aws

开发人员指南

Amazon Location Service



Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Location Service: 开发人员指南

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

Amazon Location Service 是什么 1
主要特征 1
主要优势 3
访问高质量的地理空间数据 4
无缝集成 4
拥有您的数据 4
降低运营成本 5
快速发展 5
支持的区域 5
定价模式 8
使用条款和数据属性 8
数据质量和覆盖范围 9
路径覆盖范围定义 9
地点地址定义 10
地点 Points-of-Interest (POI) 定义 10
地图数据覆盖范围定义 10
国家/地区数据质量 10
开始使用 29
首次设置账户 29
首次设置身份验证 33
选择一个 API 34
创建您的第一个 “地图和地点” 应用程序 35
创建您的第一个 Geofence and Trackers 应用程序 51
iOS 51
Android 59
身份验证 69
选择身份验证方法 69
使用 API 密钥 70
创建 API 密钥 71
调用 API 73
最佳实践 77
使用 Amazon Cognito 78
使用亚马逊 Cognito 和亚马逊定位服务 78
创建一个 Amazon Cognito 身份池 81

在 Web 中使用身份池 83
使用 IAM 83
受众 84
使用身份进行身份验证 84
使用策略管理访问 87
Amazon Location Service 如何与 IAM 协同工作 89
Amazon Location Service 如何处理未经身份验证的用户 96
基于身份的策略示例 96
故障排除 107
映射 110
特征 110
常见使用案例 111
独立地图 APIs 112
显示地图 112
地图概念 113
地图术语 113
本地化和国际化 114
地图样式 117
地图样式概述 117
AWS 地图样式 117
标准地图风格 118
单色地图风格 118
混合地图风格 119
卫星地图风格 119
标准地图风格 120
单色地图风格 129
混合地图风格 133
地图 APIs 135
映射 135
动态地图 138
静态地图 150
操作方法 155
动态地图 155
静态地图 205
管理成本和使用量 257
最佳实践 258

定价 260
配额和使用量 260
位数 262
什么是地方？ 262
特征 1
使用案例 144
地点 APIs 概览 264
地点概念 264
地点术语 265
查询和偏差 266
过滤 267
本地化和国际化 1772
联系方式和开放时间 1773
其他功能 1774
IntendedUse 1776
地方 APIs 1777
位数 1777
地理编码 1780
反向地理编码 1783
自动完成 1785
GetPlace 1787
在附近搜寻 1789
搜索文本 1792
建议 1794
操作方法 1797
使用地理编码 1797
使用 ReverseGeocode 1880
使用 “自动完成” 1900
使用 “获取地点” 1911
使用 SearchNearby 1923
使用 SearchText 1946
使用 “建议” 1995
管理成本和使用量 2023
最佳实践 2023
地点定价 2023
配额和使用量 2025

路线 2028
特征 2028
使用案例 2030
APIs 2031
路线概念 2032
术语 2032
地点（起点、目的地、航点和轨迹） 2033
时间（出发和到达） 2038
如何（出行模式、避让和排除） 2039
交通意识 2043
优化路线和航点顺序 2044
司机日程安排和通知 2045
路线 APIs 2047
计算路线 2053
计算路径矩阵 2059
计算等值线 2060
优化航点 2062
Snap to R 2064
操作方法 2065
使用 CalculateRoutes 2066
使用 CalculateIsolines 2156
使用 CalculateRouteMatrix 2171
使用 OptimizeWaypoints 2180
使用 SnapToRoads 2193
管理成本和使用量 2198
最佳实践 2198
航线定价 2200
路线配额和使用情况 2202
地理围栏 2205
使用案例 2206
地理围栏概念 2207
常用术语 2207
开始使用 2207
操作方法 2208
根据地理围栏评估设备位置 2209
通过以下方式对事件做出反应 EventBridge 2210

管理 资源 2216
管理成本和使用量 2223
定价 2223
配额和使用量 2224
跟踪器 2226
特征 2227
使用案例 2228
追踪器概念 2229
常用术语 2230
开始使用 2231
创建跟踪器 2231
创建未经身份验证的身份池 2236
使用设备位置更新跟踪器 2238
获取设备的位置记录 2240
列出您的设备位置 2241
操作方法 2244
验证位置 2244
链接到地理围栏集合 2246
使用 AWS IoT 和 MQTT 进行跟踪 2247
管理追踪器 2255
管理成本和使用量 2259
定价 2259
配额和使用量 2260
开发人员工具 2263
SDKs 和框架 2263
开发者教程 2264
SDKs 按语言分类 2291
按语言划分的地图渲染 SDK 2294
API 和 CLI 参考 2296
API 2296
CLI 2296
示例和学习资源 2299
演示 2299
样本 2299
GitHub 2300
与集成 AWS 2301

资源管理 2302
管理配额 2303
使用标签管理资源 2316
管理账单和费用 2319
使用创建资源 AWS CloudFormation 2319
监控和审计 2320
使用 Amazon 进行监控 CloudWatch 2321
使用监控和登录 AWS CloudTrail 2327
最佳实践 2333
资源管理 2333
账单和成本管理 2333
配额和使用量 2333
安全性 2335
数据保护 2335
数据隐私 2336
数据留存 2336
静态数据加密 2337
传输中数据加密 2349
事件响应 2349
日志记录和监控 2349
合规性验证 2350
恢复能力 2351
基础结构安全性 2351
AWS PrivateLink 适用于 Amazon 地点 2351
Amazon Location Service 的亚马逊 VPC 终端节点类型 2352
使用 Amazon Location AWS PrivateLink Service 时的注意事项 2353
为 Amazon Location Service 创建接口终端节点 2353
从亚马逊定位接口终端节点访问亚马逊定位 API 操作 2355
更新本地 DNS 配置 2355
为亚马逊位置创建亚马逊 VPC 终端节点策略 2356
配置和漏洞分析 2356
防范混淆代理问题 2356
最佳实践 2357
安全性 2357
文档历史记录 2360
AWS 词汇表 2361

..... mmccclxii

Amazon Location Service 是什么

Amazon Location Service 是一项完全托管的服务，可让您在应用程序中轻松添加定位功能。Amazon Location 使用高质量的地理空间数据来提供地图、地点、路线、追踪器和地理围栏等功能。除了这些功能外，Amazon Location 还可以帮助您保持对数据的控制以保护隐私和安全。它使您能够简单地使用开发人员工具访问位置数据，并通过随附的监控和管理功能更快地将应用程序迁移到生产环境。

该视频概述了 Amazon 定位服务。

亚马逊 Location Service 简介

借助 Amazon Location Service，您可以为应用程序添加强大的位置数据和功能。这包括地图、地点（搜索和地理编码）、路线、地理围栏和跟踪器等功能。Amazon Location 为基于位置的服务 (LBS) 提供覆盖全球的高质量数据。凭借其经济实惠的数据、地理围栏和追踪器功能以及内置的健康监测指标，您可以构建支持位置的复杂应用程序。

借助 Amazon Location，您可以控制组织的数据。它通过删除客户元数据和账户信息来匿名化发送给数据提供商的查询。此外，敏感的跟踪和地理围栏位置信息（例如设施、资产和人员位置）永远不会离开您的 AWS 账户。这种方法有助于保护敏感信息免受第三方侵害，保护用户隐私，并降低应用程序的安全风险。Amazon Location 还确保亚马逊和第三方都无权出售您的数据或将其用于广告。

借助 Amazon Location Service，您可以轻松无缝地将您的应用程序与其他 AWS 服务集成，从而简化开发工作流程并加快部署速度。它可以与亚马逊 AWS CloudTrail CloudWatch EventBridge、亚马逊和 AWS Identity and Access Management。借助这种集成，您可以轻松管理资源和成本，监控运行状况指标并自动响应事件，这要归功于其内置的监控、安全性和合规性功能。

Amazon 定位的主要功能

Amazon Location 提供了一套全面的功能，可增强您基于位置的应用程序和服务。本页概述了可用的关键功能，包括地图、地点、路径以及地理围栏和追踪器。利用这些功能，无论是提供实时位置情报、启用基于位置的服务，还是优化物流和运输作业，您都可以根据自己的特定用例量身打造丰富的位置感知体验。

Amazon Location 提供以下功能：

映射

Amazon Location Service Maps 可让您可视化位置信息，并且是许多基于位置的服务功能的基础。Amazon Location Service 提供动态和静态地图。

动态地图允许您使用地图图块创建交互式地图，并可以选择使用预先构建的地图样式，例如标准、单色、混合和卫星。您可以使用地图渲染引擎（例如）将动态地图内容（图块、样式、字形和精灵）拼接在一起。 MapLibre

静态地图允许您创建预渲染的非交互式地图图像，这些图像显示固定的地理区域，以便嵌入到应用程序中，无需复杂的渲染器。

有关更多信息，请参阅 映射。

位数

Amazon Location Service Places 允许您将搜索和地理编码功能集成到您的应用程序中。您可以：

- 使用地理编码（正向和反向）将地点（例如管理区域、街道或地址）转换为地理坐标，反之亦然。
- 使用地点搜索（“文本” 和 “附近”）搜索兴趣点并获取有关联系方式、接入点和营业时间的信息。
- 使用自动完成功能或建议，根据用户输入自动填充或预测地址或地点。您也可以使用 “获取地点” 按地点 ID 获取地点详情。

有关更多信息，请参阅 位数。

路线

Amazon Location Service Routes 允许您查找路线、服务区域，以及优化和分析路线。您可执行以下操作：

- 根据 up-to-date道路网络和实时交通信息估算行驶时间和距离。
- 根据业务需求的距离和时间阈值创建服务区或等值线。
- 计算多个起点和目的地的矩阵（时间和距离），并使用相同的矩阵进行路径规划。
- 将 GPS 轨迹与最近的路段对齐，以提高车辆跟踪和路线可视化的准确性。
- 找到前往多个目的地的最有效订单，也称为解决旅行推销员问题。

有关更多信息，请参阅 路线。

地理围栏

Amazon Location Service 地理围栏可让您的应用程序在设备进入或离开定义的地理边界（称为地理围栏）时进行检测和采取行动。检测到地理围栏漏洞 EventBridge 时，自动向 Amazon 发送进入或退出事件。这使您可以启动下游操作，例如向目标发送通知。

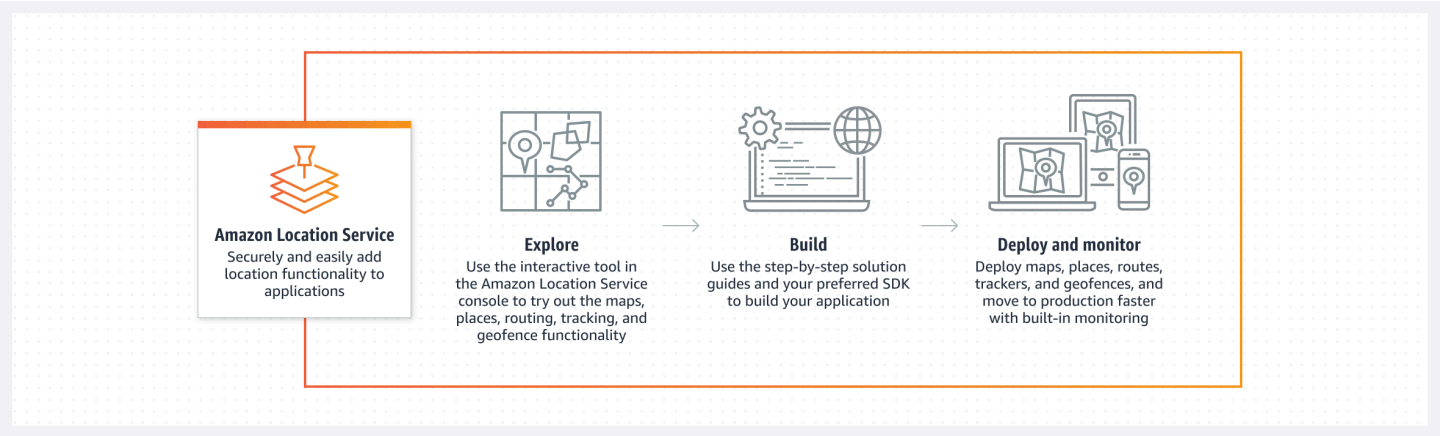
有关更多信息，请参阅 地理围栏。

跟踪器

Amazon Location Service 追踪器允许您检索正在运行支持跟踪的应用程序的设备的当前位置和历史位置。您还可以将跟踪器与 Amazon Location Service 地理围栏链接起来，根据地理围栏自动评估来自您设备的位置更新。跟踪器可以在存储或根据地理围栏进行评估之前筛选尚未移动的位置更新，从而帮助您降低成本。

有关更多信息，请参阅 跟踪器。

Amazon Location Service 的主要优势



本节概述了 Amazon Location Service 的主要优势。主要好处是：

- 访问高质量的地理空间数据-将可靠、完整 up-to-date的数据集成到应用程序中，以做出明智的决策，提供精确的见解并优化运营。
- 无缝集成-通过与 AWS 服务的无缝集成，快速部署和加快应用程序开发。Amazon Location 可与各种 AWS 服务无缝集成，从而加快开发、部署、监控和安全合规速度。
- 拥有您的数据-保护敏感的客户信息，保护用户隐私，降低应用程序的整体安全风险。该服务提供强大的数据保护措施，包括数据匿名化、加密以及与 AWS 安全服务的集成。

- 降低运营成本 ——它通过 pay-as-you-go模型提供高质量的地理空间数据，无需前期承诺或最低费用。
- 快速开发-该服务允许将地理空间数据轻松集成到应用程序中，从而加快开发周期并简化与其他 AWS 服务的集成。

访问高质量的地理空间数据

Amazon Location Service 为将地理空间功能集成到应用程序中提供了全面的解决方案，重点在于四个主要优势：无缝 AWS 集成、数据所有权和隐私、降低成本和快速开发。通过利用 AWS 基础设施和安全措施，该服务使您能够快速实施基于位置的功能，同时保持对数据的控制并降低运营成本。

无缝集成

借助 Amazon Location Service，您可以通过与 AWS 服务的无缝集成来快速部署和加快应用程序开发。Amazon Location 与一套 AWS 服务无缝集成，可简化开发工作流程，加快部署，并提供强大的监控、安全性和合规性。它与之配合使用 AWS CloudFormation 以实现一致的资源配置，Amazon CloudWatch 用于跟踪指标， AWS CloudTrail 记录账户活动，与Amazon配合使用， EventBridge 用于使用事件驱动的架构。 AWS Lambda Amazon Location 还利用 AWS Identity and Access Management (IAM) 进行安全访问控制和资源标记，以实现高效的组织和管理。这种全面的集成可以缩短开发周期，提高运营可见性，并遵守安全最佳实践。

有关更多信息，请参阅 与集成 AWS。

拥有您的数据

借助 Amazon Location Service，您可以保护用户隐私，保护您的敏感信息，并降低应用程序的安全风险。Amazon Location 提供强大的数据控制、权限和安全访问措施，确保贵组织的数据保持私密性和安全性。它可以匿名化查询，加密静态和传输中的敏感位置数据，并与 AWS Key Management Service (AWS KMS) 集成，因此您可以使用自己的加密密钥。您保留对自己数据的全部权利，没有第三方访问权限进行广告或转售。Amazon Location 还与 AWS Identity and Access Management Amazon Cognito 无缝集成，以实现安全的身份验证和访问控制，利用久经考验的安全服务 AWS 来保护您的数据和应用程序。

有关更多信息，请参阅 安全性。

降低运营成本

借助 Amazon Location Service，您可以访问来自值得信赖的数据提供商提供的经济实惠的高质量地理空间数据。Amazon Location 不需要预先承诺，也没有最低费用。您只需按实际用量付费。根据您的应用程序向服务提出的每个请求对位置数据进行计费。

参见以下案例研究：

- 数字医疗系统 Halodoc 使用亚马逊 Location Service 降低了 88% 的成本
- 网络服务提供商 Geo.me 使用亚马逊定位服务降低了 90% 的地理空间成本

快速发展

Amazon Location Service 使开发人员能够将地理空间数据和功能快速集成到他们的应用程序中。通过提供完全托管服务 APIs，Amazon Location 无需构建和维护复杂的位置数据基础设施。该服务还与 AWS CloudFormation、 CloudWatch、 CloudTrail和等 AWS 服务无缝集成 EventBridge，允许开发人员利用这些工具来构建高效的事件驱动架构。这种集成简化了开发流程，加快了基于位置的体验的原型设计、迭代和部署。借助 Amazon Location，组织可以通过地理空间功能快速增强其产品，从而减少 time-to-market并允许开发人员专注于核心产品开发。

... 由于入门和实施 Amazon Location Service 非常容易，因此我们的应用程序在 2 周内就启动并投入生产了。我们以这种速度发展的能力确保了为州交通部的审查日期做好准备。我们已经与州政府进行了审查，他们批准了我们的解决方案，我们现在正在将其推广到多个公司部门，以满足他们即将到来的州监管需求...

查尔斯·埃文斯，Alkon 司令部首席技术官

Amazon Location 支持的区域

Amazon Location Service 可在全球多个 AWS 地区使用。本页列出了当前部署和运行该服务的区域。它提供的信息可帮助您确定最适合您的应用程序和工作负载的区域。本指南涵盖了各区域的功能可用性的任何潜在变化或限制。这些信息至关重要，有助于做出有关部署位置的明智决策、确保符合数据驻留要求以及根据与最终用户或运营中心的远近优化性能。

Amazon Location 可在以下 AWS 区域使用，使用列出的终端节点。

有关更多信息，请参阅 AWS service endpoint。

区域名称	区域代码	终端节点	协议	可用版本
美国东部（俄亥俄州）	us-east-2	geo.us-east-2.amazonaws.com	HTTPS	当前和上一页
美国东部（弗吉尼亚州北部）	us-east-1	geo.us-east-1.amazonaws.com	HTTPS	当前和上一页
美国西部（俄勒冈州）	us-west-2	geo.us-west-2.amazonaws.com	HTTPS	当前和上一页
亚太地区（孟买）	ap-south-1	geo.ap-south-1.amazonaws.com	HTTPS	当前和上一页
亚太地区（新加坡）	ap-southeast-1	geo.ap-southeast-1.amazonaws.com	HTTPS	旧版
亚太地区（悉尼）	ap-southeast-2	geo.ap-southeast-2.amazonaws.com	HTTPS	当前和上一页
亚太地区（马来西亚）	ap-southeast-5	geo.ap-southeast-5.amazonaws.com	HTTPS	旧版
亚太地区（东京）	ap-northeast-1	geo.ap-northeast-1.amazonaws.com	HTTPS	当前和上一页

区域名称	区域代码	终端节点	协议	可用版本
加拿大（中部）	ca-central-1	geo.ca-central-1.amazonaws.com	HTTPS	当前和上一页
欧洲（法兰克福）	eu-central-1	geo.eu-central-1.amazonaws.com	HTTPS	当前和上一页
欧洲地区（爱尔兰）	eu-west-1	geo.eu-west-1.amazonaws.com	HTTPS	当前和上一页
欧洲（伦敦）	eu-west-2	geo.eu-west-2.amazonaws.com	HTTPS	当前和上一页
欧洲（西班牙）	eu-south-2	geo.eu-south-2.amazonaws.com	HTTPS	当前和上一页
欧洲地区（斯德哥尔摩）	eu-north-1	geo.eu-north-1.amazonaws.com	HTTPS	当前和上一页
南美洲（圣保罗）	sa-east-1	geo.sa-east-1.amazonaws.com	HTTPS	当前和上一页
AWS GovCloud（美国西部）	us-gov-west-1	地理。 us-gov-west-1.amazonaws.com geo-fips。 us-gov-west-1.amazonaws.com	HTTPS	旧版

定价模式

Amazon Location Service 不需要预先承诺，也没有最低费用。您只需按实际用量付费。Amazon Location 为前三个月的使用提供免费试用（适用使用配额）。根据您的应用程序向服务提出的每个请求对位置数据进行计费。除了 Amazon Location Service 免费试用之外，您还需要为应用程序向该服务提出的请求付费。如需了解更多定价信息，请访问我们的定价页面。

定价桶

每个 API 可能以不同的价位提供多种功能，并且价格可能因您通过请求参数请求的功能而异。

例如，对于`CalculateRoutes`，在使用、或`Truck`模式时 `CarPedestrian`，将按核心价格区间向您收费。使用其他出行模式（例如`Scooter`模式）时，将按高级价格区间向您收费。当您请求计算通行费成本时，将按高级价格段向您收费。

将向您收取费用的特定定价区间将在`PricingBucket`响应字段中返回。要查看每个 API 和功能的定价，请参阅 Amazon Location Servic e 定价页面，了解有关 API 功能、请求参数和相应定价的详细信息。有关更多信息，请参阅以下主题：

- the section called “定价”
- the section called “地点定价”
- the section called “航线定价”
- the section called “定价”
- the section called “定价”

使用条款和数据属性

在使用 Amazon Location Service 之前，请务必遵守所有适用的法律要求，包括适用于使用提供商的许可条款。

- 有关 AWS 要求的更多信息，请参阅《AWS 服务条款》第 82 节。
- 有关更多信息 SLAs，请参阅 Amazon 营业地点 SLA。

Amazon Location Service 利用来自多个来源的数据，如下所示：

- HERE Technologies 允许 AWS 客户访问底图、地理编码、反向地理编码和街道地址数据。HERE提供的地图会根据平均每天500万次更新不断进行验证，以提供900多个地图属性并保持准确的信息。

点击此处了解更多信息。

- GrabMaps 为 AWS 客户提供各种样式的地图图块、地理编码功能、反向地理编码功能和路径选项。如今，GrabMaps 它每月为八个东南亚国家的8000多亿次搜索、建议、旅行和趋势提供支持。

  进一步了解 GrabMaps。

- Esri 允许 AWS 客户访问各种样式的地图图块、地理编码功能、反向地理编码功能和街道地址数据。如今，Esri的技术已在超过35万家组织中部署，其中包括90家财富100强公司。

  了解有关 Esri 的更多信息。

- AWS 允许客户访问根据日光分布 OpenStreetMap (OSM) 和其他开放数据源构建的全球底图。地图样式经过专业设计，可支持不同的应用程序和用例，包括物流和交付，以及网络和移动环境中的数据可视化。OSM 是一个开放数据项目，拥有超过一百万的地图制作者社区，贡献者每天更新超过 500,000 个要素。

  了解有关开放数据的更多信息。

使用来自 Amazon Location Service 的数据时，您应以合理显眼的方式通知您的产品和/或服务的最终用户（例如，在最终用户条款中插入参考文献），该链接突出显示了我们的数据源及其来源，以供归因。此外，默认情况下，使用 Amazon Location Service 渲染动态地图时，归因会显示在地图的右下角。

- Esri 的数据源归因
- GrabMaps'数据源归因
- 以下是数据源归因

数据质量和覆盖范围

Amazon Location Service 的地图数据覆盖全球。每个国家/地区的所有亚马逊位置地图数据质量都被归类为全面、良好或一般。路径、地点地址和地点 Points-of-Interest (POI) 数据的地图数据是分开分类的。

路径覆盖范围定义

- 全面-完全覆盖一个国家的道路网络。能够计算一个国家内所有地址之间的最佳路线。
- 良好-覆盖一个国家内的大多数道路。提供计算一个国家内所有地址之间的路线的功能。
- 公平 ——主要城市的道路和城市之间的主要道路的覆盖范围。

地点地址定义

- 全面-全面覆盖一个国家内的所有地址，并提供完整的地址详情。
- 良好-覆盖主要城市和部分城市和街道的地址。
- 公平 ——主要城市的覆盖范围。
- 不支持-不支持某个国家/地区内的地址。

地点 Points-of-Interest (POI) 定义

- 全面-覆盖业务以支持最佳搜索体验。
- 良好-尽最大努力覆盖企业。
- 公平-仅提供基本和备受瞩目的具有里程碑意义的POI信息。
- 不支持-国内不支持 POIs 。

地图数据覆盖范围定义

- 全面-包括详细 point-to-point的地图和道路网络。
- 良好-包括对 point-to-point和地点的支持。
- Fai@@ r-仅包括位置和高级公路等级几何形状。

国家/地区数据质量

国家/地区	ISO A3	ISO A2	路线覆盖范围	地点地址覆盖范围	放置 POI 覆盖范围	地图数据覆盖范围
阿富汗	AFG	AF	不支持	公平	不支持	公平
阿尔巴尼亚	ALB	AL	好	全面	好	全面
阿尔及利亚	DZA	DZ	好	全面	不支持	全面

国家/地区	ISO A3	ISO A2	路线覆盖范围	地点地址覆盖范围	放置 POI 覆盖范围	地图数据覆盖范围
美属萨摩亚	ASM	AS	公平	好	不支持	好
安道尔	AND	AD	全面	全面	全面	全面
安哥拉	AGO	AO	好	全面	不支持	全面
安圭拉岛	AIA	AI	公平	好	不支持	公平
南极洲	ATA	AQ	不支持	公平	不支持	公平
安提瓜和巴布达	ATG	AG	公平	公平	不支持	公平
阿根廷	ARG	AR	全面	全面	全面	全面
亚美尼亚	ARM	AM	公平	好	不支持	好
阿鲁巴岛	ABW	AW	公平	好	不支持	好
澳大利亚	AUS	AU	全面	全面	全面	全面
奥地利	AUT	AT	全面	全面	全面	全面
阿塞拜疆	AZE	AZ	好	全面	不支持	全面
巴哈马	BHS	BS	全面	全面	好	全面
巴林	BHR	BH	全面	全面	好	全面

国家/地区	ISO A3	ISO A2	路线覆盖范围	地点地址覆盖范围	放置 POI 覆盖范围	地图数据覆盖范围
孟加拉国	BGD	BD	好	好	不支持	全面
巴巴多斯	BRB	BB	不支持	好	不支持	好
白俄罗斯	BLR	BY	好	全面	全面	全面
比利时	BEL	BE	全面	全面	全面	全面
伯利兹	BLZ	BZ	公平	好	不支持	好
贝宁	BEN	BJ	公平	好	不支持	好
百慕大	BMU	BM	不支持	好	不支持	好
不丹	BTN	BT	不支持	公平	不支持	公平
玻利维亚	BOL	BO	好	全面	不支持	全面
波斯尼亚和黑塞哥维那	BIH	BA	好	全面	全面	全面
博茨瓦纳	BWA	BW	好	全面	好	全面
巴西	BRA	BR	好	全面	全面	全面
英属印度洋领地	IOT	IO	不支持	公平	不支持	公平

国家/地区	ISO A3	ISO A2	路线覆盖范围	地点地址覆盖范围	放置 POI 覆盖范围	地图数据覆盖范围
英国主权基地区域	GBS	GB				全面
英属维尔京群岛	VGB	VG	公平	公平	不支持	公平
文莱达鲁萨兰国	BRN	BN	好	全面	公平	全面
保加利亚	BGR	BG	全面	全面	全面	全面
布基纳法索	BFA	BF	公平	好	不支持	好
布隆迪	BDI	BI	公平	好	不支持	好
柬埔寨	KHM	KH	好	好	公平	全面
喀麦隆	CMR	CM	好	全面	不支持	全面
加拿大	CAN	CA	全面	全面	全面	全面
佛得角	CPV	CV	公平	好	不支持	好
开曼群岛	CYM	KY	全面	全面	好	全面
中非共和国	CAF	CF	公平	好	不支持	好
乍得	TCD	TD	公平	好	不支持	好
智利	CHL	CL	全面	全面	全面	全面

国家/地区	ISO A3	ISO A2	路线覆盖范围	地点地址覆盖范围	放置 POI 覆盖范围	地图数据覆盖范围
中国	CHN	CN	公平	公平	不支持	公平
圣诞岛	CXR	CX				全面
科科斯（基林）群岛	CCK	CC				全面
哥伦比亚	COL	CO	全面	全面	公平	全面
科摩罗	COM	KM	公平	好	不支持	好
刚果	COG	CG	公平	好	不支持	好
刚果，民主共和国	COD	CD	公平	好	不支持	好
库克群岛	COK	CK	不支持	公平	不支持	公平
哥斯达黎加	CRI	CR	好	全面	好	全面
克罗地亚	HRV	HR	全面	全面	全面	全面
古巴	CUB	CU	不支持	好	不支持	好
库拉索	CUW	CW	公平	好	不支持	公平
塞浦路斯	CYP	CY	全面	全面	不支持	全面

国家/地区	ISO A3	ISO A2	路线覆盖范围	地点地址覆盖范围	放置 POI 覆盖范围	地图数据覆盖范围
塞浦路斯联合国中立区						全面
捷克	CZE	CZ	全面	全面	全面	全面
丹麦	DNK	DK	全面	全面	全面	全面
吉布提	DJI	DJ	公平	好	不支持	好
多米尼加	DMA	DM	公平	公平	不支持	公平
多米尼加共和国	DOM	DO	好	全面	公平	全面
厄瓜多尔	ECU	EC	好	好	不支持	好
埃及	EGY	EG	公平	全面	公平	全面
萨尔瓦多	SLV	SV	公平	好	不支持	好
赤道几内亚	GNQ	GQ	公平	好	不支持	好
厄立特里亚	ERI	ER	公平	好	不支持	好
爱沙尼亚	EST	EE	全面	全面	全面	全面
史瓦帝尼	SWZ	SZ	好	全面	好	全面

国家/地区	ISO A3	ISO A2	路线覆盖范围	地点地址覆盖范围	放置 POI 覆盖范围	地图数据覆盖范围
埃塞俄比亚	ETH	ET	好	全面	不支持	全面
福克兰群岛	FLK	FK	不支持	全面	不支持	全面
法罗群岛	FRO	FO	全面	全面	不支持	全面
斐济	FJI	FJ	不支持	好	不支持	好
芬兰	FIN	FI	全面	全面	全面	全面
法国	FRA	FR	全面	全面	全面	全面
法属圭亚那	GUF	GF	好	全面	好	全面
法属玻里尼西亚	PYF	PF	公平	好	不支持	好
加蓬	GAB	GA	公平	好	不支持	好
冈比亚	GMB	GM	公平	好	不支持	好
加沙地带	PSE	PS				好
格鲁吉亚	GEO	GE	公平	好	不支持	好
德国	DEU	DE	全面	全面	全面	全面
加纳	GHA	GH	好	全面	不支持	全面

国家/地区	ISO A3	ISO A2	路线覆盖范围	地点地址覆盖范围	放置 POI 覆盖范围	地图数据覆盖范围
直布罗陀	GIB	GI	全面	全面	全面	全面
希腊	GRC	GR	全面	全面	全面	全面
格陵兰	GRL	GL	不支持	公平	不支持	公平
格林纳达	GRD	GD	公平	公平	不支持	公平
瓜德罗普	GLP	GP	好	全面	公平	全面
关岛	牙龈	GU	公平	好	好	好
危地马拉	GTM	GT	好	全面	不支持	全面
根西岛	GGY	GG	全面	全面	全面	全面
几内亚	GIN	GN	公平	好	不支持	好
几内亚比绍	GNB	GW	公平	好	不支持	好
圭亚那	GUY	GY	公平	好	不支持	好
海地	HTI	HT	公平	好	不支持	好
洪都拉斯	HND	HN	公平	好	不支持	好
中国香港	HKG	HK	全面	全面	公平	全面
匈牙利	HUN	HU	全面	全面	全面	全面
冰岛	ISL	IS	全面	全面	全面	全面

国家/地区	ISO A3	ISO A2	路线覆盖范围	地点地址覆盖范围	放置 POI 覆盖范围	地图数据覆盖范围
印度	IND	IN	好	全面	好	全面
印度尼西亚	IDN	ID	好	全面	好	全面
伊朗	IRN	IR	不支持	公平	不支持	公平
伊拉克	IRQ	IQ	公平	好	不支持	好
爱尔兰	IRL	IE	全面	全面	全面	全面
马恩岛	IN	IM	全面	全面	好	全面
以色列	ISR	IL	全面	全面	公平	全面
意大利	ITA	IT	全面	全面	全面	全面
科特迪瓦	CIV	CI	好	全面	好	全面
牙买加	JAM	JM	好	好	不支持	好
日本	JPN	JP	全面	全面	全面	公平
泽西岛	JEY	JE	全面	全面	好	全面
约旦	JOR	JO	全面	全面	公平	全面
哈萨克	KAZ	KZ	好	全面	公平	全面
肯尼亚	KEN	KE	全面	全面	公平	全面
基里巴斯	KIR	KI	不支持	公平	不支持	公平
科索沃	KOS	XK	好	全面	不支持	全面
科威特	KWT	KW	全面	全面	公平	全面

国家/地区	ISO A3	ISO A2	路线覆盖范围	地点地址覆盖范围	放置 POI 覆盖范围	地图数据覆盖范围
吉尔吉斯斯坦	KGZ	KG	不支持	好	不支持	好
老挝	LAO	LA	不支持	公平	不支持	公平
拉脱维亚	LVA	LV	全面	全面	全面	全面
黎巴嫩	LBN	LB	全面	全面	公平	全面
莱索托	LSO	LS	好	全面	好	全面
利比里亚	LBR	LR	公平	好	不支持	好
利比亚	LBY	LY	公平	好	不支持	好
列支敦士登	LIE	LI	全面	全面	全面	全面
立陶宛	LTU	LT	全面	全面	全面	全面
卢森堡	LUX	LU	全面	全面	全面	全面
澳门	MAC	MO	全面	全面	好	全面
马达加斯加	MDG	MG	公平	好	不支持	好
马拉维	MWI	MW	好	全面	不支持	全面
马来西亚	MYS	MY	全面	全面	全面	全面
马尔代夫	MDV	MV	公平	好	不支持	好
Mali	MLI	ML	公平	好	不支持	好

国家/地区	ISO A3	ISO A2	路线覆盖范围	地点地址覆盖范围	放置 POI 覆盖范围	地图数据覆盖范围
马耳他	MLT	MT	全面	全面	全面	全面
马绍尔群岛	MHL	MH	不支持	公平	不支持	公平
马提尼克	MTQ	MQ	好	全面	不支持	全面
毛里塔尼亚	MRT	MR	公平	好	不支持	好
毛里求斯	MUS	MU	全面	全面	不支持	全面
马约特岛	MYT	YT	全面	全面	好	全面
墨西哥	MEX	MX	全面	全面	公平	全面
密克罗尼西亚联邦	FSM	FM	不支持	公平	不支持	公平
摩尔多瓦	MDA	MD	公平	全面	全面	全面
摩纳哥	MCO	MC	全面	全面	全面	全面
蒙古	MNG	MN	不支持	好	公平	好
黑山共和国	MNE	ME	好	全面	全面	全面
蒙特塞拉特岛	MSR	MS	不支持	公平	不支持	公平
摩洛哥	MAR	MA	好	全面	公平	全面

国家/地区	ISO A3	ISO A2	路线覆盖范围	地点地址覆盖范围	放置 POI 覆盖范围	地图数据覆盖范围
莫桑比克	MOZ	MZ	公平	全面	公平	全面
缅甸	MMR	MM	好	公平	不支持	全面
纳米比亚	NAM	NA	好	全面	公平	全面
瑙鲁	NRU	NR	不支持	公平	不支持	公平
尼泊尔	NPL	NP	公平	好	不支持	好
荷兰	NLD	NL	全面	全面	全面	全面
新喀里多尼亚	NCL	NC	公平	好	不支持	好
新西兰	NZL	NZ	全面	全面	全面	全面
尼加拉瓜	NIC	NI	公平	好	不支持	好
尼日尔	NER	NE	公平	好	不支持	好
尼日利亚	NGA	NG	好	全面	不支持	全面
纽埃岛	NIU	NU	不支持	公平	不支持	公平
诺福克岛	NFK	NF	好	全面	不支持	全面
朝鲜	PRK	KP	不支持	公平	不支持	公平
北马其顿	MKD	MK	好	全面	好	全面

国家/地区	ISO A3	ISO A2	路线覆盖范围	地点地址覆盖范围	放置 POI 覆盖范围	地图数据覆盖范围
北马里亚纳群岛	MNP	MP	不支持	好	不支持	好
挪威	NOR	NO	全面	全面	全面	全面
阿曼	OMN	OM	全面	全面	公平	全面
巴基斯坦	PAK	PK	不支持	公平	不支持	公平
帕劳群岛	PLW	PW	公平	好	不支持	好
巴拿马	PAN	PA	好	全面	公平	全面
巴布亚新几内亚	PNG	PG	公平	好	不支持	好
西沙群岛						全面
巴拉圭	PRY	PY	好	全面	好	全面
秘鲁	PER	PE	好	全面	好	全面
菲律宾	PHL	PH	好	全面	公平	全面
皮特凯恩	PCN	PN	不支持	公平	不支持	公平
波兰	POL	PL	全面	全面	全面	全面
葡萄牙	PRT	PT	全面	全面	全面	全面

国家/地区	ISO A3	ISO A2	路线覆盖范围	地点地址覆盖范围	放置 POI 覆盖范围	地图数据覆盖范围
波多黎各	PRI	PR	好	全面	公平	全面
卡塔尔	QAT	QA	全面	全面	公平	全面
留尼汪	REU	RE	全面	全面	不支持	全面
罗马尼亚	ROU	RO	全面	全面	好	全面
俄罗斯	RUS	RU	好	全面	好	全面
卢旺达	RWA	RW	公平	好	好	好
圣巴泰勒米	BLM	BL	全面	全面	不支持	全面
圣赫勒拿岛	SHN	SH	公平	好	不支持	好
圣基茨和尼维斯	KNA	KN	不支持	好	不支持	好
圣卢西亚岛	LCA	LC	公平	公平	不支持	公平
圣马丁	MAF	MF	全面	公平	不支持	公平
圣皮埃尔和密克隆群岛	SPM	PM	公平	公平	不支持	公平

国家/地区	ISO A3	ISO A2	路线覆盖范围	地点地址覆盖范围	放置 POI 覆盖范围	地图数据覆盖范围
圣文森特和格林纳丁斯	VCT	VC	不支持	好	不支持	好
萨摩亚群岛	WSM	WS	不支持	公平	不支持	公平
圣马力诺	SMR	SM	全面	全面	全面	全面
圣多美和普林西比	STP	ST	公平	好	不支持	好
沙特阿拉伯	SAU	SA	全面	全面	公平	全面
塞内加尔	SEN	SN	好	全面	不支持	全面
塞尔维亚	SRB	RS	好	全面	全面	全面
塞舌尔	SYC	SC	全面	全面	不支持	全面
塞拉利昂	SLE	SL	公平	好	不支持	好
新加坡	SGP	SG	全面	全面	全面	全面

国家/地区	ISO A3	ISO A2	路线覆盖范围	地点地址覆盖范围	放置 POI 覆盖范围	地图数据覆盖范围
圣尤斯特歇斯岛、萨巴岛、博奈尔岛	BES	BQ	不支持	公平	不支持	公平
荷属圣马丁	SXM	SX	全面	公平	不支持	公平
斯洛伐克	SVK	SK	全面	全面	全面	全面
斯洛文尼亚	SVN	SI	全面	全面	全面	全面
所罗门群岛	SLB	SB	不支持	公平	不支持	公平
索马里	SOM	SO	公平	好	不支持	好
南非	ZAF	ZA	全面	全面	全面	全面
南格鲁吉亚和南桑威奇是	SGS	GS				全面
韩国	KOR	KR	公平	公平	不适用	公平
南苏丹	SSD	SS	公平	好	不支持	好
西班牙	ESP	ES	全面	全面	全面	全面
南沙群岛		PG				全面

国家/地区	ISO A3	ISO A2	路线覆盖范围	地点地址覆盖范围	放置 POI 覆盖范围	地图数据覆盖范围
斯里兰卡	LKA	LK	公平	全面	不支持	全面
苏丹	SDN	SD	不支持	好	不支持	好
苏里南	SUR	SR	公平	好	不支持	好
斯瓦尔巴群岛和扬马延群岛	SJM	SJ	全面	全面	不支持	全面
瑞典	SWE	SE	全面	全面	全面	全面
瑞士	CHE	CH	全面	全面	全面	全面
Syria	SYR	SY	不支持	好	不支持	好
中国台湾	TWN	TW	全面	全面	好	全面
塔吉克斯坦	TJK	TJ	不支持	公平	不支持	公平
坦桑尼亚	TZA	TZ	好	全面	不支持	全面
泰国	THA	TH	好	全面	公平	全面
东帝汶	TLS	TL	不支持	好	不支持	好
多哥	TGO	TG	公平	好	不支持	好
托克劳	TKL	TK	不支持	公平	不支持	公平
汤加	TON	TO	不支持	好	不支持	好

国家/地区	ISO A3	ISO A2	路线覆盖范围	地点地址覆盖范围	放置 POI 覆盖范围	地图数据覆盖范围
千里达和多巴哥	TTO	TT	不支持	好	不支持	好
突尼斯	TUN	TN	好	全面	不支持	全面
土耳其	TUR	TR	好	全面	好	全面
北塞浦路斯土耳其共和国						全面
土库曼斯坦	TKM	TM	不支持	公平	不支持	公平
特克斯和凯科斯群岛	TCA	TC	不支持	公平	不支持	公平
图瓦卢	TUV	TV	不支持	公平	不支持	公平
乌干达	UGA	UG	好	全面	不支持	全面
乌克兰	UKR	UA	好	全面	好	全面
阿拉伯联合酋长国	ARE	AE	全面	全面	公平	全面
英国	GBR	GB	全面	全面	全面	全面
United States of America	USA	美国	全面	全面	全面	全面

国家/地区	ISO A3	ISO A2	路线覆盖范围	地点地址覆盖范围	放置 POI 覆盖范围	地图数据覆盖范围
乌拉圭	URY	UY	好	全面	全面	全面
美属维尔京群岛	VIR	VI	全面	全面	公平	全面
乌兹别克斯坦	UZB	UZ	公平	好	不支持	好
瓦努阿图	VUT	VU	不支持	好	不支持	好
梵蒂冈城	VAT	VA	全面	全面	全面	全面
委内瑞拉	VEN	VE	好	全面	公平	全面
越南	VNM	VN	好	全面	全面	全面
瓦利斯群岛和富图纳群岛	WLF	WF	公平	好	不支持	好
约旦河西岸						
西撒哈拉	ESH	EH	好	全面	不支持	全面
也门	YEM	YE	不支持	好	不支持	好
赞比亚	ZMB	ZM	好	全面	不支持	全面
津巴布韦	ZWE	ZW	好	全面	不支持	全面

开始使用 Amazon Location Service

本主题可帮助您开始使用 Amazon Location Service。按照以下步骤创建您的第一个应用程序，并了解如何根据常见用例选择合适的 Amazon Location API。

1. 设置您的 AWS 账户和访问权限。

   如果你还没有账户，系统会提示你创建一个账户。使用 AWS 免费套餐，您可以获得三个月的 Amazon Location Service 免费套餐。

   如果您已经有一个账户，则需要提供亚马逊定位服务（Amazon Location Service）的访问权限。

   继续the section called “首次设置账户”讨论这个话题。
2. 了解并设置身份验证。

   要使用 Amazon Location，必须向用户授予访问 APIs 构成亚马逊位置的资源的权限。API 密钥、Amazon Cognito 和 AWS Identity and Access Management (IAM) 是授予对资源和操作访问权限的三个选项 (APIs)。

   继续身份验证讨论这个话题。
3. 创建您的第一个定位应用程序。

   请参见the section called “创建您的第一个 “地图和地点” 应用程序”构建您的第一个 “Hello World” 应用程序。
4. 选择正确的 API。

   Amazon Location 提供了丰富的地点、路线、地图、地理围栏和追踪器，可解决各种业务用例。APIs

   要详细了解如何选择 Amazon Location API，请参阅了解the section called “选择一个 API”更多信息。

设置您的 账户

本部分介绍您需要执行哪些操作才能使用 Amazon Location Service。对于想要使用 Amazon Location 的用户，您必须拥有 AWS 账户 并已设置访问 Amazon Location 的权限。

我是新手 AWS

注册获取 AWS 账户

如果您没有 AWS 账户，请完成以下步骤来创建一个。

报名参加 AWS 账户

1. 打开https://portal.aws.amazon.com/billing/注册。
2. 按照屏幕上的说明操作。

   在注册时，将接到电话或收到短信，要求使用电话键盘输入一个验证码。

   当您注册时 AWS 账户，就会创建AWS 账户根用户一个。根用户有权访问该账户中的所有 AWS 服务 和资源。作为最佳安全实践，请为用户分配管理访问权限，并且只使用根用户来执行需要根用户访问权限的任务。

AWS 注册过程完成后会向您发送一封确认电子邮件。您可以随时前往 https://aws.amazon.com/并选择“我的账户”，查看您当前的账户活动并管理您的账户。

创建具有管理访问权限的用户

注册后，请保护您的安全 AWS 账户 AWS 账户根用户 AWS IAM Identity Center，启用并创建管理用户，这样您就不会使用 root 用户执行日常任务。

保护你的 AWS 账户根用户

1. 选择 Root 用户并输入您的 AWS 账户 电子邮件地址，以账户所有者的身份登录。AWS Management Console在下一页上，输入您的密码。

   要获取使用根用户登录方面的帮助，请参阅《AWS 登录 用户指南》中的 Signing in as the root user。
2. 为您的根用户启用多重身份验证（MFA）。

   有关说明，请参阅 I A M 用户指南中的为 AWS 账户 根用户（控制台）启用虚拟 MFA 设备。

创建具有管理访问权限的用户

1. 启用 IAM Identity Center。

有关说明，请参阅《AWS IAM Identity Center 用户指南》中的 Enabling AWS IAM Identity Center。

2. 在 IAM Identity Center 中，为用户授予管理访问权限。

   有关使用 IAM Identity Center 目录 作为身份源的教程，请参阅《用户指南》 IAM Identity Center 目录中的使用默认设置配置AWS IAM Identity Center 用户访问权限。

以具有管理访问权限的用户身份登录

- 要使用您的 IAM Identity Center 用户身份登录，请使用您在创建 IAM Identity Center 用户时发送到您的电子邮件地址的登录网址。

  有关使用 IAM Identity Center 用户登录的帮助，请参阅AWS 登录 用户指南中的登录 AWS 访问门户。

将访问权限分配给其他用户

1. 在 IAM Identity Center 中，创建一个权限集，该权限集遵循应用最低权限的最佳做法。

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的 Create a permission set。

2. 将用户分配到一个组，然后为该组分配单点登录访问权限。

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的 Add groups。

我已经有 AWS 账号了

授予访问亚马逊 Location Service 的权限

默认状态下，您的非管理员用户没有权限。在他们访问 Amazon Location 之前，您必须通过附加具有特定权限的 IAM policy 来授予权限。在授予对资源的访问权限时，请务必遵循最低权限原则。

> **Note**
>
> 有关向未经身份验证的用户授予访问 Amazon Location Service 功能的权限（例如，在基于 Web 的应用程序中）的信息，请参阅 身份验证。

以下示例策略授予用户访问所有 Amazon Location 操作的权限。有关更多示例，请参阅the section called “基于身份的策略示例”。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "geo:*"
        "geo-maps:*"
        "geo-places:*"
        "geo-routes:*"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

要提供访问权限，请为您的用户、组或角色添加权限：

- 中的用户和群组 AWS IAM Identity Center：

  创建权限集合。按照《AWS IAM Identity Center 用户指南》中创建权限集的说明进行操作。

- 通过身份提供商在 IAM 中托管的用户：

  创建适用于身份联合验证的角色。按照《IAM 用户指南》中针对第三方身份提供商创建角色（联合身份验证）的说明进行操作。

- IAM 用户：
  - 创建您的用户可以担任的角色。按照《IAM 用户指南》中为 IAM 用户创建角色的说明进行操作。
  - （不推荐使用）将策略直接附加到用户或将用户添加到用户组。按照《IAM 用户指南》中向用户添加权限（控制台）中的说明进行操作。

在创建使用 Amazon Location Service 的应用程序时，您可能需要一些用户拥有未经身份验证的访问权限。有关这些使用案例，请参阅使用 Amazon Cognito 启用未经身份验证的访问。

使用 Amazon Location Service 控制台进行身份验证

> **Note**
>
> 要了解有关身份验证的更多信息，请参阅身份验证。

要使用 Amazon Location Service，必须向用户授予访问 APIs 构成亚马逊位置的资源的权限。默认情况下，Amazon 地址 APIs 需要进行身份验证才能使用。您可以使用 Amazon Cognito 或 API 密钥为匿名用户提供身份验证和授权。

the section called “创建您的第一个 “地图和地点” 应用程序”在本教程中，该应用程序具有匿名使用方式，这意味着您的用户无需登录。在本教程中，您将创建用于示例应用程序的 API 密钥。

按照以下步骤创建您的第一个 API 密钥。

1. 在 Amazon Location 控制台中，从左侧菜单中选择 API 密钥。
2. 在 API 密钥页面上，选择创建 API 密钥。
3. 在创建 API 密钥页面中，填写以下信息：
   - 名称——您的 API 密钥的名称，例如 `MyHelloWorldApp`。
   - 描述 – API 密钥的可选描述。
   - 操作——指定您要使用此 API 密钥授权的操作。您必须至少选择 Geo-maps: get* 和 geo-places: Search*。
   - 到期时间 –（可选）添加 API 密钥的到期日期和时间。有关更多信息，请参阅 the section called “最佳实践”。
   - 引用站点 –（可选）添加一个或多个可以使用 API 密钥的域名。例如，如果 API 密钥是为了允许应用程序在网站上运行`example.com`，那么你可以将其设置`*.example.com/`为允许的反向链接。
   - 标签 –（可选）向 API 密钥添加标签。

> **Important**
>
> 我们建议您通过设置到期时间或反向链接（如果不是两者兼而有之）来保护 API 密钥的使用。

4. 选择创建 API 密钥以创建 API 密钥。
5. 在 API 密钥的详情页面上，您可以看到有关您创建的 API 密钥的信息。

   选择“显示 API 密钥”，然后复制密钥值以供本the section called “创建您的第一个 “地图和地点” 应用程序”教程的后面部分使用。键值的格式为 `v1.public.a1b2c3d4...`。

选择正确的 API

本主题可帮助您根据您可能希望使用基于位置的数据和服务解决的常见用例来选择 Amazon Location Service API。

映射

地图为各种应用程序提供了对动态和静态地图类型的访问。有关更多信息，请参阅映射。

- 动态地图：可以实时自定义的交互式地图，允许用户平移、缩放和叠加数据。有关更多信息，请参阅the section called “动态地图”。
- 静态地图：不带交互元素的显示特定位置或路线的地图的静态图像，适用于交互性有限的应用程序。有关更多信息，请参阅the section called “静态地图”。

路线

路径提供了计算位置之间优化路径的功能。这些功能支持需要物流规划、距离计算和路线优化的应用程序。用户还可以将位置点与道路对齐，以提高准确性。有关更多信息，请参阅路线。

- CalculateIsolines：根据行驶时间或距离生成等值线，可用于定义服务区或可达区域。有关更多信息，请参阅the section called “计算等值线”。
- CalculateRouteMatrix：提供多个出发地和目的地之间的距离和行程时间的矩阵，为后勤和行程计划提供支持。有关更多信息，请参阅the section called “计算路径矩阵”。
- CalculateRoutes：计算多站导航的 point-to-point优化路线，包括可自定义的路线偏好。有关更多信息，请参阅the section called “计算路线”。
- OptimizeWaypoints：优化航点顺序以获得最高效的行驶路线，从而最大限度地减少距离或时间。有关更多信息，请参阅the section called “优化航点”。
- SnapToRoads：将坐标与最近的道路对齐，通过将点捕捉到已知道路来提高 GPS 精度。有关更多信息，请参阅the section called “Snap to R”。

位数

地点使应用程序能够搜索、查找和检索有关兴趣点、地址和特定位置的详细信息。这些功能通过提供上下文和改善搜索功能中的用户体验来增强基于位置的服务。有关更多信息，请参阅位数。

- 地理编码：将地址或地名转换为地理坐标（经度、纬度），支持需要 address-to-location转换才能进行制图和空间分析的应用程序。有关更多信息，请参阅 the section called “反向地理编码”。
- 反向地理编码：将地理坐标转换为最近的地址或地名，为位置提供背景信息。有关更多信息，请参阅the section called “反向地理编码”。
- 自动完成：为用户输入的文本建议潜在的补全内容，从而提高搜索输入的效率。有关更多信息，请参阅the section called “自动完成”。
- GetPlace：检索有关指定地点的详细信息，包括地址、联系方式和营业时间等属性。有关更多信息，请参阅the section called “GetPlace”。
- SearchNearby：在给定地理点的指定半径范围内查找适合 “我附近” 搜索的地方。有关更多信息，请参阅the section called “在附近搜寻”。
- SearchText：允许基于关键字或短语的文本搜索地点或兴趣点，非常适合按名称或描述查找地点。有关更多信息，请参阅the section called “搜索文本”。
- 建议：在用户键入时提供搜索词建议，增强搜索相关性和用户体验。有关更多信息，请参阅the section called “建议”。

地理围栏

地理围栏允许应用程序定义地理边界并监控这些区域内的出入事件。功能包括创建、更新和删除地理围栏，以及配置通知或触发器，以便在被跟踪的设备跨越地理围栏边界时进行基于位置的操作。非常适合在预定义区域内进行基于距离的通知、安全监控和资产跟踪。有关更多信息，请参阅地理围栏。

跟踪器

通过跟踪，可以实时监控设备或资产的位置。功能包括添加被跟踪的设备、更新其位置数据和检索历史位置数据。跟踪器可用于管理车队、监控人员以及通过提供 up-to-date位置数据和移动模式来确保贵重资产的安全。有关更多信息，请参阅跟踪器。

创建您的第一个 Amazon 位置地图和地点应用程序

在本节中，您将使用 “地图和地点” 创建您的第一个应用程序。

先决条件：

如果您已经在the section called “首次设置身份验证”步骤中创建了 API 密钥，那就开始吧。

如果您尚未创建 API 密钥，the section called “首次设置身份验证”请在继续构建应用程序之前按照以下步骤操作。如果您有任何疑问，请参阅the section called “使用 API 密钥”和，了解the section called “支持的区域”更多信息。

Web

以下是使用 MapLibre GL JS 创建 Amazon Location Service 地图应用程序的 step-by-step教程。本指南将引导您完成设置地图、添加样式选项和启用地点搜索功能的过程。

设置初始页面

在本节中，我们将设置初始页面和文件夹结构。

添加所需的库和样式表

创建 `index.html` 文件。要渲染地图，你需要 MapLibre GL JS 和 MapLibre GL Geocoder。您将添加 MapLibre 和 Geocoder 样式表和脚本。 JavaScript

将以下代码复制并粘贴到您的`index.html`文件中。

```
<!DOCTYPE html>
<html lang="en">
<head>

    <title>Amazon Location Service - Getting Started with First Map App</title>
    <meta charset='utf-8'>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="Interactive map application using Amazon Location
 Service">

    <!--Link to MapLibre CSS and JavaScript library for map rendering and visualization
 -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/maplibre-gl@4.7.1/dist/
maplibre-gl.css" />
    <script src="https://cdn.jsdelivr.net/npm/maplibre-gl@4.7.1/dist/maplibre-gl.js"></
script>

    <!--Link to MapLibre Geocoder CSS and JavaScript library for place search and
 geocoding -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@maplibre/maplibre-gl-
geocoder@1.7.0/dist/maplibre-gl-geocoder.css" />
    <script src="https://cdn.jsdelivr.net/npm/@maplibre/maplibre-gl-geocoder@1.7.0/
dist/maplibre-gl-geocoder.js"></script>

    <!--Link to amazon-location JavaScript librarie -->
    <script src="https://cdn.jsdelivr.net/npm/@aws/amazon-location-utilities-auth-
helper@1"></script>
```

```
    <script src="https://cdn.jsdelivr.net/npm/@aws/amazon-location-client@1.2"></
script>

    <!-- Link to the first Amazon Location Map App's CSS and JavaScript -->
    <script src="utils.js"></script>
    <link rel="stylesheet" href="style.css"/>


</head>
<body>
    <main>

    </main>
    <script>
        // Step 1: Setup API Key and AWS Region
        // Step 2.1 Add maps to application
        // Step 2.2 initialize the map
        // Step 3: Add places features to application
        // Step 3.1: Get GeoPlaces instance. It will be used for addion search box and
 map click functionality
        // Step 3.2: Add search box to the map
        // Step 3.3.: Setup map click functionality
        // Add functions
    </script>
</body>
</html>
```

创建地图容器

在 HTML 文件的<body>元素下，在 HTML 中创建一个用于保存地图的 元素。你可以在 CSS 中设置其样式，以根据应用程序的需要设置尺寸。您必须从我们的 GitHub 存储库中下载 CSS 文件。style.css这将帮助你专注于业务逻辑。

将style.css和index.html文件保存在同一个文件夹中。

从中下载style.css文件GitHub。

```
<main role="main" aria-label="Map Container">
    <div id="map"></div>
</main>
```

添加 API 密钥和 AWS 区域详情

将您在中the section called “使用 API 密钥”创建的 API 密钥以及创建密钥的 AWS 区域添加到此文件中。

```
<!DOCTYPE html>
<html lang="en">
.....
.....
<body>
    <main role="main" aria-label="Map Container">
        <div id="map"></div>
    </main>
    <script>
        // Step 1: Setup API Key and AWS Region
        const API_KEY = "Your_API_Key";
        const AWS_REGION = "Region_where_you_created_API_Key";
        // Step 2: Add maps to application
            // Step 2.1 initialize the map
            // Step 2.2 Add navigation controls to the map
        // Step 3: Add places feature to application
            // Step 3.1: Get GeoPlaces instance. It will be used for addion search box
 and map click functionality
            // Step 3.2: Add search box to the map
            // Step 3.3.: Setup map click functionality
    </script>
</body>
</html>
```

将 Map 添加到您的应用程序

在本节中，我们将向应用程序添加地图功能。在开始之前，您的文件应位于此文件夹结构中。

如果尚未这样做，请从中下载该`style.css`文件GitHub。

```
|---FirstApp [Folder]
|-------------- index.html [File]
|-------------- style.css [File]
```

创建用于初始化地图的函数

要设置地图，请在行后创建以下函数`//Add functions。initializeMap(...)`

选择初始中心位置和缩放级别。在本示例中，我们将地图中心设置为加拿大温哥华，缩放级别为 10。添加导航控件便于缩放。

```
      /**
       * Initializes the map with the specified style and color scheme.
       */
      function initializeMap(mapStyle = "Standard", colorScheme = "Dark") {
          const styleUrl = `https://maps.geo.${AWS_REGION}.amazonaws.com/v2/styles/
${mapStyle}/descriptor?key=${API_KEY}&color-scheme=${colorScheme}`;
          const map = new maplibregl.Map({
              container: 'map',                // The ID of the map container
              style: styleUrl,                 // The style URL for the map
              center: [-123.116226, 49.246292], // Starting center coordinates
              zoom: 10,                        // Initial zoom level
              validateStyle: false             // Disable style validation
          });
          return map;                          // Return the initialized map
      }
```

初始化地图

`initializeMap(...)`致电初始化地图。或者，你可以在`initializeMap`函数之后用你喜欢的样式和配色方案对其进行初始化。有关更多样式选项，请参阅the section called “地图样式”。

```
      // Step 1: Setup API Key and AWS Region
      const API_KEY = "Your_API_Key";
      const AWS_REGION = "Region_where_you_created_API_Key";

      // Step 2.1 Add maps to application
      // Step 2.2 initialize the map
      const map = initializeMap("Standard","Light");

      // Step 3: Add places features to application
```

在浏览器`index.html`中打开即可查看地图的实际运行情况。

添加导航控件

或者，您可以向地图添加导航控件（缩放和旋转）。这应该在致电后完成`initializeMap(...)`。

```
        // Step 2.1 initialize the map
        const map = initializeMap("Standard","Light");

        // Step 2.2 Add navigation controls to the map
        map.addControl(new maplibregl.NavigationControl());

        // Step 3: Add places features to application
```

查看地图代码

恭喜您！您的第一个应用程序已准备好使用地图。index.html在浏览器中打开。确保与style.css位于同一个文件夹中index.html。

你的最终HTML应该是这样的：

```
        <!DOCTYPE html>
        <html lang="en">
        <head>

           <title>Amazon Location Service - Getting Started with First Map App</title>
           <meta charset='utf-8'>
           <meta name="viewport" content="width=device-width, initial-scale=1">
           <meta name="description" content="Interactive map application using Amazon
 Location Service">

           <!-- Link to MapLibre CSS and JavaScript library for map rendering and
 visualization -->
           <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/maplibre-gl@4.7.1/
dist/maplibre-gl.css" />
           <script src="https://cdn.jsdelivr.net/npm/maplibre-gl@4.7.1/dist/maplibre-
gl.js"></script>

           <!-- Link to MapLibre Geocoder CSS and JavaScript library for place search
 and geocoding -->
           <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@maplibre/
maplibre-gl-geocoder@1.7.0/dist/maplibre-gl-geocoder.css" />
           <script src="https://cdn.jsdelivr.net/npm/@maplibre/maplibre-gl-
geocoder@1.7.0/dist/maplibre-gl-geocoder.js"></script>

           <!-- Link to amazon-location JavaScript library -->
```

```
            <script src="https://cdn.jsdelivr.net/npm/@aws/amazon-location-utilities-
auth-helper@1"></script>
            <script src="https://cdn.jsdelivr.net/npm/@aws/amazon-location-
client@1.2"></script>

            <!-- Link to the first Amazon Location Map App's CSS and JavaScript -->
            <script src="utils.js"></script>
            <link rel="stylesheet" href="style.css"/>
        </head>

        <body>
            <main role="main" aria-label="Map Container">
                <div id="map"></div>
            </main>
            <script>
                const API_KEY = "Your_API_Key";
                const AWS_REGION = "Region_where_you_created_API_Key";

                function initializeMap(mapStyle, colorScheme) {
                    const styleUrl = `https://maps.geo.${AWS_REGION}.amazonaws.com/v2/
styles/${mapStyle}/descriptor?key=${API_KEY}&color-scheme=${colorScheme}`;

                    const map = new maplibregl.Map({
                        container: 'map',                // ID of the HTML element for
 the map
                        style: styleUrl,                 // URL for the map style
                        center: [-123.116226, 49.246292], // Initial map center
 [longitude, latitude]
                        zoom: 10                         // Initial zoom level
                    });
                    map.addControl(new maplibregl.NavigationControl());
                    return map;
                }

                const map = initializeMap("Standard", "Light");

            </script>
        </body>
        </html>
```

向您的应用程序添加地点

在本节中，我们将为应用程序设置添加地点功能。从下载 JavaScript 文件 GitHub，utils.js。

在开始之前，您的文件应采用以下文件夹结构：

```
|---FirstApp [Folder]
|-------------- index.html [File]
|-------------- style.css [File]
|-------------- utils.js [File]
```

创建要创建的函数 GeoPlaces

要添加搜索功能，请使用AuthHelper和初始化GeoPlaces类AmazonLocationClient。在中的</script>标签前添加以下getGeoPlaces(map)函数index.html。

```
       /**
        * Gets a GeoPlaces instance for Places operations.
        */
       function getGeoPlaces(map) {
           const authHelper = amazonLocationClient.withAPIKey(API_KEY, AWS_REGION);
                  // Authenticate using the API key and AWS region
           const locationClient = new
amazonLocationClient.GeoPlacesClient(authHelper.getClientConfig()); // Create a
GeoPlaces client
           const geoPlaces = new GeoPlaces(locationClient, map);
                   // Create GeoPlaces instance
           return geoPlaces;
                   // Return the GeoPlaces instance
       }
```

创建向应用程序添加搜索框的函数

在</script>标签前添加以下addSearchBox(map, geoPlaces)renderPopup(feature)、和createPopup(feature)函数index.html，以完成搜索功能设置。

```
       /**
        * Adds search box to the map.
        */
       function addSearchBox(map, geoPlaces) {
           const searchBox = new MaplibreGeocoder(geoPlaces, {
               maplibregl,
               showResultsWhileTyping: true,                    // Show results while
typing
```

```
                debounceSearch: 300,                                // Debounce search
 requests
                limit: 30,                                          // Limit number of
 results
                popuprender: renderPopup,                           // Function to render
 popup
                reverseGeocode: true,                               // Enable reverse
 geocoding
                zoom: 14,                                           // Zoom level on
 result selection
                placeholder: "Search text or nearby (lat,long)"  // Placeholder text
 for search box.
            });

            // Add the search box to the map
            map.addControl(searchBox, 'top-left');

            // Event listener for when a search result is selected
            searchBox.on('result', async (event) => {
                const { id, result_type } = event.result;                    // Get
 result ID and type
                if (result_type === "Place") {                               // Check
 if the result is a place
                    const placeResults = await geoPlaces.searchByPlaceId(id); // Fetch
 details for the selected place
                    if (placeResults.features.length) {
                        createPopup(placeResults.features[0]).addTo(map);     // Create
 and add popup for the place
                    }
                }
            });
        }

        /**
         * Renders the popup content for a given feature.
         */
        function renderPopup(feature) {
            return `
                <div class="popup-content">
                    <span class="${feature.place_type.toLowerCase()} badge">
${feature.place_type}</span><br>
                    ${feature.place_name}
                </div>`;
        }
```

```
        /**
         * Creates a popup for a given feature and sets its position.
         */
        function createPopup(feature) {
            return new maplibregl.Popup({ offset: 30 })      // Create a new popup
                .setLngLat(feature.geometry.coordinates)     // Set the popup position
                .setHTML(renderPopup(feature));              // Set the popup content
        }
```

在应用程序中添加搜索框

按照第 3.1 节中的getGeoPlaces(map)定义调用addSearchBox(map, geoPlaces)来创建GeoPlaces对象，然后调用将搜索框添加到应用程序中。

```
        // Step 2: Add maps to application
        // Step 2.1 initialize the map
        const map = initializeMap("Standard","Light");
        // Step 2.2 Add navigation controls to the map
        map.addControl(new maplibregl.NavigationControl());

        // Step 3: Add places feature to application
        // Step 3.1: Get GeoPlaces instance. It will be used for adding search box and
map click functionality
        const geoPlaces = getGeoPlaces(map);
        // Step 3.2: Add search box to the map
        addSearchBox(map, geoPlaces);
```

您的地点搜索已准备就绪。index.html在浏览器中打开即可查看其运行情况。

添加功能以在用户点击地图时显示弹出窗口

创建addMapClick(map, geoPlaces)用于在用户点击地图时显示弹出窗口的函数。在</script>标签前添加此函数。

```
        /**
         * Sets up reverse geocoding on map click events.
         */
```

```
       function addMapClick(map, geoPlaces) {
           map.on('click', async ({ lngLat }) => {                    // Listen for
click events on the map
               const response = await geoPlaces.reverseGeocode({ query: [lngLat.lng,
lngLat.lat], limit: 1, click: true }); // Perform reverse geocoding

               if (response.features.length) {                        // If there are
results
                   const clickMarker = new maplibregl.Marker({ color: "orange" }); //
Create a marker
                   const feature = response.features[0];              // Get the
clicked feature
                   const clickedPopup = createPopup(feature);         // Create popup
for the clicked feature
                   clickMarker.setLngLat(feature.geometry.coordinates) // Set marker
position
                       .setPopup(clickedPopup)                        // Attach popup
to marker
                       .addTo(map);                                   // Add marker
to the map

                   clickedPopup.on('close', () => clickMarker.remove()).addTo(map); //
Remove marker when popup is closed
               }
           });
       }
```

调用函数添加地图点击功能

要启用地图点击操作，请在包含地图的行addMapClick(map, geoPlaces)之后调用addSearchBox(map, geoPlaces)。

```
       // Step 3: Add places feature to application
       // Step 3.1: Get GeoPlaces instance. It will be used for adding search box and
map click functionality
       const geoPlaces = getGeoPlaces(map);
       // Step 3.2: Add search box to the map
       addSearchBox(map, geoPlaces);
       // Step 3.3: Setup map click functionality
       addMapClick(map, geoPlaces);
```

查看 “地图和地点” 应用程序

恭喜您！您的第一个应用程序已准备就绪，可以使用 “地图和地点” 了。index.html在浏览器中打开。确保style.css和utils.js位于同一个文件夹中index.html。

你的最终HTML应该是这样的：

```
<!DOCTYPE html>
<html lang="en">
<head>

   <title>Amazon Location Service - Getting Started with First Map App</title>
    <meta charset='utf-8'>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="Interactive map application using Amazon Location
 Service">

    <!--Link to MapLibre CSS and JavaScript library for map rendering and visualization
 -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/maplibre-gl@4.7.1/dist/
maplibre-gl.css" />
    <script src="https://cdn.jsdelivr.net/npm/maplibre-gl@4.7.1/dist/maplibre-gl.js"></
script>

    <!--Link to MapLibre Geocoder CSS and JavaScript library for place search and
 geocoding -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@maplibre/maplibre-gl-
geocoder@1.7.0/dist/maplibre-gl-geocoder.css" />
    <script src="https://cdn.jsdelivr.net/npm/@maplibre/maplibre-gl-geocoder@1.7.0/
dist/maplibre-gl-geocoder.js"></script>

    <!--Link to amazon-location JavaScript librarie -->
    <script src="https://cdn.jsdelivr.net/npm/@aws/amazon-location-utilities-auth-
helper@1"></script>
    <script src="https://cdn.jsdelivr.net/npm/@aws/amazon-location-client@1.2"></
script>

    <!-- Link to the first Amazon Location Map App's CSS and JavaScript -->
    <script src="utils.js"></script>
    <link rel="stylesheet" href="style.css"/>


</head>
<body>
```

```
    <main role="main" aria-label="Map Container">
        <div id="map"></div>
    </main>
    <script>
        // Step 1: Setup API Key and AWS Region
        const API_KEY = "Your_API_Key";
        const AWS_REGION = "Region_where_you_created_API_Key";


        // Step 2: Add maps to application
        // Step 2.1 initialize the map
        const map = initializeMap("Standard","Light");
        // Step 2.2 Add navigation controls to the map
        map.addControl(new maplibregl.NavigationControl());

        // Step 3: Add places feature to application
        // Step 3.1: Get GeoPlaces instance. It will be used for addion search box and
 map click functionality
        const geoPlaces =  getGeoPlaces(map);
        // Step 3.2: Add search box to the map
        addSearchBox(map, geoPlaces);
        // Step 3.3.: Setup map click functionality
        addMapClick(map, geoPlaces);



        /**
         * Functions to add maps and places feature.
         */

         /**
         * Initializes the map with the specified style and color scheme.
         */
        function initializeMap(mapStyle = "Standard", colorScheme = "Dark") {
            const styleUrl = `https://maps.geo.${AWS_REGION}.amazonaws.com/v2/styles/
${mapStyle}/descriptor?key=${API_KEY}&color-scheme=${colorScheme}`;
            const map = new maplibregl.Map({
                container: 'map',                   // The ID of the map container
                style: styleUrl,                    // The style URL for the map
                center: [-123.116226, 49.246292],   // Starting center coordinates
                zoom: 10,                           // Initial zoom level
                validateStyle: false                // Disable style validation
            });
            return map;                             // Return the initialized map
```

```
        }

        /**
         * Gets a GeoPlaces instance for Places operations.
         */
        function getGeoPlaces(map) {
            const authHelper =  amazonLocationClient.withAPIKey(API_KEY, AWS_REGION);
                    // Authenticate using the API key and AWS region
            const locationClient = new
amazonLocationClient.GeoPlacesClient(authHelper.getClientConfig()); // Create a
GeoPlaces client
            const geoPlaces = new GeoPlaces(locationClient, map);
                    // Create GeoPlaces instance
                return geoPlaces;
                    // Return the GeoPlaces instance
        }

         /**
         * Adds search box to the map.
         */

        function addSearchBox(map, geoPlaces) {
            const searchBox = new MaplibreGeocoder(geoPlaces, {
                maplibregl,
                showResultsWhileTyping: true,                    // Show results while
typing
                debounceSearch: 300,                             // Debounce search
requests
                limit: 30,                                       // Limit number of
results
                popuprender: renderPopup,                        // Function to render
popup
                reverseGeocode: true,                            // Enable reverse
geocoding
                zoom: 14,                                        // Zoom level on
result selection
                placeholder: "Search text or nearby (lat,long)"  // Place holder text
for search box.
            });

            // Add the search box to the map
            map.addControl(searchBox, 'top-left');

            // Event listener for when a search result is selected
```

```
            searchBox.on('result', async (event) => {
                const { id, result_type } = event.result;                    // Get
 result ID and type
                if (result_type === "Place") {                               // Check
 if the result is a place
                    const placeResults = await geoPlaces.searchByPlaceId(id); // Fetch
 details for the selected place
                    if (placeResults.features.length) {
                        createPopup(placeResults.features[0]).addTo(map);    // Create
 and add popup for the place
                    }
                }
            });
        }

        /**
         * Renders the popup content for a given feature.
         */
        function renderPopup(feature) {
            return `
                <div class="popup-content">
                    <span class="${feature.place_type.toLowerCase()} badge">
${feature.place_type}</span><br>
                    ${feature.place_name}
                </div>`;
        }

        /**
         * Creates a popup for a given feature and sets its position.
         */
        function createPopup(feature) {
            return new maplibregl.Popup({ offset: 30 })      // Create a new popup
                .setLngLat(feature.geometry.coordinates)     // Set the popup position
                .setHTML(renderPopup(feature));              // Set the popup content
        }

        /**
         * Sets up reverse geocoding on map click events.
         */
        function addMapClick(map, geoPlaces) {
            map.on('click', async ({ lngLat }) => {                    // Listen for
 click events on the map
                const response = await geoPlaces.reverseGeocode({ query: [lngLat.lng,
 lngLat.lat], limit: 1, click:true }); // Perform reverse geocoding
```

```
                if (response.features.length) {                         // If there are
results
                    const clickMarker = new maplibregl.Marker({ color: "orange" }); //
Create a marker
                    const feature = response.features[0];               // Get the
clicked feature
                    const clickedPopup = createPopup(feature);          // Create popup
for the clicked feature
                    clickMarker.setLngLat(feature.geometry.coordinates) // Set marker
position
                        .setPopup(clickedPopup)                         // Attach popup
to marker
                        .addTo(map);                                    // Add marker
to the map

                    clickedPopup.on('close', () => clickMarker.remove()).addTo(map); //
Remove marker when popup is closed
                }
            });
        }

    </script>
</body>
</html>
```

探索更多

您已经完成了快速入门教程，应该对如何使用 Amazon Location Service 来构建应用程序有所了解。要了解更多关于 Amazon Location 的信息，您可以查看以下资源：

- 查询建议详情-考虑扩展`GeoPlaces`类或使用类似的方法`ReverseGeocode`来获取有关 `Suggestion` API 返回的结果的更多详细信息。
- 为您的业务需求选择合适的 API-要确定最适合您需求的亚马逊定位 API，请查看此资源：the section called “选择一个 API”。
- 查看 Amazon Location “操作方法” 指南 ——访问亚马逊定位服务开发者指南，获取教程和其他资源。
- 文档和产品信息-如需完整文档，请访问 Amazon Location Service 开发者指南。要了解有关该产品的更多信息，请前往 Amazon Location Service 产品页面。

创建您的第一个 Geofence and Trackers 应用程序

在本节中，您将创建一个演示使用 Amazon 定位地理围栏和追踪器的主要功能的应用程序。这些应用程序演示了跟踪器和地理围栏如何使用 Lambda、 AWS IoT和 Amazon Location 功能的组合进行交互。选择 iOS 或安卓平台开始使用。

在开始构建应用程序之前，请按照中的步骤授the section called “首次设置身份验证”予适当的访问权限。

继续按照您的首选操作系统执行以下步骤：

- 创建一个 iOS 应用程序
- 创建 Android 应用程序

创建一个 iOS 应用程序

按照以下步骤使用 Amazon Location Service 构建 iOS 应用程序。

从中克隆项目文件GitHub。

为您的应用程序创建 Amazon Location 资源

AWS 账户准备就绪后，您可以生成 Amazon Location Service 资源。这些资源对于执行所提供的代码片段至关重要。

> **Note**
>
> 如果您还没有创建 AWS 账户，请创建一个 AWS 账户。

首先，您需要创建 Amazon Cognito 身份池 ID，请按以下步骤进行操作：

1. 在 AWS 控制台中，导航到 Amazon Cognito 服务，然后从左侧菜单中选择身份池，然后选择创建身份池。
2. 确保选中 “访客访问权限”，然后按 “下一步” 继续。
3. 接下来，创建一个新 IAM 角色或使用现有 IAM 角色。
4. 输入身份池名称，并确保身份池可以访问您将在下一个步骤中创建的地图和追踪器的 Amazon Location（geo）资源。
5.

现在，您需要在 AWS Amazon Location 控制台中创建地图并设置地图样式，请按以下步骤操作：

1. 导航至 Amazon Location 控制台中的 “地图” 部分，然后选择 “创建地图” 以预览可用的地图样式。
2. 为新地图资源指定名称和描述。记录您分配给地图资源的名称，本教程稍后将使用该名称。
3. 选择地图样式时，请考虑地图数据提供程序。有关更多详细信息，请参阅 AWS 服务条款的第 82 节。
4. 接受 Amazon 位置条款和条件，然后选择创建地图。创建地图后，您可以通过放大、缩小或向任意方向平移来与地图进行交互。

使用 Amazon Location 控制台创建跟踪器

1. 打开 Amazon Location Service 控制台。
2. 在左侧导航窗格中，选择跟踪器。
3. 选择创建跟踪器。
4. 填写所有必填字段。
5. 在位置筛选下，选择最适合您打算如何使用跟踪器资源的选项。如果未设置位置筛选，则默认设置为 TimeBased。有关更多信息，请参阅本指南和 Amazon Location Service 追踪器 API 参考 PositionFiltering 中的追踪器。
6. 选择创建跟踪器完成操作。

创建地理围栏收藏集

创建地理围栏集合时，您可以使用控制台、API 或 CLI。以下步骤将指导您完成每个选项。

使用 Amazon Location 控制台创建地理围栏集合：

1. 打开 Amazon Location Service 控制台，网址为 https://console.aws.amazon.com/location/。
2. 在左侧导航窗格中，选择 Geofence 集合。
3. 选择创建地理围栏集合。
4. 为集合提供名称和描述。
5. 在以目标 CloudWatch 为目标的 EventBridge 规则下，您可以创建一条可选 EventBridge 规则来开始对地理围栏事件做出反应。这样，Amazon Location 就可以将事件发布到亚马逊 CloudWatch 日志。
6. 选择创建地理围栏集合。

使用 Amazon 位置创建地理围栏集合： APIs

使用 Amazon 定位地理围 APIs栏中的 CreateGeofenceCollection 操作。以下示例使用 API 请求创建名为 *GOECOLLECTION_NAME* 的地理围栏集合。

```
POST /geofencing/v0/collections
Content-type: application/json
    {
        "CollectionName": "GOECOLLECTION_NAME",
        "Description": "Geofence collection 1 for shopping center",
        "Tags": {
            "Tag1" : "Value1"
                }
    }
```

使用 AWS CLI 以下命令创建地理围栏集合：

使用 create-geofence-collection 命令。以下示例使用创建名 AWS CLI 为的地理围栏集合。*GOECOLLECTION_NAME*

```
aws location \ create-geofence-collection \
    --collection-name "GOECOLLECTION_NAME" \
    --description "Shopping center geofence collection" \
    --tags Tag1=Value1
```

将跟踪器关联到地理围栏集合

要将跟踪器链接到地理围栏集合，您可以使用控制台、API 或 CLI。以下步骤将指导您完成每个选项。

使用 Amazon Location Service 控制台将跟踪器资源链接到地理围栏集合：

1. 打开 Amazon Location 控制台。
2. 在左侧导航窗格中，选择跟踪器。
3. 在设备跟踪器下，选择目标跟踪器的名称链接。
4. 在链接的地理围栏集合下，选择链接地理围栏集合。
5. 在链接的地理围栏集合窗口中，从下拉菜单中选择一个地理围栏集合。
6. 选择链接。
7. 在您关联跟踪器资源后，它将处于活动状态。

使用 Amazon 位置将追踪器资源链接到地理围栏集合： APIs

使用 Amazon 位置追踪器 APIs中的AsssociateTrackerConsumer 操作。以下示例使用一个 API 请求，该请求使用其亚马逊资源名称 (ARN) ExampleTracker 与地理围栏集合相关联。

```
POST /tracking/v0/trackers/ExampleTracker/consumers
Content-type: application/json
        {
           "ConsumerArn": "arn:aws:geo:us-west-2:123456789012:geofence-
collection/GOECOLLECTION_NAME"
        }
```

使用 AWS CLI 命令将跟踪器资源链接到地理围栏集合：

使用 `associate-tracker-consumer` 命令。以下示例使用创建名 AWS CLI 为的地理围栏集合。*GOECOLLECTION_NAME*

```
aws location \
associate-tracker-consumer \
    --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-
collection/GOECOLLECTION_NAME" \
    --tracker-name "ExampleTracker"
```

将 AWS Lambda 与 MQTT 配合使用

创建 Lambda 函数：

要在 AWS IoT Core 和 Amazon Location Service 之间建立连接，您需要一个 AWS Lambda 函数来处理由 EventBridge CloudWatch 事件转发的消息。此函数将提取所有位置数据，将其格式化为 Amazon Location Service，然后通过 Amazon Location 跟踪器 API 提交。您可以通过 AWS Lambda 控制台创建此函数，也可以使用 AWS Command Line Interface (AWS CLI) 或 AWS Lambda APIs。要使用控制台创建 Lambda 函数，将位置更新发布到 Amazon Location，请执行以下操作：

1. 打开 AWS Lambda 控制台，网址为 https://console.aws.amazon.com/lambda/。
2. 从左侧导航窗格中，选择函数。
3. 选择创建函数，并确保选择从头开始创作。
4. 填写以下选框：
    - a 函数名
    - 在 “运行时” 选项中，选择 Node.js 16.x。

5. 选择创建函数。
6. 选择 代码 选项卡以打开编辑器。
7. 使用以下内容覆盖 index.js 中的占位符代码：

```
const AWS = require('aws-sdk')
const iot = new AWS.Iot();
exports.handler =  function(event) {
              console.log("event===>>>", JSON.stringify(event));
              var param = {
                endpointType: "iot:Data-ATS"
              };
              iot.describeEndpoint(param, function(err, data) {
                if (err) {
                  console.log("error===>>>", err, err.stack); // an error occurred
                } else {
                  var endp = data['endpointAddress'];
                  const iotdata = new AWS.IotData({endpoint: endp});
                  const trackerEvent = event["detail"]["EventType"];
                  const src = event["source"];
                  const time = event["time"];
                  const gfId = event["detail"]["GeofenceId"];
                  const resources = event["resources"][0];
                  const splitResources = resources.split(".");
                  const geofenceCollection = splitResources[splitResources.length -
 1];
                  const coordinates = event["detail"]["Position"];

                  const deviceId = event["detail"]["DeviceId"];
                  console.log("deviceId===>>>", deviceId);
                  const msg =  {
                      "trackerEventType" : trackerEvent,
                      "source" : src,
                      "eventTime" : time,
                      "geofenceId" : gfId,
                      "coordinates": coordinates,
                      "geofenceCollection": geofenceCollection
                    };
                  const params = {
                    topic: `${deviceId}/tracker`,
                    payload: JSON.stringify(msg),
                    qos: 0
                  };
                  iotdata.publish(params, function(err, data) {
```

```
                    if (err) {
                      console.log("error===>>>", err, err.stack); // an error
occurred
                    } else {
                      console.log("Ladmbda triggered===>>>", trackerEvent);  //
successful response
                    }
                  });
              }
            });
          }
```

8. 选择部署以保存您更新的函数。
9. 选择配置选项卡。
10. 在 “触发器” 部分，单击 “添加触发器”。
11. 在 “来源” 字段中选择 EventBridge （CloudWatch 事件）。
12. 选择 “现有规则” 单选选项。
13. 输入这样的规则名称 `-GEOFENCECOLLECTION_NAM AmazonLocationMonitor E`。
14. 单击 “添加” 按钮。
15. 这还将在权限选项卡中附加 “基于资源的策略声明”

MQTT 测试客户端

1. 打开 https://console.aws.amazon.com/iot/。
2. 在左侧导航窗格中，选择 MQTT 测试客户端。
3. 您将看到标题为 MQTT 测试客户端的部分，您可以在其中配置 MQTT 连接。
4. 配置必要的设置后，单击连接按钮，使用提供的参数建立与 MQTT 代理的连接。
5. 记下端点值。

连接后，您可以使用 MQTT 测试客户端界面中提供的相应输入字段订阅 MQTT 主题或向主题发布消息。接下来，您将附上 MQTT 策略：

1. 在左侧菜单的管理下，展开安全性选项，然后单击策略。
2. 单击创建策略按钮。
3. 输入策略名称。
4. 在策略文档上，选择 JSON 选项卡。

5. 复制粘贴下面显示的策略，但请务必使用您的 *REGION* 和 *ACCOUNT_ID* 更新所有元素：

```
{
    "Version": "2012-10-17",
    "Statement": [
                {
                  "Action": [
                    "iot:Connect",
                    "iot:Publish",
                    "iot:Subscribe",
                    "iot:Receive"
                  ],
                  "Resource": [
                    "arn:aws:iot:REGION:ACCOUNT_ID:client/${cognito-
identity.amazonaws.com:sub}",
                    "arn:aws:iot:REGION:ACCOUNT_ID:topic/${cognito-
identity.amazonaws.com:sub}",
                    "arn:aws:iot:REGION:ACCOUNT_ID:topicfilter/${cognito-
identity.amazonaws.com:sub}/",
                    "arn:aws:iot:REGION:ACCOUNT_ID:topic/${cognito-
identity.amazonaws.com:sub}/tracker"
                  ],
                  "Effect": "Allow"
                }
              ]
}
```

6. 选择创建按钮完成操作。

设置示例应用程序代码

为了设置示例代码，您必须安装以下工具：

- Git
- XCode 15.3 或更高版本
- iOS 模拟器 16 或更高版本

使用以下步骤设置示例应用程序代码：

1. 从这个网址克隆 git 存储库：https://github.com/aws-geospatial/amazon-location-samples-ios/tree/main/tracking-with-geofence-notifications。

2. 打开 AWSLocationSampleApp.xcodeproj 项目文件。
3. 等待包解析过程。
4. 可选：在项目导航菜单上ConfigTemplate.xcconfig，重命名为Config.xcconfig并填写以下值：

```
IDENTITY_POOL_ID = `YOUR_IDENTITY_POOL_ID`
MAP_NAME = `YOUR_MAP_NAME`
TRACKER_NAME = `YOUR_TRACKER_NAME`
WEBSOCKET_URL = `YOUR_MQTT_TEST_CLIENT_ENDPOINT`
GEOFENCE_ARN = `YOUR_GEOFENCE_COLLECTION_NAME`
```

使用示例应用程序

设置示例代码后，您现在可以在 iOS 模拟器或物理设备上运行该应用程序。

1. 构建并运行应用程序。
2. 该应用程序将要求您提供位置和通知权限。您需要允许这些权限。
3. 点击 “Cognito 配置” 按钮。
4. 如果您尚未填写 `config.xcConfig` 文件中的值，则需要使用您之前在配置屏幕中创建的资源值填写该字段。

```
IDENTITY_POOL_ID = `YOUR_IDENTITY_POOL_ID`
MAP_NAME = `YOUR_MAP_NAME`
TRACKER_NAME = `YOUR_TRACKER_NAME`
WEBSOCKET_URL = `YOUR_MQTT_TEST_CLIENT_ENDPOINT`
GEOFENCE_ARN = `YOUR_GEOFENCE_COLLECTION_NAME`
```

5. 保存配置
6. 现在，您将能够看到时间、距离和精度的筛选选项。根据需要使用这些选项。
7. 前往应用程序中的 “跟踪” 选项卡，您将看到地图和 “开始追踪” 按钮。
8. 如果您已在模拟器上安装该应用程序，则可能需要模拟位置更改。这可以在 “功能”-> “位置” 菜单选项中完成。例如，选择 “功能”-> “位置”-> “高速公路大道”。
9. 点击 “开始追踪” 按钮。此时，您应该会在地图上看到跟踪点。
10. 该应用程序还会在后台跟踪位置。因此，当您在后台移动应用程序时，它会要求您提供在后台模式下继续跟踪的权限。
11. 您可以通过点击 “停止跟踪” 按钮来停止跟踪。

创建 Android 应用程序

按照以下步骤使用 Amazon Location Service 构建 iOS 应用程序。

从中克隆项目文件GitHub。

为您的应用程序创建 Amazon Location 资源

AWS 账户准备就绪后，您可以生成 Amazon Location Service 资源。这些资源对于执行所提供的代码片段至关重要。

> **Note**
>
> 如果您还没有创建 AWS 账户，请创建一个 AWS 账户。

首先，您需要创建 Amazon Cognito 身份池 ID，请按以下步骤进行操作：

1. 在 AWS 控制台中，导航到 Amazon Cognito 服务，然后从左侧菜单中选择身份池，然后选择创建身份池。
2. 确保选中 “访客访问权限”，然后按 “下一步” 继续。
3. 接下来，创建一个新 IAM 角色或使用现有 IAM 角色。
4. 输入身份池名称，并确保身份池可以访问您将在下一步中创建的地图和追踪器的 Amazon Location（geo）资源。
5.

现在，您需要在 AWS Amazon Location 控制台中创建地图并设置地图样式，请按以下步骤操作：

1. 导航至 Amazon Location 控制台中的 “地图” 部分，然后选择 “创建地图” 以预览可用的地图样式。
2. 为新地图资源指定名称和描述。记录您分配给地图资源的名称，本教程稍后将使用该名称。
3. 选择地图样式时，请考虑地图数据提供程序。有关更多详细信息，请参阅 AWS 服务条款的第 82 节。
4. 接受 Amazon 位置条款和条件，然后选择创建地图。创建地图后，您可以通过放大、缩小或向任意方向平移来与地图进行交互。

使用 Amazon Location 控制台创建跟踪器

1. 打开 Amazon Location Service 控制台。
2. 在左侧导航窗格中，选择跟踪器。
3. 选择创建跟踪器。
4. 填写所有必填字段。
5. 在位置筛选下，选择最适合您打算如何使用跟踪器资源的选项。如果未设置位置筛选，则默认设置为 TimeBased。有关更多信息，请参阅本指南和 Amazon Location Service 追踪器 API 参考 PositionFiltering 中的追踪器。
6. 选择创建跟踪器完成操作。

创建地理围栏收藏集

创建地理围栏集合时，您可以使用控制台、API 或 CLI。以下步骤将指导您完成每个选项。

使用 Amazon Location 控制台创建地理围栏集合：

1. 打开 Amazon Location Service 控制台，网址为 https://console.aws.amazon.com/location/。
2. 在左侧导航窗格中，选择 Geofence 集合。
3. 选择创建地理围栏集合。
4. 为集合提供名称和描述。
5. 在以目标 CloudWatch 为目标的 EventBridge 规则下，您可以创建一条可选 EventBridge 规则来开始对地理围栏事件做出反应。这样，Amazon Location 就可以将事件发布到亚马逊 CloudWatch 日志。
6. 选择创建地理围栏集合。

使用 Amazon 位置创建地理围栏集合： APIs

使用 Amazon 定位地理围 APIs栏中的 CreateGeofenceCollection 操作。以下示例使用 API 请求创建名为 *GOECOLLECTION_NAME* 的地理围栏集合。

```
POST /geofencing/v0/collections
Content-type: application/json
    {
    "CollectionName": "GOECOLLECTION_NAME",
    "Description": "Geofence collection 1 for shopping center",
    "Tags": {
        "Tag1" : "Value1"
            }
```

```
    }
```

使用 AWS CLI 以下命令创建地理围栏集合：

使用 create-geofence-collection 命令。以下示例使用创建名 AWS CLI 为的地理围栏集合。*GOECOLLECTION_NAME*

```
aws location \ create-geofence-collection \
    --collection-name "GOECOLLECTION_NAME" \
    --description "Shopping center geofence collection" \
    --tags Tag1=Value1
```

将跟踪器关联到地理围栏集合

要将跟踪器链接到地理围栏集合，您可以使用控制台、API 或 CLI。以下步骤将指导您完成每个选项。

使用 Amazon Location Service 控制台将跟踪器资源链接到地理围栏集合：

1. 打开 Amazon Location 控制台。
2. 在左侧导航窗格中，选择跟踪器。
3. 在设备跟踪器下，选择目标跟踪器的名称链接。
4. 在链接的地理围栏集合下，选择链接地理围栏集合。
5. 在链接的地理围栏集合窗口中，从下拉菜单中选择一个地理围栏集合。
6. 选择链接。
7. 在您关联跟踪器资源后，它将处于活动状态。

使用 Amazon 位置将追踪器资源链接到地理围栏集合： APIs

使用 Amazon 位置追踪器 APIs中的AsssociateTrackerConsumer 操作。以下示例使用一个 API 请求，该请求使用其亚马逊资源名称 (ARN) ExampleTracker 与地理围栏集合相关联。

```
POST /tracking/v0/trackers/ExampleTracker/consumers
Content-type: application/json
    {
    "ConsumerArn": "arn:aws:geo:us-west-2:123456789012:geofence-
collection/GOECOLLECTION_NAME"
    }
```

使用 AWS CLI 命令将跟踪器资源链接到地理围栏集合：

使用 `associate-tracker-consumer` 命令。以下示例使用创建名 AWS CLI 为的地理围栏集合。*GOECOLLECTION_NAME*

```
aws location \
associate-tracker-consumer \
    --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-
collection/GOECOLLECTION_NAME" \
    --tracker-name "ExampleTracker"
```

将 AWS Lambda 与 MQTT 配合使用

创建 Lambda 函数：

要在 AWS IoT Core 和 Amazon Location Service 之间建立连接，您需要一个 AWS Lambda 函数来处理由 EventBridge CloudWatch 事件转发的消息。此函数将提取所有位置数据，将其格式化为 Amazon Location Service，然后通过 Amazon Location 跟踪器 API 提交。您可以通过 AWS Lambda 控制台创建此函数，也可以使用 AWS Command Line Interface (AWS CLI) 或 AWS Lambda APIs。要使用控制台创建 Lambda 函数，将位置更新发布到 Amazon Location，请执行以下操作：

1. 打开 AWS Lambda 控制台，网址为 https://console.aws.amazon.com/lambda/。
2. 从左侧导航窗格中，选择函数。
3. 选择创建函数，并确保选择从头开始创作。
4. 填写以下选框：
   - a 函数名
   - 在 “运行时” 选项中，选择 Node.js 16.x。
5. 选择创建函数。
6. 选择 代码 选项卡以打开编辑器。
7. 使用以下内容覆盖 index.js 中的占位符代码：

```
const AWS = require('aws-sdk')
const iot = new AWS.Iot();
exports.handler =  function(event) {
    console.log("event===>>>", JSON.stringify(event));
    var param = {
        endpointType: "iot:Data-ATS"
        };
    iot.describeEndpoint(param, function(err, data) {
        if (err) {
```

```
            console.log("error===>>>", err, err.stack); // an error occurred
        } else {
                var endp = data['endpointAddress'];
                const iotdata = new AWS.IotData({endpoint: endp});
                const trackerEvent = event["detail"]["EventType"];
                const src = event["source"];
                const time = event["time"];
                const gfId = event["detail"]["GeofenceId"];
                const resources = event["resources"][0];
                const splitResources = resources.split(".");
                const geofenceCollection = splitResources[splitResources.length -
1];
                const coordinates = event["detail"]["Position"];

                const deviceId = event["detail"]["DeviceId"];
                console.log("deviceId===>>>", deviceId);
                const msg =  {
                    "trackerEventType" : trackerEvent,
                    "source" : src,
                    "eventTime" : time,
                    "geofenceId" : gfId,
                    "coordinates": coordinates,
                    "geofenceCollection": geofenceCollection
                };
   const params = {
       topic: `${deviceId}/tracker`,
       payload: JSON.stringify(msg),
       qos: 0
            };
   iotdata.publish(params, function(err, data) {
       if (err) {
           console.log("error===>>>", err, err.stack); // an error occurred
       } else {
           console.log("Ladmbda triggered===>>>", trackerEvent);  // successful
response
              }
       });
      }
     });
    }
```

8. 选择部署以保存您更新的函数。
9. 选择配置选项卡。

10. 在 “触发器” 部分，单击 “添加触发器”。
11. 在 “来源” 字段中选择 EventBridge （CloudWatch 事件）。
12. 选择 “现有规则” 单选选项。
13. 输入这样的规则名称 `-GEOFENCECOLLECTION_NAM AmazonLocationMonitor E`。
14. 单击 “添加” 按钮。
15. 这还将在权限选项卡中附加 “基于资源的策略声明”

MQTT 测试客户端

1. 打开 https://console.aws.amazon.com/iot/。
2. 在左侧导航窗格中，选择 MQTT 测试客户端。
3. 您将看到标题为 MQTT 测试客户端的部分，您可以在其中配置 MQTT 连接。
4. 配置必要的设置后，单击连接按钮，使用提供的参数建立与 MQTT 代理的连接。
5. 记下端点值。

连接后，您可以使用 MQTT 测试客户端界面中提供的相应输入字段订阅 MQTT 主题或向主题发布消息。接下来，您将附上 MQTT 策略：

1. 在左侧菜单的管理下，展开安全性选项，然后单击策略。
2. 单击创建策略按钮。
3. 输入策略名称。
4. 在策略文档上，选择 JSON 选项卡。
5. 复制粘贴下面显示的策略，但请务必使用您的 *REGION* 和 *ACCOUNT_ID* 更新所有元素：

```
{
 "Version": "2012-10-17",
 "Statement": [
                {
                  "Action": [
                    "iot:Connect",
                    "iot:Publish",
                    "iot:Subscribe",
                    "iot:Receive"
                     ],
                  "Resource": [
```

```
                    "arn:aws:iot:REGION:ACCOUNT_ID:client/${cognito-
identity.amazonaws.com:sub}",
                    "arn:aws:iot:REGION:ACCOUNT_ID:topic/${cognito-
identity.amazonaws.com:sub}",
                    "arn:aws:iot:REGION:ACCOUNT_ID:topicfilter/${cognito-
identity.amazonaws.com:sub}/",
                    "arn:aws:iot:REGION:ACCOUNT_ID:topic/${cognito-
identity.amazonaws.com:sub}/tracker"
                    ],
                  "Effect": "Allow"
                }
            ]
}
```

> **Note**
>
> 记录策略名称和主题名称，以便在下一个步骤中使用。

6. 选择创建按钮完成操作。

完成上述步骤后，现在，您将按如下方式更新访客角色的权限：

1. 导航至 Amazon Cognito 并打开您的身份池。然后，继续设置用户访问权限并选择访客角色。
2. 单击权限策略以启用编辑。

```
{
    'Version': '2012-10-17',
    'Statement': [
        {
            'Action': [
                'geo:GetMap*',
                'geo:BatchUpdateDevicePosition',
                'geo:BatchEvaluateGeofences',
                'iot:Subscribe',
                'iot:Publish',
                'iot:Connect',
                'iot:Receive',
                'iot:AttachPrincipalPolicy',
                'iot:AttachPolicy',
                'iot:DetachPrincipalPolicy',
                'iot:DetachPolicy'
```

```
            ],
            'Resource': [
                'arn:aws:geo:us-east-1:{USER_ID}:map/{MAP_NAME}',
                'arn:aws:geo:us-east-1:{USER_ID}:tracker/{TRACKER_NAME}',
                'arn:aws:geo:us-east-1:{USER_ID}:geofence-collection/
{GEOFENCE_COLLECTION_NAME}',
                'arn:aws:iot:us-east-1:{USER_ID}:client/${cognito-
identity.amazonaws.com:sub}',
                'arn:aws:iot:us-east-1:{USER_ID}:topic/${cognito-
identity.amazonaws.com:sub}',
                'arn:aws:iot:us-east-1:{USER_ID}:topicfilter/${cognito-
identity.amazonaws.com:sub}/*',
                'arn:aws:iot:us-east-1:{USER_ID}:topic/${cognito-
identity.amazonaws.com:sub}/tracker'
            ],
            'Effect': 'Allow'
        },
        {
            'Condition': {
                'StringEquals': {
                    'cognito-identity.amazonaws.com:sub': '${cognito-
identity.amazonaws.com:sub}'
                }
            },
            'Action': [
                'iot:AttachPolicy',
                'iot:DetachPolicy',
                'iot:AttachPrincipalPolicy',
                'iot:DetachPrincipalPolicy'
            ],
            'Resource': [
                '*'
            ],
            'Effect': 'Allow'
        }
    ]
}
```

3. 通过上述策略更改，现在已为应用程序正确配置了所有必要的 AWS 资源。

设置示例应用程序代码

1. 打开 Android Studio，从 “版本控制” 中选择 “新建” 和 “项目”
2. 导航到 Android Studio 左上角的文件菜单。
3. 从下拉菜单中选择 “新建”。
4. 选择 “版本控制中的项目”。
5. 输入存储库 URL 在出现的对话框中，找到标有 “URL” 的字段。
6. 将示例应用程序的以下 URL 复制并粘贴到此字段中：https://github.com/aws-geospatial/amazon-location-mobile-sdk-sample-app-android .git
7. 确定要克隆存储库的目录。使用默认目录或选择自定义位置。
8. 设置存储库 URL 和目录首选项后，点击 “克隆” 按钮。Android Studio 将继续将存储库克隆到您的指定位置。
9. 现在，您已将应用程序克隆到您的计算机上，可以开始使用它了。

使用示例应用程序

要使用该示例，请按照以下步骤进行操作：

- 创建 **`custom.properties`**：

  要配置 `custom.properties` 文件，请按照以下步骤进行操作：

  1. 打开您的首选文本编辑器或 IDE。
  2. 创建新的文件。
  3. 使用文件名 `custom.properties` 保存该文件。
  4. `custom.properties`使用以下代码示例更新，并用实际值替换`MQTT_END_POINT`、`POLICY_NAME`、`GEOFENCE_COLLECTION_NAME`和`TOPIC_TRACKER`：

     ```
     MQTT_END_POINT=xxxxxxxxxxxxx-xxx.xxx.us-east-1.amazonaws.com
     POLICY_NAME=xxxxxxxxx
     GEOFENCE_COLLECTION_NAME=xxxxxxxxxxxxxxxx
     TOPIC_TRACKER=xxxxxxxxxx
     ```

  5. 清理并重新构建项目。之后，您可以运行项目。
- 登录：

  要登录应用程序，请按照以下步骤进行操作：

1. 按登录按钮。
2. 提供身份池 ID、跟踪器名称和地图名称。
3. 再次按登录完成操作。

- 管理筛选器：

  打开配置屏幕，然后执行以下操作：

  1. 使用开关用户界面打开/关闭过滤器。
  2. 需要时更新时间和距离筛选器。

- 跟踪操作：

  打开跟踪屏幕，然后执行以下操作：

  - 您可以通过按下相应的按钮在前景、后台或省电模式下开始和停止跟踪。

使用亚马逊 Location Service 进行身份验证

要使用 Amazon Location Service，必须向用户授予访问 APIs 构成亚马逊位置的资源的权限。您可以使用授予对资源的访问权限的三种策略：

- 使用 API 密钥 — 要向未经身份验证的用户授予访问权限，您可以创建 API 密钥，以授予对您的 Amazon Location Service 资源和操作的只读访问权限。在您不想对每个用户进行身份验证的情况下，这很有用。例如，Web 应用程序。

  有关 API 密钥的更多信息，请参阅 the section called “使用 API 密钥”。
- 使用 Amazon Cognito——API 密钥的替代方法是使用 Amazon Cognito 授予匿名访问权限。Amazon Cognito 允许您使用策略创建更丰富的授权，以定义未经身份验证的用户可以执行的操作。

  有关使用 Amazon Cognito 的更多信息，请参阅 the section called “使用 Amazon Cognito”。
- 使用 AWS Identity and Access Management (IAM)-要向使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 进行身份验证的用户授予访问权限，请创建允许访问所需资源的 IAM 策略。

  有关 IAM 和 Amazon Location 的更多信息，请参阅 the section called “使用 IAM”。

选择身份验证方法

在类似的场景中，API 密钥和 Amazon Cognito 的使用方式类似，那么你为什么要选择一个而不是另一个呢？ 以下列表重点介绍两者之间的一些区别：

- 性能：
  - API 密钥：相对更快
  - 亚马逊 Cognito：相对较慢
- 可用性：
  - API 密钥：地图、地点和路线的 Amazon 位置 APIs
  - 亚马逊 Cognito：全部 APIs
- 与其他身份验证方法结合使用？
  - API 密钥：否
  - 亚马逊 Cognito：是的

Comparison（比较）

- API 密钥仅适用于地图、地点和路线操作。Amazon Cognito 可用于对大多数亚马逊定位服务的访问进行身份验证。 APIs
- 使用 API 密钥进行地图请求的性能通常比 Amazon Cognito 的类似场景要快。更简单的身份验证意味着在短时间内再次获得相同的地图图块时，可以减少往返服务的次数和缓存的请求。
- 借助 Amazon Cognito，您可以使用自己的身份验证流程或组合多种身份验证方法，使用 Amazon Cognito 联合身份。

  有关更多信息，请参阅 Amazon Cognito 开发者指南中的联合身份入门。

使用 API 密钥进行身份验证

> **Note**
>
> API 密钥仅适用于地图、地点和路线资源，您无法修改或创建这些资源。如果您的应用程序需要访问其他资源或为未经身份验证的用户执行操作，则可以使用 Amazon Cognito 与 API 密钥一起提供访问权限或取代 API 密钥。有关更多信息，请参阅 the section called “使用 Amazon Cognito”。

API 密钥是一个密钥值，它与您 AWS 账户中的特定 Amazon Location Service 资源或 API 以及您可以对这些资源执行的特定操作相关联。您可以在应用程序中使用 API 密钥对这些资源的亚马逊地址进行未经身份验证 APIs 的调用。

例如，如果您将 API 密钥与资源和/或 API 相关联，则使用该 `GetPlace*` API 密钥的应用程序将能够调用特定的 API 密钥 APIs。相同的 API 密钥不会授予更改或更新与其无关的任何资源或调用的 APIs 权限。

当您在应用程序 APIs 中调用 Amazon Location Service 时，您通常以有权进行 API 调用的经过身份验证的用户进行此调用。但是，在某些情况下，您不想对应用程序的每个用户进行身份验证。

例如，您可能希望任何使用该网站的人都可以使用显示您的营业地点的 Web 应用程序，无论他们是否登录。在这种情况下，一种替代方法是使用 API 密钥进行 API 调用。

有关何时使用 API 密钥的更多信息，请参阅the section called “最佳实践”。

为亚马逊 Location Service 创建 API 密钥

您可以通过亚马逊定位服务控制台或亚马逊定位 API 创建 API 密钥。 AWS CLI继续执行以下相应程序。

Amazon Location console

使用 Amazon Location Service 控制台创建 API 密钥

1. 在 Amazon Location 控制台中，从左侧菜单中选择 API 密钥。
2. 在 API 密钥页面上，选择创建 API 密钥。
3. 在创建 API 密钥页面中，填写以下信息：
   - 名称——您的 API 密钥的名称，例如 `ExampleKey`。
   - 描述 – API 密钥的可选描述。
   - 资源-在下拉列表中，选择要使用此 API 密钥访问的 Amazon Location 资源。您可以通过选择添加资源来添加多个资源。
   - 操作——指定您要使用此 API 密钥授权的操作。必须至少选择一个操作才能匹配所选的每种资源类型。例如，如果您选择了地点资源，则必须在地点操作下选择至少一个选项。
   - 到期时间 – （可选）添加 API 密钥的到期日期和时间。有关更多信息，请参阅 the section called “最佳实践”。
   - 反向链接-（可选）添加一个或多个可以使用 API 密钥的域名。例如，如果 API 密钥是为了允许应用程序在网站上运行`example.com`，那么你可以将其设置`*.example.com/`为允许的反向链接。
   - 标签 – （可选）向 API 密钥添加标签。
4. 选择创建 API 密钥以创建 API 密钥。
5. 在 API 密钥的详情页面上，您可以看到有关您创建的 API 密钥的信息。选择显示 API 密钥以查看您在调用 Amazon Location 时使用的密钥值 APIs。键值的格式为 `v1.public.a1b2c3d4...`。

AWS CLI

1. 使用 create-key 命令。以下示例创建了一个名为的 API 密钥`ExampleKey`，该密钥没有到期日期，并且可以访问单个地图资源。

```
aws location \
```

```
  create-key \
  --key-name ExampleKey \
  --restrictions '{"AllowActions":["geo:GetMap*"],"AllowResources":
["arn:aws:geo:region:map/mapname"]}' \
  --no-expiry
```

2. 响应中包含访问应用程序中的资源时要使用的 API 密钥值。键值的格式为 `v1.public.a1b2c3d4...`。要了解有关使用 API 密钥渲染地图的更多信息，请参阅 the section called “调用 API”。对 create-key 的响应如下所示：

```
{
    "Key": "v1.public.a1b2c3d4...",
    "KeyArn": "arn:aws:geo:region:accountId:api-key/ExampleKey",
    "KeyName": "ExampleKey",
    "CreateTime": "2023-02-06T22:33:15.693Z"
}
```

3. 您也可以在以后使用 `describe-key` 来查找键值。以下示例说明如何调用名为`describe-key`的 API 密钥`ExampleKey`。

```
aws location describe-key \
    --key-name ExampleKey
```

Amazon Location API

使用 Amazon 营业地点的CreateKey操作 APIs。以下示例是一个 API 请求，用于创建名`ExampleKey`为的 API 密钥，该密钥没有到期日期，并且可以访问单个地图资源。

```
POST /metadata/v0/keys HTTP/1.1
Content-type: application/json
{
"KeyName": "ExampleKey",
"NoExpiry": true,
  "Restrictions": {
    "AllowActions": [
      "geo-places:*",
      "geo-routes:*",
      "geo-maps:*"
    ],
        "AllowResources": [
            "arn:aws:geo-places:Region::provider/default",
```

```
            "arn:aws:geo-routes:Region::provider/default",
            "arn:aws:geo-maps:Region::provider/default"
        ]
  }
```

响应中包含访问应用程序中的资源时要使用的 API 密钥值。键值的格式`v1.public.a1b2c3d4...`为。

您也可以使用 DescribeKey API 在以后查找密钥的密钥值。

使用 API 密钥调用 Amazon Location API

创建 API 密钥后，您可以使用该密钥值在应用程序 APIs 中调用 Amazon Location。

API

支持 API 密钥的还有一个采用 API 密钥值的附加参数。 APIs 例如，如果您调用 `GetPlace` API，则可以填写密钥参数，如下所示

```
curl --request GET –url 'https://places.geo.eu-central-1.amazonaws.com/v2/place/
{PLACEID}?key={APIKEY}&language=jp'
```

AWS CLI

使用`--key`参数时，还应使用`--no-sign-request`参数，以避免使用 Sig v4 进行签名。

```
aws geo-places get-place --place-id $PLACEID --language jp --key $APIKEY
```

SDK (web)

使用以下代码：

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Display a map</title>
        <meta property="og:description" content="Initialize a map in an HTML element
 with MapLibre GL JS." />
        <meta charset='utf-8'>
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <link rel='stylesheet' href='https://unpkg.com/maplibre-gl@4.x/dist/
maplibre-gl.css' />
```

```
        <script src='https://unpkg.com/maplibre-gl@4.x/dist/maplibre-gl.js'></
script>
        <style>
            body { margin: 0; }
            #map { height: 100vh; }
        </style>
    </head>
    <body>

        <div id="map"></div>
        <script>

            const apiKey = "<api key>"; // check how to create api key for Amazon
 Location
            const mapStyle = "Standard";  // eg. Standard, Monochrome, Hybrid,
 Satellite
            const awsRegion = "eu-central-1"; // eg. us-east-2, us-east-1, us-
west-2, ap-south-1, ap-southeast-1, ap-southeast-2, ap-northeast-1, ca-central-1,
 eu-central-1, eu-west-1, eu-west-2, eu-south-2, eu-north-1, sa-east-1
            const styleUrl = `https://maps.geo.${awsRegion}.amazonaws.com/v2/styles/
${mapStyle}/descriptor?key=${apiKey}`;


            const map = new maplibregl.Map({
                container: 'map', // container id
                style: styleUrl, // style URL
                center: [25.24,36.31], // starting position [lng, lat]
                zoom: 2, // starting zoom
            });
        </script>
    </body>
</html>
```

SDK (iOS, Swift)

使用以下代码：

```
import UIKit
import MapLibre

class ViewController: UIViewController {
    let apiKey = "Enter your API key" // The previously-created API Key to use
    let regionName = "Enter your region name" // The service region - us-east-1, ap-
south-1, etc
```

```
    var mapView: MLNMapView!

    override func viewDidLoad() {
        super.viewDidLoad()
        loadMap()
    }

    func loadMap() {
        let styleName = "Standard" // The map style - Standard, Monochrome, Hybrid,
 Satellite
        let colorName = "Light" // The color scheme - Light, Dark

        // The Amazon Location Service map style URL that MapLibre will use to
 render the maps.
        let styleURL = URL(string: "https://maps.geo.\(regionName).amazonaws.com/v2/
styles/\(styleName)/descriptor?key=\(apiKey)&color-scheme=\(colorName)")

        // Initialize MapLibre
        mapView = MLNMapView(frame: view.bounds)
        mapView.styleURL = styleURL
        mapView.autoresizingMask = [.flexibleWidth, .flexibleHeight]
        // Set the starting camera position and zoom level for the map
        mapView.setCenter(CLLocationCoordinate2D(latitude: 49.246559, longitude:
 -123.063554), zoomLevel: 10, animated: false)
        view.addSubview(mapView!)
    }
}
```

SDK (Android, Kotlin)

使用以下代码：

```
class MapActivity : Activity(), OnMapReadyCallback {

    private lateinit var mBinding: ActivityMapBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        initializeMap(savedInstanceState)
    }

    private fun initializeMap(savedInstanceState: Bundle?) {
        // Init MapLibre
        // See the MapLibre Getting Started Guide for more details
```

```
        // https://maplibre.org/maplibre-native/docs/book/android/getting-started-
guide.html
        MapLibre.getInstance(this@MapActivity)
        mBinding = ActivityMapBinding.inflate(layoutInflater)
        setContentView(mBinding.root)
        mBinding.mapView.onCreate(savedInstanceState)
        mBinding.mapView.getMapAsync(this)
    }

    override fun onMapReady(mapLibreMap: MapLibreMap) {
        mapLibreMap.setStyle(Style.Builder().fromUri(getMapUrl())) {
            // Set the starting camera position
            mapLibreMap.cameraPosition =
 CameraPosition.Builder().target(LatLng(49.246559, -123.063554)).zoom(10.0).build()
            mapLibreMap.uiSettings.isLogoEnabled = false
            mapLibreMap.uiSettings.attributionGravity = Gravity.BOTTOM or
 Gravity.END

 mapLibreMap.uiSettings.setAttributionDialogManager(AttributionDialogManager(this,
 mapLibreMap))
        }
    }

    // Return the Amazon Location Service map style URL
    // MapLibre will use this to render the maps.
    // awsRegion: The service region - us-east-1, ap-south-1, etc
    // mapStyle: The map style - Standard, Monochrome, Hybrid, Satellite
    // API_KEY: The previously-created API Key to use
    // colorName: The color scheme to use - Light, Dark
    private fun getMapUrl() =
           "https://maps.geo.${getString(R.string.awsRegion)}.amazonaws.com/v2/
styles/${getString(R.string.mapStyle)}/descriptor?key=${BuildConfig.API_KEY}&color-
scheme=${getString(R.string.colorName)}"

    override fun onStart() {
        super.onStart()
        mBinding.mapView.onStart()
    }

    override fun onResume() {
        super.onResume()
        mBinding.mapView.onResume()
    }
```

```
    override fun onPause() {
        super.onPause()
        mBinding.mapView.onPause()
    }

    override fun onStop() {
        super.onStop()
        mBinding.mapView.onStop()
    }

    override fun onSaveInstanceState(outState: Bundle) {
        super.onSaveInstanceState(outState)
        mBinding.mapView.onSaveInstanceState(outState)
    }

    override fun onLowMemory() {
        super.onLowMemory()
        mBinding.mapView.onLowMemory()
    }

    override fun onDestroy() {
        super.onDestroy()
        mBinding.mapView.onDestroy()
    }
}
```

API 密钥最佳实践

API 密钥包含一个纯文本值，允许访问一个或多个资源或 APIs 您的资源 AWS 账户。如果有人复制了您的 API 密钥，他们就可以访问相同的资源和 APIs。为最大限度地减少潜在影响，请查看以下最佳实践：

- 限制 API 密钥

  为避免出现上述情况，最好限制您的 API 密钥。创建密钥时，您可以指定可以使用密钥的域名或反向链接。
- 管理 API 密钥的生命周期

  您可以创建无限期有效的 API 密钥。但是，如果您想创建临时 API 密钥、定期轮换 API 密钥或撤销现有 API 密钥，则可以使用 API 密钥过期时间。
  - 您可以在创建或更新 API 密钥时为其设置过期时间。

- API 密钥在到达其到期时间时将会自动停用。非活动密钥不能再用于发出请求。
- 您可以通过删除过期时间将临时密钥更改为永久密钥。
- 您可以在 API 密钥停用 90 天后将其删除。
- 如果您尝试停用最近七天内使用过的 API 密钥，系统将提示您确认是否要进行更改。

  如果您使用的是 Amazon Location Service API 或`true`，请将`ForceUpdate`参数设置为，否则您将收到错误消息。 AWS CLI

使用 Amazon Cognito 进行身份验证

您可以使用 Amazon Cognito 身份验证作为直接使用 AWS Identity and Access Management (IAM) 用户进行前端 SDK 请求的替代方案。

Amazon Cognito 为您的 Web 和移动应用程序提供身份验证、授权和用户管理。您可以使用带有 Amazon Location 的 Amazon Cognito 未经身份验证的身份池作为应用程序检索限定范围内的临时凭证的一种方式。 AWS

想要了解更多信息，请参阅 Amazon Cognito 开发人员指南中的用户池入门。

出于以下原因，您可能需要使用这种形式的身份验证：

- 未经身份验证的用户 – 如果您的网站包含匿名用户，则可以使用 Amazon Cognito 身份池。

  想要了解更多信息，请在 the section called “使用 Amazon Cognito” 上参阅本部分。
- 您自己的身份验证 – 如果您想使用自己的身份验证流程或组合使用多种身份验证方法，则可以使用 Amazon Cognito 联合身份。

  想要了解更多信息，请参阅 Amazon Cognito 开发人员指南 中的开始使用联合身份。

使用亚马逊 Cognito 和亚马逊定位服务

您可以将与未经身份验证的身份角色关联的 AWS Identity and Access Management (IAM) 策略用于以下操作：

Maps

地图操作列表

- `geo-maps:GetStaticMap`

- geo-maps:GetTile

> **Note**
>
> 上述操作的资源名称为：
>
> ```
> arn:aws:geo-maps:region::provider/default
> ```

Places

地点操作列表：

- geo-places:Geocode
- geo-places:ReverseGeocode
- geo-places:SearchNearby
- geo-places:SearchText
- geo-places:Autocomplete
- geo-places:Suggest
- geo-places:GetPlace

> **Note**
>
> 上述操作的资源名称为：
>
> ```
> arn:aws:geo-places:region::provider/default
> ```

Routes

路线操作列表：

- geo-routes:CalculateRoutes
- geo-routes:CalculateRouteMatrix
- geo-routes:CalculateIsolines

- geo-routes:OptimizeWaypoints
- geo-routes:SnapToRoads

> **Note**
>
> 上述操作的资源名称为：
>
> ```
> arn:aws:geo-routes:region::provider/default
> ```

Geofences and Trackers

地理围栏和追踪器操作列表

- geo:GetGeofence
- geo:ListGeofences
- geo:PutGeofence
- geo:BatchDeleteGeofence
- geo:BatchPutGeofence
- geo:BatchEvaluateGeofences
- geo:GetDevicePosition*
- geo:ListDevicePositions
- geo:BatchDeleteDevicePositionHistory
- geo:BatchGetDevicePosition
- geo:BatchUpdateDevicePosition

> **Note**
>
> 上述操作的资源名称为：
>
> ```
> arn:aws:geo:region:accountID:tracker/ExampleTracker
> ```

Previous version

先前版本操作列表：

- `geo:GetMap*`
- `geo:SearchPlaceIndexForText`
- `geo:SearchPlaceIndexForPosition`
- `geo:GetPlace`
- `geo:CalculateRoute`
- `geo:CalculateRouteMatrix`

> **Note**
>
> 上述操作的资源名称为：
> 映射
>
> ```
> arn:aws:geo:region:accountID:map/ExampleMap
> ```
>
> 地方
>
> ```
> arn:aws:geo:region:accountID:place-index/ExamplePlaceIndex
> ```
>
> 路线
>
> ```
> arn:aws:geo:region:accountID:route-calculator/ExampleCalculator
> ```

创建一个 Amazon Cognito 身份池

您可以创建 Amazon Cognito 身份池，允许未经身份验证的访客通过 Amazon Cognito 控制台、或 Amazon Cognito 访问您的应用程序。 AWS CLI APIs

> **Important**
>
> 您创建的资源池必须与您正在使用的 Amazon Location Service 资源位于相同 AWS 账户 且所在的 AWS 区域相同。

Console

使用 Amazon Cognito 控制台创建身份池

1. 转到 Amazon Cognito 控制台。
2. 选择 Manage Identity Pools（管理身份池）。
3. 选择创建新身份池，然后输入身份池的名称。
4. 从未经验证的身份可折叠部分中，选择启用未经验证的身份的访问权限。
5. 选择创建池。
6. 选择您希望用于身份池的 IAM 角色。
7. 展开查看详情。
8. 在未经身份验证的身份下，输入角色名称。
9. 展开查看策略文档部分，然后选择编辑以添加您的策略。
10. 添加您的策略来授予对您的资源的访问权限。

> **Note**
>
> 有关操作列表，请参阅上面的the section called “使用亚马逊 Cognito 和亚马逊定位服务”部分。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
  Sid": "RoutesReadOnly",
  Effect": "Allow",
  Action": [
        // add comma separated value from the previous section
      ],
  Resource": "value from previous section"
    }
  ]
}
```

11. 选择允许以创建您的身份池。

在网络中使用 Amazon Cognito 身份池

以下示例交换您为证书创建的未经身份验证的身份池，然后使用这些证书进行调用`CalculateIsolines`。为了简化这项工作，该示例使用了 Amazon 定位the section called “使用身份验证助手”程序。这可以代替获取和刷新凭证。

此示例使用适用于 JavaScript v3 的 AWS SDK。

```
import { GeoRoutesClient, CalculateIsolinesCommand , } from "@aws-sdk/client-geo-
routes"; // ES Modules import
import { withIdentityPoolId } from "@aws/amazon-location-utilities-auth-helper";

const identityPoolId = "<identity pool ID>"; // for example, us-
east-1:1sample4-5678-90ef-aaaa-1234abcd56ef

const authHelper = await withIdentityPoolId(identityPoolId);

const client = new GeoRoutesClient({
    ...authHelper.getClientConfig(),
    region: "<region>", // The region containing the identity pool
});

const input = {
    DepartNow: true,
    TravelMode: "Car",
    Origin: [-123.12327, 49.27531],
    Thresholds: {
        Time: [5, 10, 30],
    },
};

const command = new CalculateIsolinesCommand(input);
const response = await client.send(command);

console.log(JSON.stringify(response, null, 2))
```

用于 AWS Identity and Access Management 进行身份验证

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制谁可以通过身份验证（登录）和获得授权（具有权限）来使用 Amazon Location 资源。您可以使用 IAM AWS 服务 ，无需支付额外费用。

主题

- 受众
- 使用身份进行身份验证
- 使用策略管理访问
- Amazon Location Service 如何与 IAM 协同工作
- Amazon Location Service 如何处理未经身份验证的用户
- Amazon Location Service 基于身份的策略示例
- Amazon Location Service 身份和访问权限故障排查

受众

您的使用方式 AWS Identity and Access Management (IAM) 会有所不同，具体取决于您在 Amazon Location 中所做的工作。

服务用户 – 如果您使用 Amazon Location Service 服务来完成任务，则您的管理员会为您提供所需的凭证和权限。随着您使用更多 Amazon Location 功能来完成任务，您可能需要额外权限。了解如何管理访问权限有助于您向管理员请求适合的权限。如果您无法访问 Amazon Location 中的特征，请参阅 Amazon Location Service 身份和访问权限故障排查。

服务管理员 – 如果您在公司负责管理 Amazon Location 资源，您可能对 Amazon Location 具有完全访问权限。您有责任确定您的服务用户应访问哪些 Amazon Location 功能和资源。然后，您必须向 IAM 管理员提交请求以更改服务用户的权限。请查看该页面上的信息以了解 IAM 的基本概念。要了解有关您的公司如何将 IAM 与 Amazon Location 搭配使用的更多信息，请参阅 Amazon Location Service 如何与 IAM 协同工作。

IAM 管理员 – 如果您是 IAM 管理员，则可能需要了解有关如何编写策略以管理对 Amazon Location 的访问权限的详细信息。要查看您可在 IAM 中使用的 Amazon Location 基于身份的策略示例，请参阅 Amazon Location Service 基于身份的策略示例。

使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 IAM 用户身份或通过担 AWS 账户根用户任 IAM 角色进行身份验证（登录 AWS）。

您可以使用通过身份源提供的凭据以 AWS 联合身份登录。 AWS IAM Identity Center （ IAM Identity Center ）用户、贵公司的单点登录身份验证以及您的 Google 或 Facebook 凭据就是联合身份的示例。当您以联合身份登录时，您的管理员以前使用 IAM 角色设置了身份联合验证。当你使用联合访问 AWS 时，你就是在间接扮演一个角色。

根据您的用户类型，您可以登录 AWS Management Console 或 AWS 访问门户。有关登录的更多信息 AWS，请参阅《AWS 登录 用户指南》中的如何登录到您 AWS 账户的。

如果您 AWS 以编程方式访问，则会 AWS 提供软件开发套件 (SDK) 和命令行接口 (CLI)，以便使用您的凭据对请求进行加密签名。如果您不使用 AWS 工具，则必须自己签署请求。有关使用推荐的方法自行签署请求的更多信息，请参阅《IAM 用户指南》中的用于签署 API 请求的AWS 签名版本 4。

无论使用何种身份验证方法，您都可能需要提供其他安全信息。例如， AWS 建议您使用多重身份验证 (MFA) 来提高账户的安全性。要了解更多信息，请参阅《AWS IAM Identity Center 用户指南》中的多重身份验证和《IAM 用户指南》中的 IAM 中的AWS 多重身份验证。

AWS 账户 root 用户

创建时 AWS 账户，首先要有一个登录身份，该身份可以完全访问账户中的所有资源 AWS 服务 和资源。此身份被称为 AWS 账户 root 用户，使用您创建账户时使用的电子邮件地址和密码登录即可访问该身份。强烈建议您不要使用根用户执行日常任务。保护好根用户凭证，并使用这些凭证来执行仅根用户可以执行的任务。有关要求您以根用户身份登录的任务的完整列表，请参阅 IAM 用户指南中的需要根用户凭证的任务。

联合身份

作为最佳实践，要求人类用户（包括需要管理员访问权限的用户）使用与身份提供商的联合身份验证 AWS 服务 通过临时证书进行访问。

联合身份是指您的企业用户目录、Web 身份提供商、Identity C enter 目录中的用户，或者任何使用 AWS 服务 通过身份源提供的凭据进行访问的用户。 AWS Directory Service当联合身份访问时 AWS 账户，他们将扮演角色，角色提供临时证书。

要集中管理访问权限，建议您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中创建用户和群组，也可以连接并同步到您自己的身份源中的一组用户和群组，以便在您的所有 AWS 账户 和应用程序中使用。有关 IAM Identity Center 的信息，请参阅 AWS IAM Identity Center 用户指南中的什么是 IAM Identity Center？。

IAM 用户和群组

I AM 用户是您 AWS 账户 内部对个人或应用程序具有特定权限的身份。在可能的情况下，我们建议使用临时凭证，而不是创建具有长期凭证（如密码和访问密钥）的 IAM 用户。但是，如果您有一些特定的使用场景需要长期凭证以及 IAM 用户，建议您轮换访问密钥。有关更多信息，请参阅《IAM 用户指南》中的对于需要长期凭证的用例，应在需要时更新访问密钥。

IAM 组是一个指定一组 IAM 用户的身份。您不能使用组的身份登录。您可以使用组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，您可以拥有一个名为的群组，IAMAdmins并向该群组授予管理 IAM 资源的权限。

用户与角色不同。用户唯一地与某个人员或应用程序关联，而角色旨在让需要它的任何人代入。用户具有永久的长期凭证，而角色提供临时凭证。要了解更多信息，请参阅《IAM 用户指南》中的 IAM 用户的使用案例。

IAM 角色

I AM 角色是您内部具有特定权限 AWS 账户 的身份。它类似于 IAM 用户，但与特定人员不关联。要在中临时担任 IAM 角色 AWS Management Console，您可以从用户切换到 IAM 角色（控制台）。您可以通过调用 AWS CLI 或 AWS API 操作或使用自定义 URL 来代入角色。有关使用角色的方法的更多信息，请参阅《IAM 用户指南》中的代入角色的方法。

具有临时凭证的 IAM 角色在以下情况下很有用：

- 联合用户访问：要向联合身份分配权限，请创建角色并为角色定义权限。当联合身份进行身份验证时，该身份将与角色相关联并被授予由此角色定义的权限。有关用于联合身份验证的角色的信息，请参阅《IAM 用户指南》中的针对第三方身份提供商创建角色（联合身份验证）。如果您使用 IAM Identity Center，则需要配置权限集。为控制您的身份在进行身份验证后可以访问的内容，IAM Identity Center 将权限集与 IAM 中的角色相关联。有关权限集的信息，请参阅《AWS IAM Identity Center 用户指南》中的权限集。
- 临时 IAM 用户权限：IAM 用户可代入 IAM 用户或角色，以暂时获得针对特定任务的不同权限。
- 跨账户存取：您可以使用 IAM 角色以允许不同账户中的某个人（可信主体）访问您的账户中的资源。角色是授予跨账户访问权限的主要方式。但是，对于某些资源 AWS 服务，您可以将策略直接附加到资源（而不是使用角色作为代理）。要了解用于跨账户访问的角色和基于资源的策略之间的差别，请参阅 IAM 用户指南中的 IAM 中的跨账户资源访问。
- 跨服务访问 — 有些 AWS 服务 使用其他 AWS 服务服务中的功能。例如，当您在服务中拨打电话时，该服务通常会在 Amazon 中运行应用程序 EC2 或在 Amazon S3 中存储对象。服务可能会使用发出调用的主体的权限、使用服务角色或使用服务相关角色来执行此操作。

- 转发访问会话 (FAS) — 当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务 向下游服务发出请求的请求。 AWS 服务只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两项操作的权限。有关发出 FAS 请求时的策略详情，请参阅转发访问会话。
- 服务角色 - 服务角色是服务代表您在您的账户中执行操作而分派的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。
- 服务相关角色-服务相关角色是一种与服务相关联的服务角色。 AWS 服务服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户 ，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。

- 在 A@@ mazon 上运行的应用程序 EC2 — 您可以使用 IAM 角色管理在 EC2 实例上运行并发出 AWS CLI 或 AWS API 请求的应用程序的临时证书。这比在 EC2 实例中存储访问密钥更可取。要为 EC2 实例分配 AWS 角色并使其可供其所有应用程序使用，您需要创建一个附加到该实例的实例配置文件。实例配置文件包含该角色，并允许在 EC2 实例上运行的程序获得临时证书。有关更多信息，请参阅 IAM 用户指南中的使用 IAM 角色向在 A mazon EC2 实例上运行的应用程序授予权限。

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略是其中的一个对象 AWS ，当与身份或资源关联时，它会定义其权限。 AWS 在委托人（用户、root 用户或角色会话）发出请求时评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的结构和内容的更多信息，请参阅 IAM 用户指南中的 JSON 策略概览。

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

默认情况下，用户和角色没有权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。管理员随后可以向角色添加 IAM 策略，用户可以代入角色。

IAM 策略定义操作的权限，无关乎您使用哪种方法执行操作。例如，假设您有一个允许 `iam:GetRole` 操作的策略。拥有该策略的用户可以从 AWS Management Console AWS CLI、或 AWS API 获取角色信息。

基于身份的策略

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户托管策略定义自定义 IAM 权限。

基于身份的策略可以进一步归类为内联策略或托管式策略。内联策略直接嵌入单个用户、组或角色中。托管策略是独立的策略，您可以将其附加到中的多个用户、群组和角色 AWS 账户。托管策略包括 AWS 托管策略和客户托管策略。要了解如何在托管策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的在托管策略与内联策略之间进行选择。

基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用 IAM 中的 AWS 托管策略。

访问控制列表 (ACLs)

访问控制列表 (ACLs) 控制哪些委托人（账户成员、用户或角色）有权访问资源。 ACLs 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

Amazon S3 和 Amazon VPC 就是支持的服务示例 ACLs。 AWS WAF要了解更多信息 ACLs，请参阅《亚马逊简单存储服务开发者指南》中的访问控制列表 (ACL) 概述。

其他策略类型

AWS 支持其他不太常见的策略类型。这些策略类型可以设置更常用的策略类型向您授予的最大权限。

- 权限边界：权限边界是一个高级特征，用于设置基于身份的策略可以为 IAM 实体（IAM 用户或角色）授予的最大权限。您可为实体设置权限边界。这些结果权限是实体基于身份的策略及其权限边界的交集。在 `Principal` 中指定用户或角色的基于资源的策略不受权限边界限制。任一项策略中的显式拒绝将覆盖允许。有关权限边界的更多信息，请参阅IAM 用户指南中的 IAM 实体的权限边界。
- 服务控制策略 (SCPs) — SCPs 是 JSON 策略，用于指定中组织或组织单位 (OU) 的最大权限 AWS Organizations。 AWS Organizations 是一项用于对您的企业拥有的多 AWS 账户 项进行分组和集中管理的服务。如果您启用组织中的所有功能，则可以将服务控制策略 (SCPs) 应用于您的任何或所有

帐户。SCP 限制成员账户中的实体（包括每个 AWS 账户根用户实体）的权限。有关 Organization SCPs s 和的更多信息，请参阅《AWS Organizations 用户指南》中的服务控制策略。

- 资源控制策略 (RCPs) — RCPs 是 JSON 策略，您可以使用它来设置账户中资源的最大可用权限，而无需更新附加到您拥有的每个资源的 IAM 策略。RCP 限制成员账户中资源的权限，并可能影响身份（包括身份）的有效权限 AWS 账户根用户，无论这些身份是否属于您的组织。有关 Organizations 的更多信息 RCPs，包括 AWS 服务 该支持的列表 RCPs，请参阅《AWS Organizations 用户指南》中的资源控制策略 (RCPs)。
- 会话策略：会话策略是当您以编程方式为角色或联合用户创建临时会话时作为参数传递的高级策略。结果会话的权限是用户或角色的基于身份的策略和会话策略的交集。权限也可以来自基于资源的策略。任一项策略中的显式拒绝将覆盖允许。有关更多信息，请参阅 IAM 用户指南中的会话策略。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 AWS 确定是否允许请求，请参阅 IAM 用户指南中的策略评估逻辑。

Amazon Location Service 如何与 IAM 协同工作

在使用 IAM 管理对 Amazon Location 的访问权限之前，您应该了解哪些 IAM 功能可用于 Amazon Location。

可以与 Amazon Location Service 结合使用的 IAM 功能

IAM 特征	Amazon Location 支持
the section called “基于身份的策略”	是
基于资源的策略	否
策略操作	是
策略资源	是
策略条件键（特定于服务）	是
ACLs	否
ABAC（策略中的标签）	是

IAM 特征	Amazon Location 支持
临时凭证	是
主体权限	否
服务角色	否
服务相关角色	否

要全面了解 Amazon Location 和其他 AWS 服务如何与大多数 IAM 功能配合使用，请参阅 IAM 用户指南中与 IAM 配合使用的AWS 服务。

Amazon Location 基于身份的策略

支持基于身份的策略：是

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户管理型策略定义自定义 IAM 权限。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。您无法在基于身份的策略中指定主体，因为它适用于其附加的用户或角色。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的 IAM JSON 策略元素引用。

Amazon Location 基于身份的策略示例

要查看 Amazon Location 基于身份的策略的示例，请参阅 Amazon Location Service 基于身份的策略示例。

Amazon Location 内基于资源的策略

支持基于资源的策略：否

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

要启用跨账户访问，您可以将整个账户或其他账户中的 IAM 实体指定为基于资源的策略中的主体。将跨账户主体添加到基于资源的策略只是建立信任关系工作的一半而已。当委托人和资源处于不同位置时 AWS 账户，可信账户中的 IAM 管理员还必须向委托人实体（用户或角色）授予访问资源的权限。他们通过将基于身份的策略附加到实体以授予权限。但是，如果基于资源的策略向同一个账户中的主体授予访问权限，则不需要额外的基于身份的策略。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

Amazon Location 的策略操作

支持策略操作：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

JSON 策略的 `Action` 元素描述可用于在策略中允许或拒绝访问的操作。策略操作通常与关联的 AWS API 操作同名。有一些例外情况，例如没有匹配 API 操作的仅限权限 操作。还有一些操作需要在策略中执行多个操作。这些附加操作称为相关操作。

在策略中包含操作以授予执行关联操作的权限。

有关 Amazon Location 操作的列表，请参阅《服务授权参考》中的 Amazon Location Service 定义的操作。

Amazon Location 中的策略操作在操作前面使用以下前缀：

```
geo
```

要在单个语句中指定多项操作，请使用逗号将它们隔开。

```
"Action": [
      "geo:action1",
      "geo:action2"
         ]
```

您也可以使用通配符（*）指定多个操作。例如，要指定以单词 `Get` 开头的所有操作，包括以下操作：

```
"Action": "geo:Get*"
```

要查看 Amazon Location 基于身份的策略的示例，请参阅 Amazon Location Service 基于身份的策略示例。

Amazon Location 的策略资源

支持策略资源：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

Resource JSON 策略元素指定要向其应用操作的一个或多个对象。语句必须包含 Resource 或 NotResource 元素。作为最佳实践，请使用其 Amazon 资源名称（ARN）指定资源。对于支持特定资源类型（称为资源级权限）的操作，您可以执行此操作。

对于不支持资源级权限的操作（如列出操作），请使用通配符（*）指示语句应用于所有资源。

```
"Resource": "*"
```

要查看亚马逊定位资源类型及其列表 ARNs，请参阅服务授权参考中的 Amazon Location Service 定义的资源。要了解您可以使用哪些操作指定每个资源的 ARN，请参阅 Amazon Location Service 定义的操作。

要查看 Amazon Location 基于身份的策略的示例，请参阅 Amazon Location Service 基于身份的策略示例。

Amazon Location 的策略条件键

支持特定于服务的策略条件键：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

在 Condition 元素（或 Condition 块）中，可以指定语句生效的条件。Condition 元素是可选的。您可以创建使用条件运算符（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。

如果您在一个语句中指定多个 Condition 元素，或在单个 Condition 元素中指定多个键，则 AWS 使用逻辑 AND 运算评估它们。如果您为单个条件键指定多个值，则使用逻辑OR运算来 AWS 评估条件。在授予语句的权限之前必须满足所有的条件。

在指定条件时，您也可以使用占位符变量。例如，只有在使用 IAM 用户名标记 IAM 用户时，您才能为其授予访问资源的权限。有关更多信息，请参阅《IAM 用户指南》中的 IAM 策略元素：变量和标签。

AWS 支持全局条件密钥和特定于服务的条件密钥。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的AWS 全局条件上下文密钥。

要查看 Amazon Location 条件键的列表，请参阅《服务授权参考》中的 Amazon Location Service 的条件键。要了解您可以对哪些操作和资源使用条件键，请参阅 Amazon Location Service 定义的操作。

Amazon Location 支持条件键，允许您在政策声明中允许或拒绝访问特定地理围栏或设备。以下条件键可用于：

- `geo:GeofenceIds` 用于地理围栏操作。类型是 `ArrayOfString`。
- `geo:DeviceIds` 用于跟踪器操作。类型是 `ArrayOfString`。

以下操作可在您的 IAM policy 里与 `geo:GeofenceIds` 操作配合使用：

- `BatchDeleteGeofences`
- `BatchPutGeofences`
- `GetGeofence`
- `PutGeofence`

以下操作可在您的 IAM policy 里与 `geo:DeviceIds` 操作配合使用：

- `BatchDeleteDevicePositionHistory`
- `BatchGetDevicePosition`
- `BatchUpdateDevicePosition`
- `GetDevicePosition`
- `GetDevicePositionHistory`

> **Note**
>
> 您不能将这些条件键与 `BatchEvaluateGeofences`、`ListGeofences` 或 `ListDevicePosition` 操作配合使用。

要查看 Amazon Location 基于身份的策略的示例，请参阅 Amazon Location Service 基于身份的策略示例。

ACLs 在亚马逊所在地

支持 ACLs：否

访问控制列表 (ACLs) 控制哪些委托人（账户成员、用户或角色）有权访问资源。 ACLs 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

使用 Amazon Location 的 ABAC

支持 ABAC（策略中的标签）：是

基于属性的访问控制（ABAC）是一种授权策略，该策略基于属性来定义权限。在中 AWS，这些属性称为标签。您可以向 IAM 实体（用户或角色）和许多 AWS 资源附加标签。标记实体和资源是 ABAC 的第一步。然后设计 ABAC 策略，以在主体的标签与他们尝试访问的资源标签匹配时允许操作。

ABAC 在快速增长的环境中非常有用，并在策略管理变得繁琐的情况下可以提供帮助。

要基于标签控制访问，您需要使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 条件键在策略的条件元素中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关 ABAC 的更多信息，请参阅《IAM 用户指南》中的使用 ABAC 授权定义权限。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的使用基于属性的访问权限控制（ABAC）。

有关标记 Amazon Location 资源的更多信息，请参阅 the section called “如何使用标签”。

要查看基于身份的策略（用于根据资源上的标签来限制对该资源的访问）的示例，请参阅基于标签控制对资源的访问。

将临时凭证用于 Amazon Location

支持临时凭证：是

当你使用临时证书登录时，有些 AWS 服务 不起作用。有关更多信息，包括哪些 AWS 服务 适用于临时证书，请参阅 IAM 用户指南中的AWS 服务 与 IAM 配合使用的信息。

如果您使用除用户名和密码之外的任何方法登录，则 AWS Management Console 使用的是临时证书。例如，当您 AWS 使用公司的单点登录 (SSO) 链接进行访问时，该过程会自动创建临时证书。当您以用户身份登录控制台，然后切换角色时，您还会自动创建临时凭证。有关切换角色的更多信息，请参阅《IAM 用户指南》中的从用户切换到 IAM 角色（控制台）。

您可以使用 AWS CLI 或 AWS API 手动创建临时证书。然后，您可以使用这些临时证书进行访问 AWS。 AWS 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅 IAM 中的临时安全凭证。

Amazon Location 的跨服务主体权限

支持转发访问会话（FAS）：否

当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务 向下游服务发出请求的请求。 AWS 服务只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两项操作的权限。有关发出 FAS 请求时的策略详情，请参阅转发访问会话。

Amazon Location 的服务角色

支持服务角色：否

服务角色是由一项服务担任、代表您执行操作的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。

> **Warning**
>
> 更改服务角色的权限可能会破坏 Amazon Location 的功能。仅当 Amazon Location 提供相关指导时才编辑服务角色。

Amazon Location 的服务相关角色

支持服务相关角色：否

服务相关角色是一种与服务相关联的 AWS 服务服务角色。服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户 ，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。

有关创建或管理服务相关角色的详细信息，请参阅能够与 IAM 搭配使用的AWS 服务。在表中查找服务相关角色列中包含 Yes 的表。选择是链接以查看该服务的服务相关角色文档。

Amazon Location Service 如何处理未经身份验证的用户

使用 Amazon Location Service 的许多场景，包括在网络上或移动应用程序中显示地图，都需要允许尚未使用 IAM 登录的用户进行访问。对于这些未经身份验证的场景，您有两种选择。

- 使用 API 密钥 – 要向未经身份验证的用户授予访问权限，您可以创建 API 密钥来授予对您的 Amazon Location Service 资源的只读访问权限。在您不想对每个用户进行身份验证的情况下，这很有用。例如，Web 应用程序。有关 API 密钥的更多信息，请参阅 the section called “使用 API 密钥”。
- 使用 Amazon Cognito – API 密钥的替代方法是使用 Amazon Cognito 授予匿名访问权限。Amazon Cognito 允许您使用 IAM policy 创建更丰富的授权，以定义未经身份验证的用户可以执行的操作。有关使用 Amazon Cognito 的更多信息，请参阅 the section called “在 Web 中使用身份池”。

有关向未经身份验证的用户提供访问权限的概述，请参阅 身份验证。

Amazon Location Service 基于身份的策略示例

默认情况下，用户和角色没有创建或修改 Amazon Location 资源的权限。他们也无法使用 AWS Management Console、 AWS Command Line Interface (AWS CLI) 或 AWS API 执行任务。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。管理员随后可以向角色添加 IAM 策略，用户可以代入角色。

要了解如何使用这些示例 JSON 策略文档创建基于 IAM 身份的策略，请参阅《IAM 用户指南》中的创建 IAM 策略（控制台）。

有关 Amazon Location 定义的操作和资源类型（包括每种资源类型的格式）的详细信息，请参阅《服务授权参考》中的 Amazon Location Service 的操作、资源和条件密钥。 ARNs

主题

- 策略最佳实践
- 使用 Amazon Location 控制
- 允许用户查看他们自己的权限
- 在策略中使用 Amazon Location Service 资源
- 更新设备位置的权限

- 跟踪器资源的只读策略
- 创建地理围栏的策略
- 地理围栏的只读策略
- 渲染地图资源的权限
- 允许搜索操作的权限
- 路线计算器的只读策略
- 根据条件键控制资源访问权限
- 基于标签控制对资源的访问

策略最佳实践

基于身份的策略确定某个人是否可以创建、访问或删除您账户中的 Amazon Location 资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下指南和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的AWS 托管式策略或工作职能的AWS 托管式策略。
- 应用最低权限：在使用 IAM 策略设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 IAM 中的策略和权限。
- 使用 IAM 策略中的条件进一步限制访问权限：您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定的方式使用的，则也可以使用条件来授予对服务操作的访问权限 AWS 服务，例如 AWS CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的 IAM JSON 策略元素：条件。
- 使用 IAM Access Analyzer 验证您的 IAM 策略，以确保权限的安全性和功能性 – IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言（JSON）和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的使用 IAM Access Analyzer 验证策略。
- 需要多重身份验证 (MFA)-如果 AWS 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的使用 MFA 保护 API 访问。

有关 IAM 中的最佳实操的更多信息，请参阅《IAM 用户指南》中的 IAM 中的安全最佳实践。

使用 Amazon Location 控制

要访问 Amazon Location Service 控制台，您必须具有一组最低的权限。这些权限必须允许您列出和查看有关您的 Amazon Location 资源的详细信息 AWS 账户。如果创建比必需的最低权限更为严格的基于身份的策略，对于附加了该策略的实体（用户或角色），控制台将无法按预期正常运行。

对于仅调用 AWS CLI 或 AWS API 的用户，您无需为其设置最低控制台权限。相反，只允许访问与其尝试执行的 API 操作相匹配的操作。

为确保用户和角色可以使用 Amazon Location 控制台，请将以下策略添加到实体上。有关更多信息，请参阅《IAM 用户指南》中的为用户添加权限。

以下政策允许访问 Amazon Location Service 控制台，以便能够在您的 AWS 账户中创建、删除、列出和查看有关 Amazon Location 资源的详细信息。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GeoPowerUser",
      "Effect": "Allow",
      "Action": [
        "geo:*",
        "geo-maps:*",
        "geo-places:*",
        "geo-routes:*"
      ],
      "Resource": "*"
    }
  ]
}
```

或者，您可以授予只读权限以简化只读访问。使用只读权限时，如果用户尝试写入操作（例如创建或删除资源），则会显示一条错误消息。有关示例，请参阅 the section called “跟踪器的只读策略”。

允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管式策略。此策略包括在控制台上或使用 AWS CLI 或 AWS API 以编程方式完成此操作的权限。

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

在策略中使用 Amazon Location Service 资源

Amazon Location Service 对资源使用以下前缀：

Amazon Location 资源前缀

资源	资源前缀
地图资源	`map`
放置资源	`place-index`

资源	资源前缀
路线资源	route-calculator
跟踪器资源	tracker
地理围栏集合资源	geofence-collection

使用下面的 ARN 语法：

```
arn:Partition:geo:Region:Account:ResourcePrefix/ResourceName
```

有关格式的更多信息 ARNs，请参阅 Amazon 资源名称 (ARNs) 和 AWS 服务命名空间。

示例

- 使用以下 ARN 允许访问指定的地图资源。

  ```
  "Resource": "arn:aws:geo:us-west-2:account-id:map/map-resource-name"
  ```

- 要指定对属于特定账户的所有 map 资源的访问，使用通配符（*）：

  ```
  "Resource": "arn:aws:geo:us-west-2:account-id:map/*"
  ```

- 无法对特定资源执行某些 Amazon Location 操作，例如用于创建资源的操作。在这些情况下，您必须使用通配符 （*)。

  ```
  "Resource": "*"
  ```

要查看亚马逊定位资源类型及其列表 ARNs，请参阅服务授权参考中的 Amazon Location Service 定义的资源。要了解您可以使用哪些操作指定每个资源的 ARN，请参阅 Amazon Location Service 定义的操作。

更新设备位置的权限

要更新多个跟踪器的设备位置，您需要向用户授予访问您的一个或多个跟踪器资源的权限。您还希望允许用户更新一批设备位置。

在此示例中，除了授予对*Tracker1*和*Tracker2*资源的访问权限外，以下策略还授予对*Tracker1*和*Tracker2*资源使用geo:BatchUpdateDevicePosition操作的权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UpdateDevicePositions",
      "Effect": "Allow",
      "Action": [
        "geo:BatchUpdateDevicePosition"
      ],
      "Resource": [
        "arn:aws:geo:us-west-2:account-id:tracker/Tracker1",
        "arn:aws:geo:us-west-2:account-id:tracker/Tracker2"
      ]
    }
  ]
}
```

如果您想限制用户只能更新特定设备的设备位置，则可以为该设备 ID 添加条件键。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UpdateDevicePositions",
      "Effect": "Allow",
      "Action": [
        "geo:BatchUpdateDevicePosition"
      ],
      "Resource": [
        "arn:aws:geo:us-west-2:account-id:tracker/Tracker1",
        "arn:aws:geo:us-west-2:account-id:tracker/Tracker2"
      ],
      "Condition":{
        "ForAllValues:StringLike":{
          "geo:DeviceIds":[
            "deviceId"
          ]
        }
      }
    }
```

```
  ]
}
```

跟踪器资源的只读策略

要为 AWS 账户中的所有跟踪器资源创建只读政策，您需要授予对所有跟踪器资源的访问权限。您还需要授予用户访问操作的权限，这些操作允许他们获取多台设备的设备位置，从一台设备获取设备位置并获取位置历史记录。

在这个示例中，以下策略授予对以下操作的权限：

- geo:BatchGetDevicePosition 检索多个设备的位置。
- geo:GetDevicePosition 检索单个设备的位置。
- geo:GetDevicePositionHistory 检索设备的位置历史记录。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GetDevicePositions",
      "Effect": "Allow",
      "Action": [
        "geo:BatchGetDevicePosition",
        "geo:GetDevicePosition",
        "geo:GetDevicePositionHistory"
      ],
      "Resource": "arn:aws:geo:us-west-2:account-id:tracker/*"
    }
  ]
}
```

创建地理围栏的策略

要创建允许用户创建地理围栏的策略，您需要向允许用户在地理围栏集合上创建一个或多个地理围栏的特定操作授予访问权限。

以下政策授予对以下操作的权限*Collection*：

- geo:BatchPutGeofence 创建多个地理围栏。

- geo:PutGeofence 创建单个地理围栏。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateGeofences",
      "Effect": "Allow",
      "Action": [
        "geo:BatchPutGeofence",
        "geo:PutGeofence"
      ],
      "Resource": "arn:aws:geo:us-west-2:account-id:geofence-collection/Collection"
    }
  ]
}
```

地理围栏的只读策略

要为存储在 AWS 账户地理围栏集合中的地理围栏创建只读策略，您需要授予访问从存储地理围栏的地理围栏集合中读取的操作的访问权限。

以下政策授予对以下操作的权限*Collection*：

- geo:ListGeofences 列出指定地理围栏集合中的地理围栏。
- geo:GetGeofence 从地理围栏集合中检索地理围栏集合。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GetGeofences",
      "Effect": "Allow",
      "Action": [
        "geo:ListGeofences",
        "geo:GetGeofence"
      ],
      "Resource": "arn:aws:geo:us-west-2:account-id:geofence-collection/Collection"
    }
  ]
```

```
}
```

渲染地图资源的权限

要授予足够的权限来渲染地图，你需要授予对地图图块、精灵、字形和样式描述符的访问权限：

- `geo:GetMapTile` 检索用于在地图上有选择地渲染要素的地图图块。
- `geo:GetMapSprites` 检索 PNG 精灵表和描述其中的偏移量的相应的 JSON 文档。
- `geo:GetMapGlyphs` 检索用于显示文本的字形。
- `geo:GetMapStyleDescriptor` 检索地图的样式描述符，其中包含渲染规则。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GetTiles",
      "Effect": "Allow",
      "Action": [
        "geo:GetMapTile",
        "geo:GetMapSprites",
        "geo:GetMapGlyphs",
        "geo:GetMapStyleDescriptor"
      ],
      "Resource": "arn:aws:geo:us-west-2:account-id:map/Map"
    }
  ]
}
```

允许搜索操作的权限

要创建允许搜索操作的政策，您首先需要授予对 AWS 账户中地点索引资源的访问权限。您还需要授予访问权限，这些操作允许用户通过地理编码使用文本进行搜索，并通过反向地理编码使用位置进行搜索。

在此示例中，除了授予访问权限外*PlaceIndex*，以下策略还授予对以下操作的权限：

- `geo:SearchPlaceIndexForPosition` 允许您搜索给定位置附近的地点或兴趣点。
- `geo:SearchPlaceIndexForText` 允许您使用自由格式文本搜索地址、名称、城市或地区。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Search",
      "Effect": "Allow",
      "Action": [
        "geo:SearchPlaceIndexForPosition",
        "geo:SearchPlaceIndexForText"
      ],
      "Resource": "arn:aws:geo:us-west-2:account-id:place-index/PlaceIndex"
    }
  ]
}
```

路线计算器的只读策略

您可以创建只读策略，允许用户访问路线计算器资源来计算路线。

在此示例中，除了授予访问权限外*ExampleCalculator*，以下策略还授予对以下操作的权限：

- `geo:CalculateRoute` 计算给定出发位置、目的地位置和路径点位置列表的路线。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RoutesReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:CalculateRoute"
      ],
      "Resource": "arn:aws:geo:us-west-2:accountID:route-calculator/ExampleCalculator"
    }
  ]
}
```

根据条件键控制资源访问权限

在创建 IAM policy 以授予使用地理围栏或设备位置的访问权限时，您可以使用条件运算符更精确地控制用户可以访问哪些地理围栏或设备。为此，您可以将地理围栏 ID 或设备 ID 包含在策略的 Condition 元素中。

以下示例策略演示了如何创建允许用户更新特定设备的设备位置的策略。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UpdateDevicePositions",
      "Effect": "Allow",
      "Action": [
        "geo:BatchUpdateDevicePosition"
      ],
      "Resource": [
        "arn:aws:geo:us-west-2:account-id:tracker/Tracker"
      ],
      "Condition":{
        "ForAllValues:StringLike":{
          "geo:DeviceIds":[
            "deviceId"
          ]
        }
      }
    }
  ]
}
```

基于标签控制对资源的访问

当您创建 IAM policy 以授予使用您的 Amazon Location 资源的访问权限时，您可以使用基于属性的访问控制来更好地控制用户可以修改、使用或删除哪些资源。为此，您可以将标签信息包含在策略 Condition 元素中，以便根据资源标签控制访问权限。

以下示例策略演示了如何创建允许用户创建地理围栏的策略。这允许以下操作在名为的地理围栏集合上创建一个或多个地理围栏：*Collection*

- geo:BatchPutGeofence 创建多个地理围栏。
- geo:PutGeofence 创建单个地理围栏。

但是，只有当*Collection*标签的值为该用户的用户名时Owner，此策略才使用Condition元素来授予权限。

- 例如，如果名为的用户richard-roe尝试查看 Amazon 地点*Collection*，则*Collection*必须将其标记为Owner=richard-roe或owner=richard-roe。否则，该用户将被拒绝访问。

> **Note**
>
> 条件标签键 Owner 匹配 Owner 和 owner，因为条件键名称不区分大小写。有关更多信息，请参阅 IAM 用户指南 中的 IAM JSON 策略元素：条件。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateGeofencesIfOwner",
      "Effect": "Allow",
      "Action": [
        "geo:BatchPutGeofence",
        "geo:PutGeofence"
      ],
      "Resource": "arn:aws:geo:us-west-2:account-id:geofence-collection/Collection",
      "Condition": {
                "StringEquals": {"geo:ResourceTag/Owner": "${aws:username}"}
      }
    }
  ]
}
```

有关如何根据标签定义访问 AWS 资源的权限的教程，请参阅 AWS Identity and Access Management 用户指南。

Amazon Location Service 身份和访问权限故障排查

使用以下信息可帮助您诊断和修复在使用 Amazon Location 和 IAM 时可能遇到的常见问题。

主题

- 我无权在 Amazon Location 中执行操作
- 我无权执行 iam：PassRole

- 我想允许我以外的人访问我 AWS 账户 的 Amazon Location 资源

我无权在 Amazon Location 中执行操作

如果您收到错误提示，指明您无权执行某个操作，则必须更新策略以允许执行该操作。

当 mateojackson IAM 用户尝试使用控制台查看有关虚构 *my-example-widget* 资源的详细信息，但不拥有虚构 geo:*GetWidget* 权限时，会发生以下示例错误。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 geo:GetWidget on resource: my-example-widget
```

在此情况下，必须更新 mateojackson 用户的策略，以允许使用 geo:*GetWidget* 操作访问 *my-example-widget* 资源。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我无权执行 iam：PassRole

如果您收到一个错误，指明您无权执行 iam:PassRole 操作，则必须更新策略以允许您将角色传递给 Amazon Location。

有些 AWS 服务 允许您将现有角色传递给该服务，而不是创建新的服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为 marymajor 的 IAM 用户尝试使用控制台在 Amazon Location 中执行操作时，会发生以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

在这种情况下，必须更新 Mary 的策略以允许她执行 iam:PassRole 操作。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我想允许我以外的人访问我 AWS 账户 的 Amazon Location 资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以代入角色。对于支持基于资源的策略或访问控制列表 (ACLs) 的服务，您可以使用这些策略向人们授予访问您的资源的权限。

要了解更多信息，请参阅以下内容：

- 要了解 Amazon Location 是否支持这些特征，请参阅 Amazon Location Service 如何与 IAM 协同工作。
- 要了解如何提供对您拥有的资源的访问权限 AWS 账户 ，请参阅 IAM 用户指南中的向您拥有 AWS 账户 的另一个 IAM 用户提供访问权限。
- 要了解如何向第三方提供对您的资源的访问权限 AWS 账户，请参阅 IAM 用户指南中的向第三方提供访问权限。 AWS 账户
- 要了解如何通过身份联合验证提供访问权限，请参阅《IAM 用户指南》中的为经过外部身份验证的用户（身份联合验证）提供访问权限。
- 要了解使用角色和基于资源的策略进行跨账户访问之间的差别，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

Amazon Location Service 地图



通过 Amazon Location Service 地图，您可以访问 190 个国家/地区的底层地图数据，每天更新 500 万次。通过静态和动态地图功能，您可以灵活地满足不同的用户需求，并提供与情境相关的身临其境的地图解决方案。

静态地图提供预渲染的地理数据表示形式，使您能够嵌入高质量的视觉效果，从而增强报告并在电子邮件中提供空间背景。动态地图允许您创建交互式和响应式体验，用户可以在其中实时平移、缩放和浏览地图，同时与您的业务要求和偏好保持一致。无论您是显示实时 turn-by-turn导航、可视化基于位置的数据，还是让用户探索新领域，亚马逊的服务都能为您提供工具，以提供能引起受众共鸣的量身定制的解决方案。

特征

Amazon Location Service 提供动态地图和静态地图。

动态地图

> 您可以使用 AWS 地图样式，例如标准、单色、混合和卫星。您可以使用 AWS Map Styles 和地图渲染引擎（例如）将交互式地图添加到应用程序中 MapLibre。动态地图还支持地图手势，例如缩放、平移、缓动、飞行、俯仰、旋转和方位。

静态地图

> 无需地图 URLs 渲染引擎，即可使用静态地图在网站、报告或电子邮件中嵌入简单的地图图像。静态地图允许您根据应用程序的需要叠加标记（或图钉）、路线和多边形区域。

政治观点

> 要从国际视角切换到特定国家的地缘政治视角，请在 API 查询中使用政治观点参数。这有助于企业遵守当地法律，因为某些国家/地区要求在地图和地图数据方面遵守其特定的地缘政治观点。

常见使用案例

在您的应用程序中嵌入地图

在您的应用程序中构建地图，以创建基于位置的体验。可视化营业地点，搜索兴趣点，并帮助用户查找特定地址。启用无缝位置共享和地理标记功能，以吸引客户。使用全面的地图数据、强大的地理编码和灵活的渲染来创建根据您的需求量身定制的自定义交互式地图。无论您是在构建名录、拼车应用程序还是社交平台，都可以将动态、高质量的地图体验集成到您的应用程序中，从而提高用户参与度和业务洞察力。

用于报告或打印的静态地图

将街道地图、卫星图像和基于位置的视觉效果的图像无缝添加到您的网站、文档和应用程序中。静态地图使您能够创建可自定义的地图图像，从而提供地理背景，而无需复杂的客户端渲染。在收据上显示送货路线，在文档中包含位置详细信息，或者将地图集成到您的数字体验中。

分析和可视化数据

将您的数据叠加到高质量的地图上，以发现变革性的空间模式和趋势。让您的团队能够使用您的地理数据创建可自定义的交互式地图可视化效果。使用地图和您的数据来优化选址、规划基础设施或分析市场机会。

增强房地产体验

为潜在买家提供房地产清单的全面位置背景。显示房产的确切位置以及周边社区的详细信息，例如管辖区边界、当地企业、公园和学校。帮助买家找到前往开放日的路线。创建内容丰富、以位置为中心的房地产体验，吸引客户并向其提供信息。

打造引人入胜的旅行体验

显示展示目的地的动态地图，包括详细的街景和主要的地理特征。为游客和旅行者重点介绍酒店、餐厅和其他景点。规划户外设施，例如远足径，以帮助用户规划理想的行程。

使用地图来支持灾难响应工作

在危机期间，及时和准确的位置信息至关重要。使用测绘功能构建网站和应用程序，在火灾、飓风和洪水等即将发生的灾难期间为社区提供基本背景信息。显示显示疏散路线、安全避难所、封路和交通拥堵的动态地图，以帮助社区快速评估情况并做出明智的决定。

独立地图 APIs

API 名称	简短描述	资源
GetStyleDescriptor	检索可应用于地图的可用地图样式，例如标准、单色、混合和卫星。	AWS 地图样式和自定义
GetTile	根据指定的样式和缩放级别获取单个地图图块，从而可以在不同的细节层面上渲染地图。	瓷砖
GetStaticMap	根据特定的坐标和参数生成静态地图图像，这对于在报告或电子邮件中嵌入地图非常有用。	the section called “静态地图”

显示地图

主题	简短描述	资源
为动态地图设置样式	Amazon Location Service 提供了两种设置动态地图样式的选项，即使用预先设计的地 AWS 图样式和使用样式描述符自定义地图样式。	the section called “为动态地图设置样式” the section called “标准地图风格” the section called “单色地图风格” the section called “混合地图风格”
渲染动态地图	Amazon Location Service 建议使用MapLibre渲染引擎渲染地图。 MapLibre是用于在 Web	the section called “按语言划分的地图渲染 SDK”

主题	简短描述	资源
	或移动应用程序中显示地图的引擎。	
自定义静态地图	如何自定义使用 Amazon Location Service 生成的静态地图。	the section called “自定义静态地图”
叠加静态地图	叠加在静态地图上以增强地图的视觉表现形式。	the section called “叠加在静态地图上”

地图概念

本部分提供使用亚马逊定位服务的地图功能的基础知识。它详细介绍了各种可用的地图设计选项，还说明了如何利用 AWS 地图样式和设计原则自定义地图的外观和风格，从而确保应用程序的视觉一致性和品牌形象。

此外，本主题还涵盖了关键的地图概念，例如地图术语、本地化和国际化。

主题

- 地图术语
- 本地化和国际化

地图术语

本节定义了与 Amazon Location Service Maps 相关的关键术语，提供了对底图、矢量和栅格数据、地图渲染和地图样式等核心概念的见解。了解这些术语对于有效利用 Amazon Location Service 地图 APIs 和资源至关重要。

底图

底图为您的地图提供地理环境，存储为矢量切片图层。这些切片图层包括街道名称、建筑物和土地利用等地理特征，以供视觉参考。

Vector

矢量数据由点、线和面组成，用于在地图上表示道路、位置和区域。矢量形状也可以用作地图上标记的图标。

栅格

栅格数据是由网格组成的影像数据，通常表示地形、卫星影像或热图等连续数据。光栅图像也可以用作地图上的图标或纹理。

地图渲染

地图渲染库在运行时从 Amazon Location Service 提取数据，根据所选地图资源渲染地图。地图资源定义了数据提供者和地图样式。Amazon Location Service 建议使用 MapLibre 渲染引擎。

矢量图块

矢量切片是一种使用矢量形状存储地图数据的格式。它会根据显示分辨率进行调整并有选择地渲染功能，同时保持较小的文件大小以获得最佳性能。支持的格式：Mapbox 矢量图块 (MVT)。

地图样式

地图样式定义地图数据的颜色和其他样式信息，从而决定地图在渲染时的显示方式。Amazon Location Service 提供基于 Mapbox GL 样式规范的地图样式。

字形文件

包含编码的 Unicode 字符的二进制文件，地图渲染器使用它来显示标签。

精灵文件

一种便携式网络图形 (PNG) 图像文件，在 JSON 文件中包含小型光栅图像和相应位置描述。地图渲染器用于在地图上显示图标或纹理。

边界框

边界框由两个对角线角点定义：西北（NW）（左上）和东南（SE）（右下）点。这些点是指定地图空间范围的关键。

本地化和国际化

Amazon Location Service 支持本地化功能，允许开发人员针对特定语言和地区自定义地图。这包括对本地地名的支持以及使用不同语言呈现地图的功能。

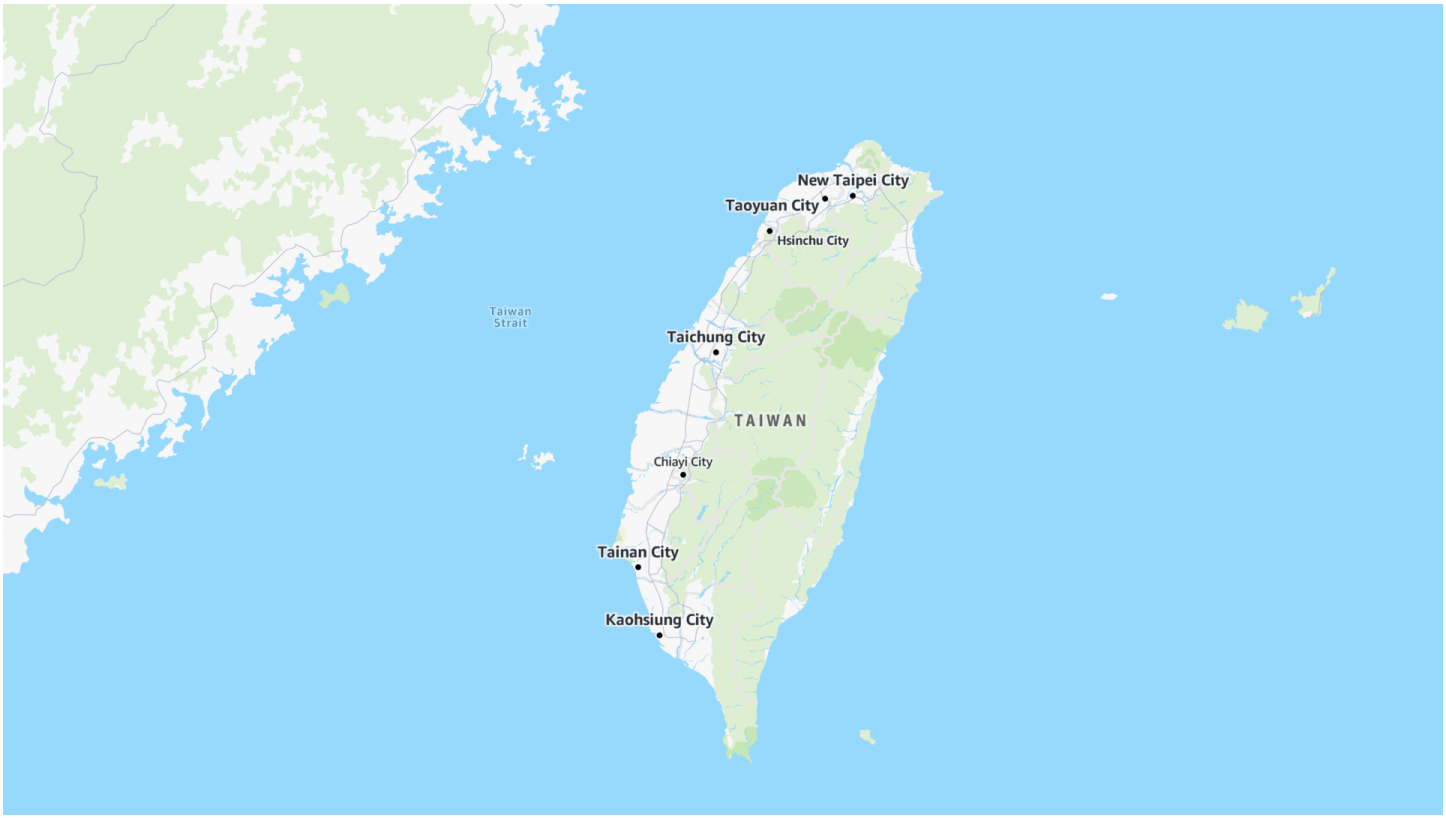
样式	政治观点	语言
Standard	阿根廷、塞浦路斯、埃及、格鲁吉亚、希腊、肯尼亚、摩	通过客户端库支持。

样式	政治观点	语言
	洛哥、巴勒斯坦、塞尔维亚、俄罗斯、苏丹、苏里南、叙利亚、土耳其、坦桑尼亚、乌拉圭	
单色	阿根廷、塞浦路斯、埃及、格鲁吉亚、希腊、肯尼亚、摩洛哥、巴勒斯坦、塞尔维亚、俄罗斯、苏丹、苏里南、叙利亚、土耳其、坦桑尼亚、乌拉圭	通过客户端库支持。
混合	阿根廷、塞浦路斯、埃及、格鲁吉亚、希腊、肯尼亚、摩洛哥、巴勒斯坦、塞尔维亚、俄罗斯、苏丹、苏里南、叙利亚、土耳其、坦桑尼亚、乌拉圭	通过客户端库支持。
卫星	可选	可选

语言

Amazon Locat APIs ion Service 提供的地图允许您自定义地图标签和文本元素的语言。此功能使您的应用程序能够满足使用多种语言的全球受众或地区的需求。通过使用用户的首选语言显示地图，可以增强整体用户体验，使地图更易于访问并与您的多样化用户群相关。

有关更多信息，请参阅 如何为地图设置首选语言。



政治观点

默认情况下，Amazon Location Service 呈现国际视角，用虚线边框直观地表示有争议的领土。要从国际视角切换到特定国家的地缘政治视角，您必须在 API 查询中使用政治观点参数。这有助于企业遵守当地法律，因为某些国家/地区要求在地图和地图数据方面遵守其特定的地缘政治观点。

除了默认的国际视角外，Amazon Location Service 还支持以下国家的地缘政治观点：阿根廷、塞浦路斯、埃及、格鲁吉亚、希腊、肯尼亚、摩洛哥、巴勒斯坦、塞尔维亚、俄罗斯、苏丹、苏里南、叙利亚、土耳其、坦桑尼亚和乌拉圭。要激活地缘政治观点，请将适当的值传递给政治观点参数。

有关更多信息，请参阅 如何设置地图的政治视角。



AWS 地图样式和自定义

地图样式概述

要请求地图，必须先选择地图样式。地图样式定义了渲染地图的视觉外观，包括地图图块、字形和精灵的样式。地图图块可以是矢量 (MVT) 或栅格（图像）。虽然当你放大或缩小时，样式可能会发生变化，但它通常会保持一致的主题。在将样式传递到地图渲染库之前，您可以覆盖部分或整个样式。

AWS 地图样式

AWS 地图样式符合行业标准，提供精致而专业的外观。这些样式减少 time-to-market并消除了专业制图师从头开始创建地图样式的需求。这些预先设计的样式使您能够快速有效地为最终用户创建具有视觉吸引力的地图。

通过利用预先设计 AWS 的地图样式，您可以绕过从头开始设计和构建地图样式的耗时且资源密集型的过程。这可以加快您的开发过程，使您能够专注于核心功能。

地图样式名称	描述	颜色方案	支持
Standard	彩色地图风格	黑暗与光明	动态地图：是，静态地图：否

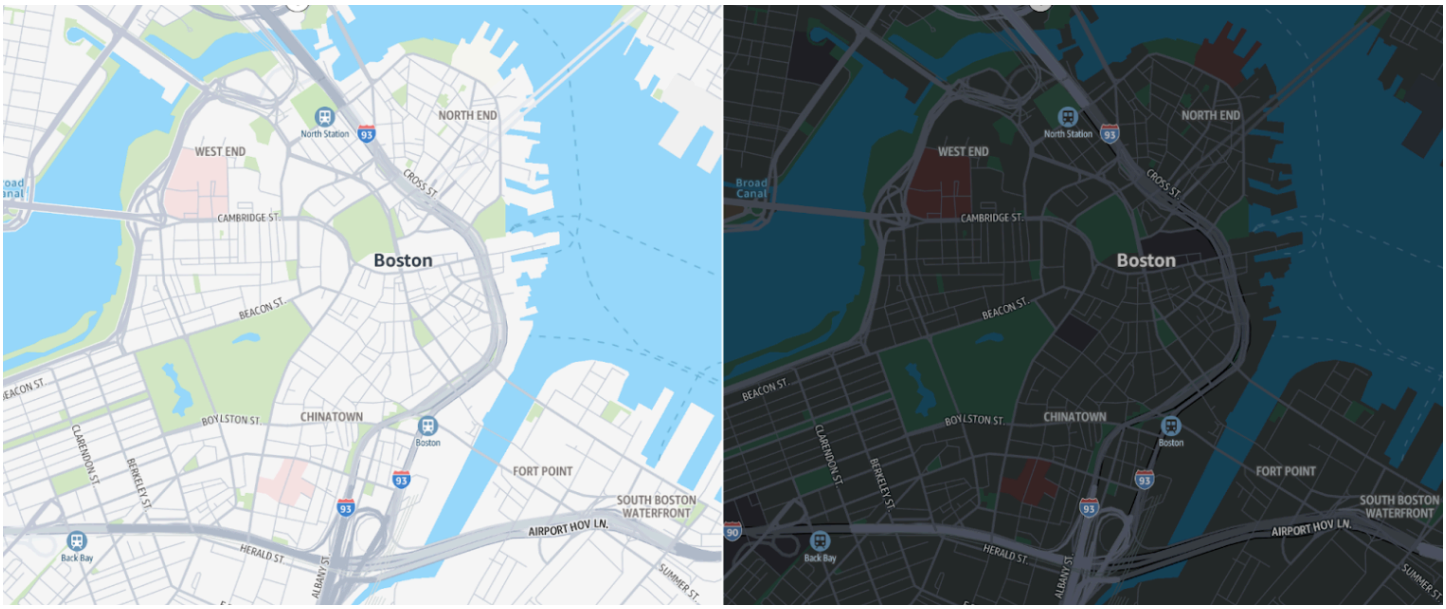
地图样式名称	描述	颜色方案	支持
单色	灰度地图样式	黑暗与光明	动态地图：是，静态地图：否
混合	卫星影像上的道路和标注叠加层	不适用	动态地图：是，静态地图：否
卫星	基于卫星影像的地图风格	不适用	动态地图：是，静态地图：是

Amazon Location Service 提供的款MapLibre GL式符合款式规范。

标准地图风格

标准地图样式是一种简洁而现代的通用地图设计，几乎可以完美地融入任何应用程序或网站。

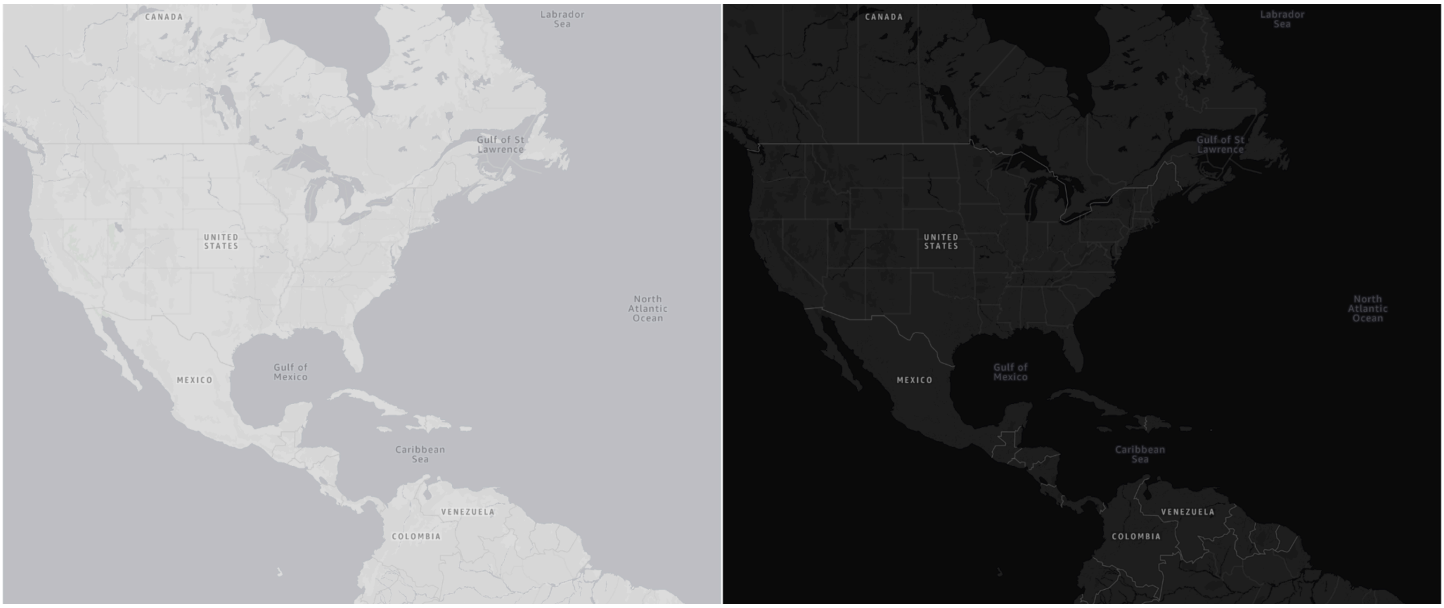
要了解更多信息，请参阅标准地图风格。



单色地图风格

Monochrome 地图样式是一种极简主义画布，具有受限的调色板，专为与数据可视化叠加层一起使用而设计。Monochrome 风格提供浅色和深色模式，可传达地理环境所需的所有基本信息。

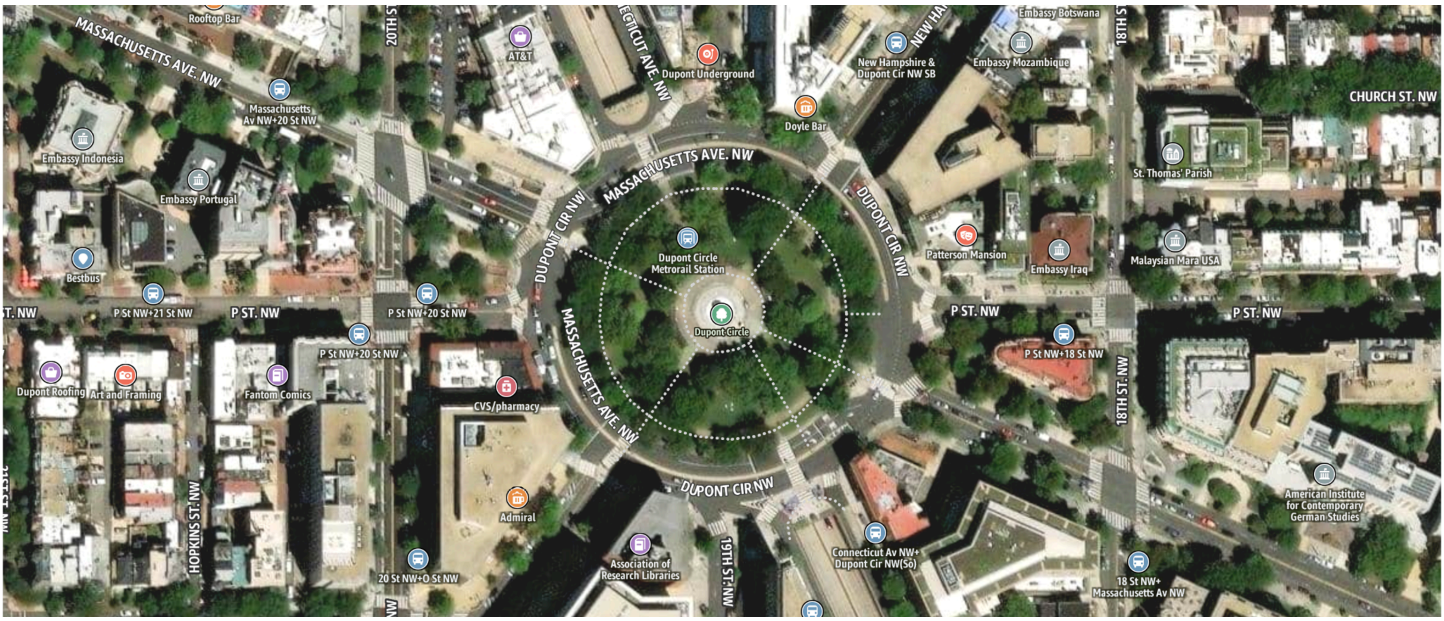
要了解更多信息，请参阅the section called “单色地图风格”。



混合地图风格

混合地图样式将全球卫星图像与矢量地图中的清晰标签和可配置的 POI 类别相结合。

要了解更多信息，请参阅the section called “混合地图风格”。



卫星地图风格

卫星地图风格呈现卫星捕获的高分辨率真实世界图像，提供真实的景观、建筑物和地形视图。这种样式通常包括最少的标签或叠加层，以保持对地理细节的关注。



标准地图风格

标准地图样式提供了简洁、现代、通用的地图设计，几乎可以无缝融入任何应用程序或网站。

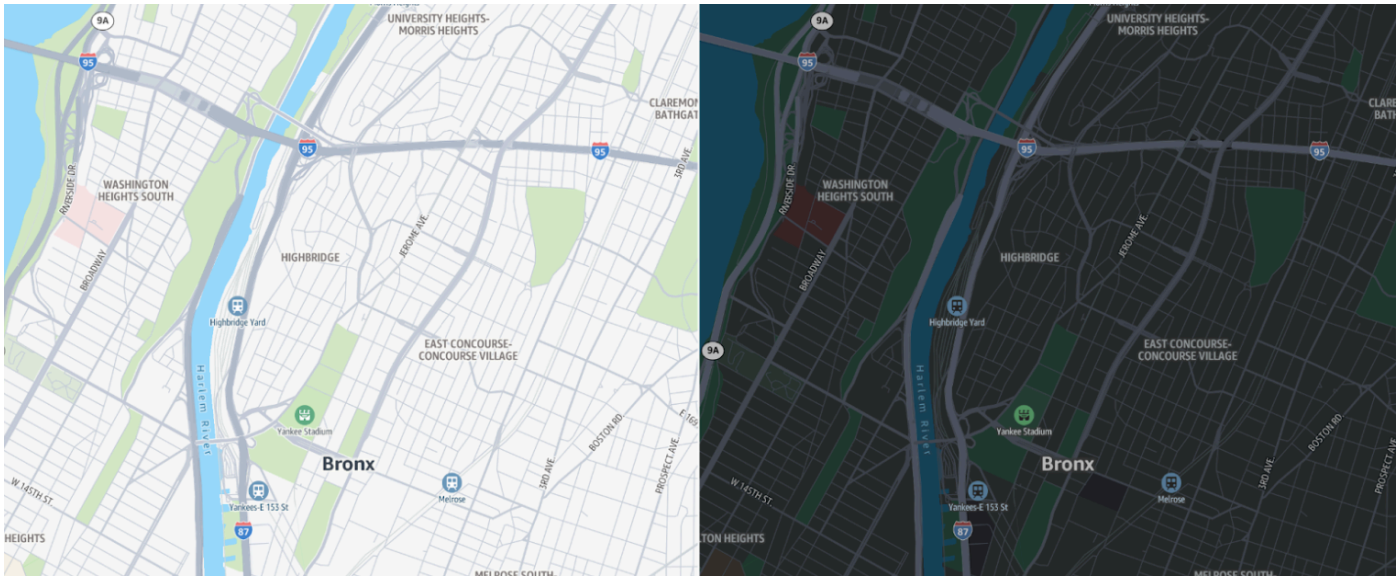
颜色方案

标准地图样式有浅色和深色两种模式。浅色模式用途广泛，可以适应任何环境，而深色模式则具有受限的调色板，旨在清晰地显示细节并在较暗的环境中保持可读性。这样可以最大限度地减少干扰，尤其是在夜间导航等场景中。

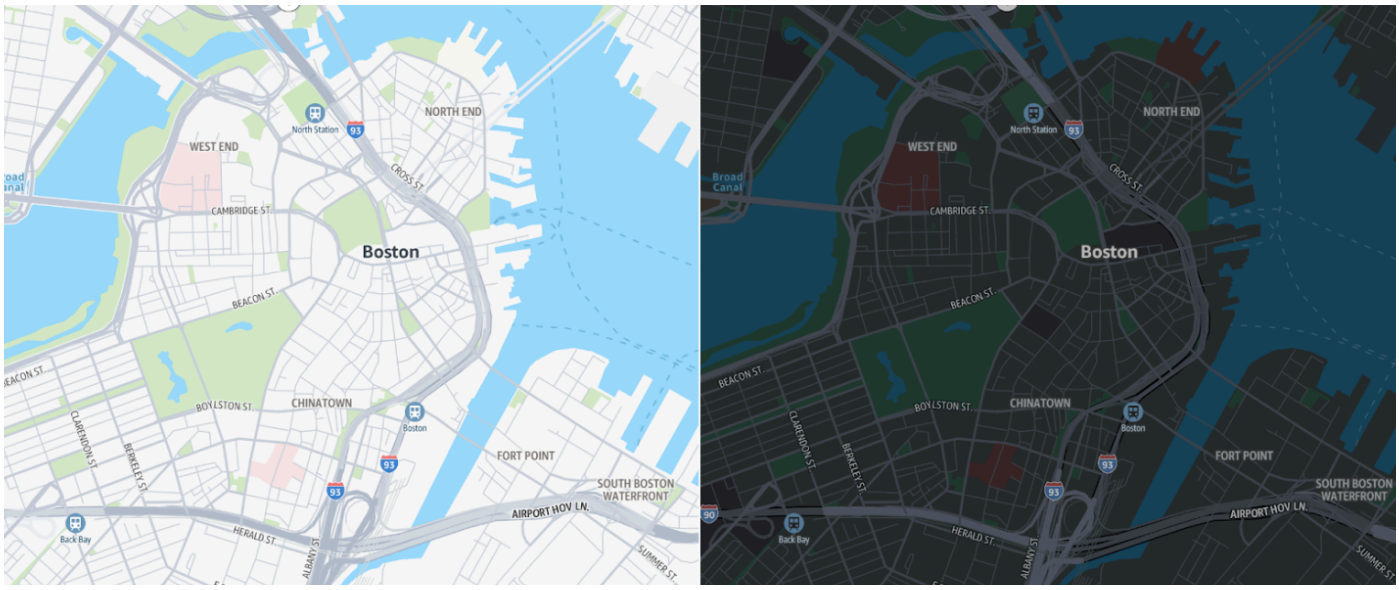
Forest


Vancouver
Surrey
Strait of Juan de Fuca
Seattle
WASHINGTON
Vancouver
Surrey
Strait of Juan de Fuca
Seattle
WASHINGTON

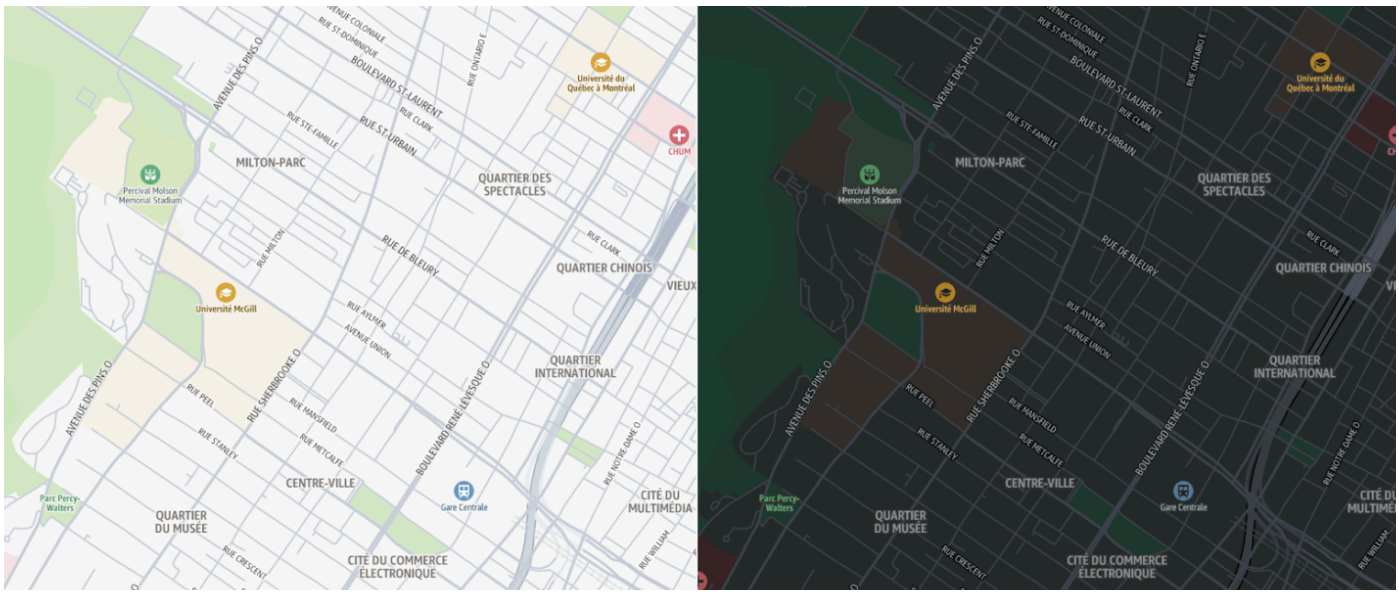
Road


9A
95
UNIVERSITY HEIGHTS-MORRIS HEIGHTS
WASHINGTON HEIGHTS SOUTH
RIVERSIDE DR
BROADWAY
HIGHBRIDGE
JEROME AVE
Highbridge Yard
EAST CONCOURSE-CONCOURSE VILLAGE
Harlem River
Yankee Stadium
Bronx
Melrose
Yankees-E 153 St
W 145TH ST
87
9A
95
UNIVERSITY HEIGHTS-MORRIS HEIGHTS
WASHINGTON HEIGHTS SOUTH
RIVERSIDE DR
BROADWAY
HIGHBRIDGE
JEROME AVE
Highbridge Yard
EAST CONCOURSE-CONCOURSE VILLAGE
Harlem River
Yankee Stadium
Bronx
Melrose
Yankees-E 153 St
W 145TH ST
87

City



Neighborhood

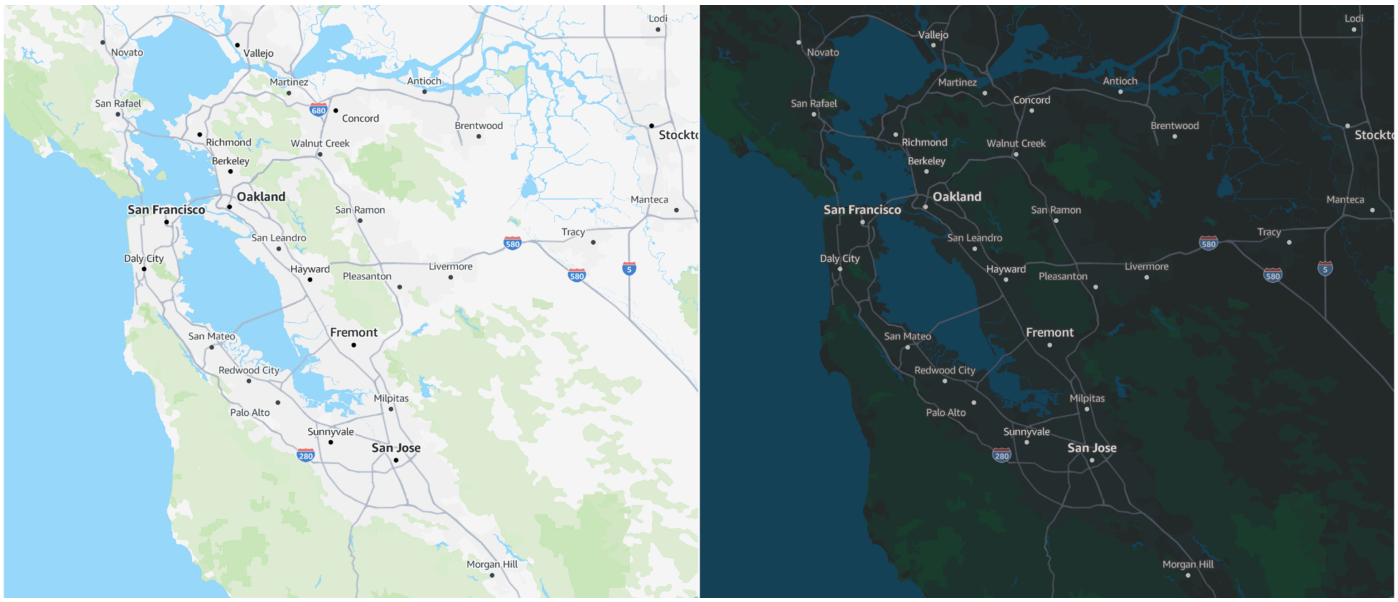


令人愉悦的现代调色板

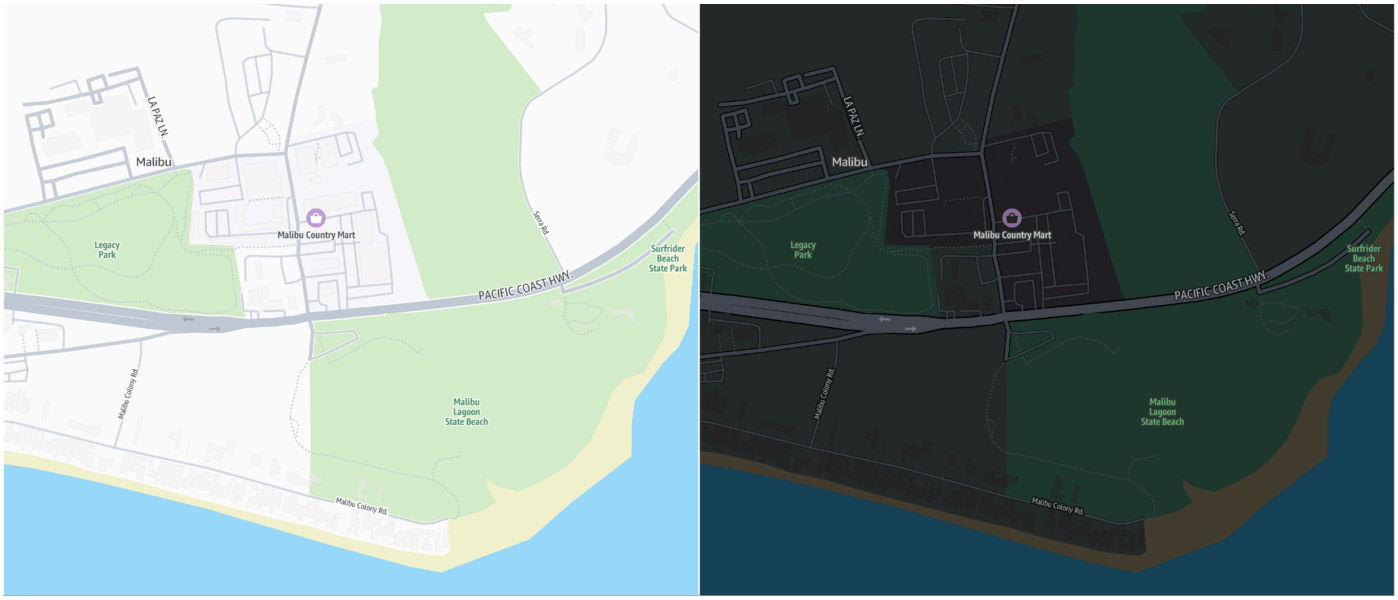
柔和的色彩提供了重要的土地利用背景信息，而不会让地图不堪重负，在高低缩放级别上都提供了有用的信息。放大后，森林、沙漠和冰川等特征为地图增添了丰富性。放大后，一系列颜色会突出显示学校、医院、休闲区（如公园和体育设施）以及商业和工业区等城市地区等重要地标。

整体风格采用连贯的调色板，包括与各自的土地利用区域相得益彰的 POI 标记。道路网络以灰色阴影显示，提供细节，而不会用明亮、分散注意力的颜色压倒地图。

Highway



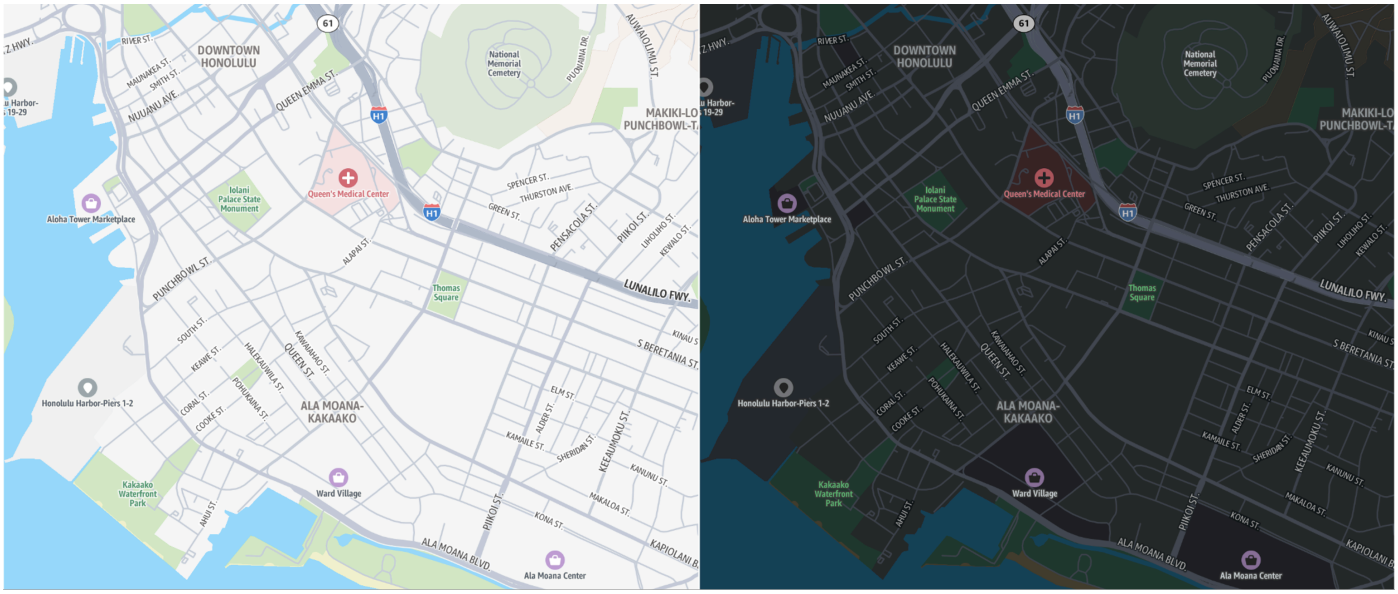
Beach



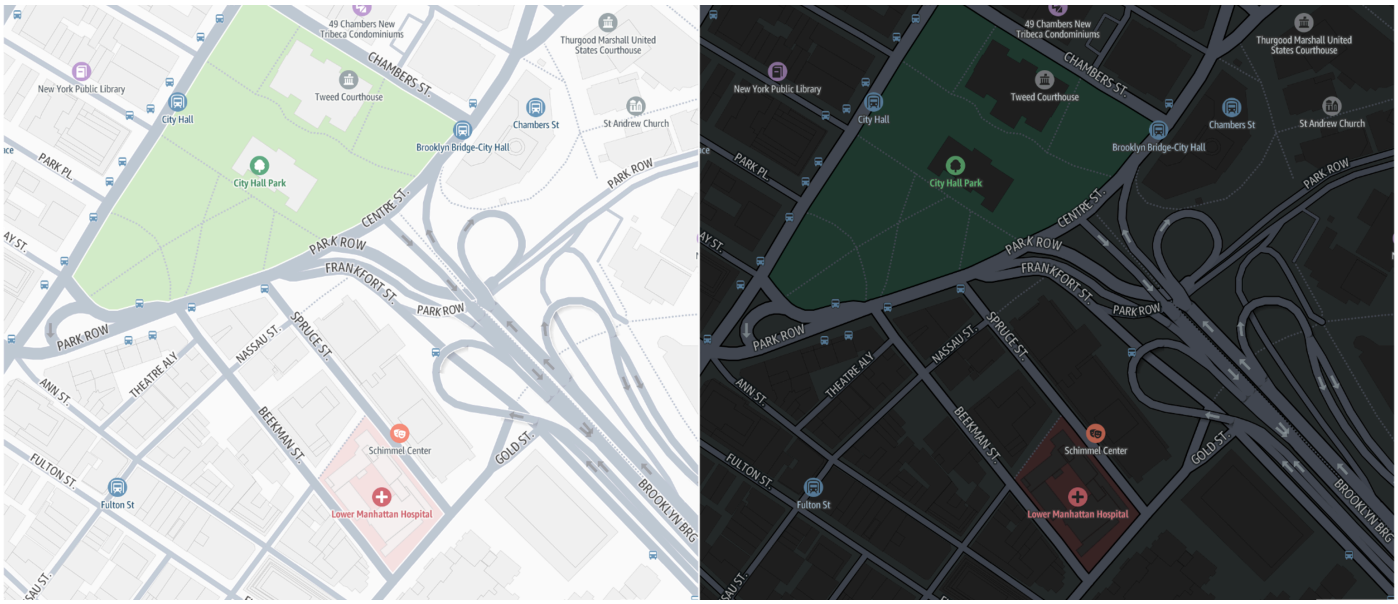
Island



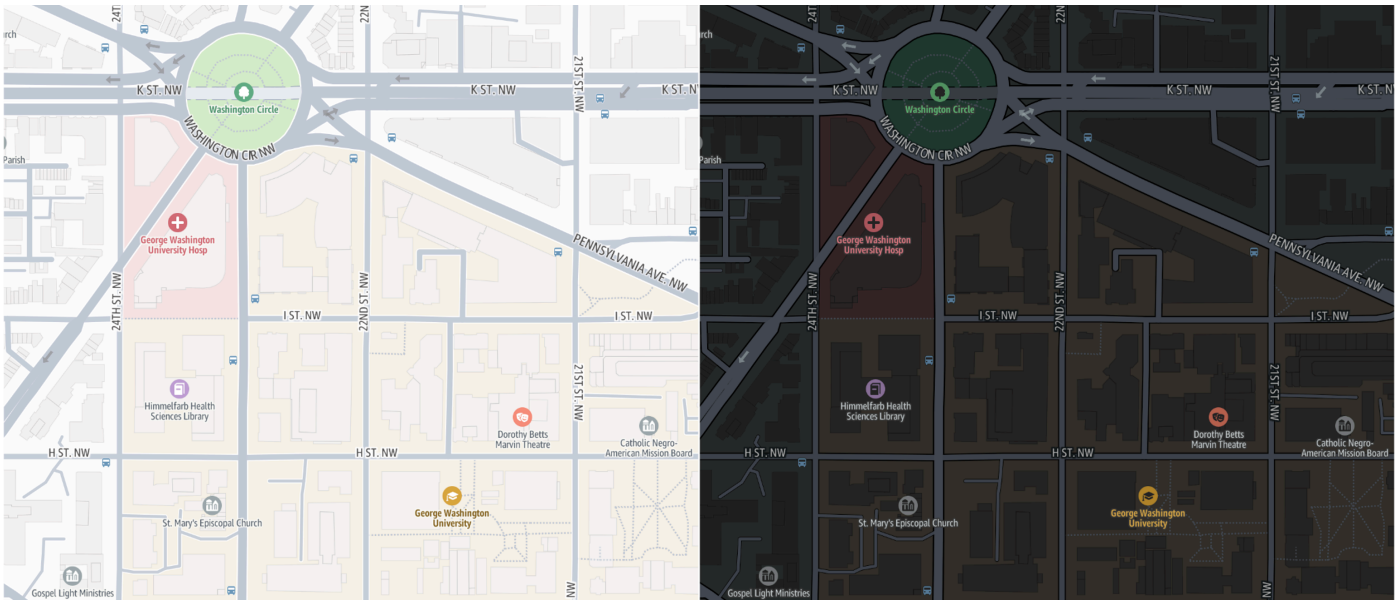
Neighborhood



Intersection

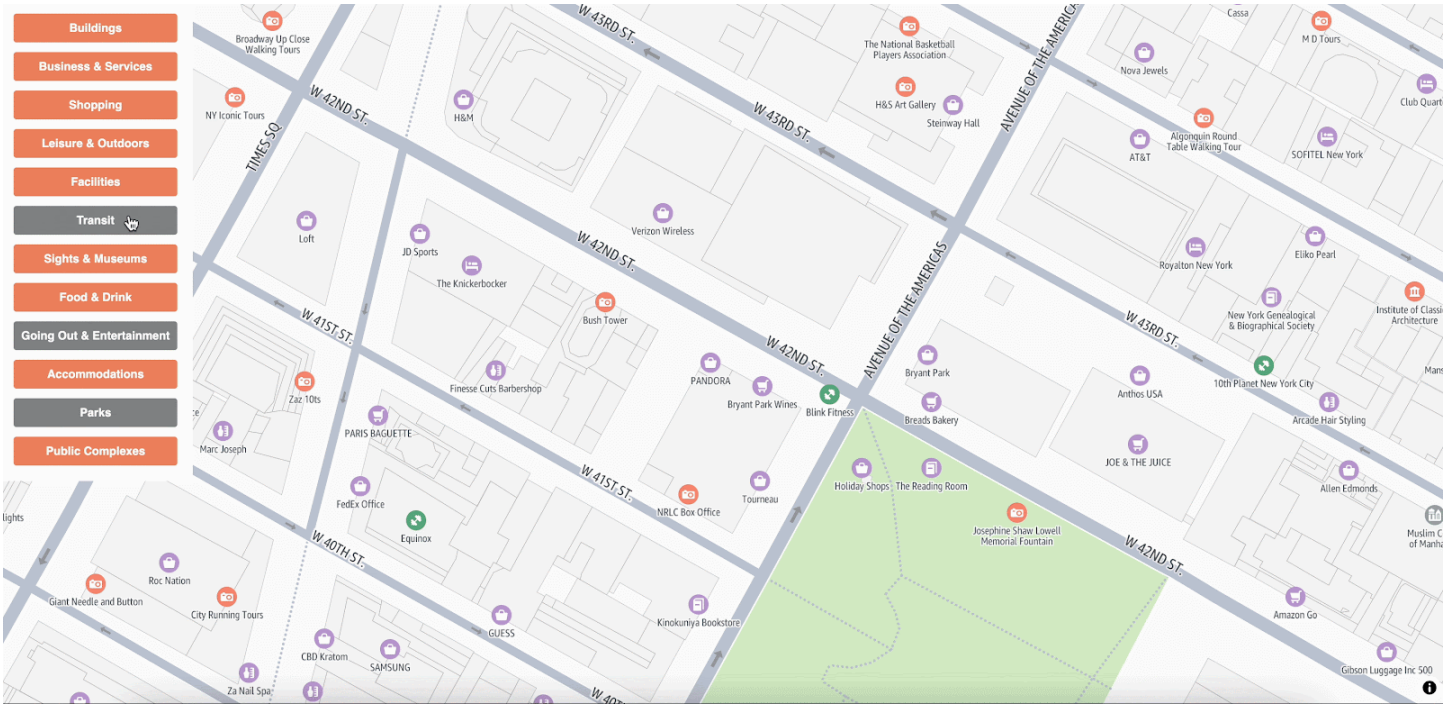


Roundabout



丰富的兴趣点 (POI)

标准地图样式支持大量可配置的兴趣点 (POIs)。只需几行代码，您就可以选择与您的用例相关的 POI 类别。

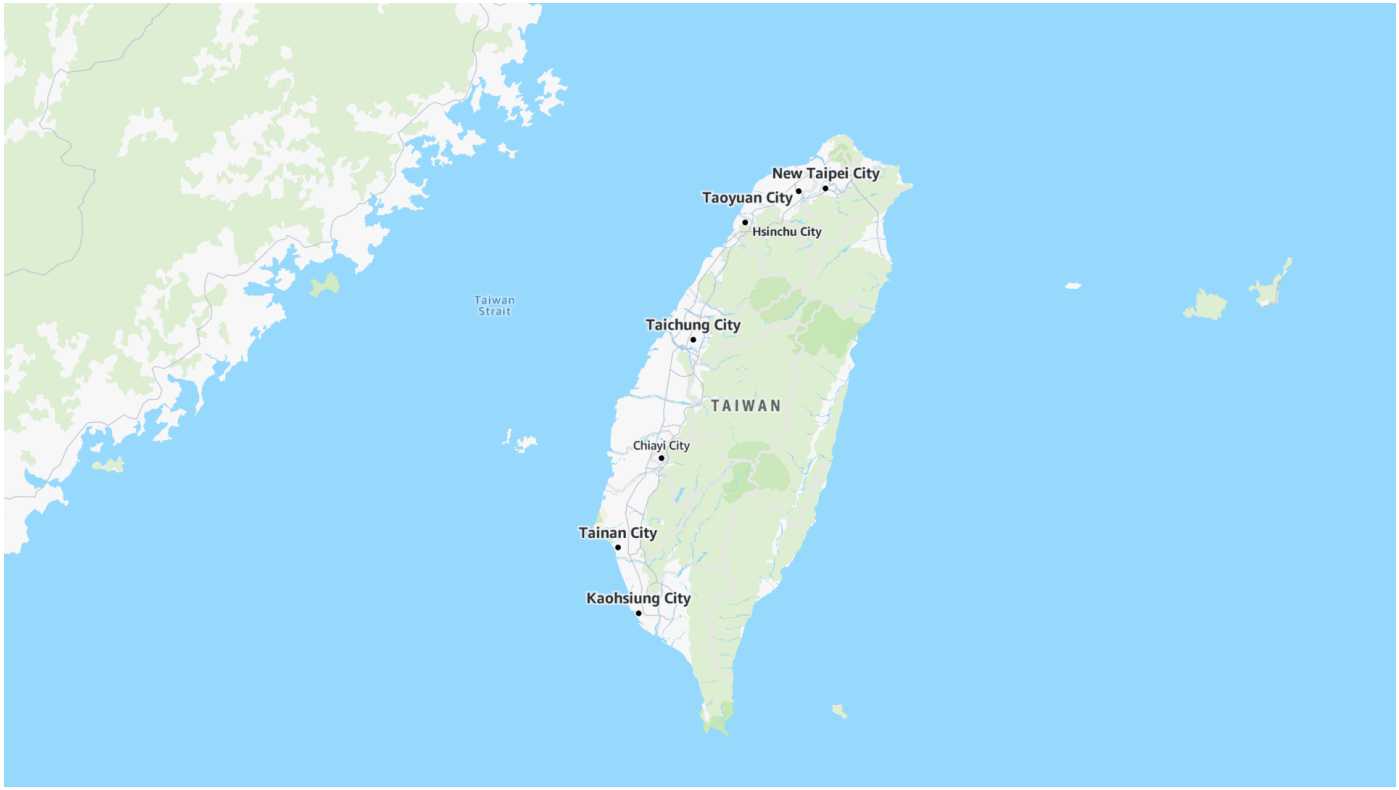


专为世界而设计

标准样式支持不同的政治观点，可确保地图为用户显示正确的边界。该样式还允许轻松切换地图标签的语言，支持数十种语言和书写系统。

要了解更多信息，请参阅the section called “本地化和国际化”。

Languages



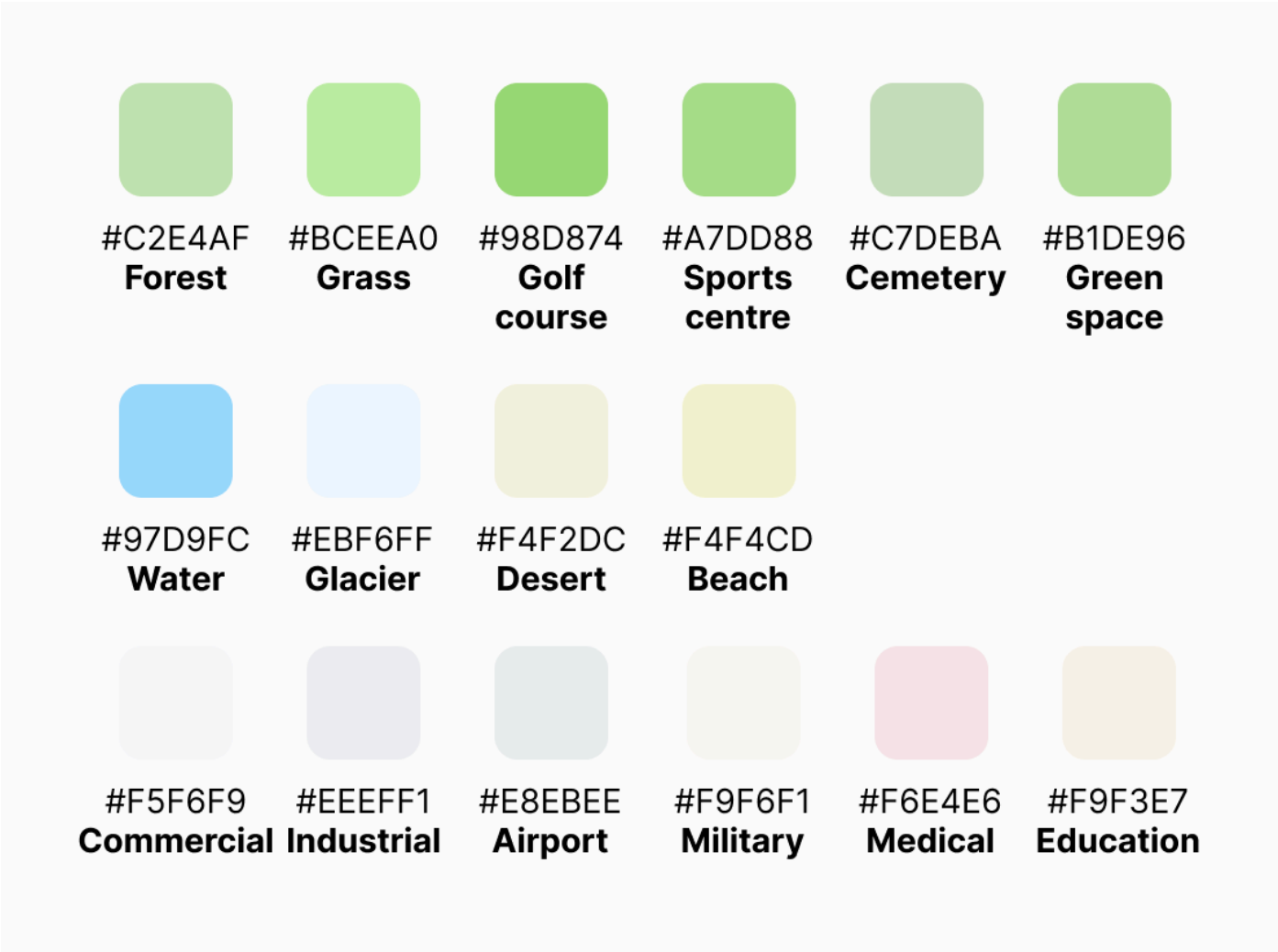
Political view



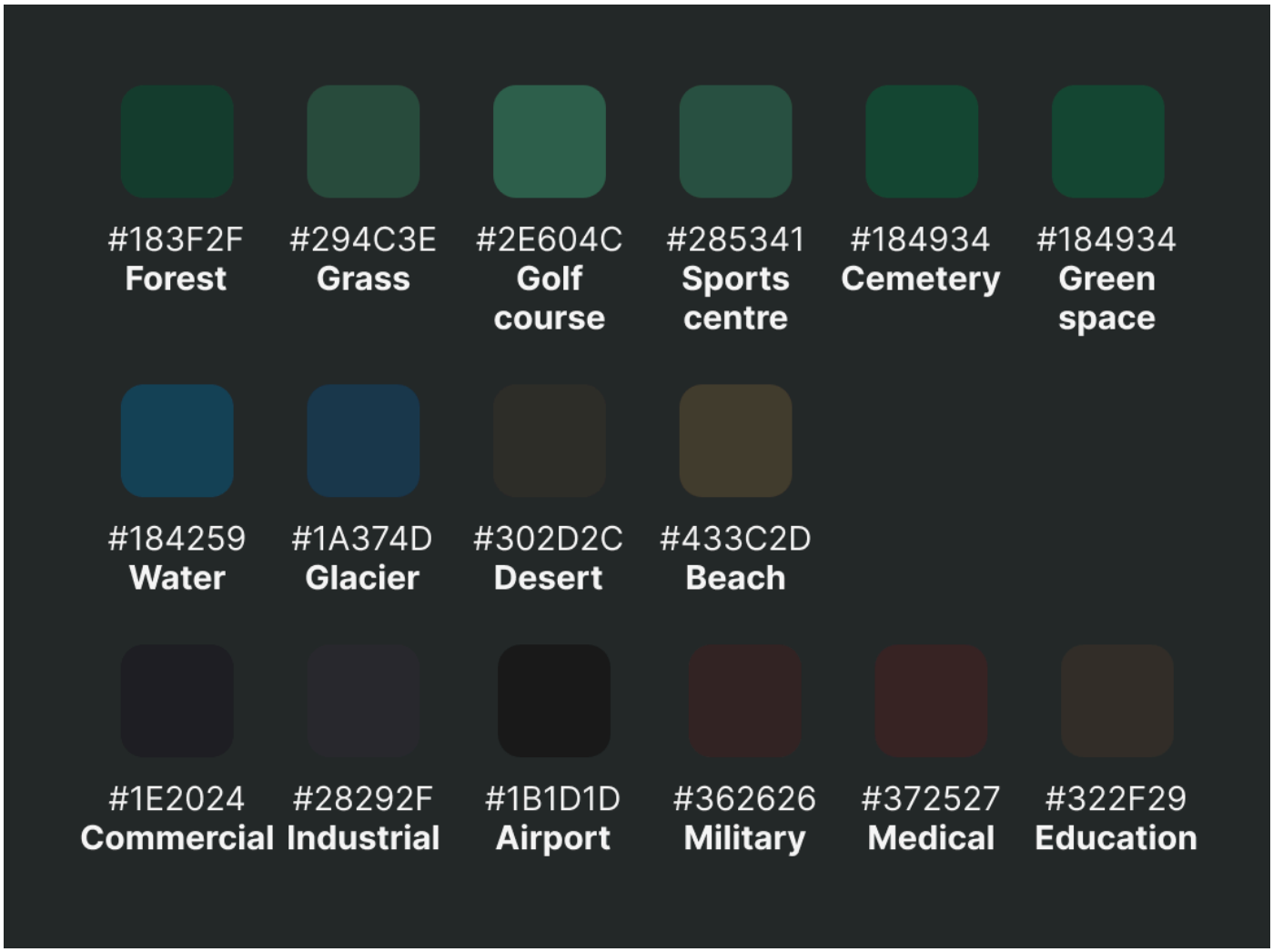
土地利用

标准地图样式使用鲜艳的色彩来表示指定的土地用途。果岭代表森林、草地、高尔夫球场、体育中心和公园。相关的颜色用于水体、冰川、沙漠和海滩。此外，诸如商业、工业、机场、军事区、医疗设施和教育区域等土地用途都以特定的充满活力的类别为特色。

Light



Dark



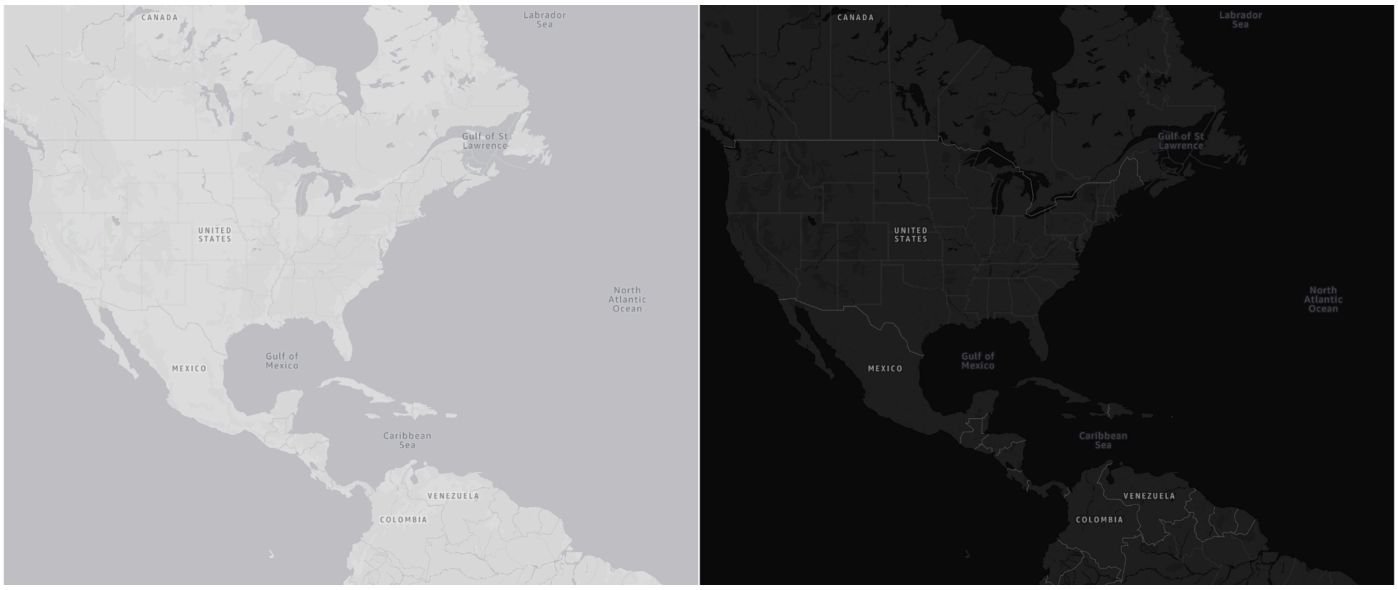
单色地图风格

Monochrome 风格是一幅极简主义画布，具有受限的调色板，专为与数据可视化叠加层一起使用而设计。这种风格支持浅色和深色模式，每种模式都传达了地理环境所需的所有基本信息。

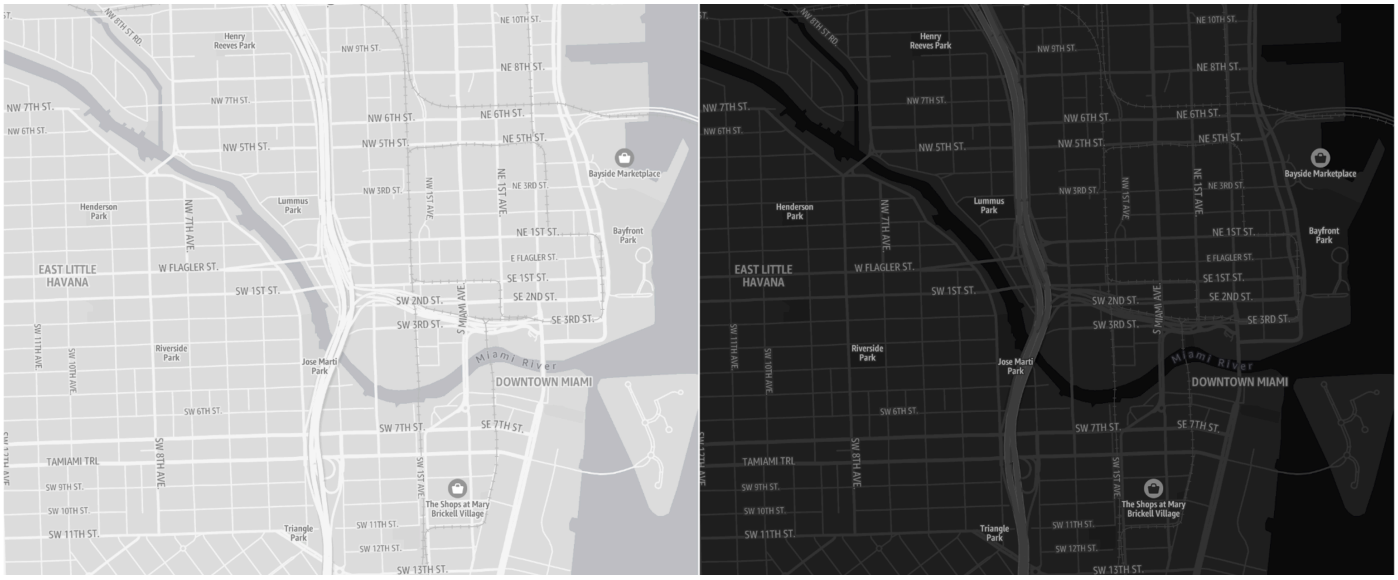
配色方案

单色样式为深色和浅色模式提供颜色选择。

Continent



Neighborhood

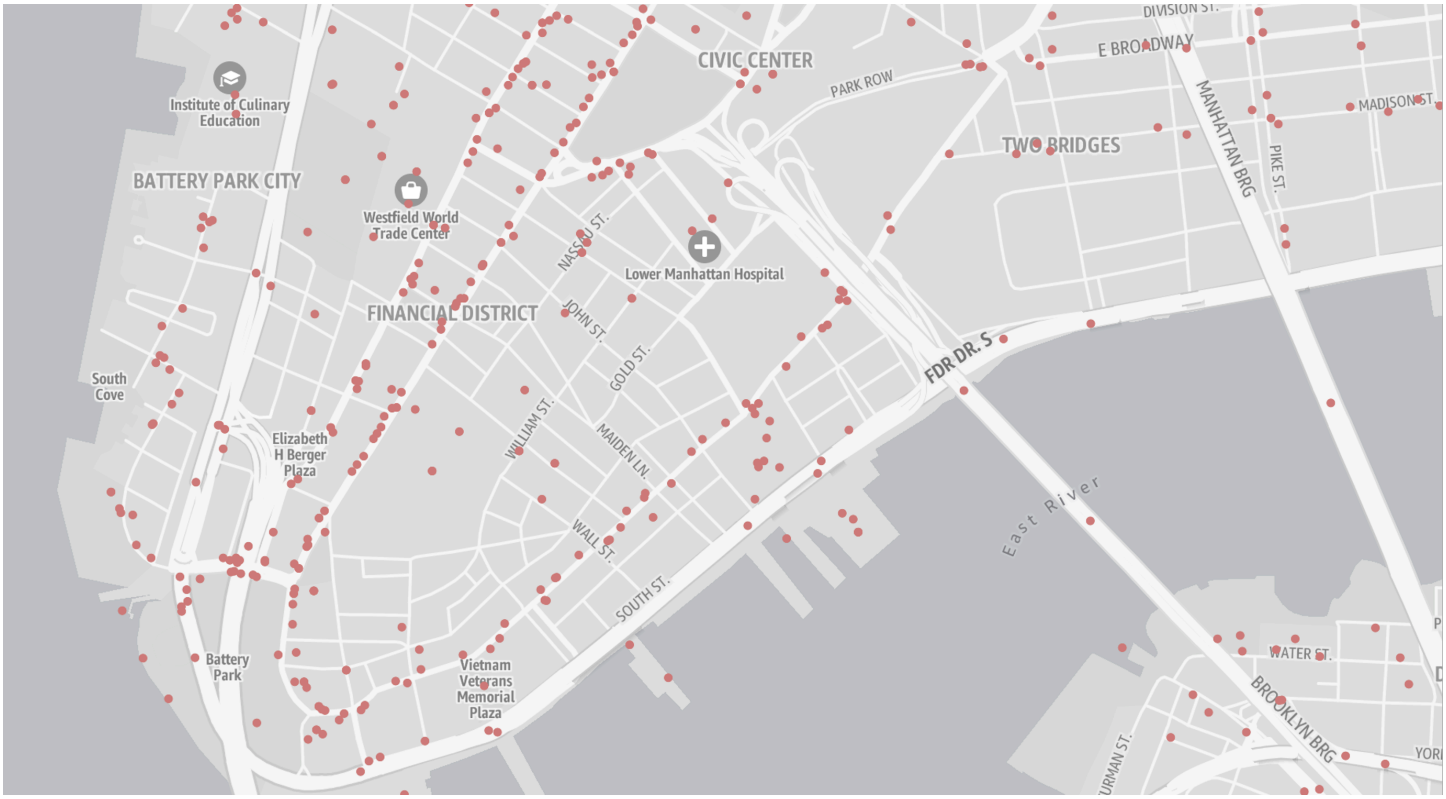


使用案例

Monochrome 风格非常适合数据可视化和极简主义的设计需求。

数据可视化

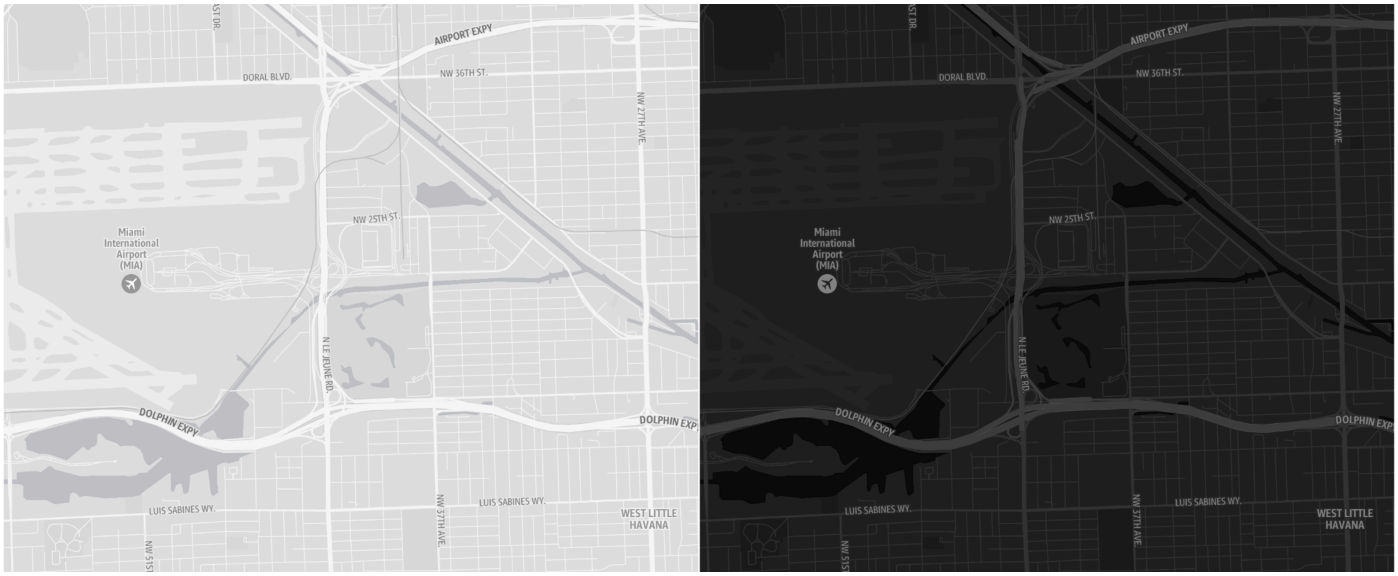
Monochrome 样式故意只使用灰色阴影，让您可以完全自由地选择数据叠加层（例如分区图、热图或点图）的颜色。



极简主义设计

为了保持地图干净不显眼，单色样式包括一组精简的兴趣点 (POIs)，用于基本要素，例如机场、公园、医院和大学。

Airport



Neighborhood



尽管单色样式包含一组精简版 POIs，但底层图块仍包含完整的 POI 数据集。这允许您显示 POIs 样式中没有视觉效果的内容。

专为世界而设计

单色样式支持不同的政治观点，可确保地图为用户显示正确的边框。该样式还允许在地图标签的语言之间轻松切换，支持数十种语言和书写系统。



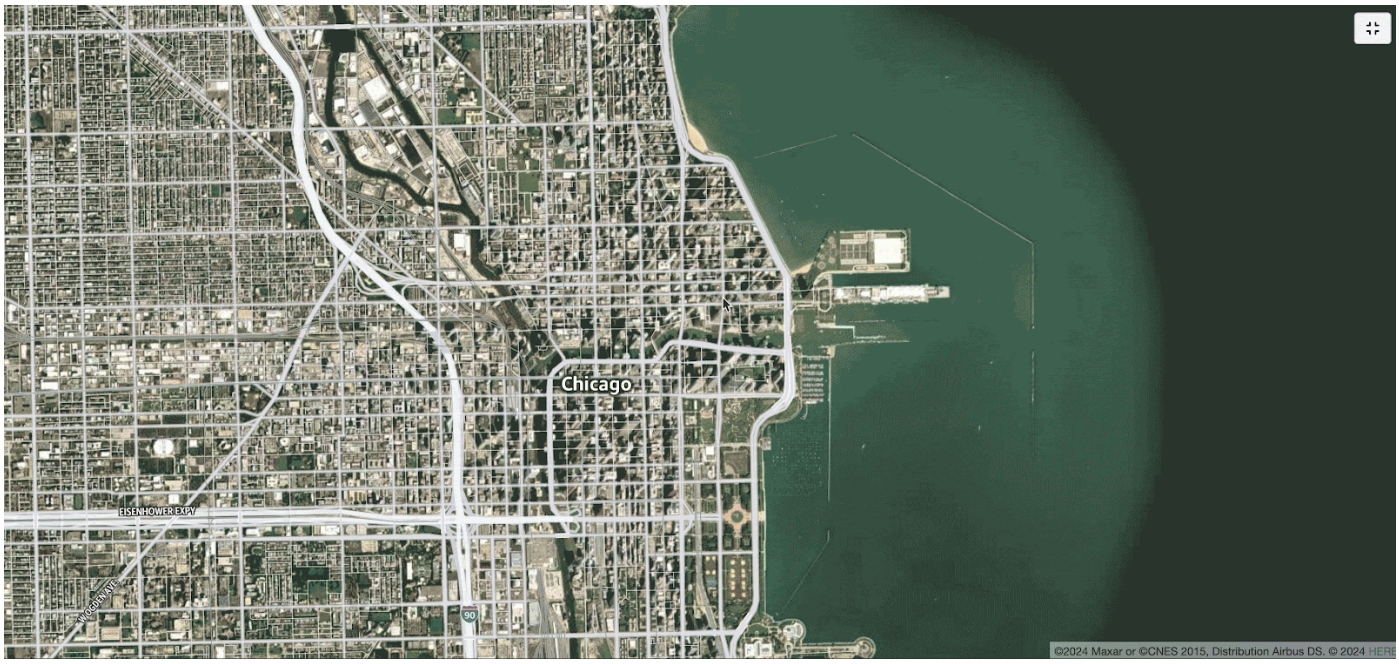
混合地图风格

混合地图样式将全球卫星影像与标准地图样式中相同的清晰标注和可配置的兴趣点 (POI) 类别相结合。这种组合提供了丰富的地理细节，同时确保了应用程序的可读性和可用性。

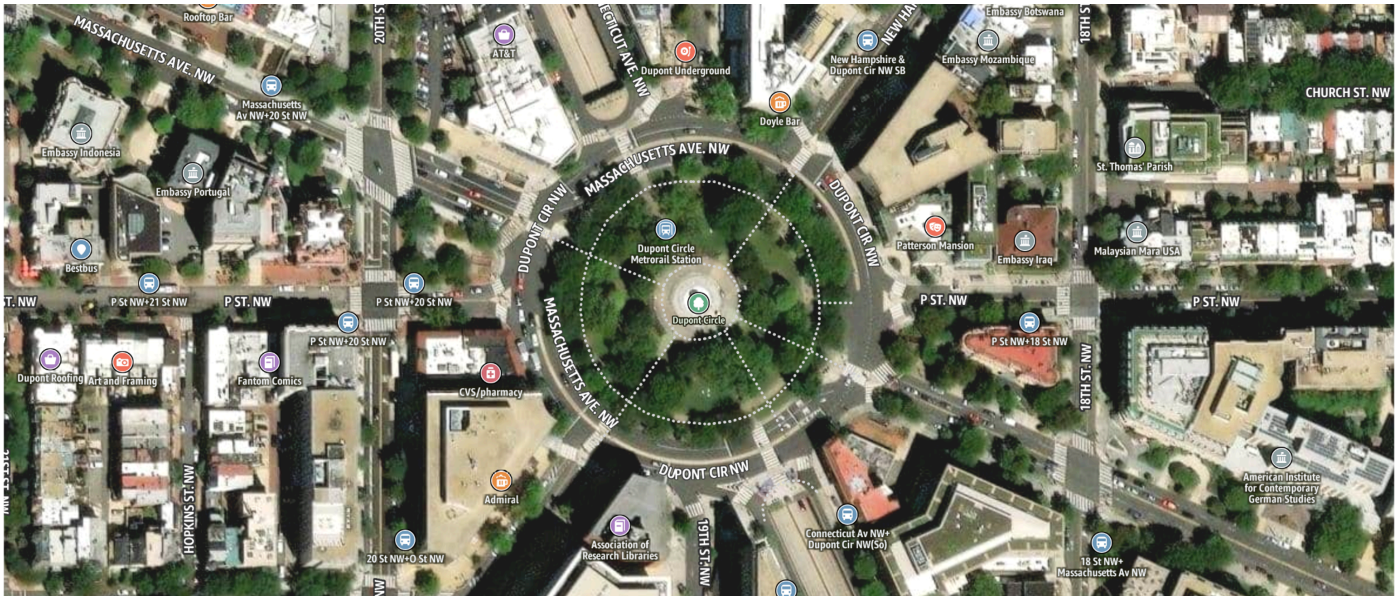
丰富的兴趣点 (POI)

POIs 这些标签是专门为对比度和可读性而设计的，为卫星层提供了必要的背景信息，而不会分散对详细图像的注意力。轻型道路线在缩小时会突出显示城市结构，放大时会逐渐消失，从而显示出更详细的街道信息。

Zoom



Neighborhood



Zoomed-in



专为世界而设计

混合风格支持不同的政治观点，可确保地图为用户显示正确的边界。这种风格还允许在地图标签的语言之间轻松切换，有数十种支持的语言和书写系统可供选择，以确保本地化体验。

地图 APIs

地图为各种应用程序提供了对动态和静态地图类型的访问。有关更多信息，请参阅映射。

- 动态地图：可以实时自定义的交互式地图，允许用户平移、缩放和叠加数据。有关更多信息，请参阅the section called “动态地图”。
- 静态地图：不带交互元素的显示特定位置或路线的地图的静态图像，适用于交互性有限的应用程序。有关更多信息，请参阅the section called “静态地图”。

下表列出了一些最适合使用地图解决的业务用例 APIs。

地图用例

以下部分介绍了许多最适合使用地图解决的业务用例 APIs。

业务需求	有用的 API	示例
显示交互式地图 支持地图手势，例如缩放、平移、缓动、飞行、俯仰、旋转和方位。	`GetTile`并`GetStyleDescriptor` 使用渲染引擎 (MapLibre)	the section called “显示地图”
向地图添加标记 例如标记、图标等。	`GetTile`并`GetStyleDescriptor` 使用渲染引擎 (MapLibre)	the section called “在地图上添加标记” the section called “在地图上添加图标”
向地图添加用户交互组件 例如，以首选语言或地缘政治视角显示地图。	`GetTile`并`GetStyleDescriptor` 使用渲染引擎 (MapLibre)	the section called “在地图上添加控件” the section called “向地图添加弹出窗口”
在地图上可视化实时数据或预先录制的数据 示例包括热图、KML、GeoJSON 要素、多边形、矩形、折线、圆形、标记等。	`GetTile`并`GetStyleDescriptor` 使用渲染引擎 (MapLibre)	the section called “在地图上添加一条线” the section called “在地图上添加多边形”
显示带有本地化功能的地图 例如，以首选语言或地缘政治视角显示地图。	`GetTile`并`GetStyleDescriptor` 使用渲染引擎 (MapLibre)	the section called “为地图设置首选语言” the section called “设置地图的政治视图”
显示静态地图图像 例如，在应用程序、电子邮件、报告或打印中使用地图图像。	`GetStaticMap`	the section called “获取特定位置的静态地图” the section called “获取特定维度的静态地图”

业务需求	有用的 API	示例
		the section called “在静态地图的半径和缩放之间做出选择” the section called “为静态地图添加比例”
向地图图像添加标记 例如标记、近距离圆圈、图标等。	`GetStaticMap`	the section called “向静态地图添加标记”
在地图图像上可视化数据 例如 GeoJSON 要素、多边形、矩形、折线、圆形等。	`GetStaticMap`	the section called “向静态地图添加一条线”
在地图上可视化现实世界的用例 示例包括路线、邻近圈等。	`GetStaticMap`	the section called “向静态地图添加路由”
在地图上可视化地点搜索和/或地理编码结果 所有 APIs 返回地理坐标，但自动完成除外。	`GetTile`并`GetStyleDescriptor` 使用带有 Places API 的渲染引擎 (MapLibre)	
在地图上绘制路线 支持航点标记。	`GetTile`并`GetStyleDescriptor` 使用带有计算路径的渲染引擎 (MapLibre)	
在地图上可视化匹配的 GPS 轨迹 支持卡车、行人、汽车和踏板车等出行模式。	`GetStyleDescriptor` 使用带有 Snap to road 的渲染引擎 (MapLibre)	

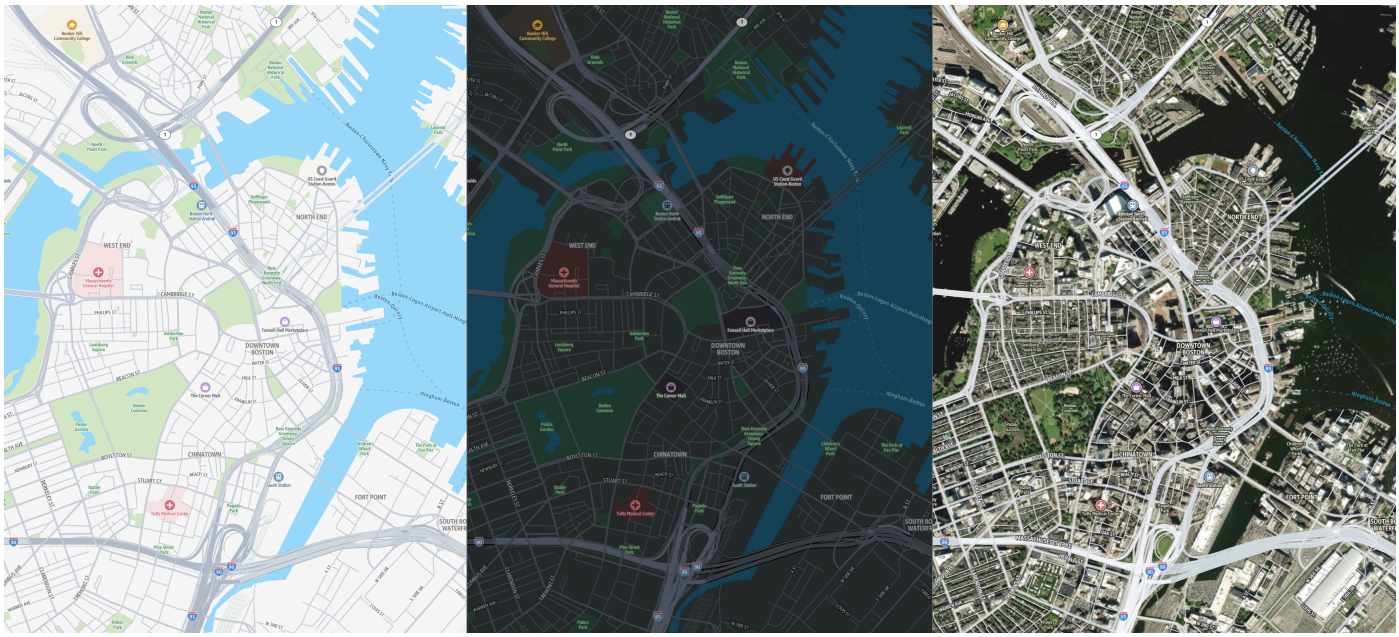
动态地图

> **Note**
>
> 您必须使用政治视图功能来遵守适用的法律，包括与绘制您通过 Amazon Location Service 访问的地图、图像和其他数据所在的国家或地区相关的法律。

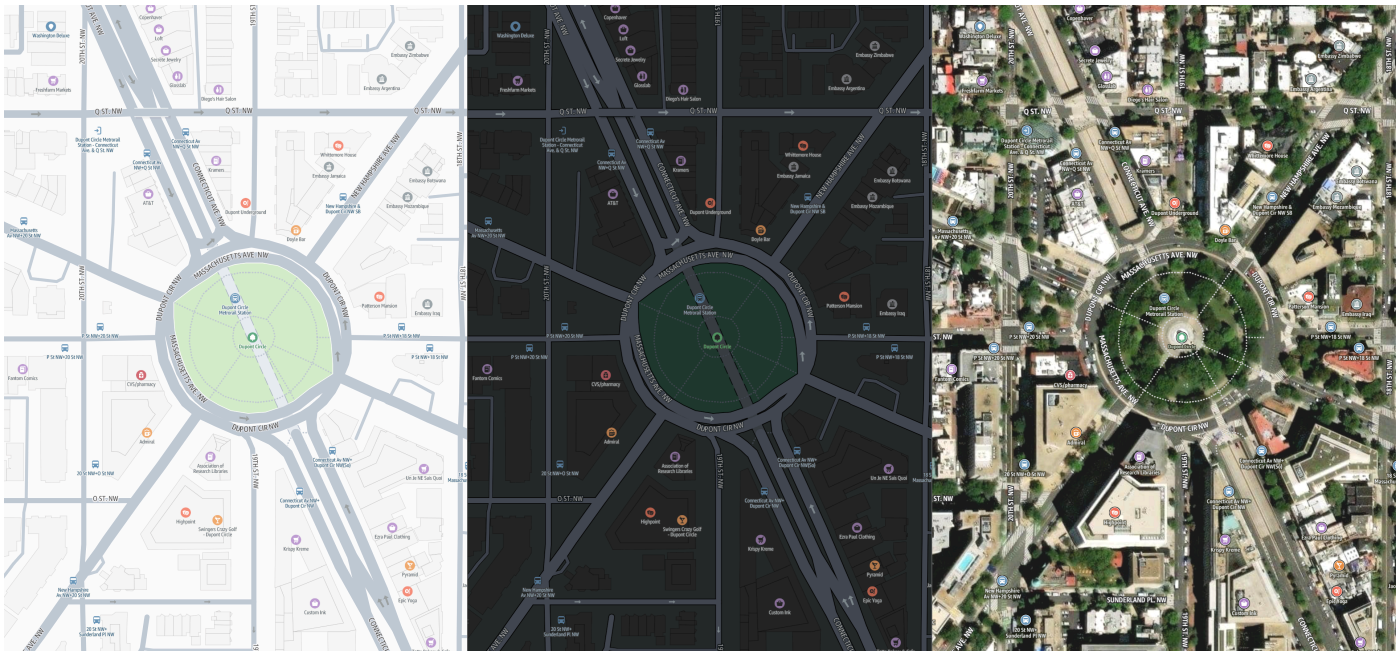
动态地图，也称为交互式地图，是支持缩放、平移、缓动、飞行、俯仰、旋转和方位等手势的数字地图。借助 Amazon Location Service，您可以构建地图应用程序，为用户提供响应式、交互式和身临其境的体验。这些地图可帮助用户根据用户输入可视化和分析实时和历史数据，从而允许他们实时平移、缩放和浏览地图。Amazon Location Service 提供的地图还支持多种语言和政治观点。

了解有关 the section called “本地化和国际化” 的更多信息。

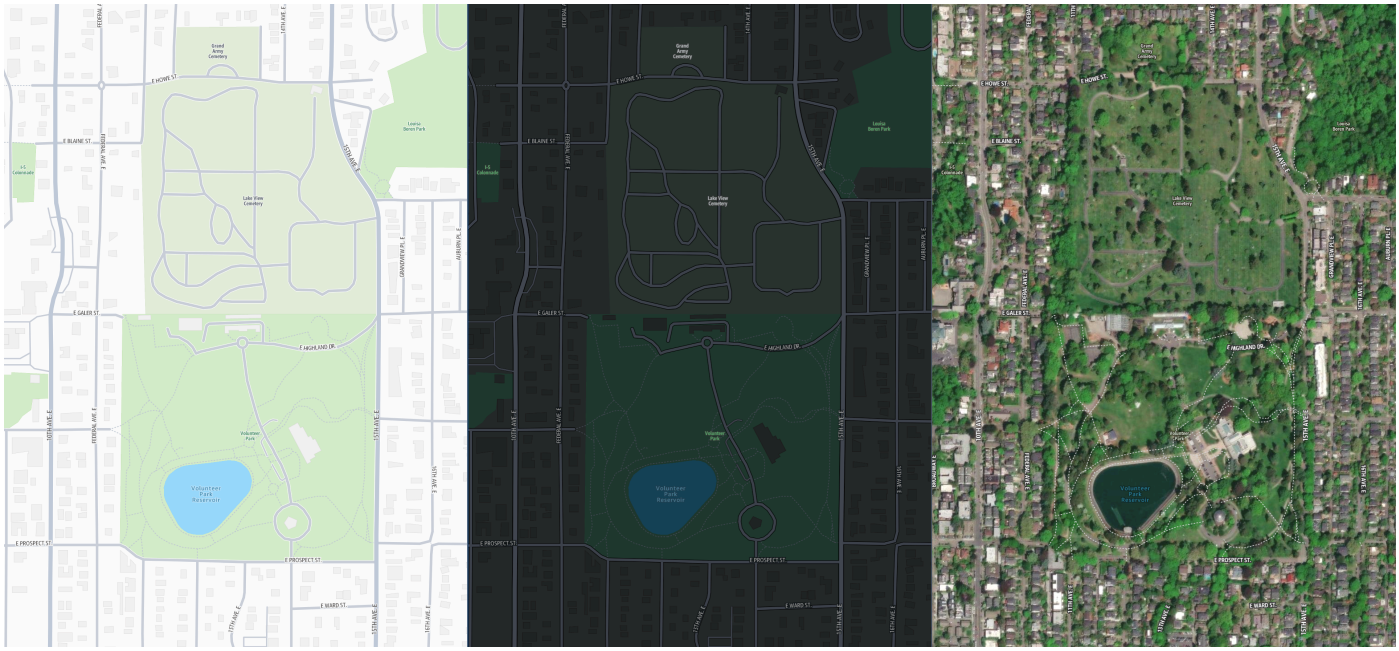
City



Roads



Park



有关 AWS 地图样式的更多信息，请参阅the section called “地图样式”。

常见使用案例

分析和可视化数据

将您的数据叠加到高质量的地图上，以发现变革性的空间模式和趋势。让您的团队能够使用您的地理数据创建可自定义的交互式地图可视化效果。使用地图和数据来优化选址、规划基础设施或分析市场机会。

增强房地产体验

为潜在买家提供房地产清单的全面位置背景。显示物业的确切位置以及周边社区的详细信息，例如管辖区边界、当地企业、公园和学校。帮助客户找到前往开放日的路线，打造内容丰富、以位置为中心的房地产体验。

打造引人入胜的旅行体验

显示显示目的地的动态地图，包括详细的街景和主要地理特征。突出游客和旅行者的景点，例如酒店、餐厅和景点。规划远足径等户外设施，以帮助用户规划理想的行程。

渲染动态地图

地图渲染引擎是一个库，负责在数字屏幕上对地图进行可视化渲染。渲染引擎将地图切片（矢量、混合、卫星）、地图数据（点、线、面）或栅格数据（影像）拼接在一起，以便在 Web 浏览器或本机应用程序中显示交互式地图。Amazon Location Service 建议使用MapLibre渲染引擎，该引擎同时支持网络和移动（iOS 和 Android）平台。 MapLibre 还提供了插件模型，并支持用户界面以各种语言进行搜索和路由。

有关更多信息，请参阅the section called “创建您的第一个 “地图和地点” 应用程序”和the section called “动态地图”。

请求地图资产

渲染引擎使用地图样式，其中包含对地图图块、精灵（图标）和字形（字体）的引用。当用户与地图交互（加载、平移或缩放）时，渲染引擎会使用所需的输入参数调用 APIs （适用于图块、精灵和字形）。您也可以 APIs 根据应用程序的需求直接调用它们。

地图图块

包含从服务器检索并由渲染引擎组装的数据的小方块，以创建交互式数字地图。

地图风格

定义地图视觉外观（例如颜色和样式）的一组规则。Amazon Location Service 遵循 Mapbox GL 风格规范。

字形文件

包含编码的 Unicode 字符的二进制文件，地图渲染器使用它来显示文本标签。

Sprite 文件

一种便携式网络图形 (PNG) 图像文件，其中包含小型光栅图像，在 JSON 文件中带有位置描述。地图渲染器用于在地图上渲染图标或纹理。

瓷砖

地图图块是预先渲染的，是较大地图的一小部分，通常显示为正方形图像。它们用于通过仅加载不同缩放级别下的可见部分来有效地显示地理数据。地图图块主要有三种类型：

图块类型

矢量地图图块

矢量地图切片将地图数据存储为几何形状（点、线、面），而不是图像。这使得可以创建在任何分辨率下都保持清晰的高质量、可缩放的地图。

栅格地图图块

栅格地图分块是表示特定地理区域的预渲染图像。与矢量切片不同，栅格切片更简单，但缺乏重新设计的灵活性。

混合地图图块

混合地图切片将矢量数据和栅格数据结合在一起。他们使用矢量数据作为核心地图元素（例如道路），而将栅格图像用于更复杂的元素，例如详细的卫星或航空摄影。

矢量切片图层

以下是 10 层矢量地图图块：

- 边界：定义行政和地理边界，包括国家、州和城市边界。
- 建筑物和地址：表示建筑物形状和详细地址点。
- 地球：显示沙漠、山脉和森林等自然特征的全球地形和地表覆盖范围。

- 土地利用：显示已分类的区域，例如公园、农田和城市区域。
- 地点：识别重要地点，例如城市、城镇和著名地标。
- 兴趣点 (POIs)：重点介绍景点、企业和其他关键地点。
- 道路：表示街道、高速公路和小径网络。
- 道路标签：为道路名称和路径编号提供文本标签。
- 交通：描绘公共交通线路，例如公共汽车、火车和地铁。
- 水：显示水体，包括湖泊、河流和海洋。

使用案例

- 获取地图图块，用于在不同的缩放级别下渲染地图的不同部分。
- 根据用户交互（例如平移和缩放）优化地图图块请求。
- 访问矢量或栅格切片以进行详细渲染。

理解请求

该请求需要以下参数：`Tileset`、`XY`、和`Z`来标识要获取的特定图块。可以选择包含该`Key`参数以进行授权。

- **`Tileset`**：指定用于获取图块的所需图块集。
- **`X`**: 地图图块的 X 轴值。
- **`Y`**: 地图图块的 Y 轴值。
- **`Z`**: 缩放值，定义图块的缩放级别。
- **`Key`**：出于授权目的，可选包含在内。

了解回应

响应包含、和`CacheControl`、等标头 `ContentType``ETag`，并包含以 MVT 格式的二进制 blob 形式的地图图块数据。这些标题用于管理缓存控制、提供内容格式详细信息以及切片的版本控制。

- **`CacheControl`**：控制地图切片的客户端缓存。
- **`ContentType`**：指定切片数据的格式。
- **`ETag`**：提供磁贴的版本标识符。
- **`Blob`**：包含 MVT 格式的矢量切片数据。

为动态地图设置样式

Amazon Location Service 提供了两种设置动态地图样式的选项：使用预先设计的地 AWS 图样式或使用样式描述符自定义地图样式。

使用预先设计 AWS 的地图样式

AWS 地图样式是符合行业标准的预定义样式，可提供精致、专业的美感。通过在 Amazon Location Service 中利用这些样式，您可以减少 time-to-market和消除专业制图师从头开始创建地图样式的需求。

有关更多信息，请参阅 the section called “地图样式”。

要了解有关预定义地图样式的更多信息，请参阅：

- the section called “标准地图风格”
- the section called “单色地图风格”
- the section called “混合地图风格”
- the section called “卫星地图风格”

使用 AWS 地图样式的好处

- 时间和资源效率： AWS 地图样式允许您绕过从头开始设计地图样式的耗时且资源密集型的过程。这使您可以专注于核心功能，同时提供具有视觉吸引力的地图。
- 专业且一致的美学：熟练的制图师遵循行业最佳实践，精心制作了 AWS 地图样式。从调色板到标签放置，每个细节都经过优化，以提高清晰度和易读性。
- 无缝集成： AWS Map Styles 与应用程序的设计语言无缝集成，为您的最终用户提供精致而一致的地图体验。

开始使用 AWS 地图样式

- 查看提供的地 AWS 图样式：在 Amazon Location Service 控制台中，导航至地图部分以浏览可用的样式。
- 选择符合您需求的样式：选择最符合应用程序设计和用户体验要求的样式。
- 整合样式：按照提供的文档，使用 Amazon Location Service 将所选样式集成到您的应用程序中，APIs 或者 SDKs。

了解有关 the section called “显示地图” 的更多信息。

使用案例

- 基于 “浅色” 或 “深色” 等配色方案自定义地图样式。
- 根据特定的政治观点或地理边界显示地图。
- 针对不同的用例优化地图样式，例如物流视图或默认视图。

理解请求

该请求支持诸如`ColorSchemeKey`、和之类的参数`PoliticalView`来定义地图的样式和显示方式。该`Style`参数是指定所需地图样式所必需的。

- **`ColorScheme`**：设置地图的色调，例如 “浅色” 或 “深色”。
- **`PoliticalView`**：指定地图可视化的政治视图。
- **`Style`**：定义地图的样式，例如 “标准” 或 “单色”。

了解回应

响应提供诸如`CacheControl`、和`ContentType`、之类的标头`ETag`，并包含作为 JSON blob 的样式描述符。标题提供了缓存信息、内容格式详细信息以及样式更改的版本控制。

- **`CacheControl`**：控制样式描述符的缓存配置。
- **`ContentType`**：表示响应格式为 JSON。
- **`ETag`**：为样式描述符提供版本标识符。
- **`Blob`**: 包含 JSON 格式的样式描述符的正文。

自定义样式描述符

要自定义地图样式，必须了解样式描述符的结构，该描述符通常是一个定义地图元素视觉表示形式的 JSON 对象。样式描述符由多个图层组成，每个图层控制特定类型的地图元素（例如道路、公园、建筑物或标注）的样式。

- 使用预定义的样式描述符作为基础：您可以从预定义的样式描述符开始，也可以使用 Maputnik 等地图样式编辑器从头开始创建样式描述符。
- 了解结构：样式描述符是一个分层 JSON 对象，其中包含图层，每个图层代表一个不同的地图元素。每个图层都有控制该元素视觉外观的属性，例如颜色、不透明度和线宽。

- 修改图层样式：根据您使用的地图样式编辑器，您可以更改现有图层或添加新图层以自定义样式。例如，您可以调整道路的颜色、修改标签的字体大小或为特定位置添加自定义图标。
- 为不同的缩放级别定义样式：地图样式编辑器允许您为不同的缩放级别定义不同的样式，这对于根据用户缩放交互控制细节级别和可见性非常有用。
- 测试和迭代：修改或创建样式描述符后，在地图上测试自定义样式以确保其按预期显示。迭代和调整，直到获得所需的视觉风格。

带有精灵的风格肖像

精灵是一种便携式网络图形 (PNG) 图像文件，其中包含小型光栅图像，例如图标、标记和其他在地图上渲染的元素。可以根据样式、配色方案和变体等参数自定义 Sprite。Amazon Location Service 通过 `GetSprites` API 提供精灵表。您也可以通过加载自己的图标集（参见the section called “在地图上添加图标”）或自定义样式描述符来加载自定义 sprite 来使用自定义图标。

使用案例

- 使用精灵表为特定样式和配色方案渲染自定义地图元素。
- 获取各种地图风格（例如标准、单色或混合）的精灵。
- 通过修改精灵自定义地图上的图标。

理解请求

该请求需要 URI 参数`ColorScheme`，例如`FileName`、和`Style`。这些参数允许根据地图的配色方案、样式和所需的特定精灵文件对精灵表进行自定义。

- **`ColorScheme`**：定义精灵的配色方案，例如 “浅色” 或 “深色”。
- **`FileName`**: 要检索的精灵文件的名称，可以是 PNG 或 JSON 文件。
- **`Style`**：指定地图样式，例如 “标准” 或 “单色”。

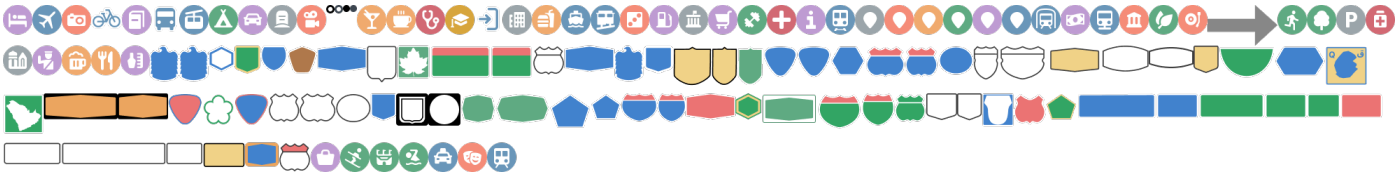
了解回应

响应包含诸如`CacheControl`、和`ContentType`、之类的标头`ETag`，并以二进制 blob 或 JSON 文件的形式返回精灵数据。这些标头提供缓存信息、响应的内容类型以及精灵数据的版本控制。

- **`CacheControl`**：精灵文件的缓存配置。
- **`ContentType`**：响应的格式，表示它包含 PNG 还是 JSON 数据。

- **`ETag`**: 精灵版本的标识符，用于缓存验证。
- **`Blob`**: 包含精灵表的主体或 JSON 偏移文件。

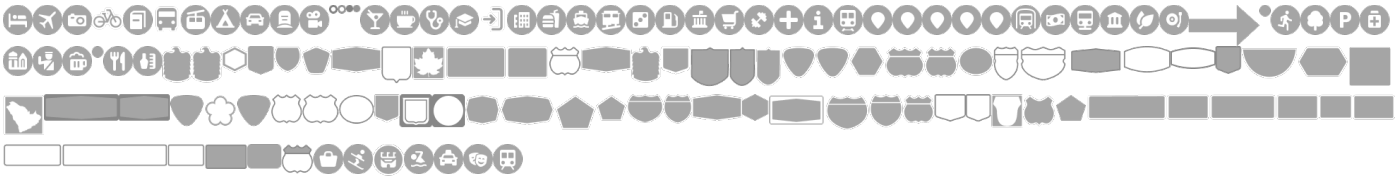
Standard Light



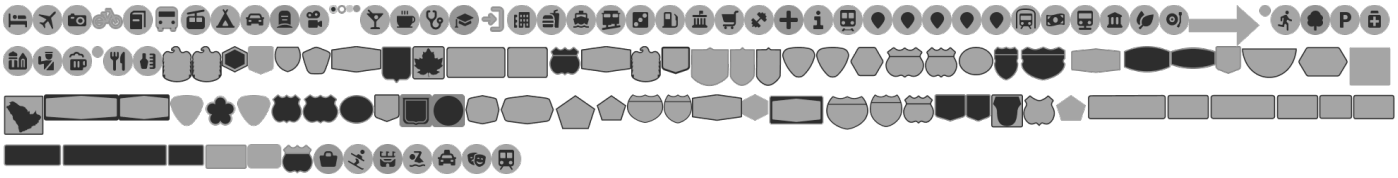
Standard Dark



Monochrome Light



Monochrome Dark



Hybrid



使用字形为标签添加样式

Glyphs 是包含编码的 Unicode 字符的二进制文件，地图渲染器使用这些字符来显示标签。Amazon Location Service 允许从字体堆栈中检索特定的字形，以便通过 `GetGlyphs` API 进行地图渲染。

使用案例

- 使用特定字体和样式在地图上渲染自定义文本。
- 获取用于本地化地图文本渲染的字形。
- 使用 Unicode 字符范围来显示地图标签和符号。
- 根据字体堆栈和字形范围优化地图字体渲染。

API 中支持的字体

API 支持以下字体：

- Amazon Ember Bold
- Amazon Ember 粗体斜体
- Amazon Ember 粗体、Noto Sans 粗体
- Amazon Ember 粗体、Noto Sans 粗体、Noto Sans Arabic 粗体
- 亚马逊 Ember 精简版 RC BdItalic
- Amazon Ember 精简版 RC 粗体
- Amazon Ember 精简版 RC 粗体斜体
- Amazon Ember Condensed RC 粗体、Noto Sans 粗体
- Amazon Ember Consending RC 粗体、Noto Sans 粗体、Noto Sans
- Amazon Ember 浓缩遥控灯
- Amazon Ember 精简版 RC Light 斜体
- 亚马逊 Ember 精简版 RC LtItalic
- Amazon Ember 精简版 RC 常规版
- Amazon Ember 精简版 RC 常规斜体
- Amazon Ember Condensed RC 常规、Noto Sans 常规
- Amazon Ember Condensed RC 常规、Noto Sans 常规、Noto Sans Arabic Condensed 常规
- 亚马逊 Ember 精简版 RC RgItalic
- 亚马逊 Ember 精简版 RC ThItalic
- 亚马逊 Ember 精简版 RC Thin
- Amazon Ember 精简版 RC 薄斜体

- 亚马逊 Ember Heavy
- Amazon Ember Heavy 斜体
- 亚马逊 Ember Light
- Amazon Ember Light 斜体
- 亚马逊 Ember Medium
- Amazon Ember 中号斜体
- Amazon Ember 中等、Noto Sans 中等
- Amazon Ember 中等、Noto Sans 中等、Noto Sans Arabic 中等
- 亚马逊 Ember 普通版
- Amazon Ember 普通斜体
- Amazon Ember 常规斜体、Noto Sans 斜体
- Amazon Ember 常规斜体、Noto Sans 斜体、Noto Sans Arabic 常规
- Amazon Ember 常规、Noto Sans 常规
- Amazon Ember 常规、Noto Sans 常规、Noto Sans Arabic 常规
- 亚马逊 Ember Thin
- Amazon Ember Thin 斜体
- AmazonEmberCdrc_bd
- AmazonEmberCdRC_ BdIt
- AmazonEmberCdrc_lt
- AmazonEmberCdRC_ LtIt
- AmazonEmberCdrc_rg
- AmazonEmberCdRC_ RgIt
- AmazonEmberCdrc_th
- AmazonEmberCdRC_ ThIt
- AmazonEmber_Bd
- AmazonEmber_BdIt
- AmazonEmber_他
- AmazonEmber_HeIt

- AmazonEmber_Lt
- AmazonEmber_LtIt
- AmazonEmber_Md
- AmazonEmber_MdIt
- AmazonEmber_Rg
- AmazonEmber_RgIt
- AmazonEmber_Th
- AmazonEmber_ThIt
- Noto Sans Blac
- Noto Sans 黑色斜体
- Noto Sans 粗体
- Noto Sans 粗斜体
- Noto Sans 超粗体
- Noto Sans 超粗斜体
- Noto Sans 超轻
- Noto Sans 超轻斜体
- Noto Sans 斜体
- Noto Sans Ligh
- Noto Sans 浅斜体
- Noto Sans 中等
- Noto Sans 中等斜体
- Noto Sans 常规
- Noto Sans 半粗体
- Noto Sans 半粗斜体
- Noto Sans Thin
- Noto Sans Thin 斜体
- NotoSans-粗体
- NotoSans-斜体
- NotoSans-中号

- NotoSans-常规
- 无常规打开，Arial Unicode MS 常规版

理解请求

该请求接受两个必需的 URI 参数，即`FontStack`和`FontUnicodeRange`，它们确定字形的字体和 Unicode 范围。`FontStack`参数指定要使用哪种字体，而`FontUnicodeRange`定义要获取的字符范围。该请求不包含正文，仅关注其功能的 URI 参数。

- **`FontStack`**：指定要检索的字体堆栈的名称。示例：“Amazon Ember Bold，Noto Sans Bold”。
- **`FontUnicodeRange`**: 要下载字形的 Unicode 字符范围。范围必须是 256 的倍数。示例：“0-255”。

了解回应

该响应以二进制 blob 的形式返回字形数据，以及用于缓存、内容类型和定价信息的 HTTP 标头。`ETag`字形数据以二进制 blob 的形式返回，以便在地图上呈现，标头提供了用于有效处理响应的额外元数据。

- **`CacheControl`**：指示客户端进行响应的缓存配置。
- **`ContentType`**：指定响应正文的格式，指明返回的字形数据的类型。
- **`ETag`**：字形版本的标识符，用于缓存验证。
- **`PricingBucket`**：表示与请求相关的定价等级。
- **`Blob`**: 以二进制 blob 形式返回的字形数据，用于呈现地图文本。

静态地图

> **Note**
>
> 静态地图仅支持卫星风格。有关更多信息，请参阅 the section called “地图样式”。

静态地图提供预渲染的地理数据表示形式，可以根据应用程序的需要选择叠加标记（或图钉）、路线和多边形区域。静态地图允许您根据可自定义的参数和数据输入生成静态（非交互式）地图图像。通过自定义叠加层、形状或应用自定义样式，Static Map 使您能够创建满足特定需求的地图可视化效果，从而增强最终用户体验并有效地传达地理信息。服务器对请求的地图图像进行自定义，并将其作为 JPEG 文件提供给客户端。您可以通过编程方式请求和生成根据您的特定要求量身定制的地图图像。

GetStaticMap API 根据指定参数（例如中心坐标、边界框或叠加层）生成地图的静态图像。API 允许自定义地图功能和样式，允许在没有交互式地图功能的 Web 或移动应用程序中使用。

常见使用案例

- Web 或移动应用程序中的嵌入式地图：静态地图图像可以高效地嵌入到网站或移动应用程序中，通过非交互式地图提供位置、路线或兴趣点的可视化效果，从而减少加载时间和数据使用量。例如，搜索引擎（例如雅虎）显示地图图像以及搜索结果 POIs。
- 电子邮件中的位置详情：静态地图图像可用于通过电子邮件共享位置信息，以帮助您的最终用户了解电子邮件的上下文。例如，送餐或拼车应用程序使用静态地图图像在包含账单和摘要的行后或送货电子邮件中显示上车/下车地点、路线或周边区域。
- 营销材料和印刷文档：可以将定制的静态地图图像整合到小册子、传单或其他印刷材料中，从而提供与内容相关的地理信息的视觉吸引力。

理解请求

该请求包括可选的 URI 参数，例如`BoundedPositions`、`BoundingBox`和`Center`，用于定义地图的可见区域和叠加层。参数`Height`和`Width`是定义图像尺寸所必需的。要了解更多信息，请参阅 the section called “自定义静态地图” 和 the section called “叠加在静态地图上”。

- `BoundedPositions`：要包含在图像中的坐标。
- `BoundingBox`：定义地图西南边缘和东北边缘的坐标。
- `Height`：指定图像的高度。
- `Width`：指定图像的宽度。
- `GeoJsonOverlay`: 用于添加叠加层的有效 GeoJSON 对象。

了解回应

响应包含、和`CacheControl`、`ContentType`之类的标头`ETag`，并以 JPEG 或 PNG 格式的二进制 blob 形式返回静态地图。标头提供静态图像的缓存控制、内容类型和版本等元数据。

- `CacheControl`：指定地图图像的缓存配置。
- `ContentType`：表示地图图像的格式（JPEG 或 PNG）。
- `ETag`：静态地图图像版本的标识符。
- `Blob`：以 JPEG 或 PNG 格式表示地图图像。

自定义静态地图

> **Note**
>
> 静态地图仅支持卫星风格。有关更多信息，请参阅 the section called “地图样式”。

本节概述了如何自定义使用 Amazon Location Service 生成的静态地图。它涵盖了各种功能，例如调整地图的位置、大小、语言、比例、叠加层和属性，使您可以根据自己的特定要求定制地图。

位置

该位置允许您定义地图的中心和边界。您可以通过设置中心坐标、边界框或使用缩放级别来确定要显示的区域来控制地图的焦点。要了解其工作原理，请参阅the section called “获取特定位置的静态地图”。

- `Center`：使用经度和纬度坐标定义地图的中心点。
- `Radius`：指定将在静态地图上显示的半径（与中心的距离）。
- `Bounding Box`：定义地图的矩形区域，通过提供左上角和右下角的坐标进行设置。
- `Zoom`：控制地图的缩放级别。缩放级别越高，在较小的区域中显示的细节越多，而缩放级别越低，在较大的区域上显示的细节越少。

尺寸和质量

您可以通过定义静态地图的尺寸（高度和宽度）和添加填充来更好地呈现标记和其他元素，从而自定义静态地图的大小和视觉质量。要了解其工作原理，请参阅the section called “获取特定维度的静态地图”。

- `Height and Width`：通过定义静态地图图像的高度和宽度（以像素为单位）来指定静态地图图像的大小。
- `Padding`：在地图边缘周围添加额外空间，以便在放置标记、线条或形状时实现更好的可视化。

扩展

比例尺提供了对地图比例的控制并定义了测量距离的单位（千米、英里）。这对于准确表示地图的大小和距离关系非常有用。要了解其工作原理，请参阅the section called “为静态地图添加比例”。

- `Scale Unit`：定义地图比例尺的单位（例如，千米或英里），允许用户在地图上准确测量距离。

叠加

您可以添加标记、用于显示路径的线、用于显示区域的多边形等。要了解其工作原理，请参阅the section called “向静态地图添加标记”the section called “向静态地图添加一条线”、或the section called “向静态地图添加路由”。

叠加在静态地图上

本节介绍如何使用 Amazon Location Service 将其他信息叠加到静态地图上。您可以通过添加各种地理要素（例如点、线和面）来自定义静态地图，以增强地图的视觉表现形式。Amazon Location Service 支持多种格式，包括 GeoJSON 和紧凑的叠加层格式，以提供灵活高效的添加叠加层的方法。

使用 geoJSON

GeoJSON 是一种多功能格式，允许您在静态地图上叠加自定义数据。通过定义点、线和面等地理要素，您可以增强地图的可视化表示，为用户提供宝贵的背景信息。GeoJSON 得到了广泛支持，在样式化和自定义地图叠加层方面提供了灵活性，使其成为显示区域、绘制路线或显示空间关系的理想格式。

借助 Amazon Location Service，您可以利用 GeoJSON 将基于位置的动态功能直接添加到静态地图上。这使您能够创建高度可定制的叠加层，这些叠加层可以根据您的特定业务需求进行定制。GeoJSON 支持多种几何类型，包括`Point`、`LineString`、`Polygon`、`MultiPolygon`、和，允许您显示各种要素，从标记和路径到复杂的区域表示。

颜色

在对 GeoJSON 功能进行样式设置时，您可以灵活地定义颜色。您可以使用不同的格式指定颜色，例如十六进制值（如 #ff0000 代表红色）或使用 alpha 透明度（如 #ff000080 表示半透明的红色）。这样可以确保您的叠加层在视觉上与地图样式保持一致。如果未指定颜色，则将应用所选地图样式的默认颜色。

绘图顺序

自定义叠加层按特定顺序绘制，以保持清晰度并避免视觉混乱。在 Amazon Location Service 中，多边形、线和点等叠加要素将显示在底图上方，但在地图标签下方。绘制顺序首先优先考虑多边形，然后是线，然后是点或标记。

测量单位

对于`width`和之类的属性`outline-width`，您可以使用不同的测量单位来指定大小，包括像素 (px)、米 (m)、千米 (km)、英里 (mi) 和百分比 (%)。百分比单位相对于默认值调整属性，从而为叠加层的样式提供更大的灵活性。

几何类型

Amazon Location Service 支持多种 GeoJSON 几何类型`Point`，例如、`LineString`、`Polygon`和。`MultiPolygon`每种几何类型都可以使用 GeoJSON 中的属性对象进行样式设置和调整，从而可以对地图上的标记、路线和区域进行广泛自定义。

带有紧凑的覆盖层

> **Note**
>
> 紧凑叠加层支持以下几何类型：点、线和面。它不支持`multiPoint`、`multiLine`或`multiPolgyon`。

紧凑叠加选项允许您使用单个查询参数在静态地图上高效地显示多个几何图形。这种简化的方法简化了请求格式并缩小了请求的大小，从而更容易传输叠加数据。客户可以在一个查询参数中输入各种几何类型及其相应的样式属性，Amazon Location Service 将通过按指定方式解析和渲染叠加层来处理繁重的工作。

使用紧凑的叠加格式时，请记住请求网址的大小有限制。尽管 Amazon Location Service 优化了查询，但请确保您的请求保持在合理范围内，尤其是在处理多个几何图形及其关联属性时。

格式

紧凑的叠加层格式结构如下：`geometry_type:geometry;property_1=value_1;property_2=value_2|geometry_type:geometry;property_1=value_1...`

每种几何类型及其样式属性均已定义。多个几何由管道运算符 (|) 分隔，每个几何的属性使用分号分隔。

支持的几何类型

Amazon Location Service 支持多种几何类型，包括`Point`、`MultiPoint`、`LineString`、`Polygon`、、和`MultiPolygon`。这些几何类型可以在同一个查询参数中使用紧凑叠加格式进行组合和设置样式。

样式属性

每个几何体都可以使用各种样式属性进行自定义，例如颜色、轮廓颜色、大小等。这些属性允许您控制地图上每个几何图形的外观，从而确保叠加层符合您的业务需求。

操作方法

本节包含各种使用方法指南和如何使用地图的示例 APIs。

示例

- 如何使用动态地图
- 如何使用静态地图

如何使用动态地图

这些操作主题提供了全面的演练，教您如何使用 Amazon Location Service 增强地图应用程序。从显示交互式地图到添加标记、线条和多边形，这些教程演示了如何利用基本功能，例如设置地图控件、添加自定义图标和处理实时数据。此外，这些主题还涵盖本地化和国际化方面，例如设置首选语言、调整政治观点以及通过根据用户的位置或视角自定义地图内容来确保遵守地区法律。

每个操作方法都旨在便于访问，并附有使用 MapLibre GL JS 在 Web 应用程序中实现这些功能的 step-by-step说明。无论您的目标是使用自定义图标和弹出窗口构建动态地图体验，还是针对特定的政治视角和语言定制地图视图，这些示例都将帮助您实现目标，同时确保为不同地区的用户提供丰富的本地化地图体验。这些教程可确保您可以充分利用 Amazon Location Service 的功能，从基本地图功能到复杂的地缘政治和本地化需求。

主题

- 如何显示地图
- 如何在地图上添加控件
- 如何在地图上添加标记
- 如何在地图上添加图标
- 如何在地图上添加一条线
- 如何在地图上添加多边形
- 如何向地图添加弹出窗口
- 如何为地图设置首选语言
- 如何设置地图的政治视角
- 如何在地图上过滤 POI

如何显示地图

Amazon Location Service 允许您使用我们的地图样式显示交互式和非交互式地图。请参阅the section called “地图样式”，了解更多信息。

交互式地图

在本示例中，您将显示一个允许用户缩放、平移、捏合和俯仰的交互式地图。

交互式地图代码示例

index.html

```
                                    <!DOCTYPE html>
                                    <html lang="en">
                                        <head>
                                            <title>Display a map</title>
                                            <meta property="og:description"
 content="Initialize a map in an HTML element with MapLibre GL JS." />
                                            <meta charset='utf-8'>
                                            <meta name="viewport"
 content="width=device-width, initial-scale=1">
                                            <link rel='stylesheet' href='https://
unpkg.com/maplibre-gl@4.x/dist/maplibre-gl.css' />
                                            <link rel='stylesheet'
 href='style.css' />
                                            <script src='https://unpkg.com/maplibre-
gl@4.x/dist/maplibre-gl.js'></script>
                                        </head>
                                        <body>
                                            <!-- Map container -->
                                            <div id="map"></div>
                                            <script>
                                                const apiKey = "<API_KEY>";
                                                const mapStyle = "Standard"; //
 e.g., Standard, Monochrome, Hybrid, Satellite
                                                const awsRegion = "eu-central-1"; //
 e.g., us-east-2, us-east-1, us-west-2, etc.
                                                const styleUrl = `https://maps.geo.
${awsRegion}.amazonaws.com/v2/styles/${mapStyle}/descriptor?key=${apiKey}`;

                                                const map = new maplibregl.Map({
```

```
                                                container: 'map', // container
 id
                                                style: styleUrl, // style URL
                                                center: [25.24, 36.31], //
 starting position [lng, lat]
                                                zoom: 2, // starting zoom
                                            });
                                        </script>
                                    </body>
                                </html>
```

style.css

```
               body { margin: 0; padding: 0; }
            html, body, #map { height: 100%; }
```

将地图平移限制在某个区域之外

在本示例中，您将限制将地图平移到定义的边界之外。

限制地图平移代码示例

index.html

```
                        <!DOCTYPE html>
                        <html lang="en">
                            <head>
                                <title>Display a map</title>
                                <meta property="og:description" content="Initialize
 a map in an HTML element with MapLibre GL JS." />
                                <meta charset='utf-8'>
                                <meta name="viewport" content="width=device-width,
 initial-scale=1">
                                <link rel='stylesheet' href='https://unpkg.com/
maplibre-gl@4.x/dist/maplibre-gl.css' />
                                <link rel='stylesheet' href='style.css' />
                                <script src='https://unpkg.com/maplibre-gl@4.x/dist/
maplibre-gl.js'></script>
```

```
                            </head>
                            <body>
                                <!-- Map container -->
                                <div id="map"></div>
                                <script>
                                    const apiKey = "<API_KEY>";
                                    const mapStyle = "Standard"; // e.g., Standard,
 Monochrome, Hybrid, Satellite
                                    const awsRegion = "eu-central-1"; // e.g., us-
east-2, us-east-1, us-west-2, etc.
                                    const styleUrl = `https://maps.geo.
${awsRegion}.amazonaws.com/v2/styles/${mapStyle}/descriptor?key=${apiKey}`;

                                    var bounds = [
                                        [90.0, -21.943045533438166], // Southwest
 coordinates
                                        [146.25, 31.952162238024968] // Northeast
 coordinates
                                    ];

                                    const map = new maplibregl.Map({
                                        container: 'map', // container id
                                        style: styleUrl, // style URL
                                        maxBounds: bounds, // Sets bounds of SE Asia
                                    });
                                </script>
                            </body>
                        </html>
```

style.css

```
    body { margin: 0; padding: 0; }
    html, body, #map { height: 100%; }
```

非交互式地图

在本示例中，您将通过禁用用户交互来显示非交互式地图。

非交互式地图代码示例

index.html

```
    <!DOCTYPE html>
    <html lang="en">
        <head>
            <title>Display a map</title>
            <meta property="og:description" content="Initialize a map in an HTML
 element with MapLibre GL JS." />
            <meta charset='utf-8'>
            <meta name="viewport" content="width=device-width, initial-scale=1">
            <link rel='stylesheet' href='https://unpkg.com/maplibre-gl@4.x/dist/
maplibre-gl.css' />
            <link rel='stylesheet' href='style.css' />
            <script src='https://unpkg.com/maplibre-gl@4.x/dist/maplibre-gl.js'></
script>
        </head>
        <body>
            <!-- Map container -->
            <div id="map"></div>
            <script>
                const apiKey = "<API_KEY>";
                const mapStyle = "Standard"; // e.g., Standard, Monochrome, Hybrid,
 Satellite
                const awsRegion = "eu-central-1"; // e.g., us-east-2, us-east-1, us-
west-2, etc.
                const styleUrl = `https://maps.geo.${awsRegion}.amazonaws.com/v2/
styles/${mapStyle}/descriptor?key=${apiKey}`;

                const map = new maplibregl.Map({
                    container: 'map', // container id
                    style: styleUrl, // style URL
                    center: [25.24, 36.31], // starting position [lng, lat]
                    zoom: 2, // starting zoom
                    interactive: false, // Disable pan & zoom handlers
                });
            </script>
        </body>
    </html>
```

style.css

```
body { margin: 0; padding: 0; }
html, body, #map { height: 100%; }
```

如何在地图上添加控件

Amazon Location Service 允许您向地图添加多个控件，包括导航、地理位置、全屏、比例和归因控制。

- 导航控件：包含缩放按钮和指南针。
- 地理定位控件：提供一个按钮，该按钮使用浏览器的地理定位 API 在地图上定位用户。
- 全屏控制：包含一个用于切换地图进入和退出全屏模式的按钮。
- 比例控制：显示地图上的距离与地面上相应距离的比率。
- 归因控制：显示地图的归因信息。默认情况下，归因控制会扩展（无论地图宽度如何）。

您可以将控件添加到地图的任何角落：左上角、左下角、右下角或右上角。

添加地图控件

在以下示例中，您将添加上面列出的地图控件。

地图控制代码示例

index.html

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Map Controls</title>
        <meta property="og:description" content="Initialize a map in an HTML
 element with MapLibre GL JS." />
        <meta charset='utf-8'>
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <link rel='stylesheet' href='https://unpkg.com/maplibre-gl@4.x/dist/
maplibre-gl.css' />
        <link rel='stylesheet' href='style.css' />
```

```
            <script src='https://unpkg.com/maplibre-gl@4.x/dist/maplibre-gl.js'></
script>
        </head>
        <body>
            <!-- Map container -->
            <div id="map"></div>
            <script>
                const apiKey = "<API_KEY>";
                const mapStyle = "Standard"; // e.g., Standard, Monochrome, Hybrid,
 Satellite
                const awsRegion = "eu-central-1"; // e.g., us-east-2, us-east-1, us-
west-2, etc.
                const styleUrl = `https://maps.geo.${awsRegion}.amazonaws.com/v2/
styles/${mapStyle}/descriptor?key=${apiKey}`;

                const map = new maplibregl.Map({
                    container: 'map', // container id
                    style: styleUrl, // style URL
                    center: [-123.13203602550998, 49.28726257639254], // starting
 position [lng, lat]
                    zoom: 10, // starting zoom
                    attributionControl: false, // hide default attributionControl in
 bottom left
                });

                // Adding controls to the map
                map.addControl(new maplibregl.NavigationControl()); // Zoom and
 rotation controls
                map.addControl(new maplibregl.FullscreenControl()); // Fullscreen
 control
                map.addControl(new maplibregl.GeolocateControl()); // Geolocation
 control
                map.addControl(new maplibregl.AttributionControl(), 'bottom-
left'); // Attribution in bottom-left
                map.addControl(new maplibregl.ScaleControl(), 'bottom-right'); //
 Scale control in bottom-right
            </script>
        </body>
    </html>
```

style.css

```
body { margin: 0; padding: 0; }
html, body, #map { height: 100%; }
```

开发者小贴士

导航控制选项

```
new maplibregl.NavigationControl({
    showCompass: true, // show or hide compass (default: true)
    showZoom: true // show or hide zoom controls (default: true)
});
```

地理定位控制选项

```
new maplibregl.GeolocateControl({
    positionOptions: { enableHighAccuracy: true }, // default: false
    trackUserLocation: true // default: false
});
```

归因控制选项

```
new maplibregl.AttributionControl({
    compact: true, // compact (collapsed) mode (default: false)
});
```

比例控制选项

```
new maplibregl.ScaleControl({
    maxWidth: 100, // width of the scale (default: 50)
```

```
        unit: 'imperial' // imperial or metric (default: metric)
    });
```

全屏控制选项

```
    map.addControl(new maplibregl.FullscreenControl({
        container: document.querySelector('body') // container for fullscreen mode
    }));
```

如何在地图上添加标记

使用 Amazon Location，您可以添加固定标记和可拖动标记，还可以根据自己的数据和偏好自定义标记的颜色。

添加固定标记

固定标记代码示例

index.html

```
    <!DOCTYPE html>
    <html lang="en">
        <head>
            <title>Add marker on a map</title>
            <meta property="og:description" content="Initialize a map in an HTML
 element with MapLibre GL JS." />
            <meta charset='utf-8'>
            <meta name="viewport" content="width=device-width, initial-scale=1">
            <link rel='stylesheet' href='https://unpkg.com/maplibre-gl@4.x/dist/
maplibre-gl.css' />
            <link rel='stylesheet' href='style.css' />
            <script src='https://unpkg.com/maplibre-gl@4.x/dist/maplibre-gl.js'></
script>
        </head>
        <body>
            <!-- Map container -->
            <div id="map"></div>
            <script>
```

```
                const apiKey = "<API_KEY>";
                const mapStyle = "Standard";  // e.g., Standard, Monochrome, Hybrid,
 Satellite
                const awsRegion = "eu-central-1"; // e.g., us-east-2, us-east-1, us-
west-2, etc.
                const styleUrl = `https://maps.geo.${awsRegion}.amazonaws.com/v2/
styles/${mapStyle}/descriptor?key=${apiKey}`;

                const map = new maplibregl.Map({
                    container: 'map', // container id
                    style: styleUrl, // style URL
                    center: [0, 0], // starting position [lng, lat]
                    zoom: 1, // starting zoom
                });

                const marker = new maplibregl.Marker() // Create fixed marker
                    .setLngLat([85.1376, 25.5941]) // Set coordinates [long, lat]
 for marker (e.g., Patna, BR, IN)
                    .addTo(map); // Add marker to the map
            </script>
        </body>
    </html>
```

style.css

```
    body { margin: 0; padding: 0; }
    html, body, #map { height: 100%; }
```

添加可拖动标记

可拖动标记代码示例

index.html

```
    <!DOCTYPE html>
    <html lang="en">
        <head>
            <title>Add draggable marker on a map</title>
```

```
            <meta property="og:description" content="Initialize a map in an HTML
 element with MapLibre GL JS." />
            <meta charset='utf-8'>
            <meta name="viewport" content="width=device-width, initial-scale=1">
            <link rel='stylesheet' href='https://unpkg.com/maplibre-gl@4.x/dist/
maplibre-gl.css' />
            <link rel='stylesheet' href='style.css' />
            <script src='https://unpkg.com/maplibre-gl@4.x/dist/maplibre-gl.js'></
script>
        </head>
        <body>
            <!-- Map container -->
            <div id="map"></div>
            <script>
                const apiKey = "<API_KEY>";
                const mapStyle = "Standard";  // e.g., Standard, Monochrome, Hybrid,
 Satellite
                const awsRegion = "eu-central-1"; // e.g., us-east-2, us-east-1, us-
west-2, etc.
                const styleUrl = `https://maps.geo.${awsRegion}.amazonaws.com/v2/
styles/${mapStyle}/descriptor?key=${apiKey}`;

                const map = new maplibregl.Map({
                    container: 'map', // container id
                    style: styleUrl, // style URL
                    center: [0, 0], // starting position [lng, lat]
                    zoom: 1, // starting zoom
                });

                const marker = new maplibregl.Marker({ draggable: true }) // Create
 draggable marker
                    .setLngLat([85.1376, 25.5941]) // Set coordinates [long, lat]
 for marker (e.g., Patna, BR, IN)
                    .addTo(map); // Add marker to the map
            </script>
        </body>
    </html>
```

style.css

```
    body { margin: 0; padding: 0; }
```

```
    html, body, #map { height: 100%; }
```

更改标记颜色

彩色标记代码示例

index.html

```
    <!DOCTYPE html>
    <html lang="en">
        <head>
            <title>Adding colorful marker on a map</title>
            <meta property="og:description" content="Initialize a map in an HTML
 element with MapLibre GL JS." />
            <meta charset='utf-8'>
            <meta name="viewport" content="width=device-width, initial-scale=1">
            <link rel='stylesheet' href='https://unpkg.com/maplibre-gl@4.x/dist/
maplibre-gl.css' />
            <link rel='stylesheet' href='style.css' />
            <script src='https://unpkg.com/maplibre-gl@4.x/dist/maplibre-gl.js'></
script>
        </head>
        <body>
            <!-- Map container -->
            <div id="map"></div>
            <script>
                const apiKey = "<API_KEY>";
                const mapStyle = "Monochrome";  // e.g., Standard, Monochrome,
 Hybrid, Satellite
                const awsRegion = "eu-central-1"; // e.g., us-east-2, us-east-1, us-
west-2, etc.
                const styleUrl = `https://maps.geo.${awsRegion}.amazonaws.com/v2/
styles/${mapStyle}/descriptor?key=${apiKey}`;

                const map = new maplibregl.Map({
                    container: 'map', // container id
                    style: styleUrl, // style URL
                    center: [0, 0], // starting position [lng, lat]
                    zoom: 1, // starting zoom
                });
```

```
                const marker = new maplibregl.Marker({ color: "black" }) // Create
 colored marker
                    .setLngLat([85.1376, 25.5941]) // Set coordinates [long, lat]
 for marker (e.g., Patna, BR, IN)
                    .addTo(map); // Add marker to the map
            </script>
        </body>
    </html>
```

style.css

```
    body { margin: 0; padding: 0; }
    html, body, #map { height: 100%; }
```

添加多个标记

多标记代码示例

index.html

```
    <!DOCTYPE html>
    <html lang="en">
        <head>
            <title>Adding multiple markers on a map</title>
            <meta property="og:description" content="Initialize a map in an HTML
 element with MapLibre GL JS." />
            <meta charset='utf-8'>
            <meta name="viewport" content="width=device-width, initial-scale=1">
            <link rel='stylesheet' href='https://unpkg.com/maplibre-gl@4.x/dist/
maplibre-gl.css' />
            <link rel='stylesheet' href='style.css' />
            <script src='https://unpkg.com/maplibre-gl@4.x/dist/maplibre-gl.js'></
script>
        </head>
        <body>
            <!-- Map container -->
            <div id="map"></div>
            <script>
```

```
                const apiKey = "<API_KEY>";
                const mapStyle = "Standard";  // e.g., Standard, Monochrome, Hybrid,
 Satellite
                const awsRegion = "eu-central-1"; // e.g., us-east-2, us-east-1, us-
west-2, etc.
                const colorScheme ="Dark"; // e.g., Dark, Light (default)
                const styleUrl = `https://maps.geo.${awsRegion}.amazonaws.com/v2/
styles/${mapStyle}/descriptor?color-scheme=${colorScheme}&key=${apiKey}`;

                const map = new maplibregl.Map({
                    container: 'map', // container id
                    style: styleUrl, // style URL
                    center: [0, 0], // starting position [lng, lat]
                    zoom: 1, // starting zoom
                });

                const locations = [
                    { lng: 85.1376, lat: 25.5941, label: 'Patna', color:
 '#FF5722' },
                    { lng: 77.1025, lat: 28.7041, label: 'Delhi', color:
 '#2196F3' },
                    { lng: 77.5946, lat: 12.9716, label: 'Bangalore', color:
 '#FF9800' },
                    { lng: 78.4867, lat: 17.3850, label: 'Hyderabad', color:
 '#9C27B0' },
                    { lng: -87.6298, lat: 41.8781, label: 'Chicago', color:
 '#4CAF50' },
                    { lng: -122.3321, lat: 47.6062, label: 'Seattle', color:
 '#FFC107' },
                    { lng: 4.3517, lat: 50.8503, label: 'Brussels', color:
 '#3F51B5' },
                    { lng: 2.3522, lat: 48.8566, label: 'Paris', color: '#E91E63' },

                    { lng: -0.1276, lat: 51.5074, label: 'London', color:
 '#795548' },
                    { lng: 28.0473, lat: -26.2041, label: 'Johannesburg', color:
 '#673AB7' },
                    { lng: -123.1216, lat: 49.2827, label: 'Vancouver', color:
 '#FF5722' },
                    { lng: -104.9903, lat: 39.7392, label: 'Denver', color:
 '#FF9800' },
                    { lng: -97.7431, lat: 30.2672, label: 'Austin', color:
 '#3F51B5' }
                ];
```

```
                // Loop through the locations array and add a marker for each one
                locations.forEach(location => {
                    const marker = new maplibregl.Marker({ color: location.color,
draggable: true }) // Create colored marker
                        .setLngLat([location.lng, location.lat]) // Set longitude
and latitude
                        .addTo(map); // Add marker to the map
                });
            </script>
        </body>
    </html>
```

style.css

```
    body { margin: 0; padding: 0; }
    html, body, #map { height: 100%; }
```

如何在地图上添加图标

Amazon Location Service 允许您在地图上添加图标，最好是 PNG 格式。您可以向特定地理位置添加图标并根据需要设置其样式。

添加静态图标

在本示例中，您将使用外部 URL 通过符号图层向地图添加图标。

静态图标代码示例

index.html

```
    <!DOCTYPE html>
    <html lang="en">
        <head>
            <title>Static icon on map</title>
            <meta property="og:description" content="Initialize a map in an HTML
element with MapLibre GL JS." />
            <meta charset='utf-8'>
```

```
            <meta name="viewport" content="width=device-width, initial-scale=1">
            <link rel='stylesheet' href='https://unpkg.com/maplibre-gl@4.x/dist/
maplibre-gl.css' />
            <link rel='stylesheet' href='style.css' />
            <script src='https://unpkg.com/maplibre-gl@4.x/dist/maplibre-gl.js'></
script>
        </head>
        <body>
            <!-- Map container -->
            <div id="map"></div>
            <script>
                const apiKey = "<API_KEY>";
                const mapStyle = "Standard";  // e.g., Standard, Monochrome, Hybrid,
 Satellite
                const awsRegion = "eu-central-1"; // e.g., us-east-2, us-east-1, us-
west-2, etc.
                const styleUrl = `https://maps.geo.${awsRegion}.amazonaws.com/v2/
styles/${mapStyle}/descriptor?key=${apiKey}`;

                const map = new maplibregl.Map({
                    container: 'map', // container id
                    style: styleUrl, // style URL
                    center: [-123.1144289, 49.2806672], // starting position [lng,
 lat] (e.g., Vancouver)
                    zoom: 12, // starting zoom
                });

                map.on('load', async () => {
                    image = await map.loadImage('https://upload.wikimedia.org/
wikipedia/commons/1/1e/Biking_other.png');
                    map.addImage('biking', image.data);
                    map.addSource('point', {
                        'type': 'geojson',
                        'data': {
                            'type': 'FeatureCollection',
                            'features': [
                                {
                                    'type': 'Feature',
                                    'geometry': {
                                        'type': 'Point',
                                        'coordinates': [-123.1144289, 49.2806672]
                                    }
                                }
                            ]
```

```
                    }
                });
                map.addLayer({
                    'id': 'points',
                    'type': 'symbol',
                    'source': 'point',
                    'layout': {
                        'icon-image': 'biking',
                        'icon-size': 0.25
                    }
                });
            });
        </script>
    </body>
</html>
```

style.css

```
body { margin: 0; padding: 0; }
html, body, #map { height: 100%; }
```

如何在地图上添加一条线

借助 Amazon Location Service，您可以将预先记录的 GPS 轨迹添加为线串，也可以将实时 GPS 轨迹添加到动态地图中。

添加预先录制的台词

在本示例中，您将预先记录的 GPS 轨迹作为 geoJSON (main.js) 添加到动态地图。为此，您需要向地图添加一个源（如 GeoJSON）和一个带有您选择的线条样式的图层。

预先录制的线路代码示例

index.html

```
<!DOCTYPE html>
<html lang="en">
    <head>
```

```
            <title>Add a line on the map</title>
            <meta property="og:description" content="Initialize a map in an HTML
 element with MapLibre GL JS." />
            <meta charset='utf-8'>
            <meta name="viewport" content="width=device-width, initial-scale=1">
            <link rel='stylesheet' href='https://unpkg.com/maplibre-gl@4.x/dist/
maplibre-gl.css' />
            <link rel='stylesheet' href='style.css' />
            <script src='https://unpkg.com/maplibre-gl@4.x/dist/maplibre-gl.js'></
script>
            <script src='main.js'></script>
        </head>
        <body>
            <!-- Map container -->
            <div id="map"></div>
            <script>
                const apiKey = "<API_KEY>";
                const mapStyle = "Standard";
                const awsRegion = "eu-central-1";
                const styleUrl = `https://maps.geo.${awsRegion}.amazonaws.com/v2/
styles/${mapStyle}/descriptor?key=${apiKey}`;

                const map = new maplibregl.Map({
                    container: 'map',
                    style: styleUrl,
                    center: [-123.126575, 49.290585],
                    zoom: 12.5
                });

                map.on('load', () => {
                    map.addSource('vancouver-office-to-stanley-park-route', {
                        type: 'geojson',
                        data: routeGeoJSON
                    });

                    map.addLayer({
                        id: 'vancouver-office-to-stanley-park-route',
                        type: 'line',
                        source: 'vancouver-office-to-stanley-park-route',
                        layout: {
                            'line-cap': 'round',
                            'line-join': 'round'
                        },
                        paint: {
```

```
                        'line-color': '#008296',
                        'line-width': 2
                    }
                });
            });
        </script>
    </body>
</html>
```

style.css

```
body { margin: 0; padding: 0; }
html, body, #map { height: 100%; }
```

main.js

```
const routeGeoJSON = {
    type: "FeatureCollection",
    features: [
        {
            type: "Feature",
            geometry: {
                type: "LineString",
                coordinates: [
                    [-123.117011, 49.281306],
                    [-123.11785, 49.28011],
                    ...
                    [-123.135735, 49.30106]
                ]
            },
            properties: {
                name: "Amazon YVR to Stanley Park",
                description: "An evening walk from Amazon office to Stanley
Park."
            }
        }
    ]
};
```

实时添加一行

在本示例中，您将模拟逐个添加新的 GPS 坐标以创建实时 GPS 追踪。这对于跟踪实时数据更新非常有用。

实时线路代码示例

index.html

```
    <!DOCTYPE html>
    <html lang="en">
        <head>
            <title>Add a line on the map in real-time</title>
            <meta property="og:description" content="Initialize a map in an HTML
 element with MapLibre GL JS." />
            <meta charset='utf-8'>
            <meta name="viewport" content="width=device-width, initial-scale=1">
            <link rel='stylesheet' href='https://unpkg.com/maplibre-gl@4.x/dist/
maplibre-gl.css' />
            <link rel='stylesheet' href='style.css' />
            <script src='https://unpkg.com/maplibre-gl@4.x/dist/maplibre-gl.js'></
script>
            <script src='main.js'></script>
        </head>
        <body>
            <!-- Map container -->
            <div id="map"></div>
            <script>
                const apiKey = "<API_KEY>";
                const mapStyle = "Standard";
                const awsRegion = "eu-central-1";
                const styleUrl = `https://maps.geo.${awsRegion}.amazonaws.com/v2/
styles/${mapStyle}/descriptor?key=${apiKey}`;

                const map = new maplibregl.Map({
                    container: 'map',
                    style: styleUrl,
                    center: [-123.126575, 49.290585],
                    zoom: 12.5
                });

                map.on('load', () => {
```

```
                    const coordinates =
 routeGeoJSON.features[0].geometry.coordinates;
                    routeGeoJSON.features[0].geometry.coordinates = [coordinates[0],
 coordinates[20]];

                    map.addSource('vancouver-office-to-stanley-park-route', {
                        type: 'geojson',
                        data: routeGeoJSON
                    });

                    map.addLayer({
                        id: 'vancouver-office-to-stanley-park-route',
                        type: 'line',
                        source: 'vancouver-office-to-stanley-park-route',
                        layout: {
                            'line-cap': 'round',
                            'line-join': 'round'
                        },
                        paint: {
                            'line-color': '#008296',
                            'line-width': 2
                        }
                    });

                    map.jumpTo({center: coordinates[0], zoom: 14});
                    map.setPitch(30);

                    let i = 0;
                    const timer = window.setInterval(() => {
                        if (i < coordinates.length) {

 routeGeoJSON.features[0].geometry.coordinates.push(coordinates[i]);
                            map.getSource('vancouver-office-to-stanley-park-
route').setData(routeGeoJSON);
                            map.panTo(coordinates[i]);
                            i++;
                        } else {
                            window.clearInterval(timer);
                        }
                    }, 100);
                });
            </script>
        </body>
    </html>
```

style.css

```
body { margin: 0; padding: 0; }
html, body, #map { height: 100%; }
```

main.js

```
const routeGeoJSON = {
    type: "FeatureCollection",
    features: [
        {
            type: "Feature",
            geometry: {
                type: "LineString",
                coordinates: [
                    [-123.117011, 49.281306],
                    [-123.11785, 49.28011],
                    ...
                    [-123.135735, 49.30106]
                ]
            },
            properties: {
                name: "Amazon YVR to Stanley Park",
                description: "An evening walk from Amazon office to Stanley
Park."
            }
        }
    ]
};
```

开发者小贴士

拟合边界：您可以通过计算直线坐标的边界来使直线与地图边界相适应：

```
const coordinates = routeGeoJSON.features[0].geometry.coordinates;
```

```
    const bounds = coordinates.reduce((bounds, coord) => bounds.extend(coord), new
maplibregl.LngLatBounds(coordinates[0], coordinates[0]));
    map.fitBounds(bounds, { padding: 20 });
```

如何在地图上添加多边形

Amazon Location Service 允许您在地图上添加多边形。您可以根据业务需求设置多边形的样式，包括添加填充和边框样式。

添加多边形

在本示例中，您将向地图添加一个多边形，并使用填充颜色和边框来设置其样式。

多边形代码示例

index.html

```
    <!DOCTYPE html>
    <html lang="en">
        <head>
            <title>Overlay a Polygon on a Map</title>
            <meta property="og:description" content="Initialize a map in an HTML
 element with MapLibre GL JS." />
            <meta charset='utf-8'>
            <meta name="viewport" content="width=device-width, initial-scale=1">
            <link rel='stylesheet' href='https://unpkg.com/maplibre-gl@4.x/dist/
maplibre-gl.css' />
            <link rel='stylesheet' href='style.css' />
            <script src='https://unpkg.com/maplibre-gl@4.x/dist/maplibre-gl.js'></
script>
            <script src='main.js'></script>
        </head>
        <body>
            <!-- Map container -->
            <div id="map"></div>
            <script>
                const apiKey = "<API_KEY>";
                const mapStyle = "Standard";
                const awsRegion = "eu-central-1";
                const styleUrl = `https://maps.geo.${awsRegion}.amazonaws.com/v2/
styles/${mapStyle}/descriptor?key=${apiKey}`;
```

```
                const map = new maplibregl.Map({
                    container: 'map',
                    style: styleUrl,
                    center: [-123.13203602550998, 49.28726257639254],
                    zoom: 16
                });

                map.on('load', () => {
                    map.addSource('barclayHeritageSquare', {
                        type: 'geojson',
                        data: barclayHeritageSquare
                    });

                    map.addLayer({
                        id: 'barclayHeritageSquare-fill',
                        type: 'fill',
                        source: 'barclayHeritageSquare',
                        layout: {},
                        paint: {
                            'fill-color': '#008296',
                            'fill-opacity': 0.25
                        }
                    });

                    map.addLayer({
                        id: 'barclayHeritageSquare-outline',
                        type: 'line',
                        source: 'barclayHeritageSquare',
                        layout: {},
                        paint: {
                            'line-color': '#008296',
                            'line-width': 2
                        }
                    });
                });
            </script>
        </body>
    </html>
```

style.css

```
    body { margin: 0; padding: 0; }
    html, body, #map { height: 100%; }
```

main.js

```
    const barclayHeritageSquare = {
        "type": "FeatureCollection",
        "features": [
            {
                "type": "Feature",
                "properties": {
                    "park_id": 200,
                    "park_name": "Barclay Heritage Square",
                    "area_ha": 0.63255076039675,
                    "park_url": "http://covapp.vancouver.ca/parkfinder/
parkdetail.aspx?inparkid=200"
                },
                "geometry": {
                    "type": "MultiPolygon",
                    "coordinates": [
                        [
                            [-123.1320948511985, 49.287379401361406],
                            [-123.13179672786798, 49.2871908081159],
                            [-123.13148154587924, 49.28739992437733],
                            [-123.1313830551372, 49.28733617069321],
                            [-123.13118648745055, 49.287208851500054],
                            [-123.13128257706838, 49.28714532642171],
                            [-123.13147941881572, 49.28727228533418],
                            ...
                            [-123.13177619357138, 49.28759081706052],
                            [-123.1320948511985, 49.287379401361406]
                        ]
                    ]
                }
            }
        ]
    };
```

如何向地图添加弹出窗口

Amazon Location Service 允许您在地图上添加弹出窗口。您可以添加简单的弹出窗口、标记上的点击触发弹出窗口、悬停触发的弹出窗口以及用于多个标记的弹出窗口。

添加你的第一个弹出窗口

在此示例中，您将添加第一个弹出窗口。

第一个弹出代码示例

index.html

```
    <!DOCTYPE html>
    <html lang="en">
        <head>
            <title>Display a map</title>
            <meta property="og:description" content="Initialize a map in an HTML
 element with MapLibre GL JS." />
            <meta charset='utf-8'>
            <meta name="viewport" content="width=device-width, initial-scale=1">
            <link rel='stylesheet' href='https://unpkg.com/maplibre-gl@4.x/dist/
maplibre-gl.css' />
            <link rel='stylesheet' href='style.css' />
            <script src='https://unpkg.com/maplibre-gl@4.x/dist/maplibre-gl.js'></
script>
        </head>
        <body>
            <!-- Map container -->
            <div id="map"></div>
            <script>
                const apiKey = "<API_KEY>";
                const mapStyle = "Standard";
                const awsRegion = "eu-central-1";
                const styleUrl = `https://maps.geo.${awsRegion}.amazonaws.com/v2/
styles/${mapStyle}/descriptor?key=${apiKey}`;

                const map = new maplibregl.Map({
                    container: 'map',
                    style: styleUrl,
                    center: [-96, 37.8],
                    zoom: 2
                });
```

```
                const popup = new maplibregl.Popup({closeOnClick: false})
                    .setLngLat([-96, 37.8])
                    .setHTML('<h1>Hello USA!</h1>')
                    .addTo(map);
            </script>
        </body>
    </html>
```

style.css

```
    body { margin: 0; padding: 0; }
    html, body, #map { height: 100%; }
```

点击标记时显示弹出窗口

在此示例中，您将在标记上附加一个弹出窗口，并在单击时显示该弹出窗口。

标记点击时弹出示例

index.html

```
    <!DOCTYPE html>
    <html lang="en">
        <head>
            <title>Display a map</title>
            <meta property="og:description" content="Initialize a map in an HTML
 element with MapLibre GL JS." />
            <meta charset='utf-8'>
            <meta name="viewport" content="width=device-width, initial-scale=1">
            <link rel='stylesheet' href='https://unpkg.com/maplibre-gl@4.x/dist/
maplibre-gl.css' />
            <link rel='stylesheet' href='style.css' />
            <script src='https://unpkg.com/maplibre-gl@4.x/dist/maplibre-gl.js'></
script>
        </head>
        <body>
            <!-- Map container -->
```

```
            <div id="map"></div>
            <script>
                const apiKey = "<API_KEY>";
                const mapStyle = "Standard";
                const awsRegion = "eu-central-1";
                const styleUrl = `https://maps.geo.${awsRegion}.amazonaws.com/v2/
styles/${mapStyle}/descriptor?key=${apiKey}`;

                const centralpark_nyc = [-73.966,40.781];
                const map = new maplibregl.Map({
                    container: 'map',
                    style: styleUrl,
                    center: centralpark_nyc,
                    zoom: 13
                });

                const popup = new maplibregl.Popup({offset: 25}).setText(
                    'Central Park, NY is one of the most filmed locations in the
 world, appearing in over 240 feature films since 1908.'
                );

                new maplibregl.Marker()
                    .setLngLat(centralpark_nyc)
                    .setPopup(popup)
                    .addTo(map);
            </script>
        </body>
    </html>
```

style.css

```
    body { margin: 0; padding: 0; }
    html, body, #map { height: 100%; }
```

将鼠标悬停在标记上时显示弹出窗口

在此示例中，您将在标记上附加一个弹出窗口，并在悬停时显示该弹出窗口。

标记悬停时弹出示例

index.html

```
    <!DOCTYPE html>
    <html lang="en">
        <head>
            <title>Display a map</title>
            <meta property="og:description" content="Initialize a map in an HTML
 element with MapLibre GL JS." />
            <meta charset='utf-8'>
            <meta name="viewport" content="width=device-width, initial-scale=1">
            <link rel='stylesheet' href='https://unpkg.com/maplibre-gl@4.x/dist/
maplibre-gl.css' />
            <link rel='stylesheet' href='style.css' />
            <script src='https://unpkg.com/maplibre-gl@4.x/dist/maplibre-gl.js'></
script>
        </head>
        <body>
            <!-- Map container -->
            <div id="map"></div>
            <script>
                const apiKey = "<API_KEY>";
                const mapStyle = "Standard";
                const awsRegion = "eu-central-1";
                const styleUrl = `https://maps.geo.${awsRegion}.amazonaws.com/v2/
styles/${mapStyle}/descriptor?key=${apiKey}`;

                const centralpark_nyc = [-73.966,40.781];
                const map = new maplibregl.Map({
                    container: 'map',
                    style: styleUrl,
                    center: centralpark_nyc,
                    zoom: 13
                });

                const marker = new maplibregl.Marker().setLngLat([-73.968285,
 40.785091]).addTo(map);
                const popup = new maplibregl.Popup({ offset: 25 })
                    .setHTML("<h3>Central Park</h3><p>Welcome to Central Park, NYC!
</p>");

                const markerElement = marker.getElement();
```

```
                markerElement.addEventListener('mouseenter', () => {
                    popup.setLngLat([-73.968285, 40.785091]).addTo(map);
                });
                markerElement.addEventListener('mouseleave', () => {
                    popup.remove();
                });
            </script>
        </body>
    </html>
```

style.css

```
    body { margin: 0; padding: 0; }
    html, body, #map { height: 100%; }
```

点击多个标记时显示弹出窗口

在此示例中，您将在多个标记上附加一个弹出窗口，并在单击时显示该弹出窗口。

点击多个标记时弹出示例

index.html

```
        <!DOCTYPE html>
        <html lang="en">
            <head>
                <title>Display a map</title>
                <meta property="og:description" content="Initialize a map in an HTML
 element with MapLibre GL JS." />
                <meta charset='utf-8'>
                <meta name="viewport" content="width=device-width, initial-scale=1">
                <link rel='stylesheet' href='https://unpkg.com/maplibre-gl@4.x/dist/
maplibre-gl.css' />
                <link rel='stylesheet' href='style.css' />
                <script src='https://unpkg.com/maplibre-gl@4.x/dist/maplibre-
gl.js'></script>
            </head>
            <body>
```

```
                <!-- Map container -->
                <div id="map"></div>
                <script>
                    const apiKey = "Your API Key";
                    const mapStyle = "Monochrome";
                    const awsRegion = "eu-central-1";
                    const colorScheme ="Light";
                    const styleUrl = `https://maps.geo.${awsRegion}.amazonaws.com/
v2/styles/${mapStyle}/descriptor?color-scheme=${colorScheme}&key=${apiKey}`;

                    const map = new maplibregl.Map({
                        container: 'map',
                        style: styleUrl,
                        center: [-123.126979, 49.2841563],
                        zoom: 15,
                        minZoom: 13,
                        maxZoom: 17
                    });

                    const locations = [
                        { id: 1, lat: 49.281108, lng: -123.117049, name: "Amazon -
 YVR11 office" },
                        { id: 2, lat: 49.285580, lng: -123.115806, name: "Amazon -
 YVR20 office" },
                        { id: 3, lat: 49.281661, lng: -123.114174, name: "Amazon -
 YVR14 office" },
                        { id: 4, lat: 49.280663, lng: -123.114379, name: "Amazon -
 YVR26 office" },
                        { id: 5, lat: 49.285343, lng: -123.129119, name: "Amazon -
 YVR25 office" }
                    ];

                    const geojson = {
                        type: "FeatureCollection",
                        features: locations.map(location => ({
                            type: "Feature",
                            properties: { id: location.id, name: location.name },
                            geometry: {
                                type: "Point",
                                coordinates: [location.lng, location.lat]
                            }
                        }))
                    };
```

```
                    map.on('load', async () => {
                        try {
                            const image = await loadImage('https://
upload.wikimedia.org/wikipedia/commons/thumb/9/93/
Amazon_Web_Services_Logo.svg/1200px-Amazon_Web_Services_Logo.svg.png');
                            map.addImage('aws', image);

                            map.addSource('places', { type: 'geojson', data:
 geojson });

                            map.addLayer({
                                'id': 'places',
                                'type': 'symbol',
                                'source': 'places',
                                'layout': {
                                    'icon-image': 'aws',
                                    'icon-size': 0.025,
                                    'icon-allow-overlap': true
                                }
                            });

                            const popup = new maplibregl.Popup({ closeButton: false,
 closeOnClick: false });

                            map.on('click', 'places', (e) => {
                                map.getCanvas().style.cursor = 'pointer';
                                const coordinates =
 e.features[0].geometry.coordinates.slice();
                                const name = e.features[0].properties.name;

 popup.setLngLat(coordinates).setHTML(name).addTo(map);
                            });

                            map.on('mouseleave', 'places', () => {
                                map.getCanvas().style.cursor = '';
                                popup.remove();
                            });
                        } catch (error) {
                            console.error('Error loading image:', error);
                        }
                    });

                    async function loadImage(url) {
                        return new Promise((resolve, reject) => {
```

```
                            const img = new Image();
                            img.crossOrigin = 'anonymous';
                            img.onload = () => resolve(img);
                            img.onerror = (error) => reject(error);
                            img.src = url;
                        });
                }
            </script>
        </body>
    </html>
```

style.css

```
    body { margin: 0; padding: 0; }
    html, body, #map { height: 100%; }
```

将鼠标悬停在多个标记上时显示弹出窗口

在此示例中，您将在多个标记上附加一个弹出窗口，并在悬停时显示该弹出窗口。

将鼠标悬停在多个标记上时弹出示例

index.html

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Display a map</title>
        <meta property="og:description" content="Initialize a map in an HTML element
 with MapLibre GL JS." />
        <meta charset='utf-8'>
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <link rel='stylesheet' href='https://unpkg.com/maplibre-gl@4.x/dist/
maplibre-gl.css' />
        <link rel='stylesheet' href='style.css' />
        <script src='https://unpkg.com/maplibre-gl@4.x/dist/maplibre-gl.js'></
script>
    </head>
```

```
    <body>
        <!-- Map container -->
        <div id="map" style="width: 100%; height: 100vh;"></div>
        <script>
            const apiKey = "You API Key";
            const mapStyle = "Monochrome";  // eg. Standard, Monochrome, Hybrid,
 Satellite
            const awsRegion = "eu-central-1"; // eg. us-east-2, us-east-1, etc.
            const colorScheme ="Light"; // eg Dark, Light (default)
            const styleUrl = `https://maps.geo.${awsRegion}.amazonaws.com/v2/styles/
${mapStyle}/descriptor?color-scheme=${colorScheme}&key=${apiKey}`;

            // Initialize the map
            const map = new maplibregl.Map({
                container: 'map', // Container id
                style: styleUrl, // Style URL
                center: [-123.126979, 49.2841563], // Starting position [lng, lat]
                zoom: 15, // Starting zoom
                minZoom: 13, // Minimum zoom level
                maxZoom: 17 // Maximum zoom level
            });

            const locations = [
                { id: 1, lat: 49.281108, lng: -123.117049, name: "Amazon - YVR11
 office" },
                { id: 2, lat: 49.285580, lng: -123.115806, name: "Amazon - YVR20
 office" },
                { id: 3, lat: 49.281661, lng: -123.114174, name: "Amazon - YVR14
 office" },
                { id: 4, lat: 49.280663, lng: -123.114379, name: "Amazon - YVR26
 office" },
                { id: 5, lat: 49.285343, lng: -123.129119, name: "Amazon - YVR25
 office" }
            ];

            // Convert locations to GeoJSON
            const geojson = {
                type: "FeatureCollection",
                features: locations.map(location => ({
                    type: "Feature",
                    properties: {
                        id: location.id,
                        name: location.name // Use the name property for popup
                    },
```

```
                    geometry: {
                        type: "Point",
                        coordinates: [location.lng, location.lat] // GeoJSON uses
 [lng, lat]
                    }
                }))
            };

            // Add the GeoJSON source and layers when the map loads
            map.on('load', async () => {
                try {
                    // Load the PNG image for the marker
                    const image = await loadImage('https://upload.wikimedia.org/
wikipedia/commons/thumb/9/93/Amazon_Web_Services_Logo.svg/1200px-
Amazon_Web_Services_Logo.svg.png'); // Ensure this URL points to a valid PNG
                    map.addImage('aws', image);

                    // Add a GeoJSON source
                    map.addSource('places', {
                        type: 'geojson',
                        data: geojson
                    });

                    // Add a layer showing the places with custom icons
                    map.addLayer({
                        'id': 'places',
                        'type': 'symbol',
                        'source': 'places',
                        'layout': {
                            'icon-image': 'aws',
                            'icon-size': 0.025, // Adjust as needed
                            'icon-allow-overlap': true // Allow icons to overlap
                        }
                    });

                    // Create a popup
                    const popup = new maplibregl.Popup({
                        closeButton: false,
                        closeOnClick: false
                    });

                    // Event handlers for mouse enter and leave
                    map.on('mouseenter', 'places', (e) => {
                        map.getCanvas().style.cursor = 'pointer';
```

```
                        const coordinates =
 e.features[0].geometry.coordinates.slice();
                        const name = e.features[0].properties.name;

                        // Set popup content and position
                        popup.setLngLat(coordinates).setHTML(name).addTo(map);
                    });

                    map.on('mouseleave', 'places', () => {
                        map.getCanvas().style.cursor = '';
                        popup.remove();
                    });
                } catch (error) {
                    console.error('Error loading image:', error);
                }
            });

            // Helper function to load the image
            async function loadImage(url) {
                return new Promise((resolve, reject) => {
                    const img = new Image();
                    img.crossOrigin = 'anonymous'; // Enable CORS
                    img.onload = () => resolve(img);
                    img.onerror = (error) => reject(error);
                    img.src = url;
                });
            }
        </script>
    </body>
</html>
```

style.css

```
        body { margin: 0; padding: 0; }
        html, body, #map { height: 100%; }
```

如何为地图设置首选语言

Amazon Location Service 允许您通过更新特定语言的样式描述符在客户端设置首选语言。您可以设置首选语言，并以嵌入的该语言显示内容。否则，它将退回到另一种语言。

> **Note**
>
> 有关更多信息，请参阅 the section called “本地化和国际化”。

将首选语言设置为日语并显示日本地图

在本示例中，您将设置更新样式以日语 (ja) 显示地图标签。

将首选语言设置为日语示例

index.html

```
   <html>
   <head>
     <link href="https://unpkg.com/maplibre-gl@4.x/dist/maplibre-gl.css"
rel="stylesheet" />
     <script src="https://unpkg.com/maplibre-gl@4.x/dist/maplibre-gl.js"></script>
     <link href="style.css" rel="stylesheet" />
     <script src="main.js"></script>
   </head>
   <body>
     <div id="map" />
     <script>
       const apiKey = "Add Your Api Key";
       const mapStyle = "Standard";
       const awsRegion = "eu-central-1";
       const initialLocation = [139.76694, 35.68085]; //Japan

       async function initializeMap() {
          // get updated style object for preferred language.
           const styleObject = await getStyleWithPreferredLanguage("ja");
          // Initialize the MapLibre map with the fetched style object
           const map = new maplibregl.Map({
               container: 'map',
               style: styleObject,
               center: initialLocation,
```

```
                zoom: 15,
                hash:true,
            });
            map.addControl(new maplibregl.NavigationControl(), "top-left");

            return map;
        }

        initializeMap();
      </script>
    </body>
    </html>
```

style.css

```
    body { margin: 0; }
    #map { height: 100vh; }
```

main.js

```
    async function getStyleWithPreferredLanguage(preferredLanguage) {
      const styleUrl = `https://maps.geo.${awsRegion}.amazonaws.com/v2/styles/
${mapStyle}/descriptor?key=${apiKey}`;

      return fetch(styleUrl)
        .then(response => {
          if (!response.ok) {
            throw new Error('Network response was not ok ' + response.statusText);
          }
          return response.json();
        })
        .then(styleObject => {
          if (preferredLanguage !== "en") {
            styleObject = setPreferredLanguage(styleObject, preferredLanguage);
          }

          return styleObject;
        })
        .catch(error => {
```

```
      console.error('Error fetching style:', error);
    });
}

const setPreferredLanguage = (style, language) => {
  let nextStyle = { ...style };

  nextStyle.layers = nextStyle.layers.map(l => {
    if (l.type !== 'symbol' || !l?.layout?.['text-field']) return l;
    return updateLayer(l, /^name:([A-Za-z\-\_]+)$/g, `name:${language}`);
  });

  return nextStyle;
};

const updateLayer = (layer, prevPropertyRegex, nextProperty) => {
  const nextLayer = {
    ...layer,
    layout: {
      ...layer.layout,
      'text-field': recurseExpression(
        layer.layout['text-field'],
        prevPropertyRegex,
        nextProperty
      )
    }
  };
  return nextLayer;
};

const recurseExpression = (exp, prevPropertyRegex, nextProperty) => {
  if (!Array.isArray(exp)) return exp;
  if (exp[0] !== 'coalesce') return exp.map(v =>
    recurseExpression(v, prevPropertyRegex, nextProperty)
  );

  const first = exp[1];
  const second = exp[2];

  let isMatch =
  Array.isArray(first) &&
  first[0] === 'get' &&
  !!first[1].match(prevPropertyRegex)?.[0];
```

```
    isMatch = isMatch && Array.isArray(second) && second[0] === 'get';
    isMatch = isMatch && !exp?.[4];

    if (!isMatch) return exp.map(v =>
      recurseExpression(v, prevPropertyRegex, nextProperty)
    );

    return [
      'coalesce',
      ['get', nextProperty],
      ['get', 'name:en'],
      ['get', 'name']
    ];
  };
```

根据最终用户的浏览器语言设置首选语言

在本示例中，您将设置更新样式以使用用户的设备语言显示地图标签。

根据浏览器语言示例设置首选语言

index.html

```
  <html>
  <head>
    <link href="https://unpkg.com/maplibre-gl@4.x/dist/maplibre-gl.css"
rel="stylesheet" />
    <script src="https://unpkg.com/maplibre-gl@4.x/dist/maplibre-gl.js"></script>
    <link href="style.css" rel="stylesheet" />
    <script src="main.js"></script>
  </head>
  <body>
    <div id="map" />
    <script>
      const apiKey = "Add Your Api Key";
      const mapStyle = "Standard";
      const awsRegion = "eu-central-1";
      const initialLocation = [139.76694, 35.68085]; //Japan
      const userLanguage = navigator.language || navigator.userLanguage;
      const languageCode = userLanguage.split('-')[0];
```

```
        async function initializeMap() {
           const styleObject = await getStyleWithPreferredLanguage(languageCode);
           const map = new maplibregl.Map({
                container: 'map',
                style: styleObject,
                center: initialLocation,
                zoom: 15,
                hash:true,
           });
           map.addControl(new maplibregl.NavigationControl(), "top-left");
           return map;
        }

        initializeMap();
      </script>
    </body>
    </html>
```

style.css

```
    body { margin: 0; }
    #map { height: 100vh; }
```

main.js

```
    async function getStyleWithPreferredLanguage(preferredLanguage) {
      const styleUrl = `https://maps.geo.${awsRegion}.amazonaws.com/v2/styles/
${mapStyle}/descriptor?key=${apiKey}`;

      return fetch(styleUrl)
        .then(response => {
          if (!response.ok) {
            throw new Error('Network response was not ok ' + response.statusText);
          }
          return response.json();
        })
        .then(styleObject => {
          if (preferredLanguage !== "en") {
            styleObject = setPreferredLanguage(styleObject, preferredLanguage);
```

```
      }

      return styleObject;
    })
    .catch(error => {
      console.error('Error fetching style:', error);
    });
}

const setPreferredLanguage = (style, language) => {
  let nextStyle = { ...style };

  nextStyle.layers = nextStyle.layers.map(l => {
    if (l.type !== 'symbol' || !l?.layout?.['text-field']) return l;
    return updateLayer(l, /^name:([A-Za-z\-\_]+)$/g, `name:${language}`);
  });

  return nextStyle;
};

const updateLayer = (layer, prevPropertyRegex, nextProperty) => {
  const nextLayer = {
    ...layer,
    layout: {
      ...layer.layout,
      'text-field': recurseExpression(
        layer.layout['text-field'],
        prevPropertyRegex,
        nextProperty
      )
    }
  };
  return nextLayer;
};

const recurseExpression = (exp, prevPropertyRegex, nextProperty) => {
  if (!Array.isArray(exp)) return exp;
  if (exp[0] !== 'coalesce') return exp.map(v =>
    recurseExpression(v, prevPropertyRegex, nextProperty)
  );

  const first = exp[1];
  const second = exp[2];
```

```
    let isMatch =
    Array.isArray(first) &&
    first[0] === 'get' &&
    !!first[1].match(prevPropertyRegex)?.[0];

    isMatch = isMatch && Array.isArray(second) && second[0] === 'get';
    isMatch = isMatch && !exp?.[4];

    if (!isMatch) return exp.map(v =>
      recurseExpression(v, prevPropertyRegex, nextProperty)
    );

    return [
      'coalesce',
      ['get', nextProperty],
      ['get', 'name:en'],
      ['get', 'name']
    ];
  };
```

如何设置地图的政治视角

Amazon Location Service 使您能够设定政治观点，确保您的申请符合当地法律。你可以设定政治观点，比较不同政治视角的地图。

> **Note**
>
> 有关更多信息，请参阅 the section called “本地化和国际化”。

设定政治观点并与国际视角进行比较

在本示例中，您将根据两种不同的政治观点创建和比较地图：国际视角和土耳其对塞浦路斯的看法。

政治观点比较示例

index.html

```
<!DOCTYPE html>
<html lang="en">
```

```
                            <head>
                                <title>Display a map</title>
                                <meta property="og:description" content="Initialize
 a map in an HTML element with MapLibre GL JS." />
                                <meta charset='utf-8'>
                                <meta name="viewport" content="width=device-width,
 initial-scale=1">
                                <link rel='stylesheet' href='https://unpkg.com/
maplibre-gl@4.x/dist/maplibre-gl.css' />
                                <link rel='stylesheet' href='style.css' />
                                <script src='https://unpkg.com/maplibre-gl@4.x/dist/
maplibre-gl.js'></script>
                                <script src="https://unpkg.com/@mapbox/mapbox-gl-
sync-move@0.3.1"></script>
                                <script src='main.js'></script>
                            </head>
                            <body>
                                <!-- Map container -->
                                <div class="maps">
                                <div id="internatinalView"></div>
                                <div id="turkeyView"></div>
                                </div>
                                <script>

                                    const apiKey = "Add Your Api Key";
                                    const mapStyle = "Standard";
                                    const awsRegion = "eu-central-1";

                                    // International perspective without political-
view query parameter
                                    const internatinalViewMapStyleUrl = `https://
maps.geo.${awsRegion}.amazonaws.com/v2/styles/${mapStyle}/descriptor?key=${apiKey}`;

                                    // Turkey perspective with political-view query
 parameter
                                    const turkeyViewMapStyleUrl = `https://maps.geo.
${awsRegion}.amazonaws.com/v2/styles/${mapStyle}/descriptor?political-view=TUR&key=
${apiKey}`;

                                    const internatinalViewMap = new maplibregl.Map({
                                        container: 'internatinalView', // container
 id
                                        style: internatinalViewMapStyleUrl, // style
 URL
```

```
                                    center: [33.0714561, 35.1052139], //
starting position [lng, lat]
                                    zoom: 7.5,
                                });

                                const turkeyViewMap = new maplibregl.Map({
                                    container: 'turkeyView', // container id
                                    style: turkeyViewMapStyleUrl, // style URL
                                    center: [33.0714561, 35.1052139],
                                    zoom: 7.5,
                                });

                                // Sync map zoom and center
                                syncMaps(internatinalViewMap, turkeyViewMap);

                                // Informational popup for international view
                                new maplibregl.Popup({closeOnClick: false})
                                    .setLngLat([33, 35.5])
                                    .setHTML('<h4>International view <br>
recognizes <br> Cyprus</h4>')
                                    .addTo(internatinalViewMap);

                                // Informational popup for Turkey's view
                                new maplibregl.Popup({closeOnClick: false})
                                    .setLngLat([33, 35.5])
                                    .setHTML('<h4>Turkey does not <br> recognize
<br> Cyprus</h4>')
                                    .addTo(turkeyViewMap);
                            </script>
                        </body>
                    </html>
```

style.css

```
                    body { margin: 0; padding: 0; }
                    html, body { height: 100%; }
                    #internatinalView, #turkeyView { height: 100%; width:
100%; }
                    .maps {
                        display: flex;
                        width: 100%;
```

```
                    height: 100%;
                }
```

如何在地图上过滤 POI

亚马逊定位服务允许您选择与您的用例相关的 POI (point-of-interest) 类别。了解有关标准地图风格 Rich POI 的更多信息

筛选 POI

在此示例中，您将显示一个交互式地图，该地图允许用户按兴趣点类别进行筛选。

Index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>POI demo</title>
    <meta property="og:description" content="" />
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <link rel="stylesheet"   href="https://unpkg.com/maplibre-gl@4.5.0/dist/
maplibre-gl.css" />
    <link rel='stylesheet' href='style.css' />
    <script src="https://unpkg.com/maplibre-gl@4.5.0/dist/maplibre-gl.js"></script>

  </head>
  <body>

    <div id="map"></div>
    <ul id="buttons"></ul>
    <script>
      //
 ----------------------------------------------------------------------------------------
     // Function to set layer state
     //
 ----------------------------------------------------------------------------------------

     const initialLayerState = {
       Buildings: {
         layers: ['poi_900_areas_buildings'],
         visibility: 'visible'
```

```
      },
      'Business & Services': {
        layers: ['poi_700_business_services'],
        visibility: 'visible'
      },
      Shopping: { layers: ['poi_600_shopping'], visibility: 'visible' },
      'Leisure & Outdoors': {
        layers: ['poi_550_leisure_outdoor'],
        visibility: 'visible'
      },
      Facilities: { layers: ['poi_800_facilities'], visibility: 'visible' },
      Transit: { layers: ['poi_400_transit'], visibility: 'visible' },
      'Sights & Museums': {
        layers: ['poi_300_sights_museums'],
        visibility: 'visible'
      },
      'Food & Drink': {
        layers: ['poi_100_food_drink'],
        visibility: 'visible'
      },
      'Going Out & Entertainment': {
        layers: ['poi_200_going_out_entertainment'],
        visibility: 'visible'
      },
      Accommodations: {
        layers: ['poi_500_accommodations'],
        visibility: 'visible'
      },
      Parks: { layers: ['poi_landuse_park'], visibility: 'visible' },
      'Public Complexes': {
        layers: ['poi_landuse_public_complex'],
        visibility: 'visible'
      }
    };

    //
-------------------------------------------------------------------------------------------
    // Setup auth and state
    //
-------------------------------------------------------------------------------------------

    let state = { ...initialLayerState };
```

```
        const API_KEY = "<api key>"; // check how to create api key for Amazon
 Location
        const AWS_REGION = "eu-central-1"; // Replace with your AWS region

        const mapStyle = "Standard";
        const colorScheme = "Light";


      // Style URL
      const styleUrl = `https://maps.geo.${AWS_REGION}.amazonaws.com/v2/styles/
${mapStyle}/descriptor?key=${API_KEY}&color-scheme=${colorScheme}`;


      // Render the map with the stylesheet
      map = new maplibregl.Map({
        container: 'map',
        style: styleUrl,
        center: [-73.966148, 40.781803],
        zoom: 17
      });

      const buttonContainer = document.getElementById('buttons');

      for (const category of Object.keys(state)) {
        const newButton = document.createElement('li');
        newButton.classList.add('button');
        newButton.classList.add('active');
        newButton.id = category;

        // Each config has a label
        newButton.textContent = category;

        // On click, we want to switch the value between 'on' and 'off'
        // We could check the config for available options in more complex cases
        newButton.onclick = () => {
          onClickCategory(category);
        };

        // Append the button to our button container
        buttonContainer.appendChild(newButton);
      }

      //
 ----------------------------------------------------------------------------------------
```

```
      // LAYER VISIBILITY FUNCTION
      //
 ----------------------------------------------------------------------------------------------

      // On click, get the stylesheet, update the language, and set the style
      const onClickCategory = category => {
        const categoryState = state[category];
        const { visibility, layers } = categoryState;

        let nextState;

        // For active button styling
        const activeButton = document.getElementById(category);

        if (visibility === 'visible') {
          nextState = 'none';
          activeButton.classList.remove('active');
        } else {
          nextState = 'visible';
          activeButton.classList.add('active');
        }

        layers.forEach(id =>
          map.setLayoutProperty(id, 'visibility', nextState)
        );

        state[category].visibility = nextState;
      };
    </script>
  </body>
</html>
```

style.css

```
    body {
        margin: 0;
        padding: 0;
      }
      html,
      body,
      #map {
        height: 100%;
```

```
}


#buttons {
  display: flex;
  flex-direction: column;
  justify-content: space-between;
  position: absolute;
  top: 0;
  min-width: 100px;
  max-height: calc(100% - 2rem);
  overflow: auto;
  padding: 1rem;
  background-color: white;
  margin: 0;
}

.button {
  display: inline-block;
  cursor: pointer;
  padding: 8px;
  border-radius: 3px;
  margin-top: 10px;
  font-size: 12px;
  text-align: center;
  color: #fff;
  background: #8a8a8a;
  font-family: sans-serif;
  font-weight: bold;
}

.button:first-child {
  margin-top: 0;
}

.active {
  background: #ee8a65;
}
```

如何使用静态地图

这些操作主题为利用 Amazon Location Service 的地图功能自定义带有叠加层和视觉元素的静态地图提供了 step-by-step指导。每份指南都会引导您完成关键任务，例如调整地图尺寸、在缩放或半径之间进行选择，以及添加各种地理空间元素，例如标记、多边形、线条和邻近圆圈。这些指南使用紧凑格式和 GeoJSON 格式，以确保样式和配置的灵活性。

无论您是在为面向客户的应用程序优化地图，还是进行地理空间分析，这些主题都提供了有关如何根据特定要求定制静态地图的明确说明。通过详细的示例，您将学习如何增强地图显示效果并有效地管理满足各种业务需求的地理数据。

主题

- 如何获取特定位置的静态地图
- 如何获取特定维度的静态地图
- 如何在静态地图的半径和缩放之间做出选择
- 如何为静态地图添加比例
- 如何向静态地图添加标记
- 如何向静态地图添加线条
- 如何向静态地图添加路由
- 如何向静态地图添加多边形
- 如何为静态地图设置语言
- 如何在静态地图上添加近距离圈

如何获取特定位置的静态地图

在本主题中，您将学习如何根据特定参数从 Amazon Location Service 检索静态地图。您将学习如何为中心位置、边界框和一组有界位置生成静态地图。提供的示例将帮助您自定义地图的宽度、高度和样式。

> **Note**
>
> 生成静态地图`map@2x`时必须通过`map`或。

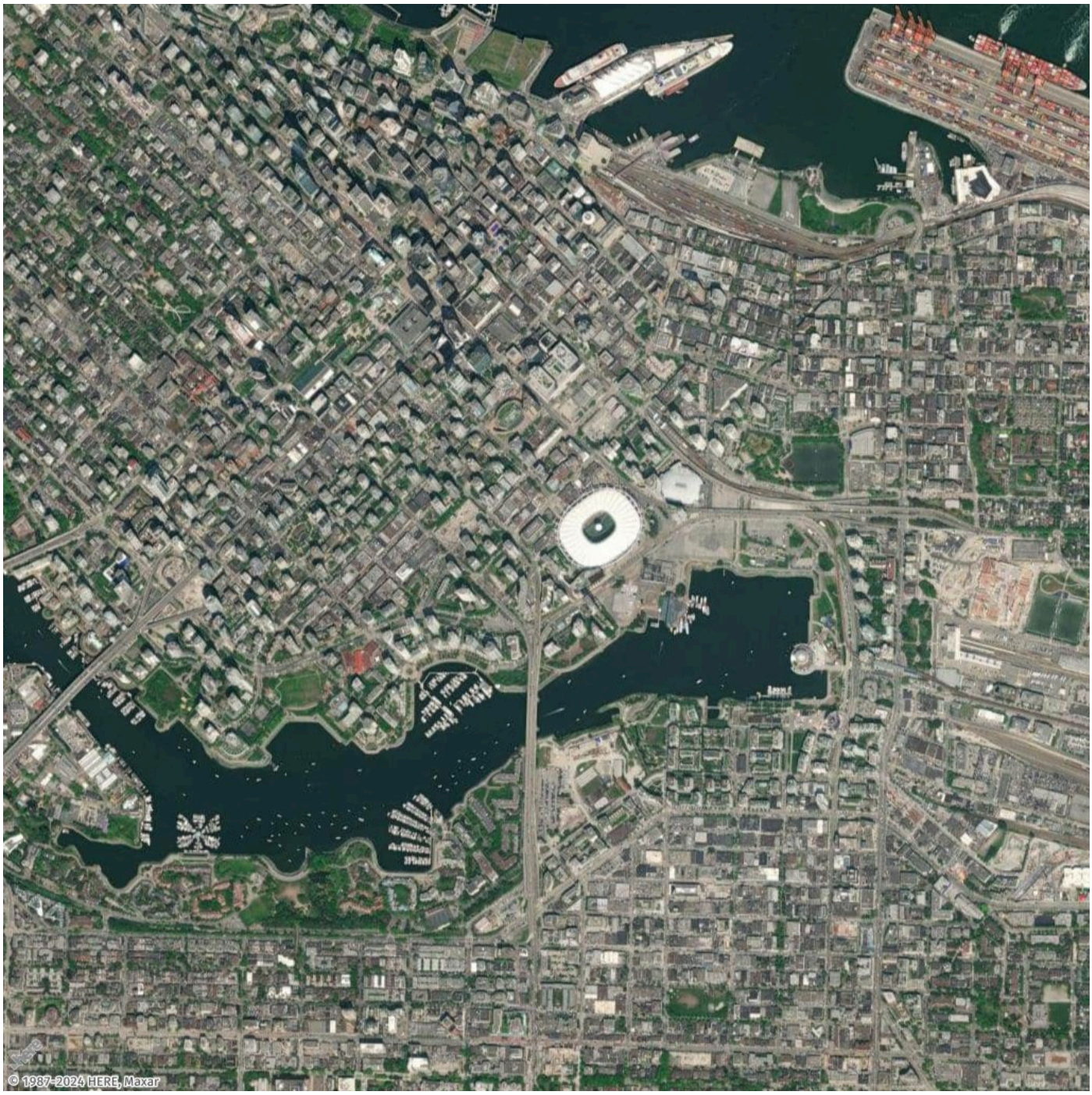
获取中心位置的地图图像

在本示例中，您将创建一个宽度为、高度为的地1024图图像，中心坐标设置为-123.1143,49.2763、位置longitude=-123.1143和latitude=49.2763，缩放级别为15。1024

Request URL

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/map?
style=Satellite&width=1024&height=1024&zoom=15&center=-123.1156126,49.2767046&key=API_KEY
```

Response image



获取边界框的地图图像

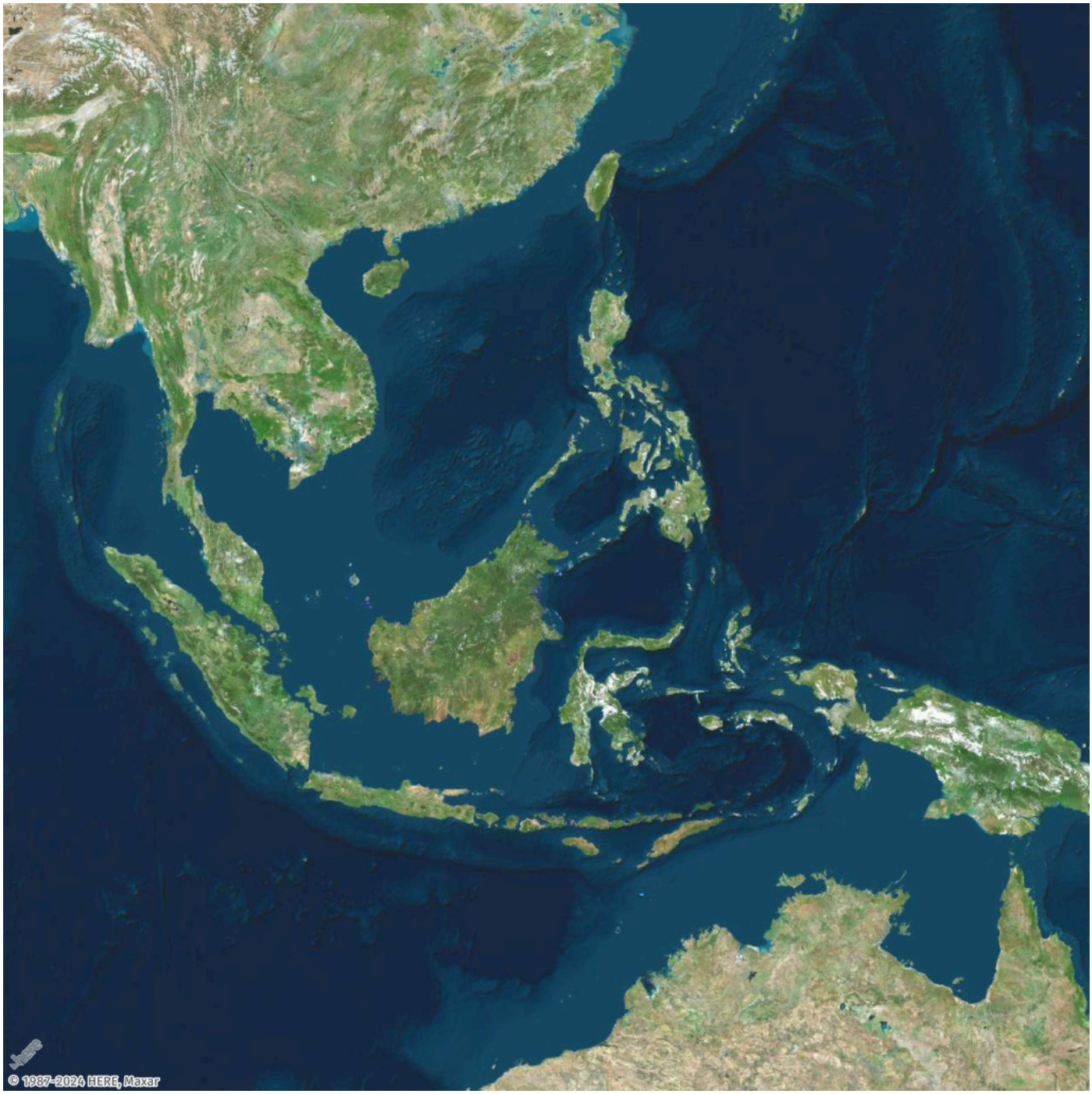
在本示例中，您将通过设置该区域的边界框来生成东南亚的地图图像。

bbox格式为{southwest_longitude},{southwest_latitude},{northeast_longitude},{northeast_latitude}。

Request URL

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/
map?style=Satellite&width=1024&height=1024&bounding-
box=90.00,-21.94,146.25,31.95&key=API_KEY
```

Response image



获取限定位置的地图图像

在此示例中，您将生成一张覆盖巴黎多个必看景点的地图，每个景点都由其坐标（经度、纬度）指定。边界位置包括：埃菲尔铁塔、卢浮宫博物馆、巴黎圣母院、香榭丽舍大街、凯旋门、圣心大教堂、卢森堡花园、奥赛博物馆、协和广场和加尼尔宫。

边界位置的格式为。`{longitude1},{latitude1},{longitude2},{latitude2} ... {longitudeN},{latitudeN}`

Request URL

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/map?
style=Satellite&width=1024&height=1024&bounded-
positions=2.2945,48.8584,2.3376,48.8606,2.3500,48.8529,2.3076,48.8698,2.2950,48.8738,2.3431,
```

Response image



如何获取特定维度的静态地图

在本主题中，您将学习如何使用 Amazon Location Service 为静态地图设置尺寸（高度和宽度）。自定义地图图像的尺寸允许您在性能、视觉质量和可用性之间取得平衡。`width`和`height`的最大值均为 1400 像素，而最小值为 64 像素。最大结果大小为 6 MB。

此外，您还可以使用bbox和bounds参数padding来确保边缘附近的重要地图要素完全可见且不会被切断。

获取具有特定高度和宽度的地图图像

在本示例中，您将创建芬兰赫尔辛基的低分辨率和中等分辨率的地图图像。

Request URL for low-resolution thumbnail

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/map?
style=Satellite&width=200&height=200&zoom=11.5&center=24.9460,60.1690&key=API_KEY
```

Response (Thumbnail 200x200)



Request URL for mid-resolution image

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/map?
style=Satellite&width=700&height=700&zoom=13&center=24.9460,60.1690&key=API_KEY
```

Response image (700x700)



获取四面都有填充的地图图像

在本示例中，您将使用芬兰赫尔辛基的几个必看地点生成地图，其坐标（经度、纬度），包括带填充和不带填充。

Request URL with padding

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/map?
style=Satellite&width=1024&height=1024&padding=150&bounded-
positions=24.9526,60.1692,24.9850,60.1465,24.9270,60.1725,24.9226,60.1826,24.9509,60.1675,24
```

Response image (with padding)



Request URL without padding

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/map?
style=Satellite&width=1024&height=1024&bounded-
positions=24.9526,60.1692,24.9850,60.1465,24.9270,60.1725,24.9226,60.1826,24.9509,60.1675,24
```

Response image (without padding)



如何在静态地图的半径和缩放之间做出选择

在本主题中，您将学习如何在使用`radius`或`zoom`何时使用 Amazon Location Service 生成静态地图之间做出选择。该`radius`参数可以更精确地控制覆盖区域，因此非常适合面向客户的应用程序，因为您知道确切的覆盖区域。当您想要调整显示的细节级别时，该`zoom`参数更适合用于地理空间分析。

有半径

在本示例中，您将使用`radius`带有中心位置的参数创建斯里兰卡的地图图像。

Request URL

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/map?
style=Satellite&width=700&height=700&center=80.60596,7.76671&radius=235000&scale-
unit=KilometersMiles&key=API_KEY
```

Response image



带变焦功能

在本示例中，您将使用zoom带有中心位置的参数创建斯里兰卡的地图图像。

Request URL

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/map?
style=Satellite&width=700&height=700&zoom=8&center=80.60596,7.76671&scale-
unit=KilometersMiles&key=API_KEY
```

Response image



如何为静态地图添加比例

在本主题中，您将学习如何在 Amazon Location Service 生成的静态地图的右下角显示比例。比例可以显示单个单位，例如`Miles`或`Kilometers`，也可以同时显示两个单位，例如`KilometersMiles`或`MilesKilometers`，其中一个单位显示在顶部，另一个单位显示在底部。

使用单个单位添加比例

在本示例中，您将显示芬兰赫尔辛基的地图，其比例设置`Kilometers`为在右下角。

Request URL

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/map?
style=Satellite&width=1024&height=1024&zoom=13.5&center=24.9189564,60.1645772&scale-
unit=Kilometers&key=API_KEY
```

Response image



用两个单位添加比例

在本示例中，您将显示芬兰赫尔辛基地图，其中Kilometers两者兼而有之，Miles并按比例显示在右下角，其中一个单位显示在另一个单位之上。

Request URL

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/map?
style=Satellite&width=1024&height=1024&zoom=14&center=24.9189564,60.1645772&scale-
unit=KilometersMiles&key=API_KEY
```

Response image



如何向静态地图添加标记

在本主题中，您将学习如何向使用 Amazon Location Service 生成的静态地图添加标记。您可以自定义标记的图标、标签、大小、颜色和其他样式选项。这允许您使用与地图目的一致的视觉线索突出显示特定位置。

标记样式

名称	类型	默认值	描述
style	字符串	marker	点几何体的样式。要创建标记，请将该值设置为 GeoJSO style N 的属性对象marker或不包含该属性。
icon	字符串	pin	标记图标类型。可用值包括：cp（近似圈）、diamond、pin、poi、
label	字符串	<empty>	要在坐标处显示的文本。要设置自动标签，请使用以下值：$alpha（拉丁字母）或$num（阿拉伯数字）。
color	颜色	视风格而定	图标颜色。
outline-color	颜色	视风格而定	图标轮廓颜色。
text-color	颜色	视风格而定	标签文字颜色。
text-outline-color	颜色	视风格而定	文本轮廓颜色。
outline-width	整数	0（已关闭）	文本轮廓宽度。

名称	类型	默认值	描述
size	字符串	medium	标签和图标大小。可用值：small、medium、large

向地图图像添加标记

在本示例中，您将在温哥华不列颠哥伦比亚广场（BC Place）的地图图像上放置一个带有标签的标记。

Geo JSON:

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [
          -123.11210,
          49.2767875
        ]
        },
      "properties": {
          "color":"#EE4B2B",
          "size":"large",
          "label":"BC Place, Vancouver",
          "text-color":"#0000FF"
       }
    }
  ]
}
```

Compact:

```
point: -123.11210,49.27678;
label=BC Place, Vancouver;
size=large;
text-color=#0000FF;
```

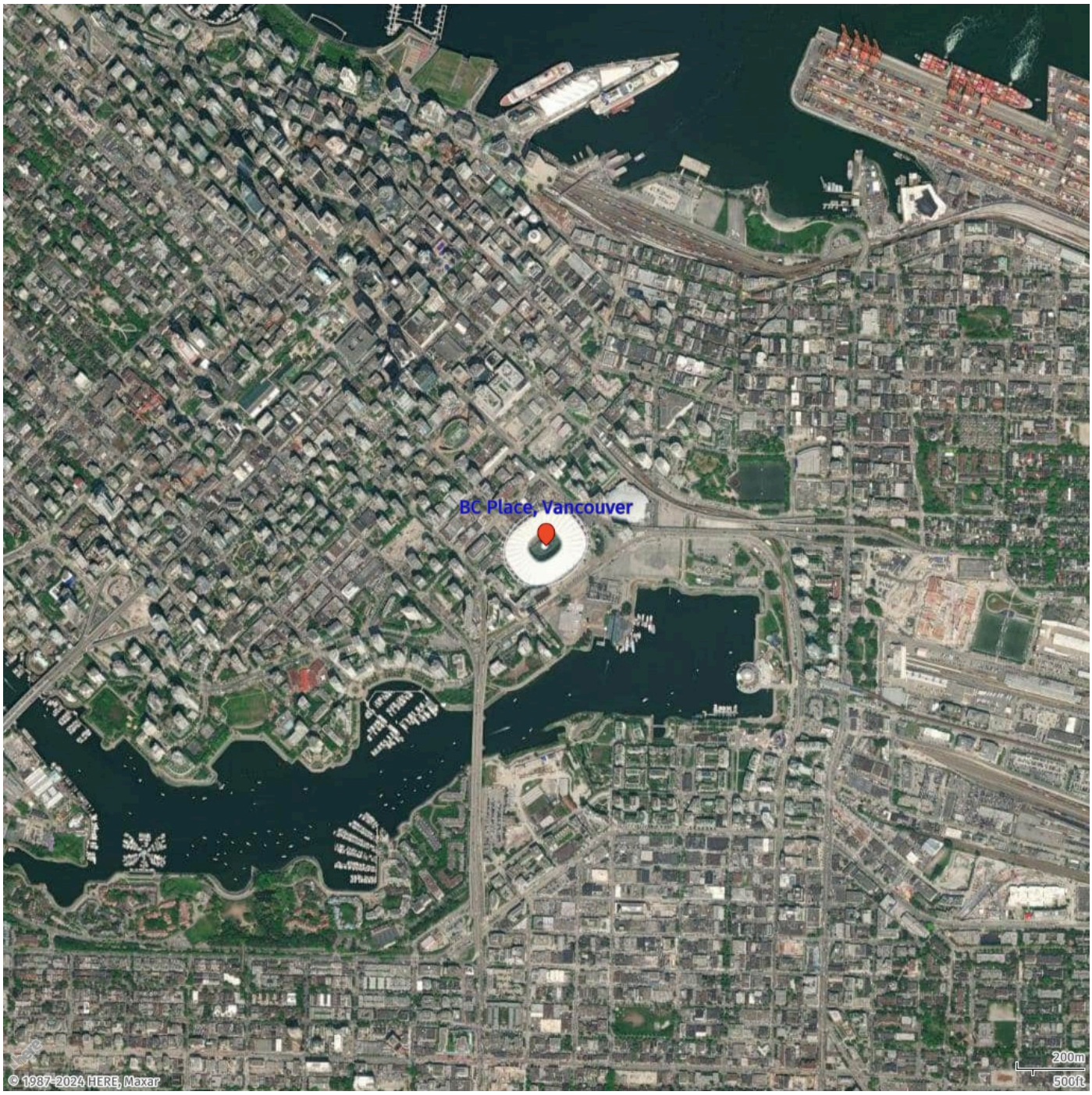
```
color=#EE4B2B
```

Request URL

使用 GeoJSON 叠加层

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/map?
style=Satellite&width=1024&height=1024&zoom=15&scale-unit=KilometersMiles&geojson-
overlay=%7B%22type%22%3A%22FeatureCollection%22,%22features%22%3A%5B%7B
%22type%22%3A%22Feature%22,%22properties%22%3A%7B%22color%22%3A%22%23EE4B2B
%22,%22size%22%3A%22large%22,%22label%22%3A%22BC%20Place,%20Vancouver%22,
%22text-color%22%3A%22%230000FF%22%7D,%22geometry%22%3A%7B%22coordinates%22%3A
%5B-123.11210974152168,49.27678753813112%5D,%22type%22%3A%22Point%22%7D%7D%5D
%7D&key=API_KEY
```

Response image



向地图图像添加多个标记

在此示例中，您将使用其坐标（经度、纬度）为芬兰赫尔辛基的必看景点添加标记。您也可以应用填充来确保地图正确容纳所有标记。

> **Note**
>
> 可用的标记图标类型包括：cp用于近距离圆圈，diamond, pin, poi, square, triangle, bubble。

Geo JSON:

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "MultiPoint",
        "coordinates": [
          [24.9526, 60.1692],
          [24.9270, 60.1725],
          [24.9226, 60.1826],
          [24.9509, 60.1675],
          [24.9566, 60.1685],
          [24.9457, 60.1674],
          [24.9397, 60.1719],
          [24.9414, 60.1715],
          [24.9387, 60.1720]
        ]
      },
      "properties": {
        "icon":"diamond",
        "label":"$num",
        "size":"large",
        "text-color":"%23972E2B",
        "text-outline-color":"%23FFF",
        "text-outline-width":2
      }
    }
  ]
}
```

Request URL

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/map?
style=Satellite&width=1024&height=1024&padding=150&geojson-overlay=%7B
%22type%22%3A%22FeatureCollection%22,%22features%22%3A%5B%7B%22type%22%3A
%22Feature%22,%22geometry%22%3A%7B%22type%22%3A%22MultiPoint%22,%22coordinates
%22%3A%5B%5B24.9526,60.1692%5D,%5B24.927,60.1725%5D,%5B24.9226,60.1826%5D,
%5B24.9509,60.1675%5D,%5B24.9566,60.1685%5D,%5B24.9457,60.1674%5D,
%5B24.9397,60.1719%5D,%5B24.9414,60.1715%5D,%5B24.9387,60.172%5D%5D%7D,%22properties
%22%3A%7B%22icon%22%3A%22diamond%22,%22label%22%3A%22%24num%22,%22size%22%3A%22large
%22,%22text-color%22%3A%22%23972E2B%22,%22text-outline-color%22%3A%22%23FFF%22,
%22text-outline-width%22%3A2%7D%7D%5D%7D&key=API_KEY
```

Response image



更改地图图像中标记的颜色

在此示例中，您将使用不同颜色的气泡标记来表示世界各地的城市。您可以自定义标记的颜色、标签和其他属性以适应地图的上下文。

Geo JSON:

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [85.1376, 25.5941]
      },
      "properties": {
        "label": "Patna",
        "icon": "bubble",
        "color": "#FF5722",
        "outline-color": "#D74D3D",
        "text-color": "#FFFFFF"
      }
    },
    .....redacted.....
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [2.3522, 48.8566]
      },
      "properties": {
        "label": "Paris, France",
        "icon": "bubble",
        "color": "#FF9800",
        "outline-color": "#D76B0B",
        "text-color": "#FFFFFF"
      }
    }
  ]
}
```

Request URL

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/map?
style=Satellite&width=1400&height=1024&padding=150&geojson-overlay=%7B
%22type%22%3A%22FeatureCollection%22,%22features%22%3A%5B%7B%22type%22%3A
%22Feature%22,%22geometry%22%3A%7B%22type%22%3A%22Point%22,%22coordinates
%22%3A%5B85.1376,25.5941%5D%7D,%22properties%22%3A%7B%22label%22%3A%22Patna
%22,%22icon%22%3A%22bubble%22,%22color%22%3A%22%23FF5722%22,%22outline-color
```

```
%22%3A%22%23D74D3D%22,%22text-color%22%3A%22%23FFFFFF%22%7D%7D,%7B%22type%22%3A
%22Feature%22,%22geometry%22%3A%7B%22type%22%3A%22Point%22,%22coordinates%22%3A
%5B105.8542,21.0285%5D%7D,%22properties%22%3A%7B%22label%22%3A%22Hanoi,%20Vietnam
%22,%22icon%22%3A%22bubble%22,%22color%22%3A%22%232196F3%22,%22outline-color
%22%3A%22%231A78C2%22,%22text-color%22%3A%22%23FFFFFF%22%7D%7D,%7B%22type%22%3A
%22Feature%22,%22geometry%22%3A%7B%22type%22%3A%22Point%22,%22coordinates%22%3A
%5B116.4074,39.9042%5D%7D,%22properties%22%3A%7B%22label%22%3A%22Beijing,%20China
%22,%22icon%22%3A%22bubble%22,%22color%22%3A%22%23FF9800%22,%22outline-color
%22%3A%22%23D76B0B%22,%22text-color%22%3A%22%23FFFFFF%22%7D%7D,%7B%22type%22%3A
%22Feature%22,%22geometry%22%3A%7B%22type%22%3A%22Point%22,%22coordinates%22%3A
%5B106.9101,47.918%5D%7D,%22properties%22%3A%7B%22label%22%3A%22Ulaanbaatar,
%20Mongolia%22,%22icon%22%3A%22bubble%22,%22color%22%3A%22%239C27B0%22,%22outline-
color%22%3A%22%237B1FA2%22,%22text-color%22%3A%22%23FFFFFF%22%7D%7D,%7B%22type
%22%3A%22Feature%22,%22geometry%22%3A%7B%22type%22%3A%22Point%22,%22coordinates
%22%3A%5B151.2093,-33.8688%5D%7D,%22properties%22%3A%7B%22label%22%3A%22Sydney,
%20Australia%22,%22icon%22%3A%22bubble%22,%22color%22%3A%22%234CAF50%22,%22outline-
color%22%3A%22%23388E3C%22,%22text-color%22%3A%22%23FFFFFF%22%7D%7D,%7B%22type
%22%3A%22Feature%22,%22geometry%22%3A%7B%22type%22%3A%22Point%22,%22coordinates
%22%3A%5B174.7633,-41.2865%5D%7D,%22properties%22%3A%7B%22label%22%3A%22Wellington,
%20New%20Zealand%22,%22icon%22%3A%22bubble%22,%22color%22%3A%22%23FFC107%22,
%22outline-color%22%3A%22%23FFA000%22,%22text-color%22%3A%22%23000000%22%7D%7D%5D
%7D&key=API_KEY
```

Response image



如何向静态地图添加线条

在本主题中，您将学习如何使用 Amazon Location Service 向静态地图添加一条线。我们将介绍支持的样式选项、如何使用 GeoJSON 定义线条以及如何应用颜色、宽度和轮廓等自定义样式。我们还将探讨如何对线宽等属性使用不同的测量单位。

支持的样式

名称	类型	默认值	描述
`color`	颜色	视风格而定	线条颜色。
`width`	整数/字符串	2	线宽。有关更多信息，请参阅 the section

名称	类型	默认值	描述
			called “向静态地图添加一条线”。
outline-color	颜色	视风格而定	线条轮廓颜色。
outline-width	整数/字符串	0（已关闭）	轮廓的宽度。有关更多信息，请参阅 the section called “向静态地图添加一条线”。

支持的单位

单位	描述
例如，整数 5	像素
例如，没有单位的字符串 "5"	像素
"px"	像素
"m"	米
"km"	Kilometers
"mi"	迈尔斯
"ft"	脚
"yd"	码数
"%"	特定属性的默认值的百分比（以像素为单位）。例如，如果的默认值为width2像素，200%则为4像素。 %是一个敏感字符，必须在请求 URL 中将其编码为%25。

添加一行

在此示例中，您将添加一条从温哥华某个地方到斯坦利公园的线。这条线是使用具有定义坐标的 GeoJSON 格式创建的。

Request

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [-123.11813, 49.28246],
          [-123.11967, 49.28347],
          [-123.12108, 49.28439],
          [-123.12216, 49.28507],
          [-123.12688, 49.28812],
          [-123.1292, 49.28964],
          [-123.13216, 49.2916],
          [-123.13424, 49.29291],
          [-123.13649, 49.2944],
          [-123.13678, 49.29477],
          [-123.13649, 49.29569],
          [-123.13657, 49.29649],
          [-123.13701, 49.29715],
          [-123.13584, 49.29847],
          [-123.13579, 49.29935],
          [-123.13576, 49.30018],
          [-123.13574, 49.30097]
        ]
      },
      "properties": {
        "name": "To Stanley Park",
        "description": "An evening walk to Stanley Park."
      }
    }
  ]
}
```

Request URL

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/map?
style=Satellite&width=1024&height=1024&padding=200&scale-
unit=KilometersMiles&geojson-overlay=%7B%22type%22%3A%22FeatureCollection
%22,%22features%22%3A%5B%7B%22type%22%3A%22Feature%22,%22geometry%22%3A%7B
%22type%22%3A%22LineString%22,%22coordinates%22%3A%5B%5B-123.11813,49.28246%5D,
%5B-123.11967,49.28347%5D,%5B-123.12108,49.28439%5D,%5B-123.12216,49.28507%5D,
%5B-123.12688,49.28812%5D,%5B-123.1292,49.28964%5D,%5B-123.13216,49.2916%5D,
%5B-123.13424,49.29291%5D,%5B-123.13649,49.2944%5D,%5B-123.13678,49.29477%5D,
%5B-123.13649,49.29569%5D,%5B-123.13657,49.29649%5D,%5B-123.13701,49.29715%5D,
%5B-123.13584,49.29847%5D,%5B-123.13579,49.29935%5D,%5B-123.13576,49.30018%5D,
%5B-123.13574,49.30097%5D%5D%7D,%22properties%22%3A%7B%22name%22%3A%22To%20Stanley
%20Park%22,%22description%22%3A%22An%20evening%20walk%20to%20Stanley%20Park.%22%7D
%7D%5D%7D&key=API_KEY
```

Response image



为线条增添造型

在此示例中，您将为上一个示例中添加的线条设置样式。这包括定义线条的颜色、宽度、轮廓颜色和轮廓宽度，以自定义地图上线条的视觉外观。

Request

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [-123.11813, 49.28246],
          [-123.11967, 49.28347],
          [-123.12108, 49.28439],
          [-123.12216, 49.28507],
          [-123.12688, 49.28812],
          [-123.1292, 49.28964],
          [-123.13216, 49.2916],
          [-123.13424, 49.29291],
          [-123.13649, 49.2944],
          [-123.13678, 49.29477],
          [-123.13649, 49.29569],
          [-123.13657, 49.29649],
          [-123.13701, 49.29715],
          [-123.13584, 49.29847],
          [-123.13579, 49.29935],
          [-123.13576, 49.30018],
          [-123.13574, 49.30097]
        ]
      },
      "properties": {
        "color": "#6d34b3",
        "width": "9m",
        "outline-color": "#a8b9cc",
        "outline-width": "2px"
      }
    }
  ]
}
```

Request URL

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/map?
style=Satellite&width=1024&height=1024&padding=200&scale-
```

```
unit=KilometersMiles&geojson-overlay=%7B%22type%22%3A%22FeatureCollection%22,
%22features%22%3A%5B%7B%22type%22%3A%22Feature%22,%22geometry%22%3A%22LineString
%22,%22coordinates%22%3A%5B%5B-123.11813,49.28246%5D,%5B-123.11967,49.28347%5D,
%5B-123.12108,49.28439%5D,%5B-123.12216,49.28507%5D,%5B-123.12688,49.28812%5D,
%5B-123.1292,49.28964%5D,%5B-123.13216,49.2916%5D,%5B-123.13424,49.29291%5D,
%5B-123.13649,49.2944%5D,%5B-123.13678,49.29477%5D,%5B-123.13649,49.29569%5D,
%5B-123.13657,49.29649%5D,%5B-123.13701,49.29715%5D,%5B-123.13584,49.29847%5D,
%5B-123.13579,49.29935%5D,%5B-123.13576,49.30018%5D,%5B-123.13574,49.30097%5D%5D
%7D,%22properties%22%3A%7B%22color%22%3A%22%236d34b3%22,%22width%22%3A%229m%22,
%22outline-color%22%3A%22%23a8b9cc%22,%22outline-width%22%3A%222px%22%7D%7D%5D
%7D&key=API_KEY
```

Response image



如何向静态地图添加路由

在本主题中，您将学习如何使用 Amazon Location Service 向静态地图添加路线。您将完成使用 CalculateRoutes API 获取路径的步骤，然后使用 GeoJSON 以及点和线的自定义样式在静态地图上对其进行可视化。

添加路径的步骤

1. 从 CalculateRoutes API 获取路线。移除落在同一条直线上的坐标以优化 LineString，从而防止查询字符串达到其极限。
2. 根据经过优化的坐标集创建 GeoJSON 对象。
3. 取第一个和最后一个点， LineString 然后添加点几何图形来标记起点和终点位置。
4. LineString 根据您的业务需求设置要点和的样式，调整颜色、大小和标签等属性。

使用紧凑风格添加路线

在本示例中，您将向中创建的线添加一条包含起点和终点的路径the section called “向静态地图添加一条线”。路径将使用自定义样式进行定义，包括起点和终点的颜色、大小和标签。

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [
          [-123.11813, 49.28246],
          [-123.11967, 49.28347],
          [-123.12108, 49.28439],
          [-123.12216, 49.28507],
          [-123.12688, 49.28812],
          [-123.1292, 49.28964],
          [-123.13216, 49.2916],
          [-123.13424, 49.29291],
          [-123.13649, 49.2944],
          [-123.13678, 49.29477],
          [-123.13649, 49.29569],
          [-123.13657, 49.29649],
          [-123.13701, 49.29715],
          [-123.13584, 49.29847],
          [-123.13579, 49.29935],
          [-123.13576, 49.30018],
          [-123.13574, 49.30097]
        ]
      },
      "properties": {
```

```
        "color": "#000000",
        "width": "20m",
        "outline-color": "#a8b9cc",
        "outline-width": "2px"
      }
    },
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [-123.11813, 49.28246]
      },
      "properties": {
        "label": "Pacific Centre",
        "icon": "bubble",
        "size": "large",
        "color": "#34B3A4",
        "outline-color": "#006400",
        "text-color": "#FFFFFF"
      }
    },
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [-123.13574, 49.30097]
      },
      "properties": {
        "label": "Stanley Park",
        "icon": "bubble",
        "size": "large",
        "color": "#B3346D",
        "outline-color": "#FF0000",
        "text-color": "#FFFFFF"
      }
    }
  ]
}
```

Request URL

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/map?
style=Satellite&width=1024&height=1024&padding=200&scale-
```

```
unit=KilometersMiles&geojson-overlay=%7B%22type%22%3A%22FeatureCollection
%22,%22features%22%3A%5B%7B%22type%22%3A%22Feature%22,%22geometry%22%3A%7B
%22type%22%3A%22LineString%22,%22coordinates%22%3A%5B%5B-123.11813,49.28246%5D,
%5B-123.11967,49.28347%5D,%5B-123.12108,49.28439%5D,%5B-123.12216,49.28507%5D,
%5B-123.12688,49.28812%5D,%5B-123.1292,49.28964%5D,%5B-123.13216,49.2916%5D,
%5B-123.13424,49.29291%5D,%5B-123.13649,49.2944%5D,%5B-123.13678,49.29477%5D,
%5B-123.13649,49.29569%5D,%5B-123.13657,49.29649%5D,%5B-123.13701,49.29715%5D,
%5B-123.13584,49.29847%5D,%5B-123.13579,49.29935%5D,%5B-123.13576,49.30018%5D,
%5B-123.13574,49.30097%5D%5D%7D,%22properties%22%3A%7B%22color%22%3A%22%23000000%22,
%22width%22%3A%2220m%22,%22outline-color%22%3A%22%23a8b9cc%22,%22outline-width
%22%3A%222px%22%7D%7D,%7B%22type%22%3A%22Feature%22,%22geometry%22%3A%22Point%22,
%22coordinates%22%3A%5B-123.11813,49.28246%5D%7D,%22properties%22%3A%7B%22label
%22%3A%22Pacific%20Centre%22,%22icon%22%3A%22bubble%22,%22size%22%3A%22large%22,
%22color%22%3A%22%2334B3A4%22,%22outline-color%22%3A%22%23006400%22,%22text-color
%22%3A%22%23FFFFFF%22%7D%7D,%7B%22type%22%3A%22Feature%22,%22geometry%22%3A%22Point
%22,%22coordinates%22%3A%5B-123.13574,49.30097%5D%7D,%22properties%22%3A%7B%22label
%22%3A%22Stanley%20park%22,%22icon%22%3A%22bubble%22,%22size%22%3A%22large%22,
%22color%22%3A%22%23B3346D%22,%22outline-color%22%3A%22%23FF0000%22,%22text-color
%22%3A%22%23FFFFFF%22%7D%7D%5D%7D&key=API_KEY
```

Response image



如何向静态地图添加多边形

可以通过在建筑物和位置周围指定一个多边形来突出显示建筑物和位置，例如五角大楼（位于华盛顿特区）。

添加单个多边形

在此示例中，您在地图上为五角大楼的建筑物添加了一个多边形（五角大楼）。

Geo JSON

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "coordinates": [
          [
            [
              -77.0579282196337,
              38.87264268371487
            ],
            [
              -77.05868880963534,
              38.87003145971428
            ],
            [
              -77.05560979468949,
              38.86876862025221
            ],
            [
              -77.05305311263672,
              38.87059509268525
            ],
            [
              -77.0546384387842,
              38.872985132206225
            ],
            [
              -77.0579282196337,
              38.87264268371487
            ]
          ]
        ],
        "type": "Polygon"
      }
    }
  ]
```

```
}
```

Request URL

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/map?
style=Satellite&width=600&height=1024&padding=200&scale-
unit=KilometersMiles&geojson-overlay=%7B%22type%22%3A%22FeatureCollection
%22,%22features%22%3A%5B%7B%22type%22%3A%22Feature%22,%22geometry%22%3A
%7B%22coordinates%22%3A%5B%5B%5B-77.0579282196337,38.87264268371487%5D,
%5B-77.05868880963534,38.87003145971428%5D,
%5B-77.05560979468949,38.86876862025221%5D,
%5B-77.05305311263672,38.87059509268525%5D,
%5B-77.0546384387842,38.872985132206225%5D,%5B-77.0579282196337,38.87264268371487%5D
%5D%5D,%22type%22%3A%22Polygon%22%7D%7D%5D%7D&key=your_API_Key
```

Response image



添加样式多边形

在此示例中，我们重新设计了前面示例中显示的多边形的样式。我们用不同的颜色画多边形 (#E3F70550)。颜色成分如下所示。

- E3 代表红色分量。
- F7 代表绿色分量。
- 05 代表蓝色分量。
- 50 表示 alpha（不透明度）分量。

Geo JSON

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "coordinates": [
          [
            [
              -77.0579282196337,
              38.87264268371487
            ],
            [
              -77.05868880963534,
              38.87003145971428
            ],
            [
              -77.05560979468949,
              38.86876862025221
            ],
            [
              -77.05305311263672,
              38.87059509268525
            ],
            [
              -77.0546384387842,
              38.872985132206225
            ],
            [
              -77.0579282196337,
              38.87264268371487
            ]
          ]
        ],
```

```
        "type": "Polygon"
      },
      "properties": {
        "color": "#E3F70550"
      }
    }
  ]
}
```

Request URL

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/map?
style=Satellite&width=1024&height=1024&padding=100&scale-
unit=KilometersMiles&geojson-overlay=%7B%22type%22%3A%22FeatureCollection
%22,%22features%22%3A%5B%7B%22type%22%3A%22Feature%22,%22geometry%22%3A
%7B%22coordinates%22%3A%5B%5B%5B-77.0579282196337,38.87264268371487%5D,
%5B-77.05868880963534,38.87003145971428%5D,
%5B-77.05560979468949,38.86876862025221%5D,
%5B-77.05305311263672,38.87059509268525%5D,
%5B-77.0546384387842,38.872985132206225%5D,%5B-77.0579282196337,38.87264268371487%5D
%5D%5D,%22type%22%3A%22Polygon%22%7D,%22properties%22%3A%7B%22color%22%3A
%22%23E3F70550%22%7D%7D%5D%7D&key=your_API_Key
```

Response image



添加多个多边形

在此示例中，我们添加了多个多边形，以突出显示纽约市的多个公园。

Geo JSON

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {
        "color": "#00800050"
      },
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [
            [
              -73.95824708489555,
              40.80055774655358
            ],
            [
              -73.9818875523859,
              40.76810261850716
            ],
            [
              -73.9729556303776,
              40.7642422333698
            ],
            [
              -73.94916953372382,
              40.79699323614054
            ],
            [
              -73.95824708489555,
              40.80055774655358
            ]
          ]
        ]
      }
    },
    {
      "type": "Feature",
      "properties": {
        "color": "#FF000050"
      },
      "geometry": {
```

```
        "type": "Polygon",
        "coordinates": [
          [
            [
              -73.94432602794981,
              40.80634757577718
            ],
            [
              -73.94607200977896,
              40.803869579741644
            ],
            [
              -73.94301654157768,
              40.80263972513214
            ],
            [
              -73.94127055974795,
              40.805099411561145
            ],
            [
              -73.94432602794981,
              40.80634757577718
            ]
          ]
        ]
      }
    },
    {
      "type": "Feature",
      "properties": {
        "color": "#0000FF50"
      },
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [
            [
              -73.9947948382843,
              40.7691390468232
            ],
            [
              -73.99564708262241,
              40.76802192177411
            ],
```

```
            [
              -73.99372953286147,
              40.76723992306512
            ],
            [
              -73.99289367783732,
              40.76835706126087
            ],
            [
              -73.9947948382843,
              40.7691390468232
            ]
          ]
        ]
      }
    }
  ]
}
```

Request URL

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/map?
style=Satellite&width=1024&height=1024&padding=100&scale-
unit=KilometersMiles&geojson-overlay=%7B%22type%22%3A%22FeatureCollection
%22,%22features%22%3A%5B%7B%22type%22%3A%22Feature%22,%22properties%22%3A%7B
%22color%22%3A%22%2300800050%22%7D,%22geometry%22%3A%7B%22type%22%3A%22Polygon
%22,%22coordinates%22%3A%5B%5B%5B-73.95824708489555,40.80055774655358%5D,
%5B-73.9818875523859,40.76810261850716%5D,%5B-73.9729556303776,40.7642422333698%5D,
%5B-73.94916953372382,40.79699323614054%5D,
%5B-73.95824708489555,40.80055774655358%5D%5D%5D%7D%7D,%7B
%22type%22%3A%22Feature%22,%22properties%22%3A%7B%22color%22%3A
%22%23FF000050%22%7D,%22geometry%22%3A%7B%22type%22%3A%22Polygon%22,
%22coordinates%22%3A%5B%5B%5B-73.94432602794981,40.80634757577718%5D,
%5B-73.94607200977896,40.803869579741644%5D,
%5B-73.94301654157768,40.80263972513214%5D,
%5B-73.94127055974795,40.805099411561145%5D,
%5B-73.94432602794981,40.80634757577718%5D%5D%5D%7D%7D,%7B%22type%22%3A
%22Feature%22,%22properties%22%3A%7B%22color%22%3A%22%230000FF50%22%7D,
%22geometry%22%3A%7B%22type%22%3A%22Polygon%22,%22coordinates%22%3A%5B%5B
%5B-73.9947948382843,40.7691390468232%5D,%5B-73.99564708262241,40.76802192177411%5D,
%5B-73.99372953286147,40.76723992306512%5D,
%5B-73.99289367783732,40.76835706126087%5D,%5B-73.9947948382843,40.7691390468232%5D
%5D%5D%7D%7D%5D%7D&key=your_API_Key
```

Response image



如何为静态地图设置语言

如果您不想使用默认语言，则可以为静态地图设置语言。

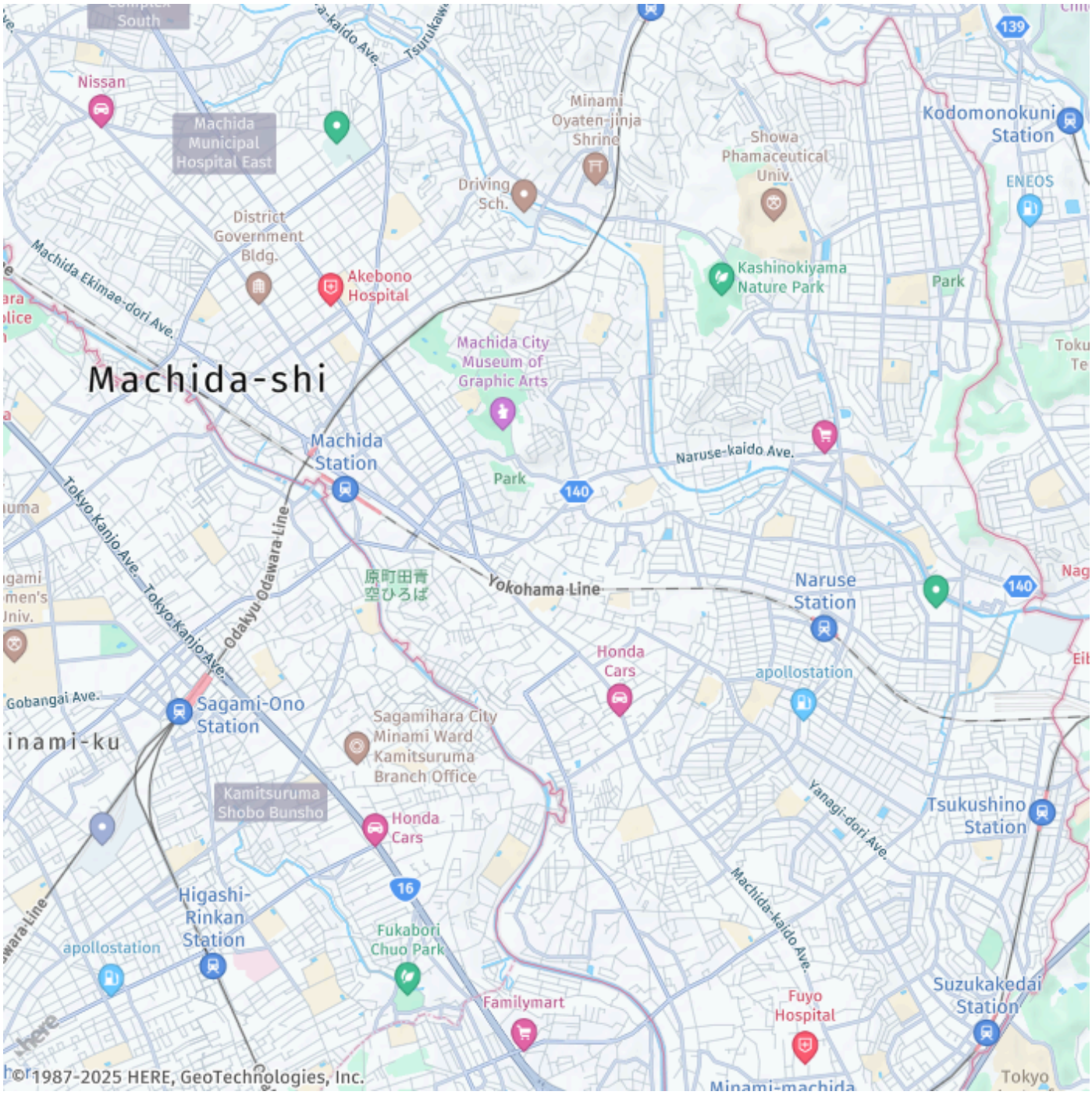
在此示例中，通常默认使用日语文本的东京地点请求的是英语，覆盖了区域默认值。

该示例显示了提供语言的请求的响应图像，以及未提供语言的请求的响应图像。相应的标签反映了显示所请求语言的地图和显示默认语言的地图之间的区别。

Request URL - English set as the language

```
https://maps.geo.us-east-1.amazonaws.com/v2/static/map?
center=139.4575,35.539&style=Standard&lang=en&height=700&width=700&zoom=14
```

Response image



Request URL - default language

```
https://maps.geo.us-east-1.amazonaws.com/v2/static/map?
center=139.4575,35.539&style=Standard&height=700&width=700&zoom=14
```

Response image



如何在静态地图上添加近距离圈

近距离圈子为报告或估计的设备或用户位置的准确性提供了重要的见解。通过在报告点周围显示一个圆圈，用户会被告知，由于各种因素会影响定位精度，实际位置可能会落在圆圈内的任何地方。这种增加的清晰度有助于决策并增强导航安全。

在此示例中，您在巴黎埃菲尔铁塔附近添加一个 500 米的近距离圈。

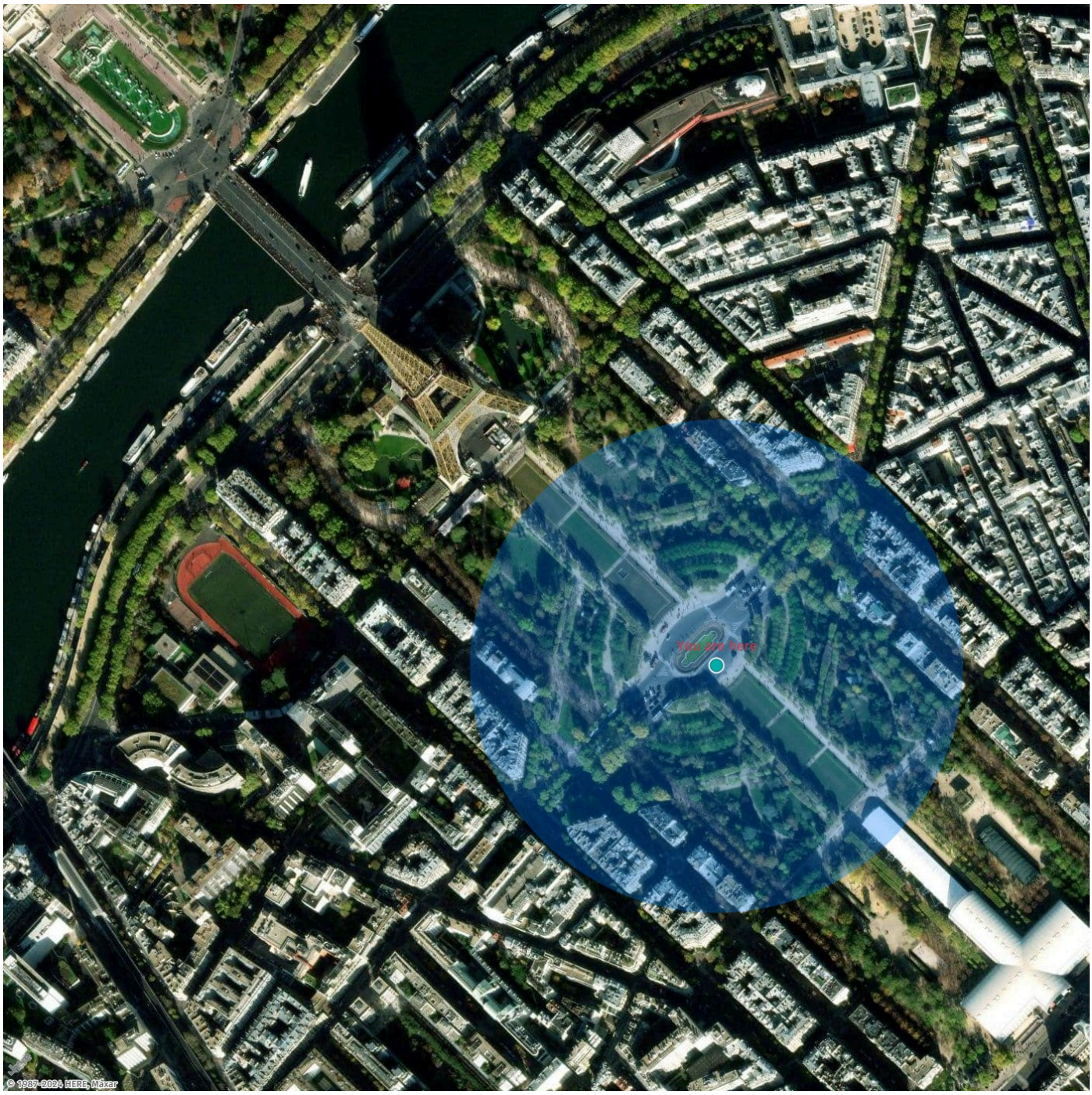
Request

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [
          2.298151,
          48.855921
        ]
      },
      "properties": {
        "style": "circle",
        "color": "#007BFF50",
        "width": "500m"
      }
    },
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [
          2.298151,
          48.855921
        ]
      },
      "properties": {
        "icon": "cp",
        "color": "#007BFF",
        "text-color":"#DC3545",
        "text-outline-color":"#007BFF",
        "outline-width":"20px",
        "label": "You are here",
        "size": "large"
      }
    }
  ]
}
```

Request URL

```
https://maps.geo.eu-central-1.amazonaws.com/v2/static/map@2x?
style=Satellite&width=700&height=700&zoom=17&center=2.2958,48.8570&geojson-
overlay=%7B%20%20%22type%22%3A%20%22FeatureCollection%22,%20%20%22features
%22%3A%20%5B%20%20%20%20%7B%20%20%20%20%20%20%22type%22%3A%20%22Feature%22,
%20%20%20%20%20%20%22geometry%22%3A%20%7B%20%20%20%20%20%20%20%20%22type
%22%3A%20%22Point%22,%20%20%20%20%20%20%20%20%22coordinates%22%3A%20%5B
%20%20%20%20%20%20%20%20%20%202.298151,
%20%20%20%20%20%20%20%20%20%2048.855921%20%20%20%20%20%20%20%20%5D
%20%20%20%20%20%20%7D,%20%20%20%20%20%20%22properties%22%3A
%20%7B%20%20%20%20%20%20%20%20%22style%22%3A%20%22circle%22,
%20%20%20%20%20%20%20%20%22color%22%3A%20%22%23007BFF50%22,
%20%20%20%20%20%20%20%20%22width%22%3A%20%22500m%22%20%20%20%20%20%20%7D
%20%20%20%20%7D,%20%20%20%20%7B%20%20%20%20%20%20%22type%22%3A%20%22Feature
%22,%20%20%20%20%20%20%22geometry%22%3A%20%7B%20%20%20%20%20%20%20%20%22type
%22%3A%20%22Point%22,%20%20%20%20%20%20%20%20%22coordinates%22%3A%20%5B
%20%20%20%20%20%20%20%20%20%202.298151,
%20%20%20%20%20%20%20%20%20%2048.855921%20%20%20%20%20%20%20%20%5D
%20%20%20%20%20%20%7D,%20%20%20%20%20%20%22properties%22%3A%20%7B
%20%20%20%20%20%20%20%20%22icon%22%3A%20%22cp%22,%20%20%20%20%20%20%20%22color
%22%3A%20%22%23007BFF%22,%20%20%20%20%20%20%20%22text-color%22%3A
%22%23DC3545%22,%20%20%20%20%20%20%20%22text-outline-color%22%3A
%22%23007BFF%22,%20%20%20%20%20%20%20%22outline-width%22%3A%2220px
%22,%20%20%20%20%20%20%20%22label%22%3A%20%22You%20are%20here%22,
%20%20%20%20%20%20%20%22size%22%3A%20%22large%22%20%20%20%20%20%7D
%20%20%20%20%7D%20%20%5D%7D&key=your_API_Key
```

Response image



管理成本和使用量

在您继续学习 Amazon Location maps 时，了解如何管理服务容量、确保遵守使用限制以及通过配额和 API 优化获得最佳结果非常重要。通过应用性能和准确性方面的最佳实践，您可以定制应用程序，以高效地处理与地点相关的查询并最大限度地提高 API 请求。

主题

- Amazon Location Service 的最佳实践
- 地图定价
- 映射配额和使用情况

Amazon Location Service 的最佳实践

使用 Amazon Location Service 时，遵循最佳实践可确保您的地图在性能、准确性和用户体验方面得到优化。本节概述了使用静态地图、地理边界和 GeoJSON 数据以增强地图功能和可视化的关键注意事项。

动态地图

以下是在 Amazon Location Service 中使用动态地图的一些最佳实践。

渲染优化 MapLibre

以下是一些 MapLibre 有助于优化 AWS 地图样式渲染的功能。有关更多信息，请参阅 the section called “地图样式”。

跳过样式验证

如果您使用的是 AWS 地图样式，`validateStyle`请设置为`false`。这将关闭加载时样式验证，从而加快初始地图加载速度。 AWS 地图样式不需要样式验证，因为它们是预先验证的。

Example

```
 const map = new maplibregl.Map({
        container: 'map', // ID of the div where the map will render
        style: 'https://maps.geo.${awsRegion}.amazonaws.com/v2/styles/${mapStyle}/
descriptor?key=${apiKey}', // Map style URL
        center: [0, 0], // Starting position [lng, lat]
        zoom: 2, // Starting zoom
        validateStyle: false, // Disable style validation for faster map load
    });
```

Explanation

- `validateStyle: true`：这样可以根据 MapLibre GL 样式规范验证地图样式。如果样式有任何问题，他们将在控制台中登录。

- 如果将其设置为`false`，则地图将跳过样式验证过程，这可能会加快加载速度，但无需进行错误检查。

预热地图

对于可能在用户浏览应用程序时多次创建和销毁地图的单页应用程序 (SPAs)，预热功能可以减少地图被销毁后重新创建地图的延迟。

此功能仅推荐用于 SPAs。

静态地图

边界，边界框（方框）

在处理地图和地理数据时，准确定义边界框 (bbox) 和边界参数至关重要，因为它们决定了感兴趣的地理区域。任何不准确之处都可能导致不良结果。

确保精确的界限

确保指定的边界精确地代表要显示的区域。即使是轻微的不准确也会裁剪或排除所需区域的某些部分，从而无法实现可视化的目的。

验证适当的缩放级别

地图的缩放级别是根据指定的边界或 bbox 自动计算的。验证生成的缩放级别是否为整个感兴趣区域提供了适当的细节和可见性。如果缩放比例过高或过低，地图可能无法有效地传达所需的信息。

检查自定义叠加层的可见性

使用带有自定义叠加层（例如 GeoJSON 要素）的 bbox 或边界时，请确保要素的范围位于生成的地图图像内。超出界限的特征可能会被剪辑或省略，从而导致可视化效果不完整或具有误导性。

在 bbox 中使用填充

使用 bbox 和 padding 参数可确保边缘附近的地图要素完全可见且不会被切断。

通过精确定义 bbox 和 bounds 参数，您可以确保地图正确表示所需的地理区域，提供适当的细节级别，并有效地整合自定义叠加层或数据层。

GeoJSON

使用 GeoJSON 数据时，通过缩小 GeoJSON 来优化查询字符串可以帮助您保持在查询字符串限制范围内，尤其是对于大型数据集。

地图定价

根据地图请求的类型和 API 调用的次数，Amazon Location Service 为其地图 API 提供具有竞争力的价格。本节概述了动态和静态地图的定价结构。

有关详细定价信息，请参阅 Amazon Location Service 定价。

动态地图

地图 API 的定价基于向该 `GetTiles` API 发出的请求数量。

其他与地图相关的内容 APIs，例如`GetGlyphs`、`GetStyleDescriptor`和，`GetSprites`都是免费的。

静态地图图像

静态地图图像的定价基于向 `GetStaticMap` API 发出的请求数量。每次请求提供静态地图图像都会计入定价计算中。

映射配额和使用情况

Amazon Location Service 对动态和静态地图都规定了特定的服务配额和使用限制。设置这些限制是为了确保所有用户的公平使用和性能效率。以下是每项服务的服务配额和可调整限制。

服务配额

Amazon Location Service 为设置默认配额 APIs 以帮助管理服务容量，可以在AWS 服务配额管理控制台中查看这些配额。您可以通过自助控制台请求增加配额，每个 API 的配额最多为默认限额的两倍。

如果配额限制超过默认限制的两倍，请通过自助服务控制台申请，自助服务控制台将自动提交支持请求。或者，也可以联系您的高级支持团队。

增加配额的请求不会直接收费，但是根据消耗的额外资源，更高的使用级别可能会导致服务成本增加。有关更多信息，请参阅 the section called “管理配额”。

动态地图

API 名称	默认	最大可调极限	超过可调整的最大限制
`GetTiles`	2000	4000	在服务配额控制台上申请或联系支持团队

静态地图

API 名称	默认	最大可调极限	超过可调整的最大限制
GetStaticMap	50	100	在服务配额控制台上申请或联系支持团队

使用限制

API 名称	限制	值
GetStyleDescriptor	每个 IP 地址每秒的最大请求数。	5000
GetGlyphys	每个 IP 地址每秒的最大请求数。	5000
GetSprites	每个 IP 地址每秒的最大请求数。	5000
GetStaticMap	压缩后的响应有效载荷大小。	6 MB
GetTiles	压缩后的响应有效载荷大小。	6 MB

术语

有关更多信息，请参阅 the section called “使用条款和数据属性”。

Amazon Location Service 位置



借助 Amazon Location Places，您可以为应用程序添加基于位置的功能。使用地点 APIs，您可以通过查询包含 108 个国家/地区超过 4 亿个地址和兴趣点 (POIs) 的综合地点数据库来搜索或对位置进行地理编码。

什么是地方？

地点是指任何特定地点，包括行政区域、地址 POIs、地理区域等。地点有各种关联信息，例如姓名、地址、坐标、类型，以及营业时间、联系方式、接入点和 POI 类别等详细信息。

- 地址：包括特定的点地址（如办公室、住宅、公寓）、街道地址和插值地址。
- 兴趣点 (POI)：包括企业（如餐馆、商店）和地标（如公园、纪念碑）。
- 行政区域：包括国家、州、省、区和邮政区等区域。
- 地理区域：包括城市、社区和地方等位置。

特征

搜索使用户 APIs 能够通过根据名称、类别或位置进行查询来发现地点、兴趣点 (POIs) 和地址。

- 地理编码 APIs：将物理位置或地址转换为地理坐标（经度和纬度）。
- 预期用途：P APIs laces 允许用户检索和利用各种应用程序的位置数据，从一次性使用到存储数据以备将来参考。默认情况下，检索数据仅供一次性使用，但可以选择永久存储数据以供分析或进一步使用。

使用案例

增强您的结账体验

当客户在网站或应用程序上输入位置时，提供有关有效街道地址的实时建议。这有助于确保准确的送货或取件地点，减少错误并创造无缝的结账体验。通过验证地址来提高购物车完成率，从而提高用户满意度并提高转化率。

欲了解更多信息，请参阅相关内容 APIs：自动完成和建议。

探索当地景点

协助客户找到最近、最相关的地方。例如，房地产客户可以找到附近的学校、杂货店和公园。通过按类别搜索和检索地址和地理编码等基本信息，建立销售服务的目标企业列表并获得见解。

有关更多信息，请参阅相关的 APIs： SearchNearby 和 SearchText。

向客户提供 up-to-date业务详情

提供有关本地企业的最新信息。使用户能够在特定区域内按公司名称或类别进行搜索，从而增强客户体验并支持业务决策。例如，帮助健身房会员找到附近的健康食品商店，或者确保送货司机在营业时间到达。

有关更多信息，请参阅相关内容 APIs： SearchNearby、 SearchText、建议和 GetPlace。

通过详细的见解丰富客户地址

利用邮政编码、地理坐标、联系方式和业务类别等其他见解，增强现有客户地址数据。使用永久存储选项将这些数据保存在数据库中，以便在营销和决策中进行战略性使用。

有关更多信息，请参阅相关内容 APIs：地理编码、反向地理编码和。 SearchText SearchNearby

查找地址的地理空间坐标以计算路线和行车路线

将地址或公司名称转换为地理空间坐标，并计算送餐或拼车服务等任务的最佳路线。使用航点排序为多个停靠点创建有效的路径。

欲了解更多信息，请参阅相关内容 APIs：建议和 SearchText。

确定地点的时区

确定特定地点（城市、地址、坐标或地点 ID）的正确时区。此功能非常适合需要精确时区的行程和日程安排应用程序的旅行应用程序。确保账单和合规记录使用相应的时间戳。

有关更多信息，请参阅相关内容 APIs：地理编码、反向地理编码和。 SearchText GetPlace

地点 APIs 概览

API 名称	描述	其他资源
自动完成	根据部分输入，自动完成功能会在用户键入时完成潜在的搜索地点或地址。自动完成功能在日本不可用。	the section called “自动完成”
地理编码	地理编码或正向地理编码是将文本地址或地点转换为地理坐标。	the section called “地理编码”
反向地理编码	反向地理编码是将地理坐标转换为人类可读的地址或地点。	the section called “反向地理编码”
GetPlace	Get Place 检索有关特定地点的详细信息，例如其地址、营业时间、联系人和其他元数据。	the section called “GetPlace”
建议	Suggest 根据用户的输入或背景（例如相关地点、兴趣点或查询词）提供智能预测或推荐	the section called “建议”
搜索文本	Search Text 提供了使用文本输入搜索地点、地址或兴趣点的功能，它会返回地名、地址、电话、类别、食物类型、联系方式、营业时间等信息。	the section called “搜索文本”
在附近搜寻	“附近搜索” 提供了在指定半径或距离地理坐标距离内搜索感兴趣点的功能。	the section called “在附近搜寻”

地点概念

以下主题提供了使用 Amazon Location Service Places API 的基础知识和最佳实践。这些概念对于在应用程序中有效实现地点功能至关重要。

主题

- 地点术语
- 查询和偏差
- 过滤

- 本地化和国际化
- 联系方式和开放时间
- 其他功能
- IntendedUse

地点术语

本节提供了一些基本定义，供您理解 Amazon Location Service Places 中的核心概念，例如位置数据类型、地理边界和筛选选项。这些条款允许准确使用地点 API 功能。

位置

地球表面的特定点，通常由地理坐标（经度和纬度）定义。位置可以代表任何地点或区域，例如城市、建筑物或兴趣点。

位数

包括行政区域、地址、兴趣点 (POI)、地理区域等的任何地点。地点通常有相关信息，例如姓名、地址、坐标、类型、营业时间、联系人和类别。

地址

包括基于点的地址（如办公室、住宅）、街道地址和插值地址，可提供精确的位置数据。

参见下面的插值定义。

辅助地址

包含辅助标识符的地址，例如套房或单元号、建筑物或楼层信息。

兴趣点 (POI)

POI 是指值得注意的地点，例如企业（如餐馆、商店）或地标（如公园、纪念碑）。

行政区域

组织地理数据的区域，例如国家、省、州、区、街区和邮政区。

地理区域

包括城市、地方和社区在内的区域，可提供更高的地理数据精细度。

位置

一组精确的坐标（经度和纬度），用于精确定位某个地点在地图上的位置。

偏置位置

地理数据中的一个参考点，可帮助确定更靠近该位置的搜索结果的优先级，从而增强搜索的相关性。

边界框

由西南和东北坐标定义的矩形地理区域，用于缩小搜索范围或在地图上显示该区域内的内容。

地点类型

根据功能或特征对地点进行分类。类型包括国家、地区、地点、地区、邮政区、街区、交叉路口、街道地址、点地址、插值地址或 POI。

类别

根据企业和地标提供的服务或活动类型（例如餐厅、酒店、学校和公园）对企业和地标进行分组。类别使用户可以更轻松地 POIs 在搜索中找到特定类型。

比赛分数

一种指示搜索结果与输入的匹配程度的指标，有助于确定相关性。

插值

使用已知地址位置作为参考点来估计未知地址的方法。

ISO 3166 国家/地区代码。

Amazon Location Service Places 使用国际标准化组织 (ISO) 3166 国家/地区代码来识别国家或地区。在 ISO 在线浏览平台上查找每个国家的代码。

查询和偏差

Amazon Location Service Places API 提供查询和偏置选项，用于检索和搜索位置数据。

查询

查询是指用于检索和搜索位置数据的输入参数。 APIs 返回结果的方式由这些查询类型决定。

过滤器类型	地理编码	反向地理编码	自动完成	获取位置	搜索文本	在附近搜寻	建议
QueryText	是	否	是	不适用	是	否	是

过滤器类型	地理编码	反向地理编码	自动完成	获取位置	搜索文本	在附近搜寻	建议
查询组件	是	否	否	不适用	否	否	否
查询职位	否	是	否	不适用	否	是	否
查询半径	否	是	否	不适用	否	是	否
查询编号	否	否	否	不适用	否	否	否
地点编号	否	否	否	是	否	否	否

偏见

“偏见位置” 是影响搜索结果的位置，优先考虑靠近偏见位置的地方。它不限制结果，但会将结果偏向指定区域。当多个地点名称相似时，此功能会优先考虑相关的位置结果。

过滤器类型	地理编码	反向地理编码	自动完成	获取位置	搜索文本	在附近搜寻	建议
BiasPosition	是	否	是	不适用	是	否	是

过滤

Amazon Location Service Places API 提供了筛选选项，可根据特定条件缩小搜索结果的范围。这些参数通过包括或排除地理限制、行政区域、业务链、POI 类别和食物类型等标准来完善结果。

本节详细介绍了 Amazon Location Service Places API 中提供的筛选参数和选项。下文将对每种筛选条件进行说明，包括其对搜索结果的影响。

地理筛选器

地理过滤器将搜索结果限制在指定的地理边界内，例如圆圈或边界框。这允许用户在搜索查询中定位特定区域。

过滤器类型	地理编码	反向地理编码	自动完成	获取位置	搜索文本	在附近搜寻	建议
圆形	否	否	是	不适用	是	否	是
边界框	否	否	是	不适用	支持	是	是

管理区域过滤器

这些过滤器允许用户将搜索结果限制在特定的管理区域，例如国家/地区或地点类型。

过滤器类型	地理编码	反向地理编码	自动完成	获取位置	搜索文本	在附近搜寻	建议
包括国家	是	否	是	不适用	支持	是	是
包括地点类型	支持	是	是	不适用	否	否	否

POI 类别筛选器

POI 类别过滤器根据兴趣点 (POI) 类别（例如特定的业务链或食物类型）来优化搜索结果。

过滤器类型	地理编码	反向地理编码	自动完成	获取位置	搜索文本	在附近搜寻	建议
包括类别	否	否	否	不适用	否	是	否
排除类别	否	否	否	不适用	否	是	否
包括业务链	否	否	否	不适用	否	是	否
排除业务链	否	否	否	不适用	否	是	否

过滤器类型	地理编码	反向地理编码	自动完成	获取位置	搜索文本	在附近搜寻	建议
包括食物类型	否	否	否	不适用	否	是	否
排除食物类型	否	否	否	不适用	否	是	否

过滤器的定义

可以使用以下条件筛选地点类型： FoodType、类别和 BusinessChainId。

圆形

由中心点（经度和纬度）和半径定义的圆形地理区域。此过滤器将结果限制在圆圈内的位置。

边界框

由两组坐标（西南角和东北角）定义的矩形地理区域，将结果限制在此框内的位置。

包括国家

将结果限制在特定国家/地区，仅允许包含指定国家/地区内的地点。

包括地点类型

将结果限制为特定类型的地点，例如国家、城市或社区。

包含和排除类别

根据特定的 POI 类别（例如餐厅或商店）优化结果。您可以添加或排除类别来调整搜索结果。

包括和排除业务链

根据特定的业务链（例如星巴克或 McDonald's）完善业绩。您可以在搜索条件中包含或排除业务链。

包括和排除食物类型

按特定类型的食物相关场所筛选结果，例如意大利菜或中国菜。您可以根据自己的要求包括或排除食物类型。

地点类型筛选条件

这些地点类型在地理搜索中提供了不同级别的精细度，允许用户优化搜索结果以仅返回特定类型，例如国家、地区、街道或个人地址级别。Amazon Location Service 支持以下地点类型：

过滤器类型	地理编码	反向地理编码	自动完成	获取位置	搜索文本	在附近搜寻	建议
PostalCode	是	否	是	不适用	否	否	否
区域	支持	是	是	不适用	否	否	否
交集	是	否	否	不适用	否	否	否
街道	支持	是	否	不适用	否	否	否
PointAddress	是	是	否	不适用	否	否	否
InterpolatedAddress	是	是	否	不适用	否	否	否

类别过滤器

本节详细介绍了 Amazon Location Service 中可用的各种兴趣点 (POI) 类别。这些类别使用户能够将搜索结果筛选到特定类型的地点，从而增强基于位置的应用程序的相关性和用户体验。

过滤器类型	地理编码	反向地理编码	自动完成	获取位置	搜索文本	在附近搜寻	建议
类别	仅在回复中	仅在回复中	不可用	仅在回复中	仅在回复中	在请求和响应中	仅在回复中

Amazon Location Service 支持各种地点类别，允许应用程序 POIs 根据用户需求进行筛选。这些类别根据地点（例如餐馆、医院或学校）缩小搜索结果范围，从而提高了搜索精度。

要检索有关地点类别的信息，请使用带有类别过滤参数的相关地点 API。有关更多信息，请参阅 Amazon Location Service API 文档。

类别名称	类别标识
成人娱乐	成人娱乐
成人用品店	成人用品店
广告营销、公关和市场研究	广告营销，_pr_and_market_research
空中缆车	Aerial_tramway
机场	机场
机场货物	机场货运
机场航站楼	机场航站楼
救护车服务	救护车服务
游乐园	amusement_park
动物公园	animal_park
公寓出租-公寓出租	公寓出租-公寓出租
水族馆	水族馆
美术馆	艺术博物馆
艺术和手工艺用品	艺术品和手工艺用品
静修处	修行所
ATM	atm
律师	律师
汽车零部件	汽车零件
汽车俱乐部	汽车俱乐部

类别名称	类别标识
汽车经销商-新车	汽车_经销商-新车
汽车经销商-二手车	汽车经销商二手车
航空	航空
B2B 餐厅服务	b2b_餐厅_服务
B2B 销售和服务	b2b_sales_and_services
羽毛球	羽毛球
面包店和烘焙食品店	bakery_and_baked_goods_store
银行	银行
宴会厅	宴会厅
酒吧或酒馆	bar_or_pub
理发师	理发师
篮球	篮球
海湾港	海湾港
海滩	海滩
住宿加早餐	床和早餐
啤酒花园	啤酒花园
自行车和自行车配件店	自行车和自行车配件店
自行车停放处	自行车停车
自行车服务	自行车服务
自行车服务和维护	自行车服务和维护

类别名称	类别标识
自行车共享地点	自行车共享地点
自行车公园	bike_park
账单支付服务	账单_付款_服务
台球-台球厅	台球泳池大厅
小酒馆	小酒馆
血库	血库
小轮车商店	bmx_shop
小轮车赛道	bmx_track
乘船渡轮	boat_ferry
划船	划船
水体	水体
身体穿孔和纹身	身体穿孔和纹身
书店	书店
边境口岸	过境
保龄球中心	保龄球中心
啤酒厂	啤酒厂
构建	building
快速公交	bus_rapid_Transit
公交车站	公交车站
公交车站	公交车站

类别名称	类别标识
商业设施	商业设施
商务服务	business_service
Butcher	屠夫
自助餐厅	自助餐厅
露营地	营地
露营徒步用品店	露营徒步_商店
露营地	营地
运河	运河
独木舟/皮划艇店	独木舟皮划艇店
汽车维修	汽车维修
汽车修理服务	汽车维修服务
洗车细节	汽车清洗细节
货运中心	货运中心
货物运输	货物运输
汽车共享地点	汽车共享地点
赌场	赌场
城堡	城堡
休闲餐饮	休闲餐饮
餐饮和其他餐饮服务	餐饮和其他食品服务
手机停车场	手机_parking_lot

类别名称	类别标识
墓地	公墓
支票兑现服务-货币兑换	支票兑现服务-货币兑换
儿童服装	儿童服装
儿童博物馆	儿童博物馆
脊椎按摩师	脊椎按摩师
教堂	教堂
雪茄和烟草店	cigar_and_tobacco_shop
电影院	电影院
市政厅	市政厅
市民社区中心	公民社区中心
服装和配饰	服装和配饰
俱乐部会所	俱乐部会所
教练学院	教练学院
鸡尾酒廊	鸡尾酒酒廊
咖啡店	咖啡店
咖啡-茶	咖啡茶
集体社区	集体社区
商业服务	商业服务
通信媒体	通信媒体
通勤火车站	通勤火车站

类别名称	类别标识
通勤火车	通勤火车
完整的休息区	complete_rest_area
计算机和软件	计算机和软件
构造	施工
消费电子产品商店	消费电子产品商店
消费品	消费品
消费者服务	消费者服务
便利店	便利店
会展中心	会议展览中心
县议会	县议会
快递员	快递员
法院	法院
COVID-19 测试现场	covid-19_testing_site
火葬场	火葬场
越野滑雪用品店	cross_country_ski_
客户服务中心	客户服务中心
乳制品	乳制品
跳舞	跳舞
熟食店	熟食店
配送入口	配送入口

类别名称	类别标识
牙医牙科诊所	牙医牙科办公室
百货商店	百货商店
折扣店	discount_store
酿酒厂	酒厂
潜水中心	潜水中心
甜甜圈店	doughnut_shop
药店	药店
药店或药房	药店_or_pharmacy
干洗和洗衣	干洗和洗衣
教育设施	教育设施
电气	电动的
大使馆	大使馆
排放测试	排放测试
工程和科学服务	工程和科学服务
娱乐和休闲	娱乐和休闲
娱乐电子产品	娱乐_电子
电动汽车电池更换站	ev_battery_swap_station
电动汽车充电站	ev_charging_station
活动空间	事件空间
设施	设施

类别名称	类别标识
家庭餐厅	家庭餐厅
全科医生执业医生	家庭全科医生诊所
农业	养殖业
快餐	快餐
渡轮码头	渡轮码头
金融和保险	金融和保险
金融投资公司	金融_投资_公司
美术	美术
精致餐饮	精致餐饮
消防部门	消防部门
健身健康俱乐部	健身健康俱乐部
地板和地毯	地板和地毯
花店	花店
鲜花和珠宝	花朵和珠宝
食品市场摊位	食品市场摊位
食品生产	食品生产
食品饮料专卖店	食品饮料_专卖店
森林、荒地或其他植被	森林、_健康_或者_其他_植被
加油站	加油站
配送和配送中心	配送和配送中心

类别名称	类别标识
殡仪馆馆长	葬礼导演
家具店	家具店
图库	画廊
赌博-彩票-博彩	gambling-lottery-betting
花园	花园
花园中心	花园中心
普通商品	杂货
礼物、古董和艺术品	礼物，_antique_and_art
玻璃和窗户	玻璃和窗户
高尔夫球场	高尔夫球场
高尔夫练习场	高尔夫练习场
高尔夫商店	golf_shop
政府办公室	政府办公室
政府或社区设施	政府或社区设施
杂货店	杂货店
招待所	招待所
Gurdwara	gurdwara
发型与美容	头发和美容
美发沙龙	美发沙龙
五金、房屋和花园	硬件、房屋和花园

类别名称	类别标识
医疗保健和医疗保健支持服务	医疗保健和医疗保健支持服务
高等教育	高等教育
高速公路出口	高速公路出口
历史纪念碑	历史纪念碑
历史博物馆	历史博物馆
曲棍球	曲棍球
假日公园	holiday_park
家居装修	家居装修
家居专卖店	主页_Specialty_store
医院	医院
医院急诊室	医院急诊室
医院或 Health Care 机构	医院或医疗机构
旅馆	宿舍
温泉	炎热的春天
酒店	酒店
酒店或汽车旅馆	酒店或汽车旅馆
人力资源和招聘服务	人力资源和招聘服务
狩猎钓鱼店	狩猎钓鱼店
氢燃料站	氢燃料站
溜冰场	溜冰场

类别名称	类别标识
倾斜轨道	倾斜轨道
室内滑雪	室内滑雪
室内运动	室内运动
工业区	工业区
内部和外部设计	内部和外部设计
网吧	网吧
调查服务	调查服务
岛屿	岛屿
IT 和办公设备服务	it_and_Office_设备_服务
爵士乐俱乐部	爵士俱乐部
珠宝商	珠宝商
卡拉OK	卡拉OK
幼儿园和托儿所	幼儿园和托儿所
湖	湖
地标景点	地标景点
园林绿化服务	园林绿化服务
语言研究	语言研究
法律服务	法律服务
休闲	空闲的
图书馆	库

类别名称	类别标识
轻轨	轻轨
现场娱乐-音乐	现场娱乐音乐
装货码头	loading_dock
装货区	加载区域
本地交通	本地公交
锁匠和安全系统服务	locksmiths_and_security_sems_services
住宿	寄宿
彩票摊位	lottery_booth
木材	木材
女佣服务	女佣服务
大型家电	主要设备
管理和咨询服务	管理和咨询服务
制造业	制造业
玛丽娜	码头
Market	市场
婚姻和婚介服务	结婚和婚介服务
医疗服务-诊所	医疗服务-诊所
集合点	会议点
男士服装	男士服装
军事基地	军事基地

类别名称	类别标识
采矿、采石和其他开采	采矿、采石和其他开采
移动零售商	移动零售商
移动服务中心	移动服务中心
模特经纪公司	模特经纪公司
汇款服务	汇款服务
单轨	单轨列车
清真寺	清真寺
汽车旅馆	汽车旅馆
摩托车配件	摩托车配件
摩托车经销商	摩托车经销商
摩托车服务和维护	摩托车服务和维护
摩托车、轻便摩托车和小型摩托车停车场	摩托车、轻便摩托车和踏板车停车
高速公路服务休息区	高速公路服务休息区
山或山	山或山
山口	山口
山峰	山峰
Mover	行动者
博物馆	博物馆
美甲沙龙	美甲沙龙
命名为 Intersection-Chowk	named_intersection-chowk

类别名称	类别标识
自然和地理	自然和地理
夜总会	夜总会
夜生活娱乐	夜生活娱乐
非门店零售商	非商店零售商
疗养院	nursing_home
越野步道起点	off_road_trailhead
越野车区	越野车辆区域
办公用品和服务商店	办公用品和服务商店
光学	光学
组织与社团	组织和社团
其他书店	其他_书店
其他图书馆	其他图书馆
其他礼拜场所	其他宗教场所
户外区域综合体	户外区域综合体
户外服务	户外服务
户外娱乐	户外娱乐
宝塔	宝塔
油漆店	油漆店
停车和骑行	park_and_ride
公园-休闲区	公园娱乐区

类别名称	类别标识
停车	parking
仅限停车场和洗手间的休息区	parking_and_restroom_only_restroom_only
停车库-停车房	parking_garage-parking_hou
停车场	parking_lot
仅限停车的休息区	仅限停车休息区
当铺	当铺
表演艺术	表演艺术
宠物护理	宠物护理
宠物用品	宠物用品
汽油加油站	汽油加油站
药房	药店
摄影	摄影
管道	管道
警察岗亭	警察箱
警务-安全	警察服务安全
警察局	警察局
邮局	邮局
邮政收集箱	postal_clection_box
电力设备经销商	电力设备经销商
小学	小学

类别名称	类别标识
印刷和出版	印刷和出版
物业管理	物业管理
精神病研究所	精神病学研究所
公共行政	公共管理
公共洗手间-厕所	公共厕所厕所
公共体育机场	公共体育机场
公共交通通道	公共交通访问权限
竞赛跑道	race_track
壁球场	壁球球场
铁路渡轮	铁路渡轮
铁路堆场	rail_yard
护林员站	Ranger_station
房地产服务	房地产服务
录制、CD 和视频	录制、_cd_and_video
娱乐中心	娱乐中心
回收中心	回收中心
注册办公室	注册_办公室
宗教场所	宗教场所
租赁和租赁	租赁和租赁
租车公司	租车_汽车_代理

类别名称	类别标识
维修和保养服务	维修和维护服务
维修服务	维修服务
水库	水库
住宅区建筑	住宅区建筑
休息区	休息区
Restaurant	餐厅
拼车接送	拼车接送
River	河
道路援助	道路援助
路边站	路边站
橄榄球	橄榄球
跑步轨道	跑步_track
跑步-步行商店	跑步-walking_shop
房车公园	rv_parks
日式旅馆	日式旅馆
风景秀丽的俯瞰休息区	scenic_overlook_rest_area
风景点	scenic_point
学校	学校
科学博物馆	科学博物馆
海港-海港	海港港

类别名称	类别标识
中学	中学
鞋类-鞋类	鞋类-鞋类
射击范围	射击范围
购物中心	Shopping_mall
短期汽车旅馆	短期汽车旅馆
神社	神社
滑板店	skate_shop
滑雪缆车	滑雪缆车
滑雪胜地	滑雪胜地
滑雪用品店	滑雪店
滑雪板商店	滑雪板商店
足球俱乐部	soccer_club
社会服务	社交服务
专业服装店	专卖店_服装店_专卖店
特色食品店	specity_food_store
专卖店	专卖店
专业行业承包商	专业贸易承包商
体育用品店	体育用品商店
体育教学和夏令营	体育教学和训练营
体育活动	体育活动

类别名称	类别标识
体育中心-体育场	体育场综合体
体育设施-场地	体育设施场地
运动场	运动场
壁球场	壁球场
存储	存储
学生住房	学生住房
冲浪店	surf_shop
甜品店	甜品店
游泳池	swiming_pool
犹太教堂	犹太会堂
Tack Shop	tack_Shop
裁缝和改装	裁缝和改动
仅限外卖和配送	仅限外卖和配送
日光浴沙龙	晒黑沙龙
Taqueria	taqueria
税务服务	税务服务
出租车站	出租车站
茶馆	茶屋
电话服务	电话服务
寺庙	寺庙

类别名称	类别标识
网球场	网球场
戏剧、音乐和文化	戏剧、_音乐_和_文化
治疗师	治疗师
轮胎维修	轮胎维修
收费站	收费站
旅游景点	旅游景点
游客信息	游客信息
拖车服务	拖车服务
玩具店	玩具店
Trailhead	trailhead
火车站	火车站
培训与发展	培训与发展
笔译和口译服务	笔译和口译服务
运输服务	交通服务
旅行社-票务	旅行社票务
卡车经销商	卡车_经销商
卡车停车	卡车停车
卡车维修	卡车维修
卡车停靠广场	卡车停靠广场
卡车清洗	cruck_wash

类别名称	类别标识
卡车-半挂车经销商-服务	卡车半经销商服务
地下列车-地铁	地下列车-地铁
海底特征	海底特征
紧急护理中心	紧急护理中心
二手二手商品商店	二手二手商品商店
实用程序	utilities
疫苗接种地点	疫苗接种网站
货车维修	货车维修
电子烟商店	vaping_store
杂货店	variety_store
兽医	兽医
视频和游戏租赁	视频和游戏租赁
视频街机游戏室	video_arcade-game_room
数据仓库	warehouse
废物和卫生	废物和卫生
水上乐园	水上乐园
水上交通	水上交通
瀑布图	瀑布
婚礼服务和新娘工作室	婚礼服务和新娘工作室
称重站	称重站

类别名称	类别标识
健康中心和服务	健康中心和服务
批发商店	批发_商店
野生动物园	wild_animal_park
野生动物保护区	野生动物避难所
葡萄酒和白酒	葡萄酒和白酒
酒庄	酒厂
女士服装	女士服装
动物园	动物园

食物类型过滤器

本节概述了 Amazon Location Service 中支持的食物类型筛选条件。这些过滤器允许您缩小地点的搜索范围，特别关注美食类型或与食物相关的兴趣点（POIs），从而为寻求特定餐饮选择的用户增强基于位置的搜索。

Amazon Location Service 中的食物类型筛选功能可帮助按特定类型的美食或食品场所优化搜索结果，支持用户对定向位置搜索的偏好。以下是支持的食物类型及其 IDs。

要应用食物类型过滤器，请 APIs 使用将`foodType`参数设置为所需美食的 Place。有关实施这些筛选条件的详细指导，请参阅 Amazon Location Service API 文档。

名称	FoodType 身份证
美国的	美国的
美式烧烤/南方	美式烧烤/南方
美国印第安人	美国印第安人
美国加州人	美国加州人

名称	FoodType 身份证
美式克里奥尔语	美式克里奥尔语
美洲原住民	美国原住民_美国人
美国灵魂食品	美国灵魂食品
美国-南方	美国南方
美国-西南地区	美国西南部
阿根廷的	阿根廷的
亚美尼亚语	亚美尼亚
亚洲的	亚洲人
澳大利亚人	澳大利亚人
奥地利人	奥地利人
阿塞拜疆语	阿塞拜
巴尔干	巴尔干半岛
波罗的海	波罗的
比利时的	比利时的
白俄罗斯人	白俄罗斯人
小酒馆	小酒馆
波希米亚	波希米亚风格
巴西人	巴西人
巴西-巴亚纳	巴西人-巴亚纳
巴西面包店	巴西面包店

名称	FoodType 身份证
Brazilian-Capixaba	brazilian-capixaba
Brazilian-Mineira	brazilian-Mineira
早餐	早餐
BrewPub	啤酒吧
不列颠群岛	British_isles
早午餐	早午餐
文莱人	文莱的
汉堡	汉堡
缅甸语	缅甸人
柬埔寨的	柬埔寨
加拿大人	加拿大人
加勒比的	加勒比海
高加索	高加索的
鸡	鸡
智利人	智利人
中文	中国人
中国-北京	中国-北京
汉语-广东话	汉语-广东话
汉语-广西	广西汉语
中式火锅	中国火锅

名称	FoodType 身份证
湖南汉语/湖北语	湖南汉语/湖北语
汉语-内蒙古语	汉语内蒙古语
中国-伊斯兰教	中国伊斯兰教
中国-江苏/浙江	中国江苏/浙江
汉语-江西	汉语-江西
中国-东北人	中国东北地区
中国西北地区	中国西北
中国粥	中国粥
中文-山东	山东汉语
汉语-上海	中国-上海
中国四川	中国四川
华裔台湾人	中国台湾人
汉语云南语/贵州	汉语/云南语/贵州
大陆	大陆的
Creperie	creperie
古巴人	古巴的
丹麦语	丹麦语
晚餐	晚餐
荷兰语	荷兰人
东欧	东欧

名称	FoodType 身份证
埃及的	埃及人
埃塞俄比亚的	埃塞俄比亚的
欧洲的	欧洲的
快餐	快餐
菲律宾人	菲律宾人
芬兰语	芬兰语
火锅	火锅
法语	法语
法语-阿尔萨斯语	法语-阿尔萨斯语
French-Auvergnate	french-auvergnate
法语-巴斯克语	法语-巴斯克语
法语-科西嘉岛	法语-科西嘉语
法国里昂奶酪	french-lyonnase
法国-普罗旺斯语	法国普罗旺斯语
French-sud-Ouest	french-sud-ouest
融合	融合
德语	德国人
希腊语	希腊语
烧烤	烧烤
清真	清真

名称	FoodType 身份证
夏威夷/波利尼西亚人	夏威夷/波利尼西亚人
热狗	hot_dogs
匈牙利语	匈牙利语
冰淇淋	冰淇淋
印度人	印度人
印度-孟加拉语	印度-孟加拉语
Indian-Goan	Indian-goan
印度语-古吉拉特语	印度-古吉拉特语
印度-海得拉巴	印度海得拉巴
Indian-Jain	Indian-Jain
Indian-Konkani	indian-konkani
印度-马哈拉施特拉邦	印度马哈拉施特拉邦
Indian-Malvani	印度马尔瓦尼
Indian-Mughlai	indian-Mughlai
印度-北印度人	印度-北部_印度人
Indian-Parsi	Indian-parsi
印度-旁遮普语	印度-旁遮普语
印度-拉贾斯坦邦	印度-拉贾斯坦语
印度-南印度人	印度-南印度人
Indian-tandoori	印度唐杜里

名称	FoodType 身份证
印度尼西亚语	印尼人
国际	国际
爱尔兰语	爱尔兰人
意大利语	意大利人
日语	日本人
日语-Chanko	日本相声
日式咖喱	日式咖喱
日本鱼/其他海鲜	日本鱼/其他_海鲜
日式饺子	日本饺子
日语-Hibachi	日语 hibachi
日式居酒屋	日式居酒屋
日式成吉思汗馆	日本成吉思汗馆
日式怀石料理	日式怀石料理
日式御好烧	日式御好烧
日式拉面	日本拉面
日式饭碗	日本饭碗
日式涮涮锅	日式涮涮锅
日式寿喜烧	日式寿喜烧
日式寿司	日本寿司
日式寿司列车	日本寿司火车

名称	FoodType 身份证
日式章鱼烧	日式章鱼烧
日式-天妇罗/其他油炸食品	日本天妇罗/其他油炸食品
日式炸猪排	日本炸猪排
日式乌冬面/其他面条	日本乌冬面/其他面条
日本-鳗鱼/鳗鱼	日本鳗鱼/anago
日式烤肉	日式烤肉
日式-烤鸡肉串/鸡肉	日式烤鸡肉串/鸡肉
犹太人/犹太洁食	犹太人/犹太洁食
烤肉串	烤肉串
韩语	韩国的
拉丁美洲	拉丁裔美国人
黎巴嫩人	黎巴嫩人
午餐	午餐
马来西亚的	马来西亚人
马耳他语	马耳他语
地中海的	地中海
墨西哥的	墨西哥的
墨西哥-瓦哈克纳	墨西哥-瓦哈克纳
Mexican-Poblana	mexican-poblana
墨西哥-韦拉克鲁萨纳	墨西哥-韦拉克鲁扎纳

名称	FoodType 身份证
墨西哥-尤卡特卡	墨西哥尤卡特卡
中东	中东地区
摩洛哥人	摩洛哥的
自然/健康	自然/健康
面条	面条
北非	北非
挪威语	挪威语
海洋的	海洋
有机	有机
巴基斯坦人的	巴基斯坦人
糕点	糕点
秘鲁人	秘鲁人
披萨	pizza
波兰语	抛光
葡萄牙语	葡萄牙语
罗马尼亚语	罗马尼亚语
俄语	俄语
三明治	三明治
斯堪的纳维亚人	斯堪的纳维亚
海鲜	海鲜

名称	FoodType 身份证
塞舌尔人	塞舌尔人
西西里	西西里人
新加坡人	新加坡人
零食和饮料	零食和饮料
汤	汤
南非的	南非_非洲
南美洲	南美
东南亚菜	东南亚洲
西班牙语	西班牙语
西班牙小吃	西班牙小吃
斯里兰卡人	sri_lankan
牛排馆	steak_house
苏里南的	苏里南人
瑞典语	瑞典语
瑞士人	瑞士人
泰语	泰国
藏族	藏族
土耳其语	土耳其语
乌克兰语	乌克兰人
纯素食	纯素主义者

名称	FoodType 身份证
素食	素食主义者
委内瑞拉人	委内瑞
越南语	越南语
也门人	也门人

商业链过滤器

本节介绍如何在亚马逊定位服务的地点 API 中使用业务链标识符筛选搜索结果。此功能允许用户根据知名业务链优化位置搜索，从而帮助增强针对特定品牌请求的搜索体验。

商业连锁店筛选器允许用户指定一个或多个连锁店标识符，从而获得有针对性的搜索结果，这对于查找特定的品牌位置（例如咖啡店、加油站）非常有用。以下是支持的业务链列表。

商业链

名称	ID
&其他故事	&Other_Storie
&STYLE	&STYLE
和披萨	和披萨
+ K 个杯赛	+_K_COPAS
+Celeste	+Celeste
+Qoutlet	+Qoutlet
0	50 Espigas y Migas
03	03
0371 澳门斗老	0371_Ao_Men_Dou_Lao
082 洗车	082_CARWASH

名称	ID
087 Jalta	087_Jalta
1 FBBank	1_ FBBank
1 停止	1_Stop
1 Stop Mart	1_Stop_Mart
1 一	1_一
1+1 披萨	1+1_Pizza
1-2-3	1-2-3
1-800-DryClean	1-800-DryClean
1-800-Flowers	1-800-Flowers
1-800-GOT-JUNK？	1-800-GOT-JUNK？
1-800-散热器和空调	1-800-Radiator_&_A/C
1. 2. 3.	1._2._3。
10 个糕点	10_PASTÉIS
10-11	10-11
5 月 10 日	5 月 10 日
100 小时咖喱 B&R	100_Hours_Curry_B&R
100 Man Volt	100_man_Volt
100 Montaditos	100_Montaditos
100 日元	100_Yen
100	100_

名称	ID
1000 i Odna Sumka	1000_i_odna_sumka
1000 i odna tufel'ka	1000_i_odna_tufel'ka
1001 巡回演出	1001_tur
100inn	100inn
101 Drogerie	101_Drogerie
101 文具乐园	101_Stationery_Paradise
1010	1010
105	105
108 Shop	108_Shop
111 药店	111_Medicine_Store
118 家旅馆	118_Inns
12 Hot Pot	12_Hot_Pot
12 mesyacev	12_mesyacev
12 mesyatsev	12_mesyatsev
120 na 80	120_na_80
121 金融 CU	121_financial_cu
123 个补充装	123 个补充装
17.5 电影院	17.5_Cinemas
1901	1901
1929 年家庭工作室	1929_Home_Studio

名称	ID
1a 汽车服务	1a_汽车服务
1 分钟	1 分钟
第一优势 FCU	1st_Advantage_FCU
1st Liberty	1st_Liberty_fcu
第一水手银行	1st_Mariner_Bank
第 1 个 MidAmerica CU	1st_ _CU MidAmerica
第一密西西比 FCU	1st_missippi_fcu
第一资源 CU	1st_resource_cu
2	2_
21 世纪	21_Century
21 世纪便利店	21_Century_便利店
21 Cineplex	21_Cineplex
21Gold	21Gold
220 伏特	220_volt
东街 231 号	231_east_ST
24 小时健身	24_Hour_Fitness
24 小时市场	24 小时_Market
241 Pizza	241_Pizza
24FAB	24FAB
24 小时停车时间	24 小时_次_停车

名称	ID
24seve	24seve
24tps	24tps
25 Chasov	25_Chasov
25 FIT	25_FIT
2Go	2Go
2N Shop	2N_Shop
2XU	2XU
2 度	2 度
第二街	2nd_street
2 比索	2 比索
3 家最佳酒店	3_Best_Hotel
3 宽带	3_宽带
3 Modas	3_Modas
3 Royalty 3	3_Royalty_3_House
3 家门店	3_Store
3 不错	3_不错
30 Shine	30_Shine
33 Pingvina	33_Pingvina
345 Store	345_Store
36	6

名称	ID
36.7	36.7
365 尽情享受购物	365_享受_购物
365PLUS	365PLUS
3 @1	3 @1
3A Takarék	3A_Takarék
3B 酒店	3B_Hotel
3Butiken	3Butiken
3 个硬币	3 个硬币
3D-GOLD	3D-GOLD
3G 移动世界	3G_Mobile_World
3 把枪	3 把枪
3M 汽车保养	3M_Car_Care
3Passos	3Passos
3Q 舒适又好吃	3q_cozy_&_Yummy
3Rivers FCU	3rivers_FCU
3Shop	3Shop
3T	3T
4 英国人	4_Engletes
4 Sea	4_Sea
4 赛季	4_Sezona

名称	ID
4 你	4_YOU
4 pieds	4_pieds
47 号街	47_Street
4D	4D
4F	4F
4 根手指	4 根手指
4all	4all
4as 市场	4as_Market
4 张床	4 张床
5 个元素	5_ELEMENT
5 Karmanov	5_Karmanov
五星限注底池	5_Star_Limit_Pot
5 个市场	5_market
5' 马里诺普洛斯	5'_Marinopoulos
5.10.15。	5.10.15。
5.11 战术	5.11_Tactical
5.7.9 门店	5.7.9_Stores
50 个 Croquetas	50_Croquetas
50 局域网	50_Lan
50 个披萨	50_Pizza

名称	ID
50 个 Sabores	50_Sabores
50 Style	50_Style
第 50 大道	第 50 大道
第 54 街	第 54_street
551 蓬莱	551_ponglely
576 KByte	576_ KByte
585 Zoloto Yuvelirniy Magazin	585_Zoloto_Yuvelirniy_Magazin
5D Cinema MAXIM	5d_cinema_maxim
第五大道	第 5 大道
5yue	5yue
5àsec	5àsec
6 猫头鹰门	6_Owl_Door
6 Ten	6_Ten
6 英寸盘子	6 英寸盘子
612 联赛	612_League
6 IXTY8 IGHT	6 IXTY8 IGHT
7 17 信用合作社	7_17_Credit_Union
7 天酒店	7_Days_Inn
7 Tian You Pin	7_Tian_You_Pin
7 Tup	7_Tup

名称	ID
7 for 全人类	7_for_all_Mankind
7 个新鲜的	7_fresh
7-11	7-11
7/24 Mix	7/24_Mix
70 年代餐厅	70S_餐厅
71 导出	71_Export
71 Settantuno	71 Settantuno
72 chealthyFood	72 chealthyFood
759Store	759Store
76	76
美国之鹰的 77kids	77kids_by_American_Eagle
7camicie	7camicie
7street	7street
第七天堂	7th_Heaven
8 MARTA Fabrika Mebeli	8_marta_fabrika_mebeli
8 Pot	8_Pot
8 种方式	8_way
8 a Huit	8_à_Huit
8090	8090
84 Lumber	84_Lumber

名称	ID
85C 面包店咖啡厅	85C_Bakery_Cafe
8funyay	8funyay
8inn	8inn
8 路	8 路
911 修复	911_Restoration
918	918
98。4 化学家	98._4_Chemist
仅限 99 美分的商店	99_Cents_Only_Stores
99 个煎饼	99_Pancakes
99 牧场市场	99_ranch_Market
99 餐厅	99_Restauran
99 Speed	99_Speed_Mart
99 辛标	99_Xin_Biao
99inn	99inn
99rent	99rent
9 轮健身	9回合_健身
9noventay9	9noventay9
9x9 文具精通	9x9_文具_精通
:CHOCOOLATE	:CHOCOOLATE
？ EZ	？ EZ

名称	ID
@ 主页	@_Home
A 和 C	A_&_C
A Aptieka	A_Aptieka
一家银行	A_Bank
一道漂亮的汤	A_漂亮的汤
A Cascata	A_Cascata
A Chau Gas	A_Chau_Gas
一支干净的香烟	A_Clean_Crean
一个更清洁的世界	A_Cleaner_World
一家家庭旅馆	A_HOME_HOTEL
A Hereford Beefstouw	A_Hereford_Beefstouw
啊啦莎	A_LA_SHA
A Loja do Gato Preto	a_loja_do_gato_preto
一个集市	A_MART
一个愉快的假期	A_不错的假期
一份不错的礼物	A_漂亮的礼物
A PADARIA PORTUG	A_PADARIA_PORTUG
Pea in the Pod	a_pea_in_the_pod
A Plus 玻璃杯	A_plus_Glas
一个二人组的地方	_TWOSOME_PLACE

名称	ID
A Xi Ya	A_Xi_Ya
阿英宝	A_英宝
一头酸奶牛	A_Yougurt_Cow
啊是的哦	a_ja_o
A −	A_-
A&A 会议室	A&A_房间
A&O	A&O
A&O 旅馆	A&O_Hostels
A&O 现实	a&o_Reality
A&P	A&P
A&P Pharmacy	A&P_Pharmacy
A&W	A&W
A&W 加拿大	A&W_Canada
A-1	A-1
A-1 面包店	A-1_BAKERY
A-1 混凝土找平	A-1_混凝土_找平
A-100	A-100
A-AAAKey 迷你仓	A-AAAKey _mini_Storage
A-Bank	A-Bank
A-K-S 连锁酒店	A-K-S_Chain_Hotel

名称	ID
A-KUAN	A-KUAN
A-Liren	A-Liren
A-MEGA	A-MEGA
A-Wear	A-Wear
A-Z domace potreby	a-z_domace_Potreby
A. Blikle	A._Blikle
安德烈·桑托斯	A._C._Santos
A. 租金	A._Rental
A. Best	A. Best
A.C. Moore	A.C._Moore
A.C.T	A.C.T
A.G. Edwards & Sons 公司	A.G._Edwards_&_Sons_Corporated
A.H. Wheeler	A.H._Wheeler
A.LANGE&SOHNE	A.LANGE&SOHNE
A.Net.Tai	A.Net.Tai
A.S. Watson 集团	A.S._Watson_Group
A.S.O	A.S.O
A.T.U 汽车服务专业版 Kazdého	A.T.U_Autoservis_Pro_Kazdého
A.T.U 汽车超市和Officina	a.t.u_autosupermarket_e_Offic
A.T.U Die Meisterwerkstatt	A.T.U_Die_Meisterwerkstatt

名称	ID
A.T.U Die Nr.1 Meisterwerkstatt	A.T.U_Die_Nr.1_Meisterwerkstatt
A.T.U Dé 完整车库	A.T.U_de_Complete_Garage
a.v.e	a.v.e
A1	A1
A1 改动	A1_改动
A1 黎巴嫩面包店	A1_Lebanese_Bakery
A1 商店	A1_Shop
A101	A101
A24	A24
A2A	A2A
A3 SPORT	A3_SPORT
A5	A5
机管局中心	aa_center
AA 连锁酒店	aa_chain_Hotel
AA 交易所	aa_交易所
AA 药房	aa_药房
AAA	AAA
AAA FCU	AAA_FCU
AAMCO 变速箱	aamco_Transmission
AB 银行	AB_Bank

名称	ID
AB 酒店	AB_Hotels
AB Vassilopoulos	ab_vassiloPoulos
ABA	ABA
ABAD 建设者	abad_Builders
ABANCA	ABANCA
ABARTH	ABARTH
减弱	减弱
ABC	ABC
美国广播公司面包店	ABC_bakery
美国广播公司银行	ABC_Bank
美国广播公司资本	ABC_Capital
美国广播公司便利店	ABC_便利店
美国广播公司烹饪工作室	ABC_Cooking_Studio
美国广播公司药店	ABC_Drug_Stores
ABC 精品葡萄酒和烈酒	ABC_fine_wine_&_Spirits
美国广播公司外语学校	ABC_forein_language_school
ABC MART	ABC_MART
ABC Pneus	ABC_pneus
美国广播公司舒赫中心	ABC_SchuhCenter
ABC Schuhe	ABC_Schuhe

名称	ID
美国广播公司商店	ABC_shop
美国广播公司供应公司	ABC_Supply_co
美国广播公司仓库	美国广播公司_仓库
ABC 大舞台	ABC_GRAND_STAGE
ABC-MART	ABC-MART
美国广播公司体育	ABC_SPORTS
ABF 货运系统	ABF_freight_System
ABFOTO	ABFOTO
阿比里尼	阿比里尼
ABK	ABK
ABN AMRO	ABN_AMRO
ABNB FCU	ABNB_FCU
ABRA 汽车车身和玻璃	abra_autobody_&_Glass
ABSA	ABSA
绝对是烧烤	绝对烧烤
ABSOLYuTBANK	ABSOLYuTBANK
万豪 AC 酒店	万豪酒店 ac_hotels_by
AC 餐厅	ac_餐厅
ACB	ACB
ACCEL	ACCEL

名称	ID
REFORM 连锁药店	ACCORDANCE_PHARM_连锁店
ACDelco	ACDelco
王牌	王牌
ACE 现金快递	ace_cash_Expr
王牌酒店	ace_Hotel
ACE 租车	ace_rent_a_Car
ACHAIKI 合作银行	ACHAIKI_COPERATIVE_BANK
ACIMA	ACIMA
ACMG FCU	ACMG_FCU
ACO 硬件	ACO_硬件
ACO Securité	aco_securité
ACREMEX	ACREMEX
ACS	ACS
战术	战术
ACTUS	ACTUS
AD	AD
AD Delhaize	ad_delhaize
AD 牛仔裤	ad_Jeans
广告生活方式	广告_生活方式
广告 VaraosaMaailma	广告_ VaraosaMaailma

名称	ID
ad-autokorjaamo	ad-autokorjaamo
阿达玛斯	阿达玛斯
ADCB	ADCB
ADEG	ADEG
阿迪拉财经	ADIRA_FINANCE
ADIYÖREM CIG KOFTE	ADIYÖREM_CIG_KÖFTE
ADNOC	ADNOC
阿道夫·多明格斯	ADOLFO_DOMINGUEZ
ADP 小型企业服务	ADP_小型企业_服务
ADT	ADT
提前进站	高级进站
AEKO 商店	AEKO_SHOP
AEON	AEON
AEON Big	Aeon_big
永旺汽车	永旺汽车
空中闪光	空中闪光
气溶胶	气溶胶
AEZ	AEZ
AFB&T	AFB&T
亚足联	亚足联

名称	ID
AFFIN HWANG 资本	AFFIN_HWANG_CAPITAL
AFFN	AFFN
AFURI	AFURI
AGARRATE CATALINA	AGARRATE_CATALINA
阿加莎	阿加莎
AGATHA RUIZ DE LA PRADA	AGATHA_RUIZ_DE_LA_PRADA
AGB	AGB
AGI PASTA	AGI_PASTA_AWAY
敏捷	敏捷
阿格拉	阿格拉
纳米比亚农业银行	纳米比亚农业银行
农业 DVLP'T 中国银行	中国农业_DVLP'T_BANK_OF
农业市场	农业市场
农业银行	农业银行
AGROPROMREDIT	AGROPROMREDIT
AGRUPA-T 旅行	AGRUPA-T_TRAVEL
AHCL Homes	ahcl_Homes
AHORN Wohlfühlhotels	ahorn_wohlfühlHotels
AIA	AIA
AIGLE	AIGLE

名称	ID
AIGLE DEPUIS	AIGLE_DEPUIS
艾格纳	艾格纳
AIIZ	AIIZ
AIK Banka	aik_banka
AILI	AILI
AIMÉ POULY	AIME_POULY
AION	AION
AIOU	AIOU
中国国航	中国国际航空
澳门航空	澳门航空
空气空间	AIR_SPACE
AIS	AIS
AITO	AITO
AJ 的精美食品	aj's_fine_Foods
味千拉面	味千拉面
AJU 客家餐厅	aju_Hakka_餐厅
AK BARS	AK_BARS
AKADEMIBOKHANDELN	AKADEMIBOKHANDELN
阿克纳酒店	AKENA_HOTEL
AKER	AKER

名称	ID
AKIBAN	AKIBAN
AKIPERTO	AKIPERTO
AKO	AKO
AKR	AKR
雅克波利斯	雅克波利斯
AKTIV BANK	AKTIV_BANK
阿克森特酒店	Akzent_Hotels
又名 SUPERBODEGA	AKÁ_SUPERBODEGA
所有连衣裙	AL_DREES
AL FARDAN 交易所	AL_FARDAN_EXCHANGE
AL FAROOJ FRESH	AL_FAROOJ_FRESH
阿尔·福尔诺	al_Forno
AL GHURAIR 交易所	AL_GHURAIR_EXCHANGE
AL广场	AL_PLAZA
AL SALAM 银行	AL_SALAM_BANK
AL-PETRO	AL-PETRO
ALAIN AFFELOU	ALAIN_AFFLELOU
ALBA	ALBA
ALCO	ALCO
ALDI NORD	ALDI_NORD

名称	ID
ALDI SÜD	ALDI_SÜD
ALDO	ALDO
ALEC-雅培实验室 ECU	Alec_-_abbott_labs_ecu
ALEF	ALEF
亚历山德罗夫斯基	亚历山德罗夫斯基
ALEMAGNA	ALEMAGNA
警报引擎零件	警报引擎零件
亚历克斯	亚历克斯
ALFA	ALFA
阿尔法罗密欧	ALFA_ROMEO
ALFAMART	ALFAMART
ALFAVIT	ALFAVIT
ALIANZA	ALIANZA
ALIMERKA	ALIMERKA
ALIOR BANK	ALIOR_BANK
ALKOSTO	ALKOSTO
全部在 CU	All_in_cu
所有套房公寓式酒店	ALL_SUITES_APPART_HOTEL
艾伦职业学院	Allen_Career_Institute
ALLEN SOLLY	ALLEN_SOLLY

名称	ID
美食联盟	ALLIANCE_GASTRONOMIQU
ALLO 电信	allo_Tele
ALLTOWN	ALLTOWN
施舍。BRAND	施舍。 _品牌
ALMAZ-KHOLDING	ALMAZ-KHOLDING
ALMI	ALMI
ALMIDECOR	ALMIDECOR
ALNO	ALNO
阿尔法银行	ALPHA_BANK
高山职业选手	ALPINE_PRO
ALPS FCU	ALPS_FCU
ALPrime 能量	ALPrime_能源
ALSARA	ALSARA
ALSEN	ALSEN
还有米洛咖啡	还有_MILO_COFFEE
ALTECH 自动寻呼蜂窝网络	ALTECH_AUTOPAGE_CELLAL
ALTOR	ALTOR
ALUMIL	ALUMIL
ALVI	ALVI
ALYA	ALYA

名称	ID
ALYSE PARC AUT	ALYSE_PARC_AUTO
上午:下午	上午:下午
安曼度假村	AMAN_RESORTS
AMANHECER	AMANHECER
AMARI	AMARI
AMART 所有运动	AMART_ALL_SPORTS
亚马逊巴西 B.B.Q.BAR	AMAZON_BRAZIL_B.B.Q.BAR
AMBEST	AMBEST
AMC AFRODITA	AMC_AFRODITA
AMC Cookware	AMC_cookware
AMC 剧院	AMC_Theatres
AMCORE 银行	amcore_Bank
阿门银行	AMEN_BANK
AMENTA 牙科诊所	AMENTA_CLÍNICA_DENTAL
美国旅馆酒店	AMERICAN_INN_HOTELS
AMERICANINO	AMERICANINO
AMF 保龄球	amf_Bowling
AMI	AMI
AMKO komerc	amko_komerc
AMOCO FCU	AMOCO_FCU

名称	ID
AMOUR	AMOUR
AMT 咖啡	amt_coffe
AMW	AMW
AMZS	AMZS
ANAC 洗车	anac_Carwash
阿纳卡普里	阿纳卡普里
ANAT	ANAT
ANCAP	ANCAP
安迪 OnCall	ANDY_ OnCall
ANECA FCU	ANECA_FCU
动物	动物
ANLUSUR	ANLUSUR
安娜威尔第	ANNA_VERDI
安妮公主	安妮_公主
ANP	ANP
ANSON'S	ANSON'S
安塔尔斯	安塔尔斯
ANTONVENETA GRUPPO MONTEPASCH	ANTONVENETA_GRUPPO_MONTEPASCHI
ANTTILA	ANTTILA
ANWB	ANWB

名称	ID
任何任何	ANY_ANY
澳新银行	澳新银行
青木	青木
AP	AP
AP	AP_
APA 联赛	APA_联赛
APANAGE	APANAGE
分开地	分开地
APCO	APCO
APCO 员工 CU	apco_employees_cu
APCOA 停车	APCOA_PARKING
亚太经合组织	亚太经合组织
APEXIM AB	APEXIM_AB
APOTEK	APOTEK
APOTIK	APOTIK
Informática 应用程序	app_Informática
苹果 2003	APPLE_2003
APPLE203	APPLE203
APSAC	APSAC
APTEKA MEDYCZNA	APTEKA_MEDYCZNA

名称	ID
尼维斯药店	APTEKA_NEVIS
APV	APV
APlus	APlus
AQUAEL 动物园	AQUAEL_ZOO
AQUILES	AQUILES
AR 酒店	AR_Hotel
阿拉伯金融机构	ARAB_FINANCE_HOUSE
咸海	咸海
ARAUJO	ARAUJO
ARB	ARB
ARC	ARC
ARCFOX	ARCFOX
ARCO	ARCO
ARD discount	广告折扣
ARDIS	ARDIS
大的	大的
ARGENTUM	ARGENTUM
阿里斯塔	阿里斯塔
阿里斯塔咖啡	ARISTA_COFFE
ARISTINO	ARISTINO

名称	ID
方舟	方舟
ARKADAS	ARKADAS
阿卡迪亚	阿卡迪亚
阿玛根	阿玛根
ARMAND THIERY	ARMAND_THIERY
阿玛尼	阿玛尼
ARMI	ARMI
ARMINE	ARMINE
ARNIKA	ARNIKA
阿诺德·巴斯克	ARNOLD_BUSCK
AROME	AROME
ARS	ARS
ARS/救援机器人	ARS/rescue_rooter
ARTURO CALLE	ARTURO_CALLE
ARYTON	ARYTON
AS	AS
AS 冒险	as_Adventur
作为国际化的	AS_国际
AS24	AS24
ASA Banka	asa_banka

名称	ID
朝日公园	朝日公园
底座	ASE_CU
ASEPEYO	ASEPEYO
ASETRA	ASETRA
阿什利	阿什利
阿什沃思	阿什沃思
亚洲图书	ASIA_BOOKS
亚洲保险	亚洲保险
ASICS	ASICS
ASISA	ASISA
ASKAR	ASKAR
ASKO	ASKO
asm-mebel'	asm-mebel'
ASN	ASN
ASNA	ASNA
ASSAI Atacadista	assaí_atacadista
ASSOS	ASSOS
阿斯顿·马丁	阿斯顿_马丁
ASTONS	ASTONS
阿斯托里亚	阿斯托里亚

名称	ID
ASTRILUZ	ASTRILUZ
ASoil	ASoil
AT&T	AT&T
AT&T 西南航空公司	at&t_西南航空
ATACADAO	ATACADAO
ATAHOTEL	ATAHOTEL
攻击	攻击
ATALAR	ATALAR
ATB	ATB
ATB 金融	atb_Financial
ATB 市场	ATB 市场
REBUL 工作室	ATELIER_REBUL
ATEbank	ATEbank
ATFBank	ATFBank
ATH	ATH
ATH 酒店	ath_Hoteles
ATHLETA	ATHLETA
ATI 物理疗法	ATI_物理疗法
ATL Mietwagen GeSmbH	atl_mietwagen_gesmbH
亚特兰特	亚特兰特

名称	ID
地图集	地图集
阿特拉斯汽车服务	ATLAS_AUTOSERVICE
阿特拉斯银行	ATLAS_BANKA
自动柜员机保释金	atm_bail_Bonds
大气层	大气层
环礁	环礁
ATOMO	ATOMO
ATS 欧洲大师	ats_Euromaster
ATWOODS 牧场和家居	atwoods_ranch_&_Home
ATWORK 人事服务	atwork_人事_服务
AU 德比	AU_DERBY
爱彼腕表	AUDEMARS_PIGUET
奥迪法玛	奥迪法玛
AUMAN	AUMAN
Aurora	Aurora
极光集团	AURORA_GROUP
吉祥凤凰	吉祥_凤凰
AUST	AUST
AUSupreme	AUSupreme
汽车 49	AUTO_49

名称	ID
汽车装甲	自动盔甲
欧罗巴汽车	汽车_EUROPA
汽车凯利	汽车_KELLY
汽车魔法	自动魔法
AUTO Q	AUTO_Q
AUTOBOY	AUTOBOY
汽车玻璃	汽车玻璃
自动烧烤	自动烧烤
汽车乐园	汽车乐园
汽车服务	汽车服务
AUTOSYS	AUTOSYS
AUXI	AUXI
艾娃	艾娃
头像	头像
AVEVE	AVEVE
AVIA	AVIA
阿维亚酒店	Avia_酒店
AVIA XPress	AVIA_ XPress
AVIA 商店	avia_Shop
中航证券	中航证券

名称	ID
AVIOLAMP	AVIOLAMP
AVIS	AVIS
安飞士预算	avis_Budget
英杰华中粮	AVIVA_COFCO
雅芳	雅芳
雅芳代表	雅芳代表
AVTOVAZBANK	AVTOVAZBANK
AVVA	AVVA
AWG	AWG
AWI	AWI
啊啊好吃的食物	AWU_DELICIOUS_FOOD
AXA	AXA
安盛保险公司	安盛公司_de_Seguros
安盛公平	Axa_equ
AXION	AXION
AXION 商店	Axion_Shop
AXTO	AXTO
AYCAN	AYCAN
阿兹慕酒店	Azimut_Hotels
亚琛银行	Aachener_Bank

名称	ID
Aakash 研究所	Aakash_Institute
Aalt 自动定位 Romandie sa	alt_auto-Location_Romandie_
Aanhuis	Aanhuis
Aargauische Kantonalbank	Aargauische_KantonalBank
亚伦兄弟	亚伦兄弟
Aaron's	Aaron's
Aava	Aava
Ab'bas 华夫饼	ab'bas_Waffle
Aba Piknik	Aba_Piknik
算盘	算盘
阿巴德斯酒店	Abades_Hoteles
Abades Triana	Abades_Triana
Abades Areas	Abades_Areas
阿班卡	阿班卡
网走信用金库	网走信用金库
Abay Bank	Abay_Bank
修道院	修道院
Abbey CU	abbey_cu
雅培在印度	雅培_印度
雅培的冰冻蛋羹	Abbott's_frozen_Custard

名称	ID
Abc Lavpri	Abc_Lavpris
Abdi	Abdi
Abele Optik	Abele_Optik
阿本森	阿本森
Abercrombie & Fitch	Abercrombie_&_Fitch
Abhydaya 合作银行	Abhydaya_合作社
Abhyudaya 合作银行	abhyudaya_co-op_Bank
Aboff's	Aboff's
Abrakadoodle	Abrakadoodle
Abrakebabra	Abrakebabra
Abrasador	Abrasador
Abri CU	abri_cu
Abroun	Abroun
Absolyut Bank	Absolyut_Bank
阿布扎比伊斯兰银行	Abu_Dhabi_Islamic_Bank
Abu Diyab	Abu_Diyab
Abuba	Abuba
Abuelo's	Abuelo's
学院银行	学院_银行
学院体育 + 户外	学院_体育_+_户外

名称	ID
阿卡迪亚 FCU	acadia_fcu
Acanthe	Acanthe
阿卡普尔科墨西哥餐厅	阿卡普尔科墨西哥餐厅
Acasia 熟食咖啡厅	Acasia_熟食咖啡厅
澳大利亚华侨银行	澳大利亚华侨银行
阿克森特旅馆	Accent_Inn
埃森哲	埃森哲
验收保险	验收_保险
存取银行	存取银行
配饰先生	配件_先生。
Accessorissimo	Accessorissimo
配饰	配饰
事故修复中心	事故维修中心
住宿大师	住宿_Guru
雅高	雅高
Accort	Accort
国际会计师	会计师_国际
账户/临时工	账户/临时工
Ace 合作银行	ace_co-op_Bank
王牌硬件	Ace_硬件

名称	ID
王牌停车	王牌停车
阿查特酒店	Achat_Hotels
Achieve CU	achieve_cu_inc
阿喀琉斯配件	阿喀琉斯_配件
Ackermans	Ackermans
Ackermans 居家舒适	Ackermans_Home_Comf
阿克兰斯-格兰杰	阿克兰斯-格兰杰
Acme 市场	acme_Markets
Acme-Savon	Acme-Savon
Acne St	暗疮工作室
阿科代克	阿科代克
Acoola	Acoola
橡子	橡子
阿科斯塔	阿科斯塔
AcquaZero	AcquaZero
操作	操作
动作脚套	动作_Footware
行动教练	行动教练
活跃的格林和罗斯	Active_Green_&_Ross
ActiveCapital 银行	ActiveCapital_银行

名称	ID
ActivoBank	ActivoBank
Acto Film	Acto_Film
实际光学	实际光学
急性炎	急性炎
Acura	Acura
阿库尼亚	阿库尼亚
Acústica Medica	Acustica_Medica
广告车库	Ad_Garage
广告 bilverkstad	ad_bilverkstad
艾达	艾达
Adabank	Adabank
Adac 汽车租赁公司	Adac_Autovermietung
足立清和信用金库	足立清和信用银行
Adairs	Adairs
Adairs Kids	Adairs_Kids
亚当巡演	Adam_Tourin
Adamjee	Adamjee
亚当斯	亚当斯
亚当斯孩子	Adams_Kids
Adana Sofrasi	Adana_Sofrasi

名称	ID
阿达尼压缩天然气	adani_cng
阿达什合作银行	adarsh_Co-operative_Bank
阿达什信贷合作社	Adarsh_Credit_Co_Operative_Society
Addide	Addide
阿迪科银行	Addiko_Bank
亚的斯国际银行	Addis_国际_银行
加法 Elle	附加_Elle
Adecco	Adecco
Adecco 就业服务	Adecco_就业_服务
Adega	Adega
阿德莱德银行	阿德莱德银行
阿德莱德停车场	阿德莱德汽车停车
Adese	Adese
阿德斯拉斯	阿德斯拉斯
Adessa	Adessa
阿迪达斯原创作品	Adidas_Originals
阿迪娜	阿迪娜
阿迪什瓦尔	阿迪什瓦尔
Aditya 汽车备件	Aditya_汽车_备件
Adiamanli Cig Kofteci Erkan Usta	Adiamanli_Cig_Kofteci_Erkan_Usta

名称	ID
Adl	Adl
阿德勒	阿德勒
海军上将	海军上将
海军上将自动机俱乐部	Admiral_Automat_Club
海军上将折扣烟草	Admiral_折扣_烟草
海军上将石油公司	海军上将_石油公司
Admore	Admore
Adnoc Oasis	adnoc_Oasis
Adobo 连接	Adobo 连接
阿多尼斯	阿多尼斯
Adopen	Adopen
亚德里亚银行	Adria_Bank
成人世界	成人世界
先进美国	先进_美国
高级汽车零部件	高级汽车零件
高级财务 FCU	高级_财务_FCU
前进电信	高级_电信
高级汽车	高级_汽车
高级测试中心	高级测试中心
优势 FCU	优势_FCU

名称	ID
优势租车	优势_租车
Advanz FCU	Advanz_FCU
Adventist Health	基督复临安息日会健康系统
冒险 CU	adventure_cu
冒险领地	冒险_领地
忠告	忠告
Adyar Ananda Bhavan	Adyar_Ananda_Bhavan
Adyar 面包店	Adyar_Bakery
Adyar Xerox	Adyar_Xerox
爱琴海	爱琴海
Aegis 汽车加油	Aegis_Auto_Gas
爱敬百货	爱敬百货店
永旺花旗超市	Aeon_Citimart
永旺购物中心	永旺购物中心
永旺购物中心	永旺购物中心
永旺超级中心	永旺超级中心
航空	航空
Aero Shake	Aero_Shak
墨西哥航空公司	墨西哥航空公司
Aeropstale	Aeropstale

名称	ID
气软	气软
Aerotek	Aerotek
伊索	伊索
附属汽车租赁	加盟租车
Affin bank	Affin_Bank
新泽西州的 Affinity FCU	Affinity_fcu_of_nj
苦难	苦难
经济实惠的假牙	负担得起的假牙
经济实惠的点火联锁	经济实惠_Ignition_Interlock
Afirme	Afirme
Aflac	Aflac
AfricOil	AfricOil
非洲银行	African_Bank
非洲骄傲酒店	African_Pride_Hotels
Afriland 第一银行	Afriland_First_Bank
Afriquia	Afriquia
非洲食品	非洲食品
Afrodita	Afrodita
Afroditi	Afroditi
Afta 市场	Afta_Marke

名称	ID
善后服务	事后服务
下午茶	下午茶
雅富顿健康与健身器材	雅富顿健康与健身器材
Afu	Afu
标签:S	标签:S
Agabang	Agabang
Agafarma	Agafarma
克服一切困难	Against_All_Odds
Agarie Megane	Agarie_Megane
Agata Meble	Agata_Meble
英国时代	年龄_英国
中介租车	中介_租赁_汽车
Agera Hompo	Agera_Hompo
Agha Khan Uni。医院实验室	Agha_Khan_Uni。 _医院_实验室
Agip	Agip
哎呀	哎呀
艾格尼丝。B	艾格尼丝。 _B
Agon	Agon
Agora	Agora
Agora 便利店	Agora_便利店

名称	ID
阿戈斯蒂尼	阿戈斯蒂尼
阿格拉姆银行	Agram_Banka
AgriBank	AgriBank
中国农业银行	中国农业银行
农业发展银行	农业发展银行
农业合作社	农业_合作社
Agrimarkt	Agrimarkt
农业集市	农业集市
Agrobanka	Agrobanka
Agrola	Agrola
阿格罗兰	阿格罗兰
Agrosoyuz	Agrosoyuz
阿贡租车	Agung_Rent_a_car
Agway	Agway
Ah Cheng 叻沙	Ah_Cheng_Laksa
aHeadHR	aHeadHR
艾亨租赁	Ahern_Rentals
艾哈迈达巴德地区合作银行	艾哈迈达巴德地区合作银行
Ahold	Ahold
AhorraMas	AhorraMas

名称	ID
Ahs-Park 酒店	AHS-Park_Hotel
Ahsen Pastaneleri	Ahsen_Pastaneleri
Ahumada	Ahumada
Ai Megane	Ai_Megane
艾尚	Ai_Shang
爱碧宠物医院	爱必宠物医院
Aibe	Aibe
爱知县 Gifu-no-Kojin志道赛门宿	Aichi_gifu-no-kojin_Shido_semmon_juku
爱知信用金库	爱知信用银行
艾达	艾达
爱果哲养生火锅	Aiguozhe_Yangusheng_Hopot
艾华	艾华
爱建证券	爱建证券_证券
瞄准邮件中心	Aim_Mail_Centers
Aima	Aima
艾尼西	艾尼西
航空学院 FCU	air_academy_FCU
航空银行	Air_Bank
空军 FCU	空军_force_FCU
法国航空	法国航空公司

名称	ID
印度航空	印度航空
液化空气美国公司	Air_Liquide_美国
新西兰航空	新西兰航空公司
空气产品公司	空气产品
AirTran 航空公司	AirTran_航空公司
Aire Serv	Aire_Serv
Aire-Master	Aire-Master
Airfel	Airfel
Airgas	Airgas
Airotel	Airotel
机场旅馆	机场旅馆
Airtel	Airtel
Airtel 支付银行	Airtel_Payments_Bank
Aisect	Aisect
Aisekiya	Aisekiya
Aiways	Aiways
艾夏艾夏	Aixia_Aixia
Aizkraukles Banka	Aizkraukles_Banka
Aji ICHIBAN	aji_ichiban
Ajilon	Ajilon

名称	ID
Ajilon 专业人员配置	Ajilon _专业人员_人员配备
Ajino Mingei	Ajino_Mingei
Ajino Tokeidai	Ajino_Tokeidai
阿杰马尔	阿杰马尔
阿治曼银行	阿治曼银行
阿治曼市中心有限责任公司	Ajman_city_centre_LLC
Ak 广场	Ak_Plaz
Akachan 百货店水谷店	Akachan_百货商店_Mizutani
Akademika	Akademika
赤坂亭	赤坂亭
阿克班克	阿克班克
Akdeniz Dondurma	Akdeniz_Dondurma
Akgüller	Akgüller
Akher Saa	Akher_Saa
Aki	Aki
秋叶原	秋叶原
Akola Dist 中央合作银行	Akola_dist_central_Co-op_Bank
Akordbank	Akordbank
Akpet	Akpet
Akpet 市场	Akpet_Marke

名称	ID
阿克兰拉尔超市	Akranlar_Sump
阿克伦	阿克伦
Aks	Aks
阿克萨发电机	Aksa_Enerator
Aktia	Aktia
AktieSport	AktieSport
Aktivoptik	Aktivoptik
Aktsept	Aktsept
Aktyv-Bank	Aktyv-Bank
Akvarel'	Akvarel'
Akyurt	Akyurt
Al Abraaj	Al_Abraaj
Al Ahali	Al_Ahali
科威特 Al Ahli 银行	al_ahli_Bank_of_科威特银行
Al Akhawain	Al_Akhawain
Al Akhdar 银行	Al_Akhdar_Bank
Al Amana 小额信贷	Al_Amana_Micro_Finance
Al Amanah Juices	Al_Amanah_Juices
Al Andalus Tapas	Al_Andalus_Tapas
阿尔安萨里交易所	Al_Ansari_Exchange

名称	ID
Al Baik	Al_Baik
Al Baraka 伊斯兰银行	Al_Baraka_Islamic_Bank
Al Barid 银行	Al_Barid_Bank
Al Dawaa 药房	Al_dawaa_药房
Al Haramain 香水	Al_Haramain_Perfumes
Al Jabr 洗衣和干洗服务	Al_Jabr_Laundry_&_Dry_Clanging
半岛电视台香水	Al_Jazeera_Perfumes
Al Khidmat	Al_Khidmat
Al Maha	Al_Maha
阿尔马纳拉药房	Al_Manara_Pharmacy
Al Mandoos	Al_Mandoos
Al Masraf	Al_Masraf
Al Maya 超市	Al_Maya_Superm
Al Meera	Al_Meera
Al Muzaini 交易所	Al_Muzaini_Exchange
阿尔纳迪药房	Al_Nahdi_Pharmacy
Al Nahhas 药房	Al_Nahhas_Pharmacy
清洁工艾尔·菲利普斯	al_phillips_the_Cleaner
Al Qurayat 金色自助餐厅	al_qurayat_Golden_Cafetery
艾尔拉哈海滩	Al_Raha_Beach

名称	ID
Al Rajhi 银行	Al_Rajhi_Bank
Al Rehaili	Al_Rehaili
Al Rehily	Al_Rehily
Al Rostamani 国际交易所	al_rostamani_国际交易所
Al Safi	Al_Safi
Al Saggaf 药房	Al_Saggaf_Pharmacy
Al Shifa	Al_Shifa
Al Tazaj	Al_Tazaj
埃及 Al Watany 银行	al_watany_Bank_of_Egypt
阿尔博格实验室	al_borg_Laborages
Al karam 工作室	al_karam_Studio
Al's 正装	al's_formal_Wear
al'yans	al'yans
Al-Anon	Al-Anon
Al-Hayat 药房	Al-Hayat_药房
Al-Maha Opticals	Al-Maha_Opticals
AlAndalus-Cineápolis	AlAndalus-Cineápolis
AlBaraka	AlBaraka
AlaTrust CU	AlaTrust_CU
阿拉巴马州农业信贷	Alabama_Ag_Credit

名称	ID
阿拉巴马州 CU	Alabama_cu
阿拉巴马州中部 CU	Alabama_central_cu
阿拉巴马州全国广播	阿拉巴马州_NBC
阿拉巴马州教师大学	Alabama_teachers_cu
Alacati Muhallebicisi	Alacati_Muhallebicisi
阿拉丁保释金	Aladdin_Bail_Bonds
阿拉丁餐厅	Aladdin's Eatery
阿拉多	阿拉多
阿拉摩	阿拉摩
Alarko	Alarko
阿拉善农村商业银行	阿拉山_农村_商业银行
阿拉斯加美国 FCU	Alaska_usa_FCU
Alatan Tur	Alatan_Tur
阿拉瓦尔银行	alawal_bank
Alba Takarekszövetkezet	Alba_Takarekszövetkezet
Albaraka Turk	Albaraka_Turk
阿尔伯特	阿尔伯特
阿尔伯特·海因	Albert_Hein
阿尔伯特和沃尔特	艾伯特和沃尔特
阿尔贝托·奥库利斯塔	Alberto_Oculista

名称	ID
阿尔贝托 VO5	阿尔贝托_ VO5
阿尔贝托的披萨	Alberto's Pizz
艾伯森	艾伯森
艾伯森市场	艾伯森市场
艾伯森市场街	艾伯森市场街
艾伯森药房	艾伯森药房
艾伯森-奥斯科	艾伯森-奥斯科
Albertsons-Savon	Albertsons-Savon
阿尔比	阿尔比
阿尔比斯	阿尔比斯
Albrekts Guld	Albrekts_Gold
Alcafé	Alcafé
阿尔坎波	阿尔坎波
阿尔卡特朗讯	阿尔卡特朗讯
Alcoa Tenn FCU	Alcoa_ten_FCU
匿名酗酒者	酗酒者_匿名
Alcon Viatges	Alcon_Viatges
Aldo Conti	Aldo_Conti
Aldo 餐厅	Aldo_restauran
Aldora	Aldora

名称	ID
啤酒	啤酒
随意的	随意的
Alef 书店	Alef_bookstore
阿列克谢夫斯基伊	阿列克谢夫斯基伊
阿伦比克制药	Alembic_制药
阿连特茹	阿连特茹
Alepa	Alepa
Aleris	Aleris
亚历克斯健身	亚历克斯_健身
AlexBank	AlexBank
亚历山大	亚历山大
亚历山大 McQueen	亚历山大_ McQueen
亚历山德拉	亚历山德拉
亚历山大酒店	亚历山大_酒店
Alexela	Alexela
Alexi Andriotti	Alexi_Andriotti
Alexon	Alexon
Alfa Star	Alfa_Star
阿尔法快车	Alfa_Expres
Alfa-Bank	Alfa-Bank

名称	ID
AlfaStrakhovanie	AlfaStrakhovanie
Alfamidi	Alfamidi
Alfapen	Alfapen
Alfemo	Alfemo
Alfil	Alfil
Alfio Raldo	Alfio_Raldo
算法 Profil Profil Mozaik	算法_配置文件_Mozaik
Algy	Algy
阿尔及利亚电信	Algérie_Telecom
Alhambra CU	alhambra_cu
阿里和阿里药房	Ali_&_Ali_Pharmacy
阿里巴巴	阿里_巴巴
阿里图片社	Ali_Image
Alians	Alians
爱丽丝·德利斯	Alice_Delice
Alimenta	Alimenta
Alimex	Alimex
Alinea	Alinea
Aliprox	Aliprox
Alisuper	Alisuper

名称	ID
哥伦比亚阿丽塔斯	Alitas_Colombianas
Alittle-Tea	Alittle-Tea
活着的 CU	Alive_cu
Aliya	Aliya
Alkaram	Alkaram
Alko	Alko
所有酒吧一号	All_Bar_One
所有咖啡厅	All_Cafe
一整天	All_Day
All My Shine	All_My_Shine
诸圣徒	所有圣徒
所有季节	所有季节
全季欧洲	全季_欧洲
All Tune & Lube	All_Tune_&_Lube
AllSaints	AllSaints
阿拉哈巴德区合作银行	Allahabad_dist_co-op_Bank
Alldays	Alldays
Allegacy FCU	Allegacy_fcu
Allegius FCU	Allegius_fcu
Allego	Allego

名称	ID
Allegra	Allegra
Allegra 打印与成像	Allegra_Print_&_Imaging
Allegro 儿科	Allegro_Pediatrics
Alleira	Alleira
艾伦俱乐部	Allen_Club
艾伦泰特房地产经纪人	Allen_Tate_Realtors
Allguth	Allguth
联盟与莱斯特	Alliance_&_Leister
联盟保证	联盟_保证
联盟银行	Alliance_Bank
联盟 CU	Alliance_cu
佛罗里达联盟 CU	Alliance_cu_of_florida
光学联盟	联盟_Optikk
联盟药房	联盟_药房
Alliant 信用合作社	Alliant_Credit_Uni
安联	安联
Allianz AYUDHYA	Allianz_Ayudhya
安联银行	安联银行
保加利亚安联银行	安联银行_保加利亚
安联会计师事务所	Allianz_C.P.

名称	ID
安联 Tiriac Asigurari	Allianz_Tiriac_Asigurari
联合银行	Allied_Bank
联合住房抵押贷款资本公司	Allied_Home_Mortgage_Capital_Corp
爱尔兰联合银行	Allied_Irish_Bank
联合石油公司	Allied_Petrolem
联合学校	Allied_School
Allied 轮胎与服务	Allied_Tires_&_Service
盟军货车线	Allied_Van_Lines
Allina	Allina
Allo	Allo
所有手机	所有手机
万能点	万能点
Allstar	Allstar
Allstate 保险	Allstate_保险
Allsup's	Allsup's
Alltel	Alltel
Alltru CU	alltru_cu
盟友时尚	盟友_时尚
阿尔玛	阿尔玛
Alma Gysyszertár	Alma_Gyögyszertár

名称	ID
Almeida Viajes	Almeida_Viajes
Almira 租车服务	Almira_汽车租赁_服务
阿尔莫拉城市合作银行	Almora_urban_co-operative_Bank
Alnatura	Alnatura
Alnatura De	Alnatura_De
Aloc-Cars	Aloc-Cars
芦荟	芦荟
在高处	在高处
阿罗哈	阿罗哈
阿隆	阿隆
Alonit	Alonit
Alouette	Alouette
Alpari	Alpari
Alpark	Alpark
阿尔卑斯山屋	Alpen_House
Alpet	Alpet
Alpet 市场	Alpet_Market
Alpha	Alpha
Alpha 复制诺基亚商店	Alpha_Copy_Nokia_Stores
阿尔法生态	Alpha_Egoci

名称	ID
阿尔法药业	Alpha_Pharm
AlphaGraphics	AlphaGraphics
AlphaMega	AlphaMega
高山	高山
阿尔卑斯银行	Alpine_Bank
Alpro	Alpro
阿尔卑斯中央信用金库	Alps_Chuo_Shinkin_Bank
Alquimobil	Alquimobil
Alrasasi Perfumes Co.	Alrasasi_Perfumes_Co.
Alser 计算机	ALSER_计算机
Alsi	Alsi
Alta 便利	Alta_Convanc
Altae	Altae
Altech Netstar	Altech_Netstar
改变它	Alter_It
Altex	Altex
Altinbas	Altinbas
Altinyildiz 经典	Altinyildiz_Classics
Altra 联邦信用合作社	Altra_Federal_Credit_Union
Altracolor	Altracolor

名称	ID
Altteul 加油站	Altteul_Gas Station
Altun	Altun
Altunbilekler	Altunbilekler
Alvina	Alvina
alza.cz	alza.cz
上午 9 点到晚上 9 点	上午 9 点到晚上 9 点
AmBank	AmBank
AmSouth 银行	AmSouth_银行
阿玛迪斯	阿玛迪斯
尼崎信用金库	尼崎信用金库
Amagerbanken	Amagerbanken
合并后的银行	合并后的银行
Amandau	Amandau
Amaral Costa Medicina Diagnóstica	Amaral_Costa_Medicina_Diagnóstica
阿玛兰特酒店	Amarante_Hotels
阿马里洛	阿马里洛
Amarine	Amarine
Amato's	Amato's
Amavita Apotheke	Amavita_Apotheke
神奇的睫毛工作室	Amazing_Lash_Studio

名称	ID
神奇的东方	神奇_东方
亚马逊 4 星	亚马逊_4 星
Amazon Books	亚马逊图书
亚马逊 Fresh	亚马逊_Fresh
Amazon Go	Amazon_Go
Amazon Hub	亚马逊_Hub
Amazon Pop Up	Amazon_Pop_Up
Amazon Surf	亚马逊_Surf
亚马逊	亚马逊
Ambar	Ambar
我爱的 Ambar	Ambar_I_LOVE
Ambardí	Ambardí
大使	大使
琥珀色	琥珀色
Ambientes 放松一下	Ambientes_Cill_Out
Ambit Rsm Bank	Ambit_Rsm_Bank
安博教育	Ambow_教育
Amcal Chemist	Amcal_Chemist
阿米吉银行	Amegy_Bank
阿米莉亚·阿兰	Amelia_Aran

名称	ID
Amerada HESS	Amerada_Hess
AmeriCU 信用合作社	Americu_Credit_Un
AmeriGas	AmeriGas
AmeriGas 丙烷	AmeriGas_丙烷
AmeriHost 旅馆	AmeriHost_旅馆
AmeriSpec	AmeriSpec
AmeriStop	AmeriStop
AmericInn	AmericInn
AmericInn 山林小屋和套房	AmericInn_Lodge_&_Suites
AmericInn 汽车旅馆	AmericInn_汽车旅馆
AmericInn 汽车旅馆和套房	AmericInn_汽车旅馆_&_套房
美式早餐	美国_早餐
美国第一信用合作社	America_First_Credit
今日美国	今日美国
美国最佳隐形眼镜和眼镜	美国_最佳_联系人_&_眼镜
美国的汽车市场	America's Car_mart
美国的 Taco Shop	America's_taco_shop
美国的轮胎	America_tire
美国航空信用合作社	美国航空_Credit_Union
美国服装	美国_服装

名称	ID
美国沥青密封涂层	美国_沥青_密封涂层
美国投资银行	美国投资银行
美国蓝鹿	美国_Bluedeer
美国癌症协会	美国_癌症_社会
美国汽车保养中心	美国汽车保养中心
美国洗车	American_Car_Wash
美式熟食店	American_Deli
美国毒品	美国_毒品
American Eagle 服装店	American_Eagle_Outfitters
美国运通	美国运通
美国公平信贷	美国_Fair_Credit
美国家庭护理	美国家庭护理
美国家庭保险	美国家庭保险
美国普通保险	美国_普通_保险
美国通用人寿公司	美国_General_Life_Company
美国问候	美国_问候
美国心脏协会	美国心脏协会
美国住房贷款	美国_房屋_贷款
美国人 HomePatient	美国人_ HomePatient
美国激光中心	American_Laser_Centers

名称	ID
美国泄漏检测	美国泄漏检测
美国退伍军人协会	American_Legion
美国医疗回应	美国_医疗_回应
美国动量银行	美国_Momentum_Bank
美式指甲	美国_Nails
美国国民银行	美国国民银行
美国国民保险	美国_国民_保险
美国石油	American_Oil
美国合作伙伴 FCU	美国_Partners_FCU
美国邮政工人工会	American_Postal_Workers_Union
美国红球	American_Red_Ball
美国红十字会	American_Red_Cross
美国储蓄银行	American_Savings_Bank
美国特色家具	American_Signature_家具
美国西南加州大学	美国_西南_CU
美国快速印刷中心	美国_Speedy_Printing_Centers
American Spirit	American_spirit_FCU
美国标准	美国_标准
美国瑞士人	美国人_瑞士人
美国轮胎仓库	American_Tire_Depot

名称	ID
美国游客	American_Tourister
美国塔	American_Tower
美国联队 FCU	American_United_FCU
美洲最具价值旅馆	Americas_Best_Value_Inn
美洲 CU	Americas_cu
美洲基督教大学	americas_christian_cu
美国企业金融	AmeriPrise_Finan
艾姆斯真脾气	Ames_True_Temper
Amica 保险	Amica_保险
Amichi	Amichi
Amici	Amici
Amico	Amico
AmicoBlu	AmicoBlu
Amigo	Amigo
Amigo 24	Amigo_24
Amigo 早午餐	朋友_Brunch
Amigos	Amigos
Amigão Supermercados	Amigão_Supermercados
阿明	阿明
埃米尔	埃米尔

名称	ID
Amirtha 冰淇淋	amirtha_ice_cream
Ammas 糕点	Ammas_Paster
Amoco	Amoco
Amoco Food Shop	Amoco_Food_Shop
Amor aos Pedaços	amor_aos_pedaços
Amores Perros	Amores_Perros
阿莫里诺	阿莫里诺
Amoud	Amoud
Ampco 系统停车	安普科_系统_停车
Greaves 的 Ampere	ampere_by_Greaves
Amplifon	Amplifon
Ampol	Ampol
阿姆雷酒店	Amrey_酒店
阿姆拉特酒店	Amrát_Hotels
阿姆拉斯酒店	Amráth_Hotels
Amstar	Amstar
美国全国铁路客运公司	美国全国铁路客运公司
安利	安利
安利全球独立企业主	Amway_Global_独立企业_所有者
América Salud	América_Salud

名称	ID
安福	An_Phuoc
AnJi	AnJi
安娜·索萨	Ana_Sousa
森蚺	森蚺
Anadolu Kuruyemis	Anadolu_Kuruyemis
Anadolu Seyehat	Anadolu_seyehat
Anadolubank	Anadolubank
Anago	Anago
Anakku	Anakku
临床实验室分析	Analisisis_Laboratório_Clínico
Anamika Khanna	Anamika_Khanna
阿南宠物医院	Anan_Pet_Hospital
安纳塔拉	安纳塔拉
阿纳斯塔西娅·比佛利山庄	Anastasia Beverly_Hills
解剖学	解剖学
安卡纳拉	安卡纳拉
Anchor (锚点)	Anchor (锚点)
AnchorBank	AnchorBank
AnchorBlue	AnchorBlue
安达雷拉	安达雷拉

名称	ID
安达兹	安达兹
Andelskassen	Andelskassen
安德森兄弟银行	安德森兄弟银行
安德森	安德森
安得拉邦 Grameena Vikas 银行	Andhra_Pradesh_Grameena_Vikas_Bank
安得拉邦州立合作银行	Andhra_pradesh_state_Co-op_Bank
Andhra Pragathi Grameena 银行	Andhra_Pragathi_Grameena_Bank
Andiamo	Andiamo
Andiccio24	Andiccio24
Andigo CU 消费者部 CCU	andigo_cu_div_of_consumers_CCU
安迪拉纳	安迪拉纳
安藤证券	安东证券
Andok's	Andok's
安德烈迪斯民宿	Andreadis_Homestores
安德烈亚尼	安德烈亚尼
安德鲁斯 FCU	Andrews_FCU
安德罗尼科的	安德罗尼科的
Ane	Ane
安妮大酒店	Ane_Grand_Hotel
安妮酒店	Ane_Hotel

名称	ID
Ane Inn	Ane_Inn
Anettes 三明治	Anettes_Sandwich
Ange	Ange
天使牙科	天使牙科
天使火锅	Angel_Hot_Pot
安琪儿妇幼医院	天使_妇幼医院
我们心中的天使	Angel_in_US
安杰洛尼	安杰洛尼
盎格鲁	盎格鲁
盎格鲁爱尔兰银行	Anglo_Irish_Bank
盎格鲁罗马尼亚银行	盎格鲁_罗马尼亚_银行
Angstrem	Angstrem
安格斯	安格斯
Angus & Coote	Angus_&_Coote
安格斯和罗伯逊	Angus_&_Robertson
Anh Ngu AMA	anh_ngu_ama
安徽汽车服务	安徽_汽车服务
安徽长丰科园镇银行	安徽_长丰_科源镇_银行
安徽郎溪农村商业银行	安徽_郎溪_农村_商业银行
安徽农村信用社	安徽_农村信用社

名称	ID
安徽舒城农村商业银行	安徽_舒城_农村通信_银行
安徽天长农村商业银行	安徽_天长_农村通信_银行
Ani Sup Utara	Ani_Sup_Utara
Aniks	Aniks
Animalis	Animalis
动画	动画
Anita papelería	Anita_papeleria
Anjappar	Anjappar
Anjuke	Anjuke
安凯	安凯
Anker	Anker
Anko	Anko
安·哈维	安_哈维
安·萨默斯	Ann_Summers
安·泰勒	安_泰勒
安·泰勒工厂	Ann_Taylor_Factory
安泰勒阁楼	Ann_Taylor_Loft
安的面包店	Ann's Bakery
安娜	安娜
安娜·里斯卡	Anna_Riska

名称	ID
安娜苏	Anna_Sui
安娜的亚麻布	Anna's_Linens
Annamalai Foods	Annamalai_Foods
安妮方丹	Anne_Fontaine
Annie's	Annie's
另一家破蛋咖啡厅	Another_broken_Egg_Café
安乐亭	安乐亭
其他装饰	Anre_Decoracion
Anten Do	Anten_Do
Anteprima	Anteprima
安东尼的餐厅	Anthony's 餐厅
人类学	人类学
昂蒂布斯-停车	昂蒂布斯-停车
Antik Batik	Antik_Batik
Antilopa	Antilopa
Antol	Antol
安东体育	Anton_Sport
安东内尔	安东内尔
安图农村商业银行	Antu_Rural_商业银行
现在可以进行任何实验室测试	Any_Lab_Test_Now

名称	ID
随时健身	任何时候_健身
Anytyme	Anytyme
Anywide	Anywide
奥德龙	奥德龙
青森信用金库	Aoi-Mori_shinkin_Bank
青木信用金库	Aoki_Shinkin_Bank
青木超级	Aoki_Super
青森县信用合作社	青森县_信用合作社
阿帕奇披萨	Apache_Pizza
停车	停车
停车	停车
Apegia	Apegia
Apel'sin	Apel'sin
Apex 租车	Apex_租车
Apex 租赁	Apex_租赁
Aplast	Aplast
Apna Bank	Apna_Bank
阿普纳集市	Apna_Bazaar
Apna Sahakari Bank	Apna_Sahakari_Bank
Apo Bank	Apo_Bank

名称	ID
阿波罗	阿波罗
阿波罗医院	阿波罗医院
阿波罗酒店	Apollo_Hotel
阿波罗光学	Apollo_Optik
阿波罗药房	阿波罗_药房
阿波罗轮胎	apollo_tyre
阿波罗尼翁	阿波罗尼翁
阿波罗尼亚	阿波罗尼亚
Apotek 1	Apotek_1
Apotek GENERI	apotek_Generik
Apotek Hjertat	Apotek_Hjertat
Apotek K-24	Apotek_K-24
Apoteka Beograd	Apoteka_Beograd
阿波特卡诺维萨德	Apoteka_Novi_Sad
Apoteket	Apoteket
Appart'City	Appart'City
阿彭策尔州立银行	Appenzeller_KantonalBank
Apple 汽车玻璃	Apple Auto_Glass
苹果金融控股（苹果银行）	Apple_Financial_Holdings_（Apple_Bank）
苹果商店	Apple_Store

名称	ID
AppleOne	AppleOne
Applebee 的	Applebee 的
苹果绿	苹果绿
应用工业技术	应用_工业_技术
应用研究助理	Applied_Research_A
Aprecio	Aprecio
Aprel'	Aprel'
Apria 医疗保健	Apria_医疗保健
四月	四月
四月英语	四月_英语
Apsara 冰淇淋	Apsara_Ice_Creams
阿普西斯酒店	Apsis_Hotels
Aptech	Aptech
Aptech 学习	Aptech_Lear
Apteka Dobroho Dina	Apteka_dobroho_Dnia
Apteka Nizkikh Tsen	Apteka_nizkikh_Tsen
Apteka Nyzkykh Tsin	Apteka_nyzkykh_Tsin
Apteka Optovykh Tsen	Apteka_Optovykh_Tsen
apteka Plus	Apteka_Plus
Apteka ot sklada	apteka_ot_sklada

名称	ID
Apteka otlichnykh tsen	apteka_otlichnykh_tsen
Apteka zdrowie	apteka_zdrowie
Aptekar'	Aptekar'
Aptekar'Evenius	Aptekar'_Evenius
Apteki Kubani	Apteki_Kubani
Apteki Kuzbassa	Apteki_Kuzbassa
Apteki Stolitsy	Apteki_Stolitsy
Apteki Stylowe	Apteki_Stylowe
Aqua Hotel & Ohla 精品酒店	Aqua_Hotel_&_Ohla_精品酒店
爪哇水族馆	Aqu_Java
Aquadis Loisirs	Aquadis_Loisirs
Aquanaut	Aquanaut
Aquascutum	Aquascutum
Aquila	Aquila
Aqui 是的	Aquí_Sí
阿拉伯非洲国际银行	Arab_African_国际银行
阿拉伯银行	Arab_Bank
阿拉伯银行公司	阿拉伯银行股份有限公司
阿拉伯投资银行	阿拉伯_投资_银行
阿拉伯以色列银行	阿拉伯_以色列_银行

名称	ID
阿拉伯约旦投资银行	阿拉伯_约旦_投资_银行
阿拉伯突尼斯银行	Arab_Tunisian_Bank
阿拉伯世界餐厅	阿拉伯世界餐厅
ArabellaSheraton	ArabellaSheraton
阿拉伯乌德	Arabian_Oud
阿拉比卡	阿拉比卡
咸脉冲	Aral_Pulse
Aramark	Aramark
Aramark 制服服务	Aramark_统一服务
阿兰斯	阿兰斯
Arap Turk Bankasi	Arap_Turk_Bankasi
Arapahoe CU	arapahoe_cu
Aras Kargo	Aras_Kargo
Arbejdernes Landbank	Arbejdernes_Landsbank
乔木毒品	Arbor_Drug
Arboretto	Arboretto
Arby's	Arby's
Arc'teryx	Arc'teryx
Arcaffe	Arcaffe
Arcantis	Arcantis

名称	ID
拱门寻呼	Arch_Paging
Archadeck	Archadeck
Archie's Trattoria	Archie's Trattoria
Archies	Archies
阿基米德	阿基米德
Arco Iris Papelería	Arco_Iris_Papelería
Arcoa	Arcoa
北极圈	北极圈
Ardaf	Ardaf
Arden B	Arden_B
Ardene	Ardene
Ardent 信用合作社	Ardent_Credit_Un
马拉多斯特竞技场	Arena_Mladost
西部竞技场	西竞技场
Aresbank	Aresbank
Aretha	Aretha
阿雷佐	阿雷佐
Argent FCU	argent_fcu
Argenta	Argenta
Argos	Argos

名称	ID
Argos 石油	Argos_Oil
Argos 加油站	Argos_加油站
阿盖尔精品酒店	阿盖尔精品酒店
阿盖尔大酒店	阿盖尔大酒店
阿盖尔酒店	Argyle_Hotel
阿盖尔度假村	阿盖尔度假村
Ari Aku	Ari_Aku
阿里亚德尼	阿里亚德尼
阿里亚尼	阿里亚尼
Arifoglu	Arifoglu
Arigaen 高尔夫	Arigaen_Golf
Arion banki	Arion_Banki
阿里郎	阿里郎
Aris 专卖店	Aris Specialty_Store
Arisa	Arisa
Aristocrazy	Aristocrazy
阿里斯顿	阿里斯顿
Aritzia	Aritzia
亚利桑那联邦大学	Arizona_Federal_cu
亚利桑那披萨公司	亚利桑那州_披萨_公司

名称	ID
Arkada	Arkada
Arko	Arko
阿玛	阿玛
Armakit	Armakit
Arman	Arman
Armand Basi	Armand_Basi
阿玛尼系列	Armani_Collezioni
阿玛尼交易所	阿玛尼交易所
阿玛尼牛仔裤	Armani_牛仔裤
阿玛尼少年	阿玛尼_少年
Armazem Paraiba	Armazem_Paraiba
Armbruster 后店	Armbruster_backShop
Armorine	Armorine
阿姆斯特朗协会 FCU	armstrong_associates_FCU
陆军和空军交换服务	陆军和空军交换服务
陆军国民警卫队	陆军_国民警卫队
陆军海军汉堡 + 墨西哥卷饼	Army_Navy_Burger_+_Burrito
阿诺德·帕尔默	Arnold_Palmer
阿诺兹	阿诺兹
香气	香气

名称	ID
芳香面包店	芳香面包店
芳香咖啡厅	Aroma_Ca
Aromatny mir	aromatniy_mir
Arow	Arow
Arpin Van Lines	Arpin_van_Lines
Arquiarmarios	Arquiarmarios
Arrachera House	Arrachera_House
Arrail	Arrail
Arreglos & Co	Arreglos_&_Co
Arrow	Arrow
弓箭灭虫者	Arrow_Exterminators
阿森纳杯	Arsenal_cu
Arsis	Arsis
Art Mozi	Art_Mozi
剃须艺术	剃须的艺术
Art'Otel	Art'Otel
ArtEspanol	ArtEspanol
Arteli	Arteli
Arteli 快车	Arteli_Exp
Arteli mars	arteli_mals

名称	ID
Artem-Bank	Artem-Bank
阿耳忒弥斯	阿耳忒弥斯
Artesania Sin Fronteras	Artesania_Sin_Fronteras
Artesanos Camiseros	Artesanos_Camiseros
食品工艺	Artesanía_de_la_Alimentacion
Artesiete	Artesiete
全日空艺术酒店	艺术酒店_ANA
Arthur	Arthur
Arthur Treacher	Arthur_Treacher's
亚瑟和佐伊	Arthur_et_Zoé
巴黎文章	巴黎文章
文章	文章
Artis	Artis
阿鲁克	阿鲁克
阿鲁拉酒店及度假村	Arula_酒店及度假村
Arun Icecreams	Arun_Icecreams
Arvanitidis	Arvanitidis
Arvest银行	Arvest_Bank
Arya Vaidya Sala	Arya_Vaidya_Sala
Aryavart 银行	Aryavart_Bank

名称	ID
Arcelik	Arcelik
如我所知	As_Know_As
As Kuruyemis	As_Kuruyemis
朝日信用金库	朝日信用金库
旭川信用金库	旭川_信用金库
Asb	Asb
Ascend 系列	上升系列
Ascension He	升天_生命值
Ascent CU	ascent_cu
Ascentra CU	Ascentra_cu
雅诗阁服务公寓	雅诗阁公寓酒店
阿斯达	阿斯达
Asesor 工具	Asesor_Tool
Asestravel	Asestravel
Ashcom 德国补充装	Ashcom_German_Refill
阿什维尔储蓄银行	Asheville_Savings_Bank
足利小山信用金库	足利_小山_信用金库
Ashion	Ashion
亚什兰	亚什兰
阿什利家具 HomeStore	Ashley_Furniture_ HomeStore

名称	ID
阿什利·斯	阿什利_斯图尔特
Ashok Leyland	Ashok_Leyand
亚洲信贷银行	亚洲_信贷银行
亚洲绿色发展银行	亚洲绿色发展银行
亚洲英雄	《亚洲英雄》
亚洲汽车	亚洲_汽车
亚太电信	亚太地区_电信
亚洲桌子	亚洲_桌子
亚洲信托银行	亚洲信托银行
亚洲联合银行	亚洲_联合银行
亚太财产保险	亚太地区_房地产_保险
亚洲1	亚洲1
亚洲电影院	亚洲电影院
亚洲的	亚洲的
阿西罗姆	阿西罗姆
提问	提问
阿斯卡里银行	Askari_Bank
Askent	Askent
Askona	Askona
Asli Borek	Asli_Borek

名称	ID
Asotsiatsia Ukraina	Asotsiatsia_Ukraina
阿斯彭牙科	Aspen_Dental
asports	asports
Aspro	Aspro
阿萨姆邦 Gramin Vikash 银行	Assam_Gramin_Vikash_Bank
阿萨姆邦石油	Assam_Oil
上帝召会 CU	God_cu 的装配
上帝召会教堂	上帝教会集会
助攻-2-卖出	助攻-2-卖出
美联银行	关联银行
联合医疗保健 CU	关联医疗保健_cu
美联社	美联社
联合学校员工 CU	关联学校_员工_cu
关联超市	关联超市
联合金融服务公司	合伙人_金融_服务
各式各样	各式各样
各式各样	各式各样
Assotsiatsiya Progress	Assotsiatsiya_Progress
保障之星	信贷之星
保证 Ami	保障_Ami

名称	ID
保证 Comar	保障_Comar
保证 Gat	保障_Gat
保证 Lloyd Tunisien	保险_Lloyd_Tunisien
马格里比亚的保证	保险_马格里比亚
保证 Salim	保险_Salim
紫苑药房	阿斯特_药房
Astera CU	Astera_cu
Astikoto	Astikoto
阿斯顿	阿斯顿
阿斯托里亚银行	Astoria_Bank
Astotel	Astotel
Astra Asigurari	Astra_Asigurari
阿斯利康银行	Astra_Bank
阿斯特拉罕俄罗斯天然气工业股份公司	阿斯特拉罕俄罗斯天然气工业股份公司
Astree 保证	Astree_Insurs
Astro	Astro
Astron	Astron
太空人队	太空人队
阿斯图里亚斯	阿斯图里亚斯
Asvio 银行	Asvio_Bank

名称	ID
Aswak Assalam	Aswak_assalam
在家里	At_Home
Atacadao	Atacadao
Atakadao Atakarejo	Atakadao_Atakarejo
Atasay	Atasay
Atasun Optik	Atasun_Optik
Atelier de Cour Celles	Atelier_de_Cour_Celles
Atem	Atem
Athena	Athena
雅典石油	Athens_Oil
运动员世界	运动员_世界
阿提拉姆酒店	阿提拉姆_酒店
亚特兰大	亚特兰大
亚特兰大面包公司	亚特兰大面包公司
亚特兰大邮政大学	Atlanta_Postal_cu
大西洋	大西洋
大西洋廉价旅馆	大西洋经济旅馆
大西洋 FCU	Atlantic_FCU
大西洋超市	大西洋超市
大西洋联合银行	大西洋联合银行

名称	ID
大西洋哈利	Atlantik_Hali
亚特兰蒂斯旅行社	Atlantis_Viajes
Atlantsolia	Atlantsolia
阿特拉斯·科普柯	Atlas_Copco
阿特拉斯本田	Atlas_Honda
阿特拉斯州际特工	Atlas_Interstate_Agent
阿特拉斯药房	阿特拉斯_药房
Atlas Van	Atlas_Van_Lines
氛围牛仔裤	氛围_牛仔裤
Atom Tost	Atom_Tost
Atomik	Atomik
亚朵酒店	亚朵酒店
Atoy	Atoy
阿特雷	阿特雷
中庭酒店	中庭_酒店
attendo	attendo
注意	注意
Attijari 银行	Attijari_Bank
Attijariwafa 银行	Attijariwafa_Bank
Attock	Attock

名称	ID
Atu	Atu
Atunas	Atunas
Au Bon Pain	Au_Bon_Pain
澳大利亚局	Au_Burea
Au Peche Mignon	Au_Peche_Mignon
Au Vieux Duluth	Au_Vieux_Duluth
Aubade	Aubade
Aubainerie	Aubainerie
奥伯特	奥伯特
Aubuchon Hardware	Aubuchon_Hardware
奥本大学 CU	Auburn_University_CU
欧尚	欧尚
欧尚城	Auchan_city
欧尚大道	Auchan_Drive
奥迪	奥迪
奥迪卡	奥迪卡
音频 2000	音频_2000
音频诊所	音频诊所
奥黛丽	奥黛丽
奥古斯塔银行	奥古斯塔银行

名称	ID
奥古斯丁	奥古斯丁
斯特拉阿姨	阿姨_斯特拉
珍妮阿姨	Auntea_Jenny
安妮阿姨的	Auntie_Anne's
AurGroup 金融 CU	AurGroup_financial_cu
Aurecon	Aurecon
Aurelia	Aurelia
Aurelio 的披萨	Aurelio's_Pizza
Aurgi	Aurgi
Aurika	Aurika
Aurobindo Pharma	Aurobindo_Pharma
Aurora FCU	aurora_fcu
Aurora Health	极光_健康_保健
Aurora 药房	极光药房
Aurora 学校 FCU	aurora_schools_FCU
澳洲面包店	澳大利亚_面包店
Ausotel	Ausotel
澳大利亚出售	澳大利亚_处置
澳大利亚宠物手机	Aussie_Pet_Mobile
澳大利亚袜队	Aussie_Sox

名称	ID
Aussino	Aussino
奥斯汀	奥斯汀
奥斯汀·里德	Austin_Reed
澳大利亚邮政	澳大利亚邮报
澳大利亚人	澳大利亚人
澳大利亚地理	澳大利亚_地理
澳大利亚军事银行	澳大利亚军事银行
澳大利亚皮肤诊所	澳大利亚皮肤诊所
奥地利精英赛	奥地利_经典
奥地利时尚酒店	奥地利_热门酒店_酒店
奥地利阿纳迪银行	奥地利_阿纳迪银行
澳洲银行	Auswide_Bank
奥亚特兰蒂斯	奥亚特兰蒂斯
正品脚	Authentic_
Auto 3000	Auto_3000
Auto 5	Auto_5
亚特兰蒂斯汽车	汽车_亚特兰蒂斯
Auto Bleue	Auto_Bleue
自动乘务员	Auto_Crew
汽车医生	自动医生

名称	ID
汽车玻璃中心	自动玻璃中心
汽车玻璃专家	汽车玻璃专家
Auto Jardin	Auto_Jardim
汽车之王	Auto_King
汽车大师赛	汽车大师赛
一号汽车	Auto_One
汽车租赁	汽车租赁
自动恢复	自动恢复
汽车保安	Auto_Securita
自动停止	Auto_Stop
汽车安全控制技术	汽车安全控制技术
车主保险	车主_保险
AutoBarn	AutoBarn
AutoCrew	AutoCrew
AutoNation	AutoNation
AutoXpress	AutoXpress
AutoZone	AutoZone
Autoasi	Autoasi
高速公路	高速公路
奥托贝拉	奥托贝拉

名称	ID
自动引用	自动引用
自动控制	自动控制
Autodoc	Autodoc
自动驾驶员	自动驾驶员
Autoescuelas JORDA	autoescuelas_Jorda
自动专家	自动专家
自动调音栏	自动调音栏
自动适配	自动适配
汽车食品	汽车食品
Autogo	Autogo
Autogol	Autogol
亲笔签名	亲笔签名
亲笔签名系列酒店	签名_收藏_酒店
Automat 1	Automat_1
自动餐厅	自动餐厅
Autonom 租车	自主租赁汽车
Autop	Autop
自动专业版	自动专业版
汽车租赁	汽车租赁
Autos Don Fundas	Autos_Don_Fundas

名称	ID
自动服务	自动服务
自动智能	自动智能
Autosur	Autosur
Autotaalglass	Autotaalglass
自动视觉	自动视觉
Autowax	Autowax
Aux	Aux
Avadian CU	Avadian_cu
阿瓦拉尔房地产	Avalar_Real_Estate
Avangard	Avangard
Avans	Avans
前卫豪斯	Avant_Haus
Avantage7	Avantage7
Avantdent	Avantdent
阿凡提	阿凡提
Avantipl	Avantipl
Avanzi	Avanzi
凯撒大道	Ave_Caesar
Avea	Avea
Aveda	Aveda

名称	ID
Aventa	Aventa
Aventa CU	Aventa_cu
安万特制药	安万特制药
大道	大道
阿维拉	阿维拉
Averitt 快车	Averitt_Exp
Avery-Hess 房地产经纪人	Avery-Hess_Realtors
Avestar	avestar_cu
Avgustina	Avgustina
阿维安卡	阿维安卡
Avias	Avias
狂热	狂热
Avin	Avin
Avitsenna	Avitsenna
Avivi	Avivi
埃文代尔	埃文代尔
Avos'ka	Avos'ka
阿沃斯卡	阿沃斯卡
Avr 租车	AVR_rent_a_car
Avs	Avs

名称	ID
Avsar Sinema	Avsar_Sinema
Avtodorstroy	Avtodorstroy
Avtokrazbank	Avtokrazbank
Avtomag	Avtomag
Avtomotiv	Avtomotiv
Avtotreyd	Avtotreyd
阿波山寨料理	Awa_Shanzhai_Cuise
Awash 保险公司	Awash_Insurance_公司
Awash 国际银行	Awash_国际银行
巧克力太糟糕了	太糟糕了_巧克力
Awlad Ragab	Awlad_Ragab
摩洛哥安盛保险	安盛保险_摩洛哥
Axa Crédit	Axa_Crédit
轴向	轴向
Axiang	Axiang
Axiom 电信	Axiom_Telecom
Axis	Axis
安讯士银行	Axis_Bank [银行]
安讯士银行	Axis_Bank [银行]
Axon	Axon

名称	ID
Axotel	Axotel
Ayakkabi Dunyasi	Ayakkabi_Dunyasi
Ayam Bakar Mas Mon	Ayam_Bakar_Mas_Mono
Ayam Bakar Wong S	Ayam_bakar_Wong_Solo
Ayam Goreng Fatmawati	Ayam_Goreng_Fatmawati
Ayam penyet-ap	ayam_penyet-ap
Ayamas	Ayamas
Aybolit	Aybolit
Aydin Turizm	Aydin_Turizm
伊洛瓦底银行	Ayearwaddy_Bank
Aygin	Aygin
Ayhan	Ayhan
艾尔斯	艾尔斯
艾尔斯酒店	艾尔斯酒店
艾西拉	艾西拉
Aytemiz 市场	Atemiz_Market
Aytemiz Petrol	Aytemiz_Petrol
Ayvalik Tost	Ayvalik_Tost
Ayyildiz	Ayyildiz
麻布茶房	Azabu_Sabou

名称	ID
Azalea City CU	Azalea_city_cu
Azbuka Vkusa	Azbuka_Vkusa
Aziatsko-Tikhookeanskiy 银行	Aziatsko-Tikhookeanskiy_Bank
Azim Ticaret	Azim_Ticaret
偶氮石油产品	偶氮石油产品
阿兹台克人	阿兹台克人
Azuline	Azuline
Acai Villa Roxa	Acaí_Villa_Roxa
Acor 出租	Acor_Rent
Acoreana Seguros	Acoreana_Seguros
巴黎机场	巴黎国际机场
艾塔	艾塔
A	A
B & B	B_&_B
B KUL	B_KUL
B Sabadell	B_Sabadell
B 旅游品牌	b_the_Travel_brand
意大利住宿加早餐旅馆	B&B_Italia
B&D	B&D
B&G 门店	B&G_store

名称	ID
B&M	B&M
B&Q	B&Q
B's mart	b's_mart
B+B Parkhaus	B+B_Parkhaus
B-Quik	B-Quik
B.	B._Young
B.O.C.	B.O.C.
B1 折扣 Baumarkt	B1_Discount_Baumarkt
B2S	B2S
B3 Takarék	B3_Takarék
BAAC	BAAC
宝宝中心	宝贝中心
BABY FACE 星球的	baby_face_Planet
婴儿房	婴儿房
BAC	BAC
BAC 圣何塞	bac_san_Jose
BACK MYHOME 连锁酒店	BACK_MYHOME_HOTEL_CHAIN
后工厂	后工厂
培根天堂	Bacon_Paradise
BADI BABY 照片	badi_baby_photo

名称	ID
BADR	BADR
巴杜拉	巴杜拉
包给你	BAG_TO_YOU
BAGAGGIO	BAGAGGIO
BAGGA	BAGGA
巴哈马	巴哈马
BAI	BAI
北京汽车	北京汽车_汽车
百信药业	百信药房
百怡超市	百艺超市
BAKMAZ	BAKMAZ
BALENO	BALENO
民谣	民谣
BALLENOIL	BALLENOIL
BALLY	BALLY
巴尔特投资银行	巴尔特投资银行
BAM	BAM
BAMBULE	BAMBULE
BAMUL Nandini	bamul_Nandini
BAN JIOU WOO	BAN_JIOU_WOO

名称	ID
BANBAJIO	BANBAJIO
BANCA ANTONIANA POPOLARE VENETA	BANCA_ANTONIANA_POPOLARE_VENETA
BANCA BRIGNONE	BANCA_BRIGNONE
BANCA CARIGE	BANCA_CARIGE
BANCA CENTROPADANA	BANCA_CENTROPADANA
意大利商业银行	意大利商业银行
BANCA CR 佛罗伦萨	BANCA_CR_FIRENZE
BANCA CRV	BANCA_CRV
BANCA D'ALBA CREDITO COOPERATIVO	BANCA_D'ALBA_CREDITO_COOPERATIVO
意大利银行	意大利银行
BANCA DEL GARDA	BANCA_DEL_GARDA
伦巴多领地银行	BANCA_DEL_TERRITORIO_LOMBARDO
亚得里亚海银行	BANCA_DELL'ADRIATICO
BANCA DELLA MARC	BANCA_DELLA_MARCA
阿斯蒂银行	BANCA_DI_ASTI
贝加莫银行	BANCA_DI_BERGAMO
合作信贷银行	BANCA_DI_CREDITO_COOPERATIVO
SARDO 信用银行	BANCA_DI_CREDITO_SARDO
BANCA di IMOLA S.p.A.	Banca_di_imola_s.p.A.
巴勒莫银行	BANCA_DI_PALERMO

名称	ID
皮亚琴察银行	BANCA_DI_PIACENZA
里米尼银行	BANCA_DI_RIMINI
萨萨里银行	BANCA_DI_SASSARI
特伦托和博尔扎诺银行	BANCA_DI_TRENTO_E_BOLZANO
欧洲移动银行	BANCA_EUROMOBILIARE
BANCA FARMAFACTORING	BANCA_FARMAFACTORING
忠意银行	BANCA_GENERALI
马拉泰斯蒂亚纳银行	BANCA_MALATESTIANA
BANCA 游行	BANCA_MARCH
BANCA MEDIOLANUM	BANCA_MEDIOLANUM
帕尔马蒙特银行	银行_蒙特_帕尔马
BANCA PADOVANA CREDITO COPERATIVO	BANCA_PADOVANA_CREDITO_COOPERATIVO
BANCA PASSADORE & C.	BANCA_PASSADORE_&_C。
大众银行 DI APRILIA	BANCA_POPOLARE_DI_APRILIA
巴里大众银行	BANCA_POPOLARE_DI_BARI
马罗斯蒂卡大众银行	BANCA_POPOLARE_DI_MAROSTICA
拉文纳大众银行	BANCA_POPOLARE_DI_RAVENNA
意大利大众银行	BANCA_POPOLARE_ITALIANA
大众银行 PUGLIA 和 BASILICATA	BANCA_POPOLARE_PUGLIA_E_BASILICATA
BANCA POPOLARE VALCONCA	BANCA_POPOLARE_VALCONCA

名称	ID
卡西纳特大众银行	Banca_popolare_del_cassinate
BANCA PUEYO	BANCA_PUEYO
BANCA SELLA	BANCA_SELLA
托斯卡纳银行	BANCA_TOSCANA
BANCA VALSABBINA S.c.p.A.	Banca_valsabbina_s.c.p.A.
BANCADRIA CREDITO COOP。三角洲	BANCADRIA_CREDITO_COOP。 _DEL_DELTA
BANCO CARREGOSA	BANCO_CARREGOSA
乌拉圭中央银行	乌拉圭中央银行
哥伦比亚银行	哥伦比亚银行
商业银行	BANCO_COMERCIAL
西班牙合作银行	BANCO_COOPERATIVO_ESPAÑOL
卡斯蒂利亚银行	BANCO_DE_CASTILLA
商业银行	BANCO_DE_COMERCIO
BALEAR 信用银行	BANCO_DE_CREDITO_BALEAR
加利西亚银行	BANCO_DE_GALICIA
BANCO DE LA MUJER	BANCO_DE_LA_MUJER
阿根廷国民银行	BANCO_DE_LA_NACION_ARGENTINA
BANCO DE VASCONIA	BANCO_DE_VASCONIA
商业银行	BANCO_DEL_COMERCIO
那不勒斯银行	那不勒斯银行

名称	ID
圣灵银行	BANCO_ESPIRITO_SANTO
金融银行	BANCO_FINANCIERO
BANCO Financiero-Agen	Banco_Financiero-Agente
BANCO GNB	BANCO_GNB
BANCO Hipotecario	Banco_Hipotecario
工业银行	工业银行
银行投资	BANCO_INVEST
马德里银行	BANCO_MADRID
BANCO MULTIVA	BANCO_MULTIVA
BANCO PENTA	BANCO_PENTA
BANCO PICHINCHA	BANCO_PICHINCHA
大众银行	BANCO_POPULA
共和国银行	BANCO_REPUBLICA
泰丰银行	BANCO_TAI_FUNG
BANCO mediolanum	banco_mediolanum
BANESTES	BANESTES
BANIF	BANIF
银行	银行
BANK ALBILAD	BANK_ALBILAD
ALJAZIRA 银行	BANK_ALJAZIRA

名称	ID
BANK ARTHA GRAHA	BANK_ARTHA_GRAHA
BANK BTPN	BANK_BTPN
BTPN Sinaya 银行	bank_btpn_Sinaya
银行 DKI	BANK_DKI
银行 DKI Syariah	bank_dki_syariah
BANK MELLI 伊朗	BANK_MELLI_IRAN
MUAMALAT 银行	BANK_MUAMALAT
阿育提亚银行	BANK_OF_AYUDHYA
北京银行	北京银行
沧州银行	沧州银行
长沙银行	长沙银行
成都银行	成都银行
重庆银行	重庆银行
大连银行	大连银行
东莞银行	东莞银行
抚顺银行	抚顺银行
赣州银行	赣州银行
广州银行	广州银行
贵阳银行	贵阳银行
杭州银行	杭州银行

名称	ID
河北银行	河北银行
江苏银行	江苏银行
吉林银行	吉林银行
锦州银行	锦州银行
九江银行	九江银行
廊坊银行	廊坊银行
兰州银行	兰州银行
辽阳银行	辽阳银行
柳州银行	柳州银行
洛阳银行	洛阳银行
南京银行	南京银行
宁波银行	宁波银行
青岛银行	青岛银行
青海银行	青海银行
泉州银行	泉州银行
台湾银行	台湾银行
天津银行	天津银行
潍坊银行	潍坊银行
西安银行	西安银行
营口银行	营口银行

名称	ID
郑州银行	郑州银行
SGB 银行	BANK_SGB
银行信号	BANK_SINAR
BANK SYARIAH MANDIRI	BANK_SYARIAH_MANDIRI
BANK WINDU	BANK_WINDU
BANOCO	BANOCO
BANQUE BEMO	BANQUE_BEMO
BANQUE BNPI	BANQUE_BNPI
BANQUE CHAIX	BANQUE_CHAIX
BANQUE CHALUS	BANQUE_CHALUS
BANQUE COURTOIS	BANQUE_COURTOIS
阿尔及利亚银行	BANQUE_D'ALGERIE
突尼斯银行	BANQUE_DE_TUNISIE
BANQUE DU CAIRE	BANQUE_DU_CAIRE
BANQUE DU LIBAN	BANQUE_DU_LIBAN
BANQUE DUBUY	DE PARSEVAL
黎巴嫩-法兰西银行	BANQUE_LIBANO-FRANÇAISE
BANQUE MISR	BANQUE_MISR
BANQUE POUYANNE	BANQUE_POUYANNE
BANQUE RAIFFEISEN	BANQUE_RAIFFEISEN

名称	ID
悦榕庄酒店及度假村	悦榕庄_酒店及度假村
宝大祥儿童购物	BAO_DA_XIANG_SHIANG_为孩子们购物
宝龙居家酒店	宝龙_如家_酒店
BAOVIET银行	宝越银行
BAPCO	BAPCO
BAQET	BAQET
酒吧 B Q PLAZA	BAR_B_Q_PLAZA
BARA 超级	BARA_超级
巴克莱	巴克莱
BAREZZITO	BAREZZITO
讨价还价的酒	BARGAIN_BOOZE
BARGELLO	BARGELLO
咖啡师	咖啡师
BARMER GEK	BARMER_GEK
巴恩咖啡厅	巴恩咖啡厅
男爵	男爵
BARTEK	BARTEK
BARWA BANK	BARWA_BANK
基础商店	Base_Shop
BASHCOMSNABBANK	BASHCOMSNABBANK

名称	ID
BASILE PETROLI	BASILE_PETROLI
基础咖啡	BASIS_COFFEE
BASLER	BASLER
MATATA INGLESA	BATATA_INGLESA
BATISTORE	BATISTORE
包豪斯	包豪斯
巴伐利亚汽油	Bavaria_petro
BAWAG P.S.K.	BAWAG_P.S.K。
BAYRAM BAL 发型设计	BAYRAM_BAL_HAIR_DESIGN
BBAC	BBAC
BBBank	BBBank
BBC-波哥大啤酒公司	BBC-Bogota_Beer_Company
烧烤鸡	BBQ_Chicken
BBVA	BBVA
BBVA 指南针	BBVA_Compas
BBVA 大陆银行	BBVA_Contern
BBVA Continental-Agente	BBVA_Continental-Agental-
BBVA 法语	BBVA_Francés
卑诗省披萨	bc_Pizza
BC 体育	BC_Sports

名称	ID
BCBGMAXAZRIA	BCBGMAXAZRIA
BCC	BCC
BCC COOPERATIVO Roma	bcc_credito_Cooperativo_Roma
BCC Carugate e e Inzago	bcc_carugate_e_e_inzago
BCC DI CAMBIANO	BCC_DI_CAMBIANO
BCD Travel 西班牙	BCD_Travel_Spain
BCEE	BCEE
BCF	BCF
BCHEF	BCHEF
BCI	BCI
BCL	BCL
BCM 银行	bcm_bank
BCP	BCP
BCP-Agente	BCP-Agente
BCR	BCR
BDC	BDC
BDL	BDL
BDO	BDO
BDS	BDS
BEA	BEA

名称	ID
比尔斯佛罗里达州	BEALLS_FLOR
BEALLS Out	Bealls_Outle
光柱	光柱
BEARES	BEARES
击败	击败
美容系列	BEAUTY_LINE
美容成功	美丽_成功
BEBA 孩子们	BEBA_KIDS
BECK'S 咖啡店	BECK'S 咖啡店
因为	因为
床和浴室	BED_&_BATH
床浴 N'TABLE	BED_BATH_N'_TABLE
床上用品世界	床上用品世界
嘟嘟声	嘟嘟声
啤酒沙龙	Beer_saloon
北奔卡车	BEIBEN_TRU
BEL-POL	BEL-POL
BELAMBRA VVF	BELAMBRA_VVF
BELEN FARMACIAS	BELEN_FARMACIAS
贝拉豆咖啡	BELLA_BEANS_COFFEE

名称	ID
BELLAGIO	BELLAGIO
BELROS	BELROS
BELWEST	BELWEST
BENGAWAN SOLO	BENGAWAN_SOLO
弯曲	弯曲
宾利	宾利
BENU	BENU
BENU Lekarna	benu_lekarna
BEOSPORT	BEOSPORT
BEREKET DONER	BEREKET_DONER
BERLIN DÖNER KEBAP	BERLIN_DÖNER_KEBAP
伯纳迪	伯纳迪
伯尼娜	伯尼娜
BEROIL	BEROIL
BERTO LUCCI	BERTO_LUCCI
伯特兰	伯特兰
BES	BES
BESO	BESO
最好的咖啡	最好的咖啡
最好的食物	BEST_FOOD

名称	ID
最好的朋友音乐	最好的朋友音乐
最好的油	BEST_OIL
BEST ONE 轮胎和服务	BEST_ONE_TIRE_&_SERVICE
最好的光学	BEST_OPITCAL
最好的旅行	最佳旅行
BEST Travnik	best_travnik
BETACAR	BETACAR
BETAGRO 商店	Betagro_shop
美好生活购物中心	BETTER_LIFE_MALL
BEYFIN	BEYFIN
BFA	BFA
BFC	BFC
BFI 金融	BFI_Finance
BFMPE	BFMPE
BGL 法国巴黎银行	BGL_BNP_PARIBAS
BGZ 法国巴黎银行	BGZ_BNP_PARIBAS
BH Posta	bh_posta
BH 电信	BH_Teleco
BHCI	BHCI
BHD	BHD

名称	ID
BHF-Bank	BHF-Bank
唐必峰	BI_FENG_TANG
双重奢华	BI_LEXURY
BI-BI	BI-BI
BI-LO	BI-LO
比亚多哥	BI_TOGO
BIALCON	BIALCON
BIALY JELEN	BIALY_JELEN
BIANCA-FASHION	BIANCA-FASHION
BIANCO 鞋	BIANCO_FOOTHERS
BIANN-BIANN-U	BIANN-BIANN-U
标	标
BIAT	BIAT
圣经	圣经
BIBI Sucos	bibi_sucos
BIBIBOP 亚洲烧烤	bibibop_Asian_Grill
BIBoVINo	BIBoVINo
BICICI	BICICI
BICIGUI	BICIGUI
BICIS	BICIS

名称	ID
BIDC	BIDC
BIDV	BIDV
BIDVest 银行	BIDVest_银行
BIERHALLE	BIERHALLE
BIFORIS	BIFORIS
五大体育用品	BIG_5_SPORTING_GOODS
大盒子	BIG_BOX
BIG Hipermercado	big_HiperMercado
大明星	BIG_STAR
BIG W	BIG_W
BIG bompreço	big_bompreço
BIG4	BIG4
BIK BOOK	BIK_BOK
BIL	BIL
BILJEMERKANT	BILJEMERKANT
比尔卡	比尔卡
BILLA	BILLA
BILLA 停下来购物	Billa_stop_&_shop
烧烤比利	BILLY_THE_GR
BIMBO'S	BIMBO'S

名称	ID
本·塞勒姆控股	BIN_SALEM_HOLDINGS
BINBANK	BINBANK
宾果游戏	宾果游戏
BIOMAX	BIOMAX
BIP BIP	BIP_BIP
BIPA	BIPA
小酒馆罗曼	BISTRO_ROMAIN
BITA'S	BITA'S
BITOQUE no PONTO	bitoque_no_ponto
BIVERBANCA	BIVERBANCA
BIWAY 时尚酒店	BIWAY_FASHION_HOTEL
商业艺术	BIZ_ART
BIZIM	BIZIM
BIZKARRA	BIZKARRA
BJ Santos	bj_Santo
BJ's	BJ's
BJ 的 Restaurant & Brewhouse	bj's_restaurant_&_Brewhouse
BJ's 批发俱乐部	bj's 批发俱乐部
BK	BK
BKCP	BKCP

名称	ID
BKS 银行	bks_bank
BKS Prem'er	bks_prem'er
BKT	BKT
黑峡谷咖啡	黑峡谷咖啡
黑峡谷厨房	黑色峡谷厨房
黑色红色白色	黑_红_白
宝珀	宝珀
BLAUMAX	BLAUMAX
BLB Eine Marke der Nord/LB	blb_eine_marke_der_nord/LB
BLC 银行	blc_bank
BLISKA	BLISKA
BLISKA 药房	bliska_Phar
方块电库	block_elect
BLOKKER	BLOKKER
开花	开花
蓝山咖啡馆	蓝山咖啡厅
蓝色方式	蓝色之路
蓝山新鲜咖啡	蓝山_新鲜_咖啡
BLUhands	BLUhands
BM	BM

名称	ID
BMB 投资银行	bmb_investment_Bank
BMCE 银行	BMCE_BANK
BMCI	BMCI
BMI FCU	BMI_FCU
BMN	BMN
BMO 蒙特利尔银行	BMO_Bank_of_Montreal
BMO 哈里斯银行	bmo_Harris_Bank
BMQ	BMQ
BMS Organics	bms_Organics
宝马	宝马
BMWi	BMWi
BNA	BNA
BNF	BNF
BNI	BNI
BNL	BNL
法国巴黎银行	法国巴黎银行_Paribas
法国巴黎银行富通	BNP_paribas_Fortis
BNT	BNT
BNT 汽车	BNT_汽车
BNZ	BNZ

名称	ID
BO.BO	BO.BO
鲍勃	鲍勃
BOBSON	BOBSON
BOBSTORE	BOBSTORE
BOC Gases	BOC_Gases
中银国际证券	中银国际_证券
BOD	BOD
身体谈话	BODY_TALK
BODZIO	BODZIO
英国央行停车	BOE 停车
BOESMANLAND BILTONG	BOESMANLAND_BILTONG
大胆	大胆
BOLIST	BOLIST
BOLV-OIL	BOLV-OIL
BON GRILLE	BON_GRILLE
博纳	博纳
BONACHERO	BONACHERO
邦咖啡	邦咖啡
BONUS 建筑保养	奖励_建筑_护理
图书中心	BOOK_CENTER

名称	ID
BOOKAZINE	BOOKAZINE
书籍 Kinokuniya	books_Kinokuniya
图书销售	图书销售
靴子广场	BOOT_SQU
展位	展位
靴子	靴子
BORGIA	BORGIA
BORNGA	BORNGA
BOS	BOS
博世汽车服务	博世汽车服务
BOSI	BOSI
BOSO	BOSO
BOTI	BOTI
秘鲁博蒂卡斯	BOTICAS_PERU
波顿牛排馆	BOTON_STEAK_HOUSE
BOUDL	BOUDL
BOULANGERIEFISCHER	BOULANGERIE_FISCHER
男孩&女孩	男孩&女孩
BP	BP
BP 2go	bp_2go

名称	ID
BP Connect	bp_Connect
英国石油快车	bp_Express
BP 商店	bp_shop
BP-中石油	BP-中石油
BPER：Banca	bper: _banca
BPET 市场	bpet_Market
BPI	BPI
BPI 家庭储蓄银行	BPI_Family_Savings_Bank
BPN	BPN
BPR	BPR
BPS-SberBank	BPS-SberBank
BPU banca	bpu_banca
BQ	BQ
BQ 酒店	bq_Hotels
BR	BR
BR狂热	br_mania
br-lelut	br-lelut
BRACAFÉ	BRACAFÉ
布拉德利的药房	BRADLEY'S 药房
布拉德洛斯	布拉德洛斯

名称	ID
BRANEW	BRANEW
布兰塔诺	布兰塔诺
BRAPPERS	BRAPPERS
BRASAYLEÑA	BRASAYLEÑA
太棒了！意大利菜	太棒了！ _CUCINA_ITALIANA
BRAX	BRAX
BRB	BRB
BRE 银行股份公司	bre_bank_SA
繁殖	繁殖
BREE	BREE
百年灵	百年灵
不来梅	不来梅
BRG Mart	BRG_mart
BRICO E.Leclerc	brico_e.leclerc
BRICO PLAN-IT	BRICO_PLAN-IT
BRICOLEX	BRICOLEX
BRICOMAN	BRICOMAN
BRICOMARCHE Muszkieterowie	bricomarche_muszkieterowie
BRICORAMA	BRICORAMA
红茶馆	BRIDAL_TEA_HOUSE

名称	ID
BRIGTS	BRIGTS
BRIO 托斯卡纳烧烤	Brio_Tuscan_Grille
不列颠尼亚	不列颠尼亚
英属印度	英属印度
BRK	BRK
BROADWAY	BROADWAY
BROJLER	BROJLER
兄弟	兄弟
BROU	BROU
棕色	棕色
BRS	BRS
布鲁诺	布鲁诺
布鲁诺·安托格尼尼	BRUNO_ANTOGNINI
BRUPHILS	BRUPHILS
BRÚJULA	BRÚJULA
BSE	BSE
BSH	BSH
BASIC	BASIC
BSN	BSN
BTA 银行	Bta_bank

名称	ID
BTCI	BTCI
BTG 国际旅行与旅游	BTG_International_Tours_Tours
BTI 银行	BTI_Bank
顺便说一句	顺便说一句
BTMU	BTMU
BTP	BTP
BTV VIER LANDER BANK	BTV_VIER_LANDER_BANK
BUCO	BUCO
BUD'S	BUD'S
BUDNI	BUDNI
水牛烧烤	水牛烧烤
自助餐快递	自助餐快递
别克	别克
BUILD-A-BEAR 工作坊	BUILD-A-BEAR_WOR
BULE MARLIN 酒吧和餐厅	Bule_Marlin_Bar&Restauran
BURAQOIL	BURAQOIL
巴宝莉	巴宝莉
BURGAN BANK	BURGAN_BANK
汉堡屋	BURGER_HOUSE
汉堡王	BURGER_KING

名称	ID
汉堡牧场	BURGER_RANCH
伯克斯奥特莱斯	BURKES_OUTLET
灌木丛	灌木丛
丛林人	丛林人
但是-S	但是-S
宝格丽	宝格丽
BW 水牛之翼	BW_BUFALO_WINGS
BW 银行	bw_bank
BWOC	BWOC
BWS	BWS
BYBLOS BANK	BYBLOS_BANK
BYGGmax	BYGGmax
BYKEA	BYKEA
BYTOM	BYTOM
香蕉	香蕉
Baazar 加尔各答	Baazar_加尔各答
Babbage's	Babbage's
Babela 的厨房	Babela's Kitchen
从婴儿到孩子	babies_to_Kids
Babies “R” Us	Babies “R” Us

名称	ID
婴儿潮	Baby_Boom
宝贝彩旗	Baby_Bunting
婴儿护理	Baby_Care
宝贝城	Baby_City
Baby Creysi	Baby_Creysi
宝贝风格	Baby_Style
Baby Walz	Baby_Walz
宝宝俱乐部	宝贝俱乐部
小象生活超市	baby_Elephant_Life_s
baby1One	baby1One
宝贝创意	宝贝创意
婴儿用品店	婴儿用品店
百家乐	百家乐
Bach Hoa XANH	bach_hoa_xanh
Bachir 冰淇淋	Bachir_Ice_Cream
Bachpan	Bachpan
Bacini's Pizzeria	Bacini's Pizzeria
Bacio di Latte	bacio_di_latte
回到床上	Back_To_Bed
后院汉堡	Back_Yard_Burgers

名称	ID
重返运动	Back_in_motion
Back.Bude	Back.Bude
巴克豪斯哈克纳	Backhaus_Hackner
Bacolod Chicken Inasal	Bacolod_Chicken_Inasal
坏屁股咖啡公司	坏屁股咖啡公司
Badenia	Badenia
Badget	Badget
Badianban Mart	badianban_Mart
Badr Al-Islami Bank	Badr_Al-Islami_Bank
Bafei	Bafei
百吉饼和百吉饼	Bagel_&_Bagel
百吉饼街咖啡厅	Bagel_Street_Cafe
百吉饼和豆子	Bagels_&_Beans
Bagelstein	Bagelstein
Bageterie 大道	Bagterie_Boulevard
Baggit	Baggit
Bagorama	Bagorama
八桂旅馆	Bagui_Inn
Baguo Buyi	Baguo_Buyi
Bagus	Bagus

名称	ID
巴哈马微风	巴哈马_微风
Bahama Buck's	Bahama_Buck's
Bahetle	Bahetle
Bahhy	Bahhy
Bahne	Bahne
Bahotu	Bahotu
巴林开发银行	巴林_开发银行
巴林融资公司	巴林_融资_公司
巴林伊斯兰银行	巴林_伊斯兰银行
巴林沙特银行	巴林_沙特银行
巴伊亚普林西比酒店及度假村	Bahía_Principe_酒店_度假村
白佳乐	Bai_Jia_Le
白佳永辉	Bai_Jia_Yong_Hui
北汽幻速	Baic_环速科技
北汽绅宝	Baic_Senova
北汽威旺	Baic_Weiwang
Baichak	Baichak
百达电气	百达_电气
贝利班克斯和比德尔	Bailey_Banks_&_Biddle
白莲	白莲

名称	ID
Bain & Company	Bain_&_Company
贝尔德	贝尔德
百味果	百味果
百兴药房	百姓_药店
百心堂药房	百心堂_药店
Baja Auto	Baja_Auto
巴哈新鲜	Baja_Fresh
Bajaj	Bajaj
Bajaj Allianz 普通保险	Bajaj_Allianz_普通_保险
Bajaj Allianz 人寿保险	Bajaj_Allianz_人寿_保险
巴贾杰电子	Bajaj_电子
Bajio 墨西哥烧烤	Bajio_Mexican_Grill
烤土豆	Baked_Potato
贝克·布伦	Baker_Brun
贝克·汉森	Baker_Hansen
贝克·诺德比	Baker_Nordby
贝克的	贝克的
贝克旅馆	Baker's Inn
贝克药房	贝克药房
贝克披萨	Baker's_Pizz

名称	ID
Baker's 双床厨房	Baker's_Twin_Kitchen
Bakeriet	Bakeriet
面包师	面包师
Bakers Del	Bakers_Delig
贝克斯广场	Bakers_Squ
Bakhyt	Bakhyt
Baki Takarék	Baki_Takarék
Bakker Bart	Bakker_Bart
Bakker Klootwijk	Bakker_Klootwijk
Bakmi GM	bakmi_GM
Bakmi Naga	Bakmi_Naga
巴科斯兄弟	Bakos_Broth
Bakso Kepala Sapi	Bakso_Kepala_Sapi
Bakso Kota Cak	Bakso_Kota_Cak_Man
Bakso Lapangan Tembak Senayan	Bakso_Lapangan_Tembak_Senayan
巴尔巴拉蒂公立学校	Bal_Bharati_Public_School
Bala	Bala
巴拉巴拉	巴拉巴拉
平衡教育	平衡_教育
巴拉罗蒂	巴拉罗蒂

名称	ID
Balcarce	Balcarce
巴尔迪尼尼	巴尔迪尼尼
Balducci 的	Balducci 的
巴黎世家	巴黎世家
Balizza	Balizza
Balla Bütor	Balla_Bütor
Ballentynes 时装中心	Ballentynes_Fashion_Centra
Ballingslöv	Ballingslöv
Bally 全面健身	全力健身
Bally's Casino 酒店	Bally's_casino_Hotel
Baloise Bank Soba	Baloise_Bank_Soba
Balt-Treyd	Balt-Treyd
波罗的海国际银行	波罗的海国际银行
波罗的海石油	波罗的海_石油
波罗的海信托银行	波罗的海信托银行
波罗的海石油公司	波罗的海石油公司
巴尔蒂斯基银行	Baltiyskiy_B
Baltneft'	Baltneft'
Baly Supermercados	Baly_Supermercados
巴尔扎克咖啡	Balzac_Coffee

名称	ID
巴马茶	Bama_Tea
小鹿斑比	小鹿斑比
Ban Juou Woo	Ban_juou_Woo
BanBif	BanBif
BanBif-Agente	BanBif-Agente
Bancorreos	Bancorreos
BanCo	BanCo
BanCoppel	BanCoppel
香蕉共和国	香蕉_共和国
香蕉树	Banana_Tree
Banc Post	Banc_Post
BancABC	BancABC
BancFirst	BancFirst
Banca Agricola Mantovana	Banca_Agricola_Mantovana
Cerea Agricola di Cerea	Banca_agricola_di_cerea
Banca Alpi Marittime	Banca_Alpi_Marittime
Banca Carime	Banca_Carime
Banca Chantunala Grischuna	Banca_Chantunala_Grischuna
喀尔巴阡商业银行	Banca_Comerciala_Carpatica
罗马商业银行	Banca_Comerciala_Romana

名称	ID
Banca Coop	Banca_Coop
Banca Cooperativa Cattolica	Banca_Cooperativa_Cattolica
Banca Crt	Banca_Crt
Banca Dell'Agricoltura	Banca_Dell'Agricoltura
Banca FIDEURAM	Banca_fideuram
Banca Intesa	Banca_Intesa
莱昂纳多银行	Banca_Leonardo
伦巴达银行	Banca_Lombarda
伦巴达和皮埃蒙特银行	Banca_Lombarda_e_Piemontese
Banca Marche	Banca_Marche
Banca Mifel	Banca_Mifel
国家劳动银行	Banca_Nazionale_del_Lavoro
Banca Nuova	Banca_Nuova
Banca Polare Polare 普利亚和巴西利卡塔大区	Banca_Polare_puglia_e_Basilicata
Banca Ponti	Banca_Ponti
Banca Pop。商业和工业	Banca_Pop。 _商业_e_Industria
Banca Pop。Dell'Etruria 和 del Lazio	Banca_Pop。 _dell'Etruria_e_del_Lazio
Banca Popol。商业和工业	Banca_Popol。 _商业_e_Industria
Banca Popolare dell'Emilia Romagna	Banca_Popolare_dell'Emilia_Romagna
Banca Popolare Pugliese	Banca_Popolare_Pugliese

名称	ID
拉齐奥大众银行	Banca_popolare_del_Lazio
Banca Popolare del Mezzogiorno	Banca_popolare_del_mezzogiorno
安科纳大众银行	Banca_popolare_di_Ancona
贝加莫人民银行	Banca_popolare_di_Bergamo
Cividale Popolare di Cividale	Banca_popolare_di_Cividale
Banca Popolare di Sondrio	Banca_popolare_di_sondrio
斯波莱托人民银行	Banca_popolare_di_spoleto
维琴察大众银行	Banca_popolare_di_Vicenza
欧洲地区银行	Banca_Regionale_Europe
Banca Romaneasca	Banca_Romaneasca
Transilvania 银行	银行_Transilvania
Banca Vesuviana	Banca_Vesuviana
Banca del Fucino	Banca_del_Fucino
皮埃蒙特银行	Banca_del_Piemonte
坎帕尼亚银行	Banca_della_Campania
Ciociaria 银行	Banca_della_ciociaria
马切拉塔省银行	Banca_della_provincia_di_macerata
提契诺州立银行	Banca_dello_stato_del_canton_ticino
博洛尼亚银行	博洛尼亚银行
卡利亚里银行	Banca_di_Cagliari

名称	ID
Credito Cooperat。Genovese	Banca_di_credito_Cooperat。 _Genovese
Banca di Credito Popolare	Banca_di_credito_Popolare
Banca di Valle Camonica	Banca_di_Valle_Camonica
维罗纳银行	Banca_di_Verona
Bancamigros	Bancamigros
Bancamía	Bancamía
Bancaribe	Bancaribe
Banco 24 小时	banco_24Horas
Banco AV 别墅	Banco_av_Villas
哥伦比亚农业银行	Banco_Agrario_de_Colombia
Banco Ahorro Famsa	Banco_Ahorro_Famsa
阿尔卡拉银行	Banco_Alcala
威尼托安布罗西亚诺银行	Banco_Ambrosiano_Veneto
阿特拉斯银行	Banco_Atlas
阿兹台克银行	Banco_Azteca
Banco BPM	banco_bpm
英国夏令时间银行	banco_bst
阿根廷毕尔巴鄂比斯开亚银行	Banco_Bilbao_Vizcaya_Argentaria
财团银行	Banco_Consorcio
Caixa Geral 银行	Banco_Caixa_Geral

名称	ID
社交账号银行	Banco_Caja_Social
Banco Caminos	Banco_Caminos
加那利银行	加那利银行
卡罗尼银行	Banco_Caroni
Ciudad 银行	Banco_Cidad
Banco Comafi	Banco_Comafi
康德尔银行	Banco_Condell
同盟银行	Banco_Confederado
大陆银行	大陆银行
Banco Credichile	Banco_Credichile
Banco Credicoop	Banco_Credicoop
乌拉圭信贷银行	银行_信贷_乌拉圭
Banco Desio Lazio	Banco_Desio_Lazio
花旗爱德华兹银行	Banco_Edwards_Citi
Efisa 银行	Banco_Efisa
恩特雷里奥斯银行	Banco_Entre_Rios
Banco Etcheverria	Banco_Etcheverría
欧洲金融银行	banco_europeo_de_Finanzas
银行外观	银行_外观
Banco Falabella	Banco_Falabella

名称	ID
Family 银行	Banco_Family
联邦银行	联邦银行
菲律宾银行	菲律宾银行
金融银行	Banco_Finantia
Sofinloc金融银行	Banco_Finantia_Sofinloc
福尔摩沙银行	Banco_Formosa
法兰西银行	Banco_Francés
GNB Sudameris 银行	banco_GNB_Sudameris
Banco Gallego	Banco_Gallego
General	Banco_General
圭亚那银行	Banco_Guayana
Banco Herrero	Banco_Herrero
乌拉圭Hipotecario银行	Banco_Hipotecario_del_Urugay
委内瑞拉工业银行	委内瑞拉工业银行
Banco Interactiones	Banco_Interactiones
国际银行	Banco_Internacional
丰沙尔国际银行	Banco_Internacional_do_Funchal
Banco Mais	Banco_Mais
Banco Mercantil	Banco_Mercantil
国家银行	Banco_Nacional

名称	ID
Ultramarino 国民银行	Banco_Nacional_Ultramarino
国家信贷银行	Banco_nacional_de_Crédito
巴拿马国民银行	Banco_Nacional_de_Panamá银行
莫桑比克机会银行	莫桑比克银行_Oportunidade_de_moçambique
巴黎银行	Banco_Paris
Banco Pastor	Banco_Pastor
巴塔哥尼亚银行	巴塔哥尼亚银行
Banco 钢琴	Banco_Piano
银行广场	Banco_Plaza
哥伦比亚大众银行	Banco_Popular_哥伦比亚
多米尼加大众银行	Banco_Popular _Dominicano
Banco Popullo & Desarrollo Comunal	banco_popula_y_y_desarrollo_comunal
葡萄牙管理银行	Banco_Portugès_de_Gestão
葡萄牙商业银行	Banco_portugès_de_negócios
Banco ProCredit	银行_ ProCredit
省银行	Banco_Provincia
欧罗巴农村银行	Banco_rural_Europa
地区银行	Banco_Regional
里奥哈银行	Banco_Rioja
Banco Ripley	Banco_Ripley

名称	ID
Banco Ripley-Agente	Banco_Ripley-Agente
圣胡安银行	Banco_San_Juan
圣克鲁斯银行	Banco_Santa_Cruz
圣达菲银行	Banco_Santa_Fe
桑坦德银行	桑坦德银行
银行安全	Banco_Security
西蒙银行	Banco_Simeon
索菲塔萨银行	Banco_Sofitasa
泰拉银行	Banco_Terra
图库曼银行	Banco_Tucuman
沃尔玛银行	Banco_Walmart
亚马逊银行	Banco_da_Amazônia
安达卢西亚银行	Banco_de_Andalucia
波哥大银行	Banco_de_Bogotá银行
智利银行	智利银行
Banco de Crédito BCredito B	banco_de_Crédito_BCP
西班牙银行	Banco_de_España
Banco de La Pampa	Banco_de_la_Pampa
莫桑比克银行	莫桑比克银行
西方银行	Banco_de_Occidente

名称	ID
葡萄牙银行	Banco_de_Portugal
火地岛银行	banco_de_tierra_del_Fuego
委内瑞拉银行	委内瑞拉银行
国家银行	Banco_de_la_Nacion
哥伦比亚共和国银行	Banco_de_la_República_Colombia
丘布特银行	banco_del_Chubut
Banco del Desarrollo	banco_del_desarrollo
布雷西亚银行	Banco_di_Brescia
加勒比银行	Banco_di_Caribe
Banco di Desio 和 della Brianza	banco_di_desio_e_della_Brianza
圣乔治银行	Banco_di_san_Giorgio
撒丁岛银行	Banco_di_Sardegna
西西里银行	Banco_di_Sicilia
巴西银行	Banco_do_Brasil
Banco do Brasil Estilo	banco_do_brasil_Stilo
东北银行	Banco_do_Nordeste
Banco wb	banco_WWB
Banco Unico	Banco_Unico
Bancobice	Bancobice
Bancoestado	Bancoestado

名称	ID
Bancoex	Bancoex
Bancofar	Bancofar
哥伦比亚银行	哥伦比亚银行
Bancomer	Bancomer
Bancoomeva	Bancoomeva
BancoPopular-e.com	BancoPopular-e.com
Bancor	Bancor
BancorpSouth	BancorpSouth
Bancwest	Bancwest
Bandana's Bar B Que and G	bandana's_bar_b_que_and_Grill
乌拉圭乐队	乐队_乌拉圭
班丹银行	Bandhan_Bank
Banefe	Banefe
Banelco	Banelco
巴内斯科	巴内斯科
Banesto	Banesto
班菲尔德宠物医院	班菲尔德宠物医院
Bang & Olufsen	Bang_&_Olufsen
班加罗尔地区合作银行	班加罗尔地区合作银行
班加罗尔苏哈达合作银行	Bangalore_souharda_co-op_Bank

名称	ID
邦恰克	邦恰克
Bangi Kopitiam	Bangi_Kopitiam
Bangiya Gramin Vikash	Bangiya_Gramin_Vikash_Bank
曼谷银行	曼谷_银行
曼谷药店	曼谷_药店
曼谷烧烤	曼谷烧烤
曼谷保险	曼谷_保险
曼谷人寿保险	曼谷_人寿保险
曼谷炒锅	曼谷炒锅
Bangor FCU	Bangor_fcu
班戈储蓄银行	Bangor_Savings_Bank
刘海	刘海
Banh Mi Tuan 地图	Banh_mi_Tuan_Map
Banjercito	Banjercito
班卓琴	班卓琴
萨尔 1 号银行	Bank_1_Saar
银行 3/4	Bank_3/4
亚齐银行	银行_亚齐
Bank Al Yousr	Bank_Al_Yousr
哈比卜银行	Bank_Al-Habib

名称	ID
Al-Maghrib 银行	Bank_Al-Maghrib
Alfalah 银行	Bank_Alfalah
阿萨法银行	银行_阿萨法
亚洲银行	Bank_Asya
奥迪银行	银行_奥迪
奥地利银行	银行_奥地利
巴厘岛BPD银行	Bank_bpd_Bali
Bank BPD DIY	bank_bpd_diy
Bank BPH	bank_bph
银行 BPS	bank_bps
博胡斯拉夫银行	Bank_Bohuslav
Bank Bukopin	Bank_Bukopin
Bumiputera 银行	Bank_Bumiputera
银行清算员	Bank_cler
中亚银行	银行_中亚
中亚银行 Syariah	中亚银行_Syariah_Syariah
达纳蒙银行	Bank_Danamon
银行 Danamon Syariah	Bank_danamon_Syariah
佐法尔银行	Bank_Dhofar
叶卡捷琳堡银行	银行_叶卡捷琳堡

名称	ID
银行经济	Bank_Ekonomi
Bank Familnyi	Bank_Familnyi
甘尼萨银行	Bank_Ganesha
Gospodarstwa krajowego 银行	Bank_gospodarstwa_krajowego
Bank Hapoalim	Bank_Hapoalim
银行住房信贷	银行_家庭_信贷
Bank Icb Bumiputera	Bank_Icb_Bumiputera
印度尼西亚银行	银行_印度尼西亚
亚洲国际银行	银行_国际_亚洲
Inteza 银行	银行_Inteza
Bank Investytsii Ta Zaoshchadzhen	银行_Investytsii_ta_Zaoshchadzhen
伊斯兰银行	银行_Islam
巴基斯坦伊斯兰银行	巴基斯坦伊斯兰银行
Bank J. van Breda & Co	bank_j._van_breda_&_co
Jabar Banten 银行	Bank_Jabar_Banten
Jabar Banten Syariah 银行	Bank_Jabar_Banten_Syariah
雅加达贾萨银行	银行_Jasa_雅加达
Jateng 银行	Bank_Jateng
Jatim 银行	Bank_Jatim
卡尔巴银行	Bank_Kalbar

名称	ID
卡尔塞尔银行	Bank_Kalsel
卡尔蒂姆银行	Bank_Kaltim
楠榜银行	Bank_Lampung
Leumi 银行	Bank_Leumi
Bank Linth	Bank_Linth
曼迪里银行	Bank_Mandiri
马斯皮恩银行	Bank_Maspion
马萨德银行	Bank_Massad
玛雅帕达银行	Bank_Mayapada
马约拉银行	Bank_Mayora
Mega 银行	银行_Mega
Mega Syariah 银行	Bank_Mega_Syariah
梅拉特银行	Bank_Mellat
商业银行	Bank_Mercantile
梅斯蒂卡银行	Bank_Mestika
中西部银行	银行_中西部
Mizrahi Tefahot 银行	Bank_Mizrahi_Tefahot
莫斯科-明斯克银行	银行_莫斯科-明斯克
莫斯科银行	Bank_Moskvy
马斯喀特银行	银行_马斯喀特

名称	ID
银行互惠银行	银行_互惠
非关税壁垒银行	bank_NTB
NTT 银行	bank_NTT
长里银行	Bank_Nagari
Narodnyy Capital 银行	Bank_Narodnyy_Kapital
印度尼西亚国家银行	印度尼西亚国家银行
印度尼西亚国家银行 Syariah	Bank_Negara_Indonesia_Syariah
内蒙古银行	内蒙古银行
高雄银行	高雄银行
第一银行	Bank_One
Otsar 银行 HaHayal	Bank_Otsar_ HaHayal
巴布亚银行	银行_Papua
Bank Pekao	Bank_Pekao
农民银行	Bank_Pertanian
马来西亚农民银行	Bank_Pertanian_马来西亚
Bank Phb	Bank_Phb
Pocztowy 银行	Bank_Pocztowy
Pundi 银行	Bank_Pundi
人民银行	Bank_Rakyat
印度尼西亚人民银行	银行_人民银行_印度尼西亚

名称	ID
印度尼西亚人民银行 Syariah	Bank_Rakyat_Indonesia_Syariah
瑞森资本银行	Bank_Renesans_Kapital
银行存款	Bank_Reshenie
Riau Kepri 银行	银行_Riau_Kepri
俄罗斯银行	Bank_Rossiya
南澳银行	Bank_SA
伊朗萨德拉特银行	银行_萨德拉特_伊朗
Sahabat Sampoerna 银行	银行_Sahabat_Sampoerna
圣彼得堡银行	Bank_Sankt-Peterburg
Saudara 银行	Bank_Saudara
Schilling & Co.	Bank_Schilling_&_Co
国立储蓄银行	Bank_Simpanan_Nasional
西纳尔马斯银行	Bank_Sinarmas
银行 SinoPac	银行_ SinoPac
索哈尔银行	Bank_Sohar
Bank Spoldzielczy Rzemiosla	Bank_Spoldzielczy_Rzemiosla
Bank Spoldzielczy we Wschowie	bank_spoldzielczy_we_wschowie
Sulutgo 银行	Bank_Sulutgo
Sumselbabel 银行	Bank_Sumselbabel
Sumut 银行	Bank_Sumut

名称	ID
印度尼西亚伊斯兰教银行	银行_Syariah_印度尼西亚
国家储蓄银行	Bank_Tabungan_Negara
伊斯兰国家储蓄银行	Bank_Tabungan_Negara_Syariah
Torgoviy Kapital 银行	Bank_Torgoviy_Kapital
Uib 银行	Bank_Uib
Bank Veles	Bank_Veles
维多利亚银行	维多利亚银行
温得和克银行	温得和克银行
Yahav 银行	Bank_Yahav
尤格拉银行	Bank_Yugra
泽尼特银行	Bank_Zenit
Bank Zachodni WBK	bank_zachodni_wbk
银行 f. Karnten u。Steiermark	bank_f._karnten_u._steierMark
阿比西尼亚银行	阿比西尼亚银行
非洲银行	非洲银行
美国银行	美国银行
鞍山银行	鞍山银行
巴林和科威特银行	巴林银行和科威特银行
保定银行	保定银行
巴罗达银行	巴罗达银行

名称	ID
贝鲁特银行	贝鲁特银行
中旅银行	JZ
朝阳银行	朝阳银行
承德银行	承德银行
中国银行	中国银行
商业银行	商业银行
交通银行	交通银行
塞浦路斯银行	塞浦路斯银行
丹东银行	丹东银行
大同银行	大同银行
东亚银行	东亚银行
英格兰银行抵押贷款	英格兰银行抵押贷款
阜新银行	阜新银行
甘肃银行	甘肃银行
希腊银行	希腊银行
贵州银行	贵州银行
海南银行	海南银行
邯郸银行	邯郸银行
夏威夷银行	夏威夷银行
衡水银行	衡水银行

名称	ID
希望银行	希望银行
葫芦岛银行	葫芦岛银行
湖州银行	湖州银行
印度银行	印度银行
爱尔兰银行	爱尔兰银行
嘉兴银行	嘉兴银行
济宁银行	济宁银行
晋中银行	晋中银行
约旦银行	乔丹银行
开伯尔银行	开伯尔银行
基加利银行	基加利银行
昆仑银行	昆仑银行
马都拉银行	马都拉银行
马哈拉施特拉邦银行	马哈拉施特拉邦银行
墨尔本银行	墨尔本银行
蒙特利尔银行	蒙特利尔银行
南昌银行	南昌银行
纽约银行	纽约银行
宁夏银行	宁夏银行
俄克拉荷马银行	俄克拉荷马银行

名称	ID
盘新银行	Panhsin 银行
盘锦银行	盘锦银行
平顶山银行	平顶山银行
旁遮普银行	旁遮普银行
秦皇岛银行	秦皇岛银行
昆士兰银行	昆士兰银行
拉贾斯坦邦银行	拉贾斯坦邦银行
日照银行	日照银行
瑞丰银行	瑞丰银行
苏格兰银行	苏格兰银行
上海银行	上海银行
上饶银行	上饶银行
绍兴银行	绍兴银行
沙迦银行	沙迦银行
石嘴山银行	石嘴山银行
斯托克顿银行	斯托克顿银行
苏州银行	苏州银行
台州银行	台州银行
西藏银行	西藏银行
铁岭银行	铁岭银行

名称	ID
东京银行	东京银行
三菱东京银行	东京三菱银行
温州银行	温州银行
乌海银行	乌海银行
邢台银行	邢台银行
菲律宾群岛银行	菲律宾群岛银行
西方银行	西部银行
Bank we Wloszczowie	bank_we_wloszcZOWIE
BankAtlantic	BankAtlantic
BankMed	BankMed
BankUnited	BankUnited
Banka Celje	Banka_Celje
Banka Kombetare Tregtare	Banka_Kombetare_Tregtare
Banka Koper	Banka_Koper
Banka Kovanica	Banka_Kovanica
BankaSrpske	BankaSrpske
银行家人寿与意外伤害公司	Bankers_Life_&_Casualty_Co
班克豪斯奥古斯特·伦茨	Bankhaus_August_Lenz
班克豪斯尼尔迈耶	Bankhaus_Neelmeyer
班基亚	班基亚

名称	ID
Banknordik	Banknordik
Bankomat	Bankomat
Bankpime	Bankpime
班克韦斯特	班克韦斯特
半木园	半木园
旗帜银行	Banner_Bank
横幅 Health	旗帜_生命值
Banorte	Banorte
班帕拉	班帕拉
板秋山西餐厅	板秋山_西餐厅
Banque Al Mawared	Banque_Al_Mawared
联合银行	Banque_Allie
Banque Banorabe	Banque_Banorabe
CIO-BRO 银行	banque_cio-bro
Bernoise 州立银行	Banque_Cantale_Bernois
纳沙泰洛斯州立银行	Banque_Cantonale_Neuchâteloise
沃州州立银行	Banque_Cantale_Vaudoise
巴塞尔州立银行	banque_cantonale_de_Bále
弗里堡州立银行	Banque_cantale_de_Fribourg
日内瓦州立银行	Banque_Cantonale_de_Genève

名称	ID
汝拉州银行	Banque_cantonale_du_Jura
瓦莱州立银行	Banque_cantonale_du_Valais
突尼斯中央银行	Banque_Centrale_de_Tunisie
Banque Coop	Banque_Coop
阿尔及利亚外贸银行	Banque_extérieure_d'Algérie
黎巴嫩联邦银行	Banque_Federal_Liban
突尼斯法兰克银行	Banque_Franco-Tunisienn
Banque J. van Breda & Co	banque_j._van_breda_&_co
Banque Jammal 信托基金	Banque_Jammal_Trust
Banque Kolb	Banque_Kolb
Banque Laydernier	Banque_Laydernier
加拿大黎巴嫩银行	Banque_Libanaise_Canadien
黎巴嫩商业银行	Banque_libanaise_de_commerc
Banque Martin Maurel	Banque_Martin_Maurel
Banque Michel Inchauspe	Banque_Michel_Inchauspé
Banque Migros	Banque_Migros
Banque Misr Liban	Banque_Misr_Liban
国家银行	国家银行
阿尔及利亚国家银行	Banque_Nationale_d'Algérie
国家投资银行	Banque_Nationale_d'Investimes

名称	ID
Banque Nuger	Banque_Nuger
帕拉蒂尼银行	Banque_Palatine
Banque Pelletier	Banque_Pelletier
大众银行	大众银行
罗纳-阿尔卑斯银行	Banque_Rhone-Alpes
沙特法兰西银行	Banque_Saudi_Fransi
塔诺银行	Banque_Tarneaud
突尼斯-科威特银行	Banque_Tuniso-Koweitienne
Zitouna 银行	Banque_Zitouna
布列塔尼银行	Banque_de_Bretagne
法兰西银行	法国银行
萨瓦银行	banque_de_Savoie
叙利亚和黎巴嫩银行	banque_de_syri_et_et_du_Liban
突尼斯和阿联酋银行	banque_de_Tunisie_et_eds_emirats
人居银行	banque_de_lHabitat
Banque de l'Industrie et du Trava	banque_de_l'Industrie_et_du_Travail
留尼汪岛银行	Banque_de_la_Reunion
Banque des Mascareignes	banque_des_mascareignes
科威特银行和阿拉伯世界银行	banque_du_koweít_et_du_monde_arabe
Banque du Liban et du Golfe	banque_du_Liban_et_du_Golfe

名称	ID
黎巴嫩洲际银行	Banque_interContinentale_du_Liban
突尼斯团结银行	Banque_Tunisienne_de_Solidarité
Banregio	Banregio
班里咖啡	Banrie_Coff
Banrisul	Banrisul
Bansal 课程	Bansal_Class
播州信用金库	播州信用金库
Bantierra	Bantierra
Banumaodu 火锅	Banumaodu_Hot_Pot
宝丰药业	宝丰_药店
宝俊	宝俊
Baon	Baon
宝庆珠宝专卖店	宝清珠宝专卖店
包商银行	包商银行
Baoshifu 糕点	Baoshifu_糕点
包头农村商业银行	包头_农村_商业银行
宝应农村商业银行	宝应_农村_商业银行
baPR BANCA AGRICOLA POP。DI RAGUSA	bapr_banca_Agricola_pop。 _DI_RAGUSA
Bar Louie	Bar_Louie
Bara Zushi	Bara_Zushi

名称	ID
烧烤万凡庄	烧烤_Wan_Fanzhuang
烧烤之国	Barbeke_Nation
Barbeques Galore	Barbeques_Galore
Barberitos	Barberitos
Barbosa Supermercados	Barbosa_Supermercados
Barbour	Barbour
Barburrito	Barburrito
巴塞洛斯	巴塞洛斯
巴塞罗	巴塞罗
巴塞罗舒适酒店	巴塞罗_康福特
巴塞罗高级房	巴塞罗_Premium
Barcos & Rodados	Barcos_&_Rodados
巴多特	巴多特
少年 Bardot	Bardot_Junio
Bare Barn	Bare_Barn
Bare Escentuals	Bare_Escentuals
Bareburg	Bareburg
Bareeze	Bareeze
巴雷索咖啡	Baresso Coffee
巴富特和汤普森	Barfoot_&_Tompson

名称	ID
特价书	Bargain_Books
特价奥特莱斯	讨价还价_Outlet
巴里斯·格罗斯	Baris_Gross
咖啡师咖啡	咖啡师_咖啡
Bariuma	Bariuma
Bark Busters	Bark_Busters
Barkers	Barkers
Barkins	Barkins
巴克斯代尔 FCU	Barksdale_FCU
Barksole	Barksole
Barnardo's	Barnardo's
巴纳斯·胡斯	Barnas_Hus
Barnes Hus	Barnens_Hus
Barnes & Noble 书商	Barnes_&_Noble_Booksellers
巴恩斯与诺布尔教育	Barnes_&_Noble_Education
Barnetts	Barnetts
Barneys 纽约	Barneys_New_York
Barnhill 的	Barnhill 的
Barnie's	Barnie's
Baroda U.P. Bank	Baroda_U.P._Bank

名称	ID
男爵领地	男爵领地
Barratts 鞋子	Barratts_Shoes
Barred's	Barred's
Barrel '	Barrel '
巴雷特金融集团	Barrett_Financial_Group
巴里加佛得角	Barriga_Verd
巴里耶尔	巴里耶尔
巴洛克	巴洛克
Bart Smit	Bart_Smit
Bartell 毒品	Bartell_Dru
巴特利特树专家	Bartlett_Tree_专家
Barton's Lumber	Barton's_Lumber
Barçin Sport	Barçin_Spor
巴斯·范德海登	bas_van_der_Heijden
基础	基础
Basellandschaftliche Kantonalbank	Basellandschaftliche_KantonalBank
地下室系统	地下室系统
Basgimpa	Basgimpa
Bashas'	Bashas'
Bashfarmatsiya	Bashfarmatsiya

名称	ID
Bashkirneft'	Bashkirneft'
Bashkirnefteproduk	Bashkirnefteproduk
Bashmachok	Bashmachok
Bashmag	Bashmag
Bashneft'	Bashneft'
基本	基本
Basic-Fit	Basic-Fit
Baskin-Robbins	Baskin-Robbins
巴斯科	巴斯科
Basler Kantonalbank	Basler_KantonalBank
Baso A Fung	Baso_A_Fung
低音	低音
巴斯浆果和模拟人生	Bass_Berry_&_Sims
Bass Pro 商店	Bass_Pro_Shops
巴塞特家居摆设	Bassett_家居_家具
堡垒	堡垒
Bata	Bata
巴淡岛	巴淡岛
Bateel	Bateel
Batelco	Batelco

名称	ID
沐浴和身体护理	Bath_&_Body_Works
浴室奇异	浴室_Bizarre
蜡染布	蜡染布
Batik Keris	Batik_Keris
巴吞鲁日	巴吞鲁日
巴吞鲁日电信公司 FCU	baton_rouge_telco_FCU
Batriq	Batriq
Battard	Battard
电池+灯泡	电池+灯泡
电池中心	电池中心
电池医生	电池_医生
电池集市	Batter_Mart
《电池世界》	《电池世界》
BauProfi	BauProfi
Bauba Style	Bauba_Style
鲍金	鲍金
Bauspezi	Bauspezi
巴伐利亚公园车库	巴伐利亚州_Parkgaragen
托架音频	Bay_Audi
Bay 听力学	Bay_Audiolo

名称	ID
海湾贸易公司	Bay_Trading_Co
BayPort CU	BayPort_CU
BayWa	BayWa
Bayada 护士	Bayada_Nurses
Bayer Heritage FCU	拜耳_Heritage_FCU
Bayerische Landesbank GZ Nurbergh	Bayerische_LandesBank_GZ_Nürnberg
Bayfin	Bayfin
Baykal'skaya Regionalnaya	baykal'skaya_regionalnaya_kompaniya
BaykalBank	BaykalBank
贝加尔投资银行	贝加尔投资银行
贝兰兹家族 CU	Baylands_family_c
Bayleys	Bayleys
Baylor Scott & White H	Baylor_Scott_&_White_Health
Baymak	Baymak
贝蒙特套房酒店	Baymont_Inn_&_Suites
Bayo	Bayo
Bayou FCU	bayou_fcu
拜拉姆	拜拉姆
湾岸医疗保健	Bayshore_医疗保健
集市	集市

名称	ID
Bazar el Regalo	bazar_el_regalo
Bazhaiemo Zdorovia	Bazhaiemo_Zdorovia
Bazille	Bazille
贝拉披萨	Baíla_Pizza
Bañarte	Bañarte
西班牙对外银行省级银行	BBVA_Banco_Provical
Bbva Bancomer	Bbva_Bancomer
bbyy	bbyy
Bca	Bca
BciNova	BciNova
Be	Be
合而为一	Be_At_One
Be For Time 茶馆	Be_For_Time_Tea_House
Be Life 酒店	Be_life_Hotels
BeFree	BeFree
BelX	BelX
海滩市政 FCU	Beach_Municipal_FCU
信标	信标
信标与桥梁	Beacon_&_Bridge
信标社区 CU	Beacon_Community_cu

名称	ID
信标照明	信标照明
珠子瑜伽	Bead_Yoga
Bealls	Bealls
二号熊	Bear_Two
Beard Papa's	Beard_Papa's
野兽	野兽
比阿特丽斯酒店	Beatriz_Hoteles
博蒙特瓷砖	博蒙特瓷砖
Beaurepaires	Beaurepaires
BeautiControl 独立顾问	BeautiControl_独立顾问
美丽	美丽
美容品牌	美容品牌
美丽小屋	美容小屋
美容农场	Beauty_Farm
Beauty	Beauty_Fit
全天美容	Beauty_Full_Days
美人马克斯	Beauty_Max
美容狂热	美容狂热
海狸	海狸
Beaver Canoe	Beaver_Canoe

名称	ID
Bebe	Bebe
Bebek Goreng H Slamet	Bebek_Goreng_H_Slamet
Bebelu Sanduíches	bebelu_Sanduíches
贝卡拉	贝卡拉
Beccos	Beccos
贝克曼·库尔特	Beckman_Coulter
Bed Bath 及其他	Bed_Bath_&_Beyond
Bedas	Bedas
床上用品专家	床上用品_专家
Beden	Beden
Beds R Us	Beds_R_Us
美正香	Bee_Cheng_Hiang
牛肉 OBrady的	Beef_'s OBrady
Beefeater	Beefeater
Beehive FCU	beeHive_fcu
直线	直线
贝多芬与公司	Beethoven_&_Company
之前和之后	之前_&_之后
Beforly	Beforly
Begemot	Begemot

名称	ID
Begemot gipermarket igrushek	begemot_Gipermarket_igrushek
Begemotik	Begemotik
贝希托酒店	Behito_Hotel
北国购物城	北国_购物城
北国超市	北果_超市
北辉木瓜牛奶	Beihuei_Papaya_Milk
北京昌平包商银行	北京_昌平_包商银行
北京菜香根餐厅	Beijing_Choi_Heung_Root_Restauran
北京高华证券	北京_高华_证券
北京现代	北京_现代
北京金悦	北京_金悦
北京体检中心	北京_体检_考试_中心
北京农村商业银行	北京_农村_商业银行
北京同仁堂	北京_通_仁_唐
BeijingWatch	BeijingWatch
Beike	Beike
Beile 机器人	Beile_Robot
成为人类	Being_Human
Beipingshifu	Beipingshifu
北山	北山

名称	ID
北尚	北尚
Beisia Denki	Beisia_Denki
贝西亚美食中心	贝西亚食品中心
贝金斯·范·莱恩斯	Bekins_Van_Lines
Beko	Beko
Bel & Blanc	Bel_&_Blanc
Bel Air	Bel_Air
Bel Canto	Bel_Canto
Bel&Bo	Bel&Bo
BelCompany	BelCompany
BelSwissBank	BelSwissBank
BelTransOyl	BelTransOyl
BelveB	BelveB
Belagroprombank	Belagroprombank
白俄罗斯银行	白俄罗斯银行
Belc	Belc
百通珠宝商	Belden_Jewelers
贝尔多纳	贝尔多纳
Belfius	Belfius
Belfos & Trufas	Belfos_&_Trufas

名称	ID
Belgazprombank	Belgazprombank
比利时华夫饼	Belgian_Waffle
比利时 Loisirs	Belgique_Loisirs
相信	相信
贝宁投资银行	贝宁投资银行
Beliy Servis	Bely_Servis
Beliy Veter TSIFROVOY	bely_veter_tsiRovoy
Belk	Belk
贝尔	贝尔
贝尔大西洋	大西洋钟声
贝尔银行	Bell_Bank
BellSouth	BellSouth
BellStores	BellStores
贝拉唐娜	Bella_Donna
Bella Gula	Bella_Gula
贝拉意大利	贝拉_意大利
贝拉维斯塔汽车旅馆	Bella_Vista_Motel
Bellacino 的披萨和研磨机	Bellacino's_pizza_&_Grinders
贝拉弗洛拉	贝拉弗洛拉
Bellco CU	bellco_cu

名称	ID
Belle Tire	Belle_Tire
Bellerose	Bellerose
Bellinipasta	Bellinipasta
Bellona	Bellona
领头羊社区 CU	belwether_Community_cu
Bellwood CU 是 Henrico FCU 的一个分区	Bellwood_cu_a_division_of_henrico_fcu
贝尔马克特	贝尔马克特
贝尔蒙特饮料	贝尔蒙特_饮料
Belorusneft'	Belorusneft'
白俄罗斯银行 Malogo Biznesa	Belorusskiy_Bank_Malogo_Biznesa
白俄罗斯国民银行	Belorusskiy_Narodniy_Bank
加拿大贝伦	Belron_Canada
贝尔通助听器中心	Beltone_Hearing_Aid_Center
Beltone Hörberatung	Beltone_Hörberatung
Belvest	Belvest
Belyi Veter	Belyi_Veter
Bembos	Bembos
Bemol	Bemol
Ben & Jerry's	Ben_&_Jerry's
本·布里奇珠宝商	Ben_Bridge_Jewel

名称	ID
本·富兰克林商店	Ben_Franklin_Store
本·谢尔曼	Ben_Sherman
长凳	长凳
基准全球酒店业	基准_全球_酒店业
基准人员配备	基准_人员配备
本迪戈银行	Bendigo_Bank
Bendon	Bendon
贝内代蒂的	贝内代蒂的
受益金融	受益人_财务
优势	优势
贝纳通	贝纳通
孟加拉建议	孟加拉语_建议
Bengawan 个人咖啡	Bengawan _Solo_Coffee
蚌埠农村商业银行	蚌埠农村商业银行
贝尼哈纳	贝尼哈纳
Benit & Co	Benit_&_Co
贝尼顿	贝尼顿
本杰明富兰克林	Benjamin_Franklin_Pl
本杰明摩尔涂料	Benjamin_Moore_Paints
Benkuaisteak	Benkuaisteak

名称	ID
Benny&Co	Benny&Co
本绍德旅游	bensaude_Turismo
本西蒙	本西蒙
床上的 Bensons	Bensons_for _Beds
宾利皮革	Bentley_Leathers
便当人	Bento_Man
Benzin 大黄色出租车	Benzin_Big_Yellow_Taxi
Benzina	Benzina
苯甲醇	苯甲醇
Benzo	Benzo
Beobank	Beobank
Bepon	Bepon
Berco's	Berco's
Berel	Berel
贝伦尼斯	贝伦尼斯
Beres 多功能车	Beres_Multicars
Berezhnaya apteka	berezhnaya_apteka
Bergner's	Bergner's
Bergere de France	Bergère_de_France
伯汉国际银行	Berhan_国际银行

名称	ID
伯克希尔银行	Berkshire_Bank
Berlanda	Berlanda
柏林银行	柏林银行_Bank
Berliner Sparkasse	Berliner_Sparkasse
柏林大众银行	Berliner_Volksbank
Berluti	Berluti
伯纳德·菲利伯特	Bernard_Philibert
伯纳州立银行	Berner_KantonalBank
Bershka	Bershka
Bertoni	Bertoni
Bertucci's	Bertucci's
Besc	Besc
Bescar	Bescar
贝西尼	贝西尼
Besson 鞋子	besson_Chaussures
最佳	最佳
Best & Less	Best_&_Less
最佳 Al-Yousifi 电子产品	Best_Al-Yousifi_电子产品
最佳银行	Best_Bank
百思买	百思买

名称	ID
最好的租车	最佳汽车租赁
最佳剪辑	Best_Cuts
最佳教育奖	最佳_教育版
最佳金融信用合作社	最佳_金融_信用合作社
最佳健身	最佳健身
最佳加油站	Best_Fuel_Station
最佳家居和电气产品	最佳家居和电气产品
最佳酒店	最佳酒店
贝斯特旅馆	美国套房
最受喜爱	最受喜爱
最佳成绩	Best_Mark
最佳市场	最佳市场
最佳超市	Best_Mart
最好的一个	Best_One
Best Pals	Best_Pals
最佳地点	最佳地点
最佳研究	最佳研究
贝斯特韦斯特酒店	贝斯特韦斯特酒店
贝斯特韦斯特高级酒店	Best_Western_Plus
贝斯特韦斯特尊贵酒店	Best Western_Premier

名称	ID
迄今为止最好的市场	迄今为止最好的市场
停车效果最佳	最佳停车位
最优惠的价格	最优惠的价格
BestDrive	BestDrive
Bestay 快捷酒店	bestay_Hotel_Expr
Bestmart36	Bestmart36
Bestn	Bestn
良品铺子	良品铺子
百特咖啡馆竞技场	投注咖啡厅竞技场
测试版	测试版
测试版杂货店	Beta_Grocery_store
Betam	Betam
倍他佩特罗	倍他佩特罗
Bethowen	Bethowen
Bethpage FCU	bethpage_fcu
Betos	Betos
贝特西·约翰逊	Betsey_Joh
贝达电气	贝塔电气
商业改善局	Better_Business_Bure
更好的英语	更好的_英语

名称	ID
Better Life 超级超市	Better_life_Hyper-Mart
美好生活超级超市	Better_life_Super-Mart
更好的地方	Better_Place
BetterLife	BetterLife
Betts	Betts
Betts Kids	Bets_Kids
贝蒂·巴克莱	Betty_Barclay
BevMo!	BevMo!
Bever Zwerfsport	Bever_Zwerfsport
Bevilles	Bevilles
Bewise	Bewise
贝门	贝门
超越银行	Beyond_Bank
Beyza Pilic	beyza_Pilic
Reichenau Bezirksparkasse	Bezirksparkasse_Reichenau
Bf Bancofácil	Bf_Bancofacil
Bg Bank	Bg_Bank
Bgn	Bgn
巴拉特合作银行	Bharat_co-op_Bank
巴拉特婚姻	Bharat_Martimony

名称	ID
巴拉特国立公立学校	巴拉特国家公立学校
巴拉特海外银行	Bharat_Overseas_Bank
巴拉特石油公司	Bharat_Petroleum
Bharkadevi 冰淇淋	Bharkadevi_Ice_Cream
Bhatia 通讯	Bhatia_通讯
Bhima 珠宝商	Bhima_Jewelers
Bhojohori Manna	Bhojohori_Manna
Bhpetro	Bhpetro
Bhpetromar	Bhpetromar
Bhs	Bhs
Bhw	Bhw
Bi Rite Elect	Bi_Rite_Electric
Bi-Mart	Bi-Mart
Bi-Mart 药房	Bi-Mart_药房
BiBa	BiBa
大	大
BilStore	BilStore
Bian 教育	边_教育
BianYiFang	BianYiFang
边立峰	边立峰

名称	ID
Bibigo	Bibigo
藏书城	藏书城
Bibliosfera	Bibliosfera
Bibo Mart	Bibo_Mart
Bic 联系人	Bic_联系方式
Bic Shuhan	Bic_Shuhan
Bicentenario	Bicentenario
Bidvest	Bidvest
Biedronka	Biedronka
Bielefelder Volksbank	Bielefelder_Volksbank
Biesiadowo	Biesiadowo
Bieszczadzka SKOK	bieszczadzka_skok
比耶夫	比耶夫
Bifanas de Vendas Nova	bifanas_de_vendas_novas
Bifarma	Bifarma
Big 10 Mart	Big_10_Mart
十大轮胎	Big_10_Tires
大苹果	Big_Apple
大苹果百吉饼	Big_Apple_Bagels
Big Apple 甜甜圈和咖啡	Big_Apple_Donuts_&_Coffee

名称	ID
大爆炸	Big_Bang
大篮子	大篮子
大集市	Big_Bazaar
大笨钟	Big_Ben
Big Big Boon	Big_Big_Boon
大咬一口	Big_Bite
大男孩	Big_Boy
大 Brothers/Big 姐妹	大兄弟/大姐妹
大捆绑包	Big_Bundle
Big C 额外版	Big_C_Extra
Big C 市场	Big_C_Market
Big C 手机	Big_C_Mobiles
Big C 超级中心	Big_C_Supercenter
大相机	Big_Camera
大厨们	Big_Chefs
Big Chick	Big_Chick
大电影院	Big_Cinema
Big D	Big_D
大爸爸	Big_Daddy
大狗	Big_Dogs

名称	ID
大费尔南德	Big_Fernand
大脚怪	Big_Foot
Big H	Big_H
大角牛排馆	Big_Horn_Steak_House
大岛 FCU	big_Island_fcu
大约翰	Big_John
大批量	Big_Lots
Big Mama	Big_Mama
大集市	Big_Mart
Big O 轮胎	Big_O_Tires
大药房	大药房
大披萨	大披萨
大红商店	Big_red_Stores
大笔储蓄家具	Big_Save_Furniture
大城英雄	Big_Town_Hero
大火车	Big_Train
Big X Picanha	Big_X_Picanha
Big Y	Big_Y
Big-A	Big-A
BigMat	BigMat

名称	ID
比格比咖啡	Biggby_Cof
Bigliquy	Bigliquy
碧珠碧姬	Bijou_Brigitte
Bijou Brio	Bijou_Brio
Bijoux España	Bijoux_España
比卡内尔瓦拉	比卡内尔瓦拉
Bike Barn	Bike_Barn
比基尼村	比基尼村
彼尔德伯格	彼尔德伯格
Bildirici	Bildirici
Bilgisayar Hastanesi	Bilgisayar_Hastanesi
Bilisten	Bilisten
比尔·米勒 Bar-B-Q	Bill_Miller_Bar-B-Q
Bill's	Bill's
比尔美元专卖店	Bill's Dollar_Stores
BillaPlus	BillaPlus
死水潭	死水潭
Billabong 高等国际学校	Billabong_High_国际学校
Billings FCU	Billings_fcu
十亿	十亿

名称	ID
Bilskadesenteret	Bilskadesenteret
Biltema	Biltema
比尔特莫尔酒店	Biltmore_Hotel
Bilxtra	Bilxtra
Bim	Bim
Bimeks	Bimeks
Bin Inn	Bin_Inn
Binalot	Binalot
绑定巧克力	Bind_Chocolate
Binder Optik	Binder_Optik
Binfenniao	Binfenniao
李冰	Bing_Lee
姜冰冰	姜冰冰
宾果游戏	宾果游戏
秉胜	秉胜
滨海轨道交通	滨海_公共交通
宾斯旺格玻璃	Binswanger_Glass
时尚之星	宾堂_时尚
生物公司	生物公司
生物节奏学院	Bio_Ritmo_Academ

名称	ID
Bio c'Bon	bio_c'_bon
BioLayf-Express	BioLayf-Express
BioPharma	BioPharma
BioTech美国	BioTech美国
Biocon	Biocon
Biofarma	Biofarma
生物地毯	生物地毯
Biomed	Biomed
Biosfera	Biosfera
Bipop Carire	Bipop_Carire
鸟类德比	Birds_Derby
伯肯斯托克	伯肯斯托克
Birla 人寿保险	Birla_Life_Insur
Birla Sun Life	Birla_Sun_Life_Insuran
伯明翰中郡	伯明翰_米德郡
伯明翰-布卢姆菲尔德 CU	伯明翰-布卢姆菲尔德_CU
生日	生日
尾西信用金库	Bisai_Shinkin_Bank
BiscoMisr	BiscoMisr
饼干维尔	饼干维尔

名称	ID
Bislet Kebab House	Bislet_Kebab_House
Bismillah 餐厅	Bismillah_Resta
Bisse	Bisse
Bissu	Bissu
西拉诺小酒馆	Bistro_Cyrano
小酒馆 Pierre	小酒馆_Pierre
位	位
BitPhone	BitPhone
咬一口	咬一口
Bitel	Bitel
Biti's	Biti's
Biutiss Next Level 美学	Biutiss_Next_Level_Estic_
Biyesen Cigköfte	biyesen_Cigköfte
Biyide	Biyide
Bizim Toptan Satis 杂志社	bizim_Toptan_Satis_Magazalari
Bizzarro	Bizzarro
比约恩·博格	Björn_Borg
Björklund	Björklund
黑安格斯牛排馆	黑安格斯牛排馆
黑熊餐厅	Black_Bear_Diner

名称	ID
黑桥食品	Black_Bridge_
黑咖啡画廊	黑咖啡画廊
黑山 FCU	black_Hills_FCU
黑色鞋带	Black_Laces
黑胡椒	黑胡椒
Black Steer	Black_Steer
黑色和棕色	黑色和棕色
布莱克的	布莱克的
黑眼豌豆	black-eyed_pea
BlackBall	BlackBall
黑胡子的海鲜岛	黑熊海鲜岛
Blackberry	Blackberry
Blackberrys	Blackberrys
二十一点披萨	二十一点_披萨
Blackmon 系泊处	Blackmon_Moorin
遮蔽	遮蔽
黑石河 FCU	blackstone_river_FCU
黑木保护者	Blackwoods_
布莱恩的农场和舰队	Blain's_farm_&_Fleet
Blake Cassels & Graydon LLP	blake_cassels_&_graydon_LLP

名称	ID
Blake's Lotaburger	Blake's_lotaBurger
Blanc du Nil	blanc_du_nil
布兰科	布兰科
布兰科·基里基	Blanko_Kiriki
Blau Verd 酒店	Blau_Verd_Hotels
Blaze 披萨	Blaze_Piz
Blazek	Blazek
Bleecker 面包店	Bleecker_Bakery
Blendz Café	Blendz_Café
祝福	祝福
Bleu Libellule	Bleu_Libellule
Bleu Marine	Bleu_Marine
Blimpie	Blimpie
BlinBeri	BlinBeri
盲课	盲课
闪烁充电	闪烁_正在充电
闪烁光学	闪烁_光学
盲目关闭	盲目关闭
幸福蛋糕	幸福蛋糕
极乐世界	幸福世界

名称	ID
Blits-Treyd	Blits-Treyd
Bloch	Bloch
布洛克之家	Block_Hou
轰动一时	轰动一时
大片娱乐	大片_娱乐
Blom Bank	Blom_Bank
埃及布洛姆银行	Blom_Bank_Egypt
Blommers 冰淇淋	Blommers_Ice_Cream
Blomsterlandet	Blomsterlandet
Bloom	Bloom
Bloomingdale	Bloomingdale
花开《化学家》	Bloos_The_Chemist
花间山酒店及度假村	Blossom_Hill_酒店及度假村
BluCurrent CU	BluCurrent_CU
蓝带酒店	蓝带酒店
蓝瓶咖啡	蓝瓶咖啡
蓝盒子	蓝色盒子
蓝十字和蓝盾	蓝十字和蓝盾
蓝色能量	蓝色能量
蓝色 FCU	blue_fcu

名称	ID
蓝色地平线酒店	蓝色地平线酒店
蓝海国际大饭店	蓝色地平线国际酒店
蓝海君华大饭店	蓝色地平线_君华_大酒店
蓝色幻觉	蓝色_幻觉
蓝色公司	Blue_Inc
蓝海酒店及度假村	Blue_Sea_酒店_度假村
蓝海豹	蓝色封印
蓝天	蓝天
蓝树	蓝树
蓝白服务	蓝白服务
BlueBay 酒店	BlueBay_酒店
BlueMobility	BlueMobility
BlueSG	BlueSG
Bluelife 电池	Bluelife_Batter
Blume 2000	Blume_2000
Blumen	Blumen_Risse
Blunives	Blunives
Blunts 鞋	Blunts_Shoes
Blz	Blz
Bm Bank	Bm_Bank

名称	ID
Bo 概念	Bo_Conce
Bo 的咖啡	bo's_coffe
Bo-rent	Bo-rent
BoBaPop	BoBaPop
BoConcept	BoConcept
BoRics	BoRics
船面条	Boat_Noodle
鲍勃埃文斯	Bob_Evans
Bob Jane T-Marts	Bob_Jane_T-Marts
Bob Sumerel 轮胎与服务	Bob_Sumerel_Tire_&_Service
Bob's	Bob's
Bob 的大男孩	Bob's_big_Boy
Bob's 折扣家具	Bob's_Discount_Furniture
Bob's Stores	Bob's Stores
Bobbi Brown	Bobbi_Brown
Bobis	Bobis
Bob 的汉堡和啤酒	Bob's_Burger_and_Brew
中银三星人寿	Boc-Samsung_Life
Bocados	Bocados
Bocage	Bocage

名称	ID
博卡琳达	博卡琳达
博卡塔	博卡塔
Bochane Rent	Bochane_Rent
Bochonok	Bochonok
Bodega Aurrera	Bodega_Aurrera
Aurrera Express 酒窖	Bodega_Aurrera_Express
身体中央	车身中央
身体手套	身体手套
body'minut	body'minut
腰铃	腰铃
身体对话	身体对话
车身科技	车身科技
波音员工信用合作社	波音_员工_信用联盟
Boekenvoordeel	Boekenvoordeel
Boerenbond	Boerenbond
Bog & idé	bog_&_idé
Bogerud Texstil	Bogerud_Texstil
Bogner	Bogner
渤海财产保险	渤海_财产_保险
渤海证券	渤海证券

名称	ID
Bohus	Bohus
沸点	Boiling_Point
Bojangles'	Bojangles'
Bokia	Bokia
Boklageret	Boklageret
Boknäs	Boknäs
Bolia	Bolia
波拉市场	Bolla_Marke
Bolle	Bolle
Boloco	Boloco
Boloni	Boloni
Bolsos Peyo	Bolsos_Peyo
Bolulu Hasan Usta	Bolulu_Hasan_Usta
Bolwell	Bolwell
Bom Preço	Bom_Preço
孟买香草和香料	Bombay_Herbs_&_Spices
孟买商业合作银行	Bombay_mercantile_Co-op_Bank
Bomi	Bomi
Bon Appetit	Bon_Appetit
Bon Bon	Bon_Bon

名称	ID
Bon Maison	Bon_Maison
Bon Marche	Bon_Marche
Bon Marché	Bon_Marché
邦沃思	Bon_Worth
Bon-Ton	Bon-Ton
BonChon 鸡	BonChon_鸡
Bona Compra	Bona_Compra
Bonafide	Bonafide
鸿运	鸿运
Bonanza 服装	Bonanza_服装
波拿巴	波拿巴
债券赢家教育	债券获奖者_教育
债券+债券	债券+债券
保税油	Bonded_Oil
Bone's	Bone's
Bonefish 烧烤	Bonefish_Grill
Bonfare 市场	Bonfare_Mark
邦戈	邦戈
博尼	博尼
博尼亚	博尼亚

名称	ID
博尼塔	博尼塔
你好	你好
Bonnie&Read	Bonnie&Read
Bono Penus	Bono_Pneus
Bono's Pit Bar-B-Q	bono's_pit_bar-b-q
Bonobo	Bonobo
邦波因特	邦波因特
Bonpreu	Bonpreu
盆景酒店	Bonsai_Hotel
奖金	奖金
图书市场 A 也是	Book_Market_A_Too
书架	Book_Rack
预订 Smile	Book_Smile
读书世界	Book_World
Booker Cash & Carry	Booker_Cash_&_Carry
预订	预订
Bookline	Bookline
Books-A-Million	Books-A-Million
自作自受	自作自受
Boons Markt	Boons_Markt

名称	ID
Boost 果汁棒	Boost_Juice_Bars
Boost移动版	Boost_Mobile
强效果汁	Booster_Juice
Boot Barn	Boot_Barn
盗版者	盗版者
靴子 Apotek	Boots_Apotek
靴子药房	靴子_药房
让化学家们大吃一惊	Boots_the_chemists
Booz 时尚酒店	Booz_Fashion_Hotels
Bora 韩国超市	Bora_Korean_Mart
边框	边框
Bordesholmer Sparkasse	Bordesholmer_Sparkasse
博格沃德	博格沃德
Borjan	Borjan
博克	博克
波罗托当铺	Boroto_Pawn_Shop
博罗沃	博罗沃
博尔曼酒店	Borrman_Hotel
Borsache	Borsache
Borsod Takarék	Borsod_Takarék

名称	ID
Boréa	Boréa
Bosa Donuts	Bosa_Donuts
博世	博世
Bosco Sport	Bosco_Sport
Boscolo	Boscolo
博斯科夫的	博斯科夫的
百色	百色
Bose 零售店	Bose_Retail_Store
Bosicor	Bosicor
波司登	波司登
波斯尼亚国际银行	波斯尼亚_国际银行
廖老板	boss_liao
Bosselman	Bosselman
波士顿	波士顿
波士顿咨询集团	波士顿咨询集团
波士顿市场	波士顿市场
波士顿披萨	波士顿_披萨
波士顿私人银行	波士顿私人银行
波士顿店	波士顿专卖店
波士顿的餐厅和体育酒吧	波士顿_餐厅_&_Sports_Bar

名称	ID
植物学的	植物学的
Boteju	Boteju
Boteko	Boteko
Boticas Arcángel	Boticas_Arcángel
Boticas Felicidad	Boticas_Felicidad
Boticas Hogar y SALUD	Boticas_Hogar_y_Salud
Boticas y Salud	Boticas_y_salud
Botika ng Bayan	botika_ng_bayan
博通教育	Botong_教育版
Bottega Veneta	Bottega_Veneta
Botterills	Botterills
Botticelli	Botticelli
Bottlemart	Bottlemart
底部美元食品	底部_美元_食物
Boubyan Bank	Boubyan_Bank
Boucheron	Boucheron
Boulanger	Boulanger
波旁威士忌 Hipermercado	Bourbon_Hipermercado
Boursorama	Boursorama
Bebé 精品店	Boutique_Bebé

名称	ID
Relógios 精品店	Boutique_dos_relógios
布伊格电信	Bouygues_Teleco
Bouza Roll	Bouza_Roll
Bowen	Bowen
鲍恩斯	鲍恩斯
Bowlero	Bowlero
Bowlmor Lanes	Bowlmor_Lanes
Box	Box
拳击手超市	拳击手_超市
美国童子军	美国童子军_of_America
Boyd's 轮胎与服务	Boyd's_tire_&_Service
Boyner	Boyner
男孩和女孩俱乐部	Boys_&_Girls_Club
欧洲男孩	Boys_Of_Europe
Bozuyuk Sut Urunleri	Bozuyuk_Sut_Urunleri
Bozzy	Bozzy
英国石油公司-中石化	英国石油公司-中石化
Bpe	Bpe
Bpet	Bpet
布拉德斯科	布拉德斯科

名称	ID
布拉德斯科 Prime	Bradesco_Prime
布拉德福德和宾利	Bradford_&_Bingley
布拉德利	布拉德利
Bragg Mutual FCU	bragg_mutual_fcu
Brahmadeodada Mane Sahakari Bank	Brahmadeodada_Mane_Sahakari_Bank
脑软件	脑软件
刹车大师	Brake_Masters
Breakes	Brakes_Plus
品牌工厂	品牌_工厂
品牌油	品牌_Oil
勃兰登堡银行	勃兰登堡银行
布兰迪利	布兰迪利
少花钱买品牌	Brands_For_Less [服装和配饰] _0
少花钱买品牌	Brands_For_Less [服装和配饰] _70
品牌奥特莱斯	品牌_奥特莱斯
布兰德斯塔	布兰德斯塔
Brandywine 生活	白兰地酒_Living
Bras N Thin	Bras_N_Things
布拉特帕克	布拉特帕克
Braum's	Braum's

名称	ID
Braun Buffel	Braun_Buffel
博朗家具中心	Braun_Mobel_Center
Braunegger	Braunegger
Braunschweigische Landessparkasse	Braunschweigische_Landessparkasse
Bravissimo	Bravissimo
肆无忌惮的头	Brazen_Head
巴西美女	巴西人_美女
巴西蝴蝶	巴西人_蝴蝶
Brd	Brd
胡言乱语	Bread_Talk
BreadTalk	BreadTalk
Breadeaux 披萨	面包_披萨
面包生活	面包生活
面包匠	面包匠
面包顶	面包顶
休息时间	休息时间
宝爵	宝爵
布雷默	布雷默
Bremer Landesbank KA 奥尔登堡	Bremer_LandesBank_ka_Oldenburg
Bremische Volksbank	Bremische_Volksbank

名称	ID
布伦特原油	Brent_Oil
Breo	Breo
Bresler 的冰淇淋	Bresler's Ice_cream
Bretagne Matériaux	Bretagne_Matériaux
布雷塔斯	布雷塔斯
Brewberrys 咖啡厅	Brewberrys_Cafe
啤狗	啤狗
Brewer FCU	Brewer_fcu
Brewers Fayre	Brewers_Fayre
Brewingz 体育酒吧和烧烤店	Brewingz_Sports_Bar_Grill
布莱恩红茶	Brian_Black_Tea
砖块	砖块
Brico	Brico
Brico Cash	Brico_Cash
Brico Depôt	Brico_Depôt
Brico IO	brico_io
Bricobox	Bricobox
Bricoma	Bricoma
Bricomarché	Bricomarché
Bridge	桥接_cu

名称	ID
普利司通	普利司通
Brigetio 变更	Brigetio_Change
明亮	明亮
光明地平线	Bright_Horizon
现在很亮！牙科	Bright_Now！ _牙科
明智证券	Bright_Smart_证券
BrightHouse	BrightHouse
BrightStar CU	BrightStar_CU
布莱顿收藏品	Brighton_收藏品
Brightview FCU	BrightView_FCU
Brillant	Brillant
Brilleland	Brilleland
辉煌	辉煌
太棒了	太棒了
Brillux	Brillux
Brio	Brio
布里奥尼	布里奥尼
Briscoes	Briscoes
布里斯托尔	布里斯托尔
布里斯托尔和西部	Bristol_&_West

名称	ID
布里斯托农场	Bristol_Farms
布里斯托尔'	布里斯托尔'
英国人	英国人
布里特酒店	Brit_Hotel
不列颠尼亚建筑协会	不列颠尼亚建筑协会
不列颠尼亚工业公司	不列颠尼亚_工业
英国航空	英国航空
英国书店和文具店	British_Bookshop_&_Stationers
英国燃料有限公司	英国_Fuels_Limited
英国心脏基金会	英国心脏基金会
英国医院	英国医院
英国减肥诊所	英国_Slimming_Clinic
英国游泳学校	英国游泳学校
BritishGas	BritishGas
Broadview 安全	广视_安全
Brocard	Brocard
Brokavia	Brokavia
Brokbiznesbank	Brokbiznesbank
破碎的蛋黄咖啡厅	Broken_Yolk_Cafe
Bronnitskiy yuvel	Bronnitsky_Yuvelir

名称	ID
Bronstein Medicina Diagnóstica	Bronstein_Medicina_Diagnóstica
布鲁克	布鲁克
布鲁克代尔	布鲁克代尔
布鲁克林银行	布鲁克林银行
布鲁克斯兄弟	布鲁克斯兄弟
布鲁克斯药房	Brooks_药房
布鲁克斯菲尔德	布鲁克斯菲尔德
小布鲁克斯菲尔德	Brooksfield_Jr.
布鲁克希尔兄弟	布鲁克希尔兄弟
布鲁克希尔的	布鲁克希尔的
布鲁克希尔药房	布鲁克希尔药房
布鲁克斯通	布鲁克斯通
兄弟汉堡	兄弟汉堡
布朗的	布朗的
Brown's Shoe	Brown'shoe_fit
布朗尼和唐宁饼	Brownies_&_Downies
Brsm	Brsm
布鲁斯·菲尔德	布鲁斯_菲尔德
Bruegger's	Bruegger's
Brumby's	Brumby's

名称	ID
Bruna	Bruna
巴黎 Brunbrun	布伦布鲁恩_巴黎
布鲁内洛·库奇内利	Brunello_Cucinelli
Brunet	Brunet
Bruno Caffe	Bruno_Caffe
不伦瑞克区	不伦瑞克区
Bruster 的真冰淇淋	Bruster's real_ice_cream
Bryant's	Bryant's
Bryc	Bryc
Brétéché	Brétéché
Bsb	Bsb
Bstdb	Bstdb
首旅南苑酒店及度假村	首旅-南苑_酒店及度假村
Btp Banque	Btp_Banque
Buba	Buba
Bubba Gump Shrimp Co.	Bubba_Gump_Shrimp_Co.
BubbleZ 敬畏	bubblez_AWE
气泡学	气泡学
buc-ee's	buc-ee's
Buca di Beppo	buca_di_beppo

名称	ID
Buccheri	Buccheri
Buchbinder	Buchbinder
巴克的披萨	Buck's_Pi
七叶树州立大学	buckeye_state_cu
Bucking B	Bucking_Bul
搭扣	搭扣
Bucodente	Bucodente
布库鲁克	布库鲁克
Bud'Zdorov	Bud'_Zdorov
Bud'te Zdorovy	bud'te_zdorovy
布达佩斯银行	布达佩斯银行
兄弟们	兄弟们
Buddy	Buddy
Buddy's 家居摆设	好友家居摆设
Buderus	Buderus
布根斯	布根斯
Budget	Budget
廉价百叶窗	预算_百叶窗
廉价排气和轮胎	预算_排气_轮胎_和_轮胎
廉价食品	预算_食物

名称	ID
廉价房东旅馆	Budget_Host_In
廉价汽油	预算_汽油
Budgetel 旅馆	Budgetel_Inn
Budu Mamoy	Budu_Mamoy
Buehlers 新鲜食品	Buehlers_Fresh Food
Bueno Pitta	Bueno_Pitta
布法罗汉堡	水牛城汉堡
布法罗交易所	水牛交易所
布法罗狂野之翼	Buffalo_Wild_Wil
布法罗之翼和戒指	Buffalo_Wings_&_Ring
Buffalucas	Buffalucas
Bufkes	Bufkes
Bufori	Bufori
错误	错误
Bugaboo Creek 牛排馆	Bugaboo_Creek_Steak_House
Bugadry	Bugadry
布加迪	布加迪
布加迪 Taste Café	Bugatti_Taste_Café
建造它	Build_it
建筑商快车	建筑商_Express

名称	ID
建筑商 FirstSource	Builders_ FirstSource
建筑商贸易仓库	建筑商_贸易仓库
BuildingStars	BuildingStars
Bukva	Bukva
bukvoed	bukvoed
保加利亚银行	宝格银行
宝格丽	宝格丽
烤肉兄弟	Bulgogi_Broth
散装谷仓	Bulk_Barn
斗牛士牛排馆	Bull_Fighter_Steak_House
Bull 酒店	Bull_酒店
斗牛	斗牛
牛蛙爱情虾	Bullfrog_Love_Shrim
Bulochnye Vol'cheka	bulochnye_vol'cheka
Bumba	Bumba
Bumbu Desa	Bumbu_Desa
保险杠到保险杠汽车零部件	Bumper_to_Bumper_Auto_Parts
Bumpers Drive-In	Bumpers_Drive-In
Bungotakada dori Sakaba	Bungotakada_dori_Sakaba
船用油	bunker_Oil

名称	ID
邦纳国际银行	Bunna_国际_银行
邦宁斯	邦宁斯
Bunnpris	Bunnpris
邦蒂集团	Bunty_Group
Buondi	Buondi
保柏	保柏
Bureau Val	Bureaux_Val
Burger 21	Burger_21
汉堡边缘	Burger_Edge
汉堡燃料	Burger_Fuel
汉堡酒廊	汉堡酒廊
完美汉堡	汉堡_完美
汉堡街	Burger_Street
威斯康星州	汉堡_威斯康星州
BurgerFi	BurgerFi
汉堡燃料	汉堡燃料
Burgerim	Burgerim
汉堡屋	Burgering_House
Burgerland	Burgerland
汉堡吧	Burgers_Bar

名称	ID
伯格维尔	伯格维尔
Burj Bank	Burj_Bank
Burlex	Burlex
伯灵顿	伯灵顿
伯灵顿北部铁路	伯灵顿_北方_铁路
伯灵顿鞋	Burlington_Shoes
Burmah	Burmah
Burn 新兵训练营	Burn_Boot_Camp
烧油	Burn_Oil
Burrito Boyz	Burrito_Boyz
Burros con Alas	burros_con_alas
Bursa Iskender	Bursa_Iskender
Bursa Kebap Evi	Bursa_Kebap_Evi
Bursa Kumas Pazari	Bursa_Kumas_Pazari
Burson 汽车零部件	Burson_Auto_Parts
伯顿	伯顿
釜山汽车金融	釜山_汽车金融
釜山银行	釜山银行
Busey Bank	Busey_Bank
商务汽车租赁	商务车_租车

名称	ID
Buslik	Buslik
Busy Beave	Busy_Beave
Busy Beaver	Busy_Beaver
但是	但是
屠夫男孩	Butcher_Boys
布泰拉市场	Butera-Market
男管家	男管家
管家	管家
管家咖啡厅	ButlersCafé
黄油 a Lee	butter_a_lee
巴特菲尔德面包店	Butterfield_B
Buways	Buways
低价购买食物	Buy-Low_Foods
立即购买	立即购买
Buyrite	Buyrite
嗡嗡声	嗡嗡声
作者：Dziubeka	作者：_Dziubeka
作者：Mascota	By_Mascota
作者：网球	By_Tennis
在赛纳河边	By_the_Saina_River

名称	ID
Bybo 牙科	Bybo_Dental
Byco	Byco
比亚迪	比亚迪
byelarusnyeft'	byelarusnyeft'
byelnyeftyekhim	byelnyeftyekhim
Byggern	Byggern
Byggkjöp	Byggkjöp
Byggmakker	Byggmakker
Bygma	Bygma
拜林银行	Byline_Bank
Bymovil	Bymovil
拜雷多	拜雷多
Byrne Dairy	Byrne_Dairy
Byrne 乳品店	Byrne_Dairy_Stores
拜伦	拜伦
BystroBank	BystroBank
byust'e	byust'e
BOBOLI	BOBOLI
Bardi Auto	Bardi_Auto
Backerei & Konditorei Thürmann	Backerei_&_Konditorei_Thürmann

名称	ID
Backerei Brunner	Backerei_Brunner
Backerei Büsch	Backerei_Büsch
Backerei Evertzberg	Backerei_Evertzberg
Backerei Gillen	Backerei_Gillen
Backerei Heitzmann	Backerei_Heitzmann
Backerhaus Veit	Backerhaus_Veit
Bebé Cash	Bebe_Cash
Bonus	Bonus
C & M 公司	C_&_M_Company
C E S CU	C_E_S_CU
C 快车	C_Expres
C Spire Wireless	C_Spire_Wireless
C Town 超市	C_Town_超市
C upC	c_upc
C&A	C&A
C&C	C&C
C&C 卡车	C&C_TRUCKS
C+C Schaper	C+C_Schaper
C-Space	C-Space
C-Store	C-Store

名称	ID
C. Rolim	C._Rolim
C.Salon	C.Salon
C.Straits Cafe	C. Straits_cafe
C: Calzatodo	C: Calzatodo
CA Crédito Agrícola	ca_credito_Agrícola
CA'NOVA	CA'NOVA
CAA	CAA
CAAR	CAAR
出租车旅馆	出租车旅馆
CABS银行	Cabs_Bank
CACO	CACO
凯撒体育博彩	凯撒_体育博彩
CAFAM	CAFAM
COTON 咖啡馆	CAFE_COTON
咖啡馆皇冠	咖啡皇冠
路易斯咖啡馆	路易斯咖啡馆
MANGOSIX 咖啡馆	CAFE_MANGOSIX
红咖啡	咖啡红色
CAFFE FRAPPESSA	CAFFE_FRAPPESSA
CAI	CAI

名称	ID
CAI 服装店	cai_clothing_stor
CAISSE D'EPARGNE	CAISSE_D'ÉPARGNE
财通证券	财通证券
莱里亚信用卡	CAIXA_DE_CRÉDITO_DE_LEIRIA
CAIXA 加利西亚	CAIXA_GALICIA
CAIXA Ontinyent	caixa_ontinyent
农村储蓄银行	CAIXA_RURAL
CAIXA 农村布里亚纳	CAIXA_RURAL_BURRIANA
阿尔库迪亚乡村储物柜	CAIXA_RURAL_DE_L'ALCUDIA
CAIXA RUL	CAIXA_RURAL_TORRENT
CAIXAGUISSONA	CAIXAGUISSONA
CAIXALTEA	CAIXALTEA
CAJA CANTABRIA	CAJA_CANTABRIA
卡哈格拉纳达	CAJA_GRANAD
墨西哥流行的盒子	CAJA_POPULA_MEXIC
CAJA RURAL	CAJA_RURAL
CAJA 农村布尔戈斯	CAJA_RURAL_BURGOS
CAJA RURAL 卡斯蒂利亚-拉曼查	CAJA_RURAL_CASTILLA-LA_MANCHA
CAJA CENTRAL	CAJA_RURAL_CENTRAL
BURGOS RURAL CAJA	CAJA_RURAL_DE_BURGOS

名称	ID
埃斯特雷马杜拉乡村民宿	CAJA_RURAL_DE_EXTREMADURA
CAJA RURAL DE GIJON	CAJA_RURAL_DE_GIJON
CAJA RURAL DE NAVARRA	CAJA_RURAL_DE_NAVARRA
CAJA RURAL DE SEGOVI	CAJA_RURAL_DE_SEGOVIA
CAJA RURAL DE VILLA	CAJA_RURAL_DE_VILLAR
CAJA RURAL DEL SU	CAJA_RURAL_DEL_SUR
格拉纳达乡村	CAJA_RURAL_GRANADA
CAJA RULAR SAN JOSE DE ALCORA	CAJA_RURAL_SAN_JOSE_DE_ALCORA
卡哈圣费尔南多	CAJA SAN_FERNANDO
CAJAMURCIA	CAJAMURCIA
CALIDA	CALIDA
CALZACOSTA	CALZACOSTA
CALZADO BUCARAMANGA	CALZADO_BUCARAMANGA
CALZADO NUEVA MODA	CALZADO_NUEVA_MODA
CALZADOS F. GAMEZ	CALZADOS_F._GAMEZ
CALZEDONIA	CALZEDONIA
CAMC	CAMC
CAMER	CAMER
大卫营	大卫营地
CAN	CAN

名称	ID
CANADIA	CANADIA
加拿大二比一披萨	CANADIAN_2_FOR_1_PIZZA
运河概念	运河概念
CANDIA STROM	CANDIA_STROM
CAP COM FCU	CAP_COM_FCU
CAPAROL	CAPAROL
开普敦鱼市场	开普敦鱼市场
开普联合集市	CAPE_UNION_MART
CAPESTORM	CAPESTORM
首都	首都
CAPPELLO	CAPPELLO
卡普里	卡普里
汽车美容师	汽车美容师
汽车美容专业人士	CAR_BEAUTY_PRO
汽车质量汽车	汽车_质量_汽车
汽车服务城市	汽车服务城市
洗车	洗车
卡拉·克鲁兹	卡拉_CRUZ
CARAMELO	CARAMELO
卡拉维拉·塞古罗斯	CARAVELA_SEGURO

名称	ID
护理	护理
货物	货物
CARIFAC	CARIFAC
CARIFANO	CARIFANO
CARIFE	CARIFE
CARINA	CARINA
CARIPARMA 农业信贷银行	CARIPARMA_CREDIT_AGRICOLE
CARIPLO	CARIPLO
CARIPRATO	CARIPRATO
CARISBO	CARISBO
CARISPEZIA CRÉDIT AGRICOLE	CARISPEZIA_CRÉDIT_AGRICOLE
亲爱的	亲爱的
卡尔顿发型国际	carlton_Hair_Internati
CARLUNAS	CARLUNAS
CARNABY	CARNABY
CARNES DE EL BIERZO LEON	CARNES_DE_EL_BIERZO_LEON
CARNES FINAS SAN JUAN	CARNES_FINAS_SAN_JUAN
卡罗来纳州埃雷拉	CAROLINA_HERRERA
CAROLL	CAROLL
CARPATICA ASIG	CARPATICA_ASIG

名称	ID
CARPLU	CARPLU
卡奎斯特	卡奎斯特
携带	携带
金牌	CARTE_D'OR
CARs	CARs
家庭与视频	CASA_&_VIDEO
安迪娜之家	卡萨_安迪纳
CASA CHEIA	CASA_CHEIA
CASA ELIAS EL BUEN MERCADO	CASA_ELIAS_EL_BUEN_MERCADO
卡萨斯	卡萨斯
CASATUDO	CASATUDO
现金区	现金区
CASSA DEI RISP。DI FORLI E ROMAGNA	CASSA_DEI_RISP。 _DI_FORLI_E_ROMAGNA
CASSA DI RISP。DI VR	VI
CASSA DI RISP。PISTOIA E LUCCHESIA	CASSA_DI_RISP。 _PISTOIA_E_LUCCHESIA
CASSA DI RISPARMIO DEL FRIULI V. G.	CASSA_DI_RISPARMIO_DEL_FRIULI_V._G。
CASSA DI RISPARMIO DEL VENETO	CASSA_DI_RISPARMIO_DEL_VENETO
CASSA DI RISPARMIO DI ASCOLI PICENO	CASSA_DI_RISPARMIO_DI_ASCOLI_PICENO
博洛尼亚储物柜	CASSA_DI_RISPARMIO_DI_BOLOGNA
CASSA DI RISPARMIO DI CENTO	CASSA_DI_RISPARMIO_DI_CENTO

名称	ID
CASSA DI RISPARMIO DI CESENA	CASSA_DI_RISPARMIO_DI_CESENA
CASSA DI RISPARMIO DI FORLI	CASSA_DI_RISPARMIO_DI_FORLI
CASSA DI RISPARMIO DI ORVIETO	CASSA_DI_RISPARMIO_DI_ORVIETO
拉文纳储物柜	CASSA_DI_RISPARMIO_DI_RAVENNA
CASSA DI RISPARMIO DI SAN MINIATO	CASSA_DI_RISPARMIO_DI_SAN_MINIATO
CASSA DI RISPARMIO DI SAVONA	CASSA_DI_RISPARMIO_DI_SAVONA
CASSA DI RISPARMIO DI SPOLETO	CASSA_DI_RISPARMIO_DI_SPOLETO
CASSA DI RISPARMIO DI VENEZIA	CASSA_DI_RISPARMIO_DI_VENEZIA
CASSA DI RISPARMIO DI VOLTERRA	CASSA_DI_RISPARMIO_DI_VOLTERRA
CASSA DI RISPARMIO PADOVA E ROVIGO	CASSA_DI_RISPARMIO_PADOVA_E _ROVIGO
CASSA DI RISPARMIO PROV。DI VITERBO	CASSA_DI_RISPARMIO_PROV。 _DI_VITER BO
CASSA PADANA BCC	CASSA_PADANA_BCC
CASSA RISP。FABRIANO E CUPRAMONT ANA	CASSA_RISP。 _FABRIANO_E_CUPRAM ONTANA
CASSE DI RISPARMIO DELL'UMBRIA	CASSE_DI_RISPARMIO_DELL'UMBRIA
卡斯特罗	卡斯特罗
猫	猫
CAT 电信	CAT_Teleco
国泰航空	国泰航空
CATTAN	CATTAN

名称	ID
CAVA	CAVA
CAZADOR	CAZADOR
混蛋	混蛋
CB	CB
CB 银行	CB_Bank
CB 门店	CB_Stores
CBA	CBA
CBC	CBC
CBC FCU	CBC_FCU
CBEST	CBEST
世邦魏理仕	世邦魏理仕
CBZ	CBZ
CC	CC
CC 的咖啡屋	cc's 咖啡屋
CC-Bank	CC-Bank
cc-英语	cc-英语
CC-OO	CC-OO
CCAA	CCAA
建行人寿	CCB_Life
CCC	CCC

名称	ID
CCITS	CCITS
CCK 生鲜超市	cck_fresh_mart
CCM	CCM
CD 一价清洁剂	CD_one_price_Cleaners
光盘仓库	光盘仓库
CD-R King	cd-r_king
CEAT	CEAT
塞西尔	塞西尔
CECOMIL	CECOMIL
CEFARM	CEFARM
CEFCO	CEFCO
名人健身	名人_健身
CELEIRO	CELEIRO
CELP	CELP
CEM	CEM
CENTONE	CENTONE
中原证券	中原证券
中央商场	CENTRAL_EMPORIUM
CENTROXOGO	CENTROXOGO
CENTRUM MEDYCZNE LIM	CENTRUM_MEDYCZNE_LIM

名称	ID
ZDROWIA 中心	CENTRUM_ZDROWIA
世纪	世纪
CEPSA	CEPSA
CEPSA 精灵	CEPSA_ELF
CERRUTI 1881	CERRUTI_1881
CEZAR	CEZAR
CFC	CFC
CFExpress	CFExpress
CFG 银行	CFG_bank
CGI	CGI
CGV	CGV
CGV 电影院	CGV_Cinemas
CH Colleccion Hogar Home	ch_coleccion_Hogar_home
查比银行	CHAABI_BANK
CHABER	CHABER
CHABUTON	CHABUTON
连锁超市	连锁超市
椅子	椅子
CHAMATE	CHAMATE
长庚生物科技	CHANG_GUNG_生物科技

名称	ID
CHANGE	CHANGE
赵义兴	赵义兴
CHARME 餐厅	魅力餐厅
CHATA POLSKA	CHATA_POLSKA
星河城堡	星河城堡
CHATTAWAK	CHATTAWAK
巴黎尚美酒店	CHAUMET_PARI
CHEAPY LAGPRISBYGG	CHEAPY_LAGGPRISBYGG
干杯	干杯
干杯餐厅	啦啦队_餐厅
CHEF'STORE	CHEF'STORE
CHELINDBANK	CHELINDBANK
成都大运汽车	CHENGDU_DAYUNAUTO
奇瑞	奇瑞
CHEVIGNON	CHEVIGNON
雪佛兰	雪佛兰
切斯·杰拉德	CHEZ_GERARD
CHIC-BOY	CHIC-BOY
CHIFON	CHIFON
CHIH HANG 租车	CHIH_HANG_CAR_RENTAL

名称	ID
中国汽车租赁	中国_汽车租赁
中国渤海银行	中国_渤海银行
中信银行国际	中国_中信银行_国际
中国龙证券	中国_龙_证券
中国东方航空	中国_东方航空
中国富强证券	中国_财富_证券
中国民族证券	中国_MINZU_证券
中国证券	中国_证券
中国南方航空	中国_南方航空
中国太平	中国_太平
中国浙商银行	中国_浙商银行
唐人街	唐人街
CHIP 儿童身份证计划	chip_the_child_ID_Program
CHIQUITO	CHIQUITO
CHJ 珠宝	CHJ_珠宝
巧克力工厂	巧克力工厂
精选服装	精选服装
重庆小天鹅	CHONGQING_CYGNET
重庆百货大楼	重庆_百货大楼
CHOPPOP	CHOPPOP

名称	ID
CHOPPIES	CHOPPIES
周大福	CHOW_TAI_FOOK
CHRIST Monters & Bijoux	Christ_Montres_&_bijoux
CHRIST Hren & Schmuck	Christ_uhren_&_schmuck
克莱斯勒	克莱斯勒
川歌玉水娇	CHUAN_GE_YU_SHUIJIAO
杨春	春阳
中美药业	CHUNG_MEI_PHARMA
钟源	钟源
CHYI	CHYI
CI Banco	CI_banco
再见宝贝厨房	再见 BABY_CUCINA
CIAR	CIAR
CIB	CIB
CIB 银行	CIB_Bank
加拿大帝国商业银行	加拿大帝国商业银行
CIBO 意式浓	Cibo_espres
CIC	CIC
中国意大利商会	中国意大利商会
CIGKOFTEM	CIGKOFTEM

名称	ID
CHI 银行	CIH_BANK
CILEK	CILEK
联昌国际银行	CIMB_BANK
联昌国际尼亚加	cimb_Niaga
CIMB Niaga Syariah	cimb_Niaga_syariah
CIMB THAI	CIMB_THAI
信达证券	信达证券
哥伦比亚电影院	哥伦比亚电影院
CINE MULTIPLEX	CINE_MULTIPLEX
CINEPLEXX	CINEPLEXX
CINERAMA	CINERAMA
CINESA	CINESA
CINWA	CINWA
CIPO & BAXX	CIPO_&_BAXX
CIRCLE K	circle_k_express
CISSE	CISSE
引用咖啡	CITE_COFFEE
CITGO	CITGO
中信证券	中信证券
雪铁龙	雪铁龙

名称	ID
雪铁龙 EUROPASS	雪铁龙_EUROPASS
CITS	CITS
城市盒子	CITY_BOX
城市链	城市链
城市钻石	城市钻石
城市交易所	城市交易所
城市牛奶	CITY_MILK
城市鞋	城市鞋子
城市商店	城市_SHOP
CItylink 电子有限公司	CItylink_电子有限公司
CJ Banks	cj_Banks
CJ 超市	CJ_Sump
CK Firo-Tour	ck_firo-Tour
CK FISCHER	CK_FISCHER
CK Satur	ck_satur
CKAUTOM	CKAUTOM
快讯	快讯
CKS	CKS
CLB	CLB
干净整洁	CLEAN_&_CLEAN

名称	ID
CLEOR	CLEOR
聪明	聪明
CLICKS	CLICKS
巴伐利亚诊所	CLINICA_BAVIERA
圣马达莱娜牙科诊所	CLINICA_DENTÁRIA_SANTA_MADALENA
CLINIQUE	CLINIQUE
CLOUD 9	CLOUD_9
CM Hustalo	cm_Hustalo
CMC	CMC
CME FCU	CME_FCU
CMH	CMH
CMIT 解决方案	CMIT_解决方案
CMK	CMK
CMR Chigköfte	cmr_cigköfte
CNA	CNA
CNEP-Banque	CNEP-Banque
CNJ	CNJ
中海油	中海油
CNS	CNS
教练	教练

名称	ID
COCCINELLE	COCCINELLE
COCCODRILLO	COCCODRILLO
驾驶舱	驾驶舱
可可时间	COCO_TIME
COCO'S	COCO'S
COCOMORE	COCOMORE
CODE	CODE
蓝色代码	CODE_BLEU
咖啡湾	咖啡湾
咖啡男孩	咖啡男孩
咖啡俱乐部	咖啡俱乐部
咖啡角	咖啡角
咖啡岛	咖啡岛
咖啡不错的咖啡馆	咖啡_NIE_CAFE
乔治的 COFFEEHOUSE	George 的 coffehouse_
咖啡店公司	咖啡店_公司
COFFINE GURUNARU	COFFINE_GURUNARU
COGICO	COGICO
冷库	冷存储
COLIN'S	COLIN'S

名称	ID
COLLOSEUM	COLLOSEUM
COLOMBRARO	COLOMBRARO
颜色加号	颜色_PLUS
哥伦布	哥伦布
来买	COME_BUY
来买	来买
舒适	舒适
舒适的酒店	舒适的酒店
漫画连接	漫画_连接
COMME CAR ISM	COMME_CA_ISM
常见性别	共同性别
公司的	公司的
罗盘	罗盘
COMPTOIR DES COTONNIERS	COMPTOIR_DES_COTONNIERS
计算机浏览器	计算机资源管理器
CONAD	CONAD
概念时装店	概念时装店
协和酒店及度假村	康科德酒店及度假村
CONFORT auto	Confort_Auto
CONFORTauto	CONFORTauto

名称	ID
康拉德	康拉德
康索尔能源	控制台_能源
消耗	消耗
首席运营官和 RIKU	COO_&_RIKU
烹饪店	烹饪店
很酷的电台	酷站
很酷的猫	很酷的猫
金融合作社	COOP_COOPERATIVA_FINANCIERA
COOP OBS	COOP_OBS
COOPERATIVA OBRERA	COOPERATIVA_OBRERA
COPCU	COPCU
COPEG	COPEG
COPIDROGAS	COPIDROGAS
科拉	科拉
CORADIA	CORADIA
珊瑚	珊瑚
CORDIEZ	CORDIEZ
角落	角落
CORONEL TAPIOCCA	CORONEL_TAPIOCCA
CORREOS	CORREOS

名称	ID
CORT 家具奥特莱斯	Cort_Furniture_Out
CORT 家具出租	Cort_Furniture_出租
CORTEFIEL	CORTEFIEL
护卫舰	护卫舰
COS	COS
宇宙	宇宙
布拉瓦海岸豪华酒店	COSTA_BRAVA_HOTELS_DE_LUXE
太阳海岸	COSTA_DEL_SOL
COSTANTIN	COSTANTIN
COSTASUR	COSTASUR
COSY-WASCH	COSY-WASCH
COTA 商业银行	Cota_Commercial_Bank
国家金融	国家_财经
信使	信使
万豪万怡酒店	万豪万怡酒店
COVIRAN	COVIRAN
舒适的茶室	COZY_TEA_LOFT
CP FCU	CP_FCU
CP 生鲜超市	CP_Fresh_mart
CP Piliq	CP_pilic

名称	ID
CPAC	CPAC
中国共产党加油站	CPC_Gas Station
CPF	CPF
CPH	CPH
CPHM	CPHM
CPI 一小时照片完成	CPI_one_hour_hour_photo_Fr
CPM FCU	CPM_FCU
CPR-手机维修	CPR-cell_phone_Rephone
CPS 小伙子	CPS_CHAPS
华润先锋	cr_Vangu
CR die Mentz	cr_die_mentz
CRABTREE & EVELYN	CRABTREE_&_EVELYN
CRACE INN	CRACE_INN
CRAI	CRAI
起重机银行有限公司	起重机银行有限公司
克劳福德	克劳福德
CRCare	CRCare
CRE	CRE
创作食物	CREATION_FOOD
CREDEM	CREDEM

名称	ID
CREDI VENETO	CREDI_VENETO
信贷融资	信贷金融
黎巴嫩信贷	CREDIT_LIBANAIS
瑞士信贷	瑞士信贷
CREDITO COPERATIVO	CREDITO_COOPERATIVO
布雷西亚合作社信贷	CREDITO_COOPERATIVO_DI_BRESCIA
CREDITO TREVIGIANO	CREDITO_TREVIGIANO
可丽饼和华夫饼	CREPES_&_WAFFLES
CRISPINO	CRISPINO
CRISTALBOX	CRISTALBOX
CRNOGORSKA 商业银行	CRNOGORSKA_KOMERCIJALNA_BANKA
克罗地亚语	克罗地亚语
克罗地亚银行	克罗地亚银行
CROCODILINO	CROCODILINO
CRODUX	CRODUX
CROMA	CROMA
CROPP	CROPP
交叉牛仔裤	交叉牛仔裤
皇冠花哨	CROWN_FANCY
皇冠汽车有限公司	CROWN_MOTORS_LTD。

名称	ID
皇冠假日酒店	皇冠假日酒店
CRÈDIT	CRÈDIT
CSC	CSC
CSC 计算机教育	CSC_计算机_教育
CSE FCU	CSE_FCU
CSE 联邦信用合作社	CSE_Federal_Credit_Union
CSL	CSL
CSL Plasma	csl_Plasma
CSN	CSN
CSOB	CSOB
CT 光学	CT_OTICAL
CTAMA	CTAMA
CTBC	CTBC
CTM	CTM
CTown	CTown
CU	CU
CU 俄克拉荷马州一号	cu_one_of_Oklaoma
CU 服务中心	cu_服务中心
南加州大学	cu_of_southern_Californ
CUBESMART	CUBESMART

名称	ID
白色胸部	CUESTA_BLAN
翠微百货	CUIWEI_百货店
CUMAR	CUMAR
积云	积云
CUPLE	CUPLE
CUPPA	CUPPA
铜氨酸	铜氨酸
CVC Viagens	cvc_Viagens
简历/药房	简历/药房
CYFROWY POLSAT	CYFROWY_POLSAT
CYRILLUS	CYRILLUS
CYTS	CYTS
Ca RRo L BoYes	Ca RRo L_ BoYes
Cabalen	Cabalen
卡宾	卡宾
Cabela's	Cabela's
卡波烧烤	Cabo_Grill
可可	可可
Cacau Show	Cacau_Show
缓存	缓存

名称	ID
仙人掌	仙人掌
仙人掌俱乐部咖啡厅	仙人掌俱乐部咖啡厅
卡德中心	Cadd_Center
Cadena Clarín	Cadena_Clarín
凯迪拉医疗保健	凯迪拉_医疗保健
凯迪拉克	凯迪拉克
凯撒的	凯撒的
凯撒世界	《凯撒世界》
凯撒	凯撒
CELONA 咖啡厅和酒吧	Cafe_&_bar_celona
咖啡厅和餐饮无印良品	Cafe_&_meal_meal_muji
Amazon 咖啡厅	Cafe_亚马逊
Amul 咖啡厅	Cafe_Amul
Bazza 咖啡厅	咖啡馆_Bazza
咖啡馆好友	咖啡馆_Buddys
咖啡馆咖啡厅	咖啡馆_咖啡厅
咖啡俱乐部	咖啡俱乐部
咖啡馆咖啡日	Cafe_Coffe_
Cafe Comme C	Cafe_comme_Ca
DOITUNG 咖啡厅	cafe_doitung

名称	ID
咖啡馆的欲望	欲望咖啡厅
Extrablatt 咖啡	Cafe_extrabl
Grazie 咖啡厅	咖啡馆_Grazie
格雷格咖啡厅	咖啡馆_Greg
Opus 咖啡厅	Cafe_Opus
里约咖啡馆墨西哥烧烤	Cafe_Rio_墨西哥_Grill
维瓦尔第咖啡厅	Cafe_Vivaldi
咖啡店	咖啡店
自助餐厅 makro	cafeteria_makro
Caffe d'Or	咖啡_D'Oro
Caffe Nero	Caffe_Nero
Caffe Ritazza	咖啡_Ritazza
Caffe Rossini	Caffe_Rossini
Caffe Uno	Caffe_Uno
Caffe moya	咖啡_moya
咖啡因	咖啡因
咖啡厅和西班牙小吃	Café_&_Tapas
咖啡和茶	Café_&_Té
安德拉德咖啡厅	Café_Andrade
Colibri 咖啡厅	Café_Colibri

名称	ID
仓库咖啡厅	Café_Depo
甜甜圈咖啡厅	Café_Donuts
欧罗巴咖啡厅	欧洲咖啡厅
Café Hillel	Café_Hillel
咖啡馆 Joe	Café_Joe
马丁内斯咖啡厅	Café_Martínez
雀巢咖啡厅	Café_Nescafé
Quindío Express 咖啡厅	Café_Quindío_Express
珊瑚咖啡	Café_de_Coral
印度咖啡厅	Café_de_Indias
德里高地咖啡厅	Café_Delhi_Hights
Ponto Café	café_do_Ponto
Boucher 咖啡厅	Café_du_Boucher
Café&Te	Café&Te
加拉加斯咖啡厅	咖啡馆_加拉加斯
Cagdas	Cagdas
卡格里	卡格里
菜百珠宝	财百_珠宝
财达证券	财达证券
Caidie 面包店	Caidie_Baker

名称	ID
才蝶轩	才蝶轩
蔡根香	蔡根香
Cailinji	Cailinji
开罗安曼银行	开罗安曼银行
Caisse Populaire Desjard	Casse_Populaire_Desjardins
盒子	盒子
Caixa Adv	Caixa_Advocats
Caxa Albalat	Caixa_Albalat
Caixa Benicarlo	Caixa_Benicarlo
Caixa d'Engin	Caixa_d'Enginyers
Caxa Geral	Caixa_Geral
Geral de Depositos	caixa_geral_de_depositos
Caixa Man	Caixa_Manresa
阿尔科拉 Caixa Rural	Caixa_Rural_Alcora
Caixa Rular	Caixa_Rural_Balears
Caixa Rural D'Algem	Caixa_Rural_d'Algemesi
Caixa Rural Gale	Caixa_Rural_Galega
Caixa Rular	Caixa_Rural_Nules
Caixa Vila-Real	Caixa_Rural_Vila-Real
Caixa Vilavella	Caixa_Rural_Vilavella

名称	ID
Caixa Rular la	caixa_rural_la_vall
Caixa Rular les	caixa_rural_les_coves
Caixa 塔拉戈纳	Caixa_Tarragona
Caixa Vinaros	Caixa_Vinaros
Caixe de Credito Agrícola	caixa_de_crédito_Agrícola
CaixaBank	CaixaBank
CaixaLaVall	CaixaLaVall
CaixalQuerie	CaixalQuerie
Caixaterrasa	Caixaterrasa
菜药古史	Caiyao_Gushi
Caja Abogados	Caja_Abogados
Caja Arequipa	Caja_Arequipa
Caja Arequipa-Agente	Caja_Arequipa-Agente
卡哈巴达霍斯	Caja_Badaj
Caja Caminos	Caja_Caminos
卡哈坎波	Caja_Campo
Caja Cusco	Caja_Cusco
Caja Cusco-Agente	Caja_Cusco-Agente
Caja Huancayo	Caja_Huancayo
Caja Huancayo-Agente	Caja_Huancayo-Agente

名称	ID
Caja Immaculada	Caja_Immaculada
伊卡哈市政厅	Caja_Municipal_Ica
Caja Municipal Ica-A	Caja_Municipal_Ica-Agente
Caja Piura	Caja_Piura
Caja Piura-Agente	Caja_Piura-Agente
热门卡哈圣胡安博斯科	Caja_Popular _San_Juan_Bosco
Caja Real del Potosi	Caja_real_del_Potosí
Caja Rural Albal	Caja_Rural_Albal
Caja Rural Almenara	Caja_Rural_Almenara
Betxi Caja Rural	Caja_Rural_betxi
Caja Rural 赌场	Caja_Rural_Casinos
Caja Rural Castellon	Caja_Rural_Castellon
Caja Rural Chilches	Caja_Rural_Chilches
科尔多瓦乡村民宿	Caja_Rural_Cordoba
Caja Rural Jaén	Caja_Rural_Jaén
新卡尔特亚乡村酒店	Caja_Rural_Nueva_Carteya
Caja 农村地区	Caja_Rular_Regional
Caja Rural VilaFames	Caja_Rural_VilaFames
Caja Rural de Albacete	caja_rural_de_Albacete
Caja Rural de Alginet	caja_rural_de_alginet

名称	ID
阿斯图里亚斯乡村民宿	Caja_rural_de_Asturias
Caja Rural de Baena	caja_rural_de_baena
Caja Rural de Canarias	Caja_rural_de_canarias
Caja Rural de Casas Ibáñez	caja_rural_de_casas_ibáñez
Caja Rural de Cañete	caja_rural_de_cañete
Caja Rural de Cheste	caja_rural_de_Cheste
Caja Rural de Real	caja_rural_de_ciudad_real
Caja Rural de Cuenca	caja_rural_de_cuenca
丰特佩拉约乡村民宿	caja_rural_de_fuentepelayo
Caja Rural de Mota del Cuervo	caja_rural_de_mota_del_cuervo
Caja Rural de Onda	caja_rural_de_onda
萨拉曼卡乡村民宿	Caja_rural_de_Salamanca
Caja Rural de Soria	caja_rural_de_soria
Caja Rural de Teruel	caja_rural_de_Teruel
托莱多乡村民宿	caja_rural_de_Toledo
乌特雷拉乡村民宿	caja_rural_de_Utrera
Caja Rural de Villamalea	caja_rural_de_villaMalea
Caja Rural de la Roda	caja_rural_de_la_roda
Caja Rural del Duero	caja_rural_del_del_Duero
Caja Sullana	Caja_Sullana

名称	ID
Caja Sullana-Agente	Caja_Sullana-Agente
Caja Tacna	Caja_Tacna
Caja Tacna-Agente	Caja_Tacna-Agente
Caja Trujillo	Caja_Trujillo
Caja Trujillo-Agente	Caja_Trujillo-Agente
Caja de Arquitectos	Caja_de_Arquitectos
巴达霍斯之屋	Caja_de_Badajo
Caja de Estremadura	caja_de_extremadura
Caja de Ingenieros	caja_de_ingenieros
Caja de Jaen	caja_de_Jaen
CajaSur	CajaSur
Cajalmendraleo	Cajalmendraleo
Cajalon	Cajalon
Cajasiete	Cajasiete
Cajasol	Cajasol
蛋糕和烘烤	Cake_&_Bake
蛋糕和烘焙食品	Cakes_and_Bakes
卡尔达披萨	卡尔达_披萨
Caliber 碰撞	口径碰撞
Caliber 住房贷款	Caliber_Home_Loon

名称	ID
机芯	机芯
Calicéo	Calicéo
加利福尼亚	加利福尼亚
加州银行与信托基金	加州_银行_&_信托
加州熊 CU	加利福尼亚熊_cu
加州壁橱	加州_壁橱
加州健身	加州健身
加州鱼烧烤	加州_Fish_Grill
加州新鲜	加州_Fresh
加州国家银行	加州国民银行
加州披萨厨房	加州_披萨_厨房
加利福尼亚共和国	加利福尼亚共和国
加州玉米饼	加州_Tortilla
Calimax	Calimax
卡里普索	卡里普索
Caliser	Caliser
致电 FCU	call_fcu
叫一辆自行车	叫_a_bike
称之为春天	Call_it_Spring
Callaghan Gorila	Callaghan_Gorila

名称	ID
Calpam	Calpam
加德士	加德士
加德士 Woolworths	加德士_Woolworths
Calumet 摄影	Calumet_Photive
加尔瓦里教堂	Calvary_Chapel
Calvin Klein	Calvin_Klein
Calypso Fitness	Calypso_Fitness
卡尔扎多出租车	Calzado_taxi
卡尔扎多斯纳瓦罗	Calzados_Navarro
Calzados Segundo Par 1 欧元	Calzados_segundo_par_1_euro
卡尔扎多斯 WhyNot	Calzados_ WhyNot
Calçados Bibi	Calçados_Bibi
Cama Cafe	CamaCafe
Camarão & Cia	Camarão_&_Cia
Cambalache	Cambalache
坎布里亚套房	坎布里亚套房酒店
剑桥	剑桥
剑桥建筑协会	剑桥建筑协会
剑桥食品	剑桥_食品
卡姆登	卡姆登

名称	ID
卡姆登国家银行	卡姆登国民银行
骆驼	骆驼
卡米洛特	卡米洛特
相机屋	相机屋
Camicado	Camicado
卡米尔·阿尔巴内	Camille_Albane
Camille La Vie	Camille_La_Vie
卡米尔的 Sidewalk Cafe	Camille's Sidewalk_Cafe
Camino FCU	camino_fcu
Camino Real	Camino_Real
Camisaria 科伦坡	卡米萨里亚_科伦坡
Camiseria Inglesa	Camiseria_Inglesa
Camp Bow 哇	Camp_Bow_Wow
大西洋营地	大西洋营地
钟楼	钟楼
露营者	露营者
露营世界	露营_世界
Indigo 露营地	Campings_Indigo
Campneus	Campneus
坎波特	坎波特

名称	ID
坎普萨	坎普萨
校园	校园
校园鞋	校园_鞋子
校园商店	校园商店
校园套房	校园套房
美国加州大学校区	Campus_usa_cu
Campeole	Campeole
Can Kardesler Kuruyemis	Can_Kardesler_Kuruyemis
加拿大面包	加拿大_面包
加拿大鹅	加拿大_鹅
加拿大之家	加拿大众议院
加拿大邮政	加拿大邮报
加拿大最具价值旅馆	加拿大_最物超所值旅馆
加拿大太平洋地区	加拿大_太平洋
加拿大轮胎	加拿大_轮胎
加拿大西部银行	加拿大西部银行
运河 M	Canal_M
运河+	Canal_Plus
Canali	Canali
卡南代瓜国家银行	加拿大国民银行

名称	ID
加拿大银行	加拿大银行
加那利群岛	加那利群岛
坎德尔伍德套房酒店	烛木套房
糖果花束	糖果_花束
苍浪亭	苍浪亭
Cango	Cango
Canifa	Canifa
坎南幼儿园和托儿所	Cannan_幼儿园_&_托儿所
Cannelle	Cannelle
罐头厂街	Cannery_Row
大炮	大炮
Cannon FCU	cannon_fcu
Cano Cornejo	Cano_Cornejo
佳能	佳能
希尔顿嘉悦酒店	Canopy_by_Hilton
Cantabil	Cantabil
康塔塔	康塔塔
食堂	食堂
坎特伯雷	坎特伯雷
Cantina Laredo	Cantina_Laredo

名称	ID
墨西哥流浪乐小酒馆	Cantina_Mariach
Cantão	Cantão
Canvas	canvas_cu
帆布咖啡厅	Canvas_Cafe
Cap N'Cork	Cap_n_Cork
CapEd CU	CapEd_CU
Capa de Ozono	Capa_de_ozono
开普航空	Cape_Air
CapeNature	CapeNature
卡皮奥	卡皮奥
CapitaMalls	CapitaMalls
CapitaMalls 亚洲	CapitaMalls_亚洲
资本银行	资本银行
首都 CU	Capital_cu
首都车	资本_汽车
资本化学家	Capital_Chemist
首都银行	首都城市银行
第一资本	Capital_one
资本证券	资本_证券
首都电影院	首都影院

名称	ID
Capitec Bank	Capitec_Bank
国会大厦	国会大厦
国会大厦	Capitol_cu
国会联邦储蓄银行	Capitol_Federal_Savings_Bank
Capodarte	Capodarte
卡布奇诺的	卡布奇诺的
卡普雷塞	卡普雷塞
Caprichos 时装和配饰	Caprichos_moda_y_complemento
Caprichosas	Caprichosas
卡普里诺	卡普里诺
卡普西斯酒店	卡普西斯_酒店
D 队长	Captain_D
Capucine	Capucine
Caputos 新鲜市场	Caputos_Fresh_Markets
汽车美容	汽车_美容
汽车便利性	汽车_便利
汽车汽油	汽车_汽油
汽车医生	Car_Medic
汽车租赁	汽车出租
保加利亚租车	汽车租赁_保加利亚

名称	ID
汽车玩具	汽车_玩具
汽车蜡	Car_Wax
Car X 汽车服务	Car_X_Auto_Service
Car'Go	Car'Go
Car-Condal	Car-Condal
CarMax	CarMax
卡拉药房	卡拉_药房
杨桃王子	Carambola Prince
卡拉特兰	卡拉特兰
碳健康	碳_健康
Carbone 的 Pizzeria	Carbone's_Pizzeria
卡片工厂	卡片工厂
卡德纳斯市场	Cardenas_Market
Cardies	Cardies
红衣主教	红衣主教
Cardo	Cardo
Cardon	Cardon
智能卡	智能卡
Cardronics	Cardronics
护理化学补习中心	Care_Chemistry_学费_中心

名称	ID
护理公园	Care_Park
护理药房	护理_药房
聪明地护理	护理_Smart
CareDent	CareDent
CareOne	CareOne
Carecross	Carecross
职业启动器教育	职业生涯_启动器_教育
卡雷斯特尔	卡雷斯特尔
汽车玻璃	汽车玻璃
CariChieti	CariChieti
驯鹿咖啡	驯鹿_咖啡
爱心药房	爱心药房
Carino 的意大利烧烤	Carino's _Italian_Grill
Carl's Jr.	carls_jr。
卡林	卡林
Carline 汽车和排气系统	Carline_Automotive_&_排气
Carlo Pazolini	Carlo_Pazolini
Carlo Rino	Carlo_Rino
CarloSantos 美发店	CarloSantos_HAIR_SHOP
卡洛斯·奥凯利的墨西哥咖啡馆	Carlos_O'Kelly's 墨西哥咖啡馆

名称	ID
Carls 家具与设计	Carls_Furniture_&_Design
Carlson Wagonlit Travel	Carlson_Wagonlit_Travel
卡尔顿	卡尔顿
卡尔顿卡	Carlton_Cards
卡尔顿燃料	Carlton_Fuels
伦敦卡尔顿酒店	卡尔顿_伦敦
卡鲁西奥的	卡鲁西奥的
Carly	Carly
卡梅尔修道院学校	卡梅尔修道院学校
卡门·斯蒂芬斯	Carmen_Steffens
卡米克电影院	卡迈克_电影院
卡恩站	Carne_Station
卡恩斯·拉莫斯	Carnes_Ramos
Vol 名单	Carnet_de_Vol
嘉年华	嘉年华
嘉年华电影院	嘉年华_电影院
嘉年华家具商	嘉年华_家具
Caro Cuore	Caro_Cuore
Caroil	Caroil
卡罗来纳博伊克斯	Carolina_Boix

名称	ID
卡罗来纳第一	Carolina First
卡罗来纳州处理	卡罗来纳州_搬运
Carolina Lemke	Carolina_Lemke
卡罗来纳州石油公司	Carolina_Petro
卡罗来纳信托基金 FCU	Carolina_trust_FCU
卡罗来纳州的Health Care系统	卡罗来纳州的健康护理系统
卡罗琳·夏娃	Caroline_Eve
停车场	停车场
地毯球场	地毯球场
一号地毯	地毯_一
地毯右	地毯_右
CarpetsPlus 彩色瓷砖	CarpetsPlus_Color_Tile
车载电话仓库	车载电话_仓库
Carpisa	Carpisa
Carrabba 的意大利烧烤	Carrabba's_Italian_Grill
家乐福	家乐福
家乐福 5 分钟	家乐福_5_minut
家乐福拜罗	家乐福_拜罗
家乐福银行	家乐福银行
家乐福联系方式	家乐福_联系我们

名称	ID
家乐福 Drogaria	家乐福_Drogaria
家乐福 GB	家乐福_GB
家乐福 Hyper	家乐福_Hyper
家乐福地点	家乐福_地点
家乐福 MAXI	家乐福_maxi
家乐福 Marinopoulos	家乐福_Marinopoulos
家乐福通行证	家乐福通行证
家乐福维亚耶斯	家乐福_Viajes
家乐福之旅	家乐福_Voyages
家乐福简介	家乐福_生物
家乐福城	家乐福城
家乐福大道	家乐福_Drive
家乐福快递	家乐福快递
家乐福市场	家乐福市场
家乐福蒙塔涅	家乐福_蒙塔涅
家乐福星球	家乐福星球
Carrera y Carrera	carrera_y_carrera
卡里安娜	卡里安娜
卡罗尔汽车燃料	卡罗尔_汽车_燃料
Carrows	Carrows

名称	ID
Carrs	Carrs
Carré Blanc	Carré_Blanc
Carrément Fleurs	Carrément_Fleurs
Carsome	Carsome
Carsone	Carsone
卡特银行和信托基金	Carter_Bank_&_Trust
Carter FCU	Carter_fcu
卡特·伦伯	Carter_Lumber
Carter's	Carter's
Carter-Cash	Carter-Cash
Carters	Carters
卡地亚	卡地亚
墨盒世界	墨盒_世界
Carulla	Carulla
卡鲁拉快车	Carulla Express
Carvalho Supermercado	Carvalho_Supermercado
Carvel	Carvel
Carvil	Carvil
Carx 轮胎和汽车	carx_tire_and_Auto
Caréco	Caréco

名称	ID
房子	房子
Casa Alvarinho	Casa_Alvarinho
Casa Batalha	Casa_Batalha
Casa Bayon	Casa_Bayon
贝拉之家	Cas_Bella
夏娃之家	Casa_Eve
卡萨格兰德	卡萨_格兰德
卡萨·乔文	Casa_Joven
Casa Pariurilor	Casa_Pariurilor
卡萨皮奥	Casa_Pio
Casa de Bolos	Casa_de_bolos
书屋	Casa_del_libro
建筑师之家	Casa_do_construc
Casa do Papel	Casa_do_papel
Casa do Pao de Queijo	casa_do_pao_de_queijo
Casa dos Relojoeiros	casa_dos_relojoeiros
Casadei	Casadei
Casas Bahia	Casas_Bahia
Casco FCU	casco_fcu
卡斯登	卡斯登

名称	ID
Casey's 杂货店	Casey's General_Store
美国现金	美国现金
现金转换器	现金转换器
现金快递	现金快递
现金组	现金集团
象牙现金	现金_象牙色
Cash Piscines	Cash_Piscines
现金+	Cash_Plus
现金记录	现金记录
现金商店	Cash_Store
Cash city Shabu-Shabu	cash_city_shabu-shabu
Cash-N-Dash	Cash-N-Dash
现金积累	现金积累
卡什兰	卡什兰
开司米	开司米
赌场	赌场
赌场咖啡厅	Casino_Cafétéria
赌场公园	Casino_Park
赌场大道	Casino_Drive
赌场商店	Casino_shop

名称	ID
卡西欧	卡西欧
卡西欧 G 工厂	casio_g-factory
Casper	Casper
Cassa Risp. della Prov. Dell'Aquila	Cassa_Risp。 _della_prov。 _Dell'Aquila
Cassa di Risparmio 帕尔马和皮亚琴察	cassa_di_risparmio_parma_e_Piacenza
Cassa di Risparmio di Carrara	cassa_di_risparmio_di_Carrara
Cassa di Risparmio di Loreto	cassa_di_risparmio_di_loreto
Cassa di Risparmio di Rieti	cassa_di_risparmio_di_Rieti
Cassa di Risparmio di Rimini	cassa_di_risparmio_di_Rimini
木薯根	木薯根
卡西迪精品店	卡西迪精品店
卡西斯	卡西斯
Cassol Centerl	Cassol_Centerlar
Castañer	Castañer
Castaño	Castaño
洛佩斯城堡	Castello_Lopes
城堡牙科	城堡牙科
卡斯特罗汽车服务	卡斯特罗汽车服务
嘉实多	嘉实多
Casu	Casu

名称	ID
休闲男士	休闲_男款
猫鞋	猫鞋
Catalo	Catalo
加泰罗尼亚	加泰罗尼亚
Catena	Catena
卡特琳娜	卡特琳娜
Caterina Leman	Caterina_Leman
国泰银行	国泰银行
国泰未来	国泰未来
国泰世华银行	国泰世华银行
凯瑟琳	凯瑟琳
天主教和社区 CU	Catholic__and__community_cu
天主教慈善	天主教慈善机构
叙利亚天主教银行	天主教叙利亚银行
Catic 健康	Catic_健康
Catmoss	Catmoss
卡托	卡托
养牛大亨	Cattle_Baron
Cau	Cau
卡瓦蒂纳音乐工作室	Cavatina_Music_Studio

名称	ID
Cavavin	Cavavin
Cavender's Boot City	Cavender's boot_City
Cavenzi	Cavenzi
恐龙洞穴	Cavern_do_dino
洞穴图书	洞穴书籍
卡夫蒙特银行	Cavmont_Bank
天哪	Cay_Thi
凯维尔	Cay_Ville
Caçula de Penus	caçula_de_pneus
Cañas & Tapas	cañas_y_tapas
Ccf	Ccf
ccmonagas	ccmonagas
Ccso	Ccso
Cdek	Cdek
CeX	CeX
Cec 银行	Cec_Bank
Ceca	Ceca
Cecauto	Cecauto
雪松瀑布社区 CU	Cedar_falls_community_cu
Cedeo	Cedeo

名称	ID
Cedico	Cedico
Cedok cestovni kancelar	cedok_cestovni_Kancelar
Celcom	Celcom
Celcom .Cube	Celcom_.Cube
Celcom Exclusive	Celcom_Exclusive
Celcom bluecube	celcom_blueCube
名人之城	名人_城市
名人发型设计讲座	Celebrity_Hair_Design_Cerener
西莉亚	西莉亚
Celikler	Celikler
席琳	席琳
Celio	Celio
CellOnly	CellOnly
地下室	地下室
Cellcom	Cellcom
手机	手机
蜂窝一号	Cellular_One
Celop	Celop
Cemex	Cemex
Cenbest	Cenbest

名称	ID
胜腾公司	Cendant_公司
Cenex	Cenex
圣塔拉	圣塔拉
半人马座	半人马座
百年纪念银行	Centenary_Bank
百年纪念银行	Centennial_Bank
卡斯蒂略中心	中心_卡斯蒂略
中央公园 CU a APCU Div	center_parc_cu__a_div_of_apcu
CenterCredit	CenterCredit
中心大楼	中心大楼
Centex 家园	Centex_Homes
厘米	厘米
Centra	Centra
Centra CU	centra_cu
中央	中央
环球中央银行	中央银行_环球银行
中央银行公司	中央银行公司
中央银行与信托	中央银行与信托
埃及中央银行	埃及中央银行
印度中央银行	印度中央银行

名称	ID
尼日利亚中央银行	尼日利亚中央银行
佛罗里达州中部 CU	佛罗里达州中部 cu_cu_of
钦奈中央合作银行	Central_Chennai_Co-op_Bank
中央城市酒店	中央城市酒店
中央合作银行	Central_Co-op_Bank
中央通信 CU	中央通信_cu
中央便利店	中央便利店
中央合作银行	中央合作银行
马德伦斯中部	Central_Madeirense
缅因州中部 FCU	Central_maine_FCU
中央购物中心	中央购物中心
中央市场	中央市场
密苏里州中部社区 CU	Central_Missouri_community_cu
中央太平洋银行	中央太平洋银行
中央停车系统	中央停车系统
中央广场	中央广场
中央租车	中央租车
中央商店	Central_Store
中央轮胎	中央轮胎
Central UP. Gas	Central_U.P._Gas

名称	ID
中央制服	中央_制服
中央高保真	中央高保真音响
中心的 FCU	centric_FCU
中心	中心
英国中部	Centro_Británico
欧洲中心口腔种植学	Centro_Europeo_Implantologia_Oral
森特罗酒店	Centro_Hotel
中心购物中心	Centro_购物中心
Centronet	Centronet
Centros Ideal	中心_理想的
Centrum Dekoracyne	Centrum_Dekoracyjne
百夫长银行	百夫长银行
Centurion 是 Crane CU 分支	Centurion_a_branch_of_cranch_cu
21 世纪	Century_21
世纪银行	世纪银行
世纪 CU	世纪_cu
世纪遗产 FCU	Century_Heritage_FCU
世纪证券	世纪证券
世纪环球影业	世纪_环球影业
CenturyFirst FCU	CenturyFirst_FCU

名称	ID
CenturyLink	CenturyLink
仪式	仪式
Ceria 99	Ceria_99
Ceriamart	Ceriamart
Cerpe	Cerpe
Cerruti	Cerruti
CertaPro 画家	CertaPro_Painters
已认证	已认证
经认证的联邦 CU	Certified_Federal_CU
Certigard	Certigard
Cervecería Cruz Bl	Cerveceria_Cruz_Blanca
Cervefrío	Cervefrío
塞尔韦拉	塞尔韦拉
Cesa Cigkofte	Cesa_Cigkofte
Ceska narodni banka	ceska_narodni_Banka
Ceska sporitelna	ceska_sporitelna
Ceskomoravska stavebni sporitelna	ceskomoravska_stavebni_sporitelna
Cesta do Povo	cesta_do_povo
Cestlavie	Cestlavie
Cetelem	Cetelem

名称	ID
Cetinkaya	Cetinkaya
Ceva	Ceva
Ceyo	Ceyo
CfC Stanbic Bank	CFC_Stanbic_Bank
中国汽车科技	Ch-Auto_Technoloy
茶白道	Cha_Bai_Dao
Cha Hua Nong	Cha_Hua_Nong
Cha Tanyong	Cha_Tanyong
茶换茶	Cha_for_tea
ChaTraMue	ChaTraMue
Chaayos	Chaayos
Chabara 555	Chabara_555
Chaffey FCU	Chaffey_fcu
Chagee	Chagee
Chago Tea & Cafe	Chago_Tea_Cafe
查希亚	查希亚
Chaihona 1	Chaihona_1
Chailease	Chailease
Chaimiyouyan	Chaimiyouyan
柴里特租赁	Chairit_租赁

名称	ID
查克拉	查克拉
苏塞国际木屋酒店	Chalet_Susse_国际
Chalkiadakis	Chalkiadakis
质询	质询
挑战者	挑战者
挑战者摩托车	挑战者_摩托车
Chaman 冰淇淋	Chaman_Ice_Cream
Champ Sports	Champ_Sports
冠军	冠军
冠军 CU	冠军_cu
冠军马里诺普洛斯	冠军_Marinopoulos
冠军抵押	冠军抵押贷款
冠军披萨	冠军_披萨
冠军溜冰场	Champion_Rink
尚普兰农场	尚普兰农场
Champps	Champps
冠军体育	Champs_Sports
香榭丽舍大街独家	香榭丽舍大街_独家
陈氏家具	Chan_家具
Chana-马自达	Chana-马自达

名称	ID
尚德里斯酒店及度假村	钱德里斯酒店及度假村
Chanello 的披萨	Chanello's_Pizza
长安福特	长安福特
长安酒店	长安酒店
彰化银行	长华银行
长顺茶	长顺茶
长安	长安
长安银行	长安银行
长安保险	长安保险
长春发展农村商业银行	长春_开发_农村通信_银行
长春农村商业银行	长春农村商业银行
长峰	长峰
长丰农村商业银行	长丰_农村_商业银行
长和	长和
长江证券	长江证券
长宁农村商业银行	长宁农村商业银行
Chango Mals	Chango_Más
长青口腔	长青口腔医疗
长生人寿	长生_生活
常熟农村商业银行	常熟农村商业银行

名称	ID
长兴农村联合银行	长兴联合农村银行
长治银行	长治银行
长治农村商业银行	长治_农村_商业银行
长治漳泽农村商业银行	长治_张泽_农村通信_银行
哈尼亚银行	Chania_Bank
禅境瑜伽	禅经_瑜伽
海峡群岛燃料	海峡群岛_燃料
Chanson	Chanson
Chantefrais	Chantefrais
Chantelle	Chantelle
Chao Doi 咖啡	Chao_doi_Coffe
Chao Inn	Chao_Inn
巢湖农村商业银行	巢湖_农村_商业银行
Chapayom	Chapayom
卓别林	卓别林
章节	章节
Char-Grill Yakitori 一番	Char-grill_Yakitori_ichiban
字符	字符
查兰加	查兰加
木炭吃	Charcoal_Eats

名称	ID
给你的汽车充电	为你的汽车充电
ChargePoint	ChargePoint
Chargefox	Chargefox
Chargemaster	Chargemaster
Chargy	Chargy
Charinyeh	Charinyeh
查尔斯和基思	Charles_&_Keith
查尔斯·施瓦布公司	Charles_Schwab_Corp
查尔斯·泰尔惠特	Charles_Tyrwhitt
Charles Vögele Mode	Charles_Vögele_Mode
Charley's	Charley's
Charleys 费城牛排	Charleys_Philly_Steaks
查理·布朗的牛排馆	Charlie_Brown's SteakHouse
查理·奥特洛	Charlie_Outlaw
查理的鸡	Charlie's Chicken
夏洛特	夏洛特 [鞋子-鞋子]
夏洛特	夏洛特 [休闲餐饮]
夏洛特·鲁斯	Charlotte_Russe
查理披萨	查理_披萨
Charmant Friseur & Cosmetik	Charmant_Friseur_&_Kosmetik

名称	ID
Charmettes	Charmettes
Charmhotel	Charmhotel
迷人的查理	迷人_查理
迷人的购物中心	Inc
迷人的雪冰	Charmy_Snow_Ice
Charrua	Charrua
图表之家	Chart_Hous
宪章	宪章
宪章通讯	宪章通讯
CharterOak FCU	Charter_oak_FCU
第一章银行	Charter_One_Bank
特许的	特许的
查鲁埃尔	查鲁埃尔
Charvet	Charvet
Chase	Chase
Chashengju	Chashengju
黑貂城堡	Chateau_de_Sable
Chatime	Chatime
周记	Chau_Kee
周伟洁	Chau_Wei_Jue

名称	ID
尚美	尚美
Chaussland	Chaussland
Chaussea	Chaussea
Chawla Jewelers	Chawla_Jewelers
Chaynaya Lozhka	Chaynaya_Lozhka
车田停车场	Che_Tin_Parking
CheBanca!	CheBanca!
Chebey 牙科连锁店	Chebey_Dental_Chain
Check 'n Go	check_'n_go
查看乐趣	check_fun
使用现金支付	Check_into_Cash
办理登机手续旅行	办理登机手续_旅行
CheckSmart	CheckSmart
检查员	检查员
Checker 汽车零部件	Checker_Auto_Parts
跳棋	跳棋
Checkers 杂货店	Checkers_杂货店
核对清单	核对清单
Checkmate 服务	Checkmate_服务
结账	结账

名称	ID
切达的刮刮厨房	Cheddar's_Scratch_Kitch_
Chedraui	Chedraui
Chedraui selecto	chedraui_Selecto
Cheeburger Cheeburger	Cheeburger_Cheeburger
厚脸皮	厚脸皮
Cheer Young	Cheer_Young_Ed
干杯宫	欢呼宫
奶酪	奶酪
天堂里的芝士汉堡	天堂里的芝士汉堡
厨师兼酿酒师	Chef_and_Brewer
Chefenxiang	Chefenxiang
Chelneft'	Chelneft'
切尔西建筑协会	切尔西建筑协会
切尔滕纳姆和格洛斯特	Cheltenham_&_Gloucester
车里雅宾投资银行	车里雅宾投资银行
Chem-Dry	Chem-Dry
ChemStation	ChemStation
Chemcel FCU	chemcel_fcu
化学银行	化工银行
化学家仓库	化学家_仓库

名称	ID
Chemmanur 信贷与投资	Chemmanur_信用_&_投资
Chemmanur 珠宝商	Chemmanur_Jewelers
Chemmart	Chemmart
Chemnitzer 花环	chemnitzer_BlumenRing
Chemolak	Chemolak
Chempion	Chempion
化学+	化学+
晨宝教育	晨宝_教育
成都农村商业银行	成都_农村_商业银行
诚豪证券	成豪证券_证券
成客大酒店	成科酒店
诚信物流	诚信物流_物流
成悦	成悦
陈记顺和牛肉	陈记_顺和_牛肉
Chenjisanjidi	Chenjisanjidi
陈丽吉	陈丽吉
钦奈中央合作银行	Chennai_Central_Co-op_Bank
钦奈公司合作银行	Chennai_Corporation_Co-op_B
晨钟跆拳道	陈忠_跆拳道
Cheogajip	Cheogajip

名称	ID
CherryBerry 酸奶吧	CherryBerry_Yougurt_Bar
Chesanyama	Chesanyama
切萨皮克百吉饼面包店	Chesapeake_Bagel_Bakery
Chesco	Chesco
柴郡建筑协会	柴郡建筑协会
切斯特	切斯特
切斯特的	切斯特的
切斯特斯	切斯特斯
Chetyre lapy	chetyre_Lapy
雪佛龙	雪佛龙
雪佛龙 FCU	雪佛龙_FCU
雪佛龙食品市场	雪佛龙_Food_Mart
Chevys	Chevys
Chez Choux	Chez_Choux
Chez Cora	Chez_Cora
Chi Chi's	Chi_Chi's
Chi Shang Bian Dang	Chi_Shang_Bian_Dang
Chi Shang 粉丝包	Chi_Shang_fan_Bao
迟香何拉	Chi_Xiang_He_La
千源加油站	Chian_uen_加油站

名称	ID
赵灵龙	赵灵龙
千叶信用金库	千叶信用金库
千叶金证券	Chibagin_证券
Chibis	Chibis
Chibo	Chibo
芝加哥披萨	芝加哥_披萨
芝加哥产权公司	芝加哥_标题_公司
芝加哥的	芝加哥的
chick-fil-A	chick-fil-A
chickabiddy	chickabiddy
鸡肉厨师	Chicken_Chef
鸡肉小屋	鸡舍小屋
鸡肉快乐	Chicken_Deligh
鸡肉快车	鸡肉快递
鸡肉旅馆	Chicken_Inn
鸡王	Chicken_King
鸡肉厨房	鸡肉_厨房
舔鸡	Chicken_Licken
Chicken Out 烤肉店	Chicken_Out_Rotisserie
鸡肉沙拉小鸡	Chicken_Salad_Chick

名称	ID
鸡肉片	Chicken_Slice
鸡肉 Tikka	Chicken_Tikka
鸡肉零食	Chicken_Treat
Chicken-Mast	Chicken-Mast
Chickin Tyme	Chickin_Tyme
Chicking	Chicking
Chico 的	Chico 的
Chico 的全美小酒馆	Chico's_all_American_Bistro
Chico 的 Off The Rack	Chico's_off_the_rack
Chico 的餐厅	Chico's Restaurant
Chicony	Chicony
首席	首席
建延	建延
ChienShiang 早餐和早午餐吧	ChienShiang_早餐&早午餐_酒吧
儿童保育网络	儿童保育网络
儿童果园	儿童果园
儿童之地	儿童之地
儿童时光日托中心	儿童时光_日托_中心
智利快车	智利快车
辣椒屋	Chili_House

名称	ID
Chili's Grill 和酒吧	Chili's_Grill_&_Bar
Chili's Too 餐厅	Chili's too_restaurant
Chilim Balam	Chilim_Balam
辣椒豆	Chilli_Beans
Chilo	Chilo
Chima	Chima
Chimarrao	Chimarrao
Chimarrano	Chimarrano
Chin's	Chin's
中国蚂蚁	China_Ant
中国银行	中国_银行
中国书店	中国_书店
中国自助餐	中国_自助餐
中国中信银行	中国_中信银行
中国汽车专家	中国_汽车_专家
中国建设银行	中国_建设_银行
中国大陆保险	中国_大陆_保险
中国国家开发银行	中国发展银行
国家开发银行证券	中国_发展_银行_证券
中国远程教育控股有限公司	China_Distance_Edu_Holdings_Ltd

名称	ID
中国教育	China_Edu
中国光大银行	中国_光大银行
中国出口和信用保险	中国_出口与信贷_保险
中国快递	中国快递
中国电影都市生活影院	中国_电影_都市_生活_影院
中国民俗餐厅	China_folk's Restaurant
中国银河证券	中国_银河_证券
中国花园	中国_花园
中国之家	中国之家
中国旅馆	China_Inn
中国保险	中国_保险
中国投资证券	中国_投资_证券
中国人寿保险公司	中国_人寿_保险_公司
中国招商银行	中国_招商_银行
招商证券	中国_招商_证券
中国明生银行股份有限公司	中国_Mingsheng_Banking_Corp
中国移动	中国_移动
中国太平洋保险股份有限公司 有限公司	中国_太平洋_保险_公司 _Ltd
中国邮政保险	中国邮政保险
中国邮政证券	中国邮政证券

名称	ID
中国推广	中国_推广
中国可再生能源	China_re
中国资源	中国_资源
中国资源先锋	中国_资源_Vanguard
中国南方城市	中国_南方城市
中国茶	中国_茶
中国电信	中国_电信
中国旅行社（香港）有限公司	中国_旅行_服务_ (香港) _Ltd。
中国联通	中国_联通
盒子里的中国	中国_in_box
Chinai 的咖喱	Chinai's Curry
华林证券	华林证券
中信地产	中国信托_房地产
中国美食快车	中国_美食快递
中国酸菜鱼	中国_酸菜_鱼
钦丰银行	Chinfon_Bank
Chinfond 自助餐和娱乐	Chinfond_自助餐与娱乐
正健健身	Ching_Fitness
Ching Shin	Ching_Shin
Chinmaya Vidyalaya	Chinmaya_Vidyalaya

名称	ID
钦莱	钦莱
Chip 7	Chip_7
Chiphone FCU	chiphone_FCU
Chipotle	Chipotle
薯条和零食	薯条_&_零食
ChipsAway	ChipsAway
Chiquinho Sorvetes	Chiquinho_Sorvetes
知多信用金库	Chita_Shinkin_Bank
Chitay-gorod	Chitay-gorod
Chitayna	Chitayna
潮州花园	Chiuchow_Garden
潮州花园休憩酒店	Chiuchow_Garden_Rest
集友银行有限公司	集友_银行_股份有限公司
池州九华农村商业银行	池州_九华_农村_商业银行
Chlitina	Chlitina
克洛伊	克洛伊
Chlopskie Jadlo	Chlopskie_Jadlo
叶绿素	叶绿素
Choc'nette	Choc'nette
巧克力河口社区 FCU	chocolate_bayou_community_FCU

名称	ID
卢加诺巧克力	Chocolate_Lugano
巴西巧克力 Cacau	巧克力_巴西_可可
Choice	Choice
精选国际酒店	Choice_酒店_国际
Choices	Choices
Choithrams	Choithrams
Chokdee Dimsum	Chokdee_dimsum
创兴银行	创兴银行
尚冲	Chong_shan
崇巴火锅	Chongba_Hotpot
崇福信动物医院	崇福信_动物_医院
崇客隆	崇客隆
重庆孔亮火锅	重庆_空亮_火果
重庆刘一手火果	重庆_刘一手_Huoguo
重庆农村商业银行	重庆_农村_商业银行
重庆三峡银行	重庆_三峡银行
Chonpines	Chonpines
Chook's	Chook's
Chooka	Chooka
选择 EV	选择 EV

名称	ID
萧邦	萧邦
Chopp KREMER Ex	chopp_kremer_Express
Chorten	Chorten
Choshi商工信用组合	Choshi_商业_&_行业_信用_合作社
Choshi 信用金库	Choshi_Shinkin_Bank
Choucook	Choucook
周生生	Chow_Sang_Sang
周生生集团	Chow_Sang_Sang_Group
周大生	Chow_Tai_Seng
Chowking	Chowking
基督	基督
克里斯蒂安兄弟汽车公司	克里斯蒂安_兄弟_汽车
克里斯蒂安·迪奥	克里斯蒂安_迪奥
基督教家庭 CU	christian_family_
基督教金融大学	基督教金融_cu
克里斯蒂安·拉克鲁瓦	Christian_Lacroix
克里斯蒂安·卢布汀	克里斯蒂安_卢布汀
基督教科学教会	基督教科学教会
克里斯蒂安尼亚·贝利斯宁	Christiania_Belysning
克里斯蒂安尼亚 GlasMagasin	Christiania_ GlasMagasin

名称	ID
克里斯汀	克里斯汀
Christine LAURE	Christine_Laure
圣诞装饰	圣诞节_装饰
克里斯托弗和班克斯	Christopher_&_Banks
克里斯托弗阿里	Christopher_Ari
Christus Healt	基督健康
克里斯蒂体育	Christy_Sports
克里斯蒂市场	Christy's Market
Chroma	Chroma
镀铬心形	Chrome_Heart
Chrome 行业	镀铬工业
慢性炸玉米饼	Chronic_Tacos
计时赛	计时赛
Chronos	Chronos
克莱斯勒五星	克莱斯勒_Five_Star
Chu Chat	Chu_Chat
蔡兰的点心	Chua_lan's dim_sum
全通体育	Chuan-Tung_Sport
ChuanHsi PaTzu	ChuanHsi_PaTzu
川财证券	川财_证券

名称	ID
传仁百味	川人_百味
川味关	川味关
Chuck E Cheese	Chuck_E_Cheese's
Chuck-a-Rama	Chuck-a-Rama
咯咯笑	咯咯笑
Chugoku Rokin	Chugoku_Rokin
垂岩	垂岩
Chumbak	Chumbak
Chun K	Chun_K
唐春水	Chun_Shui_Tang
春天大酒店	春天大饭店
春夏秋东	春夏秋东
竹南农村信用合作社	Chunan_Rural_Credit_Coop_Union
钟氏料理	Chung's Cuisine
中华邮报	中华邮报
中华电信	中华电信_电信
Chungo	Chungo
中日信用金库	中日信用金库
中央劳动银行	中央劳动银行
基督教会	基督教会

名称	ID
上帝的教会	上帝的教会
预言上帝教会	预言之神教会
耶稣基督末世圣徒教会	耶稣教会_基督_后期_圣徒们
拿撒勒教堂	拿撒勒教堂
Church 的鸡	Church's Chicken
初香如香	初香如香
Chuy's	Chuy's
Château d'Ax	Château_d'ax
法国的城堡和酒店	法兰西城堡和酒店
城堡酒店系列	Châteaux_et_Hotels_Collecti
你好 Empada	cia_da_empada
Cia da Empanada	cia_da_empanada
Cia do Terno	cia_do_Terno
你好 Agip	Ciao_Agip
CiaoAgip 购物	CiaoAgip_shop
Cic Iberbanco	Cic_Iberbanco
Cicar	Cicar
Cicek Izgara	Cicek_Izgara
Cici 的披萨	Cici's_Pizza
Ciciban	Ciciban

名称	ID
Cielito Querido	Cielito_Querido
Ciencia Divertida	Ciencia_Divertida
香烟购物者	香烟_购物者
Cigkofte Battalbey	Cigkofte_Battalbey
Cigkofteci Ahmet Aybak	Cigkofteci_Ahmet_Aybak
Cigkofteci Sait Usta	Cigkofteci_sait_Usta
Cigköfteci Ömer Usta	Cigköfteci_ömer_Usta
信诺医疗集团	Cigna_Medical_Group
Cignall	Cignall
Ciming Checkup	Ciming_Checkup
Cin	Cin
辛迪的咖啡	Cindy's Coffee
CineStar	CineStar
Cinebox	Cinebox
Cinelandia	Cinelandia
3D 电影院	Cinema_3D
电影城	Cinema_City
电影公园	Cinema_Park
Cinema&Inn	Cinema&Inn
Cinemaplexx	Cinemaplexx

名称	ID
Cinemark	Cinemark
NOS 电影院	Cinemas_nos
Cinemas Pro	Cinemas_Procinal
Cinemax	Cinemax
Cinemaxx	Cinemaxx
Cinemex	Cinemex
Cineplace	Cineplace
Cineplex	Cineplex
Cinepolis	Cinepolis
Cines ABC	Cines_abc
Cines Colci	Cines_Colci
Cines Conde Duque	Cines_Conde_Duque
Cines El Punt	Cines_El_Punt
Cines IMF	cines_IMF
Cines Palafox	Cines_Palafox
Cines Renoir	Cines_Renoir
Cines Verdi	Cines_Verdi
Cinesur	Cinesur
Cineworld	Cineworld
Cinfed CU	cinfed_cu

名称	ID
Cingular 无线	Cingular_Wireless
Cinnabon	Cinnabon
辛克	辛克
Cintas 制服租赁工作服	Cintas_制服_租赁_工作服_服装
Cinzano 酒吧	Cinzano_Bar
Cinéville	Cinéville
Cipol	Cipol
Ciprotur	Ciprotur
Circle K	circle_K
Circle Reason 花园餐厅	Circle_Reason_Garden_Resta
Circont	Circont
电气回路	Circuit_Electrique
循环租车	Circular_租车
马戏马戏团	马戏团_马戏团
Cirrus 飞机	Cirrus_Arcra
思科	思科
Citadele	Citadele
馨乐庭公寓酒店	馨乐庭公寓式酒店
花旗	花旗
花旗金融	花旗财经

名称	ID
花旗趋势	城市趋势
Citibanamex	Citibanamex
花旗银行	花旗银行
花旗银行（埃及）	花旗银行_埃及
中信图书	Citic_Books
中信保诚	中信保诚
Citigo	Citigo
Citihom	Citihom
Citilab	Citilab
Citilink	Citilink
公民	公民
公民合作银行	公民合作银行
公民信贷合作银行	公民信贷合作银行
公民银行	公民银行
公民银行和信托	公民_银行_和_信托
联邦公民社区	公民_社区_联邦
城市银行	城市银行
城市烧烤	City_Barbeque
城市海滩	城市海滩
City Bites	City_Bites

名称	ID
城市混合物	城市混合物
City CU	City_cu
市中心停车	城市中心_停车
城市中心图书馆	城市中心图书馆
都市时尚	City_Chic
城市俱乐部	城市俱乐部
城市康福特旅馆	城市康福特旅馆
城市联系人	城市_联系方式
城市便利店	城市便利店
城市可丽饼	City_crepe
城市雇员信用合作社	城市_员工_信用联盟
城市快车	城市快车
城市快递便利店	City_Express_便利店
城市金融	城市_金融
城市健身	城市_健身
城市花园	城市花园
城市装备	城市装备
城市总收入	城市_Gross
城市集团酒店	酒店城市群组
城市拉尔	City_Lar

名称	ID
城市旅馆	城市小屋
城市市场	城市市场
城市市场燃料中心	城市_市场_燃料中心
城市市场药房	城市_市场_药房
城市市场	City_Markt
城市国民银行	城市_国民银行
城市公园酒店	城市公园酒店
城市停车	城市_停车
城市制药公司	City_Pharma
城市规划	城市规划
城市证监会	City_Sec
城市观光	城市_观光
城市体育	City_Sport
城市之星	City_Star
城市联合银行	城市联合银行
城市汉堡	城市_Burger
CityInn	CityInn
Cityfarma	Cityfarma
Citéa	Citéa
Citôtel	Citôtel

名称	ID
思域视频	Civic_Video
民用	民用
民航巡逻队	民用空中巡逻队
公民社区集市	Civil_Community_Mar
Civista 银行	Civista_Bank
慈溪农村商业银行	慈溪农村商业银行
Cjia公寓	Cjia_公寓
Cjia Prime 公寓	Cjia_Prime_公寓式酒店
Cjia 精选公寓	Cjia_Select_公寓
Ckp 合作银行	Ckp_Co-operative_Bank
领取 Jumper	Cleam_Jumper
Clair de Lune	clair_de_Lune
克莱尔	克莱尔
Clalit Health Serv	Calit_Health_Services
Clarine	Clarine
Clarion 系列	Clarion_系列
克拉丽奥酒店	克拉丽奥酒店
清晰度 CU	Clarity_cu
克拉克	克拉克
克拉克橡胶	Clark_Rubber

名称	ID
克拉克的 Pump-N-Shop	Clark's_pump-n-Shop
Clarks	Clarks
克拉罗	克拉罗
Classic	Classic
经典清洁剂	经典清洁剂
经典 Polo	Classic_Polo
经典咸味	Classic_Savory
经典泰迪	经典_泰迪
古典酒店	古典酒店_酒店
Clat	Clat
Claxon	Claxon
Claypot King	Claypot_King
克莱顿酒店	Clayton_酒店
干净的汽车	Clean_Car
Clean Master 汽车服务	Clean_Master_Autoservicio
Clean Master Tintorerías	Clean_Master_Tintorerías
CleanChargeSolutions	CleanChargeSolutions
CleanNet 美国	CleanNet_美国
Clear Vision 验光师	Clear_Vision_Otometrist
Clearview FCU	ClearView_FCU

名称	ID
克利尔沃特	克利尔沃特
克莱奥	克莱奥
点击鞋子	点击_鞋子
Clientis	Clientis
法国的气候	Climat_de_France
登山者	登山者
Clin d'Oeil	Clin_d'oeil
Clinicas Medicaldent	Clinicas_Medicaldent
Clinico	Clinico
cliniprev	cliniprev
Clinix	Clinix
克林顿牌	克林顿卡片
快船溪	Clipper_Creek
Clive Peeters	Clive_Peeters
Clong Kids 英语	Clong_Kids_英语
Clorofila	Clorofila
壁橱裁缝	Closet_Tailors
布料世界	布料世界
三叶草旅馆	三叶草旅舍
克洛弗代尔油漆	Cloverdale_Pa

名称	ID
克洛维亚	克洛维亚
俱乐部	俱乐部
Club 16	Club_16
阿尔多俱乐部	Club_Aldo
Avis 俱乐部	Club_Avis
地中海俱乐部	Club_Med
地中海俱乐部健身房	Club_Med_Gym
Joyview Med	Club_Med_Joyview
普拉提俱乐部	俱乐部_普拉提
Z 俱乐部！	Club_Z！
旅行俱乐部	Club_del_Viaje
Clube Turismo	Clube_Turismo
俱乐部会所旅馆	俱乐部会所_旅馆
Cluck-U Chicke	Cluck-U_Chicken
克莱德斯代尔银行	Clydesdale_Bank
克莱门特	克莱门特
颌骨牙科诊所	Clínicas_Maxilodental
诊所牙科解决方案	Clínicas_Solución _牙科
Septima 牙科诊所	Clínicas_Septima_Dental
cmyk	cmyk

名称	ID
数控	数控
Cng	Cng
Co Bigelow	Co_Bigelow
Co Co Tea	Co_Co_Tea
合作社	合作社
合作银行	合作社_银行
肯尼亚合作银行	肯尼亚合作银行
合作药房	合作社_药房
Co-Z 8 酒店	Co-Z_8_Hotel
合作银行	Co-op_Bank
合作社埃及	合作社_埃及
Co-op Kwa Jirani	Co-op_kwa_Jirani
合作药房	合作社_药房
合作食品	合作社_食品
CoCo	CoCo
CoCo 一番屋	CoCo_Ichibanya
CoCo 茶	CoCo_Tea
CoCo一番屋	CoCo一番屋
联盟	联盟
Coalve	Coalve

名称	ID
海岸	海岸
海岸服装店	Coast_服装店
海岸天然气	Coast Gas
CoastHills CU	CoastHills_CU
沿海	沿海
沿海社区 FCU	Coastal_Community_FCU
沿海农场和牧场	沿海农场和牧场
沿海租赁	Coastal_Hire
沿海集市	Coastal_Mart
沿海度假	沿海度假
海岸地区	海岸地区
海岸线 FCU	Coastline_FCU
Cobasi	Cobasi
科布	科布
补鞋匠 Plus	Cobbler_Plus
Coborn's	Coborn's
Coccimarke	Coccimarke
Cocina de María	Cocina_de_María
Coccart	Coccart
Cocokara 精	Cockara_Fine

名称	ID
椰子屋	Coconut_House
Cocoyanen	Cocoyanen
忍者代码	Code_Ninjas
Codi	Codi
Codi Cash	codi_Cash
Coen	Coen
Coffe&Co。	Coffe&Co。
咖啡 #1	Coffee_ #1
咖啡豆	咖啡豆
咖啡豆厂	Coffe_Beanery
咖啡公司	咖啡公司
咖啡研究员	咖啡研究员
咖啡民谣	Coffee_
咖啡粉	咖啡_Grounz
咖啡屋	咖啡屋
咖啡生活	咖啡_生活
像咖啡一样	像咖啡一样
咖啡新闻	咖啡_新闻
咖啡共和国	咖啡_共和国
咖啡仙门店诗歌	Coffee_Senmon-Shop _P

名称	ID
咖啡时间	咖啡时间
咖啡太妃糖	咖啡_太妃糖
咖啡世界	咖啡世界
咖啡品牌	咖啡品牌
像咖啡一样	像咖啡一样
咖啡道	咖啡道
coffiz	coffiz
科夫曼	科夫曼
Cofidis	Cofidis
Cofix	Cofix
Cogo's	Cogo's
Cohen 的时尚眼镜	Cohen's Fashion_Optical
Cohens Chemist	Cohens_Chemist
Coiff&Co	Coiff&Co
投币式洗衣机 Pierrot	Coin_Laundry_Pierot
Coinstar	Coinstar
科克斯伯里书店	Cokesbury_bookstore
Col'Cacchio	Col'Cacchio
Colcci	Colcci
Colchöes Ortobom	Colchöes_ortobom

名称	ID
冷岩	Cold_Rock
Cold Stone Creamery	Cold_Stone_Creamery
Coldwell 银行家	Coldwell_Banker
Coldwell Banker 广告	Coldwell_Banker_Commical
Cole Haan	Cole_Haan
Cole Muffler	cole_Muffler
Coles	Coles
Coles Books	Coles_Books
科尔斯快车	Coles_Exp
科莱特·海曼的 Colette	colette_by_colette_Hayman
收集积分	收集积分
收藏	收藏
柯林斯	柯林斯
科洛基	科洛基
科尔马	科尔马
科尔梅纳	科尔梅纳
殖民地	殖民地
殖民地银行	Colonial_Bank
殖民地停车场	殖民地_停车
Colonya Caixa 波伦萨	Colonya_Caixa_Pollenza

名称	ID
国际 Color Glo	Color_Glo_Internaci
给我涂上我的颜色	Color_Me_Mine
Color Plus 专卖店	Color_Plus 专卖店
颜色套装	Color_Siete
色彩工作室	色彩工作室
ColorTyme	ColorTyme
科罗拉多州	科罗拉多州
科罗拉多大学	Colorado_cu
科罗拉玛岛	科罗拉玛岛
Coloramo FCU	Coloramo_fcu
颜色条	颜色条
彩盒	彩盒
游行中的色彩	Colors_on_Parade
色彩商店	色彩商店
多彩云南	色彩缤纷_云南
Colourp	Colourp
Colpatria	Colpatria
Colsubdio	Colsubdio
科尔顿牛排馆和烧烤店	Colton's Steak_House_&_Grill
哥伦比亚	哥伦比亚

名称	ID
哥伦比亚银行	哥伦比亚银行
哥伦比亚现金和信贷	哥伦比亚_现金_和_信贷
哥伦比亚天然气输送	哥伦比亚天然气输送
哥伦拜恩 FCU	Columbine_fcu
哥伦比亚	哥伦比亚
哥伦布咖啡厅 & Co	Columbus_Café_&_Co
哥伦布咖啡	哥伦布咖啡
客观学院	colegio_Objectivo
Compac	Com_Pac
Com Tho Chay	Com_Tho_Chay
ComForcare	ComForcare
昏迷	昏迷
Comarket	Comarket
组合	组合
康姆斯蒂亚	康姆斯蒂亚
来买茶	来买茶
Comeco	Comeco
Commercial Li	Comercial_Li
墨西哥商业	商业_墨西哥
Comerica	Comerica

名称	ID
彗星	彗星
彗星清洁剂	Comet_Cleaners
Comex	Comex
很熟悉	很熟悉
Comfenalco	Comfenalco
康福特酒店	舒适酒店
康福特旅馆	康福特旅馆
安慰守护者	舒适守护者
康福特套房	舒适套房
舒适	舒适
CommStar CU	CommStar_CU
商业银行	商业银行
国际商业银行	商业_银行_国际
非洲商业银行	非洲商业银行
迪拜商业银行	迪拜商业银行
埃塞俄比亚商业银行	埃塞俄比亚商业银行
科威特商业银行	科威特商业银行
卡塔尔商业银行	卡塔尔商业银行
商业信贷公司	商业_信用_公司
商业出版社	商业_媒体

名称	ID
德国商业银行	德国商业银行
商业银行	商业银行
小卖部	小卖部
佣金快递	佣金快递
佩里准将 FCU	commodore_perry_FCU
普通美分	普通美分
联邦银行	联邦银行
联邦土地所有权保险	联邦土地所有权保险
公社	公社
沟通 FCU	通信_FCU
美国传播工作者协会	美国通讯工作者
Abilene FCU 社区	Abilene_fcu 的社区
社区银行	N.A。
社区 CU	Community_cu
社区资源 FCU	社区资源_FCU
社区服务 CU	社区服务_cu
基督共同体	基督共同体
CommunityWide FCU	CommunityWide_FCU
Como Quieres Queire Qu	como_quieres_que_te_quiera
Comod	Comod

名称	ID
Comoil	Comoil
CompUSA	CompUSA
康帕尼亚	康帕尼亚
竞技公司	公司_Athletica
Companhia do	公司_do_Campo
Churrasco 公司	companhia_do_churrasco
Comparmos Banco	Compartamos_Banco
Comper	Comper
Compleo	Compleo
完成现金	完成_现金
完整音乐和完整视频	完成_音乐_&_Complete_Video
完成	完成
康普洛特	康普洛特
CompraBene	CompraBene
康普罗 Qui	Compro_Qui
1855 年法国商行	comptoirs_de_France_1855
CompuPay	CompuPay
电脑城	计算机城
电脑增强版	电脑版_Plus
计算机复兴	计算机_文艺复兴

名称	ID
计算机疑难解答程序	计算机疑难解答
《计算机乐园》	《计算机乐园》
电脑票	电脑票
伯爵夫人	伯爵夫人
巴里伯爵夫人	伯爵夫人_du_Barry
佛罗伦萨市	Comune_di_Firenze
米兰市	comune_di_Milano
Comwell	Comwell
Con Cabeza	Con_Cabeza
康威货运	con-way_Freight
Conbipel	Conbipel
集中精力	集中精力
概念俱乐部	概念俱乐部
康科德定制清洁剂	Concord_Custom_Cleaner
康科德宠物食品和用品	Concord_Pet_Foods_&_Supplies
协和酒店	协和酒店_酒店
混凝土	混凝土
Concung.com	Concung.com
Condeco	Condeco
康迪斯	康迪斯

名称	ID
避孕套商店	Condis_Shop
康迪斯快车	康迪斯快递
Condislife	Condislife
秃鹰	秃鹰
神鹰电子	康多尔电子
圆锥区	Cone_Zone
信心	信心
Conforama	Conforama
Confortel	Confortel
Cong Ca Ph	Cong_Ca_Phe
Cong Ty Co Phan Tap Doan Hoa Sen	Cong_ty_Co_Phan_Tap_Tap_Doan_Hoa_Sen
粥的故事	Congee_Story
刚果烧烤	Congo_Grille
针叶树	针叶树
Conn's	Conn's
Connect CU	连接_cu
Connect 听力	连接听力
互联信用合作社	互联信用合作社
连接	连接
连接 CU	连接_cu

名称	ID
连接 FCU	Connects_FCU
Connex CU	connex_cu
Connexus	connexus_cu
康纳	康纳
康诺科	康诺科
Conoil	Conoil
康拉德电子	康拉德电子
康拉德的轮胎快车和全面的汽车保养	Conrad's_tire_express_&_total_car_care
康罗伊的花	Conroy's Flowers
康萨德城	Consad_City
Consvo	Consvo
Jane 的托运	托运_by_Jane
联合银行	合并银行
综合配电器	整合式电气分销商
Consort 酒店	联盟_酒店
恒定药房	康斯坦特药房
Instrua	Instrua
建筑和商业银行	建筑与商业银行
Construrama	Construrama
消费者专业版 CU	消费者_专业_cu

名称	ID
联系 Gb	联系人_Gb
联系酒店	联系我们_酒店
《当代》	《当代》
Conti's	Conti's
大陆	大陆
美国大陆航空	大陆航空_航空公司
大陆信贷	大陆_信贷
大陆论坛酒店	大陆_论坛_酒店
大陆酒店	洲际酒店_酒店
大陆贷款	大陆_贷款
大陆	大陆
大陆 Bom Dia	Continte_Bom_Dia
大陆模型	Continte_Modelo
康提公园	康提公园
康蒂合作伙伴	康蒂合作伙伴
轮廓	轮廓
轮廓快车	轮廓_Express
便捷食品市场	便捷_Food_Mart
宏观便利	便利 ncia_makro
Converse	Converse

名称	ID
烹饪门	Cook_Door
Cook out	Cook_Out
Cook-Shop	Cook-Shop
饼干工厂	Cookie_Factory
饼干人	Cookie_Man
饼干的设计	cookies_by_design
Cookmal	Cookmal
烹饪面包店	面包店厨师
很酷的博客	很酷_博客
很酷的孩子	Cool_Kids
酷河	Cool_River
很酷的旅行	酷旅游
冷藏箱	冷藏箱
酷威	酷威
Coop 论坛	Coop_Forum
Coop 福岛	Coop_Fushima
Coop Konsum	Coop_Konsum
Coop 已标记	Coop_Marked
Coop Mart	Coop_Mart
Coop Mega	Coop_Mega

名称	ID
Coop Neara	Coop_Neara
Coop Ok	Coop_Ok
Coop Pank	Coop_Pank
Coop Prix	Coop_Prix
库珀鹰酒庄和餐厅	Cooper's_hawk_winery_&_restaurant
多德卡尼斯合作银行	Dodecanese 合作银行_bank_of_doceCanese
奥罗米亚合作银行	Oromia合作社银行_bank_of
Coopeuch	Coopeuch
Coopland	Coopland
Cooplands	Cooplands
Cooratiendas	Cooratiendas
Coosa Pines FCU	coosa_pines_FCU
Copec	Copec
新奥尔良谷地	Copeland's_of_New_Orleans
copemarket	copemarket
哥伦比亚石油公司	哥伦比亚石油公司
Copifam	Copifam
Coppel	Coppel
铜烟囱	Copper_Chimney
铜州 CU	Copper_state_cu

名称	ID
Copperchimney	Copperchimney
科普斯	科普斯
国敦	国敦
复制常规	副本_General
copyrap	copyrap
跟单交易	跟单交易
CorVel	CorVel
珊瑚博彩公司	珊瑚_博彩公司
珊瑚酒店及度假村	珊瑚酒店和度假村
珊瑚旅行	Coral_Travel
科拉利亚俱乐部	科拉利亚俱乐部
Cordial Canarias 酒店及度假村	Cordial_加那利群岛_酒店_度假村
核心生活食堂	Core_Life_Eatery
CorePlus FCU	CorePlus_FCU
CoreStates 银行	CoreStates_银行
核心力量瑜伽	核心力量_瑜伽
Coricraft	Coricraft
科里奥利电信	科里奥利_电信
杯装玉米	Corn_in_cup
街角面包店	角落_面包店

名称	ID
角落面包店咖啡厅	角落_面包店_咖啡厅
角落储藏室	角落_储藏室
拐角停靠点	角落停止
街角商店	专卖店
Corner's Deli	Corner's_deli
基石大学	Cornerstone_cu
基石通信金融CU	Cornerstone_comm_financial_cu
康宁临床实验室	康宁_临床_实验室
CorningWare Corelle Revere Store	CorningWare_Corelle_Revere_Store
Corné 皇家港	Corné_Port_Royal
Corp Banca	Corp_Banca
Corpbanca	Corpbanca
Corpo 10	Corpo_10
Dermoestetica 公司	公司_Dermoestetica
企业商业银行	企业_商业_银行
公司住宿	企业住宿
Correios	Correios
阿根廷邮政	Correo_Artino
哥斯达黎加邮政	Correos_de_Costa_Rica
Corry FCU	corry_fcu

名称	ID
Corsalys	Corsalys
Corso Como	Corso_Como
Corte & Cose	Corte_&_Cose
Cortesi	Cortesi
科鲁斯	科鲁斯
Cory	Cory
Cosas Singulares	Cosas_Singulares
Cosas de Casa	环境氛围
CoseCose	CoseCose
Cosi	Cosi
Cosmed	Cosmed
Cosmedica	Cosmedica
Cosmeticar	Cosmeticar
化妆品	化妆品
化妆品市场	化妆品_市场
科斯莫	科斯莫
Cosmo Nails	宇宙指甲
Cosmo 餐厅	宇宙餐厅
CosmoProf	CosmoProf
宇宙	宇宙

名称	ID
宇宙银行	宇宙银行
Cosmos Berry	Cosmos_Berry
Cospa	Cospa
削减成本的人	Cost_Cutters
成本加上世界市场	成本加世界市场
成本更低	成本更低
科斯塔	科斯塔
布拉瓦海岸酒店	Costa_Brava_酒店
歌诗达咖啡	Costa_Coff
东海岸旅游业	costa_este_de_turismo
哥斯达黎加维达	Costa_Vida
好市多	好市多
好市多天然气	Costco_Gas
成本削减者	成本削减者
Cosway	Cosway
Cote	Cote
科托	科托
科托服装店	Coto_Clothing_Store
科茨沃尔德户外	Cotswold_Outdoor
披萨小屋旅馆	小屋旅馆_披萨

名称	ID
Cottman 全面汽车保养	Cottman_Total_Auto_Care
棉花俱乐部	Cotton_Club
科顿县	Cotton County
棉花 Ginny	Cotton_Ginny
Cotton on	Cotton_On
棉花补丁咖啡厅	Cotton_Patch_Cafe
棉花贸易商	棉花交易者
CottonOn 身体	CottonOn_Body
CottonOn 孩子们	CottonOn_Kids
科顿菲尔德	科顿菲尔德
Cottonwood 社区 FCU	Cottonwood_Community_FCU
Cotélac	Cotélac
Cou Cou	Cou_Cou
Couche-Tard	Couche-Tard
Tolleurs de Tollens	couleurs_de_Tollens
倒计时	倒计时
伯爵夫人	伯爵夫人
乡村银行	国家银行
乡村自助餐	乡村自助餐
乡村博览会	乡村博览会

名称	ID
碧桂园假日酒店	乡村花园_假日_酒店
碧桂园凤凰酒店	乡村花园_凤凰城_酒店
乡村壁炉旅馆	Country_Hearth_
乡村旅馆	乡村旅馆
丽笙乡村酒店及套房	Country_inn_&_Suites_by_radisson
乡村厨房	乡村_厨房
乡村市场	国家_市场
乡间小路	乡村之路
乡村风格	乡村风格
乡村健康	乡村健康
CountryMark	CountryMark
乡村酒店	乡村_酒店
全国住房贷款	全国_房屋_贷款
County	County
县级市场	县市场
TX 情侣	couple_tx
Courtepa	Courtepa
Courtheoux	Courtheoux
法院	法院
表兄弟	表兄弟

名称	ID
Cova	Cova
Covabra Supermercado	Covabra_Supermercado
Covan 全球搬家	Covan_全球_移动
考文垂建筑协会	考文垂建筑协会
工作服	工作服
封面故事	封面故事
Coville	Coville
牛仔鸡	牛仔_鸡
牛仔杰克的	Cowboy_Jack's
牛仔队	牛仔队
Cowin Auto	Cowin_Auto
Cox & Kings	Cox_&_Kings
Cox Health	Cox_Health
注册会计师保险	注册会计师_保险
Crabby Joe 的	Crabby_Joe's
破解	破解
Cracker Barrel 老乡村商店	Cracker_Barrel_old_Country_
手工蛋糕	手工蛋糕
起重机 CU	crane_cu
Crate & Barrel	Crate_&_Barrel

名称	ID
火山口和货轮	Craters_&_Freighters
Crave 美式厨房和寿司吧	Crave_American_Kitchen_&_Sushi_Bar
CrawsHAWS	CrawsHAWS
蜡笔	蜡笔
Crazy 8	Crazy_8
Crazy Bowls & Wr	Crazy_Bowls_&_Wraps
疯狂的克拉克的	Crazy_Clark's
疯狂的约翰斯	疯狂的约翰斯
疯狂的台词	Crazy_Line
疯狂的衬衫	Crazy_Shirts
奶油石	Cream_stone
Creamistry	Creamistry
奶油旅馆	Creamy_Inn
奶油切片	奶油切片
创意	创意
国际创意色彩	创意_色彩_国际
CrediFriuli	CrediFriuli
CrediScotia	CrediScotia
CrediScotia-Agente	CrediScotia-Agente
信贷银行	Credins_Bank

名称	ID
保加利亚农业信贷银行	Credit_Agricole_保加利亚
埃及农业信贷银行	Credit_Agricole_Egy
信贷银行	信贷银行
欧洲信贷银行	信贷_欧洲_银行
信用积极银行	信用_Positive_Bank
信用合作社	信用联盟
信用合作社 1	Credit_Union_1
信用合作社一号	Credit_union_one
丹佛信用合作社	丹佛信用合作社
信贷乌拉尔银行	Credit_ural_Bank
信贷基金大众银行	Credit-und_VolksBank
Credito Artigian	Credito_Artigiano
Credito Emiliano	Credito_Emiliano
Credito Siciliano	Credito_Siciliano
Credito Valtellinese	Credito_Valtellinese
Creditplu	Creditplus_银行
克雷兰	克雷兰
Creme Fraiche	Creme_Fraiche
Cremolatti	Cremolatti
Creno 的披萨	Creno's_Pizz

名称	ID
令人毛骨悚然	令人毛骨悚然
Crescendo	Crescendo
Crestcom	Crestcom
CreverBanca	CreverBanca
Crevier	Crevier
船员服装公司	Crew_Clothing_Co
Crick & Co	Cricket_&_Co
板球旅馆	Cricket_Inn
板球无线	板球_无线
犯罪制止者	Crime_Stoppers
Crisol	Crisol
保鲜盒	保鲜盒
水晶酒店及度假村有限责任公司	Cristal_Hotels_and_resorts_LLC
水晶水疗中心	Cristal_Spa
小动物控制	Critter_Control
Croasonho	Croasonho
鳄鱼	鳄鱼
鳄鱼国际	鳄鱼_国际
Crocs	Crocs
牛角包快递	牛角包_Express

名称	ID
法兰西红十字会	Croix-Rouge_Française
Crosby's 便利店	Crosby's 便利店
CrossCafe	CrossCafe
CrossFit	CrossFit
克罗斯兰经济型一室公寓	克罗斯兰经济工作室
Crosspoint FCU	crosspoint_fcu
十字路口	十字路口
填字游戏	填字游戏
王冠	王冠
皇冠汉堡	Crown_Burgers
皇冠城堡国际	皇冠城堡_国际
皇冠中心	皇冠中心酒店
皇冠炸鸡	Crown_Fried_Chicken
皇冠力帆	Crown_Lifan
皇冠酒	Crown_Liquors
皇冠斯特林套房酒店	Crown_Sterling_Suites
皇冠奖杯	皇冠_奖杯
皇冠假日度假村	皇冠假日度假村
皇冠假日套房酒店	皇冠假日套房酒店
皇冠云朵酒店	Crowning_Clouds_Hotel

名称	ID
克鲁塞马尔·克鲁塞罗斯	Crucemar_Cruceros
邮轮策划师	邮轮_策划者
CruiseOne 专家	CruiseOne_专家
Crumbl 饼干	Crumbl_Cookies
紧缩健身	Crunch_Fitness
外壳披萨	外壳_披萨
克鲁兹·罗哈	Cruz_Roja
佛得角	Cruz_Verde
Crysanna	Crysanna
水晶宝石	Crystal_Bijou
水晶闪光	水晶闪光
水晶玉	Crystal_Jade
水晶玉我的面包	Crystal_Jade_My_Bread
橙色水晶酒店	水晶橙色酒店
安道尔信贷银行	Crédit_Andorra
法国农业信贷银行	Credit_Agricole
摩洛哥农业信贷银行	Crédit_Agricole_du_Maroc
Crédit Coopératif	Credit_Coopératif
Crédit Foncier	Credit_Foncier
海事信贷银行	Credit_Maritime

名称	ID
Crédit Mutuel	Credit_Mutuel
Credit Mutuel de Bretagne	Crédit_mutuel_de_Bretagne
Credit Mutuel du Senégal	Crédit_mutuel_du_Senegal
阿尔及利亚大众信贷	Crédit_Populaire_d'Algérie
摩洛哥信贷银行	Crédit_du_Maroc
Credit du Nord	Credit_du_Nord
脑脊液市场	Csf_Market
Cu Cha Dan Fan	Cu_Cha_Dan_Fan
Cub Foods	Cub_Foods
Cub 药房	小熊药房
Cubus	Cubus
Cue	Cue
崔峰源	崔峰源
Cuisine	Cuisine_Plus
Cuisinella	Cuisinella
美食参考文献	Cuisines_Références
Cukiernia Sowa	Cukiernia_Sowa
Culligan 加拿大	Culligan_Canada
库里根水系统	Culligan_Water_Systems
Cult.Fit	Cult.Fit

名称	ID
Cultura	Cultura
英国文化	Cultura_Inglesa
Culver's	Culver's
精液书	Cum_Books
坎伯兰建筑协会	坎伯兰建筑协会
坎伯兰农场	坎伯兰农场
Cura	Cura
库拉索	库拉索
Curaleaf	Curaleaf
希尔顿古玩系列	希尔顿古玩收藏_by_hilton
Curiol	Curiol
咖喱冠军	咖喱冠军
Currys	Currys
Currys Donuts	Currys_Donuts
Currys 电脑世界	Currys_PC_World
Currys.Digit	Currys.Digit
曲线	曲线
Cut & Style 沙龙	Cut_&_Style_Salon
剪裁和颜色	cut_and_colors
削减健身中心	Cutting_up_Fitness_Center

名称	ID
Cvetopttorg	Cvetopttorg
Cybelar	Cybelar
网络探索	网络探索
数码港	数码港
骑自行车的欢乐	Cycle_Joy
Cyclon	Cyclon
Cygte	Cygte
塞浦路斯 CU	塞浦路斯_CU
塞浦路斯合作银行	塞浦路斯_合作银行
Cass na Herbate	czas_na_Herbate
捷克斯车 CS	czechocar_CS
CORTÉ 路线	COTE_ROUTE
Celebres Viajes	Celebres_Viajes
Girls 代码	Codigo_Girls
Côté Café	Côte_Café
Côté France	法国海岸
Côté Jardin	Côte_Jardin
D C Books	D_C_Books
D Mart	D_Mart
D Six	D_Six

名称	ID
D&D 牙科中心	D&D_Centros_Dentales
D&G Autocare	D&G_Autocare
D&P Parfum	D&P_Parfum
D&R	D&R
D&W 食品中心	D&W_Food_Centers
D&X	D&X
D'Cost 海鲜	D'_Cost _海鲜
'ANGELO	'ANGELO
D'Damas	D'Damas
D'Mart	D'Mart
D'Oliva al FORNO	d'Oliva_al_Forno
D'S damat	ds_damat
d'arte	d'arte
D 型链	D 型链
D-Lima	D-Lima
D-Mart	D-Mart
D SMART	D SMART
D 风格	D 风格
D-VIAGEM	D-VIAGEM
D.L. Evans Bank	D.L._Evans_Bank

名称	ID
D. 电话	D. 电话
D.R. Horton	D.R._Horton
D1	D1
D2	D2
从米兰来	DA_MILANO
大神林	DA_SHEN_LIN
DA YU LANG	DA_YU_LANG
达契亚	达契亚
大方便利店	大方便利店
DAHON	DAHON
DAHSING BANK	DAHSING_BANK
大门	大门
大发	大发
DAKS	DAKS
DAMART	DAMART
DAMING 光学	该死的光学
达米勒	达米勒
DANBO MÖBLER	DANBO_MÖBLER
丹达拉	丹达拉
DANEE	DANEE

名称	ID
DANIJA	DANIJA
DANKAN	DANKAN
DAPHINE	DAPHINE
亲爱的	亲爱的
DAS	DAS
DAS FUTTERHAUS	DAS_FUTTERHAUS
DAS 市场	达斯_市场
冲进去	DASH_IN
大通证券	大通证券
DATS	DATS
DATS 24	DATS_24
DATSUN	DATSUN
DAV 学院	DAV 学院
DAV 学校	Dav_School
大卫琼斯	DAVID_JONES
大卫人	DAVID_PEOPLE
达芬奇咖啡	达芬奇咖啡
今天一天	今天_今天
DAYALI	DAYALI
DBS	DBS

名称	ID
DCA	DCA
DCB 银行	DCB_bank
DCH	DCH
DCH FOOD	DCH_FOOD
DCM	DCM
DCM	DCM_
DCM	DCM
DCS	DCS
DDP	DDP
DE CASA	DE_CASA
DE 巧克力咖啡	DE_COLATE_COFFEE
DE LINDEBOOM APOTHEEK	DE_LINDEBOOM_APOTHEEK
DE voorzorg Apotheken	de_voorzorg_apotheken
DE-STA-CO	DE-STA-CO
DEAN & DELUCA	DEAN_&_DELUCA
十日谈	十日谈
十项全能	十项全能
DECIMAS	DECIMAS
DECJUBA	DECJUBA
DEEN 超市	Deen_Supermarkte

名称	ID
DEFACHANG	DEFACHANG
DEITECH	DEITECH
DEK 计算机中心	dek_computer_Cent
DEKORPEN	DEKORPEN
DEKRA	DEKRA
DEL SOL	DEL_SOL
DELAISU	DELAISU
DELFI	DELFI
DELFIN	DELFIN
DELHAIZE	DELHAIZE
熟食兄弟	DELI_BROSTER
DELI 糕点店	deli_the_pastry_Shop
DELIFR	DELIFR
DELIKATESO	DELIKATESO
DELKO	DELKO
三角洲酒店	达美酒店
DEMLIK	DEMLIK
DENI CLER	DENI_CLER
DENIRO 汉堡系列	Deniro_Hamburgueseria
DENKO	DENKO

名称	ID
丹尼斯	丹尼斯
牙科中心 Adontología Avanzada	dental_center_odontología_Avanzada
DENTIX	DENTIX
DENTTOS Sonrisas Blancas	denttos_sonrisas_Blancas
DEPASO	DEPASO
DER Reiseburo	der_reiseburo
DERENTANG	DERENTANG
DERIMOD	DERIMOD
绝望	绝望
DESPO	DESPO
DETAILCar	DETAILCar
DEVERNOIS	DEVERNOIS
DEVK	DEVK
DEVRED	DEVRED
DEXIA	DEXIA
DEXIM	DEXIM
德庄	德庄
DF	DF
DFAC	DFAC
DFPV	DFPV

名称	ID
DFSK	DFSK
DHB	DHB
DHIPAYA 保险	DHIPAYA_保险
DHL	DHL
DI SANTINNI	DI_SANTINNI
DIA 市场	dia_mark
DIACOR	DIACOR
对角线	对角线
DIAMANTINE	DIAMANTINE
钻石眼花	DIAMOND_DAZZLE
戴安娜·简斯	DIANA_JANES
黛安·冯·弗斯滕伯格	diane_von_Furstenberg
DICICO	DICICO
DICKIES	DICKIES
DICOS	DICOS
滴滴	滴滴
DIELMAR	DIELMAR
柴油	柴油
DIFFERENCE	DIFFERENCE
DIKSI	DIKSI

名称	ID
DIMEGLIO	DIMEGLIO
你的 TÖJMAND	DIN_TÖJMAND
DINA	DINA
迪纳摩	迪纳摩
DING GO	DING_GO
叮茶	DING_TEA
DIO	DIO
迪翁	迪翁
DIONWIRED	DIONWIRED
直接卫星电视	DIRECT_SAT_TV
直接光学	DIREKT_OPTIK
DIS	DIS
DISA	DISA
折扣	折扣
碟	碟
DISPANO	DISPANO
DITA	DITA
DIVARESE	DIVARESE
DIY 仓库	自制仓库
DK	DK

名称	ID
DK Fuel	dk_fuel
DK-benzin	DK-benzin
DKNY	DKNY
DKNY 牛仔裤	DKNY_JEANS
DNA KAUPPA	DNA_KAUPPA
DNA 天然	DNA_Natural
脱氧核糖核酸存储	脱氧核糖核酸存储
DNB	DNB
DNS	DNS
自己动手	自己动手
玩得开心	DO_WELL_FUN
DOBROTsEN	DOBROTsEN
码头工人	码头工人
DOCKX	DOCKX
躲避	躲避
多哈银行	DOHA_BANK
多哈伊斯兰银行	DOHA_ISLAMIC_BANK
DOJA	DOJA
十字星	十字星
DOK	DOK

名称	ID
DOLLAMA	DOLLAMA
美元商店	美元商店
DOLLY DIMPLE 的	DOLLY_DIMPLE'S
DOLPHY	DOLPHY
DOM.RF	DOM.RF
DOMANI	DOMANI
圆顶	圆顶
自治权	自治权
DOMO 消费类电子产品专卖店	domo_消费类电子产品专卖店
多纳泰罗	多纳泰罗
东升白火	DONG_SENG_BAI_HUO
东风商用	东风_广告
东风日产	东风_日产
东海证券	东海证券
东兴证券	东兴证券
DONQ	DONQ
DORAGLASS	DORAGLASS
DORCUS	DORCUS
DORMERO	DORMERO
希尔顿逸林酒店	希尔顿逸林酒店

名称	ID
DPaschoal	DPaschoal
ZDROWIE 博士	DR_ZDROWIE
DR.MED	DR.MED
DRACOIL	DRACOIL
港龙航空	港龙航空
DRAGONSTAR	DRAGONSTAR
DRESSMANN	DRESSMANN
饮酒快车	DRINKING_EX
超级便宜的药物	DROGAS_SUPER_BARATAS
DROGASMIL	DROGASMIL
DROGERIE Natura	Drogerie_natura
DROMEAS	DROMEAS
药店健康	药店_健康
DRUNI	DRUNI
干洗 TO-YOUR-DOOR	干洗上门
DRY-B-LO	DRY-B-LO
美国干洗	DRYCLEAN_USA
DS	DS
DSCANSO	DSCANSO
DSK 银行	dsk_bank

名称	ID
DSW	DSW
DTDC	DTDC
DTLR 别墅	DTLR_VILLA
DTR	DTR
DU	DU
杜芳子	DU_FANG_ZI
迪拜银行	迪拜银行
DUBU HOUSE	DUBU_HOUSE
杜成快餐	都城_快餐
DUCTBUSTERS	DUCTBUSTERS
DUDALINA	DUDALINA
正好在南方	DU_SOUTH
DUFFIN DAGELS	DUFFIN_DAGELS
DUHANPROMET	DUHANPROMET
杜卡	杜卡
DUMOND	DUMOND
DUNKIN'COFFE	DUNKIN'COFFE
都喜天丽酒店	DUSIT_THANI
DV8	DV8
DVF	DVF

名称	ID
DWONKU	DWONKU
DYO	DYO
DYZURNA	DYZURNA
DZS	DZS
Da Bei 照片	Da_Bei_Photo
Da Fu Lan	Da_Fu_Lan
Da Grasso	Da_Grasso
Da 杂货店	da_杂货店
大龙怡火锅	Da_Long_Yi_Hopt
大牛火锅	大牛火锅
大鹏火锅	Da_Pang_Hot_Pot
大埔铁板烧	da_Pu_Tie_Ban_Shao
大信店	Da_Xin_Shop
Da Yung's Tea	Da_Yung's Tea
大寨门	Da_Zhai_Men
DaVita 透析中心	DaVita_透析中心
大安药房	达安_药房
大宾家川川火锅	大宾家_川川_火锅
达布尔	达布尔
爸爸面团	Daddy_Dough

名称	ID
爸爸甜心吐司	Daddy_Sweety_Toast
达迪	达迪
大地影院	Dadi_Cinema
达迪学校	Dadi_School
大东	大东
Daduizhang	Daduizhang
大长健	Dae_Jang_Gum
大邱银行	大邱银行
Daerduo	Daerduo
大宇	大宇
大宇快车	大宇快车
Dafang	Dafang
达吉	达吉
Dagli'Brugsen	Dagli'Brugsen
达戈斯蒂诺超市	达戈斯蒂诺_超市
Dagwinkel	Dagwinkel
大河寿司	Dahe_Sushi
Dai ABank	Dai ABank
大一未来信用金库	Daichimirai_Shinkin_Bank
Daigaku Juken Navio	Daigaku_juken_Navio

名称	ID
第一电池	第一电池
第一劝业信用合作社	Daichi_kangyo_Credit_合作社
大黑药房 100日元店	大黑_药品_100日元_店
Dailo	Dailo
每日新鲜	每日新鲜食品
每日价格	每日价格
每日的	每日的
DailyDae	DailyDae
大明火锅	大明_火锅
乳品市场	Dairy_Mart
乳业女王	乳制品女王
乳业谈话	Dairy_Talk
大沙林	大沙林
大师教育	大师_教育
大创	大创
Daisy Mart	Daisy-Mart
Daisyosuisan	Daisyosuisan
大东协信用合作社	Daitokyo_Credit_合作社
达科他手表公司	达科他手表公司
西达科他州 CU	dakota_West_cu

名称	ID
达克辛比哈尔邦格拉明银行	Dakshin_Bihar_Gramin_Bank
Dal-Tile	Dal-Tile
戴尔·卡内基训练	戴尔_卡内基培训
大连农村商业银行	大连农村商业银行
达尔迈耶斯巴克胡斯	Dallmeyers_Backh
大龙火锅	大龙火锅
Daly 连衣裙	Daly_Dress
达马凯	达马凯
达曼·卡什	Daman_Cash
达玛斯珠宝	大马士革_珠宝
Damat Tween	Damat_Tween
达姆拉汽油	Damla_Petrol
达姆拉汽油市场	Damla_Petrol_Market
Damon's Grill	Damon's_Grill
达姆罗家具	达姆罗_家具
丹丹韩宝	Dan_dan_Han_Bao
Dan Dooley	Dan_Dooley
丹·墨菲的	Dan_Murphy's
Dan-yo 信用合作社	dan-yo_credit_合作社
DanHen	DanHen

名称	ID
Dana Choga 的厨房	dana_choga's Kitchen
Dana Choga 的餐厅	dana_choga's 餐厅
达纳特酒店及度假村	达纳特酒店及度假村
丹波	丹波
跳舞的狼	Dancing_Wolves
丹东农村商业银行	丹东农村商业银行
花花公子迷你超市	Dandy_Mini_Mart
戴恩县 CU	dane_county_cu
Dang Quang Watch	Dang_Quang_Watch
Dangee Dums	dangee_Dums
丹格菲尔德	丹格菲尔德
达尼亚	达尼亚
大良水饺	大良_水饺
丹尼尔·卡辛	Daniel_Cassin
丹尼尔·赫希特	Daniel_Hechter
丹尼尔·惠灵	丹尼尔_惠灵顿
丹尼尔皮革	丹尼尔_皮革
Danmaya Suisan	danmaya_suisan
Dansk Kroferie	Dansk_Kroferie
丹斯克银行	Danske_Bank

名称	ID
Danske Mobler	danske_Mobler
但丁咖啡	但丁咖啡
丹比乌斯酒店	Danubius_酒店
丹阳农村商业银行	丹阳_农村_商业银行
道明城教育	道明成_教育
daxiaoman	daxiaoman
达芙娜·莱文森 HDL	daphna_levinson_HDL
达芙妮的希腊咖啡馆	Daphne's_Greek_Cafe
大桥外国语学校	大桥_外语_学校
Dar Al Amane	dar_Al_Amane
达尔阿萨法	dar_Assafaa
Dar e Arqam School	dar_e_arqam_school
Daredzhani	Daredzhani
大吉岭	大吉岭
达灵顿建筑协会	达灵顿建筑协会
达荣和	达荣和
达雷尔·李	Darrell_Lea
Darry Ring	Darry_Ring
Darty	Darty
Daruton	Daruton

名称	ID
达斯啤酒屋	Das_Bierhaus
破折号	破折号
大申银行	大申银行
Dashichazuke En	dashichazuke_En
大信新都汇	dasin_新都汇
Dasin Youkejie	dasin_Youkejie
Dat Backh	Dat_Backhus
数据医生	数据_医生
数据快车	数据快车
Datart	Datart
Datelli	Datelli
大同证券	大通证券
大同华	大同华
Davanni's Pizza & Hoagies	Davanni's_pizza_&_hot_Hoagies
戴夫和巴斯特的	Dave_&_Buster's
Dave's Cosmic Subs	Dave's_cosmic_Subs
Davey Tree 专家	Davey_Tree_专家
大卫劳伦斯	大卫_劳伦斯
大卫·威克利家园	David_Weekley_Homes
大卫的	大卫的

名称	ID
大卫的新娘	David's_Bridal
大卫的茶馆	David's_Tea_House
戴维斯超市	大卫斯超市
戴维斯	戴维斯
戴维森石油和天然气公司	戴维森石油和天然气
Davivienda	Davivienda
达瓦尔的开放世界	dawar's Open_World
达希豪	达希豪
大兴安岭农村商业银行	大兴安岭_农村_商业银行
白天和黑夜	白天和夜晚
Day Air	day_air_cu
戴刘易斯药房	Day_Lewis_Pharmac
日光沙拉	Day_Light_Salads
美好的一天早午餐和咖啡	Day_Nice_Brunch_&_Coffee
黎明	黎明
Dayjet	Dayjet
日光甜甜圈	Daylight_Donuts
Daymet CU	daymet_cu
戴斯酒店及套房	Days_酒店与套房
戴斯酒店	Days_Inn

名称	ID
戴斯酒店商务场所	Days_inn_Business_Pl
这样的日子	天_Like_This
戴斯套房	日间套房
daystop	daystop
大章	大章
大中	大中
达州市商业银行	达州市商业银行
Dazzle	Dazzle
Db 汽车共享	Db_汽车共享
Db Mart	Db_Mart
Dbank	Dbank
Dcrepes	Dcrepes
De Bakkerswinkel	De_Bakkerswinkel
戴比尔斯	De_Beers
De Beren Eetcafés	De_Beren_Eetcafés
De Cosas Hogar & Moda	de_Cosas_Hogar_&_Moda
De Cosas	Hogar y Moda
De Flores 和 Floreros	de_flores_y_floreros
De Graaf 车底下	De_Graaf_under-Car
De Harense Smid	De_Harense_Smid

名称	ID
Historike	De_Historke
德马德里	De_Madrid
De Pannekoekenbakker	de_Pannekoekenbakker
De Pizzabakkers	De_Pizzabakkers
De Slegte	De_Slegte
De Smet & Van Diest	de_Smet_&_van_Diest
De Tuinen	De_Tuinen
De Vere	De_Vere
De Woon 专家	De_Woon 专家
de5en5	de5en5
DeAngelo的比萨店公司	DeAngelo's_pizzeria_compan
deBorla	deBorla
DeFacto 服装	DeFacto_服装
DeHui	DeHui
DeSerres	DeSerres
DeVry 大学	DeVry_大学
交易	交易
交易	交易
Dealz	Dealz
Deamplace 酒店及度假村	Deamplace _酒店及度假村

名称	ID
Dearzhao 创意餐厅	Dearzhao_Creative_Resta
Deb Shops	Deb_Shops
德拜	德拜
德百家店	德百家店
Debenhams	Debenhams
Debonairs 披萨	Debonairs_Pizz
德卡	德卡
Decenio	Decenio
甲板墙壁	Deck_The_Walls
装饰家居	Deco_Home
装饰与你	Decor_&_You
Dedeman	Dedeman
德德曼酒店及度假村	德德曼_酒店及度假村
Deefly 酒店及度假村	Deefly_酒店及度假村
深	深
深度睡眠酒店	深度睡眠酒店
Deepika	Deepika
国防银行	国防银行
玷污	玷污
脱落	脱落

名称	ID
德加光学	Dega Optical
Degani	Degani
Degirmen Pastaneleri	Degirmen_Pastaneleri
德固萨银行	Degussa_Bank
德汉牙科诊所	Dehan_Dental_Clinic
Dehner	Dehner
Deichmann	Deichmann
德伊勒市中心有限责任公司	deira_city_centre_LLC
Dejavu	Dejavu
德卡家具装饰	Deka_Mobiliario-Decoracion
Dekamarkt	Dekamarkt
Dekk1	Dekk1
Dekkmann	Dekkmann
甲板服务	甲板服务
Dekteam	Dekteam
德尔福尔诺	Del_Forno
德尔诺特 CU	del_norte_cu
Del Papa	Del_Papa
德尔里奥	Del_Rio
Del Taco	Del_Taco

名称	ID
德尔苏尔	del_sur
德尔塔银行	达美银行
Delavska hranilnica	delavska_Hranilnica
特拉华州北部	特拉华州北部
德尔巴德	德尔巴德
Delboni Auriemo	Delboni_Auriemo
Delek	Delek
德里美味	德里_美味
德里公立学校	德里公立学校
德里国家合作银行	德里州立合作银行
Delhivery	Delhivery
熟食店	熟食店
熟食咖啡厅	熟食咖啡厅
熟食教育	熟食教育
Deli Mart	Deli_Mart
熟食时间	Deli_Time
Deli de Luca	deli_de_Luca
迪利亚斯	迪利亚斯
Delicass	Delicass
美味火锅	Delicious_Hotpot

名称	ID
美味	美味
Delika	Delika
熟食中心	Delikatesy_Centrem
Delina's	Delina's
德利威	德利威
德尔卡	德尔卡
Dell	Dell
戴尔 Anno	Dell_Anno
德拉维亚酒店	Della_Via
德尔蒙面包店	德尔蒙面包店
德勤	德勤
德尔福银行	Delphi_Bank
Delsey	Delsey
Delta	Delta
达美航空	达美航空
三角洲索尼克	Delta_Sonic
达美贸易	达美_交易
达美快递	达美快递
deltasul	deltasul
德伦牙科	Delun_Dental

名称	ID
豪华旅馆	Delux_Inn
Delvaux	Delvaux
热带熟食店	Delírio_Tropical
德迈生	德迈生
Demark	Demark
Demi Plié	Demi_Plié
Demirdokum	Demirdokum
Den Jyske Sparekasse	den_Jyske_Sparekasse
Den Sorte Gryde	Den_Sorte_Gryde
Dencio's	Dencio's
Denizbank	Denizbank
Denizen	Denizen
丹纳	丹纳
Denns De	Denns_De
Denny's	Denny's
牙科公司	牙科公司
牙科医生	牙科医生
牙科医院	牙科_医院
DentalWorks	DentalWorks
牙齿咬伤	牙齿咬伤

名称	ID
牙科	牙科
Dentalis	Dentalis
Dentalive	Dentalive
Dentzano USA	Dentzano_USA
丹佛社区 CU	Denver_community_cu
丹佛床垫	丹佛_床垫
拒绝体育	否认_体育
Denza	Denza
丹泽尔驱动器	丹泽尔驱动器
Depanneur Voisin	Depanneur_Voisin
机动车辆部	机动车辆部
depilex	depilex
车库	车库
Deppon	Deppon
德清农村商业银行	德清农村商业银行
Der Beck	Der_Beck
Der Backer Feihl	Der_Backer_Feihl
Der Kalchreuther Backer	Der_Kalchreuther_Bácker
Der Mann	Der_Mann
Derag	Derag

名称	ID
德朗升级版	Derang_Plus
德比酒店	德比_酒店
德比郡建筑协会	德比郡建筑协会
Derendinger	Derendinger
德雷斯	德雷斯
Deriden	Deriden
Desa	Desa
德斯坎普斯	德斯坎普斯
下山	下山
Deseret Book	Deseret_Book
Deseret First CU	deseret_first_cu
沙漠工业	Deseret_Industries
沙漠河流大学	Desert_rivers_cu
沙漠日光浴沙龙	沙漠太阳晒黑沙龙
Deshabhimani	Deshabhimani
德尚县银行	德尚县银行
deshyovaya apteka	deshyovaya_apteka
设计酒店	设计_酒店
设计触手可及	触手可及的设计
设计师鞋库	设计师_鞋类_仓库

名称	ID
设计师市场	设计师市场
设计画廊	设计画廊
Designtorget	Designtorget
Desigual	Desigual
Dessange	Dessange
目的地孕妇	目的地_孕妇
目的地 XL	目的地_XL
目的地信用合作社	目的地_信用合作社
Destroy	Destroy
详细信息	详细信息
Deti	Deti
德茨基·米尔	detskie_mir
detskiy tsentr Syma	detskiy_tsentr_syoma
德意志Apotheker-and Arztebank	deutsche_apotheker-_und_ArzteBank
德意志银行	德意志银行
德意志信用银行股份公司	deutsche_Creditbank_ag
德意志租金保险	Deutsche_Rentenversicherung
Deutschepost	德国邮政_德国
埃塞俄比亚开发银行	埃塞俄比亚开发银行
赞比亚开发银行	赞比亚开发银行

名称	ID
菲律宾开发银行	菲律宾开发银行
开发信贷银行	开发_信贷_银行
戴维斯保险	Deves_保险
德维安	德维安
恶魔进化	恶魔进化
德文信贷	德文信贷
杜威害虫防治	Dewey_Pest_Control
杜威披萨	Dewey's Pizza
德克斯一号	Dex_One
德克斯特	德克斯特
德友	德友
德悦	德悦
德月楼	德月楼
德泽尔纳斯洛文尼亚银行	Dezelna_Banka_Slovenije
德州银行	德州银行
dfscala	dfscala
达巴-埃斯特德 1986 德里	Dhaba_-_Estd._1986_Delhi
Dhanalakshmi Srinivasan Chit Funds	Dhanalakshmi_Srinivasan_Chit_Funds
Dhanlaxmi 银行	Dhanlaxmi_Bank
Dhobilite	Dhobilite

名称	ID
Di	Di
Di PUNTO	di_punto
Di Piu Milano	Di_Piu_Milano
Di Wang Shibu jiang mu	di_wang_Shi_Bu_Jiang_Mu_Ya
狄义顺	狄义顺
骰子	骰子
DisCom	DisCom
迪亚	迪亚
对角线酒店	对角线_酒店
给床打电话	dial_a_bed
Dial-A-Story	Dial-A-Story
方言	方言
拨号大奖	拨号大奖
迪亚尔苏尔	迪亚尔苏尔
钻石银行	钻石银行
钻石汽车玻璃	钻石汽车玻璃
钻石银行	Diamond_Bank
钻石泡沫	钻石_泡沫
钻石湖 FCU	Diamond_Lakes_FCU
钻石停车	钻石停车

名称	ID
钻石凯旋玻璃	钻石_Triumph_Glass
钻石信托银行	钻石信托银行
钻石鸟画颜料	Diamond_Vogel_Paint
Dian Pelangi	Dian_Pelangi
戴安娜	戴安娜
戴安娜法拉	戴安娜_法拉利
点太香火锅	点太香_火锅
点月成香	点月成香
Diaoye	Diaoye
Dibo	Dibo
Dick Blick 艺术材料	Dick_Blick_Art_Materials
迪克的体育用品	Dick's_Sporting_Goods
迪克的翅膀和烧烤	Dick's_Wings_&_Grill
Dickey 的烧烤炉	dickey's Barbcue_pit
Didongviet	Didongviet
不来梅Sparkasse	Die_Sparkasse_Brem
迪博尔德	迪博尔德
迭戈	迭戈
Dien May Xanh	Dien_May_Xanh
Dienthoaivui	Dienthoaivui

名称	ID
Dierbergs	Dierbergs
饮食中心	饮食中心
Dietplus	Dietplus
Diez Euros	Diez_Euros
Diffa	Diffa
不一样的 Twist 椒盐脆饼公司	Differt_Twist_Pretzel_Co
Dig Inn	Dig_Inn
Digi	Digi
数字化	数字化
数字设备公司	数字设备公司
数字 FCU	digital_fcu
数字手机	数字_移动
Digiturk	Digiturk
尊严 Health	尊严_健康
Digros	Digros
Dikaya Orkhideya	Dikaya_Orkhideya
迪莱克	迪莱克
Dilek Hali	Dilek_Hali
Diletto CAFé	Diletto_Café
莳萝的	莳萝的

名称	ID
Dillard's	Dillard's
狄龙	狄龙
狄龙燃料中心	狄龙燃料中心
狄龙市场	狄龙市场
狄龙 Marketplace 燃料中心	Dillons_Marketplace_Fuel_Cen
狄龙市场 Marketplace	狄龙_市场_药房
狄龙药房	狄龙斯药房
昏暗	昏暗
Dim Dou Duk	Dim_Dou_Duk
Momo 点心局	Dim_Sum_Bureau_of_Momo
一毛钱储蓄银行	Dime_Savings_Bank
迪莫尼	迪莫尼
鼎泰丰	丁泰峰
Din-X	Din-X
DinSko	DinSko
Dinarsu Hali	Dinarsu_Hali
迪娜兹	迪娜兹
Dindigul Thalappakatti	Dindigul_Thalappakatti
Dindigul Thalappakatti Restaurant	Dindigul_thalappakatti_Restaurant
Diner's	Diner's

名称	ID
DineroExp	DineroExp
丁歌	Ding_Ge
丁丁香	丁丁香
定海大洋农村商业银行	定海_海洋_农村_商业_银行
dinghongding 澳门元	定中鼎_澳门_美元
晚餐	晚餐
Dion's	Dion's
迪奥娜	迪奥娜
狄奥尼索斯	狄奥尼索斯
迪奥	迪奥
外交官	外交官
Dippin'Dots	Dippin'_Dots
直接汽车保险	直接汽车保险
直接能量	直接能源
DirectBuy	DirectBuy
方向 CU	方向_cu
Direkten	Direkten
Direktna Banka	Direktna_Banka
迪拉姆快车	迪拉姆快车
Dirigo FCU	dirigo_fcu

名称	ID
Dirk van den Broek	dirk_van_den_broek
Dirx	Dirx
Dis-Chem 药房	dis-chem_Pharmacies
美国残疾退伍军人	残疾人_美国_退伍军人
迪斯科	迪斯科
折扣银行	Discount_Bank
折扣汽车和卡车租赁	折扣_汽车和卡车_租赁
折扣药店	Discount_Drug_Mart
折扣药店	Discount_Drug_Store
折扣市场	Discount_Markt
折扣店 Attax	Discount_Store_Attax
折扣轮胎	Discount_Tire
探索公园	探索公园
探索玩具教育顾问	Discovery_Toys_教育_顾问
Disgor	Disgor
菜网	菜网
Diskont	Diskont
Dislub Energia	Dislub_Energia
Disma	Disma
迪士尼商店	迪士尼商店

名称	ID
巴黎迪士尼乐园	巴黎迪士尼乐园度假村
Disport	Disport
Dispreu	Dispreu
Dist 合作银行 Lakhimpur Kheri	dist_co-op_bank_lakhimpur_kheri
DistriCenter	DistriCenter
编辑管理	编辑管理
Dithmarscher vr-Bank	dithmarscher_vr-Bank
Ditsch	Ditsch
Ditt Apotek	Ditt_Apotek
著名女歌唱家	著名女歌唱家
Diva 服装店	Diva_Attires_Store
Divani & Divani	Divani_&_Divani
迪瓦尼斯连锁酒店	迪瓦尼斯_连锁酒店
多样化	多样化
多元化会员 CU	多元化成员_cu
Diverta	Diverta
Diviavelodi	Diviavelodi
Divino Fogão	Divino_Fogão
Divizion	Divizion
Dixie Chili and Deli	dixie_chili_and_deli

名称	ID
Dixika	Dixika
Dixis	Dixis
Dixons	Dixons
Dixy	Dixy
迪克西鸡	Dixy_Chicke
Djaweb	Djaweb
Djelo	Djelo
Djezzy	Djezzy
德鲁斯兰银行	Djurslands_Bank
Dlya Vas	Dlya_Vas
Dlya berezhlivykh	dlya_berezhlivykh
dnB NORD	dnb_Nord
Dna	Dna
Do Fu	Do_Fu
尽力而为	尽力而为
不做	do_mo
Do&Me	Do&Me
DoRego的	DoRego的
Dobriy Aptekar'	Dobriy_Aptekar'
Dobriy Den'	Dobriy_den'

名称	ID
Dobriy Doktor	Dobriy_doktor
Dobropek	Dobropek
Dobrya leki	dobrya_leki
DocMorris	DocMorris
DocMorris APOTEK	DocMorris_APOTEK
Doce	Doce
Dochki i Synochki	dochki_i_synochki
码头游戏	Dock_Games
船坞	船坞
IT 博士	docteur_it
博门塔尔博士	doctor_Bormental
脚医生	Doctor_Feet
Beck & Stone 医生宠物护理	Doctors_Beck_&_Stone_Pet_Care
国防部	国防部
道奇五星	Dodge_Five_Star
道奇的南方风格	Dodge's_Southern_Style
渡渡鸟披萨	渡渡鸟披萨
Dodoni	Dodoni
多德森害虫防治	Dodson_Pest_Control
Doeland	Doeland

名称	ID
Dogger	Dogger
Dogta	Dogta
狗托邦	狗托邦
Doi Chaang 咖啡	Doi_Chaang_Coffee
Doikham	Doikham
Doktor	Doktor
Alvik 博士	doktor_Alvik
Dontor Dent	doktor_Dent
斯托列托夫博士	doktor_Stoletov
Dolar Shop 经典时尚火锅	Dolar_Shop_Classic_Fashion_Hotpot
Dolce & Gabbana	Dolce_&_Gabbana
杜嘉酒店及度假村	Dolce_Hotels_&_Resorts
甜蜜的爱	Dolce_Love
Dolce Vita	Dolce_Vita
Dolcis	Dolcis
美元银行	美元银行
美元折扣	美元_折扣
美元将军	美元_将军
多拉旅馆	Dollar_Inn
美元之王	美元之王

名称	ID
Dollar 租车	美元_租车_汽车
美元节俭	美元_节俭
美元树	美元树
美元与理智	Dollars_&_Sens
Dolores Promesas	Dolores_Promesas
海豚教育版	海豚_教育
多伦多海鲜自助餐	Dolundo _海鲜_自助餐
Dom Knigi	Dom_Knigi
Dom Ksiazki	Dom_Ksiazki
Dom na Kinoto	dom_na_kinoto
Dom'Protec	Dom'Protec
Domashnyaya apteka	domashnyaya_apteka
多马斯科	多马斯科
Domayne	Domayne
Domestik	Domestik
Dominick's	Dominick's
自治领土	自治领土
多米诺骨牌	多米诺骨牌
多米诺披萨	Domino's Pizza
Domo	Domo

名称	ID
Domotekhnika	Domotekhnika
Domotel	Domotel
Domy	Domy
唐·贝利萨里奥	Don_Belisario
Don Domat	Don_Domat
Don Jediondo	Don_Jediondo
唐璜酒店	唐胡安酒店
唐·巴勃罗的墨西哥厨房	Don_pablo's_mexican_Kitchen
Don Pealo tabak-tisk	don_pealo_tabak-tisk
Don Pepe Pizzéria	Don_Pepe_Pizzéria
唐氏鞋和皮革维修	Don's_shoe_&_Leather_Repairy
Don-Don	Don-Don
DonDonDown	DonDonDown
DonMono	DonMono
道南海町信用金库	Donan_umichica_Shinkin_Bank
多纳托斯	多纳托斯
Donatto	Donatto
Dondon-an	Dondon-an
Doner Kebab	Doner_Kebab
东海堂烘焙坊	东海堂_面包店

名称	ID
董鹏	董鹏
董毅日胜	董义日生
东阿银行	东亚银行
东亚百货商店	Donga_百货大楼
东白	东白
东北敦	东北敦
东驰	东驰
Dongenergy	Dongenergy
东方百佳超市	东方_百家_超市
东风雪铁龙	东风_雪铁龙
东风风行	东风_风兴
东风本田	东风_本田
东风本田售后服务	东风_本田_售后_服务
东风本田直销店	东风_本田_直销店
东风柳汽乘龙	东风_刘启_成龙
东风标致	东风_标致
东风瓦索尔卡车	东风_瓦索尔_卡车
东风悦达起亚	东风_悦达_起亚
东港农村商业银行	东港农村商业银行
东莞农村商业银行	东莞_农村_商业银行

名称	ID
东莞证券	东莞_证券
东湖	东湖
东台农商银行	东台农村商业银行
Donguri Kyowakoku	Donguri_Kyowakoku
东乡农商银行	东乡_农村_商业银行
东兴楼	东兴楼
东阳农村商业银行	东阳农村商业银行
东营银行	东营银行
东营沙龙	东营沙龙
东园餐厅	东源_餐厅
东造鱼头	东造鱼头
东至农村商业银行	东直农村商业银行
Donhi	Donhi
Donhorbank	Donhorbank
Donner & Reuschel	Donner_&_Reuschel
Donskaya apteka	donskaya_apteka
甜甜圈连接	甜甜圈连接
甜甜圈之王	Donut_King
甜甜圈	甜甜圈
双倍浓缩咖啡	多皮奥_意式浓缩咖啡

名称	ID
零号双手	Doppio_Zero
Dor Alon	Dor_Alon
多拉	多拉
多雷维奇病理学	多雷维奇_病理学
多林特	多林特
Dormeo	Dormeo
Dormity	Dormity
多萝西·珀金斯	Dorothy_Perkins
帝盛酒店及度假村	帝盛酒店及度假村
多特蒙德大众银行	Dortmunder_Volksbank
多萨广场	dosa_Plaza
Dosenbach-Ochsner Schuhe and Sport	dosenbach-ochsner_schuhe_und_Sport
Dosteen 餐厅	多斯汀_餐厅
圆点	圆点
Dotti	Dotti
双倍披萨和鸡肉	Double_Double_Pizza_&_Chick
双蛋	双蛋
Double Kwik	Double_Kwik
Doubledave 的披萨工厂	Doubledave's_PizzaWor
逸林	逸林

名称	ID
道吉烧烤	Dougie's_Grill
道格拉斯	道格拉斯
道格拉斯和戈登房地产经纪人	Douglas_&_Gordon_Estate_Agents
道格拉斯·汉利·摩尔病理学	道格拉斯_汉利_摩尔_病理学
同济茶	Douji_Tea
Dover-Phila FCU	dover-phila_fcu
向东加州大学	down_east_cu
Downtowner 汽车旅馆	Downtowner_Motor_Inn
Dpiu	Dpiu
Ahorro 博士	Dr_Ahorro
艾琳娜·埃里斯博士	Dr_Irena_Eris
康苏尔塔博士	Dr._Consulta
马滕斯博士	Dr._Martens
Max Lékarna 博士	Dr._Max_Lekarna
帕德洛博士	Dr._Padlo
雷迪博士的实验室	Dr._reddy's 实验室
黑胶博士	Dr._Vinyl
马克斯博士	马克斯博士
Dracik	Dracik
龙酒店	Dragon_Hotel

名称	ID
Dragon-I	Dragon-I
戏剧孩子	Drama_Kids
Drapers & Damons	Drapers_&_Damons
梦之岛	梦之岛
梦幻地带	Dream_Zone
DreamMaker 浴室和厨房	DreamMaker_浴室_&_厨房
梦想家	梦想家
Dreammap	Dreammap
梦指甲和身体	Dreamnails_&_Body
梦想	梦想
Dreams-on	Dreams-on
德雷尼科夫租车	Drenikov_Rent_a_Car
德累斯顿银行	德累斯顿银行
德累斯顿大众银行 Raiffeisenbank	Dresdner_Volksbank_Raiffeisenbank
DressCode	DressCode
Dressbarn	Dressbarn
Drexel Heritage	Drexel_Heritage
Drfish	Drfish
饮品	饮品
Drinkstore	Drinkstore

名称	ID
假日开车	假日驾驶
DriveTime 二手车	DriveTime_二手车
司机之选	司机的选择
Droga Raia	Droga_Raia
Droga Rede	Droga_Rede
Drogabel	Drogabel
Drogabella	Drogabella
Drogafarma	Drogafarma
Drogal	Drogal
Drogamais	Drogamais
Drogaria Araujo	Drogaria_Araujo
Drogaria Atacadão	Drogaria_Atacadão
Drogaria Boa Saúde	Drogaria_Boa_Saude
Drogaria Catarinense	德罗加里亚_卡塔琳娜州
Drogaria Coop	Drogaria_Coop
额外的 Drogaria	Drogaria_Ex
Drogaria Onofre	Drogaria_Onofre
Drogaria Rosário	Drogaria_Rosário
Drogaria 圣保罗	Drogaria_Sao_Paulo
Drogaria Total	Drogari_Total

名称	ID
Drogaria Venancio	Drogaria_Venancio
德罗加里亚斯 Farmagnus	Drogarias_Farmagnus
Drogarias Farmelhor	Drogarias_Farmelhor
Drogarias Max	Drogarias_Max
Drogarias Ofertão	Drogarias_Ofertão
Drogarias Pacheco	Drogarias_Pacheco
Drogarias Tamoio	Drogarias_Tamoio
Drogarias 超受欢迎	Drogarias_ultra_Popul
Drogasil	Drogasil
自然 Drogeria	Drogeria_Natura
Drogueria Alemana	Drogueria_Alemana
Drogão Super	Drogão_Super
Dromedar	Dromedar
Dropa	Dropa
Dros	Dros
Drova	Drova
药品商场	Drug_Emporium
毒品 Segami	Drug_Segami
关药店	药店_Seki
毒品 Yutaka	药品_Yutaka

名称	ID
德拉蒙德高尔夫	Drummond_Golf
德拉蒙德的	德拉蒙德的
醉猴子	Drunken_Monkey
德鲁里旅馆	Drury_Inn
Druxy's	Druxy's
干燥中心	干燥中心
干洗快递	干洗_快递
干洗站	干洗站
DryWash	DryWash
Dryman	Dryman
dsquared2	dsquared2
Dt Detalles	Dt_Detalles
大田物流	DTW_Logist
杜佳	杜佳
Du pareil au méme	du_pareil_au_mème
DuPage CU	DuPage_CU
杜安·里德	Duane_Reade
迪拜水电管理局	迪拜_电力与水务局
迪拜伊斯兰银行	迪拜伊斯兰银行
迪拜市政府	迪拜市政府

名称	ID
都柏林之家	Dublin_House
杜布拉维卡	杜布拉维卡
杜卡蒂	杜卡蒂
Duchamania	Duchamania
公爵夫人	公爵夫人
Duck Thru Food	Duck_Thru_Food_Stores
鸭王	鸭王
美国管道医生	duct_doctor_USA
Dufy	Dufy
杜克	杜克
杜克服装	公爵服装
Dulce Luna	Dulce_Luna
德卢斯贸易公司	Duluth_Trading_Co
多乐士	多乐士
多乐士油漆中心	多乐士油漆中心
Dulzia	Dulzia
饺子老虎	Dumpling_Tiger
杜纳斯酒店及度假村	杜纳斯酒店及度假村
邓肯的	邓肯的
沙丘	沙丘

名称	ID
Dunelm Mill	Dunelm_Mill
邓弗姆林建筑协会	邓弗姆林_建筑_协会
邓纳姆的体育	Dunham's Sports
登喜路	登喜路
敦化农商银行	敦化_农村_商业银行
Dunkin'	Dunkin'
Dunkin'Donuts	Dunkin'_Donuts
邓禄普	邓禄普
邓恩兄弟咖啡	邓恩兄弟咖啡
Dunn's Famous	Dunn's_Famous
邓恩-爱德华兹油漆	邓恩-爱德华兹_Paints
Dunnes Stores	Dunnes_Stores
邓恩斯	邓恩斯
杜诺苏萨	杜诺苏萨
Dupaco 社区 CU	dupaco_community_cu
杜邦	杜邦
Duraclean	Duraclean
德贝尔药房	德贝尔_药房
榴莲	榴莲
榴莲糖果酒店	榴莲糖果酒店

名称	ID
榴莲之星酒店	榴莲之星酒店
Durlinger	Durlinger
Duron 涂料和墙布	Duron_Paints_&_Wallcoverings
Dursty	Dursty
都市 118 静轩	dushi_118_Jingxuan
荷兰兄弟咖啡	荷兰兄弟咖啡
Dve Palochki	Dve_Palochki
Dvizhenie	Dvizhenie
Dxl	Dxl
Dyadya Denyor	Dyadya_Denyor
dyat'kovo	dyat'kovo
Dymocks	Dymocks
Dyna Jaya	Dyna_Jaya
Dynabyte	Dynabyte
王朝 Tiletop	Dynasty_Tiletop
Dyneff	Dyneff
Dyno	Dyno
dyoshevo	dyoshevo
Dyrego Zoo	Dyrego_Zoo
戴森	戴森

名称	ID
Dzherelo	Dzherelo
天珠银行	Dzi_Bank
Dzintars	Dzintars
DECECERAM	DECECERAM
DEMA	DEMA
Dackia	Dackia
十项全能	十项全能
Deltakarek	Del_Takarek
Del-Zalai Egyesült Takarekszöv。	del-Zalai_Egyesult_Takarekszöv。
德利弗兰斯	德利弗兰斯
Depil Tech	depil_Tech
Döbbe Backerei	Döbbe_Báckerei
鄂梅	E_Mei
马上就好	E_PRONTO
鄂章门	E_张门
电子银行	电子银行
电子积木教育	电子积木_教育
E-KAFFI	E-KAFFI
电子市场交易者	电子智能交易者
E-Markt	E-Markt

名称	ID
电子光学	电子光学
电子贸易	电子贸易
电子装	电子装
E-Z 8 汽车旅馆	E-Z_8_Motel
E-Z Mart	E-Z_Mart
E-Z Serve	E-Z_Serve
E-Z 商店	E-Z_Shop
E-Z 停止	E-Z_Stop
e-laad.nl	e-laad.nl
意大利电动汽车	意大利电动汽车
E. LECLERC	E._LECLERC
E. 勒克莱尔快车	e._leclerc_Express
E. 太阳银行	E._Sun_Bank
E. Wedel	E._Wedel
E.C.A	E.C.A
E.Leclerc	E.Leclerc
E.Lecler DRIVE	e.leclerc_Drive
E.ON	E.ON
E.SUN	E.SUN
e.nopi	e.nopi

名称	ID
E1	E1
E1 家居专卖店	e1_home_Specialty_store
e2Matrix	e2Matrix
EA Anyw	ea_Anywh
轻松购物	EASY_SHOP
EATALY	EATALY
伊顿	伊顿
EB Connect	eb_connect
EB Games	EB_Games
EBH	EBH
EBN	EBN
EBO	EBO
EBS	EBS
ECE	ECE
ECO	ECO
生态服务	生态服务
ECOFFEE	ECOFFEE
ECONSAVE	ECONSAVE
ECOS	ECOS
ECS	ECS

名称	ID
欧洲货币单位立方英镑	ECU_CU
ED	ED
EDEKABANK	EDEKABANK
伊甸城酒店	伊甸城酒店
EDF	EDF
EDIYA 咖啡	EDIA_COFFEE
EDM	EDM
EDMC	EDMC
EDP Comercial	edp_Comercial
EDUSCHO	EDUSCHO
埃德温	埃德温
EE	EE
EFCU 金融 FCU	efcu_financial_fcu
EFE	EFE
EFFIA	EFFIA
EFFIA EV	EFFIA_EV
EFOR	EFOR
EFU	EFU
EG	EG
EGESOK	EGESOK

名称	ID
EGV	EGV
埃及阿拉伯土地银行	埃及_阿拉伯_土地_银行
EH Hoteles	EH_Hoteles
EIBank	EIBank
EIX 酒店	Eix_Hotels
EKO	EKO
EXPRESS-VOLGA	EXPRESS-VOLGA
EL	EL
EL ARBOL	EL_ARBOL
EL CLUB DE LA MILANESA	EL_CLUB_DE_LA_MILANESA
EL GLOBO	EL_GLOBO
EL NOBLE	EL_NOBLE
EL TORITO SINALOENSE	EL_TORITO_SINALOENSE
EL TUNEL	EL_TUNEL
EL-SALG	EL-SALG
ELA	ELA
伊兰旅馆	ELAN_INN
ELC	ELC
ELDOM	ELDOM
电动快车	电气快车

名称	ID
电子仓库	电子仓库
电子世界	电子世界
电子网	电子网
优雅	优雅
ELEKTROSILA	ELEKTROSILA
大象之王干洗	ELEPHANT_KING_FOR_DRY_CLEAN
ELISE	ELISE
ELIT	ELIT
精英	精英
瑞典精英酒店	瑞典精英酒店
ELIXIA	ELIXIA
ELIZONDO	ELIZONDO
ELKJÖP	ELKJÖP
ELLE	ELLE
ELLUS	ELLUS
埃隆	埃隆
ELTODO	ELTODO
ELplus	ELplus
EM Home	em_home
EM-TÉ	EM-TÉ

名称	ID
EMARAT MISR	EMARAT_MISR
EMILBANCA	EMILBANCA
阿联酋伊斯兰银行	阿联酋伊斯兰银行
阿联酋航空黎巴嫩银行	阿联酋航空_黎巴嫩_银行
EMPADA BRASIL	EMPADA_BRASIL
世纪金源大饭店及度假村	世纪金源公园_酒店及度假村
EMPEREUR	EMPEREUR
皇帝钻石	EMPEROR_DIMOND
英皇珠宝	EMPEROR_珠宝
英皇钟表和珠宝	EMPEROR 手表和珠宝
EMPORIO ARMANI	EMPORIO_ARMANI
EMSPO	EMSPO
ENAPO	ENAPO
ENERGOPETRO	ENERGOPETRO
ENERPETROLI	ENERPETROLI
英文主页	英语_HOME
ENOC	ENOC
ENVI	ENVI
ENZED	ENZED
ENZO	ENZO

名称	ID
EP	EP
EP: 电子合作伙伴	EP: 电子_合作伙伴
EPA	EPA
EPASSA	EPASSA
EPENGLE	EPENGLE
EPHARMA	EPHARMA
EPIC 药房	Epic_Pharmacies
缩影	缩影
EPOQUE 酒店	EPOQUE_酒店
EPPCO	EPPCO
等价物	等价物
EQUUS	EQUUS
时代	时代
新时代	ERA_NOVA
ERA 房地产	时代_房地产
呃	呃
埃里克·邦帕德	ERIC_BOMPARD
ERICEIRA SURF & SKATE	ERICEIRA_SURF_&_SKATE
ERICEIRA 冲浪店	ERICEIRA_SURF_SHOP
伊利保险	erie_Insurance

名称	ID
ERMITAZAS	ERMITAZAS
欧内斯特·琼斯	ERNEST_JONES
EROSKI	EROSKI
EROSKI 中心	eroski_Center
EROSKI viajes	eroski_Viajes
ES 电子销售公司	es_Electro
ESACDA	ESACDA
ESAF 小型金融银行	esaf_small_Finance_Bank
ESAY JOY	ESAY_JOY
ESB	ESB
ESCADA	ESCADA
ESCLAT	ESCLAT
ESCLAT OIL	ESCLAT_OIL
ESKAN 银行	ESKAN_BANK
ESL 快递	ESL_Express
ESOTIQ	ESOTIQ
ESP	ESP
埃斯波森德	埃斯波森德
浓缩咖啡屋	ESPRESSO_HOUSE
精神	精神

名称	ID
ESSELUNGA	ESSELUNGA
ESSEN	ESSEN
必不可少的	必不可少的
ESTELAR	ESTELAR
ETA	ETA
ETAPE AUTO	ETAPE_AUTO
ETEKA	ETEKA
永恒的	永恒的
道德观	道德观
埃特纳	埃特纳
ETOLA	ETOLA
ETRO	ETRO
ETSAN	ETSAN
EURO APTEKA	EURO_APTEKA
EURO REPAR 汽车服务	Euro_repar_car_servic
EUROBIT	EUROBIT
EUROCASH	EUROCASH
EUROINS	EUROINS
EUROKIOSK	EUROKIOSK
欧洲大师	欧洲大师

名称	ID
EUROMEDICA	EUROMEDICA
EUROMOBIL	EUROMOBIL
EUROMODA	EUROMODA
EURONICS	EURONICS
欧洲银行	欧洲银行
EUROPAER	EUROPAER
EUROPAM	EUROPAM
欧洲网球公开赛	欧洲网球公开赛
欧洲综合大楼	欧洲综合大楼
EUROSCOOP	EUROSCOOP
EUROSKO	EUROSKO
EUROSPAR	EUROSPAR
欧洲之星酒店	欧洲之星酒店
欧洲站	欧洲站
EUROSpin	EUROSpin
EUROTALLER	EUROTALLER
EUROVAISTINE	EUROVAISTINE
EV Connect	ev_Connec
EVBox	EVBox
EVCARD	EVCARD

名称	ID
光大证券	光大证券
常青洗衣	常青洗衣房
不断增长的银行	永恒增长银行
EVF	EVF
EVO Banco	evo_banco
EVSE 有限责任公司 WebNet	EVSE_LLC_ WebNet
EValu9	EValu9
EVgo	EVgo
EVnet荷兰	EVnet荷兰
EWOPOINT	EWOPOINT
EWIVA	EWIVA
前石油	前石油
EXALCO	EXALCO
很棒的唐宫	极好_TANG_PALACE
ELCELSIOR	ELCELSIOR
进出口银行	EXIM_BANK
泰国进出口	EXIM_TAILAND
进出口之旅	Exim_Tours
EXIsport	EXIsport
专家级精英	专家级精英

名称	ID
快递货物	快递货运
快捷玻璃	快捷玻璃
EXTROB	EXTROB
EXTREM3	EXTREM3
EZ Go	ez_go
EZ Lube	ez_Lube
EZ MONEY 典当	ez_money_Pawn
EZ Mart	ez_mart
EZ.KON	EZ.KON
EZMONEY 贷款服务	ezmoney_Loan_Servic
EZPAWN	EZPAWN
EZZA 集团	EZZA_GROUP
鹰	鹰
老鹰男孩披萨	老鹰男孩_披萨
老鹰房屋抵押贷款	鹰牌房屋抵押贷款
路易斯安那鹰队 FCU	eagle_Louisiana_FCU
老鹰工作室	老鹰工作室
老鹰	老鹰
Eaka 365 酒店	Eaka_Hotel_365
Earl Scheib 油漆和车身	Earl_Scheib_Paint_&_Body

名称	ID
伯爵	伯爵
早期学习中心	早期学习中心
地球票价	地球票价
工程机械师 CU	Earthmover_CU
宜居酒店	Ease_Hotel
Easehome	Easehome
东部	东部
东海岸	东海岸
东方钓具	East Fishing_Tackle
东海海鲜餐厅	东海_海鲜_餐厅
东区 Mario's	东区_马里奥的
东玉木医疗保健	东方_塔玛基_医疗保健
华美银行	东方西岸
东西方民族美食	东西方_民族_食物
芝加哥以东	芝加哥以东
EastWest 银行	EastWest_银行
Eastday Houkal	Eastday_Houkal
东部	东部
东方银行	东方银行
东方山地运动	东部_山地_体育

名称	ID
东方先锋	Eastern_Pioneer
东方地产	Eastern_Realty
东方茶	Eastern_Tea
Eastex	Eastex
Eastmill FCU	eastmill_FCU
Eastravel	Eastravel
东西方联合银行	东西方_Unibank
Easy by RHB	easy_by_rhb
简易车	Easy_Car
轻松的汽车生活	Easy_Car_Life
Easy Inn	Easy_Inn
随心所欲	Easy_Spiri
EasyFast	EasyFast
随心所欲	随心所欲
居然之家	居然之家
easypaisa	easypaisa
吃	吃
Eat'n Park	Eat'n_Park
EatZi的	EatZi的
伊顿酒店	Eaton_Hotel

名称	ID
Eatzzaa 披萨	Eatzzaa_Pizz
Ebar	Ebar
Ebby Halliday 房地产	Ebby_Halliday_Real_Estate
埃伯哈德制造	Eberhard_制造业
斯图加特电动自行车站	斯图加特电动自行车站
Ebohr	Ebohr
Ebs 建筑协会	Ebs_建筑学会
Ecco	Ecco
Ecco 假日	ecco_Holiday
Ecf	Ecf
Echarm	Echarm
宜尚Plus酒店	Echarm_Plus_Hotel
鄂城	鄂城
回声英语	回声_英语
Eci	Eci
埃克德快车照片	Eckerd_Express_Ph
埃肯福德银行	Eckernförder_Bank
Eckersley's	Eckersley's
环保自动柜员机	eco_atm
生态市场	生态市场

名称	ID
环保赛车	Eco_Racing_Car
环保租车	Eco_Rent_A_Car
生态商店	生态商店
环保标志	环保标志
EcoChoice 臭虫控制	EcoChoice_Bed_Bug_Control
EcoWater 系统	EcoWater_系统
Ecobank	Ecobank
Ecogreen 汽车	Ecogreen_Car
Ecolaundry	Ecolaundry
Ecoline 清洗	Ecoline_Wash
Econet	Econet
Econet 无线	econet_Wireless
伊克诺旅馆	Econo_Lodge
经济食品	经济食品
Econogas	Econogas
经济型汽车租赁	经济型_租车
经济旅馆	经济旅馆
经济型住宿系统	经济型住宿系统
Econotax	Econotax
Ecowash	Ecowash

名称	ID
埃德·哈里	Ed_Harry
Ed's Easy Diner	ed's_easy_diner
Eda 牙科	eda_Dental
Edah	Edah
Eddie Bauer	Eddie_Bauer
埃迪火箭队	埃迪_火箭队
Eddie V's	Eddie_V's
Edeka	Edeka
Edel'veys	Edel'veys
雪绒花	雪绒花
伊甸园	伊甸园
伊甸园	Eden_Park
Eden 鞋	Eden_Shoes
伊甸村	Eden_Village
Edenrobe	Edenrobe
埃德加斯	埃德加斯
Edgars 活跃	edgars_active
Edhi	Edhi
可食用的布置	可食用的安排
Edina Realty	Edina_Realty

名称	ID
爱丁堡毛纺厂	爱丁堡毛纺厂
版本	版本
日本的江户	日本江户时代
Edoardos	Edoardos
江户前寿司	江户前_寿司
Eds	Eds
教育第一 CU	教育第一_cu
教育社区 CU	教育_社区_cu
教育用品商	教育_Outfitters
edukids	edukids
爱德华琼斯投资	Edward_Jones _投资
埃德温·沃茨高尔夫	Edwin_Watts_Golf
Eegee's	Eegee's
Efe Tur	Efe_Tur
Effu	Effu
Efloussy	Efloussy
Eg3	Eg3
Ege Seramik	Ege_Seramik
Egepen	Egepen
鸡蛋和东西	Eggs_'n_things

名称	ID
Eggs Up G	Eggs_Up_Gril
聪明的	聪明的
Eggsuis	Eggsuis
Eglin FCU	eglin_fcu
利己主义者	利己主义者
Egon	Egon
Egreen	Egreen
埃及海湾银行	埃及海湾银行
爱媛信用金库	爱媛信用金库
Ehinger 大众银行	Ehinger_Volksbank
eiDer	eiDer
Eic 教育	Eic_教育
Eic 教育培训	Eic_教育_训练
Eicher	Eicher
艾格峰	艾格峰
艾琳·费舍尔	Eileen_Fisher
爱因斯坦兄弟	爱因斯坦兄弟
爱因斯坦咖啡	Einstein_Kaffee
Eka	Eka
叶卡捷琳堡国家城市银行	叶卡捷琳堡国家城市银行

名称	ID
Eki	Eki
Ekipashop	Ekipashop
生态市场	生态市场
Ekol	Ekol
Ekolas	Ekolas
Ekomar	Ekomar
Ekomini	Ekomini
Ekonika	Ekonika
Ekono	Ekono
Ekonom	Ekonom
生态广场	生态广场
Expert	Expert
Ekspobank	Ekspobank
Expres-Bank	Expres-Bank
Ekyog	Ekyog
El Ahorro 超市	El_Ahorro_Supermark
El Buen Sabor	El_Buen_Sabor
El Caballo	El_Caballo
El Chico	El_Chico
El Corral	El_Corral

名称	ID
El Corralito	El_Corralito
El Corte Inglés	El_Corte_Inglés
El Costeñito	El_Costeñito
El Dedal de la Tata	el_dedal_de_la_Tata
El Domyaty	El_Domyaty
El Ezaby	El_Ezaby
El Farolito	El_Farolito
El Fenix	El_Fenix
El Feshawy	El_Feshawy
El Fornet d'en Rossen	el_fornet_d'en_rossend
El Gallito	El_Gallito
El Ganso	El_Ganso
El Gato Pisao	El_Gato_Pisao
El Gigante del Colchon	el_gigante_del_Colchon
El Guardabosques	El_Guardabosques
El Jardín de Teca	el_Jardin_de_teca
艾尔莱西药房	el_Leithy_Pharmacies
El Minero	El_Minero
El Mokhtabar	El_Mokhtabar
El Molí Pan y Café	el_molí_pan_y_Café

名称	ID
El Molí Vell	El_Molí_Vell
艾尔蒙特	El_Monte
艾尔蒙特房车	el_monte_rv
El Mundo de 3 Pesos	el_mundo_de_3_pesos
El Mundo del Pintor	el_mundo_del_pintor
El Neel	El_Neel
El Nilein Bank	El_Nilein_Bank
El Nopal	El_Nopal
El Palacio de Hierro	el_palacio_de_hierro
El Papalote	El_Papalote
艾尔帕蒂奥	El_Patio
El Pez Gastrotiend	El_Pez_Gastrotiendas
El Pollo Loco	El_Pollo_Loco
El Pollo Pepe	El_Pollo_Pepe
El Porton	El_Porton
El Punto Magico	El_Punto_Magico
阿根廷牧场 Asador	El_rancho_Asador_Argentino
El Ribot Blau	El_Ribot_Blau
El Rincon de Guijuelo	el_rincon_de_guijuelo
El Rincon de María	el_rincon_de_María

名称	ID
埃尔里奥布兰科	El_Rio_Blanco
El Shabrawy	El_Shabrawy
El Sharkawy	El_Sharkawy
El Shorouq	El_Shorouq
艾尔超级	El_Super
El Taco 手提包	El_Taco_Tote
El Tawhid W El Nour	El_tawhid_W_El_Nour
El Torito	El_Torito
El Toro 墨西哥餐厅。& Cantina	El_Toro_Mexican_Rest。 _&_Cantina
El Vaquero	El_Vaquero
El Zaeim	El_Zaeim
El que no corre vuela	el_que_no_corre_Vuela
El'dorado	El'dorado
Elan	Elan
伊兰英语	ELAN_English
Eldan	Eldan
Eldi	Eldi
埃尔多拉多	埃尔多拉多
EleRent	EleRent
伊莱克特拉酒店及度假村	Electra_酒店及度假村

名称	ID
Electrabel	Electrabel
电动	电动
电气公司	Electric _Co
让美国电气化	电动_美国
加拿大电气化	Electrify_加拿大
电动驱动	电动驱动
伊莱克斯	伊莱克斯
伊莱克斯主页	伊莱克斯_主页
电子解决方案	电子解决方案
由迪克·史密斯提供动力的电子设备	Electronics_powered_by_dick_Smith
电子行星	电子行星
Electrotherm 印度有限公司	电热_印度_有限公司
Elefante Azul	Elefante_Azul
Eleftheroudakis	Eleftheroudakis
优雅光学	优雅_Optik
优雅	优雅
Elektro PROT	elektro_Proton
Elektromotive	Elektromotive
Elemag	Elemag
元素	元素

名称	ID
元素 FCU	Element_fcu
元素金融 FCU	Elements_Financial_FCU
元素按摩	元素_按摩
大象酒吧	大象酒吧
大象与城堡	大象与城堡
埃莱特罗·马特乌斯	Eletro_Mateus
Eletro 购物	电子购物
Eletroshow Gbarbosa	Eletroshow_Gbarbosa
Eletrosom	Eletrosom
海拔汉堡	海拔_Burger
海拔汉堡餐厅	海拔_汉堡_餐厅
Elexia 健身	Elexia_健身
精灵	精灵
Elf Antar	Elf_Antar
Elgiganten	Elgiganten
Eli 精品店	Eli_Bout
埃利安托	埃利安托
Eliel Figueirédo	Eliel_Figueirédo
Elife 购物中心	elife_mall
Elin	Elin

名称	ID
Elis	Elis
Elisa Shopit	Elisa_Shopit
精英租车	Elite_Car _Rental
精英咖啡	精英咖啡
精英照片	Elite_Foto
精英学习	精英学习
Elixian 科技	Elixian_科技
伊丽莎白	伊丽莎白
伊丽莎白雅顿红门水疗中心	Elizabeth_Arden_Red_Door_Spas
麋鹿旅馆	Elks_Lodge
艾拉·巴奇	Ella_Bache
艾拉·鲁格	Ella_Rouge
Elledí	Elledí
Ellena	Ellena
埃勒·蒙坦	Eller_Montan
Ellerines	Ellerines
埃利亚诺斯咖啡公司	Ellianos 咖啡公司
埃利尼卡市场	Ellinika_Market
Elliotts 的非百老汇熟食店	Elliotts'_Off_Broadway_Deli
埃利普斯银行	Ellips_bank

名称	ID
艾尔玛	艾尔玛
Elmer's	Elmer's
Elmo	Elmo
Elna	Elna
艾尔普莱斯	艾尔普莱斯
Elvi	Elvi
Elvi 杂货店	elvi_Grocery_store
爱丽舍西部酒店	Elysee_West_Hotel
极乐世界医疗保健	Elysium_医疗保健
Elyza's	Elyza's
Em	Em
伊曼纽尔·伯格	Emanuel_Berg
Emarat	Emarat
Embarcadeiro 41	Embarcadeiro_41
大使馆套房	大使馆套房
徽章	徽章
Emelya	Emelya
翡翠城冰沙	翡翠城_冰沙
名誉退休老年人生活	Emeritus_Senior_Living
Emery FCU	emery_fcu

名称	ID
移民储蓄银行	移民储蓄银行
埃米利奥·尤斯特	Emilio_Yuste
Emirates	Emirates
阿联酋航空驾驶学院	阿联酋航空驾驶学院
阿联酋航空印度国际交易所	阿联酋航空_印度_国际交易所
阿联酋航空NBD	阿联酋航空_NBD
阿联酋邮政	阿联酋航空_邮政
Emme Piu	Emme_Piu
Emo	Emo
情感	情感
Empadinhas Barnabé	Empadinhas_Barnabé
重点珠宝	重点_珠宝
帝国	帝国
哈萨克斯坦帝国	帝国_哈萨克斯坦
帝国餐厅	帝国_餐厅
Emporiki 银行	Emporiki_Bank
安普里奥	安普里奥
赋能 CU	赋权_cu
企业银行	emprise_Bank
Empório 车身商店	empório_body_Store

名称	ID
Emro	Emro
Emsi	Emsi
Emslándische Volksbank	Emslándische_Volksbank
Emyco	Emyco
在路上	En_Route
En-Suey	En-Suey
enBW	enBW
EnTie 商业银行	EnTie_商业银行
Enbee	Enbee
Enbright	enbright_cu
Encompass	包含_生命值
Enderpen	Enderpen
恩德萨	恩德萨
Endesa X	Endesa_X
Eneco	Eneco
Eneco EV	eneco_ev
Enel X	Enel_X
Eneos	Eneos
Ener-Gi	Ener-Gi
EnerGas	EnerGas

名称	ID
EnergoMashBank	EnergoMashBank
Energobank	Energobank
能量厨房	能量_厨房
能量动力	Energy_Power
Enerhobank	Enerhobank
远轻信用金库	Engaru_Shinkin_Bank
恩根	恩根
英语优先	英语_First
英语第一教育	英语_第一_教育
增强	增强
Eniseyskiy Ob'edinyonniy 银行	eniseyskiy_ob'edinyonniy_Bank
Enjaz 银行服务	Enjaz_银行_服务
享受	享受
慢慢享受鳗鱼的乐趣	慢慢享受 eels_the_els_
Enklere Liv	Enklere_Liv
巨大的	巨大的
Enoteca	Enoteca
Enotel	Enotel
丰富	丰富
恩里科·马里内利	Enrico_Marinelli

名称	ID
远州信用金库	Enshu_Shinkin_Bank
Entel	Entel
企业	企业
企业银行	企业银行
企业汽车销售	企业_汽车_销售
企业商用卡车	企业_商用_卡车
Enterfarma	Enterfarma
环境服务专业人员	环境服务_专业人员
Evision CU	Envision_cu
羡慕	羡慕
Enza 主页	Enza_Home
Enztalbank	Enztalbank
Biosupermarkt Eo	eo_der_biosupermarkt
Epanouir	Epanouir
Epcot	Epcot
史诗之翼	Epic_Wings
Eplehuset	Eplehuset
爱普生	爱普生
Epyon	Epyon
Eqodom	Eqodom

名称	ID
Equa bank	equa_ban
厄瓜多尔	厄瓜多尔
赤道	赤道
赤道商业银行	赤道商业银行
Equinor	Equinor
春分	春分
装备	装备
等同的	等同的
Equishare CU	equishare_cu
公平小额信贷	Equitas_Micro_Finance
股票银行	Equity_Bank
股权公司住房	股权_公司_住房
Equity	Equity_One
Erbert & Gerbert's	Erbert_&_Gerbert's
Erbi	Erbi
Erciyes Börek	Erciyes_Börek
鄂尔多斯	鄂尔多斯
鄂尔多斯农村商业银行	Erdos_Rural_商业银行
Erfurter Bank	Erfurter_Bank
伊利社区 FCU	erie_community_fcu

名称	ID
Erie FCU	erie_fcu
Erik's Delicafe	Erik's_delicafe
呃	呃
Ermapub	Ermapub
Ermenegildo Zegna	Ermenegildo_Zegna
Ernie 的加油站	Ernie's fuel_stop
安永会计师事务所	安永会计师事务所
恩斯汀的家人	恩斯汀的家族
Erntebrot	Erntebrot
Ero Car 保加利亚有限公司	Ero_Car_Bulgaria_Ltd。
埃罗斯基城	Eroski_City
情色之城	Erotic_City
Erpen	Erpen
埃罗卡	埃罗卡
Erste Bank	Erste_Bank
Erzgebirgsparkasse	Erzgebirgsparkasse
Erätukku	Erätukku
Es Teler 77	es_Teler_77
逃生旅行	逃生_旅行
伴游	伴游

名称	ID
埃森彭	埃森彭
埃斯卡	埃斯卡
埃斯库拉普	埃斯库拉普
埃斯马尔诺娃	埃斯马尔诺娃
激光太空	Espaço_Laser
Espaço Arabe	Espaço_Arabe
EspañaDuero	EspañaDuero
Espetinhos mimi	espetinhos_mimi
电子竞技	电子竞技
Esquire 沉积解决方案	Esquire_存款_解决方案
绅士厨房	《时尚先生》_厨房
Esquires 咖啡屋	Esquires_Coffee_H
《时尚先生》咖啡屋	Esquires_咖啡屋
Essar	Essar
随笔石油	essay_Petrolem
安信证券	精华_证券
必备美容	必备_美容
埃森扎	埃森扎
EssiLor	EssiLor
Essmann 的 Backstube	essmann's backstube

名称	ID
埃索	埃索
Esso Autocare	Esso_Autocare
埃索快车	埃索快车
Esso&Feoso	Esso&Feoso
Estapar	Estapar
Estel'Adoni	Estel'_Adoni
美容中心	美容中心
Estilos	Estilos
埃斯托里尔图书	Estoril_Books
河口	河口
Etalon	Etalon
Etam	Etam
Etam 内衣	Etam_Lingery
Etap 酒店	etap_Hotel
Etapes Hoteliers Corses	Etapes_Hoteliers_Corses
伊桑·艾伦	Ethan_Alle
Ethel Austin	Ethel_Austin
Etiler Marmaris	Etiler_Marmaris
Etisalat	Etisalat
埃特纳小吃店	Etna_Snack_Mart

名称	ID
Etonkids	Etonkids
Etos	Etos
Ets之旅	Ets_Tur
爱丽小屋	Etude_House
Etxart & Panno	Etxart_&_Panno
余仁生	Eu_Yan_Sang
EuYanSang	EuYanSang
Eukanuba	Eukanuba
欧亚银行	欧亚银行
尤里卡鞋	Eureka_Shoes
Eurest	Eurest
欧元	欧元
6000 欧元	Euro_6000
欧洲停车场	Euro_Car_Parks
欧洲儿童	Euro_Kids
欧洲太阳队	Euro_Sol
欧元+	欧元+
欧洲-保加利亚文化中心	欧洲-保加利亚文化中心
EuroBic	EuroBic
EuroGsm	EuroGsm

名称	ID
EuroMarket	EuroMarket
欧洲控股	欧洲控股
EuroPark	EuroPark
Euroaptieka	Euroaptieka
欧洲银行	欧洲银行
欧洲银行盈丰银行	Eurobank_EFG
EurocityBank	EurocityBank
欧洲封面	欧洲封面
Eurodif	Eurodif
欧元美元	欧元美元
Eurofirany	Eurofirany
欧洲乐园	欧洲乐园
Euromar 酒店	Euromar_酒店
Euronet	Euronet
欧洲石油公司	欧洲石油公司
欧罗巴	欧罗巴
欧罗巴艺术鞋	Europa_Art_Shoes
欧洲汽车公司	欧洲汽车公司
Europcar 互联网	Europcar_Interrent
欧洲中央银行	欧洲_中央银行

名称	ID
欧洲蜡中心	European_Wax_Center
欧洲制药公司	欧洲制药公司
Europiu	Europiu
欧洲大奖	欧洲大奖
Eurorent	Eurorent
欧洲索尔	欧洲索尔
欧洲之星霍图萨	欧洲之星_Hotusa
欧洲轮胎	欧洲轮胎
Euzkadi	Euzkadi
Eva	Eva
伊娃·阿尔法罗	Eva_Alfaro
Evangelische Creditgenossenschaf	Evangelische_Kreditgenossenschaft
埃文斯	埃文斯
埃文斯维尔教师 FCU	Evansville_teachers_FCU
Evco House of H	EVCO_House_of_hose
Even	Even
埃文尼亚酒店	埃文尼亚_酒店
活动影院	活动_电影院
事件	事件
永远闪耀	永远闪耀

名称	ID
EverSmile	EverSmile
永远最好	永远最好
Evercare 临终关怀和姑息治疗	Evercare_Hospice_&_Palliative_Care
埃弗伦斯中央大学	everence_cu
Everence FCU	everence_fcu
珠峰	珠峰
珠穆朗玛峰银行	珠穆朗玛峰银行
恒大电影院	恒大影院
恒大城	恒大城
恒大广场	恒大广场
长荣银行集团	长荣银行集团
Evergreen	Evergreen_cu
长荣信用合作社	长荣信用合作社
永恒	永恒
EveryGo	EveryGo
日常链	Everyday_Cha
日常购物	Evyday_Shop
Everyones FCU	Everyones_FCU
除水之外的一切	除水之外的一切
Everywine	Everywine_S

名称	ID
Evie	Evie
Evisu	Evisu
Evkur	Evkur
Evok	Evok
Evraziya	Evraziya
Evrodent	Evrodent
Evroil	Evroil
Evroopt	Evroopt
Evropa	Evropa
Evropeyskiy Trastoviy Ban	Evropeyskiy_trastoviy_Ban
埃夫罗罗斯	埃夫罗罗斯
Evroset'	Evroset'
Ewash	Ewash
Ewenki 包商农村银行	鄂温基_包商_农村银行
Ex Libris	Ex_Libris
Exame Imagem & Laboratorio	exame_imagem_e_laboratório
Excelsior Gama Supermercad	Excelsior_Gama_Supermercado
Exchange	Exchange
Excite CU	Excite_cu
独家图书	独家图书

名称	ID
独家变更	独家_变更
独家酒店	独家_酒店
英国独家酒店	独家_酒店_英国
行政旅馆	Executive_Inn
行政人员Tans	Executive_Tans
Exel	Exel
Exel Inn	Exel_Inn
Exel 商店	Exel_Shop
Exlent ELECTRO	Exellent_Electro
排气专业人士	Exaust_Pros
埃克西德生活	Exide_Life
EximBank	EximBank
存在	存在
exist.ru	exist.ru
退出	退出
Exito 快车	Exito_Expr
Exki	Exki
Exor石油公司	Exor_Petrolem
探险	探险
Exper	Exper

名称	ID
Experim	Experim
ExpertTire	ExpertTire
Expete	Expete
世博酒店及度假村	世博会_酒店及度假村
埃及出口开发银行	埃及出口_开发_银行
Express	Express
Express	Express
快递就业专业人士	Express_就业_专业人士
快速兑换	Express_
快捷时尚	快递_时尚
快递 Kuchnia Marche	Express_Kuchnia_March
快车道	快车道
快速润滑油	极速润滑油
快捷超市	极速超市
快速换油	Express_Oil_Ch
快速停车	快速停止
快递商店	快递门店
快递联盟	Express_
Family Inn 快递	Express_by_family_
额外增强	Exta_Plus

名称	ID
美国长住酒店	延长_逗留_美国
长住豪华酒店	长住_豪华_酒店
额外	额外
额外食物	额外食物
额外群组模型	Extra_Grupo_Modelo
额外玩具	Extra_Leker
额外空间存储	额外空间_存储空间
额外超级市场	Extra_Supermercad
ExtraMile	ExtraMile
额外法尔玛	额外法尔玛
极品皮塔饼	Extreme_Pita
极致披萨	极致_披萨
埃克森	埃克森
视力水平学习中心	视力水平学习中心
Eyecare Plus	Eyecare_Plus
眼镜世界	眼镜_世界
Eyelit	Eyelit
Eyemart 快车	Eyemart_Exp
眼睛	眼睛
Eyeshow	Eyeshow

名称	ID
Ezucooo	Ezucooo
F 租一辆车	F_Rent_A_Car
F&F	F&F
F 方块	F 方块
F. Leitner	F._Leitner
F. O. S	F._O._S
F.P.Journe	F.P.Journe
F.T.S	F.T.S
F.Y.OLD TREE 咖啡馆	F.Y.OLD_TREE_CAFE
F24	F24
F5	F5
FAA FCU	FAA_FCU
FAB	FAB
FABIO SALSA	FABIO_SALSA
FABUS	FABUS
面对面的面条屋	FACE_TO_FACE_NOODLES_HOUSE
FAMILYA	FAMILYA
芳园客栈	方圆客栈
粉丝	粉丝
FANTAWILD	FANTAWILD

名称	ID
远东百货公司	远东百货商店
农场	农场
FARM Burguers	Farm_Burguers
法玛西亚·莫雷洛斯	FARMACIA MORELOS
圣特雷莎药房	FARMACIA_SANTA_TERESA
FARMACIAS DE DESCUENTO UNION	FARMACIAS_DE_DESCUENTO_UNION
FARMAPLUS	FARMAPLUS
农民超市	农民超市
FARMÁCIA SANT'ANA	FARMÁCIA_SANT'ANA
FAST AÇAI	FAST_AÇAI
快速贴合	FAST_FIT
快速健身房 24	FAST_GYM_24
快速购物	快速购物
快速修复珠宝和手表维修	快速修复珠宝和手表维修
快箱	快箱
快速标志	快速标志
父亲和儿子	父亲和儿子
一汽	一汽
一汽奔腾	Faw_besturn
一汽红旗	一汽_红旗

名称	ID
一汽丰田	一汽丰田
FAWORI	FAWORI
FAYAQUE	FAYAQUE
FBC 银行	FBC_Bank
FBZ	FBZ
FDH 银行	FDH_Bank
联邦存款保险公司	联邦存款保险公司
FDS FUERA DE SERIES	FDS_FUERA_DE_SERIE
FE 交易	FE 交易
FEBO	FEBO
感觉	感觉
感觉周期	感觉周期
腾飞宇翔	FEI_TENG_YU_XIANG
FENEBERG	FENEBERG
芬恩·赖特·曼森	FENN_WRIGHT_MANSON
FEROBRAKE	FEROBRAKE
费里奥尼	费里奥尼
FESTYTORTAS	FESTYTORTAS
FET 内衣	fet_内衣
FETnet	FETnet

名称	ID
FFC	FFC
FH 教育	FH_教育
FHB 银行	FHB_Bank
菲亚特	菲亚特
FIDELIDADE	FIDELIDADE
FIDEURAM	FIDEURAM
嘉年华	嘉年华
FIGENPA	FIGENPA
FIITJEE	FIITJEE
电影城	电影城
菲洛和朋友们	FILOU_&_FRIENDS
FILPEN	FILPEN
FINCA	FINCA
发现	发现
金融国内银行	FINDOMESTIC_BANCA
精美的牛肉餐厅	上等牛肉餐厅
FINECO	FINECO
芬兰酒店	芬兰_酒店
FINN FLARE	FINN_FLARE
FINNKINO	FINNKINO

名称	ID
FINNPARK	FINNPARK
FIRATBORU	FIRATBORU
FIRATPEN	FIRATPEN
第一资本	第一资本
第一种药物	第一种药物
第一站	FIRST_STOP
FIRSTAR	FIRSTAR
第一站	第一站
FISH & CO	FISH_&_CO
FISK	FISK
健身服务	健身服务
健身 24 七	FITNESS_24_SEVEN
五	五
五便士	五便士
旗楼	旗楼
FLAIRHOTEL	FLAIRHOTEL
快闪市场	快闪市场
调味爆米花	调味爆米花
飞行中心	飞行中心
FLO	FLO

名称	ID
FLO 小酒馆	flo_bras系列
传单	传单
飞鹰	FLYING_EAGLE
FM 风格	FM_Style
FMB	FMB
FMFB	FMFB
FNB	FNB
折叠后悬挂	折叠后悬挂
FOLIEKe	FOLIEKe
FONA	FONA
FONE4 手机	FONE4_手机
FOOD MARCHE 美食街	Food_marche_food_Court
FOOK TAI	FOOK_TAI
FOOTIN	FOOTIN
FORA-BANK	FORA-BANK
预测	预测
森林必需品	森林必需品
外汇	外汇
外汇银行	外汇银行
理想形式	FORMA_IDEALE

名称	ID
张福尔摩沙	FORMOSA_CHANG
福尔摩沙富茂加油站	Formosa_fu_mao_Gao 加油站
福尔摩沙加油站	福尔摩沙加油站
福尔摩沙泰亚加油站	Formosa_tai_ya_Ga_Gas Station
一级方程式	FORMULE_1
FORNET	FORNET
要塞	要塞
财富餐厅	财富餐厅
CU 论坛	FORUM_CU
论坛餐厅	论坛_餐厅
论坛购物	论坛_购物
FORWARD	FORWARD
FOSCHINI	FOSCHINI
FOTOJOKER	FOTOJOKER
FOTOLAB	FOTOLAB
FOTOMAX	FOTOMAX
福田	福田
四柱酒店	四柱酒店
福朋喜来登酒店	喜来登福朋喜来登酒店
FPT	FPT

名称	ID
FQ&L	FQ&L
香味	香味
弗兰卡	弗兰卡
法国阿诺	FRANCE_ARNO
FRANCK MULLER	FRANCK_MULLER
弗兰克教务长	FRANCK_PROVOST
FRANCO FERUZZI	FRANCO_FERUZZI
弗兰萨银行	弗兰萨银行
FRANSMANNI	FRANSMANNI
弗朗茨系列	FRANZ_COLLEC
弗雷泽住宅	弗雷泽公寓
弗雷泽套房	FRASER_SUITES
短语	短语
弗雷德	弗雷德
弗雷德·佩里	FRED_PERRY
新科	新科
新鲜停止	新鲜停止
FRETEX	FRETEX
FRISTO	FRISTO
FRIULADRIA CRÉDIT AGRICOLE	FRIULADRIA_CRÉDIT_AGRICOLE

名称	ID
FROIZ	FROIZ
FRONTWAVE	前波_CU
FSP	FSP
富山檀香木	FU_SHAN_SANDAL_WOOD
付元豪	FU_YUAN_HAO
富甸银行	富电银行
富士	富士
富士自行车	富士自行车
藤山	藤山
全力以赴	全力以赴
富勒姆宫	FULUM_PALACE
欢乐的一天	FUN_DAY
FUNK'N'SOUL	FUNK'N'SOUL
时髦的鱼	FUNKY_FISH
FURLA	FURLA
FURMART	FURMART
FUSSL Modestraße	fussl_modestraße
FUTA 快车	Futa_Expres
FUZZIWIG 的糖果工厂	FUZZIWIG'S CANDY_FACTORY
FX 酒店	FX_Hotel

名称	ID
Faasos	Faasos
Fabi	Fabi
Fabiana Ricco	Fabiana_Ricco
法比亚尼	法比亚尼
Fabindia	Fabindia
法比奥·卢奇	Fabio_Lucci
法国面料	Fabricas_de_Francia
织物兰	织物兰
Fabrico	Fabrico
面料水疗中心	面料水疗中心
Fabrika	Fabrika
Fabrika Kachestva	Fabrika_Kachestva
Fabrika Obuvi	Fabrika_Obuvi
Fabrique	Fabrique
Fabrique Stenugnsbageri	Fabrique_Stenugnsbageri
Faces 电影院	面孔_影院
Factorie	Factorie
工厂卡和派对专卖店	工厂卡片和派对奥特莱斯
Faeton	Faeton
Fagmöbler	Fagmöbler

名称	ID
Fahasa	Fahasa
Faigo 火锅	Faigo_Hot_Pot
淑女	Fair_Lady
公平竞赛赌场	公平竞赛赌场
公平竞赛体育酒吧	Fair_Play_Sport_Bar
公平交易	公平交易
费尔菲尔德 FCU	Fairfield_FCU
万豪费尔菲尔德客栈	万豪费尔菲尔德酒店
费尔菲尔德万豪酒店	费尔菲尔德万豪酒店
费尔菲尔德度假村	费尔菲尔德度假村
费尔蒙特酒店	费尔蒙酒店
公平的价格	公平的价格
公平价格快递	Fairprice_Xpr
球道市场	球道市场
费尔威瑟	费尔威瑟
费尔伍德	费尔伍德
仙女	仙女
仙境酒店	Fairyland_Hotel
费萨尔伊斯兰银行	费萨尔_伊斯兰银行
信仰	信仰

名称	ID
Fakir	Fakir
事实	事实
Falafel Baribu'a	falafel_baribu'a
Falkensteiner	Falkensteiner
福尔里弗市政大学	Fall_river_Municial_cu
Fallas 折扣店	Fallas_Discount_Store
Falles Kids	Fallas_Kids
Fallas Paredes	Fallas_Paredes
Familia	Familia
Familia 酒店	Familia_酒店
家庭大奖赛	家庭大奖赛
家庭银行	家庭银行
家庭基督教商店	家庭_基督教_商店
家族企业	家族_公司
家庭美元	家庭_美元
家庭快车	全家快递
家庭票价	家庭票价
家庭理财 CU	家庭_财务_cu
Family First of NYY	Family_first_of_ny_fcu
家庭食品	家庭_食物

名称	ID
家庭地平线 CU	家庭_Horizons_cu
家庭旅馆	Family_Inn
家庭厨房	家庭_厨房
全家便利店	全家便利店
家庭药房	家庭药房
家庭储蓄 CU	Family_savings_cu
家庭安全 CU	家庭_安全_cu
家庭视频	家庭_视频
家庭大奖赛	家庭大奖赛
家庭鞋	家庭鞋
著名的阿莫斯	Famous_Amos
来自中国江南的著名厨师	著名厨师_来自中国江南_中国
著名的戴夫的	Famous_Dave's
著名鞋类	著名鞋类
Fana Sparebank	Fana_Sparebank
花式馅料	Fancy_fillings
方	方
方索公社	Fang_Suo_Commune
泛华公司	泛华股份有限公司
凡丽酒店	泛丽酒店

名称	ID
Fann parfumerie	fann_parfumerie
房利美糖果	Fannie_May_Creftions
神奇的家具	神奇的家具
神奇山姆斯	Fantastic_Sams
Fantastico	Fantastico
Fanzz	Fanzz
远东银行	Far_Eastern_Bank
远东国际银行	远东国际银行
法兰达酒店	Faranda_酒店
车道	车道
Fargerike	Fargerike
农场男孩	Farm_Boy
农场信贷服务	农场_信用_服务
农场新鲜	农场_Fresh
农场新鲜药房	农场_新鲜_药房
农场服务机构	农场服务机构
农场来源	农场来源
农场商店	Farm_Store
Farm-Liga	Farm-Liga
Farma	Farma

名称	ID
Farma & Cia	Farma_&_Cia
Farma & Farma	Farma_&_Farma
Farma Red	Farma_Red
Farma Total	Farma_Total
Farma-Plus	Farma-Plus
FarmaCenter	FarmaCenter
Farmacia 拉巴斯	Farmacia_La_Paz
法玛西亚巴黎	法玛西亚_巴黎
巴斯德药房	Farmacia_Pasteur
法玛西亚 SAAS	Farmacia_Sa
圣米格尔药店	Farmacia_San_Migel
法玛西亚托雷斯	法玛西亚_托雷斯
环球药房	环球药业
美国广播公司	Farmacias_Abc
Farmacias Benavides	Farmacias_Benavides
Farmacias Farmax	Farmacias_Farmax
Farmacias GI	Farmacias_GI
Farmacias 瓜达拉哈拉	Farmacias_瓜达拉哈拉
圣安娜药房	Farmacias_Santa_Ana
Farmacias SFA	Farmacias_SFA

名称	ID
圣安东尼奥药店	Farmacias_San Antonio
Farmacias 旧金山阿西斯	Farmacias_san Francisco_de_asis
Farmacias San Pablo	Farmacias_San_Pablo
类似药房	Farmacias_Similares
Farmacias del Ahorro	Farmacias_del_aHorro
卡门药房	Farmacias_del_Carmen
Simi 博士的药房	Farmacias_del_dr._simi
Farmaciile Dona	Farmaciile_Dona
法玛城	法玛城
法尔马达尔	法尔马达尔
Farmais	Farmais
Farmakon	Farmakon
Farmakopeyka	Farmakopeyka
Farmakor	Farmakor
Farmalider	Farmalider
Farmani	Farmani
Farmaofertas	Farmaofertas
Farmooliva	Farmooliva
Farmapronto	Farmapronto
Farmarosa	Farmarosa

名称	ID
Farmatodo	Farmatodo
Farmatotal	Farmatotal
Farmatsiya	Farmatsiya
Farmco	Farmco
Farmekonom	Farmekonom
农夫男孩	农民_男孩
农夫杰克	农民杰克
农民	农民
农商银行	Farmers_&_Merchants_Ban
农民银行和信托基金	农民_银行_和_信托
农民家用家具	农民_家居_家具
农民保险	农民_保险
农民国家银行	农民国家银行
农民联盟	农民联盟
农民联盟保险	农民联盟_保险
农业	商业和工业 Groc Star
农田	农田
法姆伦德	法姆伦德
农场市场	农场市场
安杰洛尼药房	Farmácia_Angeloni

名称	ID
药房大道	Farmácia_Avenida
Farmácia GBarbosa	Farmácia_ GBarbosa
Farmácia Maxxi	Farmácia_Maxxi
永久药房	药房_永久药房
Preço Preço Propular	Farmácia_Preço_Propular
关联药房	Farmácias_associadas
Farmácias Capilé	Farmácias_Capilé
Farmácias FTB	Farmácias_FTB
Farmácias Guararapes	Farmácias_Guararapes
Farmácias Multmais	Farmácias_Multmais
Farmácias Nissei	Farmácias_Nissei
Farmácias So Joan	Farmácias_Sao_Joano
Farnas	Farnas
Farruco	Farruco
Faruk Güllüoglu	Faruk_Güllüoglu
快速 Gas Plus	Fas_Gas_Plus
Fas mart	Fas_mart
FasTracKids	FasTracKids
法西诺	法西诺
时尚错误	Fashion_Bug

名称	ID
Big Bazaar 的时尚	Fashion_at_big_Bazaar
Fasol'	Fasol'
法斯宾德酒店	Fassbind_Hotels
快速	快速
快速 & 新鲜	Fast_&_Fresh
Fast 109 连锁酒店	Fast_109_Hotel_Chain
快速汽车贷款	快速汽车贷款
快速摩托车服务	快速摩托车服务
快速停止	Fast_Stop
快速通道叫出租车	快速通道_呼叫_出租车
FastFrame	FastFrame
FastLane	FastLane
法斯塔意大利面	Fasta_Pasta
Fastenal	Fastenal
Fasthotel	Fasthotel
系好了	系好了
Fastop	Fastop
Fastrac	Fastrac
Fastrack	Fastrack
Fastrip	Fastrip

名称	ID
Fat Bastard Burrito	Fat_Bastard_Burrito
胖爸爸	Fat_Daddy
胖脸	Fat_Face
Fatayer Ala Altayer Bakery	Fatayer_Ala_Altayer_Bakery
脂肪汉堡	脂肪汉堡
信仰真主	Fath_Allah
Fatto a Mano	fatto_a_mano
Fatz Cafe	Fatz_Cafe
Fauji 基金会学校	Fauji 基金会_学校
Faulu	Faulu
Faunatar	Faunatar
Interniforita	Favorita_interni
最喜欢的市场	最喜欢的市场
一汽吉林	一汽_吉林
一汽马自达	一汽_马自达
一汽车主	Faw_Owner
一汽-大众	一汽-大众
FaxCopy	FaxCopy
费萨尔银行	Faysal_Bank
Fazal Din	Fazal_Din

名称	ID
Fazoli's	Fazoli's
Fbb	Fbb
美联储快报	Fed_Express
FedEx	FedEx
FedEx 已授权 ShipCenter	FedEx_已授权_ ShipCenter
FedEx 投递箱	FedEx_Drop_Box
FedEx 员工信贷协会	FedEx_员工_信用协会
FedEx 货运	FedEx_运费
FedEx 办公室	FedEx_办公室
FedEx 船舶中心	FedEx_Ship_Center
联邦银行	联邦银行
尼日利亚联邦抵押贷款银行	尼日利亚联邦抵押贷款银行
联邦储蓄银行	联邦储蓄银行
联邦保险	联邦保险
费德里科的墨西哥菜	Federico's_Mexican_Food
Feel	Feel
Feeloom Hotel	Feeloom_Hotel
Fego Caffé	Fego_Caffé
Fei Zap Mai Laan	Fei_Zap_Mai_Laan
Feidachu Lajio Chaoo Chaoo	Feidachu_Lajaio_Chaorou

名称	ID
飞迪汽车	飞迪汽车有限公司
费拉诺瓦	Feira_Nova
Feira dos Tecidos	feira_dos_tecidos
飞跃教育	飞跃_教育
Felber	Felber
Felfela	Felfela
菲利克斯	菲利克斯
Felna	Felna
Felsozsolca Takarék	Felsozsolca_Takarék
Felta	Felta
费尔特里内利	费尔特里内利
芬迪	芬迪
Fenerium	Fenerium
FengYu	FengYu
丰城农村商业银行	丰城_农村_商业银行
丰和工作室	丰和工作室
风华	风华
丰汇学校	凤汇学校
奉贤农村商业银行	丰县_农村_商业银行
丰泽园	丰泽园

名称	ID
芬尼西亚银行	Fenicia_Bank
FenNo	FenNo
FercheGas	FercheGas
弗格斯酒店	弗格斯_酒店
弗格森	弗格森
Ferguson FCU	Ferguson_FCU
弗格森广场	Ferguson_Plare
花瓣上的蕨类植物	Ferns_n_Petals
芬伍德	芬伍德
费罗尼亚酒店	Feronia_Hotel
法拉利	法拉利
法拉利正装和婚纱	法拉利_正装_&_Bridal
费雷	费雷
Ferrellgas	Ferrellgas
Ferroli	Ferroli
新鲜的花园	Fesh_Garden
费斯塔	费斯塔
Fester	Fester
节日中心	节日中心
节日食品	节日_食物

名称	ID
Festval Cascavel	Festval_Cascavel
Feu Vert	Feu_Vert
FeuVert	FeuVert
FiaBank	FiaBank
Fialka	Fialka
Fibabanka	Fibabanka
Fibanc	Fibanc
Ficcus	Ficcus
富达银行	Fidelity_Bank
富达投资	忠诚投资_投资
富达全国产权保险	Fidelity_National_Title_保险
Fidocombank	Fidocombank
基准	基准
字段	字段
美国嘉年华	Fiesta_Americana
嘉年华旅馆	Fiesta_Inn
嘉年华玛特	Fiesta_Mart
嘉年华沙龙	Fiesta_Salon
第五三银行	Fifth_Third_Bank
百分之五十	百分之五十

名称	ID
费加罗	费加罗
费加罗的披萨	Figaro's_Pizza
Fika 咖啡和食品	Fika_Coffee_&_Food
Fila	Fila
Filiberto's	Filiberto's
Filippa K	Filippa_K
菲利皮的披萨石窟	Filippi's Pizza_Grotto
Filli Boya	Filli_Boya
Fillo Kargo	Fillo_Kargo
Filoil	Filoil
Filta	Filta
Fimotel	Fimotel
FinComún	FinComún
Fina	Fina
Fina Bank	Fina_Bank
Finam	Finam
金融中心 First CU	Financial_CU_CU
Financial Health	财务_健康_FCU
金融一号 CU	Financial_one_cu
金融信托 FCU	Financial_trust_FCU

名称	ID
金融信心	金融家_信心
金融信心代理商	金融家_Confianza-Agente
Finansovyi Soiuz Bank	Finansovyi_Soiz_Bank
Finansy Tat Kredyt	Finansy_ta_Kredyt
Finbank	Finbank
Finbond 互惠银行	Finbond_Mutual_Bank
芬卡圣诞老人 VeraCruz	Finca_Santa_ VeraCruz
Finhomestimal 银行	Findomestic_Banka
精美超市	Fine_Fare_Sup
上等葡萄酒和好烈酒	Fine_Wine_&_Good_Spirits
Fineksbank	Fineksbank
Finex CU	finex_cu
手指甲 2Go	Finger_Nails_2Go
Finibanco	Finibanco
终点线	Finish_Line
芬克	芬克
Fino 支付银行	Fino_Payments_Bank
Finrostbank	Finrostbank
Fintro	Fintro
Fio banka	Fio_Banka

名称	ID
菲奥尼亚银行	Fionia_Bank
佛罗伦萨	佛罗伦萨
Fipola	Fipola
火山	火山
消防员社区 CU	消防员_社区_cu
萤火虫	萤火虫
Firefly CU	Firefly_cu
火狐自行车	火狐_Bikes
Firehouse 潜艇	Firehouse_Subs
Firelands FCU	Firelands_FCU
凡世通	凡世通
火区	火区
Firinci Orhan	Firinci_Orhan
Firkin Pub	Firkin_Pub
首次激活	First_Act
第一联盟银行	First_Alliance_Bank
第一美国银行	First_American_Bank
第一美国 CU	First_American_cu
第一美国国民银行	第一美国国民银行
第一银行	First_Bank

名称	ID
一垒	First_Base
第一个 CU	first_cu
第一辆车出租	First _Car _租赁
第一中央 CU	first_central_cu
首选理发师	First Choice_Haircuters
首选酒	首选酒精饮料
第一公民银行	第一公民银行
第一城市纪念碑银行	第一城市纪念碑银行
头等舱	头等舱
第一商业银行	第一商业银行
第一联邦银行	第一联邦银行
第一社区银行	第一社区银行
第一社区 CU	first_Community_cu
第一联合银行	First_Consolidated_Bank
第一便利银行	第一便利银行
第一达科他国家银行	First_Dakota_National_Bank
第一农民银行和信托基金	First_Farmers_Bank_&_Trus
第一联邦	第一_联邦
加州第一联邦银行	加州第一联邦银行
第一联邦社区银行	First_Federal_社区_银行

名称	ID
第一金融银行	第一财经银行
第一财经 CU	First_Financial_cu
第一佛罗里达州大学	First_florida_cu
第一将军 CU	first_General_cu
第一海湾银行	First_Gulf_Bank
第一夏威夷银行	First_Hawaiian_Bank
第一地平线银行	第一地平线银行
第一酒店	First_酒店
第一独立银行	第一独立银行
第一州际银行	第一州际银行
第一投资银行	第一投资银行
第一夫人	First_Lady
第一次飞跃	First_Lea
第一市场银行	第一市场银行
第一招商银行	First_Merchants_银行
中西部第一银行	First_Midwest_Bank
第一密苏里州大学	first_missouri_cu
全国第一名	全国 First
第一国民银行	第一国民银行
阿拉斯加第一国民银行	阿拉斯加第一国民银行

名称	ID
第一国民银行和信托公司	First_National_Bank_and_trust_C
北加州第一国民银行	北加州第一国民银行
奥马哈第一国民银行	奥马哈第一国民银行
第一全国房地产	First_National_房地产公司
第一个纽约 FCU	first_new_York_FCU
第一尼亚加拉	第一_尼亚加拉
第一北部 CU	first_northern_cu
第一先锋队 FCU	first_pioneers_FCU
第一共和国银行	第一共和国银行
第一证券银行	第一证券银行
第一服务 CU	第一服务_cu
First Service	First_service_FCU
第一南岸	第一南岸
第一州立银行	第一州立银行
第一州立社区银行	第一州社区银行
一线队房地产	First_Team_Real_Estate
首次过境	首次公交
第一信托银行	第一信托银行
第一信任 CU	first_trust_cu
第一美国社区 CU	first_us_community_cu

名称	ID
弗吉尼亚第一银行	First_Virginia_Bank
第一只手表	First_Watc
First Weber	First_Weber
第一女性银行	First_Women_Bank
FirstBank	FirstBank
FirstLight FCU	FirstLight_FCU
FirstMerit 银行	FirstMerit_银行
FirstCr	FirstCr
第一只宠物	第一只宠物
Firstpet 市场	第一只宠物市场
菲舍尔	菲舍尔
Fish & Chip Co	Fish_&_Chip_Co
Fish & Go	Fish_&_Go
炸鱼和薯条	Fish_and_Chips
鱼城烧烤	Fish City_Grill
Fish Inn	Fish_Inn
鱼市场	鱼市场
鱼窗清洁	Fish_Window_Cleaning
Fish180	Fish180
FishMore	FishMore

名称	ID
Fishaways	Fishaways
费舍尔汽车零部件	Fisher 汽车零部件
鱼贩	鱼贩
Fissman	Fissman
Fit Body Boot	Fit_Body_Boot_Camp
健康护理仓库	fit_care_depo
Fit24	Fit24
FitCurves	FitCurves
FitFlop	FitFlop
健身 19	Fitness_19
健身公司	健身公司
健身第一	健身_优先
健身房	健身房
健身小屋	Fitness_Hut
健身 One	Fitness_One
健身时间	Fitness_Time
一起健身	一起健身
美国健身	Fitness_USA
健身世界	健身世界
健身 24Seven	健身 24Seven

名称	ID
Fitnessdk	Fitnessdk
健身工厂	健身工厂
Fitsme	Fitsme
Fitzsimons CU	fitzsimons_cu
五个以下	Five_Below
加州大学五郡分校	five_county_cu
五个家伙	Five_Guys
五颗星	Five_Star
五星鸡	Five_star_Chicken
五点钟	五点钟
Fivestar	Fivestar
修复	修复
修复自动	Fix_Auto
固定价格	固定价格
修复白银	修复_白银
Fixet	Fixet
Fixus	Fixus
Fiyda	Fiyda
Fj Delikatesy	Fj_Delikatesy
Fladeo	Fladeo

名称	ID
旗帜酒店	Flag_Hotel
弗拉格斯塔银行	Flagstar_Bank
Flam's	Flam's
火焰肉鸡	Flame_Broiler
弗拉门戈	弗拉门戈
Flamengo-kvetiny	Flamengo-kvetiny
火烈鸟电影院	Flamingo_Cinema
Flamol	Flamol
Flanco	Flanco
弗拉尼根的	弗拉尼根的
Flare	Flare
快闪咖啡	Flash_Co
快闪食品	Flash_Foo
Flash	Flash Wash
平顶烧烤炉	平顶烧烤
实例集	实例集
舰队农场	Fleet_Farm
舰队之脚	Fleet_Feet
舰队燃料	舰队燃料
Fleischerei Richter	Fleischerei_Richter

名称	ID
Flekkefjord Sparebank	Flekkefjord_Sparebank
Fleming's Prime 牛排馆和葡萄酒吧	Fleming's_Prime_Steakhouse_&_wine_Bar
Flessabank	Flessabank
弗莱彻	弗莱彻
Fleur De Lis FCU	fleur_de_lis_fcu
Fleur de Sel	Fleur_de_Sel
Fleury Medicina e Saúde	fleury_medicina_e_saúde
Flexa	Flexa
Flinkster	Flinkster
百翔花园酒店	Fliport_花园酒店
Fliport 酒店	Fliport_Hotel
Flisekompaniet	Flisekompaniet
Flo 服装店	Flo_Clothing_Stor
Flo&Jo	Flo&Jo
Flop	Flop
防洪	洪水控制
地板和装饰	地板和装饰
地板覆盖物国际	地板覆盖物_国际
美国地板协会	地板_美国
地板额外服务	Flooring_Xtra

名称	ID
楼层直销	地板_直播
花朵咖啡厅	flor_do_Café
弗洛拉尔酒店	Floral_Hotel
弗洛森特鲁姆福尔蒂诺娃	florcentrum_Foltyn
弗洛雷斯塔	弗洛雷斯塔
佛罗里达州立大学 CU	佛罗里达州立大学加州大学
弗洛里斯	弗洛里斯
Floristik 99	Floristik_99
弗洛里托	弗洛里托
弗洛里亚	弗洛里亚
弗洛尔斯海姆	弗洛尔斯海姆
Florín 2000	Florin_2000
Flota PIEPRZYK Moc Jakosc Zysk	flota_pieprzyk_moc_jakosc_zysk
Flou Flou	Flou_Flou
Flowerama	Flowerama
Flowerbox	Flowerbox
鲜花烘焙公司	Flowers_Baking_公司
弗洛伊德	弗洛伊德
法拉盛银行	法拉盛银行
Fly	Fly

名称	ID
飞车	飞车
飞行蛋糕	飞行_蛋糕
飞行花园	飞行花园
飞行 J	flying_J
飞行器	飞行机器
飞虎	Flying_Tiger
哥本哈根飞虎队	Flying_Tiger_哥本哈根
飞行 V	Flying_V
飞乐园	飞乐园
Flügger	Flügger
Flügger Farve	Flügger_Farve
Flügger farg	flügger_farg
焦点 CU	focus_cu
Fogo de Chao	fogo_de_chao
Fogo de Chao	fogo_de_Cháo
Fogon Do Brasil	Fogon_Do_Brasil
Fokstrot	Fokstrot
Fokus Optik	Fokus_Optik
文件夹	文件夹
叶酸	叶酸

名称	ID
Folkets 简介	Folkets_Bio
Folktandvarden	Folktandvarden
Follett	Follett
Folli Follie	Folli_Follie
Folunsi	Folunsi
方达	方达
Attawfiq 小额信贷基金会	基金会_Attawfiq_Micro_Finance
Fone Zone	Fone_Zone
Fonocentro	Fonocentro
丰塔纳鸡	丰塔纳_Chicken
Fonville Morisey Realty	Fonville_Morisey_Realty
FoocoMart	FoocoMart
食物 4 减	Food_4_Less
Food 4 Less 燃料中心	Food_4_Less_Fuel_Center
Food 4 Less 药房	Food_4_Less_药房
食物基础知识	食物基础知识
美食集市	Food_Bazaa
美食城	Food_City
快餐	Food_Fast
食品巨头	食物巨人

名称	ID
美食大厅	Food_Hall
食物之狮	Food_Lion
美食爱好者市场	Food_LOVER's Market
食品超市	Food_Mart
美食广场	美食广场
美食共和国	食物共和国
食物之路	Food_Way
食物称重机	食物称重
美食世界超市	美食世界_超市
食物-燃料	食物-燃料
FoodBox	FoodBox
FoodMaxx	FoodMaxx
Foodary	Foodary
Foodissimo	Foodissimo
美食乐园	美食乐园
Foodman	Foodman
食品公司	Foods_Co
Foods Co 燃料中心	Foods_Co_Fuel_Center
食品公司药房	Foods_Co_Phar
美食城	美食城

名称	ID
食品工厂	食品工厂
美食世界	美食世界
美食专区	美食专区
Fook	Fook
福明堂中医学 居中对齐	Fook_Ming_Tong_Chinese_Med。 _中心
Foone	Foone
脚部动作	足部动作
脚柜	Foot_Lock
足部解决方案	足部解决方案
Footaction	足迹_美国
脚盘	脚盘
Foothill FCU	foothill_fcu
Foothills CU	Foothills_cu
Foothills FCU	Foothills_fcu
Fora	Fora
Force	Force
福特	福特
福特 Rent-A-Car	福特_租车
Fordonsgas Sverige AB	fordonsGas_sverige_AB
Fore	Fore

名称	ID
预测	预测
ForestHill	ForestHill
永远的 21	Forever_21
永恒旅馆	Forever_Inns
永远全新	Forever_New
Forma Viva Jewelers	Forma_Viva_Jewelers
Formen	Formen
Formet Chelik Kapi	Formet_Celik_Kapi
Formi	Formi
Formido	Formido
福尔摩沙	福尔摩沙
福尔摩沙汽车	福尔摩沙_汽车
台塑光学	Formosa_Optical
福尔摩沙手表	Formosa_Watch
福尔摩沙	福尔摩沙
公式	公式
迪瓦纳方程式	Formul_Divana
一级方程式赛车汽车中心	一级方程式赛车_汽车中心
Zdorov'ya Formula	Formula_zdorov'ya
公式电影院	Formula_kino

名称	ID
一级方程式	一级方程式
Fornetti	Fornetti
福尔诺罗马诺	Forno_Romano
Forstinger	Forstinger
Forstèdernes Bank	Forstèdernes_Bank
阿塔卡迪斯塔堡	Fort_Atacadista
布拉格堡 FCU	fort_bragg_FCU
金融堡CU	Fort_Financial_cu
Fort Sill FCU	fort_sill_FCU
长处	长处
强项遗产	Forte_Heritag
Forte Posthou	Forte_PostHOUSE
ForteBank	ForteBank
ForteFarma	ForteFarma
Fortera CU	Fortera_cu
Fortinos	Fortinos
富通银行	Fortis_Bank
富通银行	Fortis_Banque
富通医疗保健	富通医疗保健
Fortuna Plus	Fortun_Plus

名称	ID
财神银行	财神银行
财富	财富
财富汽车	财富汽车公司
财富证券	财富证券
Fortunoff 后院商店	Fortunoff_Backyard_Stor
四十个眨眼	Forty_Winks
Forum	Forum
论坛电影院	论坛_电影院
Forus 银行	forus_Bank
Fosco	Fosco
佛山农村商业银行	佛山_农村_商业银行
化石	化石
福斯特的好莱坞	福斯特_好莱坞
促进冻结	Fosters_Free
照片优先	Foto_First
Foton Rowor	Foton_Rowor
Fotoprix	Fotoprix
创始人证券	创始人_证券
喷泉轮胎	喷泉_轮胎
四道菜	四道菜

名称	ID
四片树叶	Four_leaves
四季	四季
四季 FCU	Four_seasons_FCU
中国四季酒店	四季_酒店_中国
四方形	四方形
四星	Four_Star
四星披萨	四星披萨
FourStar	FourStar
富尔诺斯	富尔诺斯
Fownes	Fownes
福克斯	福克斯
福克斯纤维	狐狸纤维
福克斯孩子	Fox_Kids
福克斯租车	Fox_Rent_A_Car
福克斯的披萨窝	Fox's_pizza_den
福克斯玛特	福克斯玛特
福克斯顿房地产经纪人	福克斯顿_房地产经纪人
飞龙酒店	香水_酒店
Fran's Café	Fran's Café
France Loisirs	France_Loisirs

名称	ID
FranceCars	FranceCars
弗朗西斯卡的	弗朗西斯卡的
弗朗切斯科·唐尼	Francesco_Donni
法郎弗兰克	法郎弗兰克
Franco Ceccato	Franco_Ceccato
Francoc	Francoc
阿萨多 Frango	Frango_Assado
弗兰肯伯格银行	弗兰肯伯格银行
Frankfinn	Frankfinn
法兰克福储蓄银行	法兰克福储蓄银行
法兰克福大众银行	法兰克福_大众银行
Frankie & Benny's	Frankie_&_Benny's
富兰克林	富兰克林
富兰克林·科维	Franklin Covey
富兰克林-萨默塞特足球俱乐部	富兰克林-萨默塞特_FCU
富兰克林斯	富兰克林斯
Fransa	Fransa
Frantour	Frantour
Fraser Health	Fraser_Health
老鹰兄弟会	Braternal_Of_Eagles

名称	ID
弗雷德和金杰	Fred_&_Ginger
弗雷德·洛亚保险	Fred_Loya_Insurance
弗雷德·迈耶	Fred_Meyer
弗雷德·迈耶燃料中心	Fred_Meyer_Fuel_Center
弗雷德·迈耶珠宝商	Fred_Meyer_Jewelers
弗雷德·迈耶药房	Fred_Meyer_Pharmacy
Fred's	Fred's
弗雷德药房	弗雷德的药房
Freddo	Freddo
Freddy's Frozen Kust	Freddy's_frozen_custard
好莱坞的弗雷德里克	好莱坞弗雷德里克的
弗雷德里克和路易莎	Fredrik_&_Louisa
自由选择烟草师	Free_Choice_Tobacconist
自由的人	自由_People
空闲装备	空闲时间装备
免费1 汽油	Free1_gas
FreeV	FreeV
Freebirds World 墨西哥卷饼	Freebirds_World_Burrito
自由	自由
Freedom 第一名	Freedom_1st_cu

名称	ID
自由船俱乐部	Freedom_Boat_Club
自由家具	Freedom_家具
高速公路	高速公路
高速公路保险	Freeway _保险
弗莱贝格银行	弗莱伯格银行
弗莱堡州立银行	Freiburger_KantonalBank
Freightliner	Freightliner
弗莱辛格银行	Freisinger_Bank
弗雷塔斯	弗雷塔斯
弗雷尔森	弗雷尔森
弗里蒙特银行	弗里蒙特银行
弗里蒙特 FCU	Fremont_FCU
法国面包店	法式面包店
法语连接	法语_连接
Fresc Co	Fresc_Co
壁画	壁画
费森尤斯肾脏护理	Fresenius_kidney_Care
新鲜	Fresh [杂货] _42
新鲜	Fresh [杂货] _22
新鲜且准备就绪	Fresh_&_Ready

名称	ID
新鲜卷饼	Fresh_Burritos
新鲜的选择	新鲜的选择
清新之城	Fresh_Cit
新鲜的外套画家	Fresh_Coat_Painter
新鲜的花园	清新花园
新鲜汉伯	Fresh_Hanber
新鲜的衣物	新鲜衣物
新鲜的传奇	全新传奇
生鲜市场	新鲜市场
生鲜市场药房	生鲜市场_药房
生鲜超市	Fresh Mart
新广场联盟	Fresh_Plaza_Un
新鲜的茶	新鲜_茶
新鲜百里香	新鲜百里香
FreshCorner	FreshCorner
FreshMenu	FreshMenu
Freshens 冰沙公司	Freshens_Smoothie_公司
生鲜市场	生鲜市场
清新自然	清新自然
弗雷赛	弗雷赛

名称	ID
Fressi	Fressi
Fressnapf	Fressnapf
弗雷斯塔	弗雷斯塔
Fri Cykler	Fri_Cykler
星期五的	星期五的
星期五的项目	Friday's Project
朋友集市	Friend_Mar
友好的 7-11	友好_7-Eleven
友好的电脑	Friendly_PC
友情快车	友情快递
友好的杂货商	Friendly_Grocer
友好酒店康福特茵酒店	友好酒店_舒适旅馆
友好酒店品质旅馆	友好酒店_优质旅馆
友好集市	Friendly_Mar
友好超市	友好超市
友谊赛	友谊赛
好友	好友
朋友和家人 CU	朋友和家人_cu
友谊旅馆	友谊旅馆
友谊&波罗	友谊&波罗

名称	ID
弗里斯兰银行	Friesland_Bank
Frisby	Frisby
Frisch's Big Boy	frisch's_big_Boy
Frischmann Aisengart	Frischmann_Aisengart
Friseur Klinck	Friseur_Klinck
Friseur Masson	Friseur_Masson
Friskis & Svettis	Friskis_&_Svettis
Frito-Lay	Frito-Lay
Frituur La Patata	Frituur_La_Patata
弗里茨	弗里茨
Froggy	Froggy
边境调整器	Frontier_Adjusters
边境饼干	Frontier_Biscuit
边境社区 CU	Frontier_Community_cu
弗罗斯特银行	Frost_Bank
冰雪奇缘瓶	冷冻瓶
冰封之心	《冰雪奇缘》
水果和蔬菜城	Fruit_&_Veg_City
水果世界	水果世界
FruitFlowers	FruitFlowers

名称	ID
富有成果	富有成果
Fruitisimo	Fruitisimo
水果流行音乐	Fruits_P
果味混合物	Fruity_Mix
Frullati Cafe	Frullati_Cafe
弗鲁塔里亚金色	Frutaria_Gol
Fry's	Fry's
弗莱的电子产品	fry's_Electr
弗莱的燃料中心	Fry's_Fuel_Center
Fry's Market	Fry's_Marketp
Fry's Marketplace	Fry's Marketplace_Fuel_Center
弗莱的 Marketplace	Fry's_marketplace_Ph
弗莱的药房	Fry's_Pharmac
Frávega	Frávega
弗雷德里克·莫雷诺	Frédéric_Moreno
Fso	Fso
富成牛肉	福成牛肉
富记五金水暖工程	Fu_Kee_金属与水暖_工程
富林汉堡	fu_lin_Haburger
傅乔	Fu_Qiao

名称	ID
付小仙	付小仙
FuKouJu	FuKouJu
富邦证券	富邦证券
Fuddruckers	Fuddruckers
富迪	富迪
Fudzi	Fudzi
FuelUp	FuelUp
富尔特集团酒店	富尔特集团_酒店
Fufu 拉面	Fufu_Ramen
府谷农村商业银行	Fugu_Rural_商业银行
富士一番	富士_一番
富士老友	富士_老友
富士信用金库	富士信用金库
富士超市	富士超市
富士寿司	富士寿司
Fuji_jpn	Fuji_jpn
福建国辉大酒店	福建_国汇_大酒店
福建海峡银行	福建_海峡银行
福建农村商业银行	福建_农村_商业银行
富士胶片	富士胶片

名称	ID
藤冈	藤冈
富士通	富士通
富士屋餐厅	富士屋_餐厅
福井信用金库	福井信用金库
福之鸟	福之鸟
福冈日比木信用金库	福冈_Hibiki_Shinkin_Bank
福冈县信用合作社	福冈县_信用合作社
福岛信用金库	福岛信用金库
完整	完整
饱腹食物	Full_Food
全方位贷款	全谱贷款
Fuller's Inns	Fuller's Inns
全市集市	全市集市
Fullok	Fullok
已全面晋升	完全晋升 [Specialty_store] _18
已全面晋升	完全晋升 [Specialty_store] _46
富尔顿银行	富尔顿银行
富尔顿食品	富尔顿_食品
Fulu Mart	Fulu_Mart
富临渔人码头餐厅	fulum_Fisherman's_Wharf_Restaurant

名称	ID
富临餐厅	Fulum_餐厅
富伦药店	富伦药店
趣味与基础知识	Fun_&_Basics
Fun Bean 的	Fun_Bean's
趣味电影院	Fun_Cinema
船中	船中
资助 Sino Life	Funde_Sino_Life
时髦的冷冻酸奶	Funky_Frozen_酸奶
Funny Docin	Funny_Docin
Funnyland	Funnyland
Funtastic 赌场	Funtastic_Casino
欢乐世界	欢乐世界
Furgocar	Furgocar
FurnCity	FurnCity
家具城	家具城
家具医生	Furniture_Medic
家具排	Furniture_Ro
家具村	家具_Village
愤怒	愤怒
Furr's	Furr's

名称	ID
Furshet	Furshet
富润牛肉	Furun_Beef
Fusion 卡尔扎多斯	Fusion_Calzados
PARKn商店融合	Fusion_by_ SH PARKn OP
Fust	Fust
Futaba Tosho	Futaba_Tosho
富利	富利
未来小子	Future_Kid
阜新农商银行	阜新农村商业银行
Fuxinji 猪肉熟炖鸡	富心记_猪肉_Yripe_stew_Chicken
富阳农村商业银行	富阳农村商业银行
福音音乐	Fuyin_Music
福州农村商业银行	福州_农村_商业银行
Fuzzy's Taco Shop	fuzzy's_taco_shop
Fx Inn	Fx_Inn
物理学	物理学
巧克力工厂	Fábrica_di_Chocolate
Fächerrad	Fächerrad
Félegyházi Pekség	Félegyházi_Pekség
Fökusz Takarék	Fökusz_Takarék

名称	ID
Formula Academia	Formula_Academia
Forde Sparkasse	Forde_Sparkasse
Fürstlich Castell'sche Bank	Fürstlich_Castell'sche_Bank
G	G
G H BANK	G_H_BANK
G Petroli	G_Petroli
G Tec 教育	G_Tec_Educati
G&E 超市	G&E超市
宝洁超市	G&G_超市
G&M	G&M
G&V	G&V
G'ZOX	G'ZOX
G-MED	G-MED
G-Park	G-Park
G-SPORT	G-SPORT
G-STAR RAW	G-STAR_RAW
G-超级	G-超级
G.R.A.S.	G.R.A.S.
G20	G20
G2000	G2000

名称	ID
G500	G500
G7 Mart	G7_Mart
GAASTRA	GAASTRA
GAB Robins	Gab_robins
GAC ACURA	GAC_ACURA
广汽三菱	广汽_三菱
广汽三菱	广汽_三菱
广汽丰田	GAC_TOYOTA
广汽 Zxauto	Gac_zxAuto
GADIS	GADIS
GAES	GAES
GAIL Gas	Gail_gas
学田	学田
晚会快车	晚会快车
GALAMART	GALAMART
老佛爷百货	Galeries_Lafayette
伽马	伽马
GANSO	GANSO
GANT	GANT
间隙	间隙

名称	ID
GAPCO	GAPCO
花园旅馆	花园旅馆
GARMIN	GARMIN
气体	气体
煤气牛仔裤	Gas_Jeans
天然气快递	天然气快递
喘着粗气	喘着粗气
GATE 磨损	Gate_Wear
蛋糕屋	GATEAUX_HOU
GATI	GATI
GAZ Detali Mashin	gaz_detali_mashin
凝视	凝视
GB	GB
GB 快递	GB_Expr
GB 酒店	GB_Hotels
GBS 银行	GBS_Bank
GBarbosa	GBarbosa
GCB	GCB
GCHU 酒店	GCHU_HOTEL
GCR	GCR

名称	ID
GCU	GCU
GDH INN	GDH_INN
通用电气货币银行	ge_money_Bank
GEA	GEA
GECO tabak-tisk	geco_tabak-tisk
吉利	吉利
GEGA OIL	GEGA_OIL
GEICO	GEICO
通用光学	通用_OPTICA
GENIKI 银行	geniki_bank
GENNIS'S	GENNIS'S
GENNIS'S ohoh-mini	Gennis's_ohoh-mini
GENO BANK ESSEN	GENO_BANK_ESSEN
GEOX	GEOX
杰拉德·达雷尔	GERARD_DAREL
GERMANO DE SOUSA	GERMANO_DE_SOUSA
日耳曼诺斯	日耳曼诺斯
格里·韦伯	GERRY_WEBER
GETITNOW.GR	GETITNOW.GR
GEZER	GEZER

名称	ID
GF 酒店	GF_Hoteles
广发证券	GF_SECURITIES
GFA FCU	GFA_FCU
GG TABAK	GG_TABAK
GHC 医疗中心	GHC_medical_Centre
GHETALDUS	GHETALDUS
GHN	GHN
GHT 酒店	GHT_Hotels
巨型自行车	Giant_Bicles
GIEFFE	GIEFFE
GIEVES&HAWKES	GIEVES&HAWKES
GIFFARINE	GIFFARINE
GIGANTTI	GIGANTTI
吉娜劳拉	GINA_LARA
GINKOO	GINKOO
GIOSEPPO	GIOSEPPO
乔瓦尼·加利	GIOVANNI_GALL
GIPPO	GIPPO
吉兰多拉	吉兰多拉
GIRARD-PERREGAUX	GIRARD-PERREGAUX

名称	ID
GIUNTI al Punto	Giunti_al_punto
GIVe	GIVe
GIZIA	GIZIA
GL 咖啡厅	gl_cafe
一瞥	一瞥
玻璃清洁	玻璃清洁
玻璃色调薄膜	玻璃色薄膜
玻璃驱动器	玻璃驱动器
格伦菲尔德	格伦菲尔德
全局包装	全局包装
GLOBEKS	GLOBEKS
GLOBI	GLOBI
GLOBUL	GLOBUL
肾小球	肾小球
格洛丽亚酒店	GLORIA_HOTEL
格洛丽亚度假村	格洛丽亚度假村
GLORIA 眼镜	Gloria_眼镜
GM	GM
通用雪佛兰	通用_雪佛兰
GMAC 保险	GMac_保险

名称	ID
GMAC抵押贷款	Gmac_Mortage
GMAC 房地产	Gmac_Real_Estate
GMC	GMC
GNC	GNC
GNC Livewell	GNC_Livewell
GO	GO
GO COLORS！	GO_COLORS！
去远足	去徒步旅行
去市场	去市场
去运动	GO_SPORT
GOCCO	GOCCO
GODIVA	GODIVA
GOFFIN BANK	GOFFIN_BANK
黄金	黄金
金牌健身房	GOLD'S_GYM
戈尔达斯	戈尔达斯
金车	金车
金咖啡	金咖啡
金鹰	GOLDEN_EAGLE
金鹰酒店及度假村	金鹰酒店及度假村

名称	ID
金色旅馆	GOLDEN_INNS
金美洲虎	GOLDEN_JAGUAR
金色女士们的照片	GOLDEN_LADIES_PHOTO
金色郁金香	GOLDEN_TULIP
GOLDMET 旅馆	GOLDMET_INN
金州证券	金州证券
GOLDWAGEN	GOLDWAGEN
GOLFINO	GOLFINO
GOLFTEC	GOLFTEC
很好享受	很好享受
价格中心不错	价格不错 CENTRE
GOOD2走吧	GOOD2走吧
固特异	固特异
GORDONS 化学家	GORDONS_CHEMISTS
美食公园美食广场	美食公园美食广场
GOcarmat	GOcarmat
GOfit	GOfit
GOportraits	GOportraits
GP Parsik Sahakari Bank	gp_parsik_sahakari_Bank
GPBank	GPBank

名称	ID
优雅光学	优雅光学
GRAIN DE MALICE	GRAIN_DE_MALICE
格兰美利亚酒店及度假村	美利亚大酒店及度假村
GRANCO FCU	GRANCO_FCU
大酒店	大酒店
君悦酒店	君悦酒店
美爵酒店	美居大酒店
宏伟的光学	大光学
奶奶的家	奶奶的家
GRANDOCEAN	GRANDOCEAN
GRANIER	GRANIER
GRANNGÅRDEN	GRANNGÅRDEN
长城证券	长城证券
希腊之光	GREEK_LIGHT
绿十字	绿色十字架
绿色家具	绿色家具
绿草	绿草
绿心	绿心
GREGGS	GREGGS
格雷戈里	格雷戈里

名称	ID
GREYBOX 咖啡	GREYBOX_COFFE
GRIDINN	GRIDINN
格里夫PLUS	格里夫_PLUS
GRN VACACIONES	GRN_VACACIONES
昏昏欲睡	昏昏欲睡
GROMADA	GROMADA
GROS	GROS
GROS MERCAT	GROS_MERCAT
GRS Baterías	grs_baterías
PEGAS 集团	GRUPA_PEGAS
金融集团 Monex	Grupo_Financiero_Mone
莲花集团	GRUPO_LOTU
GRUPO Marquez	Grupo_marquez
GRUPOTEL	GRUPOTEL
GRYCAN	GRYCAN
格兰意式浓缩咖啡	GRÃO_ESPRESSO
GS	GS
GS 加德士	gs_caltex
GS Mahanagar 合作银行	gs_mahanagar_Co-operative_Bank
GS Store	GS_store

名称	ID
GS 超市	GS_Sump
GS	GS_
GS25	GS25
GSC	GSC
GSPC天然气	Gspc_gas
GT	GT
GTA 酒店	GTA_Hotel
GTBank	GTBank
GTO	GTO
GU	GU
瓜纳巴拉	瓜纳巴拉
唐光生	唐光胜
广济药业	广济药业
广西北部湾银行	广西_北部_海湾银行
广州餐厅	广州_餐厅
GUI	GUI
GUIJI	GUIJI
桂林银行	桂林银行
GUINOT 研究所	Guinot_Institut
古井酒店	GUJING_HOTEL

名称	ID
GULDFYND	GULDFYND
海湾银行	GULF_BANK
海湾国际银行	海湾国际银行
GUMAC	GUMAC
GUODA36524 便利店	GUODA36524_便利店
国都证券	国都证券
国森	国森
国元证券	国元证券
GUSTTIMO	GUSTTIMO
GUTA	GUTA
GUTA-BANK	GUTA-BANK
GUTMANN	GUTMANN
GUY LAROCHE	GUY_LAROCHE
GWK Travelex	gwk_Travelex
GYMBABY	GYMBABY
Gabor	Gabor
加布里埃尔	加布里埃尔
广汽菲亚特	Gac_Fiat
Gac Gonow	Gac_Gonow
Gac Hino	Gac_Hino

名称	ID
Gacel	Gacel
Gad	Gad
加达姆牙科诊所	加达姆牙科诊所
Gadzooks	Gadzooks
Gaee	Gaee
Gaga Cafe	Gaga_Cafe
Gaibom	Gaibom
Gailli Markey	Gailli_Markey
节日	节日
Galactico	Galactico
银河石油	银河_石油
Galaxias	Galaxias
银河公司	Galaxy_&_Co
银河影院	银河影院
Gale's	Gale's
Galeria	Galeria
Galeria Alkoholi	Galeria_Alkoholi
考夫霍夫画廊	Galeria_Kaufhof
美容画廊	Galerie_de_Beauté
加利	加利

名称	ID
加利加利	Gali_Gali
加利西亚	加利西亚
伽利略	伽利略
Galito's	Galito's
Gall & Gall	Gall_&_Gall
Galle Watch	Galle_Watch
画廊	画廊
Galleix	Galleix
画廊酒店	画廊_酒店
Viajes 画廊	Gallery_Viajes
加洛度假酒店	Galo_Resort_酒店
Galp	Galp
伽马	伽马
蒲郡信用金库	蒲郡信用金库
Gambrinus	Gambrinus
Game (游戏)	Game (游戏)
游戏现金	Game_Cash
游戏疯狂	游戏_Crazy
游戏狂热	游戏狂热
周六比赛	Game_Sat

名称	ID
游戏 X 改变	Game_X_Change
GameStop	GameStop
游戏工作坊	游戏_工作坊
游戏作品	游戏作品
Gamm vert	gamm_vert
Gan's Food	Gan's_food
岩原学园	Ganbaru_Gakuen
甘德山	甘德山
甘德户外	Gander_户外
Ganezzi	Ganezzi
Gangar Eyenation	Gangar_Eyenation
Gangyagou	Gangyagou
甘家杰	甘家杰
甘娇	甘娇
Ganko-en	Ganko-en
甘肃农村信用社	甘肃农村信用联盟
甘肃榆中农村合作银行	甘肃_豫中_农村_Coop_Bank
赣榆农商银行	甘肃农村商业银行
Gaodijie56	Gaodijie56
高思教育	高思_教育

名称	ID
间隙车身	Gap_Body
GapKids	GapKids
加拉巴托斯	加拉巴托斯
车库广告	Garage_ad
车库泰克	Garage_Tek
加兰蒂银行	Garanti_Bank
Garanti Bankasi	Garanti_Bankasi
Garbanzo 地中海新鲜	Garbanzo_Mediteranean_Fresh
Garbarino	Garbarino
花园法院	花园球场
花园酒店	花园酒店
花园岭	花园岭
Garderobe Mannen	Garderobe_Mannen
加伦塔	加伦塔
加菲猫的	加菲猫的
Garis	Garis
Garlam	Garlam
加兰酒店	Garland_Hotel
Garmoniya zdorov'ya	garmoniya_zdorov'ya
印度尼西亚鹰航空	鹰航_印度尼西亚

名称	ID
美国天然气公司	美国天然气公司
煤气中心	Gas_Center
煤气城	Gas_City
煤气之王	Gas_King
煤气超市	Gas_Mart
Gas N Shop	Gas_N_Shop
Gas Rite	Gas_Rite
Gas and Go	gas_and_go
汽油和迷你超市	煤气和迷你超市
Gas+	Gas+
Gasamat Oil	Gasamat_Oil
Gasland	Gasland
Gasman	Gasman
加索利纳特级	Gasolina_Extra
Gasomax	Gasomax
Gasprocar	Gasprocar
Gast im Schloße	gast_im_schlos
Gastec	Gastec
Gaston y Daniela	gaston_y_daniela
大门	大门

名称	ID
盖特城银行	Gate_City_Bank
Gateau	Gateau
盖特论坛	盖特论坛
盖特威地铁 CU	Gateway_metro_cu
盖特威报摊	Gateway_Newstan
收集 FCU	gather_fcu
Gatidis	Gatidis
Gator's Dockside	Gator's_dockside
Gatta	Gatta
Gatti's Pizza	Gatti's Pizza
Gauchos	Gauchos
高迪	高迪
Gault & Millau	Gault_&_Millau
高蒙	高蒙
加夫拉诺维奇	加夫拉诺维奇
盖洛德酒店	Gaylord_Hotel
Gaz	Gaz
俄罗斯天然气石油公司	俄罗斯天然气石油公司
俄罗斯天然气银行	俄罗斯天然气银行
Gazenergobank	Gazenergobank

名称	ID
Gazith	Gazith
Gazman	Gazman
俄罗斯天然气工业股份公司	俄罗斯天然气工业股份公司
俄罗斯天然气工业银行	俄罗斯天然气工业银行
俄罗斯天然气工业股份公司'	俄罗斯天然气工业股份公司'
格但斯克大教堂 Ksiazki	Gdanski_Dom_Ksiazki
Gedimat	Gedimat
Gedizpen	Gedizpen
极客小队	极客小队
Geeks on Call	Geeks_on_Call
吉恩和理查兹	Geen_&_Richards
Geers	Geers
Geetanjali 沙龙	Geetanjali_Salon
Gegele	Gegele
Gelaoguan	Gelaoguan
Gelateria Parmalat	Gelateria_Parmalat
Gelatissimo	Gelatissimo
意式冰淇淋	Gelato_Italiano
Geldhof	Geldhof
Geldmaat	Geldmaat

名称	ID
格丽酒店	Geli_Hotel
Gelson's	Gelson's
GemeenteAmsterdam	GemeenteAmsterdam
GemeenteRotterdam	GemeenteRotterdam
GemeenteUtrecht	GemeenteUtrecht
双子座	双子座
Gemotest	Gemotest
Gems 全日制餐厅	Gems_All_Day_Dining_Restaurant
GenFed 金融 CU	GenFed_financial_cu
Genbank	Genbank
源台	源台
吉恩·马蒂诺	Gene_Martino
常规	常规
通用合作银行	普通合作银行
通用涂料	通用_Paint
通用裤子公司	General_Pants_Co.
忠意保险	忠意保险
忠意银行	Generali_Bank
忠意中国	将军_中国
忠意全球援助	一般_全球_援助

名称	ID
Generika 药店	Generika_DrugStore
创世纪	创世纪
创世纪健身	创世记_健身
Geneviève Lethu	Geneviève_Lethu
成吉思烧烤	Genghis_Grill
Geniki Taxydromiki	Geniki_Taxydromiki
Genisys CU	genisys_cu
元气寿司	Genki_Sushi
GenoBank DonauWald	GenoBank_DonauWald
Genobank Rhön-Grabfeld	Genobank_Rhön-Grabfeld
Genol	Genol
Genossenschaftsbank Unterallgau	Genossenschaftsbank_Unterallgau
Genpa	Genpa
Genuardi's	Genuardi's
Genviet	Genviet
GeoStar	GeoStar
GeoVista CU	GeoVista_CU
GeoJIT	GeoJIT
Geonet	Geonet
Georg Peck	Georg_Peck

名称	ID
乔治·詹森	George_Jensen
乔治竞争考试学院	George_School_Of_竞技考试
乔治的	乔治的
乔治敦卡夫 CU	Georgetown_kraft_cu
格鲁吉亚汽车典当	公司
格鲁吉亚遗产 FCU	Georgia_Heritage_FC
格鲁吉亚咖啡馆	乔治亚咖啡厅
乔治亚太平洋地区	乔治亚太平洋地区
Georgio 的烤箱新鲜披萨公司	Georgio's Oven_fresh_pizza_co
Geraer Bank	Geraer_Bank
Gerber 碰撞与玻璃	Gerber_Collision_&_Glass
Gerbes	Gerbes
格贝斯药房	Gerbes_药房
Gerkens un Partneri	gerkens_un_Partneri
德裔美国人银行 ITM	German_American_bank_itm
Gerry 的烧烤	Gerry's_Grill
GetGo	GetGo
盖蒂	盖蒂
Gewusst wie	gewusst_wie
Ghanshyam 超级市场	Ghanshyam_超级市场

名称	ID
加亚药房	Ghaya_药房
郭吉	Gi_Guo
GiFi	GiFi
贾科莫·孔蒂	Giacomo_Conti
Gialen	Gialen
Giani 冰淇淋	Giani_Ice_Cream
巨人	巨人
巨型药店	Giant_Drugstore
巨鹰	Giant_Eagle
巨鹰快车	巨人鹰快车
巨鹰市场区	巨鹰市场区
巨型食品店	Giant_Food_Stores
巨型汽油	Giant_汽油
巨型药房	巨人药房
巨虎	Giant_Tiger
Giassi Supermercados	Giassi_Supermercado
吉布斯	吉布斯
基甸国际	Gideons_国际
岐阜工商信用组合	岐阜_商业_&_行业_信用_合作社
岐阜信用金库	岐阜_信用金库

名称	ID
Gigafit	Gigafit
Giligan 的餐厅	Giligan's Restaurant_
Gilma	Gilma
吉梅内斯	吉梅内斯
Gimsa	Gimsa
生姜	生姜
Gingko	Gingko
Gini & Jony	Gini_&_Jony
贾金兰	Ginlan_Jia
吉诺·帕斯科	Gino_Pasco
Gino Passcalli	Gino_Passcalli
Gino's East	Gino's_East
Gino 的披萨	Gino's_Pizza
Ginos's/VIP	Ginos's/VIP
Gintell	Gintell
银座梅甘娜	Ginza_Megane
银座雷诺阿	银座_雷诺阿
佐丹奴	佐丹奴
乔治·阿玛尼	Giorgio_Armani
Giperavto	Giperavto

名称	ID
Gipfel	Gipfel
Giphar	Giphar
Gippokrat	Gippokrat
GiraMondo 维亚耶斯	GiraMondo_Viajes
长颈鹿	长颈鹿
长颈鹿	长颈鹿
长颈鹿学习	长颈鹿_学习
Giramundo	Giramundo
Girbaud	Girbaud
Girias	Girias
吉里亚斯投资	吉里亚斯_投资
Git-N-Go	Git-N-Go
Gitem	Gitem
独家赠送	赠送_独家
纪梵希	纪梵希
吉夫拉尔	吉夫拉尔
Gjun	Gjun
冰川山 CU	Glacier_Hills_cu
格拉登酒店	格拉登酒店
魅力镜头	Glamour_Shots

名称	ID
Glarner Kantonalbank	Glarner_KantonalBank
格拉斯菲特	格拉斯菲特
Glashutte	Glashutte
美国玻璃展	玻璃_美国
玻璃城 FCU	glass_city_FCU
玻璃医生	玻璃_医生
玻璃匠人	玻璃匠人
格拉森斯	格拉森斯
GlaxoSmithKline	GlaxoSmithKline
GlaxoSmithKline 药品	GlaxoSmithKline_药品
Gleaner	Gleaner
Gleetour 酒店及度假村	欢乐之旅_酒店及度假村
格伦市场	格伦市场
格伦代尔联邦银行	格伦代尔联邦银行
Glengarry	Glengarry
格伦马克制药公司	Glenmark_制药
Glenn 体育酒吧	Glen_Sportsbar
滑行涂料	Glidden_Paints
Glidden 专业油漆中心	Glidden 专业油漆中心
Glingerie	Glingerie

名称	ID
闪光	闪光
Gll Wonton	Gll_Wonton
全球艺术	全球_艺术
全球 CU	Global_cu
全球现金	全球_现金
全球德西	全球_德西
全球饮食	全球_饮食
全球教育	全球_教育
环球港	环球港
环球之家	全球之家
全球药房	全球_药房
全球风格	全球风格
全球电信商店	Global_Teleshop
环球宝库	环球财富银行
全球厢式货车线路	全球货车线路
地球村组织	全球_村庄_组织
全球箱子	全球箱子
地球	地球
环球旅行家	环球旅行家
Globus	Globus

名称	ID
Globus Baamarkt	Globus_Baumarkt
Globus EV	Globus_ev
Globus 时尚	Globus_时尚
Glocken Báckerei	Glocken_Báckerei
凯莱大酒店	格洛丽亚大酒店
格洛丽亚·简的	Gloria_Jean's
格洛丽亚牛仔裤	格洛丽亚牛仔裤
凯莱宫海水浴及酒店	Gloria_Palace_Thalasso _&_Hotels
光荣牙科	Glorious_Dental
荣耀时光烧烤	Glory_Days_Grill
Glou	Glou
Gluck	Gluck
Glue Store	胶水存储
Glusco	Glusco
Gm 加油站	通用汽车加油站
Gme Hotel	Gme_Hotel
格蒙德	格蒙德
Gmpbaby	Gmpbaby
Gnóstica	Gnóstica
Go Gas	Go_Gas

名称	ID
Go Local	Go_Local
Go Mart	Go_Mart
去户外	去户外活动
去寿司	Go_Sushi
Go Vita	Go_Vita
Go Wow	Go_Wow
Go wash 洗涤中心	go_wash_centro_de_Lavado
走吧！旅行	走吧！ _旅行
Go'on	Go'on
Go69 披萨	Go69_Pizz
GoCar	GoCar
五知草村	五知草村
Goco	Goco
戈达德学校	Goddard_School
教父的	教父的
Godfreys	Godfreys
Godrej Interio	Godrej_Interio
Godteland	Godteland
Gogi	Gogi_House
Gogoro	Gogoro

名称	ID
Goiko Grill	Goiko_Grill
Gokkusasagi	Gokkusasagi
Golbary	Golbary
黄金海岸 FCU	Gold_coast_FCU
Gold Entimes 健身俱乐部	金奖_Entimes_Fitness_Club
金牌大师	Gold_Meister
金星辣椒	Gold_Star_Chili
GoldLions 100 酒店	金狮_100_Hotel
Goldcross	Goldcross_C
金色 1 CU	Golden_1_cu
Golden 1 信用合作社	Golden_1_Credit_Union
金拱门	Golden_Arch
黄金大道	黄金大道
金小鸡	Golden_Chick
黄金圈 CU	Golden_circle_cu
金畜栏	Golden_Corral
金纸杯蛋糕	Golden_纸杯蛋糕
金龙	Golden_Dragon
金象	金色_大象
金叉	Golden_Fork

名称	ID
黄金未来	Golden_Future
金加仑	Golden_Gallon
金汉斯	Golden_Hans
黄金酒店	Golden_Hotels
Golden Krust	Golden_krust_Bakery
金小牛广场	Golden_Mavericks_Plaza
金平原加州大学	Golden_Plains_cu
金玫瑰	Golden_Rose
金光闪耀	Golden_Shine
金勺	Golden_Spoon
金太阳证券	金太阳证券
GoldenFork	GoldenFork
Goldenwest CU	GoldenWest_cu
金发姑娘	金发姑娘
金马克	金马克
金大师	金大师
Goldoil	Goldoil
Goldoil 市场	Goldoil_Market
金牌健身房快车	Golds_Gym_Expr
Goldsmiths	Goldsmiths

名称	ID
高尔夫	高尔夫
高尔夫等	Golf_Etc。
银河高尔夫	Golf_Galaxy
高尔夫总部	高尔夫总部
高尔夫小镇	Golf_Town
美国高尔夫	Golf_USA
Golff	Golff
Golfshop	Golfshop
Golfsmith	Golfsmith
Goli	Goli
Gomax	Gomax
国美	国美
Gong Cha	Gong_Cha
美好的一天好时光茶	Good_day_good_time_Tea
美味的食物	Good_Eats
好感觉健身	良好的感觉_健身
Good Me	Good_Me
早上好	早上好
早安酒店	早安酒店
Good Pharm	Good_Pharm

名称	ID
善良的撒玛利亚人协会	Good_Samaritan_Society
味道不错	好味道
美好时光	Good_Time
Good Times 汉堡和冰雪奇缘蛋奶冻	Good_Times_Burgers_&_Frozen_Custard
GoodLife 健身	GoodLife_健身
Goodcents 熟食店新鲜订阅	Goodcents_deli_fresh_subs
Goodfood 厨房	好食物_厨房
Goodhope 轮胎	好希望_轮胎
美好的生活	美好的生活
GoodsMarket	GoodsMarket
开局良好	开局良好
古德韦尔	古德韦尔
善意	善意
古迪之家	Goody_Hou
Goody's	Goody's
古迪汉堡屋	Goody's Burger_House
Goooodo	Goooodo
Goorin Bros	Goirin_Bros
GorZdrav	GorZdrav
Gorditas Doña Tota	Gorditas_Doña_Tota

名称	ID
戈德曼斯	戈德曼斯
戈登的珠宝商	Gordon's_Jewelers
Gordão Lanches	Gordão_Lanches
Gorenjska Banka	Gorenjska_Banka
Gorilka	Gorilka
戈尔曼	戈尔曼
天哪	天哪
Gostop	Gostop
被掩盖了	Gotcha_covere
后藤之王	Goto_King
Goubuli	Goubuli
Gountsidis	Gountsidis
美食	美食
Gourmet K	Gourmet_Korv
美味云吞团	Gourmet_Wonton
政府储蓄银行	政府_储蓄_银行
Goyourt	Goyourt
Goz Group	Goz_Group
Gozde Optik	Gozde_Optik
戈佐	戈佐

名称	ID
Gping 健康	Gping_Wellen
GrapAri Telkomsel	Grapari_telkomsel
Grab & Go 便利店	Grab_Go_Convencience_Store
Grab&Go	Grab&Go
GradePower 学习	GradePower_学习
Graebel	Graebel
Graeter's 冰淇淋	Graeter's Ice_cream
Graf	Graf
Grafschafter 大众银行	Grafschafter_Volksbank
格兰杰游戏	Grainger_Games
格兰杰工业供应	Grainger_工业_供应
Grama Vidiyal 小额信贷	Grama_Vidiyal_Micro_Finance
格拉梅迪亚	格拉梅迪亚
格拉敏银行	Grameen_Bank
Grameena Vikas 银行	Grameena_Vikas_Bank
格兰	格兰
Gran Deposito	Gran_Deposito
大男爵	Grand_Barony
大国泰证券	国泰证券
大城市旅馆	Grand_City_Inn

名称	ID
大百货商店	大百货
Frais	Grand_Frais
洲际大酒店	洲际大酒店
君悦酒店	君悦酒店
格兰豪宅餐厅	Grand_Mansion_餐厅
地铁公园大酒店	大都会公园酒店
豪门酒店及度假村	Grand_Noble_酒店及度假村
大光学	格兰德光学
大公园	大公园
大奖赛	大奖赛
大租车站	大租车站
万豪豪华公寓	万豪豪华公寓
丽岑大酒店及度假村	Grand_Rezen_Hotel_Resort
大购物中心	大购物中心
格兰云天大酒店	Grand_Skylight_
大联盟	大联盟
GrandBuy	GrandBuy
GrandOptical	GrandOptical
GrandVision 作者：Fototica	GrandVision_by_Fototica
Grandes etapes Françaises	Grandes_etapes_Françaises

名称	ID
祖母印象	祖母_印象
葛兰奇	葛兰奇
花岗岩城食品和啤酒厂	Granite_City_Food_&Brewery
花岗岩州立大学	Granite_state_cu
花岗岩变换	花岗岩_转换
奶奶的自助餐	Granny's Buffet
格拉西姆	格拉西姆
Gratsiya	Gratsiya
Graubundner Kantonalbank	Graubundner_KantonalBank
Gravis	Gravis
格雷巴电气	Graybar_电动
格拉茨自行车	Graz_Bike
谢谢	谢谢
汽车修理工	Grease_Monke
很棒的美国百吉饼	Great_American_Bagel
很棒的美式饼干	很棒的 American_Cookie
大盆地 FCU	Great_basin_FCU
很棒的加元商店	Great_Canadian_Dollar_Stor
很棒的剪辑	精彩片段
很棒的伊利 FCU	Great_erie_fcu

名称	ID
精彩表情牙科中心	Great_Expressions_牙科中心
很棒的画面效果很好	Great_Frame_Up
大丰收面包公司	Great_Harvest_Bread_Co.
很棒的家庭房地产	Great_Home_Realty
五大湖区 CU	Great_Lakes_cu
很棒的 NorthWest FCU	太棒了_ _F NorthWest CU
大苏格兰社区市场	Great_Scot_社区_市场
大南方银行	Great_Southern_Bank
澳大利亚大南方银行	澳大利亚大南方银行
很棒的牛排和土豆公司	Great_Steak_&_Potato_Company
很棒的牛排餐厅	很棒的牛排餐厅
大树药房	大树药房
长城	Great_Wall
长城荣耀证券	长城_荣耀_证券
长城汽车	长城汽车
长城华西银行	长城_西部_中国_银行
大西方银行	Great_Western_Bank
大西部金融服务公司	Great_Western_金融_服务
西部大酒店	Great_Western_Inn
很棒的包裹	Great_Wraps

名称	ID
GreatGas	GreatGas
大联盟 FCU	Greater_Alliance_FCU
大银行	Greater_Bank
大内华达州 CU	Greater_nevada_cu
大新奥尔良 FCU	Greater_new_Orleans_FCU
大德克萨斯州 FCU	Greater_texas_FCU
大联盟	Greater_Union
GreaterManchester	GreaterManchester
很棒的锅	很棒的锅
Grecotel	Grecotel
希腊人的比萨店	Greek's_Pizzeria
Green	Green
绿咖啡 Nero	Green_Caffe_Nero
绿色磨坊	Green_Mill
Green Mill 餐厅和酒吧	Green_Mill_Restaurant_Bar
绿油	Green_Oil
绿点	Green_Poin
绿色服务	绿色服务
绿茶餐厅	绿茶餐厅
东方绿树酒店	Green_Tree_Eastern_Hotel

名称	ID
绿树旅馆	Green_Tree_Inn
绿色趋势	绿色趋势
绿谷杂货店	Green_Valley_Grocel
绿色洗涤	Green_Wash
绿色之路	Green_Way
绿轮电动车	green_wheel_ev
绿色乡村 FCU	绿色乡村_FCU
GreenFlux	GreenFlux
格林国王旅馆	Green_King_Inns
绿海龟	绿色海龟
Greenelf	Greenelf
绿意咖啡厅	绿地咖啡厅
Greengo	Greengo
Greenhalgh's	Greenhalgh's
格陵兰岛搞笑	Greenland_being_Funy
格陵兰岛直销全球商品	格陵兰岛_直销_全球商品
绿党	绿党
格林威治	格林威治
格林威治专卖店	格林威治专卖店
格林伍德	格林伍德

名称	ID
Greet-Tree 健身俱乐部	GreetTree_Fitness_Club
格雷戈里亚迪	格雷戈里亚迪
Gregory's	Gregory's
Grekiska kolgrillsbaren	grekiska_kolgrillsBaren
Gretel's Pretzels	Gretel's_pretzels
灰狼	Grey_Wolf
Greyder	Greyder
Greyhound 巴士线路	Greyhound_Bus_Lines
Greyhound Package 快递	灰狗_Package_Express
Greylock FCU	Greylock_fcu
Grido	Grido
Griff	Griff
格里戈里亚迪	格里戈里亚迪
Grigoris	Grigoris
Griletto	Griletto
烧烤 66	Grill_66
烧烤了	烧烤了
Grimminger	Grimminger
格里莫尔迪	格里莫尔迪
Grip	Grip

名称	ID
Grisino	Grisino
Gristedes	Gristedes
Griswold 家庭护理	Griswold_Home_Care
Griswold 特别护理	Griswold_Special_Care
杂货店	杂货_奥特莱斯
Groseri	Groseri
Grossi Toidukaubad	Grossi_Toidukaubad
地面回合	地面_回合
H 组 21	Group_H_21
群马	群马
Groupama-Avic 财产保险	Groupama-Avic_财产_保险
Grozd'	Grozd'
Groß-Gerauer Volksbank	Groß-Gerauer_Volksbank
Group Dusa	Grup_Dusa
Group Hatipoglu	Grup_Hatipoglu
Grupo 7 Viajes	Grupo_7_Viajes
Grupo AN	Grupo_an
Aventoura 集团	Grupo_Aventoura
Grupo Balaña	Grupo_Balaña
巴厘岛集团	Grupo_Bali

名称	ID
Boxito 集团	Grupo_Boxito
Grupo CGA	grupo_CGA
Grupo Desfotur	Grupo_desfotur
欧罗巴家具集团	Grupo_Europa_Muebles
Grupo Fariones	Grupo_Fariones
Grupo Glauka	Grupo_Glauka
Grupo Jose Lourenço	Grupo_Jose_Lourenco
Grupo Nabeiro	Grupo_Nabeiro
Grupo ODA Viajes	Grupo_oda_Viajes
旅行小组	Grupo_On_Travel
Grupo Playa	Grupo_Play_Sol
Qualitas 集团	集团_Qualitas
东南亚集团	Grupo_Sea
Grupo Star Viajes	Grupo_Star_Viajes
Grupo Viajes Ribermoon	Grupo_Viajes_Ribermoon
虚拟小组	群组_虚拟
Gránit Bank	Gránit_Bank
Grüne Tankstelle	Grüne_tankstelle
Gsm Box	Gsm_Box
Gu Lov	Gu_Love

名称	ID
顾露顾露	gu_Lu_Gu_Lu
鳄梨调味酱墨西哥食品	鳄梨酱_墨西哥_食品
Guadalupe CU	Guadalupe_cu
关公园	Guan_Park
广南大皮法	广南_大_皮_发
广东	广东
广东发展银行	广东发展银行
广东华兴银行	广东_华兴银行
广东国际大酒店	广东_国际酒店
广东南粤银行	广东_南粤银行
广芳酒店	广芳_酒店
光明	光明
广西临桂农村商业银行	广西_Lingui_Rural_Comm_Bank
广西农村信用社	广西农村信用联盟
广元市包商贵民银行	广元市_包商_贵民银行
广州香蕉叶	广州_香蕉叶
广州友谊商店	广州_友谊门店
广州本田	广州_本田
广州农村商业银行	广州_农村_商业银行
广州证券	广州_证券

名称	ID
冠亚英语学校	Guana_英语学校
守护者	守护者
卫报银行	监护人银行
卫报 CU	Guardian_cu
守护者 CU	Guardians_cu
Guarita	Guarita
Gubernskie Apteki	Gubernskie_Apteki
古奇	古奇
Guenpin Fugu	Guenpin_Fugu
Guerin	Guerin
娇兰	娇兰
Guerrisol	Guerrisol
猜猜	猜猜
宾客宿舍	访客宿舍
Guha Roy 美食店和酒店	Guha_roy_Food_Joints_and_Hotels
桂城购物中心	贵城_购物中心
Guido's	Guido's
公会抵押	公会抵押贷款
桂林国民县银行	桂林国民县银行
贵仁香	贵仁香

名称	ID
广元贵上县银行	桂上县_广元银行
吉他中心	Guitar_Center
吉塔特酒店	Guitart_酒店
贵阳农村商业银行	贵阳农村商业银行
古吉拉特邦天然气	古吉拉特邦_天然气
古尔·艾哈迈德	古尔_艾哈迈德
Gulen Boya	Gulen_Boya
海湾	海湾
海湾非洲银行	Gulf_African_Bank
海湾能源	Gulf_Energy
海湾商店	Gulf_Store
古连天	古连天
海鸥	海鸥
Gullfunn	Gullfunn
格列佛	格列佛
枪	枪
群马未来信用合作社	Gunma_Mirai_Credit_合作社
群马县信用合作社	群马县_信用合作社
Gunze 体育	Gunze_Sports
国安居	国安居

名称	ID
国大药房	国大药房
Guoduomei	Guoduomei
国购广场	国购广场
国联证券	国联证券
国人保险	国人保险
国盛电器	国盛电器
国盛药业	国盛药业
国泰百货	国泰百货
国泰君安证券	国泰_君安证券
国信	国信
Gural Porselen	Gural_Porselen
古尔冈格拉明银行	Gurgaon_Gramin_Bank
Gurgoan Gramin	Gurgoan_Gramin
Gusticus	Gusticus
古斯托	古斯托
Gusto 服装店	Gusto_Clothing_Store
Guthrie Bowron	Guthrie_Bowron
Guthrie's	Guthrie's
古天道香	古天道香
Guwa Yiguoxian	Guwa_Yiguoxian

名称	ID
固阳包商农村银行	固阳_包商_农村银行
Guzman y Gomez	guzman_y_Gomez
Gya Hotel	Gya_Hotel
健身天使	Gym_Angel
Gymboree	Gymboree
金宝贝奥特莱斯	Gymboree_O
Gymguyz	Gymguyz
Gyomuyo Shokuhin Super	Gyomuyo_shokuhin_Super
Gyoza Shokudo Maruken	Gyoza_shokudo_Maruken
Gyu shige	gyu_shige
Gyu-Kaku	Gyu-Kaku
Gyutan Yaki 仙台海米	Gyutan_Yaki_Sendai_Hemmi
Gysyszerter	Gysyszerter
Gyöngy Patika	Gyöngy_Patika
Gyöngyös-Matra Takarék	Gyöngyös-Matra_Takarék
GEMO	GEMO
GUNAYDIN	GUNAYDIN
Gádoros Takarék	Gadoros_Takarek
Gartnerbank	Gartnerbank
Gäubodenbacker Hahn	Gäubodenbacker_Hahn

名称	ID
巨人	巨人
巨人赌场	巨人赌场
光学将军	Genérale_d'Optique
Gönull Kahvesi	Gönul_Kahvesi
Görtz	Görtz
Gülaylar	Gülaylar
Güllüoglu	Güllüoglu
Gümüssuyu Hali	Gumüssuyu_Hali
H 酒店	H_Hotel
H Mart	H_Mart
哈塞缪尔	H_SAMUEL
H&B Housing	H&B_Housing
H&H Reznictvi	H&H_reznictvi
H&M	H&M
H&M Home	H&M_Home
H&O	H&O
H&R Block	H&R_Block
H'elite 酒店	H'elite_Hotel
H+	H+
H-E-B	H-E-B

名称	ID
H-酒店	H-酒店
H. Skrzydlewska	H._Skrzydlewska
H.Stern	H.Stern
H10	H10
H2	H2
H2O	H2O
H2O Plus	H2O_Plus
H4	H4
H: 连接	H: 连接
HAHR BENZIN	HAAHR_BENZIN
HABIServe	HABIServe
八番拉面	八番拉面
HAFELE	HAFELE
HAI	HAI
海南航空	海南航空
发型快递	发型快递
HAIRLEKIINI	HAIRLEKIINI
HAIYA	HAIYA
海悦花园酒店	海悦花园酒店
Halbos-skrine na miru	Halbos-skrine_na_miru

名称	ID
HALCON	HALCON
哈林	哈林
哈尔伯格斯金奖	HALLBERGS_G
汉堡	汉堡
HAMOE 联动酒店	HAMOE_LINKAGE_HOTEL
韩丽轩	HAN_LI_XUAN
HAN'S	HAN'S
花一	花一
HANAK kuchyne a interiery	Hanak_kuchyne_a_interiery
HANDLAR'N	HANDLAR'N
手咖啡	手咖啡
HANDU	HANDU
杂工	杂工
悬而未决	HANG_TEN
杭州风情	杭州风格
HANKO 寿司	HANKO_SUSHI
汉林	汉林
翰林茶馆	HANLIN_TEA_HOUSE
HANNASHAN 韩国烧烤	HANNASHAN_KOREAN_BBQ
汉庭酒店	HANTING_HOTEL

名称	ID
汉庭海友客栈	Hanting_hi_inn
郝怡佳	HAO_YI_JIA
幸福的味道	HAPPINESS_TASTE
快乐又有趣	HAPPY_&_FUNNY
快乐烧烤	快乐烧烤
快乐的孩子们	快乐_孩子们
欢乐谷	快乐谷
哈尔滨银行	哈尔滨银行
艰难地逃跑	HARD_OFF
HARIBO	HARIBO
哈里斯酒店	HARRIS_HOTELS
哈里·威尔逊	HARRY_WILSON
丰收节	丰收节
哈维斯超市	哈维斯_超市
HASCOL	HASCOL
哈万	哈万
哈瓦鸡	哈瓦鸡肉
夏威夷天堂	夏威夷天堂
HAYAL Kahvesi	Hayal_kahvesi
HBC	HBC

名称	ID
HBL 快车	HBL_Expres
HCG	HCG
HCT 运输	HCT_运输
高清捐赠者	HD_DONER
HD Iskender	HD_iskender
高清电源	HD_Suppl
HDBank	HDBank
HDFC 银行	HDFC_bank
HDFC 所以说	HDFC_ERGO
HDFC 住房贷款	HDFC_Home_Loan
何善元	HE_SHAN_YUAN
健康药店	健康药店
重型	重型
重型工具	重型工具
合肥百货大楼	合肥_百货大楼
HEGII	HEGII
HELADOS 必应	Helados_Bing
希腊银行	HELLENIC_BANK
你好！ YUJIWUDI	你好！ YUJIWUDI
下摆	下摆

名称	ID
HEMA	HEMA
HEMMAKWELL	HEMMAKWELL
摇摆不定	摇摆不定
HEMOIL 市场	血液_市场
HEMTEX	HEMTEX
亨德森	亨德森
亨得利	亨得利
恒泰证券	恒泰证券
亨利的	亨利的
赫拉	赫拉
康宝莱	康宝莱
草本学家	草本学家
HERE	HERE
HERKULES	HERKULES
爱马仕	爱马仕
HERNANDO TRUJILLO	HERNANDO_TRUJILLO
唐英雄	HERO_TANG
埃雷罗	埃雷罗
赫斯伯格	赫斯伯格
HEXAL	HEXAL

名称	ID
六角线	六角线
HF	HF
HFC	HFC
氢氟碳化合物银行	HFC_bank
HG 娱乐	HG_娱乐
HG Hoteles	HG_Hoteles
嗨熊猫	嗨_PANDA
高保真公司	HI-FI_CORP
HIB Petrol	Hib_petro
HIBBETT 体育	HIBBETT_SP
HIBER	HIBER
日比谷花坛	HIBIYA_KADAN
HICKEY 的药房	HICKEY'S 药房
HIGH'S	HIGH'S
HILACITY	HILACITY
山顶	山顶
山顶	山顶
HIPER USERA	HIPER_USERA
HIPERCOR	HIPERCOR
HIPOT	HIPOT

名称	ID
HIPOTEKARNA BANKA	HIPOTEKARNA_BANKA
HIPOTELS	HIPOTELS
HIRIKO	HIRIKO
命中	命中
HIT 加油站	Hit_petrol_Station
HITEA	HITEA
HITO 三明治 CHARBROIL	HITO_SANDWICH_CHARBROIL
嗨	嗨
HKB	HKB
HKT	HKT
HM	HM
HM 酒店	HM_Hotels
HM 工作室	HM_Studio
HMS市场	HMS_Markets
HMT	HMT
HMV	HMV
海航商务酒店	HNA_BUSINESS_HOTEL
海航快捷酒店	HNA_express_inn
海航大酒店	HNA_GRAND_HOTEL
HNOSS	HNOSS

名称	ID
HOBBEX	HOBBEX
爱好大厅	HOBBY_LOBBY
HOF	HOF
海天堂	HOI_TIN_TONG
HOKIBA	HOKIBA
你好	你好
你好 CASA	HOLA_CASA
HOLANDA	HOLANDA
假日 KTV	HOLIDAY_KTV
HOLLYS	HOLLYS
主页	主页
国内市场	家居市场
家居超市	HOME_MART
家庭哨兵	HOME_SENTRY
家居商店	HOME_STORE
HOMECENTER	HOMECENTER
HOMESENSE	HOMESENSE
鸿兴海鲜酒家	洪兴海鲜餐厅
洪雅	HONG_YA
洪敏	洪敏

名称	ID
HONORATO	HONORATO
希望 1967	HOPE_1967
HOPE 内衣	希望_内衣
水平	水平
霍恩巴赫	霍恩巴赫
霍恩格雷斯光学	HORNE_GRACE_OTICAL
卢斯医院	HOSPITAL_DA_LUZ
热	热
HOT & roll	Hot_&_roll
炙手可热的烧烤	HOT_FOR_LONG_BBQ
热油	HOT_OIL
火锅自助餐	火锅自助餐
火锅拉面	火锅拉面
热空间	HOT_SPACE
热鞋	热鞋
豪泰租赁与租赁	Hotai_租赁_&_出租
酒店中心	酒店中心
热点	热点
HOTUSA	HOTUSA
呼叫者	呼叫者

名称	ID
小时选择	时长选择
琥珀之屋	琥珀之屋
弗雷泽之家	HOUSE_OF_FRASER
餐厅怎么样	HOW_WAY_RESTAURAN
HP	HP
HPDTC	HPDTC
HPJ Jewelers	HPJ_Jewelers
HRVATSKA POSTANSKA BANKA	HRVATSKA_POSTANSKA_BANKA
汇丰银行	汇丰银行
汇丰卓越	汇丰银行_Premier
汇丰农村银行	汇丰银行_农村银行
HSH Nordbank	HSH_Nordbank
HIN GAU CHAYAO 药店	HSIN_GAU_CHYAO_DRUGSTORE
HSM 圣米歇尔酒店	HSM_酒店_圣米歇尔
HSN	HSN
HT Eronet	ht_eroNet
HTC	HTC
HTMBoxes	HTMBoxes
华冠	华冠
华丽佳和	华丽佳和

名称	ID
HUALUXE	HUALUXE
黄河晶都大酒店	黄河京都酒店
华融证券	华融证券
华生堂	华生堂
华泰证券	华泰证券
华天大酒店	华天酒店
华天客栈	华天客栈
哈德逊湾	哈德森湾
HUDY Sport	Hudy_sport
汇丰堂餐厅	HUI_FENG_TANG__餐厅
汇金百货	HUI_JIN_百货大楼
许刘山	HUI_LAU_SHAN
惠威	惠威
人类	人类
悍马	悍马
湖南餐厅	湖南_餐厅
鸿福堂控股	HUNG_FOOK_TONG_HOLDINGS
洪瑞晨	HUNG_RUI_CHEN
饥饿的狮子	HUNGRY_LION
HUNGUEST 酒店	HUNGUEST_酒店

名称	ID
HUSA	HUSA
赫斯基体育用品专卖店	Husky_Sporting_Goods_store
HUZAR	HUZAR
HWC Roasters	HWC_Roasters
凯悦	凯悦
以凯悦为中心	以凯悦为中心
齐拉拉凯悦酒店	HYATT_ZILARA
凯悦 ZIVA	HYATT_ZIVA
凯悦酒店	Hyatt_House
HYPCO	HYPCO
HYPO-ALPE-ADRIA 银行	HYPO-ALPE-ADRIA_BANK
现代百货店	现代百货店
HZ Hörmittelzentalen	hz_Hörmittelzentelzentalen
HaBenleumi	HaBenleumi
HaShnizelia	HaShnizelia
哈根达斯	哈根达斯
Haan-Gar	Haan-Gar
哈斯	哈斯
哈班酒店	哈班_酒店
哈比卜银行	Habib_Bank

名称	ID
苏黎世哈比卜银行股份公司	Habib_bank_ag_Zurich
哈比卜大都会银行	Habib_Metropolitan_Bank
哈比卜的	哈比卜的
人类栖息地	人类栖息地
哈布图尔酒店	哈布图尔_酒店
Haci Bekir	Haci_Bekir
Haci Sayid	Haci_Sayid
墨西哥庄园	Hacienda_Mexicana
Hacioglu	Hacioglu
伦敦哈克特	Hackett_London
哈迪姆	哈迪姆
HadraMoot	HadraMoot
哈飞	哈飞
Haffner's	Haffner's
Hafo	Hafo
哈加拉	哈加拉
哈格兰	哈格兰
Hagener Stadtbackerei Kamp	Hagener_Stadtbackerei_Kamp
Haggen 食品与药房	Haggen_Food_&_Pharmacy
萩山口信用金库	Hagi_Yamaguchi_Shinkin_Bank

名称	ID
Hagkaup	Hagkaup
马海	Hai_Ma
HaiDiLaoHuoGuo	HaiDiLaoHuoGuo
海昌海洋公园	海昌海洋公园
海底捞	海底捞
Haigh's	Haigh's
海拉姆·科皮蒂亚姆	Hailam_Kopitiam
海门农商银行	海门_农村_商业银行
海明教育	海明_教育
海南农村信用社	海南_农村_信用社
海南美食	Hainanese_Delights
海宁农村商业银行	海宁农村商业银行
海青教育	海青_教育
发型俱乐部	发型俱乐部
Cuttery	Hair_cuttery
头发表情	头发_表达式
发型大师	发型大师
十一号美发沙龙	Hair_Salon_Eleven
发型师	发型师
海尔豪斯仓库	发廊_仓库

名称	ID
发型大师	发型大师
Haishige	Haishige
海通	海通
海翔商业	海翔_商业
海印海记潮汕牛肉火锅	海印海鸡_潮山_牛肉_火锅
海韵音乐	海韵_音乐
Hajdu Takarék	Hajdu_Takarék
Hak Ka Hut	Hak_Ka_Hut
博多一番通	博多_一番通
客家县银行	客家县银行
Hakmar	Hakmar
Hakuritabai Hanbey	Hakuritabai_Hanbey
白山信用金库	白山信用金库
Halcon Viajes	Halcon_Viajes
Haldiram's	Haldiram's
半价图书	半价书籍
Halfords	Halfords
哈利法克斯	哈利法克斯
Halk Ekmek	Halk_Ekmek
哈尔克银行	哈尔克银行

名称	ID
Hallensteins	Hallensteins
Hallertauer Volksbank	Hallertauer_Volksbank
Halles dis	Halles_dis
Hallhuber	Hallhuber
霍尔马克	霍尔马克
万圣节之城	Halloween_City
万圣节快车	万圣节快报
Halo-Apteka	Halo-Apteka
哈洛宁	哈洛宁
Halows	Halows
Halpa-Halli	Halpa-Halli
霍尔斯特德街熟食店	Halsted_Street_Deli
Halyk Bank	Halyk_Bank
滨松岩田信用金库	滨松_岩田_信用金库
Hamashbir Letzarkhan	Hamashbir_Letzarkhan
HambDas	HambDas
汉堡能源	汉堡_能源
汉堡能量电动汽车	Hamburg_Energie_EV
汉堡 Sparkasse	Hamberger_Sparkasse
汉堡大众银行	汉堡_大众银行

名称	ID
Bairro 汉堡店	汉堡店_do_Bairro
汉堡诺斯特拉	汉堡_诺斯特拉
哈密市商业银行	哈密市商业银行
汉密尔顿	汉密尔顿
Hamleys	Hamleys
锤子	锤子
锤子硬件	锤子_硬件
汉普郡	汉普郡
希尔顿汉普顿酒店	希尔顿汉普顿酒店
Han-Dee Hugo 的	Han-Dee_Hugo's
HanLin 韩国餐厅	HanLin_韩国_餐厅
HanLuoYuan 韩式烧烤	HanLuoYuan_Korean_BBQ
花正	花正
花松	花松
哈纳罗俱乐部	Hanaro_Club
Hanaro Mart	Hanaro_Mart
Hanco	Hanco
汉考克银行	汉考克银行
Hancock FCU	Hancock_fcu
汉考克面料	汉考克_面料

名称	ID
精心挑选的酒店	Hand_picked_Hotels
手部和石头按摩和面部水疗	手和石头按摩和面部水疗
半田信用金库	Handa_Shinkin_Bank
Handelsbanken	Handelsbanken
Handi Plus	Handi_Plus
处理	处理
Hands	Hands_Be
方便的 Andy	HANDY_Andy
方便的超市	Hand_Mart
方便的储藏室	方便的储藏室
方便的方法	随心所欲
杂工连接	杂工_连接
杂工很重要	Handyman_Matters
Hanen	Hanen
恒生银行	Hang_Seng_Bank
汉格诊所	衣架诊所
恒隆宫	Hanglung_Palace
杭州解白	杭州_捷百
杭州联华	杭州_联华
杭州联合银行	杭州联合银行

名称	ID
杭州新开元大酒店	杭州_新开元_酒店
汉科超级	Hanko_Super
阪急绿洲	阪急_Oasis
翰林教育	翰林_教育
翰林茶室	翰林茶室
翰林写作	汉林_写作
汉林源	汉林源
汉娜	汉娜
汉娜专卖店	汉娜专卖店
汉纳福德	汉纳福德
汉纳福德药房	汉纳福德_药房
汉娜	汉娜
饭能信用金库	Hanno_Shinkin_Bank
汉诺威大众银行	汉诺威_大众银行
Hanouty	Hanouty
汉斯·安德斯	Hans_Anders
Hans im Glück	Hans_im_glück
汉萨班卡	汉萨班卡
Hansapank	Hansapank
汉萨同盟银行	Hanseatic_Bank

名称	ID
汉森和戴斯维克	Hansen_&_Dysvik
汉腾	汉腾
汉庭高级酒店	汉庭酒店_高级_酒店
韩华银行	韩华银行
HaoShangHao	HaoShangHao
HaoXiangLai	HaoXiangLai
豪德火锅	Haode_Hot_Pot
浩汉冒才	Haohan_Maocai
郝家香	郝家香
郝家源	郝家源
好客酒店	Haoke_Hotel
Haoledi	Haoledi
Haolinju	Haolinju
Hap Chan	Hap_Chan
正在发生	正在发生
Happi Mobiles	Happi_Mobiles
幸福集市	Happiness_Mart
欢乐酒吧和烧烤店	Happy_Bar_&_Grill
快乐盒子	快乐盒子
快乐中心	快乐中心

名称	ID
幸福的家庭	快乐_家庭
快乐食物	快乐_食物
快乐的目标	快乐进球
哈利快乐	Happy_Harry
欢乐马电影院	Happy_Horse_C
快乐 Joe's	Happy_Joe
快乐柠檬	快乐_柠檬
开心玛特	快乐玛特
开心见面	快乐见面
快乐的包装工和搬运工	Happy_Packers_and_movers
快乐的小狗	快乐小狗
快乐租房	快乐租房
快乐之路旅馆	Happy_Road_Inn
幸福的州立银行	Happy_State_Ban
快乐牛排	快乐牛排
快乐的	快乐的
快乐的披萨	Happy's Pizza
Happyday KTV	Happyday_KTV
Hapro Mart	Hapro_Mart
Haprofood	Haprofood

名称	ID
Hapvida	Hapvida
Harald Nyborg	Harald_Nyborg
Harashin NARUS	Harashin_narus
Harat's Pub	Harat's_Pub
哈尔滨农村商业银行	哈尔滨农村商业银行
港口货运工具	港口_货运_工具
HarborLight CU	HarborLight_CU
Harborstone CU	Harborstone_cu
海港广场	海港广场
哈考特	哈考特
硬摇滚咖啡厅	硬摇滚咖啡厅
硬摇滚赌场	硬摇滚赌场
Hard Rock 酒店	Hard_Rock_Hotel
Hard Rock 酒店和赌场	Hard_Rock_Hotel_&_Casino
Hardee's	Hardee's
Hardee's /Red Burrito	Hardee's_/_red_Burrito
哈丁的	哈丁的
哈丁的友好市场	Harding's 友好市场
硬件 Hank	硬件_汉克
Hardy's Health 商店	Hardy's Health_Stores

名称	ID
Haritha Honda	Haritha_Hond
哈雷酒店	Harley_Hotel
哈雷戴维森	哈雷戴维森
Harmon AutoGlass	Harmon_ AutoGlass
Harmon's 邻里杂货店	Harmon's_neighborhood_Gro
和谐医疗	和谐_医疗
和谐	和谐
哈罗德的面包	Harold's_bread
哈罗德的鸡舍	Harold's_chicken_shack
竖琴	竖琴
Harrah's	Harrah's
Harris + Hoole	Harris_+_Hoole
哈里斯围巾	Harris_Scarfe
哈里斯·泰特	Harris_Teeter
哈里斯·泰特药房	Harris_Teeter _药房
哈利和大卫	Harry_&_David
哈里·诺曼房地产经纪人	Harry_Norman_Realtors
哈里·拉姆斯登的	Harry_Ramsden's
海瑞温斯顿	Harry_Winston
哈利和马克斯	Harry_and_Max

名称	ID
哈利的	哈利的
Harrys	Harrys
哈特	哈特
哈特福德 FCU	Hartford_FCU
哈特劳尔	Hartlauer
Hartleys	Hartleys
哈佛先锋医疗协会	哈佛_Vanguard_Medical_Associates
丰收食品	收获_食物
收获能量	收获_能量
收割机	收割机
Harvester 金融CU	收割机_财务_cu
哈维酒店	Harvey_酒店
哈维·诺曼	哈维_诺曼
哈维世界旅行	哈维_世界_旅行
哈维的	哈维的
Harveys 家居专卖店	Harveys_Home_Specialty_St
哈里亚纳邦城市燃气公司	Haryana_City_Gas
Harzer Volksbank	Harzer_Volksbank
Harzsparkasse	Harzsparkasse
有 Kahve Evi	Has_Kahve_Evi

名称	ID
波崎	波崎
Hashi	Hashi
Hasta 20	Hasta_20
前往 La Vista	Hasta_La_Vista
匆忙	匆忙
Hat Shack	Hat_Shack
帽子世界	Hat_World
帽子专区	Hat_Zone
Hatai Mart	Hatai_Mart
Havaianas	Havaianas
哈瓦那	哈瓦那
Havas Voyages	Havas_Voyages
Havmor	Havmor
Havoline Express 润滑油	Havoline_xpress_Lube
夏威夷中部 FCU	Hawaii_central_FCU
夏威夷社区 FCU	Hawaii_community_FCU
夏威夷执法部门 FCU	夏威夷_执法_FCU
夏威夷州立大学 FCU	Hawaii_state_FCU
HawaiiUSA FCU	Hawaiiusa_fcu
夏威夷金融 FCU	Hawaiian_financial_FCU

名称	ID
Hawes & Curtis	Hawes_&_Curtis
华泰汽车	华泰汽车
温德姆豪顿套房酒店	温德姆酒店 Hawthorn_Suites_by_w
海亚特房地产	Hayatt_Real_Estate
海斯	海斯
淡褐色	淡褐色
为 Tails 做好准备	Heads_up_for_tails
Health 100	Health_100
Health 2000	健康_2000
Health Ban	健康银行
Health Care 家庭 CU	医疗保健_家庭_cu
Health First	健康第一_药房
Health House	健康之家_药房
Heath Lane	健康通道
Health Life	健康_生活_国际
Health Management Associat	健康管理助理
Health Mart P	健康智能药房
HealthCare 同事 CU	HealthCare_associates_cu
HealthCity	HealthCity
HealthEast	HealthEast

名称	ID
HealthNet FCU	HealthNet_FCU
HealthShare CU	HealthShare_CU
健康公园	健康公园
健康镜病理学	健康镜_病理学
健康之路医疗诊所	Healthway_Medical_Cl
健康之选	健康之选
健康习惯	健康习惯
健康生活	健康生活
哈特兰银行	哈特兰_银行
哈特兰加州大学	哈特兰_cu
哈特兰旅馆	哈特兰_旅馆
Heaton's	Heaton's
天堂最好	天堂_最好
天堂火腿	天堂火腿
希伯罗斯银行	Hebros_Bank
Hechalou	Hechalou
鹤城农村商业银行	合城_农村_商业银行
Hedgren	Hedgren
Heekcaa	Heekcaa
合肥科技银行	合肥_科技银行_合肥_科技银行

名称	ID
Hefest	Hefest
Hefu-Noodel	Hefu-Noodel
呵呵谷	呵呵谷
嘿	嘿
Hei Mien Tsai	Hei_mien_Tsai
Heide-Backerei Meyer	Heide-Backerei_Meyer
海德堡大众银行	Heidelberger_Volksbank
海登海默大众银行	Heidenheimer_Volksbank
高地财经	身高_财经
黑龙塘	黑龙塘
Heinen 的	Heinen 的
平和堂	平和堂
和记餐饮	和记_餐饮
合佳酒店	合家_酒店
Hekikai 信用金库	Hekikai_Shinkin_Bank
Helados Bambi	Helados_Bambi
Helados 苏丹娜	Helados_Sultana
海伦娜化学	Helena_Chemical
海伦斯	海伦斯
直升机超级市场	Heli_Super_Mart

名称	ID
赫利俄斯	赫利俄斯
赫利俄斯酒店	Helios_Hotels
Helios 手表	Helios_Watch
螺旋结构	螺旋结构
地狱披萨	Hell_Pizza
Hellanor	Hellanor
希腊邮政银行	Hellenic_Postbank
你好	你好
你好酒店	你好_酒店
你好孩子们	Hello_Kids
Hellweg	Hellweg
Helm	Helm
委内瑞拉银行Helm	委内瑞拉Helm_bank_de_
Helmisimpukka	Helmisimpukka
和龙农村商业银行	和龙农村商业银行
帮助网	Help_Net
Help-U-Sell 房地产	Help-u-Sell_Real_Estate
Helthin	Helthin
Helthin 药店	Helthin_Drugstore
Helzberg 钻石	赫尔茨伯格_钻石

名称	ID
Hemkop	Hemkop
Hemmabutikerna	Hemmabutikerna
鸡舍市场	Hen_House_Mark
HengDa 酒店和度假村	HengDa_酒店_&_度假村
恒安标准人寿	恒安_标准_生活
恒驰	恒驰
恒启教育	恒起_教育
衡山酒店及度假村	衡山_酒店及度假村
汉高粘合剂技术	汉高_粘合剂_技术
Henleys	Henleys
亨利·本德尔	Henri_Bendel
亨利·鲍彻	Henri_Boucher
亨利·劳埃德	Henri_Lloyd
Henrico FCU	Henrico_fcu
亨利市场	Henry's Market
Heraldus	Heraldus
Herbaceous 系列	草本系列
草本咖啡厅	Herbal_Cafe
康宝莱独立分销商	Herbalife_独立分销商
Herberger's	Herberger's

名称	ID
Herbária	Herbária
Herco Creaciones	Herco_Creaciones
Herfy	Herfy
Hering	Hering
遗产	遗产
Heritage	Heritage_FCU
遗产家族 CU	Heritage_family_cu
Heritage	遗产_Fresh
遗产旅馆	Heritage_Inn
遗产南加州大学	Heritage_south_cu
传统食品	遗产_食物
Hermes Fleischwaren	Hermes_Fleischwaren
爱马仕 PaketShop	爱马仕_ PaketShop
Hermes Pardini	Hermes_Pardini
Herner Sparkasse	Herner_Sparkasse
英雄	英雄
英雄 Motocorp	Hero_Motocorp
苍鹭食品	Heron_Foods
Herry	Herry
Hertitage 新鲜	Hertigage_Fresh

名称	ID
赫兹	赫兹
赫兹设备租赁	赫兹_设备_出租
Hervis	Hervis
Hesperia	Hesperia
Hess	Hess
Hess Bookstore	Hess_bookstore
赫斯快车	Hess_Expres
Het Huis	Het_Huis
HexaByte	HexaByte
Hexaware 技术	六进制软件_技术
Hexie Health	Hexie_健康_保险
嘿 Tea	Hey_Tea
Heyraud	Heyraud
Heyten	Heyten
合众口腔	合众口腔_口腔
嗨	嗨
Hi-Fi Klubben	Hi-Fi_Klubben
高寿命	高寿命
Hi-Mart	Hi-Mart
Hi-Q	Hi-Q

名称	ID
HiMountain	HiMountain
Hibachi-San	Hibachi-San
Hibernia	Hibernia
日比谷	小町托里
日比谷花坛风格	Hibya-Kadan_Style
Hichef	Hichef
Hickory Tavern	Hickory_Tavern
Hiden Hotel	Hiden_Hotel
Hidesign	Hidesign
Hidrosina	Hidrosina
海发汽油	Hifa_Petrol
东山口信用金库	东山口_信用金库
Higginbotham's	Higginbotham's
High & Migh	High_&_Mighty
高触感高科技	High_Touch_High_Tech
胜利大酒店	High_Victory_Hotel
高地	高地
高地咖啡	高地_咖啡
Highmark FCU	Highmark_fcu
公路技术	Highway _ 技术

名称	ID
HigienSec	HigienSec
希兰德	希兰德
希兰德燃料中心	Hilander_Fuel_Center
希兰德药房	希兰德_药房
希尔菲格	希尔菲格
Hillard Lyons	Hillard_Lyons
希勒市场	Hiller's Market
希尔顿	希尔顿
希尔顿花园酒店	希尔顿花园酒店
希尔顿度假大酒店	希尔顿度假大酒店
希尔顿韦纳	Hilton_Weiner
喜马偕尔邦格拉明银行	Himachal_Pradesh_Gramin_Bank
喜马偕尔邦州立合作银行	Himachal_pradesh_state_Co-op_Bank
喜马拉雅光学	Himalaya_Optical
姬路信用金库	姬路信用金库
Himo	Himo
Hinckley & Rugby 建筑协会	Hinckley_&_Rugby_Building_Society
印度斯坦汽车	印度斯坦_汽车
印度斯坦石油公司	印度斯坦_石油公司
日野	日野

名称	ID
嘻哈	Hip_Hop
HipShop	HipShop
超市中心	HiPer_Centro
Hiper bompreço	Hiper_bompreço
HiperDino	HiperDino
HiperSimply	HiperSimply
HiperSol	HiperSol
Hipercompra	Hipercompra
HiPerfarma	HiPerfarma
Hipoteku banka	Hipoteku_Banka
枚方信用金库	枚方_信用金库
平冈	平冈
平冢信用金库	平冢_信用金库
Hirepool	Hirepool
宏金证券	广银证券
广岛信用金库	广岛信用金库
广岛县信用组合	广岛县_信用合作社
广岛市信用组合	广岛市_信用合作社
海信	海信
金历史	Histoire_d'Or

名称	ID
痛苦史	Histoire_de_Pains
瑞典的历史酒店	瑞典的历史酒店
希特康汽车中心	Hitcon_Automóvil_Center
Hk Benzin	Hk_Benzin
Hl Markt	Hl_Markt
哈哈	哈哈
Helb I kifle	hleb_i_kifle
Hlobus	Hlobus
黄河手机	Hoang_Ha_Mobile
Hoang Phuc	Hoang_Phuc
Hoau	Hoau
Hoba	Hoba
霍布斯	霍布斯
HobbyTown 美国	HobbyTown_美国
流浪汉储藏室	Hobo_Pantry
Hockhua 补品	Hockhua_Tonic
霍金·斯图尔特	Hocking_Stuart
Hofer	Hofer
Hoff	Hoff
霍夫曼	霍夫曼

名称	ID
Hofmeister Brot	Hofmeister_Brot
Hofpfisterei	Hofpfisterei
Hog's Breath 咖啡馆	Hog's_breath_ca
Hogi Yogi	Hogi_Yogi
Hogl	Hogl
Hogo 颜色	Hogo_Color
HokBen	HokBen
Hokka Holka	Hokka_Hokka
北海道烤芝士塔	北海道_Baked_Cheese_Tart
北海道信用金库	北海道_信用金库
北门信用金库	Hokumon_Shinkin_Bank
北尾信用合作社	Hokuo_Credit_合作社
北莲商店	北莲店
北陆劳动银行	Hokuriku_Labour_Bank
北星信用金库	Hokusei_Shinkin_Bank
幸运的Hokuyu	Hokuyu_Luck
Hol'Ab！	Hol'Ab！
你好 Travel	你好旅行
霍尔顿	霍尔顿
Holden Hdt	Holden_Hdt

名称	ID
Holden 特种车辆	Holden_Special
假日	假日 [便利店] _12
假日	假日 [便利店] _65
节日发型	Holiday_Hair
假日酒店	Holiday_Inn
假日酒店俱乐部度假	Holiday_Inn_Club_Vacations
快捷假日酒店	快捷假日酒店
快捷假日酒店及套房	Holiday_Inn_Express_Hotel_&_Suites
花园庭院假日酒店	Holiday_Inn_Garden_Court
假日酒店及套房	Holiday_Inn_Hotel_&_Suites
假日度假酒店	Holiday_Inn_Resort
精选假日酒店	Holiday_Inn_Select
精选假日酒店及套房	Holiday_Inn_Select_&_Suites
假日 SunSpree 度假酒店	Holiday_Inn_ _Resort SunSpree
假日之星	Holiday_Star
假日车站商店	假日_车站商店
Holiland	Holiland
霍兰德和巴雷特	Holland_&_Barrett
荷兰面包店	荷兰_面包店
荷兰花卉折扣	荷兰_Blumen_Discount

名称	ID
荷兰布鲁门马克	Holland_Blumen_Mark
荷兰赌场	荷兰赌场
荷兰酒店	荷兰_酒店
Hollister	Hollister
好莱坞	好莱坞
好莱坞面包店	好莱坞面包店
好莱坞巨型购物中心	好莱坞_Megaplex
好莱坞指甲	好莱坞_钉子
好莱坞黑人	Hollywood_Tans
福尔摩斯广场	Holmes_Place
Holy Rosary CU	Holy_rosary_cu
家居与小屋	Home_&_Cottage
家居特价商品	主页_特价商品
家庭业务	主页_业务
家庭护理援助	主页_护理_援助
家居中心	主页_中心
美国家居清洁中心	Home_Cleaning_Centers_of Americ
主页颜色	主页_颜色
家居装饰师系列	家居_Decorators_系列
主页等	主页_等

名称	ID
主页” 联邦银行	主页_联邦银行
家用硬件	主页_硬件
家庭帮手	主页_Helpers
家园酒店	主页_酒店
如家旅馆	Home_Inn
改为在家老年护理	Home_Instead_Senior_Care
首页原味鸡肉	首页_原创_鸡肉
家居用品商	主页_Outfitters
家居奥特莱斯	主页_Outlet
家居升级版	主页_Plus
Home Plus	主页_Plus_Stores
家庭专业版	Home_Pro
房屋储蓄和贷款	主页_储蓄_&_贷款
家庭储蓄银行	主页_储蓄_银行
主页 Smile	主页_微笑
甜蜜的家	首页_Sweet_Home
家乡	Home_Town
国内市场专卖店	主页_市场_专卖店_专卖店
家居加快递	首页_Plus_Expres
家庭修复	家庭修复

名称	ID
Home2 套房	Home2_Suites
HomeBanc	HomeBanc
HomeGoods	HomeGoods
HomeStreet 银行	HomeStreet_银行
HomeTeam 检验服务	HomeTeam_检查_服务
HomeTown 自助餐	HomeTown_自助餐
HomeVestors	HomeVestors
Homeart	Homeart
Homebase	Homebase
家富超市	Homeful_Sum
如家Plus	如家酒店_Plus
如家酒店精选	如家酒店_精选
家园	家园
Homemart	Homemart
霍姆普雷斯旅馆	家乡_旅馆
天堂里的家	Homes_in_heaven
霍姆斯特德一室公寓套房	Homestead_Studio_Suites
家庭观察 CareGivers	Homewatch_ CareGivers
霍姆伍德套房酒店	霍姆伍德套房酒店
Homey Hotel	Home_Hotel

名称	ID
Homla	Homla
Hommerson 赌场	Hommerson_Casino's
本田	本田
Hondos 中心	Hondos_Center
蜜露甜甜圈	Honey_Dew_Donuts
蜂蜜农场	蜂蜜农场
HoneyBaked 火腿公司	HoneyBaked_Ham_Company
Honeycombe L	蜂窝_休闲
蜜月甜品	蜜月_甜品
霍尼韦尔	霍尼韦尔
中国香港	Hong_Kong
香港宽带网络	香港_宽带_网络
香港快运	香港_快递
丰隆银行	丰隆银行
丰隆伊斯兰银行	Hong_Leong_Islamic_Bank
Hong 餐厅	洪_餐厅
Hong Thai 旅行社	Hong_Thai _旅游_服务
HongZhuangYuan	HongZhuangYuan
红鼎豆老	Hongding_豆老
洪恩教育	Hongen_教育

名称	ID
Honghan 跆拳道	Honghan_跆拳道
HongkongPost	HongkongPost
鸿茂水饺	Hongmao_水饺
洪尼饮食文化	洪尼_饮食_文化
红旗超市	Hongqi_超市
Hongweiyijia	Hongweiyijia
宏信证券	宏信证券
宏运大外天	宏运_大外天
Honigman	Honigman
檀香山 FCU	Honoluulu_FCU
荣誉酒店	Honor_Hotel
Honsel	Honsel
连帽衫	连帽衫
Hoogvliet	Hoogvliet
宏达证券	宏大证券
Hooters	Hooters
跳舞	Hop_Hop
跳进去	Hop_in
HopLop	HopLop
Hopdoddy 汉堡吧	Hopdoddy_Burger_Bar

名称	ID
Hope FCU	Hope_fcu
希望面条	希望面条
Hopeshow	Hopeshow
霍普韦尔 FCU	Hopewell_fcu
贺拉斯·曼恩保险	Horace_Mann_保险
地平线经销商	地平线分销商
Horizon 分销商-	Horizon 分销商_-_弗雷斯诺
激素 Oideya	Hormone_Oideya
Hornbacher's	Hornbacher's
Horunga-Mart	Horunga-Mart
Hosler 西餐厅	Hosler_Western_Restaurn
Hospes	Hospes
美国医院公司	美国医院企业协会
卢西亚达斯医院	医院_卢西亚达斯
国际酒店业	酒店业_国际
Hosta Piknik	Hosta_Piknik
国际青年旅舍	国际青年旅舍
Hosten	Hosten
又热又脆的	Hot_&_Crispy
热面包	Hot_Breads

名称	ID
热汉堡	Hot_Burger
棍子上的热狗	hot_dog_on_a_stick
热门食物	Hot_Stuff_Food
热门披萨	热销_披萨
热门话题	热门话题
Hot&Cold	冷热两用
Hot-Star 大炸鸡	Hotstar_Large_fried_Chicken
81 号酒店	酒店_81
88 号酒店	酒店_88
贝莱尔酒店	Hotel_Belleaire
Chocolat 酒店	酒店_Chocolat
丹恩酒店	Hotel_Dann
埃斯特拉酒店	Hotel_Estelar
英迪格酒店	Hotel_Indigo
今旅酒店	Hotel_Jen
萨加尔酒店	Hotel_Sagar
桑蒂卡酒店	酒店_Santika
塞拉酒店	酒店_Sierra
斯瓦加特酒店	Hotel_Swagat
蒂米什瓦拉酒店	酒店_蒂米什瓦拉

名称	ID
州长酒店	Hotel_des_Gouverneurs
Du Vin & Bistro 酒店	Hotel_du_vin_&_Bistro
拉卡萨酒店	Hotel_la_Casa
科图尔萨酒店	酒店_科图尔萨
厄尔巴岛酒店	酒店_厄尔巴
全球酒店	酒店_Globales
加尔比酒店	酒店_Garbi
金码头酒店	酒店_Marina_d'Or
波塞冬酒店	酒店_波塞冬
Hoteles RH	酒店_RH
圣米歇尔酒店	酒店_圣米歇尔
桑托斯酒店	酒店_桑托斯
太阳之翼酒店	酒店_太阳之翼
奥尼克斯酒店	Hotels_Onix
南格兰德酒店	Hotels_du_Grand_Sud
霍滕科酒店	Hotenco_Hoteles
Hotic	Hotic
Hotlink	Hotlink
火锅故事	Hotpot_Story
Houlihan's	Houlihan's

名称	ID
房子和家居	House_&_Home
自制蛋糕	自制蛋糕
家庭医生	家庭医生
艺术之家	House_Of_Art
甜甜圈屋	House_Of_Donuts
糖果屋	House_of_Cand
马萨巴之家	House_of_masaba
油漆屋	House_of Paint
旅行之家	旅行之家
房子。	房子。
HouseMaster	HouseMaster
家庭银行	家庭银行
住房与开发银行	住房与开发银行
贸易与金融住房银行	Housing_bank_for _trade_ank_For_
休斯顿 FCU	Houston_FCU
休斯顿餐厅	休斯顿餐厅
Hoveround	Hoveround
霍维玛酒店	Hovima_Hotels
多么甜蜜	太棒了
霍华德眼镜	Howard_Eyewear

名称	ID
霍华德·汉纳房地产	Howard_Hanna_Real_Estate
霍华德·约翰逊	Howard_Johnson
霍华德存储世界	Howards_Storage_World
你好	你好
Hoyer	Hoyer
Hoyts	Hoyts
Hradecka pekarna	Hradecka_pekarna
Hranilnica LON	Hranilnica_lon
Hrant	Hrant
Hruska	Hruska
Hrvatska Lutrija	Hrvatska_Lutrija
克罗地亚邮政	Hrvatska_Posta
克罗地亚邮政莫斯塔尔	Hrvatska_posta_mostar
杨新东	杨新东
Hsv	Hsv
热门酒店	Htop_Hotels
HuHot 蒙古烧烤	HuHot_蒙古烧烤
华保险	华为保险
华南银行	华南银行
华南证券	华南证券

名称	ID
华智林	林华智
Hua's 餐厅	Hua's 餐厅
华安证券	华安证券
华晨影城	华晨_Cineplex
华晨超市	华晨_超市
华驰牙科全国	华驰牙科_全国
华创证券	华创证券
H uada-milk-tea	H uada-milk-tea
华富证券	华富证券
淮安农村商业银行	淮安农村商业银行
淮北农商银行	淮北_农村_商业银行
淮南商业银行	淮南_商业银行
怀远农村商业银行	怀远农村商业银行
华健	华健
华金证券	华金证券
华昆酒店	华昆_酒店
华联购物中心	华联商城
华联北京超市	华联-北京_超市
华联/上海华联超市	华联/上海华联_超市
华林	华林

名称	ID
华美口腔	华美口腔_牙科
欢昌KTV	Huanchang_ktv
黄成老马	黄成老马
HuangJiHuang 三酱炖锅	HuangJiHuang_Three-sauce_Simmer_Po
黄海汽车	黄海_汽车
皇佳教育	黄家_教育
黄山太平农村商业银行	黄山_太平_农村_商业银行
潢尚	潢尚
皇尚便捷超市	皇山_便利超市
皇尚超市	皇山_超市
黄山交易中心	黄山_交易中心
黄尚煌	黄尚煌
黄岩农村合作银行	黄岩_农村_合作社_银行
华普超市	华普_大卖场
华强广场	华强广场
华融香江银行	华融香江银行
华润超市	华润_超市
华生超市	华声_超级市场
华氏药业	华氏药业
华松	华松

名称	ID
华泰保险	华泰_保险
华图	华图
华为	华为
华西建设县银行	华西_建设_县_银行
华西证券	华西证券
华夏银行	华夏银行
华夏眼科医院	华夏眼科医院
华夏当铺	华夏_当铺
华讯大酒店	华讯酒店
华瑶城	华瑶城
华义	华义
华谊环球影城	华谊环球影院
华谊名都	华谊_名都
华悦百货	华悦_百货_门店
华仔沙龙	Huazai_Salon
华之友	华之友
华中成	华中成
湖北银行	湖北银行
湖北大运	湖北_大运
湖北省农村信用社	湖北_农村信用合作社

名称	ID
HubeiInsun 电影院	HubeiInsun_Cinema
湖滨农商银行	湖滨_农村_商业银行
宇舶表	宇舶表
Hubo	Hubo
Hubrol	Hubrol
Hucc-Coop	Hucc-Coop
哈克食品店	Huck's Food_store
胡达餐厅	胡达餐厅
哈德尔之家	HuddleHouse
哈德逊城储蓄银行	哈德森城储蓄银行
哈德逊河社区 CU	Hudson_river_Community_cu
哈德逊联合银行	哈德森联合银行
哈德森咖啡	哈德森咖啡
Huf-Bau	Huf-Bau
Hugendubel	Hugendubel
休斯电气	休斯_电气
Hughes FCU	Hughes_FCU
Hugo Boss	Hugo_Boss
Huguosi Cuisine	Huguosi_Cuise
虎汇超市	Huhui_超市

名称	ID
惠宁	惠宁
惠佳	惠佳
Huijia Kids	惠佳_孩子
Huimiba	Huimiba
徽商银行	徽商银行
徽商农村银行	徽商农村银行
惠友	惠友
虎津餐厅	胡金_餐厅
呼伦贝尔农村商业银行	呼伦贝尔_农村_商业银行
Humac	Humac
人力	人力
人性化	人性化
Humayuan	Humayuan
Hummel	Hummel
Humpty's	Humpty's
Humus Eliyahu	Humus_Eliyahu
湖南中餐厅	湖南_中餐厅
湖南华融农村商业银行	湖南_华融_农村_商业银行
珲春农村商业银行	Hunchun_农村_商业银行
百年教育	百次_教育

名称	ID
宏安药店	Hung-An_Drug_Store
饥饿的马	饥饿的马
Hungry Howie 的披萨	Hungry_howie's_Pizza
Hungry's	Hungry_Jack's
匈牙利 Takarek	Hungária_Takarék
Hunkemöller	Hunkemöller
Hunky Dory	Hunky_Dory
狩猎油	Hunt_Oil
狩猎和钓鱼	狩猎和钓鱼
亨廷顿学习中心	亨廷顿学习中心
霍果诗佳	Huo_Guo_Shi_Jia
Huopangzi Feichangbao	Huopangzi_Feichangbao
胡庆余堂	胡庆余堂
飓风烧烤和翅膀	Hurricane_Grill_&_Wings
Hurtan	Hurtan
Hush Puppies	Hush_Puppies
湖上乡镇银行	湖商农村银行
爱斯基摩犬	爱斯基摩犬
Hussel	Hussel
Hustler 成人用品店	Hustler_Adult_Shop

名称	ID
Husumer Volksbank	Husumer_Volksbank
Hutaoli 音乐餐厅和酒吧	Hutaoli_Music_Restaurant_Bar
哈钦教育	哈钦_教育
和记电信	Hutchison_Telecom
胡同	胡同
Huttopia	Huttopia
Hvb Bank	Hvb_Bank
华泰银行	华泰银行
华宝证券	华宝_证券
Hwy 55 Burgers Shakes & Fries	Hwy_55_Burgers_Shakes_&_Fries
Hy-Vee	Hy-Vee
Hy-Vee 市场格栅快车	Hy-vee_Market_Grille_Expr
凯悦嘉轩酒店	Hyatt_Place
凯悦酒店	凯悦_丽晶酒店
凯悦公寓俱乐部	凯悦公寓俱乐部酒店
凯悦度假村	凯悦度假村
海得拉巴区中央合作银行	Hyderabad_dist_central_Co-op_Bank
水电	水电
德士古水电	Hydro_德士古
赞美诗摄影工作室	赞美诗_照片_工作室

名称	ID
兵库信用金库	兵库_信用金库
兵库县信用合作社	兵库县_信用合作社
晓星	晓星
Hype DC	hype_dc
Hyper U	Hyper_U
希波利翁	希波利翁
超级市场	超级市场
超新星	超新星
超级鞋	超级鞋
超级巨星	超级巨星
Hypo	Hypo
Hypotecni banka	Hypotecni_Banka
Hypowereinsbank	Hypowereinsbank
Hysteric Glam	hysteric_Glam
现代 AutoEver	现代_ AutoEver
现代工程公司	现代_工程
现代格洛维斯	现代_Glovis
现代 KEFICO	现代_kefico
现代摩比斯	现代_摩比斯
现代汽车公司	现代汽车公司

名称	ID
现代石油银行	现代_Oilbank
现代 Rotem	Hyundai_Rotem
现代制铁	现代_钢铁
现代坦迪斯	现代_Transys
现代威亚	现代_威亚
哈达	哈达
Hastens	Hastens
巴里埃尔酒店	Hotels_Barrière
Höffner	Höffner
Höks HESTSPORT	Höks_hastSport
Höyer	Höyer
我能读	我能读懂
我简直不敢相信这是酸奶	我_简直不敢相信_这是酸奶
我自己动手 Robotec	我_DIY_Robotec
我愿意	我愿意
我停车	我_我_停车
我喜欢 10 欧元	我_love_10_euros
我喜欢 Yoo	我_Love_Yoo
I Partea	我_Partea
我是	我_am

名称	ID
我有房子	i_House
I&M 银行	I&M_Bank
I-Ball	I-Ball
我租房	我租房
I.P.ZONE	I.P.ZONE
I.T	I.T
爱荷华州维亚耶斯	ia_viajes
IBA-莫斯科	IBA-莫斯科
IBACCO	IBACCO
IBC银行	IBC_Bank
IBERDOEX	IBERDOEX
IBERDOEX 加油站	Iberdoex_Petrol_Station
伊比利亚银行	伊比利亚银行
IBEROSTAR	IBEROSTAR
IBEU	IBEU
IBL 银行	IBL_BANK
IBP	IBP
IBUSZ	IBUSZ
ICA	ICA
ICA 银行	ICA_Bank

名称	ID
ICA Kvantum	ica_kvantum
ICA Maxi	ica_maxi
ICA 超市	ica_Sump
工商银行	工商银行
ICE 设计	ice_Design
ICHIBENOR	ICHIBENOR
ICICI 银行	ICICI_Bank
艾达	艾达
想法	想法
额外创意	Idea_Ex
超级创意	idea_sup
IDER	IDER
IDFC 第一银行	IDFC_First_Bank
IDS	IDS
IENA	IENA
是	是
如果巴黎 RIVE GAUCHE	IF_PARIS_RIVE_GAUCHE
IFA	IFA
IFA 酒店	ifa_Hotel
IFB 积分	ifb_Points

名称	ID
IFK	IFK
伊加	伊加
IGA 白酒	Iga_Liquir
IGS	IGS
IH 信用合作社	ih_credit_Uni
IHOP	IHOP
IIFL 证券	IIFL_证券
IIHT	IIHT
III Forks	iii_forks
IIII Intermoda	iiii_intermoda
宜家	宜家
IKILER	IKILER
哎呀	哎呀
IKON 办公解决方案	ikon_Office_解决方案
IL 露台	IL_Patio
美丽岛	ILE_DE_BEAUTE
ILEY	ILEY
成像仪	成像仪
IMAX	IMAX
IMB	IMB

名称	ID
IMEI	IMEI
帝国停车	帝国停车
重要度假村	重要度假村
冲动完全健身	冲动_全面_健身
IMS	IMS
伊娜	伊娜
INATEL	INATEL
INCANTO	INCANTO
INCITY	INCITY
INDECO	INDECO
印度花园	印度花园
印度厨房	印度厨房
靛青	靛青
INDISKA	INDISKA
兴业银行	工业银行
工业证券	工业证券
英菲尼迪	英菲尼迪
ING	ING
荷兰国际集团银行	ing_bank
直接拨打	ING_DRICT

名称	ID
荷兰国际集团-DiBa	荷兰国际集团-DiBa
INGLOT	INGLOT
INGO	INGO
INKA	INKA
INKILAP KITAPEVI	INKILAP_KITAPEVI
INLAB 实验室	inlab_laboratório
INNO GALERIA	INNO_GALERIA
INSIDE BY MELIA	INSIDE_BY_MELIA
INOV	INOV
INOVA FCU	INOVA_FCU
inova-Chiphone FCU 合作处	inova-chiphone_FCU_Partnership_B
INOX	INOX
INSOTEL	INSOTEL
INTERINA	INTERINA
通过装饰书房进行室内装饰	interiors_by_decorating_den
INTERMARCHÉ	INTERMARCHÉ
混合	混合
Interni Armarios	interni_armarios
进步银行	进步银行
鞋间	鞋间

名称	ID
SPAR	SPAR
INTERSPAR RESTAVRACIJA	INTERSPAR_RESTAVRACIJA
运动间	运动间
INTESA SANPAOLO	INTESA_SANPAOLO
银泰百货	INTIME_百货商店
INVITRO	INVITRO
INXX	INXX
INZONE	在区域
ION	ION
离子性	离子性
IP	IP
IP MATIC	IP_MATIC
IPO 汽车美容中心	IPO_CAR_BEAUTY_CENTER
IPT	IPT
IQ	IQ
IQ 007	IQ_007
IQ 玩具	IQ_TOY
是咖啡	IS_COFFEE
ISA	ISA
伊莎贝尔	伊莎贝尔

名称	ID
伊势丹	伊勢丹
ISKU	ISKU
泰国伊斯兰银行	泰国伊斯兰银行
伊斯兰国际阿拉伯银行	伊斯兰国际_阿拉伯_银行
ISSIMO	ISSIMO
ISTARSKA CREDITNA BANKA UMAG	ISTARSKA_KREDITNA_BANKA_UMAG
五十铃	五十铃
IT CITY	IT_CITY
IT NIOS	IT_NIOS
板长寿司	ITACHO_SUSHI
ITAKA	ITAKA
意大利的热门家庭酒店	意大利_家庭_酒店
板前寿司	板前寿司
ITM	ITM
ITTO BILDIQUIDE	ITTO_BILLINIQUIDE
ITV	ITV
IU	IU
IU 信用合作社	iu_credit_Uni
IU 酒店	iu_Hotel
IUSACELL	IUSACELL

名称	ID
IVB	IVB
常春藤屋	IVY_HOUSE
万国表	万国表
IZB	IZB
IZELMAN	IZELMAN
IZIBUILD	IZIBUILD
艾德花园	Iade_Garden
我是 gym	我是 gym
茨城县信用合作社	Ibaraki-ken_credit_合作社
Iberdrola	Iberdrola
Iberent	Iberent
伊伯索尔	伊伯索尔
伊比尔	伊比尔
宜必思尚品酒店	宜必思尚品酒店
Ibox 银行	Ibox_Bank
Ibérica de Luminacion	Ibérica_de_iluminacion
Ic Bank	Ic_Bank
Ica Bensin	Ica_Bensin
Ica Neara	Ica_Neara
工商银行-安盛人寿	工商银行-Axa_Life

名称	ID
冰淇淋秀	Ice_Cream_Show
小奈的冰茶	Xiao_nai 的冰茶
IceCoBar	IceCoBar
IceMellow	IceMellow
冰山	冰山
冰岛	冰岛
Icha Icha	Icha_Icha
Ichiban	Ichiban
一番拉面	一番_拉面
一番寿司	一番_寿司
Ichido	Ichido
Ichier 牙科诊所	Ichier_Dental_Clinic
一护馆	一护馆
市井信用金库	Ichii_Shinkin_Bank
一兰	一兰
Ichiroku Hompo	Ichiroku_Hompo
Ici Paris	ici_paris_XL
Iciar de la Concha	iciar_de_la_concha
结冰	结冰
ico	ico

名称	ID
Iconinc	Iconinc
创意	创意
伊德比银行	Idbi_Bank
想法库	理念_银行
创意制造者	创意制造者
理想的 CU	理想_cu
理想的电气供应商	理想的_电气_供应商
理想	理想
想法	想法
出光兴产	Idemitsu_Kosan
Idemöbler	Idemöbler
Ident-AKid	Ident-AKid
Ideya 银行	Ideya_Bank
瑞士田园诗酒店	Idyll_Hotels_Schweiz
ifnp	ifnp
伊格鲁	伊格鲁
伊瓜特米	伊瓜特米
Igym	Igym
Ihlas	Ihlas
Ihlas 家居专卖店	Ihlas_Home_Specialty_Store

名称	ID
Ihr Platz	Ihr_Platz
Iib 银行	IIB_Bank
饭田信用金库	饭田信用金库
饭冢信用金库	Iizuka_Shinkin_Bank
伊乔伊酒店	Ijoy_Hotel
Ikari	Ikari
伊卡里咖啡	伊卡里_咖啡
伊克巴尔	伊克巴尔
艾克的	艾克的
Iki	Iki
Ikidz 英语学校	ikidz_school_of_English
Ikkenme Sakaba	Ikkenme_Sakaba
Ikkousha	Ikkousha
Ikkyu	Ikkyu
伊克利尔	伊克利尔
Iklimsa	Iklimsa
Il Fornaio	Il_Fornaio
Il Makiage	Il_Makiage
伊拉里亚	伊拉里亚
Ilektroniki	Ilektroniki

名称	ID
Ilha 3	Ilha_3
蓝岛	Ilha_Azul
青岛	Ilha_Verde
Illik pekarstvi	illik_pekarstVI
我爱健身	我爱健身
Ilusion	Ilusion
伊玛	伊玛
图像	图像
幻影馆	幻影馆
想象一下家	Imagine_Home
Imagix	Imagix
Imc	Imc
Imeksbank	Imeksbank
imerco	imerco
Imex banka	imex_Banka
Imly	Imly
沉浸式酒店及度假村	沉浸式_酒店及度假村
Imo 的披萨	imo's_Pizza
Impark	Impark
帝国的	帝国的

名称	ID
帝国银行	帝国银行
产品帝国	帝国_产品
Imperiya Sumok	Imperiya_Sumok
Inplozia	Inplozia
Império Bonanca	Império_Bonanca
In & Out	In_&Out
在操场上	在游乐场里
In Time	In_Time
In-N-Out 汉堡	In-N-Out_Burger
InFirst FCU	InFirst_FCU
InPost	InPost
InTown 套房	InTown_套房
稻本寺	稻本寺
Inbursa	Inbursa
Inc. 5	Inc._5
Incenta a Pathways Div of Pat	incenta_a_div_of_pathways_financial_cu
Inchin 的竹园	inchin's Bamboo_Garden
Inci	Inci
Incommons银行	Incommons_Ban
不可思议的连接	不可思议的连接

名称	ID
Ind'l 和 com'l 中国银行	ind'l_and_com'l_bank_of_China
Indalo Tapas	Indalo_Tapas
英达拉保险	Indara_保险
Indeks-Bank	Indeks-Bank
AMSOIL 独立经销商	独立_amsoil_经销商
独立银行	独立银行
独立楼层	独立楼层
独立的 Kirby 分销商	独立_Kirby_分销商
沃特金斯独立合伙人	独立_Watkins_Associate
独立人士	独立人士
Index 生活购物中心	首页_生活购物中心
印度邮政支付银行	印度_Post_Payments_银行
印度风格	印度风格
印度汽车天然气	印度_汽车_天然气
印度银行	印度银行
印度咖啡屋	印度咖啡屋
Indian Health	印度健康服务
印度餐 Thali-ya	印度餐_Thali-ya
印度石油	印度_石油
印度海外银行	印度_海外_银行

名称	ID
印度地形	印度地形
印第安纳州会员信用合作社	印第安纳州_会员_信用联盟
印度农场	Indo_Farm
Indomaret	Indomaret
室内	室内
因陀罗	因陀罗
Indraprastha Gas	Indraprasta_Gas
Indue	Indue
印度河	印度河
印度工业银行	印度工业银行
韩国兴业银行	韩国工业银行
工业信用合作社	工业信贷联盟
工业银行	工业银行
工业	工业
工业孩子	工业_童装
Ineng 的特别烧烤	ineng's_Special_BBQ
无限	无限
无限 FCU	infinity_fcu
Infor	Infor
Infosys	Infosys

名称	ID
Infuze CU	infuze_cu
Ing Bank Slaski	Ing_Bank_Slaski
Ing Vysya	Ing_Vysya
英格尔斯天然气快车	英格尔斯_Gas_Express
英格尔斯市场	英格尔斯_市场
英格尔斯药房	英语_药房
Ingosstrakh	Ingosstrakh
英格拉姆微型	Ingram_Micro
InkaFarma	InkaFarma
Inkley 的相机和图像	Inkley's_camera_&_image
内陆食品商店	Inland_Food_Store
内陆石油	内陆石油公司
Inlove	Inlove
Inmarko	Inmarko
Inmedio	Inmedio
客栈套房酒店	旅馆_套房_酒店
湾港旅馆	亲笔签名集
InnSunrise	InnSunrise
内蒙古伊金霍洛银行	内蒙古_伊金霍洛_银行
悦诗风吟	悦诗风吟

名称	ID
旅店老板	旅店老板
Inno	Inno
Innovatur	Innovatur
美国旅馆	美国因斯_of_america
Inokom	Inokom
Inova Drogarias	Inova_Drogarias
Inprombank	Inprombank
Insail 酒店	Insail_Hotels
里面	里面
Insinuante	Insinuante
失眠咖啡	失眠_咖啡
失眠饼干	Insomnia_Cookies
Inspect-It 一	inspect-it_1st
灵感	灵感
Instaloan	Instaloan
即时印记	即时印记
即时打印	Insty_Prin
Insureone 保险公司	Insureone_Insurance_Agen
Inta Juice	Inta_Juice
Integra银行	整合银行

名称	ID
Integramedica	Integramedica
Intehbank	Intehbank
智能办公	智能办公
国际米兰	Inter_Dan
Inter'ernaya Lavka	inter'ernaya_lavka
欧洲际酒店	欧洲际酒店_酒店
欧洲间银行	欧洲间银行
因特尔酒店	因特尔酒店
InterCity	InterCity
InterCityHotel	InterCityHotel
InterContinental	InterContinental
InterPark	InterPark
银行同业	银行同业
布隆迪银行同业拆借银行	银行同业_布隆迪
银行同业经纪人	银行同业经纪人
Intercare	Intercare
陶瓷间	陶瓷间
交汇处	交汇处
Intercommerz	Intercommerz
洲际银行	洲际银行_银行

名称	ID
折扣间隔	折扣间隔
界面金融集团	接口_金融_集团
花间植物	花间植物
Interforma	Interforma
Interfunk	Interfunk
临时 HealthCare	中期_ HealthCare
Interior He	Interior_Health
Interlingua 学校	国际语言学校
Intermarché 联系方式	Intermarché_contact
国际市场快车	Intermarché_Express
Intermarché HYPER	intermarche_Hyper
Intermarché 地点	Intermarché_Location
Intermarché Petrol	Intermarche_Petrol
Intermarché SUPER	intermarche_Super
山间医疗保健	山间医疗保健
Intermède	Intermède
Internafta	Internafta
美国国税局	内部收入服务
国际	国际
国际资产银行	国际资产银行

名称	ID
国际电气工人兄弟会	国际电气工人兄弟会
国际纸业公司	国际_Paper_Co
国际青年旅舍	国际青年旅舍
Interoil	Interoil
Interoptik	Interoptik
国际停车	国际停车
Interpetro	Interpetro
Interquest 探测犬	Interquest_detection_Canines
Interra CU	interra_cu
因特网	因特网
州际电池	州际电池
互教	互教
Intertop	Intertop
国际玩具展	国际玩具展
Intertur 酒店	Intertur_Hotels
面试	面试
Inthanin 咖啡	Inthanin_coffee
Intima	Intima
Intima Secret-Lili Pink	Intima_Secret-Lili_Pink
Intratuin	Intratuin

名称	ID
Intropia	Intropia
Intrust 银行	Intrust_Bank
Inturotel	Inturotel
Inul Vizta	Inul_Vizta
投资	投资
投资银行	投资银行
InvestKapitalBank	InvestKapitalBank
Investacor	Investacor
投资银行	投资银行
投资者银行	投资者_银行
投资者诊所	投资者_诊所
投资信托银行	Investrust_Bank
Investorgbank	Investorgbank
因维萨酒店	Invisa_Hoteles
邀请	邀请
Invivo	Invivo
Invázio	Invázio
井尾信用金库	Io_Shinkin_Bank
爱荷华大学	爱荷华大学
Ipca 实验室	Ipca_实验室

名称	ID
Ipek	Ipek
Ipekyol	Ipekyol
伊皮兰加	伊皮兰加
Ipoint	Ipoint
一风堂	一风堂
Irbis	Irbis
艾里斯女孩	Iris_Girls
爱尔兰人寿和永久	爱尔兰人生活和永久
爱尔兰全国建筑协会	爱尔兰全国建筑协会
Irkutskaya apteka	irkutskaya_apteka
Irma	Irma
铁山	铁山
铁石银行	Ironstone_Bank
欧文	欧文
是银行	Is_Bank
艾萨克著名的烤三明治	Isaac's Famous_grilled_Sandwiches
伊莎多拉	伊莎多拉
Isbir Yatak	Isbir_Yatak
我看见孩子们	我看见孩子们
石井体育	石井_体育

名称	ID
石烧牛排 Zei	Ishiyaki_Steak_Zei
Ishka	Ishka
Iskamed	Iskamed
伊斯兰开发银行	伊斯兰发展银行
岛上 FCU	Island_fcu
岛屿喷墨机	岛_Ink-Jet
Islands	Islands
Isler Kitapevleri	Isler_Kitapevleri
伊斯马尔	伊斯马尔
Iso	Iso
Ispark	Ispark
以色列邮报	以色列邮报
Issac's Deli	issac's_deli
Isseg Farmacias	Isseg_Farmacias
一肖尼寿喜烧	Isshoni_sukiyaki
伊斯坦布尔赛亚哈特	Istanbul_Seyahat
Istarski 超市	Istarski_超级市场
Istikbal	Istikbal
Istrobanka	Istrobanka
It Oil	It_Oil

名称	ID
这是一场磨难	这是 _a_grind
只是午餐	这只是午餐
是皮肤	它是 _skin
板垣市	板垣市
Itaka Nedvizhimost'	Itaka_Nedvizhimost'
italclean	italclean
意大利餐饮 DonA	意大利_餐饮_Dona
意大利披萨	意大利_披萨
Italianni	Italianni
伊塔普阿	伊塔普阿
伊塔乌	伊塔乌
Itau Personalité	Itau_Personalité
Itc	Itc
Ithmaar 银行	Ithmaar_Bank
Ithmaar Holding B.S.C.	Ithmaar_Holding_B.S.C.
伊藤洋华堂	伊藤洋华堂
Itoku	Itoku
伊万	伊万
Ivar's	Ivar's
依维柯	依维柯

名称	ID
伊冯娜	伊冯娜
常春藤幼儿园	常春藤幼儿园
岩城	岩城
Ixe	Ixe
Ixora	Ixora
Iyo Seimen	Iyo_Seimen
Izbyonka	Izbyonka
Izhkombank	Izhkombank
伊佐德	伊佐德
伊佐德奥特莱斯	Izod_Outlet
J·亚历山大的	J_Alexander
J Crew	J_Crew
J Jill	J_Jill
J Sainsbury	J_Sainsbury
J Young	J_Young
J	J_
强生食品	J&J_Foods
J&T 快车	J&TExpress
J'CODE	J'CODE
J. Alves	J._Alves

名称	ID
J·布尔纳索斯	J._Bournazos
J·克里斯托弗的	J._Christoph
J.D. Byrider	J._D._Byrider
J.A. Russell	J.A_Russell
J.Co 甜甜圈和咖啡	J.Co_Donuts_&_Coffee
J.G.T.	J.G.T.
JA	JA_
JA-SS	JA-SS
江淮汽车	JAC_Motors
杰克和琼斯	JACK_&_JONES
JACK&MAGIC PE	JACK&MAGIC_PEA
JACK'S 油漆和五金件	JACK'S PAINT_&_硬件
成龙电影院	JACKIE_CHAN_CINEMA
翡翠花园	JADE_GARDEN
JADRANSKA BANKA	JADRANSKA_BANKA
JAEGER-LECOULTRE	JAEGER-LECOULTRE
JAF 公路服务	Jaf_Road_Service
JAG Club	Jag_club
美洲虎	美洲虎
JAM	JAM

名称	ID
JANDAIA	JANDAIA
日本 C.R.C.	JAPAN_C.R.C.
日本家庭中心	日本家居中心
日本医疗	日本_医疗
贾里尔书店	JARIR_BOOKSTORE
杰森	杰森
杰森斯	杰森斯
JASONS 集市	Jasons_market_Plac
JASPAL	JASPAL
贾斯珀	贾斯珀
JATOMI 健身	JATOMI_FITNESS
JAX FCU	JAX_FCU
JAXQuickfit 轮胎	JAXQuickfit_轮胎
爵士生活俱乐部	爵士生活俱乐部
JA	JA
JB HiFi	JB_ HiFi
JB Sports	JB_Sports
JB 的餐厅	JB's_Restaurant
JC	JC
JCB	JCB

名称	ID
JCPenney	JCPenney
JCPenney 家居摆设	JCPenney_家居_家具
京东景车汇	JD_jingcheHui
京东体育	JD_Sports
JD Wetherspoo	JD_Wetherspoon
让-克洛德·比吉纳	JEAN-CLAUDE_BIGUIN
牛仔裤和夹克	牛仔裤_&_夹克
牛仔裤褶边	Jeans_Fritz
JENSENS BÖFHUS	JENSENS_BÖFHUS
JERUK MADU PAK ALI	JERUK_MADU_PAK_ALI
喷气式飞机	喷气式飞机
JET 服装店	Jet_clothing_store
JETPET	JETPET
JETTI	JETTI
JH 学院	JH_Academy
朱健	朱健
贾阳吉	JIA_YANG_JI
佳华商场	JIAHUA_EMPORIUM
佳慧	佳慧
嘉伦宾馆	嘉璐安酒店

名称	ID
JIANG 鱼丸	江鱼丸
嘉泰连锁商务酒店	嘉泰连锁商务酒店
佳怡快递	佳易快递
悬臂	悬臂
捷佳咖啡	捷佳咖啡
吉米钥匙	JIMMY_KEY
金百丸	JIN_BAI_WAN
金发佳	JIN_FA_JIA
JIN JIN WOK	JIN_JIN_WOK
金老汉	JIN_LAO_HAN
JIN MAN LOU	JIN_MAN_LOU
金茂机械与五金	金茂_机械_&_硬件
JIN YUH TARNG	JIN_YUH_TARNG
JINS	JINS
JINTONE 酒店	JINTONE_HOTEL
JIO GROUP	JIO_GROUP
JIP	JIP
吉泰酒店	吉泰_酒店
JJB Sports	jjb_Sports
JK	JK

名称	ID
JK哈里斯	JK_Harris
JK 轮胎	JK_Tyres
JLD Dragon	JLD_Dragon
JLL	JLL
JMCH	JMCH
JMMC	JMMC
JNBY	JNBY
JNE	JNE
JOA	JOA
琼	琼
JOANN 面料和手工艺品商店	Joann_Fabric_and_craft_stores
JOAQUIM CHAVES	JOAQUIM_CHAVES
JOE 与果汁	JOE_&_THE_JUICE
约翰·克雷格	约翰·克雷格
约翰·约翰	约翰_约翰
约翰玩家	约翰_玩家
约翰逊	约翰逊
JOKER 专卖店	Joker_Specialty_store
JOM	JOM
JONAK	JONAK

名称	ID
JOOIL	JOOIL
JOOIL 市场	jooil_Market
哈哈哈！	哈哈哈！
约旦科威特银行	约旦_科威特_银行
JORYA	JORYA
期刊标准	期刊_标准
欢乐管理	欢乐管理
欢乐来电牛排	JOYCALLER_STEA
乔伊斯	乔伊斯
JOYINN	JOYINN
JOYJO	JOYJO
乔伊玛特	乔伊玛特
JOYSOUND	JOYSOUND
JPMorgan Chase	JPMorgan_Chase
JRC GameCentrum	JRC_GameCentrum
JRS 快车	JRS_Expres
JS 银行	JS_Bank
JS 酒店	JS_Hotels
JSF	JSF
JSP	JSP

名称	ID
JT	JT
JUDEHUATIAN	JUDEHUATIAN
聚和酒店	JUHE_HOTEL
果汁和茶店	JUICE_&_TEA_SHOP
果汁起作用了	Juice_works
JULES	JULES
巨型电子设备	巨型电子
珍宝专卖店	jumbo_Specialty_stor
UMPIN GYM U.S.A.	JUMPIN_GYM_US.A.
JUN CHA	JUN_CHA
JUN 男士	Jun_men
军门	军门
贾斯汀	贾斯汀
JUTECO	JUTECO
JW 万豪酒店	JW_Marriott
JWOS	JWOS
JYSK	JYSK
JZ 证券	JZ_Securities
Jac	Jac
杰克·阿斯特的	Jack_Astor's

名称	ID
杰克·高恩房地产经纪人	Jack_Gaughen_Realtor
杰克·沃尔夫斯金	杰克·沃尔夫斯金
杰克·沃尔夫斯金商店	Jack_Wolfskin_Stor
盒子里的杰克	Jack_in_the_box
Jack's	Jack's
杰克的家庭餐厅	Jack's 家庭_餐厅
大奖	大奖
杰克逊地区 FCU	Jackson_area_FCU
杰克逊·休伊特税务局	Jackson_Hewitt_Tax_Serv
杰克森斯	杰克森斯
杰奎琳·里乌	Jacqueline_Riu
雅克·维特	Jacques_Vert
Jacqui-E	Jacqui-E
翡翠料理	Jade_Cuisine
Jaeger	Jaeger
Jaeger 女装	Jaeger_Ladieswear
Jaeger Man	Jaeger_Man
Jaez Express 酒店	Jaez_Express_Hotel
Jafs	Jafs
Jag	Jag

名称	ID
Jager	Jager
Jahonts	Jahonts
Jahra 合作社协会	Jahra_Co-Operative_Societ
Jai Laxmi Bank	Jai_Laxmi_Bank
入狱	入狱
斋浦尔中央合作银行	Jaipur_Central_Co-operative_Bank
贾卡门	贾卡门
Jalgaon Dist 中央合作银行	Jalgaon_dist_central_Co-op_Bank
Jalgaon Janata Sahakari Bank	Jalgaon_Janata_Sahakari_Bank
牙买加蓝	牙买加_蓝色
牙买加和印度咖啡	Jamaica_y_café_de_las_Indias
Jamba	Jamba
詹姆斯·康尼岛	詹姆斯_康尼岛
詹姆斯·乔纳	詹姆斯_乔纳森
詹姆斯·乔伊斯·科菲特尔	James_Joyce_coffetel
詹姆森旅馆	Jameson_Inn
Jamie 的意大利语	Jamie's _Italian
杰米波拉银行	Jami_Bora_Bank
查谟和克什米尔银行	Jammu_&_Kashmir_Bank
简·伯格	Jan_Burger

名称	ID
扬·林德斯	Jan_Linders
Jan-Pro	Jan-Pro
Jana 小型金融银行	Jana_Small Finance_Bank
Janakalyan Sahakari Bank	Janakalyan_Sahakari_Bank
Janalaxmi 合作银行	janalaxmi_co-op_Bank
Janaseva Sahakari 银行	Janaseva_Sahakari_Bank
Janata Sahakari 银行	Janata_Sahakari_Bank
Jane Eyre	Jane_Eyre
Jani-King	Jani-King
珍妮和杰克	Janie_and_Jack
Jankalyan Sahakari Bank	Jankalyan_Sahakari_Bank
Janney	Janney
Jansahakari 银行	Jansahakari_Bank
Jantize	Jantize
日本主页	日本_主页
日本肉食清泉馆	日本_肉类_Seisenkan
日本照片	日本_照片
日本租车	日本_租车
日本亭	Japan_Tei
日本船	日本船

名称	ID
Japanika	Japanika
Jaquar	Jaquar
Jardiland	Jardiland
Jared 珠宝画廊	Jared_the_galleria_of Jewelry
贾维斯	贾维斯
Jasman	Jasman
Jasmi's	Jasmi's
茉莉	茉莉
茉莉花盛开旅馆	Jasmine_Bloom_inn
Jason's Deli	Jason's deli
爪哇城	Java_City
哈维尔·西莫拉	Javier_Simorra
Jawed Habib 美发和美容沙龙	Jawed_Habib_Hair_&_Beauty_Salons
Jay C	Jay_C
Jay C 燃料中心	Jay_C_Fuel_Center
Jay C 药房	Jay_C_Pharmacy
Jay Jays	Jay_Jays
贾亚拉杰汽车局	Jayaraj_汽车代理商
Jaycar	Jaycar
Jaymart	Jaymart

名称	ID
佳宇国际影院	Jayu_国际_影院
爵士乐	爵士乐
JazzCash	JazzCash
爵士乐运动	爵士乐运动
爵士披萨	爵士乐_披萨
Jau Serve 超级市场	Jaú_Serve_Supermercados
Jea Tzoei Wang	Jea_Tzoei_Wang
Jean Coutu	Jean_Coutu
Jean Delatour	Jean_Delatour
让·路易斯·大卫	Jean_Louis_David
让·路易丝·大卫	Jean_Louise_David
让-马克·马尼亚蒂斯	Jean-Marc_Maniatis
牛仔裤中心	Jeans_Center
牛仔裤俱乐部	Jeans_Club
牛仔裤实验室	Jeans_Lab
牛仔裤仓库	牛仔裤_仓库
牛仔裤装	牛仔裤_Wear
Jeanswest	Jeanswest
吉普	吉普
吉普服装	Jeep_Apparel

名称	ID
吉普乡村 FCU	jeep_country_FCU
吉普五星	Jeep_Five_Star
杰夫·德·布鲁日	jeff_de_Bruges
杰弗牛排	杰弗牛排
杰斐逊金融大学	Jefferson_Financial_cu
杰斐逊飞行员人寿保险公司	Jefferson_Pilot_Life_Ins_Co
杰斐逊大学医院	杰斐逊大学_医院
耶和华见证人	耶和华见证人
济州银行	济州银行
Jeld Wen	Jeld_Wen
Jello	Jello
果冻兔子	Jelly_Bunny
钟仁聪	钟振聪
Jeng Jong	Jeng_Jong
珍妮·巴顿	Jenni_Button
詹妮弗敞篷车	詹妮弗_敞篷车
珍妮·克雷格减肥中心	Jenny_Craig_Weight_Loss_Center
Jennyfer	Jennyfer
Jeno 的披萨	Jeno's_Pizza
全北银行	全北银行

名称	ID
Jeordie's	Jeordie's
杰拉的果汁	Jera's_juice
Jerail	Jerail
耶利米的意大利冰	Jeremiah's_italian_ice
Jernia	Jernia
Jerns	Jerns
Jeronymo	Jeronymo
杰里学校	Jerry_School
Jerry 的订阅和披萨	Jerry's_subs_&_Pizza
泽西岛	泽西岛
泽西岛 Mike's Subs	Jersey_mike's_Subs
泽西岛海岸 FCU	Jersey_shore_fcu
Jeshing	Jeshing
Jespac	Jespac
小杰斯珀	Jesper_Junio
杰西卡 McClintock	Jessica_ McClintock
杰索普斯	杰索普斯
Jesters	Jesters
喷气食品店	Jet_Food_Shop
Jet Food 商店	Jet_Food_Stores

名称	ID
捷特玛特	Jet_Mart
杰特的披萨	Jet's Pizza
Jet-Black	Jet-Black
JetStream FCU	JetStream_FCU
捷泰航空	捷泰航空
Jetoil	Jetoil
捷达	捷达
喷气式飞机	喷气式飞机
Jetty Surf	Jetty_Surf
Jewel-Osco	Jewel-Osco
珠宝设计公司	珠宝_设计_公司
Jexer	Jexer
贾坎德邦格拉明银行	Jharkhand_Gramin_Bank
吉光美味炸鸡	Ji_Guang_Delicious_Fried_Chicken
贾贝	Jia_Bei
贾乐元	Jia_le_Yuan
王佳	王佳
JiaJiaYue	JiaJiaYue
JiaLiHua 旅馆和酒店	JiaLiHua_旅馆_&_酒店
JiaYi 酒店	JiaYi_酒店

名称	ID
JiaYiHotel	JiaYiHotel
Giabian Go	Jiabian_Go
佳得利超市	家得利_超市
Jiaheyipin	Jiaheyipin
佳会教育	佳辉_教育
佳吉快递	佳极速运
佳佳吉	佳佳吉
佳乐佳	佳乐佳
佳丽	佳丽
佳美口腔	佳美口腔_牙科
佳梦	佳梦
吉安玛特	吉安玛特
江边城外巫山 Kaoquanyu	江边_成外_巫山_Kaoquanyu
江川友	江川友
江都农村商业银行	江都_农村_商业银行
江海证券	江海_证券
江杰	江杰
傅江南	Jiannan_Fu
江南农村商业银行	江南_农村_商业银行
江苏滨海农村商业银行	江苏_滨海_农村通信_银行

名称	ID
江苏长江商业银行	江苏_长江_商业银行
江苏大丰农村商业银行	江苏_大丰_农村通信_银行
江苏富宁农村商业银行	江苏_富宁_农村通信_银行
江苏高淳农村商业银行	江苏_高春_农村通信_银行
江苏高邮农村商业银行	江苏_高邮_农村通信_银行
江苏靖江农村商业银行	江苏_靖江_农村通信_银行
江苏句容农村商业银行	江苏_句容_农村通信_银行
江苏涟水农村商业银行	江苏_涟水_农村_商业银行
江苏民丰农村商业银行	江苏_民丰_农村通信_银行
江苏濮州农村商业银行	江苏_濮州_农村通信_银行
江苏如皋农村商业银行	江苏_如皋_农村_商业银行
江苏射阳农村商业银行	江苏_射阳_农村通信_银行
江苏泗洪农村商业银行	江苏_思宏_农村通信_银行
江苏泗阳农村商业银行	江苏_泗阳_农村通信_银行
江苏信义农村商业银行	江苏_信义_农村_商业银行
江苏叙义农村商业银行	江苏_徐义_农村_商业银行
江苏仪征包商农村银行	江苏_仪征_包商_农村银行
姜堰农村商业银行	姜堰_农村_商业银行
建湖农商银行	建湖_农村_商业银行
健民牙科诊所	健敏牙科诊所

名称	ID
健民大药房	健民药业_药店
娇龙豆老	Jiolong_doulao
嘉泰乐	嘉泰乐
吉比	吉比
Jidori Semmonten 居酒屋 Jidokko Kumiai	Jidori_Semmonten_Isakaya_Jidokko_Kumiai
强杰	Jie_Qiang
Jiffy	Jiffy
Jiffy Lube	Jiffy_Lube
Jiffy Sto	Jiffy_Stop
Jigsaw	Jigsaw
日盛国际银行	日盛国际银行
Jika Udon	Jika_Udon
吉拉·加兹阿巴德银行	Jila_Ghaziabad_Bank
吉林春城农村商业银行	吉林_春城_农村_商业银行
吉林龙井农村商业银行	吉林_龙井_农村_商业银行
吉林省农村信用社	吉林_省_农村_信用社
吉林九台农村商业银行	吉林九台农村商业银行
Jim & Jill	Jim_&_Jill
Jim 'N Nick's Bar-B-Q	jim_'n_nick's_bar-b-q
Jim's Resturants	Jim's Resturants

名称	ID
Jimailong	Jimailong
Jimboy's Tacos	Jimboy's_tacos
吉米格兰特斯	吉米_Grants
吉米爵士	Jimmy_Jazz
吉米·约翰的	Jimmy_John's
Jimmy 相当	Jimmy_Fairl
Jimmy's 杀手炸鱼薯条	Jimmy's Killer_fish_&_Chips
Jimmy 的杀手烧烤炉	Jimmy's Killer_Grills
Jimmy's 杀手大虾	Jimmy's Killer_Prawns
金心意大利面	Jin_Heart_Past
金江	姜进
金祥强郭成	金祥强郭成
金贝	Jin_bei
Jin mart	jin_mart
Jin's Inn	Jin's Inn
JinDingXuan	JinDingXuan
JinSun 证券	JinSun_证券
JinYi 电影院	JinYi_电影院
金贝	金贝
晋城银行	晋城银行

名称	ID
晋城农村商业银行	晋城_农村_商业银行
Jinderlimin	Jinderlimin
金都百货	金都百货
景阁老灶火锅	Jing_Ge_Lao_Zao_Hotpot
静涵教育	韩静_教育
景莱酒店	京莱酒店
景威餐厅	京味餐厅
JingYa	JingYa
京城138家客栈&酒店	景城_138_客栈&酒店
景德镇农村商业银行	景德镇_农村_商业银行
京东便利店	京东_便利店
京东五星	京东_五星_五星
Jingduxiefu	Jingduxiefu
精功	精功
精功超市	精功_超市
晶晶教育	京京_教育
晶晶幼儿园	晶晶_幼儿园
景客隆	景客隆
静丽瑜伽	Jingli_Yoga
京伦饭店	景伦酒店

名称	ID
金谷农村商业银行	金谷_农村_商业银行
金贵黄潮	金贵黄潮
景县农村商业银行	景县_农村_商业银行
静源瑜伽	静源瑜伽
Jingyuerong	Jingyuerong
金华银行	金华银行
济宁包商农村银行	济宁_包商_农村银行
锦江之星	锦江客栈
锦江之星精选酒店	锦江_客栈_精选
锦江之星风尚酒店	锦江_客栈_风格
晋江农村商业银行	晋江_农村_商业银行
金卡酒店	Jinka_Hotel
金陵	金陵
金泉道	金泉道
晋商银行	晋商银行
金盛国际家居装饰	金盛国际_家居_装饰
劲松口腔医院	劲松_口腔医院
金威训练	金威_训练
金怡国际影城	金怡_国际_影院
锦州农村商业银行	锦州_农村_商业银行

名称	ID
Jio Store	Jio_Store
jio-bp	jio-bp
吉群学校	Jiqun_School
吉视传媒	Jishi_Media
九国一汤	Jiu_Guo_Yi_Tang
九井加油站	九井加油站
九天之家	九天_首页
九头鹰餐厅	九头英餐厅
久光	久光
久豪租车	久豪_租车
九毛九喜贝菜	Jiumaojiu_xibeacia
九台龙嘉乡村银行	九台_龙家_乡村银行
九月山	九月山
九洲商业大厦	九洲商业大厦
九洲超市	九洲_超市
鸡西农商银行	鸡西农村商业银行
Jmc	Jmc
江铃福特	Jmc_Ford
Jo Mercer	Jo_Mercer
JoS。A. 银行	JoS。 _A._BANK

名称	ID
骑师	骑师
Joe Weider	Joe_Weider
Joe 的汽车公园	Joe's Auto_Parks
Joe's Crab Shack	Joe's_crab_shack
Joe's Kwik Mart	Joe's_kwik_mart
Joes	Joes
乔伊牙科	Joey_Dental
只有 Joey's	Joey's_only
慢跑内衣	Jogè_Lingery
John Dory 的	John_Dory's
约翰·加菲尔德	约翰·加菲尔德
约翰·汉考克	约翰·汉考克
约翰·雅各布斯	John_Jacobs
约翰·L·斯科特房地产	John_L._Scott_Real_Estate
约翰·刘易斯	约翰·刘易斯
城南信用金库	Johnan_Shinkin_Bank
约翰尼的	约翰尼的
约翰尼火箭队	约翰尼_火箭队
约翰尼的	约翰尼的
约翰尼市场	Johnny's_Mar

名称	ID
约翰尼的披萨屋	Johnny's Pizza_Hous
Johnson 健身与健康	约翰逊_健身_&_健康
约翰斯顿和墨菲	Johnston_&_Murphy
约翰斯通供应	约翰斯通_供应
城北信用金库	Johoku_Shinkin_Bank
联合出版	联合出版
小丑	小丑
Jokers	Jokers
Jola 模式	Jola_Mode
Jolidon	Jolidon
Joliesse	Joliesse
Jollibean	Jollibean
Jollibee	Jollibee
Jolly Hotel	Jolly_Hotel
Jomo	Jomo
Jon Smith Subs	Jon_Smith_Subs
Jones	Jones
琼斯靴匠	Jones Bootmaker
琼斯纽约	Jones_New_York
Jonwayauto	Jonwayauto

名称	ID
Joom Zap Hut	Joom_Zap_Hut
Joplin Metro CU	Joplin_Metro_cu
Jorcel Chausseur	Jorcel_Chausseur
约旦阿赫利银行	Jordan_Ahli_Bank
约旦 CU	Jordan_cu
约旦商业银行	约旦商业银行
约旦迪拜伊斯兰银行	约旦_迪拜_伊斯兰银行
约旦伊斯兰银行	约旦伊斯兰银行
Jordan's Kwik Stop	Jordan's_kwik_stop
Jordbárpikene	Jordbárpikene
Jos Alukkas	Jos_Alukkas
Josco 珠宝商	Josco_Jewelers
Josephino 的	Josephino 的
Joshin	Joshin
约书亚·多尔	Joshua_Doore
Joshu's Shoarma	Joshua's_shoarma_Grill
Jouetland	Jouetland
Jouetland B Comme Bebé	Jouetland_B_Comme_Bebé
Jouetland Bébéland	Jouetland_Bébéland
旅程	旅程

名称	ID
Jounrey	journey_cu
旅程的尽头	Journey's end
旅程	旅程
Journeys	Journeys_Kidz
Joventour Viajes	Joventour_Viajes
乔维亚· Financial/FKA 拿骚埃德	jovia_Financial/fka_Nassau_ed
大悦城	Joy_City
欢乐英语	快乐_英语
Joy 酒店	欢乐酒店
乔伊·玛特	Joy_Mart
欢乐与和平	欢乐与和平
乔亚酒店	Joya_Hotel
Joyalukkas	Joyalukkas
Joyca	Joyca
乔伊斯·莱斯利	Joyce_Leslie
Joymax	Joymax
欢乐一号	Joyous_One
Jr 食品超市	Jr_Food_Mart
Jr 食品商店	Jr_Food_Store
Jreck Subs	Jreck_Subs

名称	ID
Js 休闲装	Js_Casual
Jstone 意大利餐厅	Jstone_Italian_Restauran_
Jtexp 快递	jtexp_Express_Del
Jtour Inn	Jtour_Inn
居宝元	ju_Bao_Yuan
胡安·马埃斯特罗	Juan_Maestro
胡安·波洛	Juan_Pollo
胡安·瓦尔迪兹咖啡厅	Juan_Valdez_Café
Jubilant Lifesciences	Jubilant_Lifesciences
银禧保险	银禧_保险
Jubin Frères	Jubin_frères
Jubitom	Jubitom
Juelmin 保险服务	Juelmin_保险_服务
觉薇	觉薇
Jugo 果汁	Jugo_Juice
Juick Up	Juice_It_Up
果汁沙龙	Juice_Salon
果汁停止	Juice_Stop
JuicePoint	JuicePoint
Juiland	Juiland

名称	ID
Juicy Couture	Juicy_Couture
Jula	Jula
朱利安格雷夫斯	Julian_Graves
Julie Guerlande	Julie_Guerlande
Julien d'Orcel	Julien_d'Orcel
胡利奥	胡利奥
胡里奥·塞佩达	Julio_Cepeda
Julius Baer	Julius_Baer
巨大的	巨大的
巨额现金和随身携带	Jumbo_Cash_&_Carry
Jumbo Home 专卖店	Jumbo_Home_Speciality_Stor
巨额分数	Jumbo_Scor
巨型海鲜	巨型_海鲜
大放异彩	大放异彩
朱美拉	朱美拉
跳跃	跳跃
套头衫	套头衫
Jun 酒店	Jun_Hotel
朱娜	朱娜
朱奈德·贾姆希德	Junaid_Jamshed

名称	ID
Junaid for 香水	junaid_for_perfumes
聚能教育	聚能_教育
Junge	Junge
丛林小屋和度假村	丛林小屋和度假村
Junker	Junker
君来大饭店	君来酒店
朱诺	朱诺
俊义	俊义
朱皮公园	Jupi_Park
Jurki	Jurki
Jurlique	Jurlique
陪审团	陪审团
巨森	巨森
只是烘烤	只是烘烤
刚出炉了	Just_Bak
只是刹车	只是刹车
只要买自行车就行了	只需要购买周期
只是削减	Just_Cuts
只是牛仔裤	Just_Jean
只是墨西哥人	只是墨西哥人

名称	ID
只是轮胎	Just_Tries
稍等片刻	just_a_momen
Just-A-Buck	Just-A-Buck
JustHome	JustHome
Justherb	Justherb
正义	正义
正义 FCU	Justice_fcu
Justo & Bueno	Justo_&_Bueno
日德兰德银行	日德兰德银行
Juvelirtorg	Juvelirtorg
巨阳	巨阳
Juzishu	Juzishu
Jysan 银行	jysan_bank
Jyske Bank	Jyske_Bank
Jyu-Jyu Karubi	Jyu-Jyu_Karubi
Jászárokszállás 和 Vid。Takareksz。	jászárokszállás_es_vid。 _Takareksz。
Jarnia	Jarnia
K&G 时尚超市	K&G_Fashion_Superstore
K&H Bank	K&H_Bank
K&H 国际学校	K&H_国际学校

名称	ID
K&K	K&K
K&N'S	K&N'S
K&U Báckerei	K&U_Báckerei
K&W Cafeterias	K&W_Cafeterias
K 的壁橱	K's_Closet
K's PARK	k's_park
K+K	K+K
K+K Schuhcenter	K+K_Schuhcenter
K-城市市场	K-城市市场
K-Ext	K-Ext
韩国市场	韩国市场
K-RAUTA	K-RAUTA
K-Rep 银行	K-Rep_Bank
K-State FCU	k-state_fcu
K-超市	K-超市
K拳击	K拳击
KA HO 餐厅	KA_HO_RESTAURAN
KAGETSU ARASHI	KAGETSU_ARASHI
KAHLS THE & KAFFEHANDE	KAHLS_THE_&_Kaffehandel
KAI BO 食品超市	KAI_BO_FOOD_SUPER

名称	ID
开元义菊	Kai_yuan_yi_ju
KAIBIN-INN	KAIBIN-INN
KAIRIKISHI	KAIRIKISHI
KAISERDOM 酒店	KAISERDOM_HOTEL
KAKADU	KAKADU
KALIFA	KALIFA
卡洛里亚斯	卡洛里亚斯
KAMA 阿育吠陀	kama_Ayurveda
KAMMI	KAMMI
木棉酒店	KAPOK_HOTEL
KAREN MILLEN	KAREN_MILLEN
KARUSEL'	KARUSEL'
KASB 银行	Kasb_bank
KASIKORNBANK	KASIKORNBANK
KASSELER BANK	KASSELER_BANK
KASTOR	KASTOR
考艾岛	考艾岛
KAV 租車	KAV_autoverHur
KAYSERI MUTFAGI	KAYSERI_MUTFAGI
KB 主页	kb_Home

名称	ID
KB 国民银行	kb_kookmin_Bank
KBC	KBC
爱尔兰比利时联合银行	KBC_Bank_Ireland
比利时联合银行	kbc_Banka
KBL	KBL
KBP Foods	kbp_Foods
KCB	KCB
KCELL	KCELL
KCT 信用合作社	KCT_Credit_Union
KDB 银行	KDB_Bank
KDL OLIMP	KDL_OLIMP
KEB Hana Bank	keb_hana_bank
KEBUKE	KEBUKE
KEDDY	KEDDY
KEDR	KEDR
KEFAN	KEFAN
小桶	小桶
肯巴金融 CU	kemba_financial_cu
KENNARDS HIRE	KENNARDS_HIRE
KENOL	KENOL

名称	ID
KENT&CURWEN	KENT&CURWEN
肯伍德	肯伍德
KENZO	KENZO
KERAIA	KERAIA
KEROPETROL	KEROPETROL
凯里酒店	KERRY_HOTEL
KEWEN 瑜伽	KEWEN_YOGA
KF-Park	KF-Park
肯德基	肯德基
凯基银行	KGI_Bank
凯基证券	KGI_证券
汉特-曼西斯基银行	汉蒂-曼西斯基_银行
KHS	KHS
KIABI	KIABI
KICHIRI 休闲餐饮	Kichiri_Casual_Dining
踢	踢
KIFIDIS	KIFIDIS
KIKO	KIKO
KIMIA FARMA	KIMIA_FARMA
金普顿	金普顿

名称	ID
善良	善良
KIND Hörzcentralen	Kind_HörzCentralen
KINDANCE INT'L INT'L	KINDERDANCE_INT'L_INC
国王车 KAVALAN	KING_CAR_KAVALAN
KING KOIL	KING_KOIL
国王体育	KING_SPORT
特大弹簧床	特大弹簧床
国王面包店	国王面包店
KINEPOLIS	KINEPOLIS
KIP	KIP
KIPS	KIPS
KIRISHIAVTOSERVIS	KIRISHIAVTOSERVIS
KIRJAKASSI	KIRJAKASSI
KIRKPINAR	KIRKPINAR
吉斯顿酒店及度假村	KISTON_酒店及度假村
KITEA	KITEA
KITSCHEN	KITSCHEN
猕猴桃	猕猴桃
KIWOKO	KIWOKO
KK 超级市场	kk_super_mart

名称	ID
KLER	KLER
KLG	KLG
KLINK	KLINK
KLM	KLM
km-prona	km-prona
KNEZ 汽油	KNEZ_PETROL
KNEZ 汽油市场	KNEZ_PETROL_MARKET
骑士酒店	KNIGHTS_HOTEL
克尼·多布罗夫斯基	KNIHY_DOBROVSKY
KNPC	KNPC
KOA	KOA
KOAJ	KOAJ
小林	小林
KOBIL	KOBIL
KODOMO	KODOMO
KODi 折扣	KODi_折扣
KOH-I-NOOR papirnictvi	koh-i-noor_papirnictVI
KOHIKAN	KOHIKAN
科勒	科勒
KOI Cafe	Koi_cafe

名称	ID
KOI The	koi_the
KOI Thé	Koi_thé
小岛 x BIC CAMERA	Kojima_x_bic_Camera
科尔波特	科尔波特
KOM	KOM
KOMANDOR	KOMANDOR
KOMANDOR SuperMarket	KOMANDOR_ SuperMarket
商业银行	商业银行
COMFORT	COMFORT
KONIKOM	KONIKOM
KONZUM	KONZUM
KOOKAI	KOOKAI
KOOKENKA	KOOKENKA
KOOP Zaedno na Pazar	koop_zaedno_na_pazar
KOPITIAM	KOPITIAM
KORKMAZ	KORKMAZ
KOSOVA 汽油	KOSVA_PETROL
KOTIPZZA	KOTIPZZA
KOUFU	KOUFU
KOÇAK GOLD	KOÇAK_GOLD

名称	ID
毕马威会计师事务所	毕马威会计师事务所
KPN	KPN
KPetrom	KPetrom
KQTEA	KQTEA
KR Bakes	KR_Bakes
KRACK ZAPATERIAS	KRACK_ZAPATERIAS
克拉科夫斯基克雷登斯	KRAKOWSKI_KREDENS
欧洲信贷银行	CREDIT_EVROPA_BANK
CRITIKOS	CRITIKOS
KRUNGTHAI 银行	KRUNGTHAI_BANK
KRÈFEL	KRÈFEL
KS Bakers	ks_Bakers
堪萨斯州银行	KS_Bank
KSFE	KSFE
KSIEGARNIE NOVA	KSIEGARNIE_NOVA_DUO
KSW FCU	KSW_FCU
KTDC	KTDC
KTM	KTM
KUBENZ	KUBENZ
库卡	库卡

名称	ID
KUKUXUMUSU	KUKUXUMUSU
KULTAJOUSI	KULTAJOUSI
KUMON	KUMON
KUSMI TEA	KUSMI_TEA
KUTAHYA SERAMIK	KUTAHYA_SERAMIK
科威特国际银行	科威特国际银行
KV FCU	KV_FCU
KVARNER PUNAT	KVARNER_PUNAT
KVB Rad	kvb_rad
KVK	KVK
KWFT	KWFT
KWIATY POLSKIE	KWIATY_POLSKIE
KWIK SAVE	KWIK_SAVE
KWIK STAR	KWIK_STAR
KWIKSPAR	KWIKSPAR
KWL Aqua	kwl_Aqua
KWL GmbH	kwl_gmbH
KWSB	KWSB
KYMCO	KYMCO
Ka 国际	Ka_国际

名称	ID
KaBloom	KaBloom
Kaba	Kaba
Kababji 餐厅	Kabji_Restaurant
Kabhi B	Kabhi_B
Kabova	Kabova
Kadernictvi Klier	Kadernictvi_Klier
Kadoil	Kadoil
卡多尔市场	Kadoil_Market
Kafe Betawi	Kafe_Betawi
Kafe Rosneft'	Kafe_Rosneft'
Kaffebrenneriet	Kaffebrenneriet
卡夫卡斯	卡夫卡斯
Kagonoya	Kagonoya
鹿儿岛工业信用合作社	鹿儿岛_Kogyo_Credit_合作社
鹿儿岛信用金库	鹿儿岛信用金库
鹿儿岛崇光信用金库	鹿儿岛_崇光_信用金库
Kahoots	Kahoots
Kahta Cigkofte	Kahta_Cigkofte
Kahve Diyari	Kahve_Diyari
Kahve Dükkani	Kahve_Dükkani

名称	ID
Kahve Dünyasi	Kahve_dünyasi
凯惠	Kai_Hui
开元大酒店	开元大酒店
开元关堂	唐开元
开元曼居酒店	开元_曼_居_Hotel
Kai Yuan Ming	Kai_Yuan_Ming_ting
开元度假村	开元度假村
凯宾公寓	凯宾_公寓
Kaidonno	Kaidonno
开芬	开芬
凯拉斯	凯拉斯
Kailash Parbat	Kailash_Parbat
开平农村商业银行	开平农村商业银行
Kaisen Yatai Okuman	Kaisen_Yatai_Okuman
凯撒永久化	Kaiser_Permanent
Kaiser Permanente Healt	Kaiser_Permanente_Health_
Kaitori Okoku	Kaitori_Okoku
Kaiyang Wichainburi	Kaiyang_Wichainburi
开元	开元
开元证券	开元证券

名称	ID
Kajiken	Kajiken
Kakao	Kakao
Kakujo Gyorui	Kakujo_Gyorui
Kakuyasu Beer-to-Tetsunabe Gyoza 3 6 5 Sakaba	kakuyasu_beer_to-_to-_tetsunabe_gyoza_3_6_5_Sakaba
Kal Tire	Kal_Tire
Kalakap	Kalakap
Kaliko	Kaliko
Kalina Oil	Kalina_Oil
Kalinka	Kalinka
Kallappanna Awade IJS Bank	kallappanna_awade_ijs_Bank
Kalunga	Kalunga
Kalupur 商业合作银行	Kalupur_Commercial_Co-op_Bank
Kayan Jewelers	Kalyan_Jewelers
Kalyan Silks	Kalyan_Silks
Kamado Gohan Sachifukya	Kamado_Gohan_Sachifukya
Kamadoka	Kamadoka
镰仓光悦	镰仓_甲越
Kamalaya	Kamalaya
卡曼工业科技	Kaman_工业_技术
Kamasdu	Kamasdu

名称	ID
Kamashin	Kamashin
Kamat yatrinivas	kamat_yatrinivas
卡姆达尔	卡姆达尔
龟有信用金库	Kameari_Shinkin_Bank
神井餐厅	神井_餐厅
Kamil Koça	Kamil_Koça
Kamiseta	Kamiseta
甘榜鸡舍	Kampong_Chicken_House
Kamps	Kamps
神仓	神仓
神奈川信用金库	神奈川县_信用金库
金泽信用金库	金泽信用金库
Kanbawza 银行	Kanbawza_Bank
坎迪	坎迪
Kandon	Kandon
Kaneko Gankyo-ten	Kaneko_Gankyo-ten
Kanesue	Kanesue
Kangaiduo	Kangaiduo
袋鼠快车	袋鼠快车
Kangela	Kangela

名称	ID
Kanghodong Baekjeong	Kanghodong_Baekjeong
坎格拉中央合作银行	Kangra_Central_合作社_银行
康真	康真
Kankatala 纺织品	Kankatala_纺织品
鹿川制面	Kanokawa_Seimen
Kanouka	Kanouka
Kanseki	Kanseki
坎斯基	坎斯基
Kantaro	Kantaro
Kantata	Kantata
康提糖果	Kanti_Swets
KantsPark	KantsPark
Kantsler	Kantsler
Kanz	Kanz
Kanzelsberger	Kanzelsberger
坎兹勒	坎兹勒
Kaoba	Kaoba
Kaoil	Kaoil
Kapika	Kapika
KappAhl	KappAhl

名称	ID
卡拉	卡拉
卡拉·菲林	Kara_Firin
Karaage Yukari	Karaage_Yukari
Karaage Daikichi	Karaage Daikichi
卡拉卡	卡拉卡
卡拉奇面包店	卡拉奇_面包店
卡拉达格	卡拉达格
卡拉OK Mac	Karaoke_Mac
卡拉OK Mokomoco	Karaoke_Mokomoko
卡里	卡里
Kari Baby	Kari_Baby
Karims	Karims
Karisma	Karisma
Karlova pekarna	Karlova_pekarna
Karmelkorn	Karmelkorn
卡纳塔克邦银行	卡纳塔克邦银行
卡纳塔克邦格拉明银行	Karnataka_Gramin_Bank
卡纳塔克邦合作社Apex银行	Karnataka_State_Co-op_Apex_Bank
卡纳塔克邦 Vikas Grameena 银行	Karnataka_Vikas_Grameena_Bank
Karo Fil'm	Karo_Fil'm

名称	ID
Karry	Karry
卡尔施塔特	卡尔施塔特
Kartal Yuvasi	Kartal_Yuvasi
Kartopliana Khata	Kartopliana_Khata
Karubiya Daifuku	Karubiya_Daifuku
Karur Vysya Bank	Karur_Vysya_Bank
Karwei	Karwei
Kasa Stefczyka	Kasa_Stefczyka
Kasap Doner	Kasap_Doner
Kashemir i Shyolk	kashemir_i_shyolk
Kashieca	Kashieca
Kashikaigishitsu 我的空间	Kashikaigishitsu_My_Space
Kasmir Hali	Kasmir_Hali
Kasnet-Agente	Kasnet-Agente
Kaspi 银行	Kaspi_Bank
Kasseler Sparkasse	Kasseler_Sparkasse
Kastrati	Kastrati
Katahdin FCU	katahdin_fcu
凯特·希尔	Kate_Hill
Kate Spade	Kate_Spade

名称	ID
加德满都	加德满都
Katies	Katies
Katik Döner	Katik_Döner
Katsu Ten	Katsu Ten
胜花亭	胜花亭
胜仓	胜仓
考夫兰	考夫兰
Kaufland EV	Kaufland_ev
考夫曼	考夫曼
Kaufpark	Kaufpark
考蒂利亚学院	Kautilya_academy
Kaveri Grameena Bank	Kaveri_Grameena_Bank
河内屋	河内屋
川口信用金库	川口信用金库
Kawan 村庄	Kawan_Villag
川崎信用金库	川崎信用金库
凯珠宝商	Kay_Jewelers
Kaya	Kaya
Kaymo	Kaymo
Kayseria	Kayseria

名称	ID
KazKom	KazKom
KazMunayGaz	KazMunayGaz
Kazanskie Apteki	Kazanskie_Apteki
Kazar	Kazar
风之町	风之町
Kazoku-te	Kazoku-te
卡兹邮报	卡兹邮报
Kai	Kai
Kb 的合理价格	KB's_Fair_price
Kcas	Kcas
肯德基唯一的咖啡	肯德基的_only_Coffee
柯浩	Ke_Hao
Kearny Bank	Kearny_Bank
Kebap Savvas	Kebap_Savvas
Kedai Buku H Q	Kedai_Buku_H_Q
Kedai RM 2	kedai_rm_2
Kedai Rakyat 1 马来西亚	Kedai_Rakyat_1_Malaysia
Keddo	Keddo
Kedi	Kedi
Keds	Keds

名称	ID
Keds Kids	Keds_Kids
基华	Kee_Wah
Keeley Ann	Keeley_Ann
基恩牙科诊所	Keen_Dental_Clinic
KeepCool	KeepCool
Keesler FCU	keesler_fcu
基瓦	基瓦
Kegou	Kegou
Keisy	Keisy
Keith 的超市	Keith's Superstores
Keke's 早餐咖啡厅	Keke's 早餐咖啡厅
Kelebek Mobilya	Kelebek_Mobilya
克林宁	克林宁
凯勒威廉姆斯房地产	Keller_Williams_Realty
凯莉·威廉姆森	Kelley_Williamson
凯利市场	Kelley's Market
凯利服务	凯利_服务
凯利-摩尔涂料	Kelly-Moore_Paints
Kelsey's	Kelsey's
Kem	Kem

名称	ID
Kemal Tanca	Kemal_Tanca
Kemba CU Inc.	kemba_cu__inc.
凯宾斯基	凯宾斯基
肯	肯
KenJo 市场	KenJo_市场
KenTire 和 AutoCare Towery's	KenTire_&_ _Towery's AutoCare
肯奇克旅馆	Kenchic_Inn
垦丁大饭店	肯丁酒店
Kendoil	Kendoil
Kendriya Vidyalaya	Kendriya_Vidyalaya
Kengee	Kengee
肯古鲁	肯古鲁
肯煌仁	肯煌仁
肯尼迪炸鸡	Kennedy_Fried_Chicken
肯尼迪传输	刹车和汽车维修
Kenneth Cole 纽约	Kenneth_Cole_New_York
肯尼的 Marketplace	Kennie's Marketplac
肯尼·罗杰斯烘焙师	Kenny_Rogers_Roasters
Kent Kwik	Kent_Kwik
KentBank	KentBank

名称	ID
肯塔基州	肯塔基州
Kenuo	Kenuo
Kenvelo	Kenvelo
喀拉拉邦快车	喀拉拉邦快车
喀拉拉邦考穆迪	喀拉拉邦_考穆迪
喀拉拉邦公路	喀拉拉邦_道路工程
Kerama Marazzi	Kerama_Marazzi
Keramundo-Welt der Fliesen	Keramundo-welt_der_fliesen
Kerasotes 剧院	Kerasotes_Theatres
Kerr Drug	Kerr_Drug
Kerr-McGee	Kerr-McGee
凯里快车	Kerry_Expr
Kerum	Kerum
Kervan Dondurma	Kervan_Dondurma
Kesoil	Kesoil
凯瓦果汁	Keva_Juice
Keventers	Keventers
凯文斯顿	凯文斯顿
键	键
关键食物	Key_Food

名称	ID
KeyBank	KeyBank
KeyMe	KeyMe
KeyStore	KeyStore
KeyWear	KeyWear
基斯通银行	Keystone_Bank
Keyuca	Keyuca
Keíto	Keíto
Kfar 'ashu'im HaSha	Kfar_ 'ashu'im HaSha
Khaadi	Khaadi
哈利吉商业银行	Khaleeji_商业银行
哈利勒	哈利勒
Khan Chacha	Khan_Chacha
Khimchistka Diana	Khimchistka_Diana
Khleb Nasushchnyy	Khleb_Nasushchnyy
Khleb Sol'	Khleb_Sol'
Kholdi Diskaunter	Kholdi_diskaunter
Kholidey Klassik	Kholidey_Klassik
Khoroshaya apteka	Khoroshaya_apteka
Khoum Credit	Khoum_Credit
Khreshchatyk	Khreshchatyk

名称	ID
库拉纳游记	Khurana_Travels
Khushhali 银行	Khushhali_Bank
KiK	KiK
KiKi	KiKi
起亚公司	起亚公司
Kiatnakin Phatra Bank	Kiatnakin_Phatra_Bank
Kibek	Kibek
Kibouken	Kibouken
Kichi-Kichi	Kichi-Kichi
孩子城堡	Kid_Castle
孩子的微笑	Kid_Smile
Kid Veda	Kid_Veda
儿童内饰	Kid_interiör
孩子对孩子	kid_to_kid
KidStok	KidStok
小孩学院	Kiddie_academy
孩子的课堂	Kiddy's Class
儿童脚柜	儿童足部储物柜
儿童探索	Kids_Ques
Kids R Us	Kids_R_Us

名称	ID
Kidsfirst 幼儿园	Kidsfirst_幼儿园
儿童酒店	kinderfreundlichen
基德斯兰	基德斯兰
Kidsplace	Kidsplace
Kidzee	Kidzee
Kiehl's	Kiehl's
Kieler Volksbank	Kieler_Volksbank
Kienlongank	Kienlongank
Kigili	Kigili
Kignas	Kignas
kiinteistömaailma	kiinteistömaailma
Kiki's 快餐	Kiki's_fast_food
Kiler	Kiler
基利姆	基利姆
Killfish 折扣吧	Killfish_Disch_Bar
Killiney	Killiney
Kiloutou	Kiloutou
金	金
Kim 牙科	Kim_Dental
金氏俱乐部	Kim's Club

名称	ID
Kimex	Kimex
金普顿酒店	金普顿_酒店
Kimukatsu	Kimukatsu
金氏农场市场	kin's_Farm_Marke
金之鸟乐	金之鸟乐
Kind Hörgeräte	Kind_Hörgeräte
Kindal	Kindal
KinderCare 学习中心	KinderCare_学习中心
志同道合医疗保健	Kindred_医疗保健
KindsWant	KindsWant
Kindy ROO	Kindy_rooo
动力学工程	动力学_工程
国王烧烤	King_BBQ
国王咖啡	国王咖啡
国王来吧	King_Come
King Jouet	King_Jouet
King Krak 自助餐厅	King_Krak_Caf
库伦国王	King_Kullen
金龙	King_Long
国王派	King_Pie

名称	ID
King Soopers	King_Soopers
King Soopers 燃料中心	King_Soopers_Fuel_Center
King Soopers 药房	King_Soopers_Phar
King Sun 餐厅	King_Sun _餐厅
国王茶	King_Tea
刀之王	刀之王
国王糖果	King's_Cred
国王国际英语	King's 国际_英语
国王城银行	King's_town_Bank
国王杂货店	King's Variety_Store
王国银行	Kingdom_Bank
国王家族餐厅	Kings_Family_ 餐厅
国王珠宝	Kings_珠宝
国王超级市场	国王超级市场
金斯米德鞋	Kingsmead_Shoes
金斯波特	金斯波特
金石书店	Kingstone_Bookstore
Kingsway 轮胎	Kingsway_Tyres
近畿产业信用合作社	Kinki_Industrial_Credit_合作社
Kinki Rokin	Kinki_Rokin

名称	ID
金洛克·安德森	金洛赫_安德森
金尼毒品	金尼_毒品
Kinno Buta	Kinno_Buta
纪之国信用金库	Kinokuni_Shinkin_Bank
Kinomaks	Kinomaks
Kinopark	Kinopark
Kinoplex	Kinoplex
Kinoplexx	Kinoplexx
Kinta	Kinta
金坛自助餐	金坛_自助餐
Kinuya	Kinuya
Kiosko	Kiosko
Kip Ling	Kip_Ling
Kipa	Kipa
吉卜林	吉卜林
Kira Moda	Kira_Moda
Kira Plastinina	Kira_Plastinina
麒麟城	麒麟城
柯克兰的	柯克兰的
Kirmizi Döner	Kirmizi_Döner

名称	ID
基罗夫斯基伊	基罗夫斯基伊
桐生信用金库	桐生信用金库
木曾寺	木曾寺
北伊势上野信用金库	Kitaise_Ueno_Shinkin_Bank
喜多方拉面番内	Kitakata_Ramen_Bannai
北见信用金库	北见信用金库
北野市场	北野市场
北大阪信用金库	北大阪_信用金库
Kitch'n	Kitch'n
厨房和桌子	厨房和桌子
厨房求解器	厨房求解器
厨房用品	厨房用品
厨房调校	Kitchen_Tune-Up
Kiton	Kiton
Kitro	Kitro
相机和图像套件	Kits_Camera_&_Image
Kiwibank	Kiwibank
Kjell & Company	Kjell_&_Company
克拉拉	克拉拉
Klassika	Klassika

名称	ID
Klasykbank	Klasykbank
Klava Davai	Klava_Davai
Kleiderbauer	Kleiderbauer
Klenger Burger	Klenger_Burger
Klenotnictvi PRÉSENCE	klenotnictvi_Présence
Klenoty Aurum	Klenoty_Aurum
克利尔	克利尔
Klinicare	Klinicare
Klirynhovyi Dim	Klirynhovyi_Dim
Klo	Klo
时钟大师	时钟大师
居琅站	Kluang_Station
Klub Fox	Klub_Fox
凯马特	凯马特
Kmoil	Kmoil
Knauber	Knauber
骑士办公室解决方案	Knight_Office_解决方案
骑士旅馆	Knights_Inn
哥伦布骑士团	哥伦布骑士
Knigomir	Knigomir

名称	ID
Knizhniy Labirint	Knizhniy_Labirint
Knizhniy Mir	Knizhniy_Mir
Knn Idiomas	kn_Idiomas
Knopka	Knopka
结	结
Knowfat！生活方式格栅	Knowfat！ _生活方式_格栅
小林光学	小林光学_光学
神户信用金库	神户信用金库
Koberce BRENO	Koberce_Breno
Koberce 趋势	Koberce_Trend
高知信用金库	Kochi_Shinkin_Bank
Kochlöffel	Kochlöffel
Kocolene	Kocolene
柯达快车	柯达快车
Kodi	Kodi
Koerich	Koerich
Kof 塔	Kof_Tower
Kofteci Ramiz	Kofteci_Ramiz
甲府信用金库	甲府信用金库
Kohl's	Kohl's

名称	ID
Kohr Bros	Kohr_Bros
Kohyo	Kohyo
科科·布莱克	椰子黑色
Kolache 工厂	Kolache_Factory
Koleso	Koleso
Kollam 至上	Kollam_Supre
Kolon Sport	Kolon_sport
Colonial Home	Colonial_Home
Kolonna	Kolonna
Koltso Urala	Koltso_Urala
Komagene	Komagene
Komeda 的咖啡	Komeda's Coffee
商业银行 AD Banja Luka	Komercijalna_banka_ad_banja_Luka
商业银行	Komercni_Banka
Komertsiinyi Industrialnyi Bank	Komertsiinyi_Industrialnyi_Bank
韩国商业投资银行	韩国商业投资银行
KommunalCredit	KommunalCredit
Komp'yuterniy Mir	komp'yuterniy_mir
Kompanero	Kompanero
Kompedan	Kompedan

名称	ID
Kompen	Kompen
Komputronik	Komputronik
Komus	Komus
Kona Grill	Kon_Grill
Koncept 家具	概念_家具
Condomeriet	Condomeriet
Konfor	Konfor
金咖啡	空咖啡
Koni Store	Koni_Store
Konkord	Konkord
Konmar	Konmar
Konnect	Konnect
河野信用金库	Kono_Shinkin_Bank
康拉德·卡塞尔	康拉德_卡塞尔
Konshinya	Konshinya
Konstandinidis	Konstandinidis
Konsul	Konsul
消费	消费
联系我们	联系我们
Kontanten	Kontanten

名称	ID
Kontigo	Kontigo
Konyali Haci Usta	Konyali_Haci_Usta
Koo Koo Roo	Koo_koo_Roo
Koolau FCU	koolau_fcu
Koong Wo Tong	Koong_Woh_Tong
Koorsen 消防与安保	Koorsen_Fire_&_Security
Kop og Kande	kop_og_Kande
哥本哈根	哥本哈根
Kopeyka	Kopeyka
Kopi Kenangan	Kopi_Kenangan
Kopp 的冰冻蛋羹	Kopp's_frozen_Custard
甲罗本店	Kora_Honten
Korablik	Korablik
Korall	Korall
Koray	Koray
韩国开发银行	韩国开发银行
韩国外汇银行	韩国交易所银行
韩国第一银行	韩国第一银行
韩国炒年糕	Korean_teokbokki
Koreana	Koreana

名称	ID
Korian	Korian
Korona	Korona
Korona TEKh 不是	Korona_ 不是 TEKh
Korona Takarék	Korona_Takarék
Corporatsiya Tsentr	Corporatsiya_Tsentr
Korus	Korus
Korzinka	Korzinka
Kosebasi	Kosebasi
Kosiuko	Kosiuko
Kosmos	Kosmos
Kot i Pyos	kot_i_pyos
Kotak Mahindra Bank	Kotak_Mahindra_Bank
Koto	Koto
Kotofey	Kotofey
Koton	Koton
Kotsovolos	Kotsovolos
Koupelny Ptacek	Koupelny_Ptacek
Koutons	Koutons
Koutubia	Koutubia
Kovai Pazhamudir Nilayam	Kovai_Pazhamudir_Nilayam

名称	ID
Kovapaint	Kovapaint
九龙大厦	九龙大厦
Koyunlu Hali	Koyunlu_Hali
科泽科德地区合作银行	科泽科德地区合作银行
Kozmo	Kozmo
科兹莫斯	科兹莫斯
Kraftman FCU	Kraftman_FCU
Kragen 汽车零部件	Kragen_Auto_Parts
喀拉喀托家具	喀拉喀托_家具
克拉科夫斯基银行 Spoldzielczy	Krakowski_Bank_Spoldzielczy
Kral Cig Kofte Osman Usta	Kral_Cig_Kofte_Osman_Usta
Krasniy Kub	Krasniy_Kub
Krasny Pishchevik	Krasniy_Pishchevik
克拉斯尼亚尔	Krasnie_Yar
Krasnoe i Beloe	krasno_i_beloe
克拉斯诺亚尔斯克石油产品	克拉斯诺亚尔斯克石油产品
Krass Optik	Krass_Optik
Kravitz	Kravitz
克雷投资银行	克雷投资银行
Kraysneft'	Kraysneft'

名称	ID
Kreatos	Kreatos
Kredobank	Kredobank
Kredyt Optyma Bank	Kredyt_Optyma_Bank
Kredyt Yevropa Bank	Kredyt_Yevropa_Bank
Kredyt-Dnipro	Kredyt-Dnipro
Kredytvest Bank	Kredytvest_Bank
Kreepy Krauly	Kreepy_Krauly
Kreis-u。迪林根城市公园大厅	kreis-_u._stadtsparkasse_Dillinge
Kreis-u。Stadtsparkasse Kaufbeuren	kreis-_u._stadtsparkasse_kaufbeuren
Kreis-u。瓦瑟堡 Stadtsparkasse	kreis-_u._stadtsparkasse_Wasserburg
施派尔县和城市公园大楼	kreis-_und_stadtsparkasse_Speyer
Kreis-und Stadtspk。Dinkelsbühl	kreis-_und_stadtspk。 _dinkelsbühl
Kreis-und Stadtspk。Erding-Dorfen	kreis-_und_stadtspk。 _Erding-Dorfen
Kreis-Sparkasse Northeim	Kreis-Sparkasse_Northei
Kreissparkasse Ahreweiler	Kreissparkasse_Ahweiler
阿尔滕基兴国王公园	Kreissparkasse_Altenkirch
Kreissparkasse Anhalt-Bitterfeld	Kreisparkasse_Anhalt-Bitterfeld
奥格斯堡国立公园	奥格斯堡市政公园
Kreissparkasse Bautzen	Kreissparkasse_Bautzen
Kreissparkasse Bersenbrück	Kreissparkasse_Bersenbrück

名称	ID
Kreissparkasse Biberach	Kreissparkasse_Biberach
Kreissparkasse Birkenfeld	Kreisparkasse_Birkenfeld
Kreissparkasse Bitburg-Prüm	Kreissparkasse_Bitburg-Prüm
伯布林根国立公园	伯布林根Kreissparkass
Kreissparkasse Börde	Kreissparkasse_Börde
Kreissparkasse Cochem-Zell	Kreissparkasse_Cochem-Zell
Kreissparkasse Döbeln	Kreissparkasse_Döbeln
杜塞尔多夫国立公园	杜塞尔多夫市立公园
Kreissparkasse Eichsfeld	Kreissparkasse_Eichsfeld
Kreissparkasse Eslingen-Nürtingen	Kreissparkasse_Eslingen-Nurtingen
Kreissparkasse Euskerchen	Kreisparkasse_Euskirchen
弗罗伊登施塔特国王公园	Kreissparkasse_Freudenstadt
Kreissparkasse Garmisch-Part。	Kreissparkasse_Garmisch-Part。
盖尔恩豪森 Kreissparkasse	Kreissparkasse_Gelnhausen
Kreissparkasse Gotha	Kreissparkasse_Gotha
Kreissparkasse Grafschaft Bentheim	Kreissparkasse_Grafschaft_Bentheim
Kreissparkasse Grafschaft Diepholz	Kreissparkasse_Grafschaft_Diepholz
Kreissparkasse Groß-Gerau	Kreissparkasse_Groß-Gerau
格平根 Kreissparkasse	Göppingen Kreissparkasse
Kreissparkasse Halle（威斯特法伦州）	Kreissparkasse_Halle_（威斯特法伦州）

名称	ID
海登海姆Kreissparkasse	Kreissparkasse_Heidenhe
Kreissparkasse Heilbronn	Kreissparkasse_Heilbron
Kreissparkasse Heinsberg	Kreisparkasse_Heinsberg
Kreissparkasse Herzotum Lauenburg	Kreissparkasse_Herzotum_Lauenburg
Kreissparkasse Hildburghausen	Kreissparkasse_Hildburghausen
Kreissparkasse Kaisslautern	Kreissparkasse_Kaisslautern
凯尔海姆 Kreissparkasse	Kreissparkasse_Kelhe
Kreissparkasse Kusel	Kreissparkasse_Kusel
科隆 Kreissparkasse	Kreissparkasse_Köln
Kreissparkasse Limburg	Kreissparkasse_Limburg
路德维希堡 Kreissparkasse	Kreissparkasse_路德维希堡
Kreissparkasse Mayen	Kreissparkasse_Mayen
Kreissparkasse Melle	Kreissparkasse_Melle
Kreissparkasse Miesbach-Tegernsee	Kreissparkasse_Miesbach-Tegernsee
慕尼黑施塔恩贝格 Kreissparkasse	Kreissparkasse_Munchen_Starnberg
北豪森 Kreissparkasse	Kreissparkasse_Nordhaus
Kreissparkasse Ostalb	Kreissparkasse_Ostalb
Kreissparkasse Osterholz	Kreissparkasse_Osterholz
Kreissparkasse Peine	Kreissparkasse_Peine
拉文斯堡 Kreissparkasse	拉文斯堡Kreissparkasse

名称	ID
罗伊特林根 Kreissparkasse	罗伊特林根 Kreissparkasse_
Kreissparkasse Rhein-Hunsrück	Kreissparkasse_Rhein-Hunsrück
Kreissparkasse Rhein-Pfalz	Kreissparkasse_Rhein-Pfalz
Kreissparkasse Rottweil	Kreissparkasse_Rottweil
Kreissparkasse Saale-Orla	Kreissparkasse_Saale-Orla
萨尔费尔德-鲁多尔斯塔特 Kreissparkasse	Kreissparkasse_Saalfeld-Rudolstadt
Kreissparkasse Saarlouis	Kreissparkasse_Saarloui
Kreissparkasse Saarpfalz	Kreissparkasse_Saarpfalz
Kreissparkasse Schluchtern	Kreissparkasse_Schluchtern
Kreissparkasse Schongau	Kreissparkasse_Schongau
Kreissparkasse Schwalm-Eder	Kreissparkasse_Schwalm-Eder
Kreissparkasse Soltau	Kreisparkasse_Soltau
Kreissparkasse St. Wendel	Kreissparkasse_St._Wendel
Kreissparkasse Stade	Kreissparkasse_Stade
施泰因福特 Kreissparkasse	Kreissparkasse_Steinfurt
Kreissparkasse Stendal	Kreissparkasse_Stendal
Kreissparkasse Syke	Kreissparkasse_Syke
Kreissparkasse Traunstein-Trostberg	Kreissparkasse_Traunstein-Trostberg
图特林根 Kreissparkasse	Kreissparkasse_Tuttling
图宾根 Kreissparkasse	Kreissparkasse_Tubingen

名称	ID
Kreissparkasse Verden	Kreisparkasse_Verden
Kreissparkasse Vulkaneifel	Kreissparkasse_vulkaneifel
魏布林根 Kreissparkasse	Kreissparkasse_Weibling
Walsrode Kreissparkasse	Kreissparkasse_Walsrode
Weilburg Kreissparkasse	Kreissparkasse_Weilburg
Kreissparkasse Wesermunde-Hadeln	Kreissparkasse_Wesermunde-Hadeln
韦斯特瓦尔德 Kreissparkasse	Kreissparkasse_Westerw
Kreissparkasse Wiedenbrück	Kreissparkasse_Wiedenbrück
Kremmerhuset	Kremmerhuset
克雷普市场	克雷普市场
克雷斯塔	克雷斯塔
Kriser 的天生宠物	Kriser's_natural_Pet
Krispy Kreme	Krispy_Kreme
Krist	Krist
克里斯特石油公司	Krist_Oil_Compan
Kro's Nest	kro's_nest
克罗格	克罗格
克罗格燃料中心	Kroger_Fuel_Center
克罗格市场	克罗格市场
克罗格 Marketplace 燃料中心	克罗格_市场_燃料中心

名称	ID
克罗格市场 Marketplace	克罗格_市场_药房
克罗格药房	克罗格_药房
Krogh Optik	Krogh_Optikk
Tía Petra 的 Kroketeria	Kroketeria_de_tía_Petra
Kronans 药店	Kronans_Apotek
克罗诺斯	克罗诺斯
Kronverk 电影院	Kronverk_Cinema
Kropka	Kropka
Kroshka Kartoshka	Kroshka_Kartoshka
Kroüp	Kroüp
Kruidvat	Kruidvat
KrungThai-Axa 人寿保险	Krungthai-Axa_life_Insuran
克魯日卡	克魯日卡
克雷洛夫斯基伊	克雷洛夫斯基伊
Krys	Krys
克里斯塔尔	克里斯塔尔
Krönan	Krönan
Ktronix	Ktronix
Kua Aina	Kua_Aina
Kuanbandeng	Kuanbandeng

名称	ID
宽甸农村商业银行	宽甸_农村_商业银行
光光超市	Kuanguang_Super_Mar
快悦快递	Kuayue Express
库班信用	Kuban_Credit
库伯物流	Kuber_Logist
库钦故居	Kuchen_House
Kuchyne Sykora	Kuchyne_Sykora
Kuchyne gorenje	kuchyne_gorenje
Kuchyne oresi	kuchyne_oresi
Kucuk Ev	Kucuk_Ev
九宝物咖啡厅	Kudamono_Cafe
Kudsan 面包店和咖啡	Kudsan_Bakery_&_Coffee
Kudu	Kudu
Kueijang Ramen	Kueijang_Ramen
Kuidon	Kuidon
Kulcs Patika	Kulcs_Patika
Kulechaowan	Kulechaowan
库尔姆巴赫银行	Kulmbacher_Bank
Kult Olymp & Hades	Kult_Olymp_&_Hades
Kum & Go	Kum_&_Go

名称	ID
熊本第一信用金库	熊本_第一_信用金库
熊鹿市	熊鹿市
Kumbakonam 过滤咖啡	Kumbakonam_Filter_Coffee
Kumfs	Kumfs
Kumo Kumo	Kumo_Kumo
金皮尔盒子	Kumpir_Box
功夫陶罐	Kung_Fu_Clay_Pot
功夫茶店	功夫茶店
功夫厨房	功夫厨房
功夫	功夫
昆仑饭店	昆仑酒店
Kunooz Alseha Pharmacy	Kunooz_Alseha_Pharmacy
昆山农村商业银行	昆山_农村_商业银行
昆怡酒店	昆怡_酒店
郭元叶	郭元叶
国泰	国泰
Kupets	Kupets
库尔多餐厅	库尔多餐厅
吴信用金库	Kure_Shinkin_Bank
Kuriniy DOM	kurini_dom

名称	ID
Kurskprombank	Kurskprombank
Kurt Geiger	Kurt_Geiger
Kusuri-no-LOV	Kusuri-no-LOV
Kuveytturk	Kuveytturk
科威特金融大厦	科威特_Finance_House
科威特石油	科威特_石油
桑名三重信用金库	Kuwana_mie_Shinkin_Bank
Kuya J 餐厅	Kuya_J_Restaurant
库兹涅茨克商业银行	库兹涅茨克商业银行
Kvartal	Kvartal
《土地法案》	Kverneland_Bil
Kvetiny Novak	Kvetiny_Novak
Kvickly	Kvickly
Kvänum	Kvänum
光州银行	光州银行
Kwantum	Kwantum
Kwik Fill	Kwik_Fill
Kwik King	Kwik_King
Kwik Kopy	kwik_Kopy
Kwik Mart	Kwik_Mart

名称	ID
Kwik SPAR	kwik_spar
Kwik Shop	Kwik_Shop
Kwik Store	Kwik_Store
Kwik Trip	Kwik_Trip
Kwik Way	Kwik_Way
Kwik-Fit	Kwik-Fit
Kyffhauserspk。Artern-Sondershshan。	Kyffhauserspk。 _Artern-Sondershsn。
Kyivstar	Kyivstar
庆南银行	庆南银行
京都碗	Kyoto_Bowl
京都中央信用金库	京都中央信用金库
京都北斗信用金库	京都_北斗_信用金库
基里亚德	基里亚德
基里亚德酒店	Kyriad_Hotel
基里亚德奇妙酒店	Kyriad_Marvelous_Hotel
基里亚德尊尚酒店	kyriad_Prestige
Kysten Rundt	Kysten_Rundt
九州 Rokin	九州_Rokin
九州石友	九州_Sekiyu
Kahyra	Kahyra

名称	ID
科隆银行	Kölner_Bank
Königsbauer	Königsbauer
Körny-Bokod Takarék	Körny-Bokod_Takarék
Küche & Co	Küche_&_Co
Kuchen Aktuell	Küchen_Aktuell
Kuchen Quelle	Küchen_Quelle
Küchentreff	Küchentreff
Kutahya Porselen	Kutahya_Porselen
L H R	L_H_R
L&L 夏威夷烧烤	L&L_Hawaiian_Barback
L'Arche	L'Arche
L'Artigiano	L'Artigiano
L'Auto e. 勒克莱尔	l'auto_e._leclerc
L'Etual'	L'Etual'
欧舒丹	欧舒丹
L'Occitane au Brésil	l'Occitane_au_brésil
L'orange Bleu	L'Orange_Bleu
L'Osteria	L'Osteria
l'angfarm 商店	l'angfarm_store
l'eau Vive	l'eau_vive

名称	ID
l'ombre de La Lune	l'ombre_de_la_lune
左呼吸	左呼吸
L.L. Bean	L.L._Bean
LA BOUCHERIE	LA_BOUCHERIE
LA CANTERA	LA_CANTERA
洛杉矶市	LA_CITY
LA ESPIGA	LA_ESPIGA
洛杉矶健身	洛杉矶健身
LA GRILLARDIERE	LA_GRILLARDIERE
温德姆拉昆塔酒店	温德姆拉昆塔酒店
啦尚音	LA_SHANG_YIN
LA TAHONA	LA_TAHONA
LA TAQUERIA	LA_TAQUERIA
LA TORRE	LA_TORRE
LA VIE SAINE	LA_VIE_SAINE
洛杉矶减肥中心	la_weight_loss_Center
LACOLADA	LACOLADA
LACOSTE	LACOSTE
LADA Detal'	Lada_detal'
女士系列	女士系列

名称	ID
LAFE 实验室	lafe_laboratório
LAGA	LAGA
拉古纳	拉古纳
LAINEGE	LAINEGE
莱商银行	莱商银行
泄漏	泄漏
LALIPCO	LALIPCO
兰博基尼	兰博基尼
LAMBSON 的雇用	LAMBSON'S_HIRE
灯具加号	LAMPS_PLUS
LAMY	LAMY
拉纳克	拉纳克
LANCEL	LANCEL
蓝旗亚	蓝旗亚
LANCY FROM25	LANCY_ FROM25
路虎	LAND_ROVER
兰迪森	兰迪森
LADROME	LADROME
朗豪酒店及度假村	LANGHAM_PLACE_酒店和度假村
拉诺酒店	LANO_HOTEL

名称	ID
拉诺国际酒店	拉诺国际酒店
LANVIN	LANVIN
老田庄	老田庄
LAOFULAND	LAOFULAND
哈哈咖啡	LAOO_COFFEE
LAPEYRE	LAPEYRE
拉里交易所	LARI_XCHANGE
激光 2000	LASER_2000
激光 ELETRO	LASER_ELETRO
LASKASAS	LASKASAS
LASSA	LASSA
LASSUS	LASSUS
拉美旅行	LATAM_Travel
拿铁	拿铁
劳拉阿什利	劳拉·阿什利
月桂树餐厅	LAUREL_餐厅
熔岩	熔岩
LAZURIT	LAZURIT
LAZZONI	LAZZONI
LBC	LBC

名称	ID
LBG 牙科	LBG_Dental
LBS 金融 CU	lbs_financial_cu
LBX 药房	LBX_PHARMACY
LC 威基基	lc_Waikik
LCBO	LCBO
LCL	LCL
旧市场	LD MARKET
LE CROBAG	LE_CROBAG
LE GRENIER	_GRENIER
MEUBLE 巨人	_GEANT_DU_MEUBLE
LE LIS BLANC	LE_LIS_BLANC
LE MUTANT	_MUTANT
LE vélo STAR	le_velo_star
租赁 BANCOLDEX	LEASING_BANCOLDEX
李	李
礼顿酒店	LEEDEN_HOTEL
LEEN BAKKER	LEEN_BAKKER
合法	合法
乐高	乐高
LEHMANNS FACHBUCHHANDLUNG	LEHMANNS_FACHBUCHHANDLUNG

名称	ID
花环花园	LEI_GARDE
LEKO	LEKO
LEKRUS	LEKRUS
LENE V	LEN_V
镜头制作者	LENS_CRAFTERS
LENTA	LENTA
狮子座	狮子座
LEO 加油站	leo_petrol_Station
勒罗伊·梅林	LEROY_MERLIN
LES FCU	LES_FCU
LES OPTICIENS 的忠告	LES_OPTICIENS_CONSEILS
LES SCHWAB 轮胎中心	LES_SCHWAB_TIRE_CENTER
LESCO 服务中心	LESCO_服务中心
休闲餐饮	休闲餐饮
LEVNÉ KNIHY	LEVNÉ_KNIHY
LEWIATAN	LEWIATAN
雷克萨斯	雷克萨斯
LFS 电影院	LFS_Cinemas
LG	LG
LG 洗衣休息室	LG_laundry_Loun

名称	ID
LH Bank	LH_bank
李宁	李宁
LIANG SHE HAN BUY	LIANG_SHE_HAN_BUY_GOOD
亮友餐饮	LIANG_YOU_CARTIN
利伯塔德	利伯塔德
解放者	解放者
LIBRO	LIBRO
LIDL	LIDL
生命医疗集团	生命医疗集团
生活方式酒店	生活方式酒店
LIGA 银行	Liga_bank
LIGOIL	LIGOIL
丽江河餐厅	丽江河餐厅
像茶店	点赞茶店
LIKELAI	LIKELAI
石灰耀斑	石灰色火花
丽娜的	丽娜的
LINDEX	LINDEX
LINE-X	LINE-X
LINEA 之旅	LINEA_TOURS

名称	ID
临商银行	临商银行
狮子	狮子
LIPO	LIPO
LIQUN	LIQUN
LISCA	LISCA
LIVRARIA ALMEDINA	LIVRARIA_ALMEDINA
丽兹酒店	Liz_Hotel
LIZ MINELLI	LIZ_MINELLI
LIZARRAN	LIZARRAN
LJ Hooker	lj_hooker
LK	LK
LK Bennett	LK_Bennett
劳埃德银行	劳埃德银行
劳埃德国际银行	劳埃德国际银行
LMG 保险	LMG_Insurance
LMI	LMI
LO DE CARLITOS	LO_DE_CARLITOS
哈哈哈哈哈	LO_TENGO
锁和锁	锁和锁
LOEWE	LOEWE

名称	ID
登录	登录
LOHAS	LOHAS
LOHBS	LOHBS
LOHIA	LOHIA
LOHO	LOHO
LOJA DAS MEIA	LOJA_DAS_MEIAS
Loko-bank	Loko-bank
棒棒糖巴黎	棒棒糖_巴黎
LOLY BALAS	LOLY_BALAS
伦敦派公司	LONDON_PIE_CO
长富宫酒店	长富宫大酒店
长亚洲JU	LONG_SIAN_JU
浪琴	浪琴
龙江银行	龙江银行
LOPESAN	LOPESAN
哈罗	哈罗
LOS AMORES	LOS_AMORES
LOS PALETEROS	LOS_PALETEROS
LOT 手机	Lot_mobiles
LOTOS	LOTOS

名称	ID
乐透最佳	Lotos_Optima
家很多	LOTS_HOME
乐天电影院	LOTTE_CINEMA
乐天百货	乐天百货
乐天市场 999	Lotte_market_999
乐天玛特	Lotte_mart
乐透	乐透
莲花	莲花
路易斯·皮恩	LOUIS_PION
路易威登	路易威登
洛夫森银行	LOVCEN_BANKA
爱宝贝工作坊	爱心宝贝工作坊
爱情共和国	爱情共和国
爱情世界	爱情世界
爱与爱	爱与爱
LOXAM	LOXAM
LOZO 购物中心	LOZO_SHOPPINGMALL
LPL 财经	lpl_Financi
LS（英国）	LS_ (英国)
LTG	LTG

名称	ID
LTK	LTK
LUBER	LUBER
LUBNERS	LUBNERS
LUBRIDANO	LUBRIDANO
卢塞恩	卢塞恩
露西的	露西的
LUGGI	LUGGI
路易吉·贝托利	LUIGI_BERTOLLI
六福珠宝	六福珠宝
LUKOIL	LUKOIL
卢克石油商店	Lukoil_shop
LUMIERE	LUMIERE
卢米埃尔展馆	LUMIERE_PAVILIONS
LUNDBERGS Handskar & VASKOR	LUNDBERGS_HANDSKAR_&_VESKOR
龙美窗帘	LUNGMEI_窗帘
LUNILABO	LUNILABO
LUPO	LUPO
LUX MED	LUX_MED
LEXAUTO	LEXAUTO
LUYU	LUYU

名称	ID
LVT LOS VESTIDO	LVT_LOS_VESTIDOS
LXR 酒店及度假村	LXR_Hotels_&_resorts
LYN	LYN
LYNK	LYNK
La 14	La_14
La Anonima	La_Anonima
La Baguette	La_Baguette
La Bamba	La_Bamba
邮政银行	La_Banque_Postale
拉贝尔省	拉贝勒省
La Bodega de la Trattoria	la_bodega_de_la_trattoria
La Bonga del Sinu	la_bonga_del_sinu
洛杉矶精品店	洛杉矶精品店
La Boíte a Pizza	la_boite_à_pizza
La Brasa Roja	La_Brasa_Roja
La Brioche Dorée	La_Brioche_Dorée
La Cabine T Sans Fil	La_Cabine_T_Sans_Fil
La Cage aux Sp	la_cage_aux_Sports
La Caja de Canarias	la_caja_de_Canarias
洛杉矶国会大厦 FCU	la_capitol_fcu

名称	ID
La Caravana	La_Caravana
La Casa de Toño	la_casa_de_toño
La Casa del Alfajor	la_casa_del_alfajor
La Cazuela	La_Cazuela
La Chelinda	La_Chelinda
La Chilanguita	La_Chilanguita
La Cigale	La_Cigale
La Clueca	La_Clueca
La Colegiala	La_Colegiala
洛杉矶大陆	洛杉矶大陆
La Criée	La_Criée
La Croissanterie	La_Croissanterie
La Despensa 快车	La_Despensa_Expres
La Destileria	La_Destileria
《经济》	La_Economía
拉埃斯特雷拉	La_Estrella
欧罗巴	La_Europea
La Fabrica del Cartucho	La_fabrica_del_cartucho
La Foir'Fouille	La_Foir'Fouille
La Fourmi	La_Fourmi

名称	ID
La Gas	La_Gas
La Genovesa	La_Genovesa
La Gran Bodega	La_Gran_Bodega
拉格兰德罗盖里亚	La_Gran_Drogueria
La Grande Recré	La_Grande_Recré
La Gull's Court	la_gull's_Court
La HALLE	La_Halle
La Halle aux Chaussures	la_halle_aux_Chaussures
La Ibérica	La_Ibérica
La Leña	La_Leña
啦萝拉	La_Lola
《黑手党》	La_Mafia
La Mafia se seenta a la mesa	la_mafia_se_se_sienta_a_la_Mesa
La Maison Ensoleille Table	La_Maison_Ensoleille_Table
La Mallorquina	La_Mallorquina
La Manon	La_Manon
洛杉矶大厦	La Mansion
La Martina	La_Martina
La Michoacana	La_Michoacana
La Michoacana 肉类市场	La_Michoacana_Meat_Market

名称	ID
La Mie Caline	La_mie_Caline
La Mode 眼镜	La_Mode_Eyear
La Mulera	La_Mulera
La Nave de Portico	la_nave_de_portico
La New	La_New
La Oca	La_Oca
巴黎人	La_Parisien
La Parrilla 墨西哥餐厅	La_Parrilla_墨西哥餐厅
Parrilla de Pollos Planes	la_parrilla_de_pollos_Planes
La Pasiva	La_Pasiva
La Pasteria	La_Pasteria
La Pataterie	La_Pataterie
香水店	La_Perfumería
La Perla	La_Perla
La Perla Gris	La_Perla_Gris
小学院	La_Petite_Academy
La Petrölea	La_Petrölea
尼科披萨	la_pizza_de_nico
拉普莱斯	La_Place
La Plateforme du Bâtiment	la_plateforme_du_bâtiment

名称	ID
La Ponderosa	La_Ponderosa
La Porchetta	La_Porchetta
La Prairie	La_Prairie
La Rapa	La_Rapa
La Rebaja	La_Rebaja
La Red Turística	La_red_Turística
拉罗萨德尔蒙特	La_Rosa_del_Monte
La Salsa	La_Salsa
La Salsa 新鲜墨西哥烧烤	La_Salsa_Fresh_Mexican_Grill
La Santé	La_Santé
La Senza	La_Senza
La Sirena	La_Sirena
La Station U	La_Station_U
La Sureña	La_Sureña
La Table	La_Table
La Tagliatella	La_Tagliatella
La Tasca	La_Tasca
La Taverne de Maítre Kanter	la_Taverne_de_maitre_kanter
La Tienda de Dolar	la_tienda_de_de_Dolar
La Tienda de los Möviles	la_tienda_de_los_möviles

名称	ID
La Vaca	La_Vaca
La Vara	La_Vara
La Veneciana	La_Veneciana
La Ventana de Colores	la_ventana_de_colores
生活克莱尔	La_Vie_Claire
玫瑰人生	La_Vie_En_Rose
La Vitoriana	La_Vitoriana
La Wash	La_Wash
La Yarda	La_Yarda
La compagnie des petits	la_compagnie_des_petits
洛杉矶模式	LA_mode
Laz-Boy	Laz-Boy
La-Z-Boy 家具画廊	La-Z-Boy_Furniture_G
LaKaffa	LaKaffa
LaMar's Donuts	LaMar's_Donuts
LaMorning	LaMorning
LaRosa的	LaRosa的
LaRosa's Pizzeria	LaRosa's_Pizzeria
LabCorp	LabCorp
Labchecap	Labchecap

名称	ID
标签 Ritu Kumar	Label_Ritu_Kumar
Label'Vie	Label'Vie
《劳工发现者》	Labor_Finders
劳动准备就绪	Labor_Ready
Laboral kutxa	Laboral_kutxa
美国实验室公司	美国实验室_公司_美国
生物分析实验室	实验室_生物分析
南方生物中心实验室	实验室_生物中心_南部
加斯帕实验室	Laboratório_Gaspar
Leme 实验室	Laboratório_Leme
保罗·阿泽维多实验室	Laboratório_Paulo_Azevedo
罗珊娜·多尔·苏亚雷斯实验室	Laboratório_Roseanne_Dore_Soares
露丝·布拉桑实验室	Laboratório_Ruth_Brazao
圣马科斯实验室	Laboratório_Sao_Marcos
温曼实验室	Laboratório_Weinmann
Gilson Cidrim 实验室	实验室_Gilson_Cidrim
标准实验室	Laboratorios_Padrano
实验室 a+	Labs_a+
Lacesar	Lacesar
Lactuca	Lactuca

名称	ID
拉达	拉达
Ladbrokes	Ladbrokes
LadeFoxx	LadeFoxx
女士锻炼快车	女士_Workout_Express
Ladurée	Ladurée
拉杜什卡	拉杜什卡
女士足柜	Lady_Foot_Lock
沙姆夫人	Lady_Sharm
鹳夫人	Lady_Stork
女士健身	女士健身
LadyM 甜点	女士_甜点
Ladyamericana	Ladyamericana
Lafu	Lafu
Lafuma	Lafuma
Lagerhaus	Lagerhaus
Lagkagehuset	Lagkagehuset
拉合尔文法学校	Lahore_Grammar_School
来必宝	来必宝
莱雅百货	莱雅_百货_大楼
湖城银行	湖城银行

名称	ID
湖边	湖边
莱克兰银行	莱克兰银行
莱克斯天然气公司	Lakes_Gas_Co
湖岸学习	湖岸_学习
湖边	湖边
Lakmé 沙龙	Lakme_Salon
叻沙小屋	Laksa_Shack
Lala Arbon 合作银行	Lala_arbon_co-op_Bank
Laline	Laline
兰博基尼酒店及度假村	兰博基尼_酒店及度假村
Lampehuset	Lampehuset
Lampemagasinet	Lampemagasinet
兰美李庄肉	Lan_Mei_Li_Zhuang_meat
Lancerto	Lancerto
兰奇超市	Lanchi_Mart
Lanchão & Cia	Lanchão_&_Cia
Lanco FCU	lanco_fcu
土地储备	Land_Bank
菲律宾土地银行	菲律宾土地银行
林肯之地 CU	Land_of_Lincoln_cu

名称	ID
兰达尔	兰达尔
陆运	陆运
陆上的 Corten	陆路_Corten
陆运 Minim	陆上_最小
陆上的 Morpheus	Landbound_Morphe
Landbouwkredit	Landbouwkredit
Landbackerei Ihle	Landbackerei_Ihle
Landbackerei Sommer	Landbackerei_Sommer
兰德斯超市	兰德斯_Superstore
巴登符腾堡州立银行	Landesbank_巴登-符腾堡州
柏林州立银行	柏林州立银行
Landesbank Kreissparkasse Sigmarin。	Landesbank_Kreissparkasse_Sigmarin。
萨尔兰德斯银行	Landesbank_Sar
Landesbausparkasse	Landesbausparkasse
奥尔登堡州立公园	LandesSparkasse_zu_Oldenburg
兰德酒店	兰德酒店
兰迪	兰迪
Landidyll	Landidyll
CU 着陆	Landings_cu
兰迪斯超市	兰迪斯_超市

名称	ID
标记	标记
地标 CU	Landmark_cu
地标保险经纪人	Landmark_保险_经纪人
兰德里的海鲜屋	Landry'seafood_House
陆地尽头	Lands'End
Landsbankinn	Landsbankinn
兰茨贝格-阿默湖银行	兰茨贝格-阿默西银行
风景时尚	风景_时尚
兰德咖啡厅	Lander_Café
陆风	陆风
Landzeit	Landzeit
莱恩·布莱恩特	Lane_Bryant
莱恩·克劳福德	莱恩_克劳福德
Lang Backerei-Konditorei	Lang_Backerei-Konditorei
朗廷酒店	Langham_Hotel
Langley FCU	langley_fcu
朗文教育	朗文_教育
Lanidor	Lanidor
Lanidor KIDS	Lanidor_Kids
Lanidor LOFT	Lanidor_Loft

名称	ID
拉诺罗	拉诺罗
拉努酒店	Lanou_Hotel
Lao Ke Fu	Lao_Ke_Fu
老太元厨房	老挝_太原_厨房
LaoViet 银行	LaoViet_银行
老坝和浙江地方美食	Laoaba_&_浙江_地方菜
Laobanlianshangyu	Laobanlianshangyu
老厂美食	老昌_食品
劳定峰	劳定峰
老凤阁品珍轩	Laofengge_品珍轩
老家烤肉	老家_Bakemeat
老牌坊	老牌坊
劳胜兴	劳胜兴
老土易昌火锅	Laotu_Echang_Hot_Pot
老王火锅	老王_火锅
老香楼	老香楼
老阳铭源眼镜	老杨_铭源_眼镜
老正兴餐厅	老正兴_餐厅
拉帕纳	拉帕纳
翻领	翻领

名称	ID
拉佩尔大道	Lapeyre_Drive
Lapin House	Lapin_House
拉皮兹·洛佩兹	拉皮兹_洛佩兹
Laptok	Laptok
Lapperre	Lapperre
拉皮亚酒店	Lappia_酒店
Lara Bags	Lara_Bags
Lara Fabio	Lara_Fabio
Laritza D	Laritza_D
Larkspur 酒店	Larkspur_Hotels
Larry's Giant Subs	Larry's_Giant_Subs
Larsen & Toubro	Larsen_&_Toubro
拉森酒店和克罗费里	Larsen_hotel_og_Kroferie
Las Alitas	Las_Alitas
拉斯卡纳斯塔斯	Las_Canastas
Las Gaoneras	Las_Gaoneras
拉斯加塔斯	Las_Gatas
Las nuevas droper	las_nuevas_droper
Las-Knigas	Las-Knigas
拉萨尔银行	Lasalle_Bank

名称	ID
Lasefense 餐厅	Lasefense_Resta
Laselva	Laselva
澳大利亚激光诊所	澳大利亚激光诊所
LaserAway	LaserAway
LasikPlus 视觉中心	LasikPlus_愿景中心
最后一英里解决方案	最后一英里_解决方案
Lastentarvike	Lastentarvike
Lastigim	Lastigim
拉斯蒂克公园	Lastik_Park
Lastik Vs	Lastik_Vs
Latah CU	latah_cu
Lateko	Lateko
拉丁美洲裔女子	拉丁美洲裔女子
拉丁裔服装店	Latina_服装店
拉脱维亚北美自由贸易协定	拉脱维亚_北美自由贸易协定
Laubman & Pank	Laubman_&_Pank
Laudamotion	Laudamotion
劳拉·阿什利家居专卖店	Laura_ashley_Home_Specialty_Stor
劳拉·塞科德	Laura_Secord
劳伦	劳伦

名称	ID
劳伦电影院	劳伦_电影院
洛朗	洛朗
劳伦森银行	Laurentian_Bank
劳里·卢米埃尔	Laurie_Lumière
熔岩	熔岩
熔岩新鲜	LavaFresh
Lava Lava el Delfín	Lava_lava_el_delfin
LavaXpres	LavaXpres
Lavalux	Lavalux
薰衣草	薰衣草
薰衣草酒店	Lavande_Hotel
Lavard	Lavard
Lavatutto	Lavatutto
Lavazza	Lavazza
拉文纳酒店及度假村	Lavenna_酒店及度假村
Lavoisier Laboratório 和 Imagem	lavoisier_laboratório_e_imagem
Lavpris Benzin	Lavpris_Benzin
Lawine	Lawine
草坪医生	草坪医生
劳森	劳森

名称	ID
Lawson 108	Lawson_108
劳顿斯	劳顿斯
律师职称	律师_职务
Laya Burger	拉亚汉堡
Laziz 披萨	懒人披萨
Le Boeuf Jardinier	Le-Boeuf_Jardinier
乐城堡	乐城堡
Le Coq Wine&Bistro	Le_Coq_Wine&Bistro
Le Cremier	Le_Cremier
Le Creuset	Le_Creuset
Le Croissant	Le_Croissant
Le Furet du Nord	Le_furet_du_Nord
Le Grand Panier B	Le-Grand_Panier_Bio
Le Groupe Jean Cutu	Le-Groupe_Jean_Coutu
Le Kiosque a Pizz	le_kiosque_à_pizza
艾美酒店	le_Meridien
Le Pain Francai	LePain_Francais
Le Pain Quotidien	Le_Pain_Quotidien
水果天堂	le_paradis_du_fruit
Petit Bofinger	Le_Petit_Bofinger

名称	ID
Le Petit Detail	Le_Petit_Detail
Le Polka	Le_Polka
Le Postiche	Le_Postiche
Le Repaire de Bacchus	le_repaire_de_Bacchus
Le Repas	Le_Repas
Le Temps des Cerises	_temps_des_cerises
乐健康	_健康
乐饼干	_Biscuit
Le comptoir de Mathilde	lecomptoir_de_mathilde
Le jardin des Fleurs	LeJardin_des_Fleurs
Le marché d'a corté	lemarche_d'a_côté
LeBags	LeBags
LePeep 餐厅	LePeep_餐厅
Leadco 社区 CU	leadco_community_cu
领袖	领袖
领导者价格	领导者_价格
领先	领先
领先优势	领先优势
前沿通信	领先_前沿_通信
世界领先的小型酒店	世界领先的小型酒店

名称	ID
Leal	Leal
精益农村商业银行	Lean_Rural_Compical_Bank
飞跃信使	Leapards_Courier
Leapmotor	Leapmotor
学习教育	学习_教育
学习快车玩具	学习_Express_Toys
LearningRx	LearningRx
LeasePlan	LeasePlan
黎巴嫩伊斯兰银行	黎巴嫩_伊斯兰银行
黎巴嫩瑞士银行	黎巴嫩_瑞士银行
Lebeau Vitres d'autos	Lebeau_Vitres_d'Autos
Lebes	Lebes
Lec Lee	Lec_Lee
Lecado Doceria	Lecadô_Doceria
lechu.ru	lechu.ru
莱西科	莱西科
LedCan 科技	LedCan_Tecnología
Leda	Leda
Ledo 披萨和意大利面	ledo_Pizza_and_Pasta
李·库珀	lee_Cooper

名称	ID
李·赫希特·哈里森	Lee_Hect_Harrison
Lee Myles 变速箱与汽车保养	lee_myles_transmissions_&_autocar
李·罗伊·塞尔蒙的	Lee_Roy_Selmon's
Lee Wrangler	Lee_Wrangler
Lee 的著名鸡肉	Lee's_famous_Chicken
Lee 的三明治	Lee's 三明治
Leeann Chin	Leann_Chin
利兹和霍尔贝克建筑协会	利兹_&_Holbeck_Building_Society
利兹建筑协会	利兹建筑协会
Leeji 传统茶店	Leeji_Tradial_Tea_Shop
Leek United 建筑协会	Leek_United_Building_
里拉宫	Leela_Palace
Lees Inn	Lees_Inn
Leezen	Leezen
Lefdal	Lefdal
Lefit	Lefit
传统	传统
传统社区 FCU	Legacy_Community_FCU
传统酒店及度假村	Legacy_酒店_度假村
合法的海鲜	合法_海鲜_食品

名称	ID
Legekaeden	Legekaeden
军团	军团
乐高教育中心	乐高教育中心
Leiner	Leiner
Leiser	Leiser
休闲俱乐部	休闲俱乐部
漯阳农村商业银行	漯阳农村商业银行
LekOptTorg	LekOptTorg
Lekar'	Lekar'
Lekarna Devetsil	Lekarna_devetsil
卢布尔雅那药店	Lekarna_卢布尔雅那
Leke Jean's	Leke_Jean's
Lele Yumurta	Lele_Yumurta
Lelecha	Lelecha
Leman Kultür	Leman_Kültür
勒芒租车	Lemans_Rent_a_Car
柠檬鱼	Lemon_Fish
柠檬酒店	Lemon_Hotel
柠檬树酒店	柠檬树酒店
柠檬水	柠檬水

名称	ID
Lemurrr	Lemurrr
LenOblFarm	LenOblFarm
Lenard's	Lenard's
莱纳德的鸡	Lenards_Chicken
LendingTree	LendingTree
Lenmed	Lenmed
Lennar Homes	Lennar_Homes
伦诺克斯 PartsPlus	Lennox_ PartsPlus
Lenny's SubShop	Lenny's_sub_shop
莱诺	莱诺
联想	联想
Lentz USA 消声器刹车和减震器	Lentz_usa_mufflers_brakes_&_Shocks
狮子座咖啡	leo_Coffee
Leo Madeiras	Leo_Madeiras
莱昂	莱昂
Leon's	Leon's
伦纳德的	伦纳德的
莱昂纳多	莱昂纳多
莱昂纳多酒店	莱昂纳多_酒店
列奥尼达斯	列奥尼达斯

名称	ID
Leonisa	Leonisa
Leopaard	Leopaard
狮子座	狮子座
Lerner 商店	Lerner_Shops
Les 3 Brasseurs	Les_3_Brasseurs
Les Briconautes	Les_Briconautes
Les Castels	Les_Castels
Les Compagnons des Saison	Les_compagnons_des_Saisons
Les Comptoirs 赌场	Les_Comptoirs_Casino
Les Domaines Qui Montent	Les_Domaines_Qui_Montent
Les Fils a Maman	les_fils_à_maman
莱斯米尔斯	Les_Mills
Les Mousquetaires	Les_Mousquetaires
Les Relais d'Alsace	Les_relais_d'Alsace
Les Routies	Les_Routiers
莱斯康	莱斯康
乐山市商业银行	乐山城市商业银行
乐诗派	乐诗派
Leslie 的游泳池用品	Leslie's 游泳泳池用品
莱索托邮政银行	莱索托_邮政银行

名称	ID
Letz Sushi	Letz_Sushi
Leumit	Leumit
洛伊特基歇尔银行	Leutkircher_Bank
Levi's	Levi's
Levingers 干洗店	Levingers_干式清洁剂
Levobank	Levobank
Levoberezhniy	Levoberezhniy
刘易斯	刘易斯
刘易斯·克拉克 CU	Lewis_clark_cu
列克星敦	列克星敦
Ley	Ley
Leysieffer	Leysieffer
Lezizhan CigKofte	Lezizhan_ CigKofte
li'l Cricket	Li'l_cricket
LiHO	LiHO
Liana	Liana
Liandongyun Go	Liandongyun_Go
联丰农村银行	联丰农村银行
梁亮郑霞	梁亮正霞
Liangeqinghuoguogongsi	Liangeqinghuoguogongsi

名称	ID
良好中餐	良好_中国_食品
良好咖啡	良好_咖啡
良品朴子	Liangpin_Puzi
凉山州交通银行	凉山州_地级_商业银行
良子	良子
联华/世纪联华超市	联华/世纪联华_超市
连佳	连佳
连胜	连胜
联讯证券	联讯证券
连云港东方农村商业银行	连云港东方农村商业银行
liaoji Bangbang	Liaoji_Bangbang
辽宁大石桥龙凤银行	辽宁_大石桥_龙丰银行
辽宁大瓦农村商业银行	辽宁_达瓦_农村_商业银行
辽宁盖州农村商业银行	辽宁_盖州_农村_商业银行
辽宁省农村信用社	辽宁农村信用合作社
辽沈银行	辽沈银行
辽源农村商业银行	辽源_农村_商业银行
自由银行	自由银行
利伯特尔	利伯特尔
自由	自由

名称	ID
自由美国商店	Liberty_American Store
自由银行	自由银行
ETFCU 的 Liberty Financial	Liberty_financial__a_division_of_etfcu
自由健身	自由_健身
自由互助	自由互助
自由加油站	Liberty_Petrol_Stati
自由保险	Liberty_Segu
自由税务服务	自由税服务
自由旅行	Liberty_Tr
天秤座银行	Libra_Bank
Libreco	Libreco
Libri	Libri
Libris	Libris
Libro	Libro
Lic	Lic
Lider	Lider
Lider obuvnoy mir	Lider_obuvnoy_mir
盖子	盖子
列支敦士登土地银行股份公司	Liechtensteinische_LandesBank_ag
LienVietPostBank	LienVietPostBank

名称	ID
Lienzo de los Gazules	Lienzo_de_los_Gazules
力帆汽车	Lifan_Auto
生活	生活
生活咖啡	生活_咖啡
生活酒店	生活酒店
生命医疗	生活_医疗保健
印度人寿保险公司	印度人寿保险公司
生命药房	生活_药房
生命储存	生命存储
生活工作室	生活工作室
生活方式	生活风格
终身健身	终身_时间_健身
救生服	生活制服
LifeWay	LifeWay
LifeWay 基督教商店	LifeWay_Christian Store
生活乐趣	生活乐趣
生活方式	生活方式
生活方式快车	生活方式_Express
生活方式度假村	生活方式_度假村
Lig-Lig	Lig-Lig

名称	ID
西甲出版社	Liga_Press
Direct	照明_直接
LightingP	LightingP
李浩	李浩
丽红	丽红
丽华快餐	丽华_快餐
丽江农村合作银行	丽江_农村_合作银行
Likoufu	Likoufu
小天使	Lil_Angels
Lil'Champ	Lil'_Champ
Liliangui Dabing 配烟熏肉	Liliangui_dabing_with_Smoked_Meat
Lilica & Tigor	Lilica_&_Tigor
漓陵农村商业银行	黎灵_农村_商业银行
Lilla coop	lilla_coop
Lilliput	Lilliput
莉莉英语	Lily_英语
利马	利马
Limanfer	Limanfer
石灰	石灰
Lime 服装店	Lime_服装店

名称	ID
限量品牌	限量版_品牌
柠檬水	柠檬水
LincOne FCU	LincOne_FCU
Lincare	Lincare
林肯	林肯
林肯合作化学家	Lincoln_Co-Op_Chemist
林肯金融集团	林肯金融集团
林肯房地产公司	林肯房地产公司
林肯理工学院	林肯技术研究所
林克拉夫特	林克拉夫特
林德纳	林德纳
Lindt	Lindt
Line	Line_Brew
Lineadecor	Lineadecor
线性的	线性的
床单	床单
衬里	衬里
临汾农村信用社	Linfen_Rural_Credit_Union
凌军教育	凌军_教育
临海农商银行	临海_农村_商业银行

名称	ID
林佳敏南菜	林家_闽南财富
链接	链接
Linos	Linos
临泉农村商业银行	林泉农村商业银行
Linzmaster	Linzmaster
狮城	Lion_City
狮子公司	狮子公司
Lion d'Or	Lion_d'Or
狮子国际银行	狮子会国际银行
狮子王教育	Lion_King_教育
门廊之狮	Lion_of_Porches
狮子的选择	Lion's Choice
Lipetskcombank	Lipetskcombank
Lippi	Lippi
力宝银行	力宝银行
Lipoti Pekség	Lipoti_Pekség
白酒大亨	Liquor_Barons
酒类中心	酒类中心
酒城	Liquor_City
白酒之王	Liquor_King

名称	ID
Liquir St	Liquor_Stax
LiquorShop	LiquorShop
酒园	酒园
澳大利亚利克兰酒庄	Liquorland_澳大利亚
Liqurland NewZealand	Liquorland _ NewZealand
里仁村	丽人村
里斯本社区 FCU	Lisbon_community_FCU
Lisichka	Lisichka
Lissac l'Opticien	Lissac_l'Opticien
Listerhill CU	ListerHill_cu
Listo！	Listo！
小大	Little_Big
小凯撒	Little_Caesars
小厨师	Little_Chef
小艾莉	Little_Elly
小将军	Little_General
小天才	小天才
Little mo&Co.	little_mo&Co。
小蒙古人	小蒙古人
小绵羊	Little_Sheep

名称	ID
小故事	Little_Stories
小味道	Little_Taste
小羔羊	Little_Lamb
Littman Jewelers	Littman_Jewelers
刘乔	Liu_Jo
刘晓	刘晓
刘美	刘美
刘富士	刘富士
六老根大舞台	Liulaogen_dawutai
浏阳农商银行	浏阳_农村_商业银行
Livat	Livat
直播电汇	Live_Wire
热闹的酒店	Lively_酒店
利物浦	利物浦
生活之家	生活之家
生活空间	生活空间
Livraria Cultura	Libraria_Cultura
库里提巴书店	Livrarias_Curitiba
Livre Keisei	Livre_Keisei
丽文	丽文

名称	ID
Liz Claiborne	liz_Claiborne
蜥蜴灌木丛	Lizard's_thicket
LjusExperten	LjusExperten
Lladro	Lladro
拉诺汽油	Llano_Petrol
Llena tu loko	llena_tu_LOKO
Llobet	Llobet
劳埃德药房	劳埃德药房
Lloyds TSB	Lloyds_TSB
LloydsApotek	LloydsApotek
LloydsPharma	LloydsPharma
Lmg	Lmg
Lml 在印度	Lml_India
Ln 酒店业	Ln_Hospit
Lnp	Lnp
充实的生活	生活的负担
Loaf 'N Jug	Loaf_'N_Jug
Lob	Lob
Loblaw	Loblaw
当地燃料	本地燃料

名称	ID
当地酒	本地_酒
LocalNet	LocalNet
Localiza	Localiza
Locatel	Locatel
Loch Fyne 餐厅	Loch_Fyne_Restaur
机车铁路午餐盒	机车_铁路_午餐_盒
Loding	Loding
Loehmann's	Loehmann's
洛伊斯酒店	Loews_Hotel
阁楼	阁楼
崇高的追求	Lofty_Pursuits
Logan's Road	Logan's Roadhouse
Logis	Logis
Logis de France	Logist_de_Fr
徽标	徽标
Logy	Logy_com
Loja AVIA	Loja_avia
Loja Gi	Loja_Gi
Loja das Sopa	Loja_das_sopas
Americanas	Lojas_Americans

名称	ID
洛哈斯大道	Lojas_Avenida
Lojas Becker	Lojas_Becker
洛哈斯·贝努伊特	Lojas_Benoit
Lojas Cem	Lojas_Cem
Lojas Centauro	Lojas_centauro
科伦坡洛哈斯	洛哈斯_科伦坡
Lojas MM	Lojas_mm
Lojas Pernambucanas	Lojas_Pernambucanas
Lojas Pompeia	Lojas_Pompeia
Lojas 卖家	Lojas_Seller
Lojas TaQi	Lojas_taqi
Lokalbrugsen	Lokalbrugsen
Lokman Hekim	Lokman_Hekim
Lokmangal 合作银行	Lokmangal_Co-operative_Bank
萝拉和丽莎	Lola_&_Liza
萝拉·波托纳	Lola_Botona
萝拉·罗卡	Lola_Roca
Lollicup	Lollicup
Lollicup 咖啡和茶	Lollicup_Coffee_&_Tea
Lollypop	Lollypop

名称	ID
Lomitos 2x1	lomitos_2x1
Lomo	Lomo
Londee	Londee
隆迪斯	隆迪斯
伦敦相机交易所	伦敦相机交易所
伦敦毒品	伦敦_毒品
伦敦炸鱼薯条	伦敦_Fish_&_Chips
孤星牛排馆	Lone_Star_Stakhouse
多头和福斯特房地产经纪人	Long_&_Foster_Realtors
多头和 McQuade	Long_&_ McQuade
长链	Long_Chain
朗·约翰·西尔弗的	Long_John_Silver
很久以前	很久以前
LongSian	LongSian
长链	长链
珑骧	珑骧
长角	长角
Longhorn 牛排馆	Longhorn_Steakhouse
隆基	隆基
龙门教育	龙门_教育

名称	ID
Longo's	Longo's
Longre	Longre
多头毒品	多头毒品
龙文教育	龙文_教育
洛尼亚	洛尼亚
朗斯代尔	朗斯代尔
Loogman	Loogman
观看	观看
Look & Liet	Look_&_Listen
LookSmart 改动	LookSmart_改动
Looks 沙龙	Looks_Salon
Loomis	Loomis
洛佩斯超级市场	Lopes_Supermercados
洛兰租车	Loran_Car _Rental
洛尔卡	洛尔卡
洛德和泰勒	Lord_&_Taylor
克里希纳勋爵银行	Lord_Krishna_Bank
洛伦佐	洛伦佐
Lorna Jane	Lorna_Jane
洛林	洛林

名称	ID
Los 3 Elefantes	Los_3_Elefantes
失去 4 个案例	los_4_ases
洛斯 Aguachiles	Los_Aguachiles
洛斯阿科斯	Los_Arcos
Los Bestias	Los_Bestias
Los Bisquets Bisquets Bis	Los_Bisquets_Bisquets_Obregon
洛斯将军	Los_Generales
Los Guerrilleros	Los_Guerrilleros
洛斯波塔莱斯	Los_Portales
Los Saldes	Los_Saldes
Los Telares	Los_Telares
Los Tocados de Crysanna	los_tocados_de_crySanna
Los cochinitos	los_cochinitos
斯洛文尼亚彩票	Loterija_Slovenije
Lotok	Lotok
很多披萨	Lots'_A_Pizza
Lotte	Lotte
乐天超市	Lotte_Super
Lotteria	Lotteria
Lottusse	Lottusse

名称	ID
莲花电子	Lotus_电子
莲花超市	莲花超市
莲花泰式餐厅	Lotus_Thai_Restai
Lotus's	Lotus's
Lotus 焕然一新	Lotus's_go_fresh
Lou & Grey	Lou_&_Grey
卢·马尔纳蒂的	Lou_Malnati's
路易斯	路易斯
路易斯·德尔海兹	Louis_Delhaize
路易斯·尼尔森	Louis_Nielsen
路易斯·菲利普	路易斯_菲利普
路易莎咖啡	路易丝咖啡
路易斯安那州足球俱乐部	Louisiana_FCU
路易斯安那州著名炸鸡	路易斯安那州_Famous_Fried_Chicken
美国路易斯安那州 FCU	路易斯安那_美国_FCU
休息室	休息室
Louviers FCU	Louviers_fcu
爱客连锁酒店	Love_Guest_连锁酒店
爱赖斯	Love_Ric
爱之乡商店	Love's Country_

名称	ID
Love's 旅行站	LOVE's_travel_st
可爱的卡片	Lovely_Card
可爱的指甲	Lovely_Nail
爱心小屋	Loving_Hut
Lovisa	Lovisa
Lowe's	Lowe's
下东区人民 FCU	Lower_East_side_peoples_fc
Lowes	Lowes
Lowes Foods	洛斯_食物
Lowy	Lowy
Loya	Loya
驼鹿小屋忠诚骑士团	Moose_Lodges 的忠诚勋章
润滑油服务中心	润滑油服务中心
Luby's	Luby's
Lucardi Juwelier	Lucardi_Juwelier
卢卡斯服务	卢卡斯服务
鹿城农村商业银行	鹿城_农村_商业银行
清晰诊断	清晰诊断
Lucid Motors	Lucid_Motors
运气	运气

名称	ID
Luck 7 Inn	Luck_7_Inn
幸运咖啡	幸运咖啡
幸运	幸运
幸运 7	Lucky_7
吉祥鸟健身俱乐部	Lucky_Bird_Fitness_Club
幸运品牌牛仔裤	幸运_品牌_牛仔裤
幸运电影院	幸运影院
幸运鱼餐厅	幸运鱼餐厅
幸运药房	幸运药房
艾伯森的 Lucky	Lucky_by_Albertsons
幸运的咖啡厅	幸运咖啡厅
卢登多村	卢登多村
卢多瓦银行	Ludova_Banka
Luella	Luella
Lufian	Lufian
德国汉莎航空公司	德国汉莎航空公司
Luhmann Tankstellen	Luhmann_tankstellen
Luhta 品牌专卖店	Luhta_Brand_Store
路易吉	路易吉
Luk Show	Luk_Show

名称	ID
Luke	Luke
露露酒店	Lulu_Hotel
露露大卖场	Lulu_Hypermarket
lululemon	lululemon
Lumbee 担保银行	Lumbee_Guaranty_B
木材城	Lumber_City
木材清算人	木材清算人
流明	流明
牙医 Lumino	Lumino_the_Dentist
露娜	露娜
Luna Grill	露娜烧烤
Luna Lunera	Luna_Lunera
Lune Bleu	Lune_Bleu
洛阳农村商业银行	洛阳_农村_商业银行
Lupa	Lupa
Lupe Tortilla	Lupe_Tortilla
Lupillos	Lupillos
Lupin	Lupin
Lush	Lush
Lush 手工化妆品	Lush_手工制作_化妆品

名称	ID
卢西塔纳	卢西塔纳
路西塔尼亚	路西塔尼亚
Lussile	Lussile
Lutz	Lutz
卢瓦克	卢瓦克
Lux Mini Spolem PSS	lux_mini_spolem_PSS
Lux Spolem PSS	lux_spolem_PSS
Luxemon 酒店	Luxemon_Hotel
纳智捷	纳智捷
芦溪牛肉火锅	Luxi_Beef_Hotpot
Luxihe Taosu	Luxihe_Taosu
蓝色地平线豪华酒店	豪华_蓝色地平线酒店
鹿源	鹿源
Luz da Lua	luz_da_lua
Luzerner Kantonalbank	Luzerner_KantonalBank
Luzete	Luzete
泸州市商业银行	泸州城市商业银行
泸州太台	泸州_台泰
吕菊	Lv_Ju
利沃夫	利沃夫

名称	ID
Lvmama	Lvmama
Lvx	Lvx
莱德尔英语	Lydell_Eglish
Lydia's Lechon	Lydia's_Lechon
Lyf & heilsa	lyf_&_Heilsa
Lyf og Heilsa	lyf_og_Heilsa
Lyfja	Lyfja
林恩·阿德勒	Lynn_Adler
Lynx	Lynx
山猫光学	Lynx Optique
Lyogkiy Shag	Lyogkiy_Shag
里昂汽车公园	Lyon_Parc_Auto
歌词	歌词
lyubimaya apteka	lyubimaya_apteka
Lyubimchik	Lyubimchik
Lyubimiy	Lyubimiy
Lyuksor	Lyuksor
Lansforsakringar	Lansforsakringar
Lan & Spar Bank	Lán_&_Spar_Bank
Lebeny-Kunsight Takarek	Lebény-Kunsziget_Takarék

名称	ID
Lekarna ipc	lekárna_ipc
Leon de Bruxelles	Leon_de_Bruxelles
Líra	Líra
Löplabbet	Löplabbet
Lövbjerg Supermarket	Lövbjerg_Supermarked
M & Co	M_&_Co
M 美容	M_Beauty
M Corner	M_Corner
M Local	M_Local
M P Rajya Sahakari Bank Maryadit	M_P_Rajya_Sahakari_Bank_Maryadit
M 陈列室	M_Showroom
M 支架	M_Stand
M&M 食品市场	M&M_Food_Market
M&P	M&P
M&S Simply Food	M&S_Simply_Food
M&T 银行	M&T_Bank
M+M tabak	m+m_tabak
M市场	M市场
M-Milk	M-Milk
M. Ercilla	M._Ercilla

名称	ID
M. 警官	M._官员
M. 石油公司	M._Petrol
m.tsubomi	m.tsubomi
m. 视频	m. 视频
M1	M1
M1 酒店	M1_Hotel
M1 加油站	M1_Petrol_Station
M2 鞋子	M2_Shoes
M24	M24
M7 Takarék	M7_Takarék
MA TSU Mi	ma_tsu_mi
MAACO	MAACO
MAC	MAC
MAC FCU	MAC_FCU
美凯龙	美凯龙
MACANNA	MACANNA
澳门斗罗	MACAO_DOULAO
麦克唐纳	麦克唐纳
MACFit	MACFit
可可夫人	MADAME_COCO

名称	ID
MADECENTRO	MADECENTRO
马德罗	马德罗
疯人院	疯人院
MAES	MAES
MAGBIT	MAGBIT
MAGNUM 4D	MAGNUM_4D
MAGRABI 光学	MAGRABI_OPTICAL
MAGY	MAGY
MAGiC 触摸	MAGiC_TOUCH
麦胜里餐厅	MAI_SENG_LI_RESTAURANT
自制模式	MAISON_MODE
大型电影院	MAJOR_CINEPLEX
MAKALU	MAKALU
永远弥补	永远化妆
化妆品店	彩妆店
马基斯广场	MAKIS_PLACE
MAKS	MAKS
MAKUUNI	MAKUUNI
MALAK AL TAWOUK	MALAK_AL_TAWOUK
MALIKS	MALIKS

名称	ID
MAM 有限公司	mam_Limite
妈妈庇护所	妈妈庇护所
MAMAWAI	MAMAWAI
猛犸象	猛犸象
男人	男人
MAN 巴士	Man_bus
法力	法力
真锅市	真锅市
MANASEER	MANASEER
芒果	芒果
MANO	MANO
制造业	制造业
MAPCO	MAPCO
MAPFRE	MAPFRE
MARASIL	MARASIL
马拉松运动	Marathon_Sport
马可波罗酒店	MARCO_POLO_HOTELS
玛丽娜·里纳尔迪	MARINA_RINALDI
玛丽蒂姆	玛丽蒂姆
市场进入	MARKET_IN

名称	ID
JASONS 的《市场广场》	Market_place_by_Jasons
MARKOR 家具	MARKOR_FURNIS
马尔科娃	马尔科娃
MARLO'S	MARLO'S
MARMALATO	MARMALATO
MARSANPIEL	MARSANPIEL
貂皮	貂皮
MARUGAME SEIMEN	MARUGAME_SEIMEN
丸井（冈山县鸟取县中国区）	MARUI_ (冈山_鸟取_中国_地区)
玛鲁蒂铃木	玛鲁蒂_铃木
MARY COHR	MARY_COHR
MAS	MAS
马萨坎巴东	MASAKAN_PADANG
玛莎拉蒂	玛莎拉蒂
MASKU	MASKU
MASLIFE	MASLIFE
MASRAF AL RAYAN	MASRAF_AL_RAYAN
质量	质量
马斯蒂娜	马斯蒂娜
MATCH	MATCH

名称	ID
MATKAHUOLTO	MATKAHUOLTO
MATRITSA	MATRITSA
矩阵	矩阵
矩阵仓库计算机	MATRIX_WAREHOUSE_PC
MATT 光学	matt_Optik
MATTINO-OBUV'	MATTINO-OBUV'
床垫谷仓	MATTRESS_BARN
MATY	MATY
MAUBOUSSIN	MAUBOUSSIN
MAX	MAX
马克斯·佩里	MAX_PERRY
MAX SPORT	MAX_SPORT
最大 T 个解决方案	MAX_T_解决方案
Max&Co。	Max&Co。
MAXBET	MAXBET
MAXCOOP	MAXCOOP
MAXEN	MAXEN
最大直径	最大直径
MAXI'S	MAXI'S
MAXIMA BAZE XXX	MAXIMA_BAZE_XXX

名称	ID
MAXIMA XX	MAXIMA_XX
MAXIMA XXX	MAXIMA_XXX
MAXUS	MAXUS
MAXXIS	MAXXIS
西安五一酒店	MAY_FIRST_HOTEL_西安
迈巴赫	迈巴赫
MAYORISTA 10	MAYORISTA_10
MAZP	MAZP
MB	MB
MB 财经	mb_财经
MB 多银行	MB_MULTIBANCO
mb-Rent	mb-Rent
MBCA	MBCA
MBG	MBG
MBO 电影院	MBO_电影院
MBT	MBT
MBT 石油公司	mbt_Petrolem
MC 体育	MC_SPORTS
MCB	MCB
MCD	MCD

名称	ID
MCI 电信公司	MCI_Telecommunications_
MCK	MCK
MCL	MCL
MCL 自助餐厅	Mcl_Cafeter
MCM	MCM
MDA Electroménager	mda_electroMénager
MDB	MDB
MDG 超市	mdg_superm
MDM 银行	mdm_bank
ME BY MELIA	ME_BY_MELIA
MEAB 银行	meab_bank
MEC	MEC
MEDCO	MEDCO
媒体风暴	MEDIA_STROM
MEDIAPRESS	MEDIAPRESS
医疗保险	医疗保险
MEDJIMURSKA BANKA	MEDJIMURSKA_BANKA
超级自行车	巨型自行车
MEGA 唐吉诃德	Mega_don_Quijote
MEGABOX	MEGABOX

名称	ID
MEGAPUNKT	MEGAPUNKT
MEGATOP	MEGATOP
MEHOOD	MEHOOD
梅州东坡餐厅	梅州东波餐厅
MEIDI-YA	MEIDI-YA
MEISHIMEIKE	MEISHIMEIKE
美怡佳酒店	美宜家_酒店
美利亚酒店及度假村	美利亚酒店及度假村
融化披萨	融化披萨
MELTZ	MELTZ
MENORKINAS	MENORKINAS
菜单	菜单
MEO	MEO
墨菲斯托	墨菲斯托
欧洲议会议员 ATM	MEPS_ATM
北海	MER_DU_NORD
MERCADONA	MERCADONA
MERCADOS ZAPATOCA	MERCADOS_ZAPATOCA
梅里达	梅里达
MERKUR	MERKUR

名称	ID
里面 MERKUR	merkur_inside
MERQUEFACIL	MERQUEFACIL
MERRELL	MERRELL
MESSAGE	MESSAGE
METRO 现金随身携带	METRO_CASH_&CARRY
麦德龙百货	Metro_百货_商店
METRO 儿童公司	METRO_KIDS_COMP
地铁货运	Metro_Kargo
METSS	METSS
MEXX	MEXX
MEZZO	MEZZO
MF banka	mf_banka
MFA	MFA
MFC	MFC
MFO Matratzen	mfo_matratzen
MG	MG
MG 社区信用合作社	mg_Community_Credit_合作社
米高梅大酒店	mgm_Grand
MGallery	MGallery
MHB	MHB

名称	ID
我的银行	MI_BANCO
我的房子 cosas de Casa	mi_casa_cosas_de_Casa
MIC.RO	MIC.RO
迈克尔·科尔斯	MICHAEL_KORS
佛罗里达州中部	MIDFLORIDA_CU
MIKASA MOOR	MIKASA_MOOR
迈克的厨房	MIKE'S KITCHEN
MIKIMOTO	MIKIMOTO
MILADYS	MILADYS
米兰咖啡	米兰咖啡
MILENIO Cosas de Casa	milenio_cosas_de_Casa
千禧酒店和度假村	千禧酒店及度假村
MILLIE'S	MILLIE'S
MILTI	MILTI
MIMMOS	MIMMOS
MINGR	MINGR
微型	微型
五矿证券	五矿_证券
民生证券	民生证券
明尤恩中心酒店	MINYOUN_CENTRAL_酒店

名称	ID
MINYOUN 皇家酒店	MINYOUN_ROYAL_酒店
MINYOUN SUNIYA 酒店	MINYOUN_SUNIYA_HOTELS
MIRABAUD	MIRABAUD
MISAKO	MISAKO
MISHKA	MISHKA
MISR	MISR
MISR 药房	misr_Pharmac
错过 21	MISS_21
甜甜圈先生	MISTER_DONUT
米西斯酒店	MITSIS_HOTELS
三菱	三菱
MIX Markt	mix_Markt
MK 餐厅	MK_餐厅
MKB 银行	MKB_Bank
MKB 联合银行	MKB_联合银行
MLDC	MLDC
MLL 酒店	mll_Hoteles
MM 果汁	mm_juice
MM 超级市场	mm_mega_Mark
嗯	嗯

名称	ID
MMartan	MMartan
MNG Kargo	mng_kargo
mo&Co。	mo&Co。
MO-BO	MO-BO
MOBALPA	MOBALPA
手机。	手机。
MOD 披萨	mod_Pizz
MODA DI CENTONE	MODA_DI_CENTONE
MODEINN	MODEINN
MODELORAMA	MODELORAMA
现代古典	现代_古典
MODIANA	MODIANA
MOF	MOF
MOGGY'S	MOGGY'S
MOHITO	MOHITO
MOI	MOI
MOIRA	MOIRA
MOJO S.PHINE	MOJO_S.PHINE
MOL	MOL
MOLTON	MOLTON

名称	ID
MOMA	MOMA
君主	君主
蒙达多里	蒙达多里
MONETA 货币银行	Moneta_money_Bank
货币兑换	货币兑换
MONKI	MONKI
MONOPETO	MONOPETO
蒙塔涅	蒙塔涅
MONTAGUT	MONTAGUT
万宝龙	万宝龙
锡耶纳的复活节山	MONTE_DEI_PASCHI_DI_SIENA
MOODO	MOODO
MOPARK	MOPARK
莫拉纳	莫拉纳
MORAVEL papirnictvi	moravel_papirnictVI
摩根	摩根
莫里森	莫里森
摩斯汉堡	MOS_BURGER
苔藓俱乐部	MOSS_CLUB
MOTEL268	MOTEL268

名称	ID
动机	动机
MOTOLEK	MOTOLEK
发动机	发动机
MOTOVAC	MOTOVAC
摩凡陀	摩凡陀
继续前进	MOVE_ON
MOil	MOil
MOil 市场	MOil_市场
MP 汽油	MP_PETROL
MPB	MPB
MPH 书店	mph_bookstore
MQ	MQ
先生自己动手做	MR_DIY
希腊先生	MR_GREEK
铁板烧先生	Mr_Teppanyaki
比尔先生	MR._BEER
BROWN 咖啡先生	布朗咖啡先生
MRF 条款和条件	MRF_T&S
MRINetwork	MRINetwork
太太	太太

名称	ID
MRW	MRW
MS More	ms_more
山地车坦克比赛	山地车坦克比赛
MTBank	MTBank
MTC 商店	MTC_shop
MTK	MTK
MTN	MTN
MTOclean	MTOclean
MTS	MTS
MTS 银行	MTS_Bank
MU-MU	MU-MU
孟泰保险	MUANG_THAI_INSURANC
曼泰人寿保险	MUANG_THAI_LIFE_ASSURAN
孟泰租赁	MUANGTHAI_LEASING
三菱日联银行	MUFG_Bank
MUGG & BEAN	MUGG_&_BEAN
无印良品	无印良品
多服务器	多服务器
多索姆	多索姆
音乐	音乐

名称	ID
MUSIMUNDO	MUSIMUNDO
MUSSI	MUSSI
野马	野马
突变体快车	突变体快车
MUYA	MUYA
MX	MX
MX-SPORT	MX-SPORT
我的地方	MY_PLACE
我的尺码	我的尺码
MYBOX	MYBOX
MYDIN	MYDIN
迈尔	迈尔
MYG 数字中心	myg_digital_Hub
MYGYM	MYGYM
MYKAL	MYKAL
MYMAY	MYMAY
MYWAY	MYWAY
MYaSNOV'	MYaSNOV'
Ma'afe Ne'eman	ma'afe_ne'eman
马安咖啡	Maan_Coffee

名称	ID
马鞍山农村商业银行	马鞍山_农村_商业银行
Mabi	Mabi
马布鲁克	马布鲁克
Mac 中心	Mac_Center
Mac 酒店	Mac_Hotels
Mac 的	Mac 的
MacEwen	MacEwen
MacStore	MacStore
Macado's	Macado's
澳门帝国茶	澳门_帝国_茶
通心粉烧烤	Macaroni_Grill
澳门街餐厅	Macau_Street _餐厅
澳门味道	Macau_Taste
Maccabai Sherutei Briut	Maccabai_sherutei_Briut
Maccabi	Maccabi
Maccabi Pharm	Maccabi_Pharm
梅斯	梅斯
Machikadoya	Machikadoya
Mack	Mack
Mackays 商店	Mackays_Stores

名称	ID
Macowens	Macowens
Macpac	Macpac
宏	宏
宏中心	宏中心
宏秒	宏秒
Macson	Macson
Macu	Macu
梅西百货	梅西百货
梅西百货家具画廊	梅西百货_家具_画廊
疯狂的绿党	Mad_Greens
疯狂的墨西哥人	Mad_Mex
疯狂的科学	疯狂科学
为大蒜而疯狂	Mad_for_Garlic
罗斯夫人	Madam_Rose
王女士咖啡厅	Madam_Wong_Cafe
夫人	夫人
Made In	Made_In
Madhi ni Khamani	madhi_ni_khamani
MadhuLoka 酒类精品店	MadhuLoka_烈酒_精品店
Madhur 快递服务	Madhur_Courier_Service

名称	ID
中央邦格拉明银行	Madhya_Pradesh_Gramin_Bank
Madhyanchal Gramin Bank	Madhyanchal_Gramin_Bank
麦迪逊县 FCU	Madison_county_FCU
Madlen	Madlen
Mado	Mado
马德拉斯咖啡屋	马德拉斯咖啡屋
前田	前田
马士基	马士基
大师披萨	Maestro_Pizza
MagNet 银行	MagNet_银行
Magarpa	Magarpa
一般杂志	Magasin_General
Magazan	Magazan
Magazin Goryashchikh Putyovok	Magazin_Goryashchikh_Putyovok
路易莎杂志	杂志_Luiza
Maggiano's	Maggiano's
Maggie T	Maggie_T
MaggieMoo的	MaggieMoo的
马焦雷	马焦雷
魔法玉米	Magic_Corn

名称	ID
魔法科斯塔布兰卡	Magic_Costa_Blanca
神奇点心	Magic_Dim_Sum
魔法英语	Magic_英语
魔法生活	Magic_Life
魔法星球	Magic_Planet
Magistr	Magistr
Magistra Lekárny	Magistra_Lekárny
Magiya Zolota	Magiya_Zolota
Magma Fincorp	Magma_Fincorp
Magnat-rd	Magnat-rd
Magnet	Magnet
磁性标志	磁性标志
Magnit	Magnit
Magnit 化妆品	马格尼特_化妆品
玉兰	玉兰
Magnolia FCU	Magnolia_fcu
Magnoliya	Magnoliya
Magnotel	Magnotel
马格南	马格南
玛格利塔披萨	玛格利塔_披萨

名称	ID
Magruder's	Magruder's
Mahalaxmi 银行	Mahalaxmi_Bank
Mahanagar Gas	Mahanagar_Gas
马哈拉施特拉邦格拉明银行	Maharashtra_Gramin_Bank
马哈拉施特拉邦天然气	马哈拉施特拉邦_天然气
马哈拉施特拉邦合作银行	Maharashtra_State_Co-op_Bank
Maharishi Vidya Mandir 学校	Maharishi_Vidya_Mandir_School
Mahasagar 游记	Mahasagar_Travels
马辛德拉	马辛德拉
Maicao	Maicao
女佣大队	Maid_Brigade
完美女仆	Maid_to_Perfection
Maid-Rite	Maid-Rite
MaidPro	MaidPro
Maidenform	Maidenform
邮箱等	邮箱等。
主赛事娱乐	主赛事_娱乐
缅因州家庭 FCU	Maine_family_FCU
缅因高地 FCU	Maine_Highlands_FCU
缅因州储蓄 FCU	Maine_savings_FCU

名称	ID
缅因州立大学	缅因州_state_cu
中国大陆	中国大陆_中国
主资源银行	主资源_银行
主打套房	Mainstay_Suites
主流精品店	主流_精品店
Mainstreet 银行	Mainstreet_Bank
主街 CU	Mainstreet_cu
MainturFurniture	MainturFurniture
Mainzer VB	Mainzer_VB
MairiedeParis	MairiedeParis
Mais Econômica	Mais_Econômica
Mais Optica	Mais_Optica
Mais1Cafe	Mais1Cafe
MaisBrasil	MaisBrasil
Maisadou Tea & Cafe	Maisadou_Tea_&_Cafe
Maisen	Maisen
Maison De L'Hu	Maison_De_L'Hui
家居装饰	Maison_Decor
Literie 之家	Maison_de_la_literie
世界之家	Maison_du_Monde

名称	ID
麦新阁精品酒店	麦兴_精品酒店
Maiya 牙科	Maiya_Dental
雄伟度假村	Majestic_Resorts
雄伟的葡萄酒仓库	雄伟的葡萄酒仓库
Majik Market	Majik_Market
永远弥补	永恒彩妆
化妆牙科	彩妆_牙科
Makeymytrip	Makeymytrip
Makotoya	Makotoya
Makovec maso-uzeniny	makovec_maso-uzeniny
马克石油公司	马克石油公司
万客隆餐饮服务	万宏_餐饮服务
宏观市场	宏观市场
马克萨维特	马克萨维特
Maksimus	Maksimus
Maky	Maky
Malabar 金和钻石	Malabar_Gold_&_Diamond
马兰面条	Malan_Noodle
Malatya Pazari	Malatya_Pazari
Malayala Manorama	Malayala_Manorama

名称	ID
Malco	Malco
Maldita	Maldita
Maldron	Maldron
Malheur FCU	malheur_fcu
马里马里	Mali_Mali
阿联酋购物中心有限责任公司	mall_of_the_emirates_LLC
Malmaison酒店	Malmaison酒店
马洛诊所	Malo_Clinic
马尔普卡快车	Malpka Express
Malubianbian	Malubianbian
Malzers Backstube	Malzers_backstube
傅妈妈的亚洲之家	mama_fu's Asian_House
后藤妈妈	Mama_Goto
Mama Roma	Mama_Roma
妈妈的鸡	mamma's_Chicken
猛犸象	猛犸象
Mammy's	Mammy's
Mampabaño	Mampabaño
活着的男人	Man_Alive
Man Divan	Man_Divan

名称	ID
人对人	Man_To_Man
Man&Manetti	Man&Manetti
马纳普拉姆金融	Manappuram_Finance
曼彻斯特及更多	曼彻斯特_&_更多
满族炒锅	Manchu_Wok
Mancino 的 Pizza & Grinders	Mancino's_pizza_&_Grinders
文华东方酒店	文华东方酒店
柑橘鸭	Mandarina_Du
曼达丽娜花园	文华花园
曼菲尔德	曼菲尔德
Mang Inasal	Mang_Inasal
漫画仓库	Manga_souko
Mangirl 公寓	Mangirl_aportent
芒果家居专卖店	Mango_home_Specialty_store
曼哈顿百吉饼	Manhattan_Bagel
玛尼	玛尼
狂躁	狂躁
烧烤狂欢	Mania_de_Churrasco
修指甲快车	美甲_Express
马尼拉格雷	Manila_Grace

名称	ID
马尼帕尔医院	马尼帕尔医院
Manish Packers 和 Movers	Manish_Packers_and_movers
马尼托	马尼托
Manjaku	Manjaku
Manlec	Manlec
Mann Hann	Mann_Hann
Manneken	Manneken
礼仪咖啡	礼貌_咖啡
万宁	万宁
Manolo Bakes	Manolo_Bakes
庄园	庄园
庄园药房	庄园_药房
马诺拉	马诺拉
Manotel	Manotel
人力	人力
曼辛格集团	Mansingh_Group
Manten Sakaba	Manten_Sakaba
口头禅	口头禅
宏利人寿-中化集团	宏利人寿-中化集团
马恩岛石油公司	Manx_Petrolem

名称	ID
满心	满心
Manyavar	Manyavar
毛泽东湾	Mao's_bay
Mapco 快车	Mapco_Expres
枫熊学校	Maple_Bear_School
枫叶酒店	Maple_Leaf_Hotel
Maplin	Maplin
三月酒店	Mar_Hotels
Marabraz	Marabraz
马拉松	马拉松
Marble Slab Creamery	Marble_Slab_Creamery
Marbodal	Marbodal
马克·凯恩	Marc_Cain
马克·奥波罗	Marc_O'Polo
马克·奥里安	Marc_Orian
Marc's	Marc's
马克药房	Marc's_Pharmac
MarcPol	MarcPol
March of Dimes	March_of_dimes
马尔凯	马尔凯

名称	ID
Marché Plus	Marche_Plus
Marché U	Marche_U
马可·阿尔达尼	Marco_Aldany
Marco 的披萨	Marco's_Pizza
Marcone	Marcone
Marconfort	Marconfort
马科斯	马科斯
Marcs	Marcs
马库斯	马库斯
马库斯剧院	Marcus_Theatres
马雷多	马雷多
Marees	Marees
马雷卡酒店连锁	Mareka_Hotel_catenation
Marella	Marella
Mareshki	Mareshki
马芬银行	Marfin_Bank
Marfin Egnatia Bank	Marfin_Egnatia_Bank
玛格丽塔	玛格丽塔
免保证金超级市场	保证金_免费_超级市场
Marhino	Marhino

名称	ID
玛丽亚·菲洛	Maria_Filo
玛丽亚的	玛丽亚的
马里亚诺的新鲜市场	马里亚诺的新鲜市场
玛丽·布拉切尔	Marie_Blachère
玛丽·卡伦德的	Marie_Callender'
玛丽·克莱尔	玛丽·克莱尔
Marie 法国车身系列	Marie_France_Bodylin
玛丽·斯托普斯诊所	Marie_Stopes 诊所
玛丽埃塔·马丁聚合	Marieta_Martin_Agregates
玛丽莲	玛丽莲
码头酒店	Marina_酒店
海军陆战队招募	海军陆战队_招募
船用 FCU	marine_FCU
水手财经	海事_金融
马里奥·米克	Mario_Mikke
Marionnaud Parfumeries	Marionnaud_Parfumeries
玛丽莎	玛丽莎
Marisfrolg	Marisfrolg
Marisol	Marisol
Marithé+François Girbaud	Marithé+Francois_Girbaud

名称	ID
玛丽蒂姆酒店	玛丽蒂姆酒店
海事银行	海事银行
Mariya Mebel'naya Fabrika	Mariya_mebel'naya_Fabrika
玛丽亚-拉	玛丽亚-拉
Marjane	Marjane
标记	标记
马克·阿伦	Mark_Aalen
马克·福尔梅勒	Mark_Formelle
Mark Pi 的	Mark_Pi's
Mark's	Mark's
马克的披萨店	Mark's_Pizzeria
Marka	Marka
Markant	Markant
Markantalo	Markantalo
市场篮子	市场篮子
市场街	市场街
美国市场 FCU	market_usa_fcu
Marketlace	集市_咖啡厅
马克格拉芬	马克格拉芬
马卡姆	马卡姆

名称	ID
Marko	Marko
Markobar	Markobar
Marks & Spencer	Marks_&_Spencer
Marktkauf	Marktkauf
马林旅行	Marlin_Travel
Marmaris Büfe	Marmaris_Büfe
旱獭	旱獭
Marmotte	Marmotte
摩洛哥电信	摩洛哥_Telecom
Maronno Rosso	Maronno_Rosso
Marqt	Marqt
苏亚雷斯侯爵	Marques_Soares
Marrachino	Marrachino
万豪酒店	万豪酒店
万豪行政公寓	万豪行政公寓
万豪度假俱乐部	万豪度假俱乐部
玛丽布朗	玛丽布朗
马斯登建筑协会	马斯登建筑协会
Marsel	Marsel
马什超市	Marsh_超市

名称	ID
元帅	元帅
马歇尔	马歇尔
马歇尔和伊尔斯利公司	Marshall_&_ilsley_Corp
马歇尔	马歇尔
Marshalls 印度	Marshalls_印度
马斯顿	马斯顿
Marstons Plc	Marstons_Plc
Mart Minas	Mart_Minas
Martabak Alim	Martabak_Alim
玛塔巴克披萨 Orins	Martabak_Pizza_Orins
Martes Sport	Martes_Sport
Marthe	Marthe
马丁石油	马丁油
马丁的	马丁的
马丁药店	Martin's Drugstore
马丁的食品市场	马丁食品市场
马丁的汽油	Martin's 汽油
马丁的超级市场	Martin's 超级市场
马丁超市	马丁超市
马丁化干洗	马丁化_干洗

名称	ID
Marty's Shoes	Marty's 鞋子
马蒂	马蒂
Maruai	Maruai
丸江	丸江
丸福咖啡店	丸福咖啡店
Marugo	Marugo
丸八	丸八
Marukame	Marukame
Maruki	Maruki
丸宫店	丸宫店
丸三证券	丸三证券
Marwa	Marwa
Marwin	Marwin
玛丽·布朗的著名鸡肉	Mary_Brown's Famous_Chicke
玫琳凯化妆品	Mary_Kay_Cosmet
玛丽帕兹	玛丽帕兹
马尔扎诺	马尔扎诺
Mas X Menos	Mas_X_Menos
Mas y Mas	mas_y_mas
MasKota	MasKota

名称	ID
Mas QMenos	Mas QMenos
马赛服装公司	马赛_服装公司
马萨拉厨房	马萨拉厨房
Masaya	Masaya
Maskom	Maskom
Masoutis	Masoutis
加州大学马萨斯湾	mass_bay_cu
马萨诸塞将军布里格姆	马萨诸塞州将军_布里格姆
地下交通铁路	公共交通铁路
MassMutual	MassMutual
MassMutual 水星人生	MassMutual_水星_生命
按摩羡慕	Massage_Envy
Massage Envy	Massage_Envy_Spa
按摩高度	按摩高度
Massimo Dutti	Massimo_Dutti
Mast Kalandar	Mast_Kalandar
主实例	主实例
AB 大师	Master_ab
Master Care 设施服务	Master_Care_设施_服务
Costura 大师	Master_Costura

名称	ID
Master 经济旅馆	Master_Econom_Inn
大师泡沫	Master_Foom
主气体	Master_Gas
主主机	主机
Master Hotéis	Master_Hotéis
康师傅厨师桌	Master_Kong_Chef's Table
MasterCuts	MasterCuts
MasterFarma	MasterFarma
MasterLink 证券	MasterLink_证券
大师级工艺	大师级工艺
大师	大师
MasterFIT	MasterFIT
大师赛	大师赛
马斯特斯旅馆	Masters_Inn
Mat & May	Mat_&_May
Mat Viet	Mat_Viet
斗牛士	斗牛士
Matahari	Matahari
马塔兰	马塔兰
马塔努斯卡谷 FCU	Matanuska_valley_FCU

名称	ID
匹配服装店	Match_clothing_stor
火柴盒	火柴盒
Matco 工具	Matco_Tools
Mateus	Mateus
Mathers	Mathers
Mathnasium	Mathnasium
玛蒂尔达	玛蒂尔达
Matim Li	Matim_Li
Matkroken	Matkroken
Matmazel	Matmazel
法国马图	Matou_France
床垫	床垫
Martzen CONCORD	Matratzen_Concord
Matrebellerna	Matrebellerna
马祖连锁站	松链站
松乐	松乐
松根	松根
松子日本料理	松子_日本料理
松本信用金库	松本信用金库
松之屋	松之屋

名称	ID
Matsuri	Matsuri
松屋	松屋
马塔纳	马塔纳
马修的选择	马修的选择
床垫折扣店	床垫_折扣店
床垫公司	床垫_Firm
床垫巨头	床垫_Giant
床垫仓库	床垫_仓库
Matöppet	Matöppet
Maui Wowi	Maui_Wowi
莫里斯·拉克鲁瓦	Maurice_Lacroix
Maurice's BBQ	Maurice's BBQ
毛里求斯电信	毛里求斯_电信
Mauro Giuli	Mauro_Giuli
毛罗·塞尔吉奥	Mauro_Sergio
毛里利奥·德·阿尔梅达	Maurílio_de_almeida
Maveks	Maveks
Maveli 专卖店	Maveli_Stores
Maverick 金融	Maverick_Finance
Maverik	Maverik

名称	ID
Maverik 市场	Maverik_Market
Mavi 牛仔裤	Mavi_Jeans
Mavis 折扣轮胎	Mavis_Discount_Tire
Mawadis	Mawadis
Max & Erma's	Max_&_Erma's
Max Bahr	Max_Bahr
马克斯·布伦纳	Max_Brenner
Max Broock 房地产经纪人	Max_Broock_Realtors
Max 服装	最大服装数量
最大能量	最大能量
马克斯时装	Max_Fashions
Max 人寿保险	最大人寿保险
Max Mart	Max_Mart
最大肌肉营养	最大肌肉_营养
马克斯纽约人寿保险	Max_Newyork_Life_Ins
Max 零售店	Max_retail_Store
Max's 餐厅	Max's 餐厅
MaxMara	MaxMara
MaxValu	MaxValu
MaxValu Tanjai	MaxValu_Tanjai

名称	ID
Maxbo	Maxbo
Maxen 教育	Maxen 教育版
马克西	马克西
Maxi & Cie	Maxi_&_Cie
Maxi Bebé	Maxi_Bebé
Maxi Bodega	Maxi_Bodega
Maxi Diez	Maxi_Diez
Maxi Farma	Maxi_Farma
Maxi Marché	Maxi_Marché
Maxi Zoo	Maxi_Zoo
Maxi-Livres	Maxi-Livres
Maxi-Makasiini	Maxi-Makasiini
Maxiahorro	Maxiahorro
Maxigrula	Maxigrula
Maxim's	Maxim's
Maxim 的香港日	Maxim's_Hong_Day
Maxim's MX	maxim's_mx
马克西姆宫	Maxim's Palace
最大限度	最大限度
Maxipark	Maxipark

名称	ID
Maxiprest 轮胎	Maxiprest_Tyres
Maxis	Maxis
Maxmat	Maxmat
Maxol	Maxol
Maxrave	Maxrave
Maxway	Maxway
麦克斯韦	麦克斯韦
麦克斯韦搬迁	Maxwell_搬迁
Maxxi Atacado	Maxxi_Atacado
Maxxi Econômica	Maxxi_Econômica
5 月 10 日	5 月 10 日
五月精品店	五月_精品店
马来亚银行	马来亚银行
Mayer's Markenschuhe	Mayer's_markenschuhe
Mayerline	Mayerline
五月花公交	五月花_Transit
Mayo Chix	Mayo_Chix
梅奥诊所	Mayo_Clinic
Mayorsa	Mayorsa
马萨特兰墨西哥餐厅	马萨特兰墨西哥餐厅

名称	ID
马自达	马自达
马佐维耶茨基地区银行	Mazowiecki_Bank_Regionalny
妈祖村	Mazu_Village
Mazzio's	Mazzio's
Mbna Corp	Mbna_Corp
Mc GREGOR	mc_Gregor
Mc 牛仔裤	Mc_Jeans
McAlister's Deli	McAlister's_deli
McArthur	McArthur
McArthurGlen 名牌直销店	McArthurGlen_设计师_奥特莱斯
McCabes 药房	McCabes_药房
McCafé	McCafé
McCarthy的	McCarthy的
McClure	McClure
McColl的	McColl的
McCormick 和 Schmick's	McCormick_&_Schmick's
McCoy的	McCoy的
McDonald的	McDonald的
McFit	McFit
McGrath	McGrath

名称	ID
McGuireWoods	McGuireWoods
McIntosh 旅馆	McIntosh_客栈
McKeown	McKeown
McKinsey & 公司	McKinsey_&_公司
McLaren	McLaren
McMenamins	McMenamins
McPaper	McPaper
McPen papirnictvi	McPen_papirnictvi
McRae的百货商店	McRae's_百货商店
McRennett	McRennett
我来药店	mecome_Drugstore
me-n-ed's Pizzeria	me-n-ed's_Pizzeria
轮上吃饭	Meals_On_Wheels
吃肉和吃	Meat_&_Eat
肉与健康	Meat_&_Fit
Mebel'Grad	Mebel'__Grad
Mebel'chernozem'ya	mebel'_chernozem'ya
Mebel'Tut Deshevle	Mebel'_Tut_Deshevle
mebel'grad	mebel'grad
麦加	麦加

名称	ID
麦加化妆品	Mecca_Cosmetica
Meccano	Meccano
机械库	机械师_银行
Mechta	Mechta
地中海海滩	Med_Playa
MedExpress 紧急护理	MedExpress_紧急护理
MedPlus	MedPlus
Meda Kuchen	Meda_Küchen
梅达斯	梅达斯
Medeia	Medeia
医疗第一	医疗第一
MediRite 药房	MediRite_药房
媒体咖啡厅 Popeye	MediaCafe_Popeye
媒体市场	Media_Markt
媒体集市	Media_Mar
媒体世界	媒体世界
medialunas del abuelo	mediaLunas_del_abuelo
中位数酒店	Median_Hotels
医疗银行	医疗银行
医疗城医疗保健	医疗_城市_医疗保健

名称	ID
医用头发修复	医疗毛发修复
医疗药房	Medicap_药房
医疗保险	医疗保险
药品专柜	Medicine_Shoppe
Medicinos Bank	Medicinos_Bankas
Mediclinic	Mediclinic
Medicross 医疗中心	Medicross_Medical_Centre
中世纪时代	中世纪时代
Medimax	Medimax
麦地那县 FCU	Medina_county_FCU
Mediq	Mediq
Mediq Apotheek	Mediq_Apotheek
Meditsina dlya Vas	Meditsina_dlya_vas
中型酒店	中型酒店
Medoil	Medoil
Medsi	Medsi
Medunitsa	Medunitsa
美杜莎电影院	Medusa_Cinema
Meed	Meed
认识新鲜	遇见新鲜

名称	ID
与所有人见面	Meet_all
米赞银行	Meezan_Bank
超级的	超级的
Mega Ba'ir	Mega_Ba'ir
Mega Cgr	Mega_Cgr
墨西哥超级商业	Mega_Comercial_Mexicana
超级影像	超级图片
兆丰国际商业银行	Mega_国际_商业_银行
巨型码头	超级玛丽娜
超级超市	超级超市
Mega Saver	Mega_Saver
兆丰证券	巨型证券
超级体育	超级体育
超级包裹	Mega_Wraps
Mega-Oyl	Mega-Oyl
MegaFon	MegaFon
Megabank	Megabank
Megaflis	Megaflis
Megalos	Megalos
Megamart	Megamart

名称	ID
Megamatte	Megamatte
Megane 1Ban	Megane_1Ban
梅甘娜药物	Megane Drug
梅甘娜之花	Megan_Flower
梅甘娜故居	Megan_House
梅甘娜小屋	Megane_Hut
Megane-no-Aizawa	Megane-no-Aizawa
Megane-no-Harada	Megane-no-Harada
Megane-no-Nagata	Megane-no-Nagata
Megapetro	Megapetro
Megasport	Megasport
Megastroy	Megastroy
Mego	Mego
惠之里	惠之里
Mehamaks	Mehamaks
Mehilainen	Mehilainen
穆罕默德·塔特利	Mehmet_tatli
美豪典雅酒店	Mehood_Elegant_Hotel
梅惠权	梅惠全
Mei&Mei	Mei&Mei

名称	ID
Meijer	Meijer
Meijer 加油站	Meijer_Gas Station
梅杰药房	Meijer _药房
Meijiale	Meijiale
Meikles	Meikles
美联中和兽医医院	美联众和_兽医_医院
Meilinge	Meilinge
Meineke 汽车保养中心	Meineke_Car_Care_Care_C
Meininger	Meininger
Meio Tom	Meio_Tom
美士林	美士林
Meister BäR 酒店	Meister_bar_Hotels
名铁协章	Meitetsu_Kyosho
Meiyiduo	Meiyiduo
魅族	魅族
Meka	Meka
Mekk	Mekk
Mekka	Mekka
Mekonomen	Mekonomen
Mekopart	Mekopart

名称	ID
墨尔本病理	墨尔本_病理学
Meli Melo	Meli_Melo
Melia	Melia
梅隆	梅隆
醇香蘑菇	Mellow_Mushr
Melodiya zdorov'ya	melodiya_zdorov'ya
旋律	旋律
Mema	Mema
MemberSource CU	MemberSource_CU
会员优势 CU	成员_优势_cu
会员 CU	成员_cu
CCNCU 会员精选 CU 分部	members_choice_cu_div_of_ccncu
会员精选 FCU	Members_Choice_FCU
会员交流 CU	会员交易所_cu
会员优先 CU	成员_first_cu
成员来源 CU	成员_source_cu
MembersOwn CU	MembersOwn_CU
Memo	Memo
孟菲斯	孟菲斯
工作中的男人	Men_At_Work

名称	ID
男士服装	男装_Wearhouse
Menards	Menards
Menchie 的冷冻酸奶	Menchie's_frozen_酸奶
门代尔	门代尔
门德雷斯	门德雷斯
Meness aptieka	Meness_aptieka
门富特酒店及度假村	Menfut_酒店_度假村
孟海晨生茶	Menghai_Chensheng_Tea
梦真教育	孟震_教育
Menicon	Menicon
梅诺	梅诺
Menta	Menta
孟席斯	孟席斯
Meralco	Meralco
Mercacentro	Mercacentro
Mercadorama	Mercadorama
眼镜市场	Mercadão_dos_Oculos
Mercafam	Mercafam
巴西商人	mercantil_do_brasil
商业银行	Mercantile_Bank

名称	ID
Mercantile Bshrs Corp	Mercantile_bshrs_Corp
Mercatone Uno	mercatone_Uno
墨卡托	墨卡托
奔驰	奔驰
商人银行	商人银行
商人轮胎和汽车	Merchant's_tire_&_auto
美居	美居
水星家具	水银_家具
水星药物	水银_药物
水星保险	水银保险
怜悯	怜悯
Mercy	Mercy_cu
Mercy Heal	怜悯生命值
Mercy Health	Mercy_Health_Sy
Meridia Community	Meridia_Community_cu
子午线	子午线
子午线信用合作社	Meridian_Credit_Union
经络油	Meridian_Oil
Merinos	Merinos
功绩	功绩

名称	ID
优异停车	Merit_Parking
美利通套房酒店	美利顿套房酒店
Meritta Bra	Meritta_Bra
Meriwest CU	meriWest_cu
Merkamueble	Merkamueble
Merko	Merko
默克赌场	Merkur_Casino
Merkur5	Merkur5
水星市场	水星市场
梅尔·诺曼	梅尔_诺曼
梅林	梅林
梅林电影院	梅林电影院
Merlinhod	Merlinhod
Meroil	Meroil
梅拉尔	梅拉尔
美林证券	Merrill_Lynch
美林财富管理	Merrill_Lynch_Wealth_Management
Merry Maids	Merry_Maids
快乐玛特	Merry_Mart
欢乐季	欢乐季

名称	ID
梅萨	梅萨
梅塞罗	梅塞罗
Meshiya Miyamoto	Meshiya_Miyamoto_Munashi
Meshur Adiyaman Cigkoftecisi	Meshur_Adiyaman_Cigkoftecisi
Meshur Peynirci	Meshur_Peynirci
梅斯拉	梅斯拉
Mester Grönn	Mester_Grönn
Mestergull	Mestergull
大都会食品	Met_Foods
MetLife	MetLife
Metalicus	Metalicus
金属投资银行	金属投资银行
Meten	Meten
Meteor	Meteor
流星停车	流星停车
Meters/Bonwe	Meters/Bonwe
Metex	Metex
Metkombank	Metkombank
Metoocate	Metoocate
Metr	Metr

名称	ID
Metrika	Metrika
地铁	地铁
大都会银行	Metro_Bank
地铁经纪人	大都会经纪人
Metro	Metro_cu
加拿大地铁	加拿大地铁_加拿大
中国地铁	Metro_China
Metro Diner	Metro_Diner
澳门轻轨电车	Metro_ligeiro_de_macau
地铁市场	Metro_Market [杂货店] _15
地铁市场	Metro_Market [杂货店] _78
麦德龙医疗大学	Metro_medical_cu
地铁摩托车	Metro_摩托车
纳米比亚地铁	Metro_纳米比亚
大都会石油	Metro_Oil
地铁公园酒店	Metro_Park_Hotel
大都会石油公司	地铁_石油
Metro 自助存储	Metro_Self Storage
Metro 鞋	Metro_Shoes
地铁超市	地铁_超市

名称	ID
Metro Turizm	Metro_Turizm
英国石油公司的地铁	Metro_at_BP
T-Mobile 的地铁	Metro_by_t-Mobile
Metro cencosud	Metro_cencosud
大都会银行	大都会银行
大都会电影院	大都会_电影院
大都会的	大都会的
大都会壁橱公司	大都会_壁橱_公司
大都会酒店	大都会酒店
大都会市场	大都会市场
大都会国民银行	大都会国民银行
大都会体育中心	大都会体育中心
大都会	大都会
Metropolradruhr	Metropolradruhr
Meta	Meta
MeuSuper	MeuSuper
Meublatex	Meublatex
Meuhedet	Meuhedet
墨西哥人	墨西哥人
Mezhtopenergobank	Mezhtopenergobank

名称	ID
Mezzaluna	Mezzaluna
意大利面中餐	mezzo_di_pasta
小米	小米
Mi Piaci	Mi_Piaci
小米商店	Mi_Store
我的老城镇	Mi_Viejo_Pueblito
Mi tienda del ahorro	mi_tienda_del_ahorro
MiNiMiNi	MiNiMiNi
Tanishq 的 Mia	mia_by_tanishq
Miam Miam	Miam_Miam
迈阿密订阅烧烤	Miami_Subs_Grill
绵堆面	绵堆面
绵阳城市商业银行	绵阳城市商业银行
云母	云母
云母硬件	云母_硬件
迈克尔·希尔	Michael_Hill
迈克尔斯	迈克尔斯
米哈尔·内格林	Michal_Negrin
Michel Dervyn	Michel_Dervyn
米歇尔·雷内	米歇尔_雷内

名称	ID
Michel 的糕点店	Michel's Patisserie
米其林	米其林
密歇根工厂！	Michigan_Works！
Mico	Mico
微型中心	微中心
Microchoix	Microchoix
微林斯	微林斯
微狂热	微狂热
Micromax	Micromax
微软商店	微软_Store
麦克罗特酒店及套房	Microtel_Inn_&_Suites
美国中部 CU	mid_american_cu
明尼苏达州中部足球俱乐部	明尼苏达州中部_FCU
中州银行和信托	Mid-State Bank_&_Trust
中州石油公司	中部各州_石油
MidFirst 银行	MidFirst_银行
MidSouth 银行	MidSouth_银行
MidSouth 社区 FCU	MidSouth_community_fcu
MidWestOne 银行	MidWestOne_银行
迈达斯	迈达斯

名称	ID
迈达斯鞋	Midas_Shoes
中海岸 FCU	Midcoast_FCU
美的	美的
米迪酒店及度假村	Midi_酒店及度假村
MidiCi 那不勒斯披萨公司	MidiCi_The_Neapolitan_Pizza_Company
米德兰加州大学	Midland_cu
美多里	美多里
中点	中点
中途岛酒店集团	Midway _酒店业
中途岛超市	中途岛_超市
中西部社区 FCU	Midwest_Community_FCU
Miel	Miel
Mifarma	Mifarma
Migato	Migato
强大的汽车零部件	Mighty_Auto_Parts
强力炸玉米饼	Mighty_Taco
Migrol	Migrol
Migrol Shop	Migrol_Shop
Migros	Migros
Migros DO IT +GARDEN	Migros_do_it+Garden

名称	ID
Migrosbank	Migrosbank
小米格尔的	Miguel's_JR
首尔米霍尔酒店	Mihoel_Seoul
工信部	工信部
Mikasa	Mikasa
迈克·戴维斯	Mike_Davis
迈克·佩罗	Mike_Pero
微电子	微电子
Mikyajy	Mikyajy
Mil MilkShakes	mil_ MilkShakes
米兰	米兰
米拉尔	米拉尔
Milavitsa	Milavitsa
Milenium	Milenium
迈尔斯	迈尔斯
里程碑	里程碑
Milex	Milex
Milio 的三明治	Milio's 三明治
牛奶屋	Milk_Houses
牛奶店	Milk_Shop

名称	ID
Milk-Agro	Milk-Agro
Milky Lane	Milky_Lane
米尔酒店及水疗中心	Mill_Hotel_&_Spa
千禧建筑信息模型	Millennium_BIM
千禧银行	千禧银行
千禧和国通	Millennium_and_copthorne
bcp 千禧年	Millennium_BCP
Miller	Miller
米勒和卡特	Miller_&_Carter
米勒电信	米勒_电信
米勒啤酒屋	Miller's Ale_House
米勒斯	米勒斯
米勒斯前哨基地	Millers_Outpost
Milli Pazar	Milli_Pazar
Milli Shoes	Milli_Shoes
Millie 的饼干	Millie's_cookies
百万	百万
米尔斯舰队农场	Mills_Fleet_Farm
CU Millstream	millstream_area_cu
米尔玛	米尔玛

名称	ID
Milo's	Milo's
Milo's 汉堡包	Milo's_Hamburgers
MiltonKeynes	MiltonKeynes
Mim	Mim
Mimco	Mimco
哑剧	哑剧
咪咪的小酒馆和面包店	mimi's_bistro_&_Bakery
Mimmin	Mimmin
Mimo & Co	Mimo_&_Co
Minds 英语学校	思想_英语_学校
我的毒品	我的毒品
Mine Yakkyoku	我的_Yakkyoku
Mineirão Atacarejo	Mineirao_Atacarejo
Ming Tien 咖啡语言	Ming_Tien_Coffee_Lan
明发	明发
名仕教育	名仕_教育
明思教育	明思_教育
明星百货	明星_百货
迷你银行	Mini_Bank
迷你卜拉欣	Mini_Brahim

名称	ID
迷你影院	迷你影院
迷你酒店	Mini_Hotel
迷你 Kalzone	Mini_Kalzone
迷你市场	迷你市场
迷你超市	Mini_Mart
迷你大奖赛	Mini_Prix
迷你石头火锅	Mini_Stone_Hot_Pot
迷你 U 存储空间	Mini_U_Storage
Minigrula	Minigrula
Miniklub	Miniklub
最低	最低
Minimercado Extr	minimercado_Extra
最低价格	最低价格
Minipreço 杂货	Minipreço_Grocery
Miniso	Miniso
迷你停靠站	迷你停靠站
Minit Mart	Minit_Mart
敏敏	敏敏
Minnco CU	minnco_cu
Minnequa Works	minnequa_works_

名称	ID
明尼苏达电力 ECU	明尼苏达州_power_ECU
明尼苏达谷 FCU	Minnesota_Valley_FCU
米妮未成年人	Minnie_Minors
Minnoco	Minnoco
Minol	Minol
敏润超市	敏润超市
闽山大酒店	闽山大饭店
闽山饭店	闽山_酒店
民生农村银行	民生_农村_银行
Minsun	Minsun
薄荷	薄荷
Mint 酒店	薄荷酒店
Minuet Petite	Minuet_Petite
Minuit	Minuit
民兵出版社	民兵_Press
Minuto Pao de Acucar	minuto_pao_de_Acucar
敏新	敏新
敏众医疗	敏中_医疗保健
Miqi	Miqi
Mir Lekarstv	mir_Lekarstv

名称	ID
Mir Obuvi	mir_Obuvi
Mir Zdorov'ya	mir_zdorov'ya
Mira La Marela	Mira_La_Marela
Mirabito	Mirabito
奇迹汽车喷漆和车身维修	Miracle_Auto_Painting_&_Body_Repa
奇迹方法	奇迹_方法
奇迹之耳	奇迹之耳
米拉达	米拉达
米拉杰电影院	Miraj_电影院
Mirako 健身	Mirako _健身
Miratorg	Miratorg
米尔迪夫市中心	米尔迪夫城市中心
镜子和绘画	镜子和画作
米罗	米罗
Carnes Parrilla 小姐	Mis_Carnes_Parrilla
Costillitas 小姐	Mis_Costillitas
我最好的假期	Mis_Mejores_Vacaciones
Misabel	Misabel
Misas	Misas
三岛信用金库	三岛信用金库

名称	ID
Mishref 合作社协会	Mishref_Co-o-operative_S
Misoxi	Misoxi
Misr 保险	Mir_Insurance
Misr 伊朗开发银行	Misr_伊朗_开发银行
Coquines 小姐	Miss_Coquines
Etam 小姐	Miss_Etam
成都的傅小姐	Fu_In_成都小姐
卡普里兹小姐	Miss_Kapriz
诗小姐	诗歌小姐
Rotenmeyer 小姐	Miss_Rotenmeyer
塞尔弗里奇小姐	Selfridge 小姐
Missha	Missha
Missil	Missil
密西西比 FCU	密西西比_FCU
Missoni	Missoni
巴索先生	Mister_Baso
Mister Bed	Mister_Bed
洗车先生	Mister_Car_Wash
夫人先生	Mister_Lady
兰德里先生	Mister_Landri

名称	ID
Mister Minit	Mister_Minit
披萨先生	先生_Pizza
墨西哥披萨先生	Mister_Pizza_Mexic
传送先生	Mister_Trans
佐尔先生	Mister_Zol
Misto Bank	Misto_Bank
Mistral	Mistral
Mit Thae 租赁	Mit_Thae_Leasing
三田制面所	Mita_Seimenjo
三田井谷	三田井谷
米切尔鱼市场	米切尔鱼市场
水户证券	美图证券
水户信用金库	水户信用金库
Mitr Sin Duan	Mitr_Sin_Duan
米特拉	米特拉
Mitre 10	Mitre_10
三菱商事能源	三菱_株式会社_能源
三菱日联信托银行	三菱_ufj_trust_Bank
三井石友	Mitsui_Sekiyu
光冈	光冈

名称	ID
Mittare 保险	Mittare _保险
Mittelbrandenburgische Sparkasse	Mittelbrandenburgische_Sparkasse
混合	混合
混合 Mateus	Mix_Mateus
混合土豆	混合土豆
混为一谈	Mix_Up
Mixc	Mixc
mixuebingcheng	mixuebingcheng
Miyabi 烧烤	Miyabi Grill
宫越屋咖啡	宫越屋_咖啡
宫崎第一信用金库	宫崎_第一_信用金库
Miyoshino	Miyoshino
Mizhilian	Mizhilian
Mizhnarodnyi Investytsiinyi Bank	Mizhnarodnyi_Investytsiinyi_Bank
Mizhnarodnyi Ipotechnyi Bank	Mizhnarodnyi_Ipotechni_Bank
Mizzoni	Mizzoni
mjstyle	mjstyle
Mk One	Mk_One
Mk2	Mk2
Mkb Romexterra 银行	Mkb_Romexterra_Bank

名称	ID
Mladinska 书籍	mladinska_knjiga
Mlin i pekare	mlin_i_pekare
Mlinar	Mlinar
mlinotest	mlinotest
mmcinem	mmcinem
Mongo Mebeli	Mnogo_Mebeli
莫莫天堂	Mo_Mo_Paradise
Mo'men	Mo'men
Mo's Mobiles	mo's_mobiles
Moat House 酒店	Moat_House_Hotels
Mobi	Mobi
Mobi Cash	Mobi_Cash
Mobifone	Mobifone
美孚	美孚
移动元素	移动元素
Mobil Mart	Mobil_Mart
移动教育工作者 CU	移动教育工作者_cu
Mobilelink	Mobilelink
Mobilia	Mobilia
法国家具展	Mobilier_de_France

名称	ID
Mobilink	Mobilink
Mobilis	Mobilis
出行信用合作社	出行_信用合作社
Mobilochka	Mobilochka
Mobily	Mobily
Mobinil	Mobinil
Mobiphone	Mobiphone
Mobistar 中心	Mobistar_Center
Mobistore	Mobistore
Moblerone	Moblerone
Moc Jakosc Zysk	Moc_Jakosc_Zysk
Moca-Loca	Moca-Loca
摩卡咖啡馆和酒吧	Mocha_cafe_&_bar
Mochachos 鸡村	Mochachos_Chicken_Villages
麻薯	麻薯
麻薯糖果	年糕_甜点
Moco	Moco
Mod 的头发	mod's_hair
Moda Giyim	Moda_Giyim
Moda Life	Moda_Life

名称	ID
Pelle 中的时尚	Moda_in_pelle
玛丽	模特_玛丽
模式关闭	Mode_Off
模特的体育用品	模特体育用品
Modelo	Modelo
Fraser 的 Modena	Modena_by_fraser
现代	现代
现代诊断与研究中心	现代诊断与研究中心
美国现代伍德曼	美国的现代木人
现代广场	现代广场
Modis	Modis
Moe 的西南烧烤	moe's_southwest_Gril
Mogu 教育	Mogu_教育
莫霍克	莫霍克
Mohácsi Takarék 银行	Mohacsi_Takarek_Bank
Moj Kiosk	Moj_Kiosk
Moja SAMOSKA	Moja_samoska
我的书店	moje_lekarna
魔法	魔法
摩卡之家	Moka_House

名称	ID
Moko Tammen Nakam	MOKO_Tammen_Nakamoto
Mol Shop	Mol_Shop
Moldura Minuto	Moldura_Minuto
莫莉幻想	莫莉_幻想
Molly Maid	Molly_Maid
Molniya	Molniya
妈妈和我	妈妈_&_我
动量	动量
Momi & Toy's	Momi_&_Toy's
Momibook	Momibook
红叶日本料理	红叶_日本料理
Momma Goldberg's Deli	momma_Goldberg's deli
Mon Cher	Mon_Cher
Mon Hue	星期一_Hue
蒙娜丽莎	Mona_Lisa
摩纳哥电信	摩纳哥_电信
Monarch 牙科	Monarch_Dental
帝王油	Monarch_Oil
莫纳克酒店	Monarque_Hoteles
Monbo	Monbo

名称	ID
Monceau Fleurs	Monceau_Fleurs
Mondi	Mondi
Mondial Pare-Brise	Mondial_Pare-Brise
蒙迪亚尔纸巾	Mondial_Tissus
Monetka	Monetka
Money Mailer	Money_Mailer
国际资金管理	资金管理_国际
理财集市	Money Mart
MoneyGram	MoneyGram
MoneyOne	MoneyOne
金钱树	金钱树
Monginis	Monginis
Mongog	Mongog
蒙古语快速停车	蒙古语_Quick_Stop
Moniberic	Moniberic
Monical 披萨	莫尼卡_披萨
Monical 的披萨	Monical's Pizza
花押字	花押字
Mono'	Mono'
Mono'bea	Mono'bea

名称	ID
Monoprix	Monoprix
Monoprix EV	Monoprix_EV
Monrif	Monrif
门罗	门罗
门罗汽车服务和轮胎中心	Monro_auto_Service_and_Tire_Centers
Monroe Telco FCU	Monroe_telco_FCU
季风	季风
季风沙龙	季风沙龙
Montagu 干果和坚果	Montagu_Dried_Fruit_&_Nuts
蒙大拿快车	蒙大拿州快车
蒙大拿烧烤	蒙大拿州烧烤
蒙大拿州迈克的	蒙大拿州_Mike's
蒙大拿州的	蒙大拿州的
蒙塔托	蒙塔托
蒙贝尔	蒙贝尔
蒙特卡罗	蒙特卡洛
蒙特帕斯基银行	Monte_Paschi_Banque
Montedison	Montedison
蒙特皮奥	蒙特皮奥
蒙特皮奥·杰拉尔	Montepio_Geral

名称	ID
Monter	Monter
蒙特祖玛的	蒙特祖玛的
蒙托汽车	Monto_Motors
蒙特塞拉特岛	蒙特塞拉特岛
月亮	月亮
MoonStar	MoonStar
摩尔人	摩尔人
Mooyah	Mooyah
Mopas	Mopas
莫尔·森	Mor_Seng
更多	更多
更多 & 更多	更多_&_更多
更多 Chaya	更多_Chaya
摩根和摩根	Morgan_&_Morgan
摩根士丹利	Morgan_Stanley
摩根士丹利史密斯巴尼	Morgan_Stanley_Smith_Barney
Mori-no-Miyako 信用金库	Mori-no-Miyako_shinkin_Bank
盛冈信用金库	盛冈_信用金库
森田	森田
Morkels	Morkels

名称	ID
Morley's	Morley's
Morni	Morni
Morning	Morning_Hill
晨间旅馆	Morning_Inn
晨鸡	晨鸡
莫宁顿	莫宁顿
摩洛哥风味	摩洛哥_Taste
莫里斯	莫里斯
莫里斯·默多克旅行	Morris_Murdock_Travel
莫里斯油	Morris_Oil
莫里斯专卖店	Morris_Specialty_store
Morskyi	Morskyi
莫顿大厦	莫顿_建筑物
莫顿的 The Steakhouse	Morton's_the_the_steak
Moschino	Moschino
莫斯科停车	莫斯科_停车
Moskommertsbank	Moskommertsbank
Moskovskaya Yarmarka	Moskovskaya_Yarmarka
莫斯科夫斯基工业银行	Moskovskiy_Industrial'nie_Bank
莫斯科夫斯基信用银行	Moskovskiy_Creditniy_Bank

名称	ID
Moskovskiy yuvelirniy zavod	moskovskiy_yuvelirniy_Zavod
Moskovskoe Vremya	Moskovskoe_Vremya
莫斯克维奇	莫斯克维奇
Mosoblbank	Mosoblbank
Mosoblfarmatsiya	Mosoblfarmatsiya
蚊子乔	Moskito_Joe
Moss Bros	Moss_Bros
Mossimo	Mossimo
Mostaza	Mostaza
Mostsvettorg	Mostsvettorg
168 号汽车旅馆	汽车旅馆_168
6号汽车旅馆	汽车旅馆_6
一号汽车旅馆	Motel_One
妈妈乳制品	母乳制品
孕产妇产妇	孕产_产妇
反对酒后驾车的母亲	Mothers_Agast_Drunk_Driving
母亲的骄傲	妈妈们_骄傲
动感行业	影视行业
摩托	摩托
摩托超市	Moto_Mart

名称	ID
摩托摄影与人像工作室	Moto_Photo_&_Portailit_Stud
元町咖啡	元町_咖啡
汽车银行	汽车银行
Motta	Motta
希尔顿座右铭	希尔顿的座右铭
Moulin Etape	Moulin_Etape
勃朗山	Mount_Blanc
夏洛特山蓟草	Mount_Charlotte_Thi
Mount Litera Zee 学校	Mount_Litera_Zee_S
梅鲁山石油公司	Mount_Meru_Petrole
美国山区信用合作社	Mountain_America_信用联盟
山地 CU	山脉_cu
山地设计	山地_设计
山地硬装	山地_硬装
迈克山披萨	Mountain_mike's _P
加州山河	山河_cu
山地仓库	山地仓库
Mountancrest	MountainCrest_cu
芒特菲尔德	芒特菲尔德
移动	移动

名称	ID
Movele	Movele
Mover 英语	Mover_英语
Movida	Movida
电影明星	Movie_Star
Movietime 电影院	Movietime_电影院
莫维弗洛尔	莫维弗洛尔
正在移动	正在移动
Movitel	Movitel
Moxe	Moxe
Moxie's	Moxie's
Moxy 酒店	Moxy_Hotels
Moya apteka	moya_apteka
Moyat 杂志	Moyat_magazin
Moza Banco	Moza_Banco
莫扎特冰淇淋	莫扎特_冰淇淋
Mpreis	Mpreis
Mps 微画颜料	Ms_Micropain
Mpsa	Mpsa
先生	先生
Bigg 先生的	Mr_Bigg's

名称	ID
离合器先生	Mr_Cluth
菲什先生	Mr_Fish
Gas 先生	Mr_Gas
英雄先生	Mr_Hero
杰夫先生	先生_Jeff
吉姆先生的披萨	Mr_Jim's_Pizza
Lube 先生	Mr_Lube
M 先生	Mr_M
迈克先生的	Mr_Mike's
Muffler 先生	Mr_Muffler
皮塔先生	Mr_Pita
普莱斯先生	Mr_Price
普莱斯先生家	Mr_Price_Home
普莱斯体育先生	Mr_Price_Sport
赖斯先生	Mr_Rice
Rooster 先生	Mr_Rooster
Sub 先生	Mr_Sub
Subb 先生	Mr_Subb
寿司先生	Mr_Sushi
Tire 先生	Mr_Tire

名称	ID
传输先生	传输先生
mr-Onion	mr-Onion
电器先生	设备先生
贝克先生	_Baker 先生
巴克先生	Mr._Bark
Bricolage 先生	Mr._Bricolage
猫先生	Mr._Cat
Cigkofte 先生	Mr._Cigkofte
Doors 先生	门先生_Doors
电气先生	_Electric
Fuel 先生	Mr._Fuel
杂工先生	Mr._Handyman
Kitsch 先生	Mr._Kitsch
Kumpir 先生	_Kumpir 先生
马克·贝克先生	Mr._Mark_Bak
Mix 先生	Mr._Mix
披萨先生	先生_Pizza
椒盐脆饼先生	Mr._Pretzels
Rooter 水暖先生	先生_Rooter_Plumbing
Wash 先生	Mr._Wash

名称	ID
Wish 先生	Mr._Wish
Blue 先生	Blue 先生
微风先生咖啡	微风先生_咖啡
Mr.Eel 的爱	Mr.eel's_LOVE
英吉利先生	英吉利先生
Pizza 先生葡萄牙	Pizza 先生_葡萄牙
视频先生	视频先生
音乐先生	音乐先生
MrMax	MrMax
Mriia	Mriia
菲尔兹夫人	Fields 夫人
健身夫人	健身 Mrs_
Vanelli 太太的	Mrs_Vanelli's
Sporty 太太	Mrs._Sporty
Winner 太太的鸡肉和饼干	mrs._winner's chicken_&_Biscuits
周太太	周太太
Mrsfaiths	Mrsfaiths
mt-Toast.com	mt-Toast.com
山地车	山地车
Mtek	Mtek

名称	ID
Mtel	Mtel
Mtel Imax 3D	mtel_imax_3D
Mtsr	Mtsr
Muaa	Muaa
Mubarak Al-Kabeer 合作社	mubarak_al-Kabeer_co-op_Society
很多运动鞋	Much_运动鞋
Muchas Gracias 墨西哥菜	Muchas_Gracias_墨西哥_食品
Mucho Burrito	Mucho_Burrito
泥湾	Mud_Bay
Mudo	Mudo
Mueble Rustiko	Mueble_Rustiko
Muerde La Pasta	Muerde_La_Pasta
松饼休息时间	Muffin_Brea
便装	便装
muj obchod	muj_obchod
无神藏	无神藏
Mujosh	Mujosh
Mukta A2 电影院	Mukta_A2_Cinemas
桑树	桑树
穆林斯和普里查德	Mullins_&_Prichard

名称	ID
多功能烘烤	多重烘烤
MultiBank	MultiBank
MultiCard	MultiCard
MultiMarket	MultiMarket
MultiOpticas	MultiOpticas
MultiPack	MultiPack
MultiPark	MultiPark
大通银行	大通银行
Multica	Multica
多重医疗药房	多重医疗_药房
Multicinemas	Multicinemas
多药气	多药气
多重农力	多重农力
Multikino	Multikino
Multiloja	Multiloja
万能的	万能的
多药业	多药业
Multiplex 联合新影院	Multiplex_United_New_
Multirama	Multirama
多重存储	多重存储

名称	ID
多约克	多约克
孟买地区中央合作银行	孟买地区中央合作银行
Mumuso	Mumuso
Mumy Room	Mumy_Room
Mundicenter	Mundicenter
世界马可	Mundo_Marco
Mundo Moody	Mundo_Moody
世界佛得角	Mundo_Verde
蒙多夫	蒙多夫
俄克拉荷马城市政ECU	俄克拉荷马城的市政行政区_ecu_of
Murco	Murco
室兰信用金库	Muroran_Shinkin_Bank
墨菲快车	墨菲快车
美国墨菲	Murphy_USA
Murphy's Deli	Murphy's_deli
武藏证券	武藏证券
马斯卡特药房	马斯喀特_药店
肌肉萎缩症协会	肌肉萎缩症_协会
火腿博物馆	Museo_del_Jamon
音乐作品	音乐作品

名称	ID
Muskogee FCU	muskogee_fcu
穆斯林商业银行	穆斯林商业银行
Mussol	Mussol
Musta Pörssi	Musta_Pörssi
野马电影院	野马影院
Musti ja Mirri	musti_ja_mirri
Muthoot 财经	Muthoot_Finance
Muthoot Fincorp	Muthoot_Fincorp
Muthoottu Mini	Muthoottu_Mini
Muthu 药房	Muthu_药房
穆蒂亚拉银行	Mutiara_Bank
Mutlu aku	mutlu_aku
互惠通用	互惠环球
互助 CU	mutual_cu
互助毒品	Mutual_Drug
相互优先 FCU	mutual_first_FCU
共同储蓄 CU	mutual_savings_cu
奥马哈互助会	Mutual_of_Omaha
我的欧尚	My_Auchan
我的化学家	My_Chemist

名称	ID
我的牙医	我的牙医
我最喜欢的松饼	My_Favorite_Muffin
My Feel 酒店	My_Feel_Hotel
我的健身房	My_Gym
我的医疗保健 FCU	my_Healthcare_fcu
我的家酒店	我的家酒店
我的自制牛排	My_Home_Steak
我的本地人	My_Local
我的鞋子	我的鞋子
我的套房公寓式酒店	My_Suite_Appart-Hotels
我的旅行	My_Travel
我的卡车清洗	My_Truck_Wash
我温暖的一天	My_Warm_Day
我的区域酒店	My_Zone_Hotel
我的市场	我的市场
MyChoize	MyChoize
MyEyeDr.	MyEyeDr.
MyMydin	MyMydin
MyPoint cu-前身为 Point Loma CU	MyPoint_cu-formerly_point_loma_cu
MyShop	MyShop

名称	ID
缅甸Apex银行	缅甸_Apex_Bank
缅甸经济银行	缅甸_经济银行
Myasnitskiy Ryad	Myasnitskiy_Ryad
Myasnoy Dvorik	Myasnoy_Dvorik
Mycom	Mycom
Mydin Mart	Mydin_Mart
我的王国	我的王国
迈尔斯	迈尔斯
我的牙科	Myour_Dental
神秘云南民族菜	神秘南方云南民族菜
MÖBELRINGEN	MÖBELRINGEN
Markische Bank	Markische_Bank
Meditel	Meditel
移动快车	Mövil_Express
Möbelix	Möbelix
Mövenpick	Mövenpick
瑞享咖啡厅	瑞享咖啡厅
Möller Bil	Möller_Bil
Müller	Müller
慕尼黑银行	慕尼黑银行

名称	ID
Müritz-Sparkasse	Müritz-Sparkasse
N-真	N-真
N.E.W. CU	N.E.W._CU
N1	N1
不！鞋子	不！ _Chaussures
NAB	NAB
NAC	NAC
NADRA 和 Sahulat	nadra_e_sahulat
名字 FCU	NAE_FCU
NAF NAF NAF	NAF_NAF
NAFNAF	NAFNAF
NAFT	NAFT
NAGEM	NAGEM
NAMB	NAMB
南城百货	南城_百货大楼
南京证券	南京证券
纳诺斯	纳诺斯
南苑客栈	南苑客栈
奈良井披萨店	NARAI_PIZZERIA
NARODNE NOVINE	NARODNE_NOVINE

名称	ID
NAS	NAS
NAS JRB CU	NAS_JRB_CU
NASBank	NASBank
NASH	NASH
纳舒厄手机	NASHUA MOBILE
NASZA APTEKA	NASZA_APTEKA
娜塔莉斯	娜塔莉斯
纳坦	纳坦
澳大利亚国民银行	澳大利亚国民银行
阿布扎比国家银行	阿布扎比国家银行
埃及国家银行	埃及国家银行
富查伊拉国家银行	富查伊拉国家银行
国家兴业银行	NATIONAL_SOCIETE_GENERALE_BANK
国家银行	国家银行
大自然的房子	自然之家
当然 JOJO	NATURALLY_JOJO
NATUREL PEN	NATUREL_PEN
NAUTALIA	NAUTALIA
导航	导航
NAVILLE	NAVILLE

名称	ID
海军蓝和绿色	海军蓝_&_绿色
纳亚拉能源	Nayara_energy
纳扎里	纳扎里
NB	NB_
NBC	NBC
NBC 资本公司	NBC_Capital_Corp
NBG 银行	NBG_Bank
NBK	NBK
NBP	NBP
NBS 银行	NBS_Bank
NBT 银行	NBT_Bank
NC	NC
NCB	NCB
NCP	NCP
一只老虎	NE_TIGER
神酒	神酒
NEFTIKA	NEFTIKA
NEONET	NEONET
新公园	新公园
NEOS BANCA	NEOS_BANCA

名称	ID
NEOSET	NEOSET
NES 租赁	新房租赁
奈斯派索	奈斯派索
筑巢	筑巢
筑巢油	NESTE_OIL
NESTE 石油快报	Neste_oil_Expres
NET	NET
NETOIL	NETOIL
NETOIL 市场	Netoil_Marke
NETTO	NETTO
NETTO 法国	Netto_Franc
NETTOYE 汽车	NETTOYE_AUTO
NEUROTH HORCENTER	NEUROTH_HORCENTER
新世纪百货	新世纪百货
新人	NEW_MAN
新世界酒店	新世界酒店
《纽约客》	NEW_YORKER
新日子	新日子
新岛咖啡	新岛咖啡
纽曼	纽曼

名称	ID
新闻咖啡厅	新闻咖啡厅
NEX	NEX
NEXA	NEXA
Ng Tiu Yan Tong Sui	ng_tiu_yan_tong_sui
NH	NH
NH 银行	NH_Bank
NH-OIL	NH-OIL
苏格兰国民保健服务	NHS_苏格兰
笔尖	笔尖
NIB 银行	Nib_bank
NIC 银行	NIC_Bank
不错 ioi	nice_ioi
NICI	NICI
NIIT	NIIT
耐克	耐克
尼尔森鞋	NILSON_SHOES
九点到九点	NINE_TO_NINE
九号西	NINE_WEST
九十四杯咖啡	九十四_咖啡
忍者	忍者

名称	ID
没有	没有
NIPCO	NIPCO
日本蓝	NIPPON_BLUE
NIRVANA FITNESS&SPA	NIRVANA_FITNESS&SPA
NIS 汽油	NIS_PETROL
日产	日产
日产聆风	NISSAN_LEAF
NISSEN	NISSEN
NITOS	NITOS
NIUTRON	NIUTRON
NIWel 美丽	NIWel_美容
NK Al'yans	nk_al'yans
NKD	NKD
NLB Banka	nlb_Banka
NLB Montenegrobanka	nlb_MontenegroBanka
NMA FCU	NMA_FCU
NMB	NMB
NNPC	NNPC
NO.1 台湾茶	NO.1_TAIWAN_TEA
NOA NOA	NOA_NOA

名称	ID
诺尔油	NOAL_OIL
NOBIL	NOBIL
贵族房子	NOBLE_HOUSE
贵族海鲜餐厅	贵族海鲜餐厅
NOCO	NOCO
游牧酒店	NOMAD_HOTELS
NOMI	NOMI
Nomos-Bank	Nomos-Bank
非单人运动	NON_SOLO_SPORT
不间断	不间断
NONO	NONO
NOODLE 13	NOODLE_13
面条吧	面条_酒吧
带水坑的面条	NOODLE_W_PUDLE
努尔伊斯兰银行	NOOR_ISLAMIC_BANK
NORDBLANC	NORDBLANC
北欧健康	北欧健康
NORDSTROM 机架	Nordstrom_rack
NORMA	NORMA
寻北者	探北者

名称	ID
北方鱼	北方鱼
北门	北门
德克萨斯州北极星银行	德克萨斯州北极星银行
德克萨斯州北极星银行-抵押贷款部	Northstar_bank_of_Texas_-_Mortgage_Division
德克萨斯州北极星银行-财资部	Northstar_bank_of_Texas_-_P&E_Division
NORVIK BANKA	NORVIK_BANKA
不是	不是
NOTTURNO ITALIANO	NOTTURNO_ITALIANO
NOU-HAU	NOU-HAU
新星	新星
新银行	NOVA_BANKA
NOVA PEIXAR	NOV_PEIXARIA
NOVA 农场	Nova_Farma
NOVA 投资	Nova_Inv
NOVAK maso-uzeniny	novak_maso-uzeniny
小说	小说
NOVIKOMBANK	NOVIKOMBANK
NOVO BANCO	NOVO_BANCO
NOVOFARM	NOVOFARM
诺富特	诺富特

名称	ID
诺富特咖啡厅	诺富特咖啡厅
NOVUS 玻璃	novus_Gl
NP-Markt	NP-Markt
NPC 24H	NPC_24H
NPC 加油站	NPC_Gas Station
NPCO	NPCO
NPD	NPD [parking_garage-parking_house]
NPD	NPD [汽油加油站]
NRMA	NRMA
NRSP 银行	NRSP_bank
NRgize	NRgize
NSCB	NSCB
NSIA 银行	NSIA_BANQUE
不是	不是
北领地商店	NT_shop
NTI 快车汽车保养	nti_express_auto_Car
NTT·	NTT_·
NU BA	NU_BA
NU PAST	NU_PASTA
NUBA Viajes	Nuba_Viajes

名称	ID
NURBANK	NURBANK
NUTRISA	NUTRISA
疯狂的巴伐利亚人	NUTTY_BAVARIAN
NW Plus	nw_plus_cu
新泽西珠宝	nwj_Jewell
NY TEAM FCU	NY_TEAM_FCU
NYCE	NYCE
NYPFC	NYPFC
NZCU	NZCU
Na Kolyosakh	Na_Kolyosakh
Na zdorov'e	na_zdorov'e
na zdorov'ye	na_zdorov'ye
NaBe 一个	NaBe_一个
Nabab	Nabab
Nacex	Nacex
国家的	国家的
Nacional Inn	Nacional_Inn
纳德邦	纳德邦
Naeem 合作社	Naeem_co-Operative_Society
Naf Naf Grill	Naf_Na_Grill

名称	ID
Naftako	Naftako
Naftal	Naftal
长野信用金库	长野信用金库
长野县信用合作社	长野县_信用合作社
长野县劳动银行	长野县劳动银行
长崎茶门	长崎_Chanmen
长城	长城
不 & Frisch	不_&_Frisch
Naheola CU	naheola_cu
纳伊夫鸡	纳伊夫_Chicken
奈因	奈因
Nails4'Us	Nails4'Us
奈瑞尔	奈瑞尔
内藤翔健	Naito_Shoken
天真蓝	Naive_Blue
Naivedyam	Naivedyam
Naixuedecha	Naixuedecha
纳佳餐厅	Najia_餐厅
中兵库信用金库	Naka-Hyogo_shinkin_Bank
中野沟信用合作社	Nakanogou_Credit_合作社

名称	ID
森山中津 Karaage	中津_炸鸡_森山
纳霍德卡	纳霍德卡
纳克沙特拉	纳克沙特拉
纳库马特	纳库马特
南安银行	Nam_A_Bank
南光石油	Nam_Kwong_Oil
Namdhari's Fresh	Namdhari's_fresh
郭楠火	郭楠火
南山人寿	Nan_Shan_Life
Nana Biosupermermer	Nana_biosupermerm
南昌农村商业银行	南昌_农村_商业银行
南城翔	南城翔
南充市商业银行	南充城市商业银行
南希的披萨	Nancy's_Pizza
南丹游记	Nandan_Travels
Nandilath G Mart	Nandilath_G_Mart
Nando's	Nando's
南芬农村商业银行	南芬农村商业银行
南海农村商业银行	南海_农村_商业银行
南华期货	南华期货

名称	ID
南京大牌堂	南京_daipaidang
南门双柔	南门_Shuanrou
南宁百货	南宁_百货_大楼
Nanobox 开车然后你	nanobox_by_car_and_you
Nanou甜甜圈屋	Nanou_Donuts_House
南浦	南浦
南台湾	南台湾
南通农村商业银行	南通农村商业银行
娜努娜娜	Nanu_Nana
南勋银行	南讯银行
南洋商业银行	南阳_商业银行
南洋乡村银行	南阳乡村银行
Nanz	Nanz
南洲牙科	南州口腔_口腔医院
纳帕汽车零部件	纳帕汽车零部件
纳帕汽车保养中心	Napa_汽车护理中心
Napapijri	Napapijri
拿坡里披萨	那不勒斯_披萨
拿破仑·珀迪斯	Napoleon_Perdis
那不勒斯	那不勒斯

名称	ID
小睡	小睡
奈良泰国料理	奈良_泰国菜
君澜大酒店	南都君澜大酒店
君澜度假村及水疗中心	君澜度假村及水疗中心
匿名毒品	麻醉品_匿名
Narodnaya 7ya sem'ya	narodnaya_7ya_sem'ya
Narodnaya apteka	narodnaya_apteka
狭窄	狭窄
纳尔斯	纳尔斯
纳尔维森	纳尔维森
纳斯卡咖啡厅	Nasca_Cafe
Nascia	Nascia
纳西姆合作社	Naseem_co-Operative_Society
Nash Krai	Nash_Krai
Nash doctor	nash_doktor
Nasha apteka	Nasha_apteka
Nashe Zoloto	Nashe_Zoloto
椰浆饭	椰浆饭
Nasi Kandar Pelita	Nasi_Kandar_Pelita
拿骚金融 FCU	Nassau_financial_FCU

名称	ID
Nassauische Sparkasse	Nassauische_Sparkasse
纳赛尔社会银行	Nasser_Social_Bank
纳赛尔体育中心	纳赛尔体育中心
Nasz Sklep	Nasz_Sklep
NatWest	NatWest
纳塔尔批发珠宝商	Natal_Wholesale_Jewelers
内森的名人	Nathan's_Famous
国家的巨型汉堡包	Nation's Giant_murgers
国家建设酒店	国家建设_酒店
国家九号汽车旅馆	National_9_Motel
全国 ATESA	National_atesa
国家汽车玻璃	全国汽车玻璃
国家银行	国民银行
哈伊马角国家银行	拉斯海玛国家银行
国家农业银行和 dvlp't	National_bank_of_agr_and_dvlp't
亚利桑那国家银行	亚利桑那国民银行
巴林国家银行	巴林国家银行
国家商业银行	国家商业银行
希腊国家银行	希腊国民银行
科威特国家银行	科威特国民银行

名称	ID
阿曼国家银行	阿曼国民银行
南卡罗来纳州国民银行	南卡罗来纳州国民银行
Umm Al Qaiwain 国家银行	National_bank_of_umm_al_qaiwain
美国国家银行	美国国民银行
全国书店	全国书店
国家停车场	国家汽车公园
全国租车	National_租车
国家公民银行	国民公民银行
国家康尼岛	国家_康尼岛
国家地下水测量师	National_地下水测量师
国家卫生实验室	国家卫生实验室
全国听力保健	全国听力护理
国家保险公司	国民保险公司
爱尔兰国民银行	爱尔兰国民银行
国家石油	国家石油公司
宾夕法尼亚国家银行股份公司	National_Penn_Banshares_Inc.
国家药房	全国_药房
国家财产检查	国家财产检查
国民储蓄	国民储蓄
国家轮胎和电池	National_Tire_&_Battery

名称	ID
National 轮胎和汽车保养	National_Tyres_&_Autocare
全国面包车线	National_Van_Lin
国家/城市	国家/城市
全国建筑协会	全国建筑协会
全国碰撞维修中心	全国碰撞维修中心
全国地板和窗帘	全国_地板_和_窗户_覆盖物
全国保险	全国_保险
原生烧烤和翅膀	Native_Grill_and_Wings
纳特拉德	纳特拉德
Nattex	Nattex
Nattex Descanso	Nattex_Descanso
自然产品	天然产品
Natura	Natura
Natura 专卖店	Natura_Specialty_st
Naturasí	Naturasí
自然护理	Natural_Care
天然杂货店	Natural_Grocers
自然厨房	自然厨房
自然劳森	Natural_Lawson
自然外观	自然外观

名称	ID
天然 Pawz	Natural_Pawz
天然新鲜冰淇淋	天然新鲜冰淇淋
纳图拉利亚	纳图拉利亚
自然化者	自然化者
Naturals 沙龙	Naturals_Salon
自然与探索	Nature_&_Decouvertes
自然厨房	自然_厨房
大自然的篮子	Nature's Basket
大自然的农场	Nature's_Farm
大自然的桌上咖啡厅	Nature's_table_Cafe
Naturcompaniet	Naturcompaniet
Natuur pop	natuur_pop
Natuurwinkel	Natuurwinkel
Natuzzi	Natuzzi
顽皮	顽皮
Naugles	Naugles
Naumi 酒店	Naumi_Hotel
Nautica	Nautica
Nav-Star	Nav-Star
Naveco	Naveco

名称	ID
Naveo CU	naveo_cu
Navis Pack & Ship	Navis_Pack_&_Ship
海军联邦信用合作社	海军_联邦信用合作社
不 Elektrodom	Nay_Elektrodom
Naya Sawera	Naya_Sawera
Naya Tours	Nay_Tours
Nayomi	Nayomi
Naza	Naza
纳扎尔	纳扎尔
NeNe 鸡	NeNe_鸡
脖子和脖子	脖子_&_脖子
Neckermann	Neckermann
内德班克	内德班克
需求	需求
尼姆拉纳酒店	Neemrana_Hotel
Neeru's	Neeru's
Neethi 医疗商店	Neethi_Medical_Stores
neft'magistral'	neft'magistral'
Neftekhimprom	Neftekhimprom
Nefteke	Nefteke

名称	ID
Netto	Netto
根岸	根岸
Nehoda Cz	Nehoda_Cz
邻里大学	邻里_cu
邻居	邻居
邻居 CU	邻居_cu
邻居 Dalianhuoshao	邻居_大连火烧
邻居 FCU	邻居_FCU
内江兴隆农村银行	内江_兴隆_农村银行
内曼·马库斯	Neiman_Marcus
Nektar	Nektar
Nekter 果汁吧	Nekter_Juice_Bar
Nelda	Nelda
Nelo Gelato	Nelo_Gelato
纳尔逊·舍嫩	Nelson_Schoenen
尼尔森的	尼尔森的
Nemetskaya obuv'	Nemetskaya_obuv'
Nena 的 FRUTERIA	Nena's_Fruteria
Nendels Inn	Nendels_Inn
Neobio 家庭公园	Neobio_Family_Par

名称	ID
Neocine	Neocine
Neofarm	Neofarm
Neoluxor	Neoluxor
那不勒斯披萨	Neopolitan_Pizza
Neotel	Neotel
NephroCare	NephroCare
Neposeda	Neposeda
Neprospi	Neprospi
Nepstar 药店	Nepstar_Drug_Store
Nervol	Nervol
雀巢咖啡架	雀巢咖啡支架
Neste K	Neste_K
雀巢	雀巢
内斯特罗石油公司	Nestro_Petrol
Neta	Neta
Netcare	Netcare
Neto	Neto
Netto Getranke-Discoun	Netto_Getránke-Discoun
Netto Marken-折扣	Netto_Marken-Discount
Netto 超级市场	Netto_Supermercados

名称	ID
Nettorama	Nettorama
网络	网络
网络视频	网络_视频
Neue Aargauer Bank	Neue_Aargauer_Bank
诺伊豪斯	诺伊豪斯
诺伊曼	诺伊曼
诺伊门	诺伊门
内华达州鲍勃队	Nevada_Bobs
内华达州立银行	内华达州立银行
涅夫斯卡娅光学	Nevskaya_Optica
新亚洲	新亚洲
新宝贝	New_Baby
纽宾凯国际酒店	New_Beacon_国际酒店
New Beacon N+ 国际酒店	New_Beacon_N+_国际酒店
新宾凯尚居酒店	New_Beacon_上居_酒店
新武道跆拳道	New_Budo_Taekwondo
新世纪	新世纪
开元名都大酒店	新世纪大酒店
新频道	新频道
新中国人寿	新中国_生活

名称	ID
全新维度 FCU	new_dimensions_fcu
新时代	New_Era
新弗兰克的	New_Frank的
新希望浸信会教堂	New_Hope_Baptist _Church
新视野 CU	新视野_cu
新视野计算机学习中心	New_Horizons_计算机_学习_中心
新华都超级中心	全新 Hua_du_Supercenter
新生活广场	新生活广场
全新面貌	New_Look
新东方	新东方
新奥尔良消防员 FCU	new_orleans_Firemens_FCU
新大谷酒店	新大谷酒店
New Pharm	New_Pharm
新季市场	新季_市场
新南方电影城	New_South_Movie_City
新星超市	新星超市
新时代证券	《新时报》_证券
新世界	新世界
新世界百货	新世界百货商店
新世界超市	新世界超市

名称	ID
纽约公司	New_York_&_Company
纽约社区银行	纽约社区银行
纽约薯条	New_York_Fries
纽约热狗	New_York_Hot_dog
纽约人寿保险公司	纽约人寿保险公司
纽约披萨	纽约披萨
纽约披萨店	New_York_Pizzeria
纽约长老会医疗保健系统	纽约长老会医疗保健系统
纽约公共图书馆	纽约公共图书馆
新西兰天然	新西兰_天然
新西兰邮政	《新西兰邮报》
NewAlliance 银行	NewAlliance_银行
NewCar	NewCar
Neway	Neway
纽卡斯尔建筑协会	纽卡斯尔建筑协会
纽卡斯尔市	纽卡斯尔_永久
纽怀特诊所	纽怀特诊所
Newk's Eatery	Newk's_eatery
纽波特牙科	纽波特牙科
新闻+	新闻+

名称	ID
Newspower	Newspower
Newu	Newu
纽约切片	Newyork_Sl
下一步 & NextStar	下一页_&_ NextStar
下一步护理	下一页_护理
次日百叶窗	下一天_百叶窗
隔壁商店	Next_door Store
下一个主页	下一页_主页
下一个故事组	下一个故事组
下一辆自行车	下一辆自行车
Nextera	Nextera
Neye	Neye
Ngern Tid Lord	Ngern_Tid_Lor
阮金	Nguyen_Kim
Ngvc	Ngvc
Nha Sach Phuong Nam	Nha_sach_Phuong_Nam
Nha Thuoc An Khang	Nha_Thuoc_An_Khang
Nha Thuoc Taymy Chau	Nha_Thuoc_Tay_My_Chau
Nhon Hoa	Nhon_Hoa
Ni Hon Ryou Ri	Ni_Hon_ryou_Ri

名称	ID
倪世海泰	Ni_Shi_Hai_Tai
Niagaras Choice FCU	Niagaras_Choice_FCU
Nib 国际银行	Nib_国际银行
Nicc Sunleaf	Nicc_Sunleaf
很高兴见到你	很高兴见到你
Nice N Easy	Nice_N_Easy
Nichii	Nichii
Nickelodeon 家庭套房	Nickelodeon_Family_Suites
Nickels 烧烤和酒吧	Nickels_Grill_and_Bar
Nicola Cafés	NicolaCafés
尼古拉斯	尼古拉斯
Nido	Nido
Nids de France	NIDS_de_France
Nidwaldner Kantonalbank	Nidwaldner_KantonalBank
聂卫平围棋学院	Nie_weiping_Go_Academy
Niedermeyer	Niedermeyer
Niederrhein。Sparkasse RheinLippe	Niederrhein。 _Sparkasse_ RheinLippe
Niederösterreichische Landesbank	Niederösterreichische_Landesbank
白天和黑夜	日夜两夜
Night 'n Day	night_'n_day

名称	ID
NightOwl	NightOwl
Nihonkai Shoya	Nihonkai_Shoya
新泻信用金库	新泻_信用金库
新泻县信用合作社	新泻县_信用合作社
Niigata-rokin	Niigata-rokin
日光	日光
Nikkori	Nikkori
尼康	尼康
Niku-no-Mansei	Niku-no-Mansei
Nilgiri 的	Nilgiri 的
Nilkamal	Nilkamal
Nille	Nille
Nilüfer	Nilüfer
Nimseeseng 租赁	Nimseeseng_Leasing
九头鸟连锁餐厅	Nineheaded_Bird_Chain_Restaurant
宁波商业银行	宁波_商业银行
宁波东海银行	宁波_东海银行
宁城包商农村银行	宁城_包商_农村银行
宁城农商银行	宁城_农村_商业银行
忍者盒子	Ninja_Box

名称	ID
Ninomaxx	Ninomaxx
立邦涂料	Nippon_Paint
Nippon-Ichi 鲷鱼烧	Nippon-Ichi_taigaki
Nirala	Nirala
Nirula's	Nirula's
妮莎	妮莎
Nishat 亚麻布	Nishat_Linen
西中国信用金库	西中国_shinkin_Bank
西兵库信用金库	西兵库_shinkin_Bank
西垣岛	西垣岛
锦荒井	nishiki_Warai
仁科食品篮	Nishina_食物篮
西尾信用金库	Nishio_Shinkin_Bank
日产蓝鸟	日产_蓝鸟
日产 TEANA	Nissan_Teana
日产达特桑	日产达特桑
日信信用金库	日信信用银行
Nitori	Nitori
牛太郎	Niu_Tai_Lang
牛科技	纽科技

名称	ID
Nizhegorodskaya Aptechnaya S	Nizhegorodskaya_Aptechnaya_Set'
不用担心	No_Fear
没有多余的装饰	没有多余的装饰
第一药房	No.1_药房
诺亚方舟连锁酒店	Noah's_ark_chain_hotel
Noah's NY Bagels	Noah's_ny_Bagels
诺布希尔食品	Nob_Hill_Foods
诺贝尔书店	Nobel_bookstore
贵族牛排馆	Noble_Family_Steak_House
贵族升级版	Noble_Plus
贵族罗曼的	Noble_Roman's
Nobu 国家银行	Nobu_National_Bank
哥伦比亚之夜	Noches_de_Colombi
Nocibé	Nocibé
Noctalia Flex	Noctalia_Flex
Noel Leeming	Noel_Leeming
Nohvas	Nohvas
Noir	Noir
诺基亚	诺基亚
Nol'Bol'	Nol'_Bol'

名称	ID
Nomad Oil	Nomad_Oil
非儿子	Non_Son
农工商超市	农工商_超市
Noni B	Noni_B
面条盒	面条_盒
面条食堂	面条_食堂
面条站	Noodle_Stat
面条和公司	面条_&_公司
努尔银行	Noor_Bank
Nor 酒店	Nor_Hotels
NorState FCU	NorState_FCU
Norauto	Norauto
Nord	Nord
诺德牙科	北方牙科
Nord Lb	Nord_Lb
Nord Picardie Carburant	Nord_Picardie_Carburant
Nord-Ostsee Sparkasse	Nord-Ostsee_Sparkasse
Nordea	Nordea
Nordea Sampo	Nordea_Sampo
诺德电影	Nordisk_Film

名称	ID
Nordjyske 银行	Nordjyske_Bank
诺德兰兹银行	诺德兰兹银行
Nordoel	Nordoel
Nordotel	Nordotel
诺德西	诺德西
Nordstrom	Nordstrom
Nordthüringer VB	NordThüringer_VB
Norfa	Norfa
诺福克南部	诺福克_南方
NorisBike	NorisBike
Norisko Auto	Norisko_Auto
Norky's	Norky's
诺兰迪亚	诺兰迪亚
Norli Libris	Norli_Libris
Norma De	Norma_De
正常	正常
诺曼	诺曼
Noroc	Noroc
诺雷尔人员配备服务	Norrell_人员配备_服务
Norsk Flid Husfliden	Norsk_Flid_Husfliden

名称	ID
Norsk Hydro	Norsk_Hydro
Norsk Kino	Norsk_Kino
挪威斯堪尼亚	Norsk_Scania
北阿拉巴马州教育工作者大学	North_Alabama_educators_cu
北美货车线路	北美_Van_Lines
北大西洋	北大西洋
北方酒店	北方酒店
北爱荷华州社区 CU	North_IOWA_Community_cu
意大利北部	意大利北部
北喀纳拉州 Gsb Co op Bank	North_kanara_gsb_co_op_Bank
北方公园	北方公园
北极星石油站	北极星石油站
NorthCountry FCU	NorthCountry_FCU
NorthStar CU	NorthStar_CU
东北银行	东北银行
CU 东北部	Northeast_CU
东北证券	东北证券
北方银行	北方银行
北方社区 CU	Northern_Communities_cu
北山 FCU	Northern_Hills_FCU

名称	ID
北印第安纳州 FCU	Northern_Indiana_FCU
北方反思	北方_Reflections
北方摇滚	Northern_Rock
北方工具和设备	Northern_工具和设备
北方信托	Northern_Trust
NorthernIrelandExecutive	NorthernIrelandExecutive
诺斯菲尔德银行	诺斯菲尔德银行
北国地区 FCU	Northland_area_FCU
北线	北线
北区医院	北边医院
Northwell He	诺斯韦尔_健康
西北航空	西北航空公司
西北基督教大学	西北_christian_cu
西北储蓄银行	西北储蓄银行
西北轮胎与服务	西北_轮胎_和_服务
西北互惠人寿保险	西北航空_共同人寿_保险
诺维克银行	Norvik_Bank
诺里奇和彼得伯勒 B.S.	Norwich_&_Peterborough_B.S.
Noryoku Kaihatsu 中心	Noryoku_Kaihatsu_Center
鼻子	鼻子

名称	ID
我们的 Caixa	Nossa_Caixa
诺斯托	诺斯托
不只是披萨	不只是披萨
Notabene	Notabene
Nothing Bundt	Nothing_Bund_Cak
只不过是面条	除了面条之外什么都没有
能登共荣信用金库	Noto_Kyoei_Shinkin_Bank
Nouvelles Fontières	Nouvelles_Fontières
Nova	nova_cu
Nova Cambios	Nova_Cambios
Nova 住房贷款	新房贷款
马里博尔新信贷银行	Nova_Creditna_Banka_Maribor
新卢布尔雅那银行	Nova_Lubjanska_Banka
诺瓦药房	Nova_药房
Nova Poshta	Nova_Poshta
Nova Rede Drogaria	Nova_rede_Drogaria
NovaCare 康复	NovaCare_康复
Novabank	Novabank
Novaya Zarya	Novaya_Zarya
Novaya apteka	novaya_apteka

名称	ID
诺夫哥罗德石油产品	诺夫哥罗德石油产品
Noviy Knizhniy	Noviy_Knizhniy
诺夫里昂酒店	Novlion_Hotel
诺沃	诺沃
Novo Shoes	新款鞋子
Novosibirskaya ptitsefabri	Novosibirskaya_ptitseFabrika
新西伯利亚市政银行	Novosibirskiy_munitsipal'niy_Bank
新西伯利亚石油产品	新西伯利亚石油产品
Novum	Novum
诺弗斯	诺弗斯
Novus 汽车玻璃	Novus_Auto_Gl
Novyi	Novyi
诺瓦咖啡	Nowwa_Coff
NrGize 生活方式咖啡厅	NrGize_生活方式_咖啡厅
新地铁	Nu_Metro
NuMark CU	NuMark_CU
NuVista FCU	NuVista_FCU
Nuance 通讯	细微差别_通讯
努比亚	努比亚
Nueva Linea	Nueva_Linea

名称	ID
Nuevo Almacén Gastrob	Nuevo_Almacen_Gastrobar
新圣达菲银行	nuevo_banco_de_santa_fe
新查科银行	nuevo_banco_del_Chaco
新地平线旅游	Nuevos_Horizontes_Viajes
Nuit D'Hotel	Nuit_d'Hotel
沼津信用金库	沼津信用金库
头号鞋	一号鞋子
Numero Uno	Numero_Uno
诺庭客栈	诺庭客栈
Nuova Banca Etruria	Nuova_Banca_Etruria
NutriSystem 减肥中心	NutriSystem_Weight_Loss_Center
坚果和香料	Nuts_n_Spices
Nuvision FCU	nuvision_fcu
Nyugat Takarék	Nyugat_Takarék
努涅斯和纳瓦罗酒店	NUÑEZ_I_NAVARRO_HOTELS
Nábytek Jamall	Nábytek_Jamall
Nádix Change	Nadix_Change
附近的商店	附近的商店
O Bee CU	o_bee_cu
O Boticário	O_Boticário

名称	ID
O C S	O_C_S
O MEDHOR BOLO DE CHOCOLO DE MUNDO	O_MELHOR_BOLO_DE_CHOCOLATE_DO_MUNDO
哦 Mamma Mia	O_Mamma_Mia
O Mundo do Café	o_mundo_do_Café
O！商店	O！ _SHOP
O'Shipka	O'_Shipka
O'Briens	O'Briens
O'CITY 酒店	'CITY_HOTEL
O'Chelle	O'Chelle
O'coffee	O'Coffee_
O'HARA	O'HARA
O'Hagan's	O'Hagan's
O'Learys	O'Learys
O'Reilly 汽车零部件	O'Reilly_Auto_Parts
O'Stin	O'Stin
'Sushi	'Sushi
O'TACOS	O'TACOS
O. El Tayeby Pharmacy	O._El_Tayeby_Pharmacy
氧气	氧气
腰带	腰带

名称	ID
OBK	OBK
OBLEAS Jose A.	obleas_jose_A。
OBRAMAT	OBRAMAT
OBRAZOVANIE	OBRAZOVANIE
OBUV.COM	OBUV.COM
OCB	OCB
华侨银行	华侨银行
华侨银行 NISP	OCBC_NISP
西方酒店及度假村	西方_酒店_度假村
海洋银行	海洋银行
OCHNIK	OCHNIK
西方的	西方的
华侨城酒店	OCT_HOTEL
OCTA+	OCTA+
八角形	八角形
OCharley的	OCharley的
ODDP 厨房	oddp_Kitchen
不在家里	不在家里
OFFCROSS	OFFCROSS
OFFERTISS	OFFERTISS

名称	ID
办公鞋	办公鞋
OGERO	OGERO
酒店	酒店
OHYA 汽车旅馆	Ohya_Motel
油！	油！
大石快递	大石快递
OISHI 格兰德	OISHI_GRAND
大石拉面	OISHI_RAMEN
OJAY	OJAY
正常	正常
OK 随时市场	好_anytime_Market
好快递	OK_Expr
不错的食物	OK_Foods
OK 家具	OK_家具
OK Grocer	OK_grocer
不错的市场	好的 Market
OK Mart	OK_mart
OK MiniMark	好_ MiniMark
OK Pastelería	Ok_Pasteleria
OK Tire	OK_tire

名称	ID
OK 值	OK_Value
OKASHI 乐园	OKASHI_LAND
好的	好的
OKKO 酒店	OKKO_Hotels
OKLAO	OKLAO
OKNOPLAST	OKNOPLAST
OKQ8	OKQ8
OKTA	OKTA
OKTAN	OKTAN
OKay	OKay
OKa伊迪	OKa伊迪
OLCO	OLCO
老张记	OLD_CHANG_KEE
老芝加哥	老芝加哥
OLIMP	OLIMP
奥林匹卡	奥林匹卡
奥利弗格兰特	OLIVER_GRANT
OLIVEYOUNG	OLIVEYOUNG
OLYMP 服装店	奥林匹克服装店
奥林匹亚体育	奥林匹亚_体育

名称	ID
阿曼阿拉伯银行	阿曼_阿拉伯银行
阿曼开发银行	阿曼开发银行
欧米茄	欧米茄
OMNI	OMNI
OMNISHOP	OMNISHOP
天哪	天哪
OMV	OMV
OMV Istrabenz	omv_istrabenz
ONAHOTELS	ONAHOTELS
一次	一次
ONE ZO	ONE_ZO
鬼火锅	ONI_HOT_POT
ONIX	ONIX
ONLY	ONLY
小野	小野
ONOFF	ONOFF
ONstyle	ONstyle
OOTCL	OOTCL
OOTOYA	OOTOYA
OPA	OPA

名称	ID
哎呀！希腊的	哎呀！ _OF_GREECE
欧宝	欧宝
25 小时开放！	OPEN_25_HS_！
打开干净	OPEN_CLEAN
OPG	OPG
相反的	相反的
OPSM	OPSM
OPTI-VUE	OPTI-VUE
OPTICA 2000	OPTICA_2000
OPTICALIA	OPTICALIA
OPTISSIMO	OPTISSIMO
选项	选项
租车选项	OPÇÃO_RENT_A_CAR
ORA	ORA
乐团	乐团
俄勒冈	俄勒冈
OREX	OREX
ORGA	ORGA
有机	有机
东方交易所	东方交易所

名称	ID
东方证券	东方证券
ORIHICA	ORIHICA
ORKAN	ORKAN
奥伦	奥伦
ORSAN	ORSAN
OS 药店	os_Drug_Store
奥斯卡	奥斯卡
OSGH 电影院	OSGH_CINEMAS
OSIM	OSIM
OST	OST
OTIFARMA	OTIFARMA
OTKRYTIE	OTKRYTIE
OTP 银行	OTP_Bank
或者	或者
OTTO 烧烤	OTTO_GRILL
OTTOWASH	OTTOWASH
你的杯子	OUR_CU
我们的投资	我们的投资
户外仓库	户外仓库
装备	装备

名称	ID
奥特莱特-J	奥特莱特-J
OVER	旅行专家
太惊慌失措了	太惊慌失措了
OVS 行业	OVS_行业
OWNDAYS	OWNDAYS
OXALIS	OXALIS
牛津公司	牛津公司
氧化物	氧化物
OXXO 服装店	oxxo_Clothing_store
OXXO 气体	OXXO_GAS
绿野仙踪能源	oz_Energia
臭氧健身	臭氧健身
橡木	橡木
橡树酒店	Oak_Hotel
橡树旅馆	Oak_Tree_Inn
奥克利	奥克利
橡树	橡树
奥克伍德	奥克伍德
奥克伍德家园	奥克伍德_Homes
Oando	Oando

名称	ID
绿洲	绿洲
绿洲咖啡厅	Oasis_Café
绿洲汽车旅馆	Oasis_Motel
奥伯班克	奥伯班克
欧贝罗伊	欧贝罗伊
Oberweis	Oberweis
带广信用金库	Obihiro_Shinkin_Bank
Obonk Steak & Ribs	Obonk_Steak_&_Ribs
Obshtinska Banka	Obshtinska_Banka
Obwaldner Kantonalbank	Obwaldner_KantonalBank
Oca	Oca
Oca 酒店	Oca_Hoteles
Ocafa	Ocafa
场合	场合
西方酒店	西方酒店
海运	海运
海洋篮子	海洋篮子
Ocean Edu	Ocean_Edu
海洋金融 FCU	Ocean_Financial_FCU
海洋人寿保险	海洋人寿保险

名称	ID
大洋州 Job Lot	Ocean_State_Job_Lot
OceanFirst 银行	OceanFirst_银行
Oceancont	Oceancont
大洋洲酒店	大洋洲_酒店
大洋银行	大洋银行
Ochaya	Ochaya
Ochkarik	Ochkarik
奥克斯纳诊所基金会	Ochsner_Clinic_基金会
Ocine	Ocine
Octave	Octave
十月	十月
小田急 OX	小田急_ox
Odalys	Odalys
Oddbins	Oddbins
奥迪安	奥迪安
奥里萨邦格拉米亚银行	Odisha_Gramya_Bank
奥里萨邦州立合作银行	奥里萨邦_州立合作社_银行
奥东托公司	Odonto_Company
Odoya	Odoya
Oekotel	Oekotel

名称	ID
Oeral	Oeral
Oeschle	Oeschle
非百老汇	Off_Broad
离开 Ya Tree	Off_Ya_Tree
Offcn	Offcn
Office	Office
Office 1 超市	办公室_1_Superstore
办公仓库	办公仓库
办公室奥特莱斯	办公室_奥特莱斯
办公用品仓库	办公_产品仓库
办公套件升级版	办公套房_Plus
办公室仓库	办公室_仓库
OfficeMate	OfficeMate
OfficeMax	OfficeMax
OfficeTeam	OfficeTeam
军官俱乐部	军官俱乐部
Officewor	Officewor
爱尔兰官方酒吧	Official_Irish_Pub
Oficentro	Oficentro
Ofiprecios	Ofiprecios

名称	ID
Ofoterrt	Ofoterrt
Ogaki-Seino 信用金库	Ogaki-seino_shinkin_Bank
Ogalo	Ogalo
小川	小川
Oggi	Oggi
Oggi Sorvetes	Oggi_Sorvetes
Ogutler	Ogutler
哦哈哈	Oh_la_la
俄亥俄州汽车贷款服务	公司
俄亥俄州 HealthCare FCU	俄亥俄州_ _FCU HealthCare
俄亥俄州储蓄银行	俄亥俄州储蓄银行
俄亥俄大学 CU	Ohio_university_CU
哦对了	哦对了
Oi	Oi
Oi2 Centros Auditivos	Oi2_Centros_Auditivos
石油	石油
石油管家	Oil_Butler
Henry's 石油罐	Oil_Can_Henry's
换油机	换油机
Oil4Wales	Oil4Wales

名称	ID
Oilcorp	Oilcorp
Oilibya	Oilibya
大分未来信用金库	大分县_Mirai_Shinkin_Bank
大分信用金库	大分县信用金库
大分县信用合作社	Oita-ken_credit_合作社
冈三证券	冈山证券
冈山信用金库	冈山_信用金库
冈崎信用金库	冈崎信用金库
Oke Shop	Oke_Shop
Okko	Okko
奥克拉茶	奥克拉茶
俄克拉荷马州中部	OklaHoma_central_cu
俄克拉荷马教育工作者	OklaHoma_educators_cu
俄克拉荷马州立大学	OklaHomas_cu
Okna Rosta	Okna_Rosta
御好烧本铺	Okonomiyaki_Hompo
大仓酒店及度假村	大仓_酒店及度假村
Ol Gud	Ol_Gud
Ol'Ga	Ol'Ga
奥拉酒店	Ola_酒店

名称	ID
古老的乡村自助餐	古老乡村自助餐
老自治领货运专线	Old_Dominion_Freight_Line
古老的英式旅馆	老式英语旅馆
老式炸鱼薯条	Old_fashioned_fish _and_Chips
老神	Old_God
老卡其色	Old_Khaki
老互助银行	Old_Mutual_Bank
老国家银行	老国民银行
老海军	Old_Navy
老第二国民银行	Old_Second_National_
老意大利面工厂	Old_Spaghetti_Factory
老庙金	Old_Temple_Gold
老城白咖啡	老城区白咖啡
老叔叔	老叔叔
老城寺庙小吃	Oldtown_Temple_Snac
奥尔登堡州立银行	Oldenburgische_Landesbank
奥兹莫比尔	奥兹莫比尔
Ole'	Ole'
Olerex	Olerex
奥尔加的厨房	Olga's Kitchen

名称	ID
橄榄和油	Olive_&_Oil
橄榄园意大利餐厅	Olive_Garden_意大利餐厅
Olivenlunden 1830	Olivenlunden_1830
奥利弗的真实食物	Oliver's_real_Food
奥利弗的超级三明治	Oliver's 超级三明治
奥利维里	奥利维里
奥利维亚皮肤和头发诊所	Olivia_Skin_&_Hair_Clinic
奥利维尔·达奇金	Olivier_Dachkin
奥利维尔·斯特雷利	Olivier_Strelli
Oljeshekkerna John	Oljeshekerna_Johnss
Ollie 的讨价还价	Ollie's Bargain
Olsten 人员配备服务	Olsten_人员配备_服务
奥尔瓦信使	Olva_Courier
Olymp	Olymp
奥林匹亚信用合作社	奥林匹亚信用合作社
Ola	Ola
奥利斯	奥利斯
Oma	Oma
Oma 家居装修和五金	Oma_家居_装修_&_硬件
Oma Saystöpankki	Oma_Saästöpankki

名称	ID
奥马哈警察 FCU	omaha_Police_FCU
奥马哈牛排	Omaha_Steaks
阿曼国际银行	阿曼国际银行
阿曼石油	Oman_Oil
Omaya	Omaya
天哪 Carr	Omc_Carr
青梅信用金库	Ome_Shinkin_Bank
欧米茄 FCU	Omega_fcu
Omker	Omker
奥姆尼酒店	全方位酒店
Omniasig	Omniasig
Omnjoi 国际影院	Omnijoi_国际_影院
Omnip	Omnip
omskiy Bekon	Omskiy_Bekon
Omuraisutei	Omuraisutei
Omusubi Gonbei	Omusubi_Gonbei
关于旅行社	on_agencias_de_Viajes
在诊所里	在诊所
在 Cue Expres	On_Cue_Expr
On Tap	On_Tap

名称	ID
在现场	On_The_Spot
准时休息和食物	On_Time_Break_&_Food
在边境	On_the_border
在逃跑	On_the_run
OnCue	OnCue
OnPath 联邦 CU	OnPath_federal_cu
Ona Saez	Ona_Saez
曾几何时	Once_upon_a_Child
Oncquest	Oncquest
Oncu	Oncu
Ondot 快递和快递货物	Ondot_Courier_&_Express_Cargo
一日画	One_Day_Paint
一百年 Yufu	一百年_Yufu
一加一超级市场	一加一超级市场
One Price 服装店	一价服装专卖店
一次烘烤	One_Roas
单一来源 FCU	one_source_FCU
一步到位	One_Step
一站式	One_Stop
一茶匙	一茶匙

名称	ID
同一个愿景 FCU	one_vision_FCU
One's Rent-A-Car	one's_租一辆车
OneMain 财务	OneMain_金融
OneWest 银行	OneWest_银行
鬼冢虎	Onitsuka_Tiger
仅限教育	仅限教育
安大略海岸 FCU	Ontario_shores_FCU
Ontrack 数据恢复	在轨数据恢复
奥努尔	奥努尔
Onur 集团	Onur_Group
Onurex	Onurex
Onyasai	Onyasai
哎呀！	哎呀！
Op Is 在 Voordeelshop	op_is_op_voordeelShop
OpConnect	OpConnect
Opan	Opan
OpelRent	OpelRent
Open 25	Open_25
开放诊所	开放诊所
开放式储藏室	开放式储藏室

名称	ID
打开你的门	打开你的门
为孩子开放	Open_for _kid
Open32	Open32
OpenCor	OpenCor
OpenWorks	OpenWorks
开放银行	开放银行
Opencel	Opencel
Opet	Opet
Opmar Optik	Opmar_Optik
Opolska SKO	opolska_skok
波尔图	波尔图
Oportun	Oportun
OPPO	OPPO
机会银行	Opportunity_Banka
Optan	Optan
光学 2000	Optic_2000
Optica Halperin	Optica_Halperin
光学中心	Optical_Center
眼镜折扣	Optical_Discount
光学快车	光学快车

名称	ID
Opticana	Opticana
Opticália	Opticália
选项 1	Opti_1
Optika 眼动力	Optika_Eyekraf
光学 SILOE	Optika_siloe
Optika Schastliviy Vzglyad	Optika_Schastliviy_Vzglyad
Optiker Bode	Optiker_Bode
Optikos 沙米	Optikos_Shami
乐观主义者	乐观主义者
Optivisão	Optivisão
Optus	Optus
奥拉加索巴	奥拉加索巴
OralKlass	OralKlass
橙色便利	橙色_便利
橙色酒店	橙色酒店
橙色网吧	橙色网吧
橙色朱利叶斯	橙色_朱利叶斯
橙茶	橙茶
橙色突尼斯衣	橙色_突尼斯
橙色理论 Fitness	橙色理论_健身

名称	ID
Oranzh	Oranzh
Oranzheviy Slon	Oranzheviy_Slon
宝珠通讯	轨道通讯
Orbis	Orbis
Orby	Orby
Orcanta	Orcanta
订单	订单
鄂尔多斯银行	鄂尔多斯银行
奥雷亚酒店	Orea_酒店
俄勒冈社区 CU	Oregon_Community_cu
奥伦莱克	奥伦莱克
有机农场 Zdrowia	Organic_Farma_Zdrowia
有机 Shizen Shokukuhin Kodawariya	Organic_Shizen_Shokuhin_Kodawariya
东方	东方
东方山水酒店	东方日光酒店
东方商业银行	东方商业银行
东方儒家酒店	东方_儒家_酒店
东方饺子王	Oriental_Dumpling_King
东方儿童艺术	Oriental_kid's_art
东方织工	Oriental_Weavers

名称	ID
东方之光酒店	Orientalshine_Hotel
Origato	Origato
原创经典	原创_经典
原版 Joe's	Original_Joe's
原创煎饼屋	Original_pancake_House
Origins	Origins
Origus	Origus
Orkin 害虫和白蚁防治	Orkin_Pest_&_Termite_Control
奥兰多大学	orlando_cu
Orlik	Orlik
奥尔洛斯基	奥尔洛斯基
奥尔良管理	奥尔良管理
Ormatek	Ormatek
奥罗米亚国际银行	奥罗米亚国际银行
Oroton	Oroton
Orsay	Orsay
Ortho 床垫	矫形床垫
美国正畸中心	美国正畸中心
Orvis	Orvis
大羚羊	大羚羊

名称	ID
大阪市信用金库	大阪市信用金库
大阪厚生信用金库	大阪_Kosei_Shinkin_Bank
大阪王将	大阪_王将
大阪信用金库	大阪_信用金库
大阪商行信用金库	大阪商行信用金库
奥萨塞特	奥萨塞特
奥斯卡奥斯卡沙龙	奥斯卡_奥斯卡_沙龙
奥斯科药房	奥斯科_药房
Oses Cigköfte	Oses_Cigköfte
OshKosh 天哪	OshKosh_b'gosh
Oshchadbank	Oshchadbank
Oshi Oshi	Oshi_Oshi
奥斯克伦	奥斯克伦
奥斯克罗巴	奥斯克罗巴
奥斯陆 Kinematografer	Oslo_Kinematografer
Osmanli Tulumbacisi	Osmanli_Tulumbacisi
Ossira	Ossira
Ostfriesische Volksbank	Ostfriesische_Volksbank
Ostharzer Volksbank	Ostharzer_Volksbank
OstseeSparkasse 罗斯托克	OstseeSparkasse_罗斯托克

名称	ID
德累斯顿 Ostsáchsische Sparkasse	Ostsáchsische_Sparkasse_Dresden
Osuuspankki	Osuuspankki
otantik Kumpir	Otantik_Kumpir
大手店	大手店
奥赛姆	奥赛姆
奥的斯电梯	奥的斯电梯
Oto Pratik	Oto_Pratik
Otog Front Banner 农村商业银行	Otog_Front_Banner_Rural_Comm_Bank
Otp banka	OTP_Banka
奥托	奥托
奥托的	奥托的
奥托's Ag	Otto's_ag
奥托餐厅	Otto's 餐厅
奥托林	奥托林
Ouachita Valley FCU	Ouachita_valley_fcu
我们的时间	我们的工作时间
出乎意料	Out_of_the_Blue
内陆牛排馆	内陆_Steakhouse
服装零售商	服装零售商
出口	出口

名称	ID
Outlet Viajes	Outlet_Viajes
奥特莱斯	奥特莱斯
Ouya	Ouya
欧亚达	欧亚达
烤箱咖啡	烤箱_咖啡
OvenFresh	OvenFresh
Ovenstory 披萨	Ovenstory_Pizz
超过最大值	超出最大值
超额租金	超额租金
Overeaters 匿名	Overaters_Anon
架空门	架空门
陆上	陆上
陆上酒	陆上酒类
海外巨龙	海外_龙
Evo Estrelado	Ovo_Estrelado
欧文	欧文
欧文的	欧文的
欧文的燃料中心	Owen's Fuel_Center
欧文药房	Owen's Pharmacy
猫头鹰照相馆	猫头鹰摄影工作室

名称	ID
乐施会	乐施会
牛津	牛津
牛津书店	牛津书店
牛津 FCU	牛津_FCU
牛津学习	牛津大学_学习
Oxxo	Oxxo
奥西布尔	奥西布尔
Oylmpus	Oylmpus
Oyo 酒店	Oyo_Hotel
Ozamu	Ozamu
Ozerki	Ozerki
Ozmosis	Ozmosis
奥苏特	奥苏特
Ozturk	Ozturk
P 和 M	p_and_m
P&K 设备	P&K_设备
P's-first	P's-first
P'tit Dej Hotel	p'tit_dej_Hotel
P.S. 时尚	P._S._Fashion
P.C. LEE CA 牛肉面条之王	P.C._LEE_CA_BEEF_NOODLE_KING

名称	ID
P.C. Richard & Son	P.C._Richard_&_Son
P.style 沙龙	p.style_Salon
P1	P1
pac'nSave	pac'nSave
太平洋	太平洋
太平洋百货商店	太平洋百货商店
太平洋证券	太平洋证券
PACL 印度	Pacl_India
PACO MARTINEZ	PACO_MARTINEZ
PAGRO DISKONT	PAGRO_DISKONT
PAIDEL	PAIDEL
pak'nSave	pak'nSave
pak'nSave Fuel	pak'nsave_Fuel
PAKO LORENTE	PAKO_LORENTE
皇家宫殿	皇家宫殿
帕拉尔熊	PALAR_BEAR
钯金酒店及度假村	Palladium _酒店_度假村
PALMERS	PALMERS
PAN-PEK	PAN-PEK
平底锅。MENESA	平底锅。 _MENESA

名称	ID
胰腺合作银行	胰腺_合作社_银行
PANELLINIA 银行	PANELLINIA_BANK
PANISHOP	PANISHOP
全景	全景
PAPELERIA DANY	PAPELERIA_DANY
PAPELERIAS LA ESCOLAR	PAPELERIAS_LA_ESCOLAR
PAPPADEAUX 海鲜厨房	PAPPADEAUX_海鲜_厨房
纸莎草	纸莎草
MELIA 的 PARADISUS	PARADISUS_BY_MELIA
PARANOiA	PARANOiA
PARFOIS	PARFOIS
巴黎法式长棍面包	PARIS_BAGUETTE
巴黎三木	PARIS_MIKI
巴黎光学	PARIS_OPTIQUE
公园精英	PARK_ELITE
公园服务中心 HÜFNER	PARK_SERVICE_HÜFNER
PARKSON	PARKSON
PARKn购物	PARKn购物
PARMIGIANI	PARMIGIANI
标准杆	标准杆

名称	ID
PARS 市场	Pars_Mark
帕森斯音乐	PARSONS_MUSI
PARTIOAITTA	PARTIOAITTA
伴侣年龄 RTV	PARTNER_AGD_RTV
PASA DONER	PASA_DONER
PASQUALE	PASQUALE
过去的时代	过去的时间
PASTAMANIA	PASTAMANIA
PASTELERIAS ROLDAN	PASTELERIAS_ROLDAN
百达翡丽	PATEK_PHILIPPE
帕特里斯·布雷尔	PATRICE_BREAL
帕特里齐亚·佩佩	PATRIZIA_PEPE
守护神	守护神
保罗	保罗
保罗和鲨鱼	PAUL_&_SHARK
PAUL 餐厅和面包店	Paul_restaurant_&_baker
PAULA CAHEN D'ANVERS	PAULA_CAHEN_D'ANVERS
PAWO	PAWO
PBDAC	PBDAC
PBKIDS	PBKIDS

名称	ID
PBW	PBW
PBZ	PBZ
PBteen	PBteen
PC CITY	PC_CITY
PC Coste	PC_Coste
PC 珠宝商	PC_Jeweller
电脑世界	PC_World
PCBox	PCBox
PCCW	PCCW
PCD 商店	PCD_STORES
PCF Sparkletots	pcf_sparkletots
PDI	PDI
PDQ	PDQ
PDV	PDV
PECSA	PECSA
PEDRA DURA	PEDRA_DURA
PEGAS Touristik	Pegas_touristik
PEGAVISION	PEGAVISION
PEMO	PEMO
PENA HOUSE	PENA_HOUS

名称	ID
PENFED 信用合作社	penfed_credit_Union
便士	便士
一分钱	一分钱
贝尔塔酒店	贝尔塔酒店
PENTIK	PENTIK
PEP	PEP
PEP 细胞	pep_cell
PEP 主页	pep_Home
PEPCO	PEPCO
完美清洁	完美干净
PERIPL	PERIPL
表象人格	表象人格
PETCO	PETCO
彼得·英格兰	彼得_英格兰
PETIMEX	PETIMEX
PETIT BATEAU	PETIT_BATEAU
PETIT PALACE 酒店	PETIT_PALACE_HOTELES
佩特拉斯	佩特拉斯
加拿大石油公司	加拿大石油公司
PETROFER	PETROFER

名称	ID
PETROIBÉRICA	PETROIBÉRICA
汽油	汽油
石油公司	汽油公司
PETROMIRALLES	PETROMIRALLES
PETRONIEVES	PETRONIEVES
秘鲁石油公司	秘鲁石油公司
石油大奖赛	石油大奖赛
PETROX	PETROX
PETSMART	PETSMART
PETstock	PETstock
标致	标致
标致租金	标致租车
PEUQUE	PEUQUE
PEZZO	PEZZO
PF Chang 的中国小酒馆	pf_Chang's_China_Bistro
PFCU	PFCU
PG 银行	PG_Bank
PG 玻璃	PG_Glass
PGG Wrightson	pgg_wrightson
PGO 滑板车	PGO_SCOOTERS

名称	ID
医药健康	PHARMA_SANTÉ
药房家族	药房家族
PHARMACY1	PHARMACY1
PHCN	PHCN
PHH 住房贷款	phh_Home_Loans
菲利普斯·纳卡斯	PHILIPPOS_NAKA
腓尼基	腓尼基
照片大厅	照片大厅
PHS	PHS
PIAGET	PIAGET
PIAO 如家旅馆	PIAO_HOME_INN
皮卡迪利	皮卡迪利
腌鱼	PICKED_FISH
PICTET	PICTET
PIEKARNIA POD TELEGRA	PIEKARNIA_POD_TELEGRAFEM
88 号码头	PIER_88
PIGPA	PIGPA
PIXIANG	PIXIANG
PIKAPOLONICA	PIKAPOLONICA
PILATOS	PILATOS

名称	ID
品良成少济功	PIN_LIANG_CHENG_SHAO_JI_GONG
PINKO	PINKO
PIONIER！	PIONIER！
PIP 印刷和营销服务	PIP_印刷_和_营销_服务
海盗	海盗
PIRMA	PIRMA
旋转木马	旋转木马
PISERRO	PISERRO
PITTARELLO	PITTARELLO
匹兹堡	匹兹堡
PIWIARNIA WARKA	PIWIARNIA_WARKA
PIXmania.com	PIXmania.com
披萨 1969 美食家	PIZZA_1969_GOURMET
披萨迷	PIZZA_FAN
意大利披萨	意大利披萨
PIZZA MOBIL	PIZZA_MÖVIL
完美的披萨	披萨_完美
PJ 的咖啡	pj's_coffe
PKO Bank Polski	pko_bank_Polski
PKU Care Health 管理中心	PKU_care_Health_Management_

名称	ID
PKnet.	PKnet.
Electro Plan	Planeo_electro
PLANEO 快点时间	Planeo_quick_Tim
星球	星球
星球时尚	星球_时尚
星球女孩	PLANET_GIRLS
奥布卡星球	PLANET_OBUCA
星球寿司	PLANET_SUSHI
星球体育	PLANETA_SPORT
厂	厂
植物	植物
铂金汉爵酒店	PLATINUM_HANJUE_HOTEL
播放	播放
PLAYASOL	PLAYASOL
广场	广场
宜人的宫殿	PLEASANT_PALAC
PLENOI	PLENOI
计划	计划
PLODINE	PLODINE
PLURICOOP	PLURICOOP

名称	ID
再加上银行	PLUS_BANK
PM	PM
PNA	PNA
PNB	PNB
PNC 银行	PNC_bank
PNC 抵押贷款	PNC_抵押贷款
PNJ	PNJ
PO FAT	PO_FAT
PO 市场	PO Market
POCO Einrichtungsmarkte	poco_einrichtungsMarkte
PODRAVSKA BANKA	PODRAVSKA_BANKA
豆荚	豆荚
POINT.P TP	POINT.P_TP
积分	积分
POL'BEYKERI	POL'_BEYKERI
极地	极地
POLAR 泡芙和蛋糕	Polar_Puffs_&_cakes
POLARN O. PYRET	POLARN_O._PYRET
POLISHOP	POLISHOP
POLISON	POLISON

名称	ID
POLLO CAMPERO	POLLO_CAMPERO
POLLO MATON	POLLO_MATON
POLOmarket	POLOmarket
POLSKIE SKLEPY DZIECIECE	POLSKIE_SKLEPY_DZIECIECE
POLY	POLY
POLYANA	POLYANA
POLZELA	POLZELA
波莫纳	波莫纳
POMPO	POMPO
PONEY	PONEY
庞蒂亚克	庞蒂亚克
小马出租	PONY_RENT
POP Pankki	pop_pankki
POPPY 华夫饼	poppy_Waffle
POP	POP
热门私人银行	大众银行_PROVADA
瓷器	瓷器
PORLAND	PORLAND
保时捷	保时捷
波特兰	波特兰

名称	ID
端口	端口
POSB	POSB
邮政附件 +	POSTAL_ANNEX+
邮政网	邮政网
陶器谷仓	POTTERY_BARN
POWER	POWER
POWER 消费电子产品商店	消费类电子产品商店
强力蜡	POWER_WAX
POYA HOME	POYA_HOME
POZEGNANIE Z AFRYKA	POZEGNANIE_Z_AFRYKA
PPCBL	PPCBL
PPG 工业公司	PPG_Industries
PPK BJELOVAR	PPK_BJELOVAR
PQ 服装	pq_服装
普拉达	普拉达
普拉萨克	普拉萨克
PRECIS PETITE	PRECIS_PETITE
预先设定的	预先设定的
PREGO GOOM	PREGO_GOURM
PREMAP	PREMAP

名称	ID
PREMIATTO	PREMIATTO
高级组	高级组
头等舱	PREMIÈRE_CLASSE
PRESTIA	PRESTIA
普雷斯顿酒类专卖店	Prestons_Liquor_Stores
PREVIDENTE	PREVIDENTE
价格和骄傲	PRICE_'N_PRIDE
英国骄傲酒店	BRITAIN_PRIDEAIN_HOTELS
PRIMA DELI	PRIMA_DELI
PRIMA 家居专卖店	Prima_home_Specialty_store
PRIMA MODA	PRIMA_MODA
PRIMAX	PRIMAX
PRIMI	PRIMI
王子餐厅	王子餐厅
Tam.Tam 公主	princesse_tam
PRINTEMPS	PRINTEMPS
PRIVATBANK DEGROOF	PRIVATBANK_DEGROOF
贝尔格莱德私人银行	PRIVREDNA_BANKA_AD_BEOGRAD
PROCHAZKA maso-uzeniny	prochazka_maso-uzeniny
PROCheck	PROCheck

名称	ID
教授	教授
个人资料更多	个人资料_PLUS
PROMART	PROMART
PRONAR	PRONAR
PRONOVIAS	PRONOVIAS
PRONTO 咖啡厅和酒吧	pronto_cafe_&_bar
PROTEA	PROTEA
质子	质子
质子库	质子库
PRVA BANKA CG	PRVA_BANKA_CG
PRYMUS-AGD	PRYMUS-AGD
PRÜNE	PRÜNE
PS	PS
PS 食品超市	PS_Food_mart
PSC 国际	PSC_国际
私营部门司银行	PSD_Bank
PSE 信用合作社公司	PSE_credit_Union_Inc
PSK 银行	PSK_bank
PSO	PSO
PSS Spolem	pss_spolem

名称	ID
PT	PT
PT Power	pt_Power
PT2000	PT2000
PTK	PTK
PTT 银行	PTT_Bank
PTT KARGO	PTT_KARGO
PTT 邮局	ptt_post_Office
PTT matik	ptt_matik
PTT 站	PTT_Station
浦东商场	PU_DONG_EMPORIUM
大众银行（香港）	PUBLIC_BANK_ (香港_KONG)
PUC	PUC
PUENG NGEE CHIAN	PUENG_NGEE_CHIANG
PUFF & PIE	PUFF_&_PIE
PULS & TRäNING	PULS_&_TRANING
彪马能源	PUMA_Energy
PUNT ROMA	PUNT_ROMA
纯	纯
PURIFICACION GARCIA	PURIFICACION_GARCIA
莆田	莆田

名称	ID
PUILO	PUILO
普易光学	普易光学
光伏油	PV_oil
PVR 影院	PVR_Cinema
PVS 芒果面包师	pvs_mango_Bakers
PVcom银行	PVcom银行
PYXIS	PYXIS
巴勃罗	巴勃罗
PacSun	PacSun
Pace	Pace
太平洋喀斯喀特 FCU	Pacific_cascade_FCU
太平洋咖啡公司	太平洋咖啡公司
太平洋健身	太平洋健身
太平洋卓越银行	Pacific_Premier_Bank
太平洋骄傲	太平洋骄傲
太平洋地区销售	Pacific_Sales
太平洋运输 FCU	太平洋_交通_FCU
太平洋西部国民银行	太平洋西部国民银行
帕西尼	帕西尼
打包和麻袋	Pack_&_Sack

名称	ID
帕科酒店	Paco_Hotel
Pactiv	Pactiv
帕吉特商业服务	Padgett_商业_服务
帕迪尼	帕迪尼
Padisah Hali	Padisah_Hali
帕加尼	帕加尼
第一页	Page_One
Pagenet	Pagenet
宝塔	宝塔
Pague Menos	Pague_Menos
帕哈拉快车	帕哈拉快车
拜县酒店	Pai_Hotel
拜县国际	Pai_国际
Paibai.Yun 酒店	Paibai.Yun_Hotel
Pain d'Or	Pain_d'Or
Pain a la Ligne	pain_à_la_LIGNE
Painewebber	Painewebber
用曲折绘画	用_A_Twist 绘画
油漆点	油漆点
朴英雄	Pak_Hero

名称	ID
Pak 邮件地址	Pak_Mail
Pak 卡塔尔 Takaful 保险	Pak_Qatar_Takaful_Insurance
Pak-a-Sak	Pak-a-Sak
帕克彭	帕克彭
皇宫电影院	Palace_Cinema
皇宫电影院	宫殿_电影院
皇宫旅馆	Palace_Inn
Palafox	Palafox
Palais des thés	Palais_des_thés
Palaispa	Palaispa
Paleo 咖啡厅	Paleo_Cafe
Palettas	Palettas
巴利语	巴利语
帕利亚酒店	Palia_酒店
Palisades FCU	palisades_FCU
钯	钯
棕榈滩棕褐色	Palm_Beach_Tan
棕榈泉油	Palm_Springs_Oil
帕尔默斯花园中心	帕默斯花园中心
Palmetta	Palmetta

名称	ID
Palmetto Bank	Palmetto_Bank
棕榈小酒馆	棕榈小酒馆
Pam	Pam
Pam Golding 地产	Pam_Golding_Proper
Pambao	Pambao
棉花堡	棉花堡
Pan Chemodan	Pan_Chemodan
Pan Cig Kofte	Pan_Cig_Kofte
Pan Pa'Ya！	Pan_Pa'_Ya！
泛太平洋	泛太平洋
帕纳戈披萨	帕纳戈_披萨
Panamericana	Panamericana
巴拿马	巴拿马
Panamá Jack	Panamá_Jack
Panaria	Panaria
Panarottis	Panarottis
松下	松下
Panatseya	Panatseya
煎饼屋	Pancake_House
潘城港	潘城港

名称	ID
Panchero 的墨西哥烧烤	Panchero's _Mexican_Grill
Pancho	Pancho
Pancho 的墨西哥自助餐	Pancho's _墨西哥_自助餐
Pancit Ng Mga Taga Malabon	Pancit_Ng_Mga_Taga_Malabon
熊猫	熊猫
熊猫快车	熊猫快车
熊猫旅馆	Panda_Inn
Pandati	Pandati
潘多拉	潘多拉
潘都物流	泛都_物流
潘杜罗	潘杜罗
潘杜罗霍比	Panduro_Hobby
潘迪安格拉玛银行	Pandyan_Grama_Bank
帕内拉面包	Panera_Bread
Panggelia	Panggelia
Panhandle efcu-panHandle	panHandle_efcu-panHandle_cu
Panier Sympa	Panier_Sympa
帕宁银行	Panin_Bank
帕尼尼	帕尼尼
Pannon Takarék	Pannon_Takarék

名称	ID
Pannon Takarék Bank	Pannon_Takarek_Bank
Pannonhalma 是 Vidéke Takarekszöv。	pannonHalma_és_vidéke_takarekszöv。
Pannonia	Pannonia
帕诺斯	帕诺斯
Pans & Company	Pans_&_Company
Panta Rhei	Panta_Rhei
Pantalooons	Pantalooons
Pantashop	Pantashop
Panvel	Panvel
攀枝花市商业银行	攀枝花_城市_商业银行
攀枝花农村商业银行	攀枝花_农村_商业银行
潘索	潘索
保罗·孔蒂	Paolo_Conte
Pap Oil	Pap_Oil
爸爸吉诺的	Papa_Gino's
爸爸约翰的	爸爸约翰的
爸爸墨菲的	Papa_Murphy's
Papa Romano 的	Papa_Romano's
爸爸的披萨要外带	papa's_pizza_to_go
Papagalino	Papagalino

名称	ID
Papantoniou Bakeries	Papantoniou_Bakeries
Papasotiriou	Papasotiriou
Paper Pl	Paper_Plu
纸品商店	Paper_Store
Papizza	Papizza
Pappa Rich	Pappa_Rich
Pappa Roti	Pappa_Roti
PappaRich	PappaRich
Pappas Bar-B-Q	Pappas_Bar-B-Q
Pappasito's Cantina	Pappasito's _cantina
Pappert's Báckerei	Pappert's_báckerei
辣椒粉	辣椒粉
Par Mar 门店	Par_mar_Stores
帕尔玛特	Par_Mart
Paraboot	Paraboot
游行	游行
半位数	半位数
Paradis	Paradis
天堂	天堂
天堂面包店	天堂面包店

名称	ID
天堂漫步	Paradise_Walk
天堂度假村	Paradisus_度假村
帕拉多尔纳西纳尔酒店	Parador_Nacional
Parallel	Parallel
帕拉莫	帕拉莫
派拉蒙银行	派拉蒙银行
派拉蒙酒店	派拉蒙酒店
Parcel P	Parcel_Plus
Parcus	Parcus
帕尔多斯鸡肉	Pardos_Chicke
没有伴侣的父母	没有伴侣的父母
Parfümerie Becker	Parfümerie_Becker
巴黎	巴黎
巴黎熟食店	Paris_Deli
巴黎画廊	Paris_Gallery
巴黎蛋糕	Paris_Gateaux
Parishop	Parishop
巴黎的	巴黎的
帕里西纳	帕里西纳
Paritate	Paritate

名称	ID
Paritetbank	Paritetbank
停车和充电	park_&_Charge
公园和套房	公园和套房
Park 'N Fly	park_'N_Fly
公园大道	公园大道
Park Bravo	park_Bravo
柏悦酒店	柏悦酒店
雷迪森公园酒店	Radisson 的 park_inn_by_by
国际公园旅馆	国际公园酒店
公园广场	公园广场
Park Rite	Park_Rite
安全停车	安全停车
帕克城	帕克城
帕克·林奇	Parker_Lynch
帕克的	帕克的
鹿特丹 Parkeren	Parkeren_鹿特丹
Parkhaus-Betriebs-GmbH 法兰克福 M	parkhaus-betriebs-Gmbh_Frankfurt_M。
停车	停车
停车 + 保护	停车_+_保护
停车快车	停车快车

名称	ID
Parkings BSM	parkings_bsm
帕克林	帕克林
Parkview 社区 FCU	Parkview_Community_FCU
帕尔梅纳拉	帕尔梅纳拉
帕尔马	帕尔马
Parousiasi	Parousiasi
Parrot-Ice	Parrot-Ice
PartSource	PartSource
合伙人 Rompetrol	Partener_Rompetro
Partiscum Takarék	Partiscum_Takarék
帕蒂齐亚·阿雷顿	Partizia_Ary
合作伙伴	合作伙伴
合作伙伴：科罗拉多大学	合作伙伴_Colorado_cu
合作伙伴 1st FCU	Partners_1st_FCU
合作伙伴 Health	合作伙伴_健康
Partouche	Partouche
美国零部件	零件_美国
Parts City Auto Parts——博雷汽车零部件	parts_city_auto_parts_-_bray's_auto_Parts
零件升级版	零件_Plus
美国派对	派对_美国

名称	ID
派对之城	派对城市
派对厅 KTV	Party_room_KTV
帕鲁奥洛	帕鲁奥洛
Pasabahce	Pasabahce
Pasaport 披萨	护照_披萨
Pasaraya Borong Sakan	Pasaraya_Borong_Sakan
Pasaraya KU	pasaraya_ku
Pasaraya MU	pasaraya_mu
Pascal Coste	Pascal_Coste
Paschim Banga Gramin Bank	Paschim_Banga_Gramin_Bank
Pascoes the Jewelers	Pascoes_the_jewelers
Pascucci	Pascucci
Pasoil	Pasoil
通道 FitnessFirst	段落_ FitnessFirst
Passarela	Passarela
百香咖啡	激情咖啡
激情美女	Passion_Beauté
Passport Heal	护照_健康_通讯
护照旅馆	Passport_Inn
护照 Seva Kendra	Passport_Seva_Kendra

名称	ID
意大利面城	Pasta_City
舒适的意大利面	Pasta_Cosy
可可意大利面	Pasta_de_Coco
意大利面屋公司	Pasta_House_Co
番茄意面	Pasta_Pomodoro
王牌面食	Pasta_Wang
Pasta 你明白了！	pasta_Ya_Gotcha！
Pastarito Pizzarito	Pastarito_Pizzarito
意大利粉彩	Pastel_Italiana
Pastelerias Esperanza	Pastelerias_Esperanza
Pasteleria Montparnasse	Pasteleria_Montparnasse
Pastes kikos	Pastes_kikos
巴斯德	巴斯德
牧师	牧师
PatPrimo	PatPrimo
Patacones	Patacones
巴塔哥尼亚	巴塔哥尼亚
Patchi	Patchi
Patco 便利店	Patco_Handy_Shop
Pateo	Pateo

名称	ID
探路者教育中心	探路者_教育_中心
路径标记	路径标记
Pathways 金融CU	Pathways_financial_c
路径	路径
Pathé	Pathé
PatikaPlus	PatikaPlus
法式蛋糕店瓦莱丽	Patisserie_Valerie
帕特里亚	帕特里亚
Patropi	Patropi
保罗咖啡厅	保罗_Cafe
保罗哈里斯	保罗_哈里斯
保罗史密斯	保罗·史密斯
保罗的仓库	保罗的仓库
暂停咖啡厅	Pause_Café
摊铺机	摊铺机
展馆	展馆
展馆药房	展馆_药房
帕夫洛夫斯卡娅·库罗奇卡	Pavlovskaya_Kurochk
帕夫斯科	帕夫斯科
Pax-Bank	Pax-Bank

名称	ID
帕克斯顿公园酒店	Paxton_Park_Hotel
少付钱	少付钱
少付燃料中心	Pay_Less_Fuel_Center
少付药房	少付药房
Pay Rite	Pay_Rite
Paychex	Paychex
Payless	Payless
无薪租车	Payless_Car_Rental
无薪鞋	Payless_Shoes
Payless Shoesource	Payless_Shoesource
帕斯	帕斯
Pcan	Pcan
和平收藏	和平系列
PeaceHealth	PeaceHealth
桃州 FCU	peach_state_FCU
孔雀商店	Peacock_Store
孔雀专卖店	Peacock's Stores
孔雀	孔雀
峰值	峰值
峰值性能	峰值性能

名称	ID
山顶药房	高峰药房
珍珠牙科	Pearl_Dental
夏威夷珍珠 FCU	pearl_Hawaii_fcu
珍珠指甲	Pearl_Nails
Pearle	Pearle
Pearle 视觉中心	珍珠视觉中心
皮尔逊	皮尔逊
Pecel Lele LELE LELA	pecel_lele_Lela
Pechki-Lavochki	Pechki-Lavochki
Peck 的市场	Peck's_Market
Pedders	Pedders
Pedemeia	Pedemeia
Pedro	Pedro
佩德罗·胡安和迭戈	Pedro_Juan_&_Diego
佩德罗·德尔·耶罗	pedro_del_Hierro
Peebles	Peebles
Peek & Cloppenburg	Peek_&_Cloppenburg
Peekaboo	Peekaboo
Peermont	Peermont
Peet 的咖啡	peet's_coffee

名称	ID
Pegadaian	Pegadaian
Peggy Sue 的	Peggy_Sue's
Pehlivanoglu	Pehlivanoglu
Pei Wei 亚洲餐厅	Pei_Wei_Asian_Diner
培杰火锅	Peijie_Hotpot
沛县农村商业银行	沛县_农村_商业银行
Pek-Nack	Pek-Nack
Pekarna Pecjak	Pekarna_Pecjak
Pekarne S ILa	Pekarne_ S ILa
Pekarstvi JECMINEK	pekarstvi_jecminek
Pekarstvi Baker 先生	Pekarstvi_mr._Baker
Pekdemir	Pekdemir
北京商会	北京会议厅
北京花园	北京花园
北京 Newface 酒店	北京_Newface_Hotel
Peko	Peko
Pelephone	Pelephone
鹈鹕州立大学	pelican_state_cu
鹈鹕服装店	Pelican_Clothing_store
Pelicana	Pelicana

名称	ID
Pelikan	Pelikan
Pelit	Pelit
Pella 门窗	Pella_windows_&_Doors
Peluquería BQ	peluqueria_bq
墨西哥石油公司	墨西哥石油公司
Pempek Pak Raden	Pempek_Pak_Raden
Pen Air FCU	pen_air_fcu
槟城路著名的 TeoChew Chendol	Penang_Road_Famous_ _Chend TeoChew ol
彭城农商银行	彭城_农村_商业银行
企鹅	企鹅
鹏友租车	鹏友_租车
Penhaligon 的	Penhaligon 的
CU 半岛	peninsula_cu
半岛酒店	半岛酒店
宾夕法尼亚联邦储蓄银行	宾夕法尼亚联邦储蓄银行
宾夕法尼亚车站东海岸潜艇	Penn_Station_East_Coast_Subs
彭宁顿斯	彭宁顿斯
Pennstar 银行	Pennstar_Bank
Pennypinchers	Pennypinchers
Pennz Mart	Pennz_Mart

名称	ID
Pennzoil 10 分钟换油	penzoil_10_minute_Oil_Change
佩诺布斯科特县 FCU	penobscot_county_fcu
Penshoppe	Penshoppe
Penske 汽车中心	Penske_Auto_Cent
Penske 卡车租赁	Penske_Truck_Retal
Pentagone	Pentagone
宾得	宾得
Penti	Penti
Pento	Pento
人员	人员
人民选择信用合作社	人民选择信用联盟
人们的电话	人们的电话
人民联合银行	人民联合银行
PeopleReady	PeopleReady
人民	人民
人民联盟 FCU	People_Alliance_FCU
人民银行	人民银行
人民至上	以人为本
人民珠宝商	人物_珠宝商
人民国家银行	人民国民银行

名称	ID
人民信托 FCU	Peoples_trust_F
Peora	Peora
Pep Boys	Pep_Boy
Pepe 牛仔裤	Pepe_Jeans
Pepe's 墨西哥餐厅	Pepe's 墨西哥餐厅
Pepecar	Pepecar
Peplos	Peplos
Pepper H	胡椒_汉密尔顿
胡椒午餐	胡椒午餐
胡椒炒	胡椒炒
Pepperjax 烧烤	Pepperjax_Gril
Pepper Artisan	Peppery_Art
Peppes 披萨	peppes_Pizz
百事可乐装瓶公司	百事可乐装瓶公司
PerSu	PerSu
Perekrestok	Perekrestok
Perekriostok	Perekriostok
Perekryostok OYL	perekryostok_oyl
Perene	Perene
完美外观	完美外观

名称	ID
完美的皮塔饼	Perfect_Pita
完美的服务	完美服务
完美的形状	完美形状
完美西装 FActory	完美套装_ FActory
完美的身体中心	完美的身体中心
高性能自行车	性能_自行车
香水狂热	香水狂热
朱莉娅香水店	香水_Julia
阿塔拉亚香水店	香水_Atalaya
Perfumerías Aromas	Perfumerías_Aromas
Primerías PRIMOR	perfumerías_Primor
香水和公司	香水_&_Companhia
Perkins	Perkins
Permata Bank	Permata_Bank
Permata Bank Syariah	Permata_Bank_Syariah
Permfarmatsiya	Permfarmatsiya
佩尔恩咖啡	Pern_coffee
perng yuh	perng_yuh
Perodua	Perodua
Perrino	Perrino

名称	ID
Perros y Burros	perros_y_burros
佩里·埃利斯	Perry_Ellis
佩里体育	Perry_Sport
波斯语	波斯语
Pershyi Investytsiinyi 银行	Pershyi_Investytsiinyi_Bank
Person 国际教育	个人_国际_教育
Pertamina	Pertamina
Pervaya Polosa	Pervaya_Polosa
Pervaya Pomoshch'	Pervaya_Pomoshch'
Pervobank	Pervobank
Pervomaisky	Pervomaisky
PesaPoint	PesaPoint
Peshekhod	Peshekhod
佩斯塔纳	佩斯塔纳
宠物中心	Pet_Center
宠物城	Pet_City
宠物食品快递	宠物_食品快递
宠物乐园	Pet_Land
宠物爱好者中心	宠物爱好者中心
宠物停靠站	Pet_Stop

名称	ID
宠物套房	宠物套房
宠物超市	宠物超市
宠物用品 Plus	宠物用品_Plus
宠物价值	Pet_Valu
Petbarn	Petbarn
皮特市场	Pete's Market
Petec	Petec
彼得·亚历山大	彼得_亚历山大
彼得·哈恩	Peter_Hahn
彼得派珀披萨	Peter_Piper_Pizz
彼得的美食家	Peter's GOURM
Peter's Gute Backstube	Peter's_gute_backstube
彼得的草地	Peter's Meadow
彼得比尔特	彼得比尔特
Peterburgskie Apteki	Peterburgskie_Apteki
Petiskeira	Petiskeira
小赌场	Petit_Casino
Petit Patapon	Petit_Patapon
PetitBistro	PetitBistro
加泰罗尼亚佩蒂斯格兰斯酒店	Petits_Grans_Hotels_de_Catalunya

名称	ID
Petland	Petland
宠物乐园折扣	Petland_折扣
Petline	Petline
Petline 市场	Petline_Mark
佩特拉	佩特拉
Petro	Petro
博茨瓦纳石油公司	博茨瓦纳石油公司
Petro快车	石油快车
肯尼亚石油公司	肯尼亚石油公司
Petro Mart	Petro_Mart
石油储藏室	石油储藏室
七号石油公司	石油_Seven
Petro 停靠中心	石油停靠中心
Petro-T	Petro-T
Petro-T 卡锁	Petro-T 卡锁
PetroChina	PetroChina
PetroLimex	PetroLimex
巴西石油公司	巴西石油公司
彼得罗卡特	彼得罗卡特
Petroci	Petroci

名称	ID
Petroda	Petroda
石油三角洲	石油三角洲
Petrofib	Petrofib
Petrogal	Petrogal
Petrokommerts	Petrokommerts
乌克兰石油公司	乌克兰石油公司
石油办公室	Petrol_Ofisi
石油	石油
Petrom	Petrom
Petromar	Petromar
Petromil	Petromil
Petromin	Petromin
Petromin快车	Petromin_Exp
Petromoc	Petromoc
Petromoc e Sasol Sarl	petromoc_e_sasol_sarl
Petron	Petron
Petron Shop	Petron_Shop
马来西亚国家石油公司	马来西亚国家石油公司
Petronas NGV	Petronas_NGV
Petronor	Petronor

名称	ID
Petroser	Petroser
彼得罗苏尔	彼得罗苏尔
石油贸易	石油贸易
彼得罗维奇	彼得罗维奇
宠物天堂	Pets_Paradise
家里的宠物	Pets_at_Home
Petz	Petz
Pevex	Pevex
Peynirci Baba	Peynirci_Baba
Pfaltzgraff	Pfaltzgraff
Pfister	Pfister
Phano	Phano
幻影	幻影
制药专家	制药专家
Pharma Pl	Pharma_Plus
PharmaPoint lekarna	PharmaPoint_lekarna
Liegeoises 热门药店	药店_Populaires_Liegeoises
药效	药效
SPAR 的药房	pharmacy_at_spar
Pharmapele	Pharmapele

名称	ID
Pharmaprix	Pharmaprix
Pharmasave	Pharmasave
Pharmavie	Pharmavie
Pharro	Pharro
第八阶段	第八阶段
Pheple FCU	pheple_fcu
菲尔的汉堡	Phil's Burger
菲尔达尔	菲尔达尔
菲律宾航空	菲律宾_航空公司
菲律宾交通银行	菲律宾交通银行
菲律宾商业银行	菲律宾_商业_银行
菲律宾国家银行	菲律宾国民银行
菲律宾储蓄银行	菲律宾_储蓄银行
菲律宾信托银行	菲律宾信托银行
飞利浦	飞利浦
菲利普斯 66	Phillips_66
费城连接	费城连接
费城椒盐脆饼工厂	Philly_Pretzel_Factory
菲尔兹咖啡	Philz_Coff
Phiten	Phiten

名称	ID
Pho 24	Pho_24
Pho Hoa	Pho_Hoa
Pho Ong Hung	Pho_ong_Hung
Phoenix	Phoenix
凤凰城儿童医院	凤凰城儿童医院
凤凰市场	凤凰市场
手机 4U	Phones_4U
摄影工作室 Turtle	照片_Studio_Turtle
Phuc Long	Phuc_Long
Phungnoi 面包店	Phungnoi_Bakery
物理	物理
Piacere	Piacere
Piaomiaofang 美发	Piaomiaofang_美发
Piatto	Piatto
Pick OuiC	pick_OUIC
皮卡德	皮卡德
Picardi Rebel	Picardi_Rebel
Picc	Picc
皮卡迪利自助餐厅	Piccadilly_自助餐厅
Piccadily	Piccadily

名称	ID
Pick 'n Save	pick_'n_Save
Pick Up Stix	pick_Up_Stix
Pick n Pay	pick_n_Pay
Pick n Pay 服装	pick_n_pay_Clothing
Pick n Pay Ex	pick_n_pay_Express
Pick n Pay 超级	pick_n_pay_Hyper
Pick n Pay 白酒	pick_n_pay_Liquir
Pick n Pay 药房	pick_n_pay_Pharmacy
Pickleman 的美食咖啡厅	Pickleman's _Goorment_Caf
选择点	选择点
Pico	Pico
Picturehou	Picturehou
Pidem	Pidem
馅饼城	Pie_City
馅饼脸	Pie_Face
碎片	碎片
拼图自动	Pieces_Aut
皮埃蒙特优势 CU	Piedmont_Advantage_c
皮埃蒙特联邦	Piedmont_Federal
Piel de Toro	Piel_de_toro

名称	ID
1 号码头进口	Pier_1_ 进口
穿孔宝塔	Percing_Pagoda
皮埃尔·马可里尼	Pierre_Marcolini
Piery	Piery
彼得罗·菲利皮	Pietro_Filipi
小猪忙碌	Pig_Busy
Piggly Wiggly	Piggly_Wiggly
色素	色素
皮拉尔·普列托	Pilar_Prieto
朝圣者	朝圣者
朝圣者清洁工	朝圣者_清洁工
支柱 CU	pillar_cu
Pillar To	Pillar_To_Post
飞行员	飞行员
Pim	Pim
Pimapen	Pimapen
田品	Pin_Tian
pin&Go！	pin&Go！
皮纳尔县 FCU	pinal_county_fcu
Pinanbao	Pinanbao

名称	ID
Pinch a Penny 泳池露台水疗中心	pinch_a_penny_pool_patio_spa
Pind Balluchi	Pind_Balluchi
Pine Bluff 棉带 FCU	pine_bluff_cotton_belt_belt_fc
松树社区 CU	pine_tree_Community_cu
菠萝服装商	Pineapple_Outfitter
平安银行	平安银行
中国平安	ping_an_of_China
平皇咖啡	Ping_Huang_Co
平河味道	Ping_River_Tast
平安证券	平安证券
Pingvin Patica	Pingvin_Patika
平阳农村商业银行	平阳_农村_商业银行
Pink Biju	pink_Biju
粉色新女孩	pink_new_Girl
Pinkberry	Pinkberry
粉红	粉红
Pinnacle Bank	Pinnacle_Bank
Pinnacle CU	pinnacle_cu
Pinoil	Pinoil
Pinot 的调色板	Pinot's_Palette

名称	ID
品尚豆老	Pinshang_豆老
Pintu Sayer	Pintu_Sayer
Pintumex	Pintumex
Pinturas Juno	Pinturas_Juno
PintxoTerapia	PintxoTerapia
品正滨	品正滨
先锋	先锋
先锋银行	Pioneer_Bank
先锋 FCU	Pioneer_FCU
先锋超市	先锋_超市
Piotr I Pawel	Piotr_I_Pawel
Piotr i Pawel po drodze	piotr_i_pawel_po_drodze
比雷埃夫斯银行	比雷埃斯银行
倍耐力	倍耐力
Pirex Papír	Pirex_Papir
Pirtek	Pirtek
Pisanello 的披萨	Pisanello's_Pizza
皮斯塔之家	Pista_House
进站	Pit_Stop
皮塔丛林	Pita Jungle

名称	ID
Pita Pit	Pita_Pit
Piticas	Piticas
Pitico	Pitico
Pittappillil 代理商	Pittappillil_机构
Pivac	Pivac
Pivdennyi	Pivdennyi
Pix 便利店	Pix_便利店
Pixel-J	Pixel-J
Piyu	Piyu
Piza e Vino	piza_e_vino
滁州农村商业银行	Pizhou_农村_商业银行
Pizza 4P	Pizza_4P's
披萨 73	Pizza_73
阿格瓦尼亚披萨	Pizza_agvania
披萨小吃	Pizza_Bite
Pizza Boli	Pizza_Boli's
披萨刺山柑	Pizza_Capers
Celentano Pizza	Pizza_Celentano
披萨鸡肉完美	披萨_鸡肉_完美
披萨厨师	Pizza_Cooc

名称	ID
披萨角	披萨专区
披萨乐趣	披萨_喜悦
披萨书房	Pizza_Den
披萨仓库	披萨_仓库
披萨快递	披萨_Express
披萨工厂	Pizza_Fabrika
披萨工厂	披萨工厂
新鲜披萨	披萨_Fresh
披萨小伙子	Pizza_伙计们
披萨天堂	披萨天堂
披萨屋	披萨屋
必胜客	Pizza_Hut
披萨旅馆	Pizza_Inn
Pizza Jardin	Pizza_Jardin
爵士披萨	披萨_Jazz
披萨 Joe's	Pizza_Joe's
披萨之王	Pizza_King
披萨 Lazza	Pizza_Lazza
披萨小派对	Pizza_Little_Party
披萨人	Pizza_Man

名称	ID
披萨 Micha	Pizza_Micha
新星披萨	Pizza_Nov
Pizza Patron	Pizza_Patron
披萨派伊	Pizza_Pai
Pepone 披萨	Pizza_pepone
披萨专业版	Pizza_Pro
披萨牧场	披萨_牧场
Raul 披萨	Pizza_Raul
披萨摇滚	Pizza_Rock
披萨小屋	Pizza_Shack
披萨片	披萨_Slice
披萨广场	披萨广场
披萨时间	Pizza_Time
Z 披萨	Pizza_Z
披萨区	Pizza_Zone
艺术披萨	Pizza_del_Arte
PizzaRev	PizzaRev
Pizzabakeren	Pizzabakeren
Pizzabulls	Pizzabulls
Pizzas Angelotti	Pizzas_Angelotti

名称	ID
Pizzaville	Pizzaville
Marino 披萨店	披萨店_Marino
皮涅罗酒店	Piñero_Hoteles
Pkbk	Pkbk
PlaceMakers	PlaceMakers
Placecol	Placecol
格子储藏室	格子_储藏室
PlainsCapital 银行	PlainsCapital_银行
Plaisir 咖啡厅	Plaisir_Cafe
星球海滩	Planet_Beach
星球现金	Planet_Cash
星球健身	行星健身
爱好星球	Planet_Hobby
好莱坞星球	星球_好莱坞
星球小子	Planet_Kids
M 星球	Planet_M
星球钉子	Planet_Nails
星球香水	Planet_Parfum
星球冰沙	星球_冰沙
星球体育商店	Planet_Sports_Stores

名称	ID
星球潜艇	Planet_Sub
冲浪星球	Planet_Surf
X 星球	Planet_X
Planet'Indigo	Planet'Indigo
行星炸弹	Planeta_Bombom
Planeta D	Planeta_D
Planeta Sibir'	Planeta_Sibir'
Planeta 寿司	Planeta_Sushi
Planeta zdorov'ya	Planeta_zdorov'ya
电子星球	Planeta_Elektroniki
Planeta fitnes	Planeta_fitnes
Planeta kolgotok	Planeta_KolgoTok
计划生育	计划生育
植物睡眠	植物_睡眠
Plantage Books 及更多	Plantage_Books_&_更多
Plantasje	Plantasje
种植者银行	Planters_Bank
普拉斯彭	普拉斯彭
Platekompaniet	Platekompaniet
白金银行	Platinum_Bank

名称	ID
铂金 FCU	Platinum_fcu
柏拉图	柏拉图
柏拉图的壁橱	柏拉图的壁橱
柏拉图	柏拉图
鸭嘴兽	鸭嘴兽
柏拉图	柏拉图
再玩一次体育	Play_It Again_Sports
玩生活	玩乐生活
玩 N 笔交易	Play_N_Trade
PlayCity	PlayCity
普拉亚保龄球	Play_Bowl
普拉亚参议员	Playa_Sen
Playabc	Playabc
普拉亚尔特	普拉亚尔特
花花公子	花花公子
PlayersRoom	PlayersRoom
游乐园	游乐园
游戏生活	游戏生活
广场	广场
阿兹台克广场	Plaza_Azteca

名称	ID
广场酒店	Plaza_Hotel
广场药房	广场_药房
Plaza 轮胎和汽车服务	Plaza_Tire_&_Auto_Service
Vea 广场	Plaza_Vea
拜托妈妈	拜托_妈妈
Plow & Hearth	Plow_&_Hearth
Pluckers 翼杆	Pluckers_Wing_Bar
Plug'n Charge	Plug'n_Charge
管道升级版	Plembing_Plus
水暖世界	Plembing_World
Plamblink	Plamblink
Plunke	Plunke
再加上	再加上
再加上 +	Plus_+
Plus 市场	加上市场
Plus Superdesco	Plus_Superdescomp
Plus15Protect	Plus15Protect
毛绒	毛绒
Plus bank	Plyus_Ban
Pmo	Pmo

名称	ID
Penue Egger	Neu_Egger
Neubox	Neubox
Neuhage	Neuhage
Pneuvita	Pneuvita
Po Karmanu	Po_Karmanu
Pochta Bank	Pochta_Bank
Pochtomat Halva	Pochtomat_Halva
口袋商店	Pocket_shop
Poczta Polska	Poczta_Polska
吊舱选择	Pod_Choice
Pod Inn	Pod_Inn
吊舱积分	Pod_Point
Pod 选择	Pod_Select
讲台	讲台
Podkarpacki Bank Spoldzielczy	Podkarpacki_Bank_Spoldzielczy
Podmotel	Podmotel
波多罗日尼克	波多罗日尼克
波德鲁日卡	波德鲁日卡
Podunavlje	Podunavlje
诗意的东方之美	诗意_东方_美丽

名称	ID
Pohona Ph	Pohona_Ph
Poidem	Poidem
Poiesz	Poiesz
点	点
Point Coop	Point_Coop
S 点	Point_S
Vert	点_Vert
Point.P	Point.P
Poisk	Poisk
Poivre rouge	Poivre_Rouge
Pojok Busana	Pojok_Busana
Pokupayka	Pokupayka
Pokupochka	Pokupochka
北极熊	Polar_Bear
Polare	Polare
北极星	北极星
北极星证券	北极星证券
Polata	Polata
Polen Tantuni	Polen_Tantuni
Polgat	Polgat

名称	ID
波尔加里银行	Polgári_Bank
波利康银行	波利康银行
波利斯公园	Polis_Park
Polisan	Polisan
Politix	Politix
Pollet	Pollet
Pollini	Pollini
Pollo Brujo	pollo_Brujo
Pollo Feliz	pollo_Feliz
波洛热带风情	Pollo_Tropic
波洛斯里奥	Pollos_Rio
Polly's Pies 餐厅和面包店	Polly's_pies_restaurant_&_Bakery
Polo	Polo
Polo 车库	Polo_Garage
马球市场	Polo_Market
Polo 服	Polo_Wear
Polska SKOK	Polska_skok
Polskie Sklady Budowlane PSB	polskie_sklady_budowlane_PSB
波尔塔瓦银行	波尔塔瓦银行
波尔顿国际服务公寓	波尔顿_国际_服务_Apt

名称	ID
Polushka	Polushka
保利国际影城	保利国际影院
Pomellato	Pomellato
Pomme de Pain	pomme_de_pain
Pommersche 大众银行	Pommersche_Volksbank
番茄披萨	Pomodori_Pizza
Pompa	Pompa
Pomponchik	Pomponchik
Ponderosa	Ponderosa
Pong	Pong
Ponkid 儿童世界	ponkid_Children's Worl
蓬	蓬
十六浦德兴海鲜火锅	Ponte_16_Tak_Heng_Seafood_Hot_Pot
Ponto Fresco	Ponto_Fresco
小马奔跑	跑小马
Poolwerx	Poolwerx
Poorvika 手机	Porvika_Mobile
波普玛特	Pop_Mart
Pop Tate	Pop_Tate's
Pop-A-Lock	Pop-A-Lock

名称	ID
教皇	教皇
Popeye 的补品	Popeye's 补品
Popeyes 路易斯安那厨房	Popeyes_Louisiana_K
Popia	Popia
杨树	杨树
Popo la mama	popo_la_mama
Poppo	Poppo
Popsy	Popsy
最受欢迎	最受欢迎
Popular-e.com	Popular-e.com
猪肉店	Pork_Shop
爪哇港口城市	Port_City_Java
港口酒店	港口酒店
潜艇港口	订阅端口
portamedic	portamedic
冒险港酒店	Portaventura_Hoteles
蓝港酒店及度假村	Portblue _酒店及度假村
Portillo's	Portillo's
波蒂略的餐厅	Portillo's 餐厅
波特兰布朗	波特兰_布朗

名称	ID
波特兰玻璃	波特兰玻璃公司
波特曼建筑协会	波特曼建筑协会
波特曼斯	波特曼斯
波尔图湾	Porto_Bay
波尔图埃迪托拉	Porto_Editora
塞古罗港	Porto_Seguro
波尔图-弗兰科	波尔图-弗兰科
Portobello Shop	Portobello_Shop
波托菲诺酒店及度假村	波托菲诺_酒店及度假村
人像创新	肖像_创新
波特说	波特说
葡萄牙	葡萄牙
Pos 印度尼西亚	Pos_印度尼西亚
波萨德斯基伊	波萨德斯基伊
Poscer	Poscer
Posojilnica Bank	Posojilnica_Bank
邮局	邮局
邮局员工 CU	邮局_员工_cu
斯洛文尼亚邮政	Posta_Slovenije
Posta Srpske	Posta_Srpske

名称	ID
邮政连接	邮政连接
中国邮政储蓄银行	中国银行邮政储蓄银行
Postanska stedionica	Postanska_stedionica
邮政银行	邮政银行
Postbank Immobilien GmbH	Postbank_immobilien_GmbH
邮政银行	邮政银行
意大利邮政	Poste_Italiane
摩洛哥邮政	Poste_Maroc
Posten	Posten
邮政财务	邮政财务
Postie	Postie
Postiljon	Postiljon
斯洛文尼亚邮政银行	Postna_Banka_Slovenije
额外的 Posto	帖子_Extra
Posto makro	Posto_makro
Postova Banka	Postova_Banka
火锅神	pot_god
土豆角	Potato_Corner
Potbelly 三明治	Potbelly_三明治店
潜力	潜力

名称	ID
Potlatch No 1 FCU	potlatch_no_1_fcu
Potlatch No 1 Financial	potlatch_no_1_financial_cu
Potrefena Husa	Potrefena_Husa
Potrefena husa	Potrefena_HUSA
Potugalia	Potugalia
Potzollcalli	Potzollcalli
庞德面包店	磅_面包店
Poundland	Poundland
Poundstretcher	Poundstretcher
为你而设	Pour_Vous
普萨达	普萨达
葡萄牙旅馆	波萨达斯_de_Portug
强力购买	Power_Buy
Power CU	power_cu
Power Financ	power_financial_cu
强力语言训练	强力_语言_训练
力量寿命	力量_Life
电力销售	Power_Sales
强力睡眠	电源_Sleep
Power-day (WuHan) 商务酒店	Power-day (WuHan) _商务_酒店

名称	ID
强力本	强力本
Powerflow 排气	Powerflow_Exausts
强力健身房	强力健身房
宝龙	宝龙
Powszechna SKOK	powszechna_skok
Poya 生活超市	Poya_Living_Mart
Pozaa 茶和咖啡	Pozaa_Tea_&_Coffe
Poziomka	Poziomka
Pozitronika	Pozitronika
Prabhat Khabar	Prabhat_Khabar
实用的汽车和货车租赁	实用_汽车_和_货车_出租
布拉格	布拉格
Pragathi Grameena 银行	Pragathi_Grameena_Bank
Praktiker	Praktiker
Prashanthi 阿育吠陀中心	Prashanthi_Ayurvedic_Centre
Prata Fina	Prata_Fina
Prathama UP Gramin Bank	Prathama_up_gramin_Bank
Pravek	Pravek
Praveks-Bank	Praveks-Bank
Praxair	Praxair

名称	ID
Prazdnichniy	Prazdnichniy
Prazeres da Picanha	prazeres_da_picanha
精密汽车	精密_汽车
精密门服务	精准门上门服务
汽车办公精度	Precision_oficina_automóvel
精确调整汽车保养	Precision_Tune_Auto_Car
精度	精度
Preem	Preem
首选 CU	preferred_cu
Prego	Prego
Prehrana	Prehrana
Premaman	Premaman
总理	总理
尊享商务中心	尊享商务中心
高级社区 CU	premier_Community_cu
总理 FCU	Premier_fcu
普瑞米尔酒店	Premier_酒店
普瑞米尔旅馆	Premier_Inn
普瑞米尔旅馆	Premier_Lodg
高级会员 CU	Premier_members_c

名称	ID
尊享停车	尊享停车位
高级紧急护理	Premier_Urgent_Care
PremierGarage	PremierGarage
首映国际影院	首映_国际_影院
Premio	Premio
高级	高级
预备课程	准备课程
长老会	长老会教堂
总统证券	证券总裁
总统超市	总裁_超市
Presotea	Presotea
新闻车	新闻车
Press Sec Tintorerie	Press_Sec_Tintoreries
新闻商店	新闻店铺
Pressbyrån	Pressbyrån
新闻咖啡厅	新闻咖啡厅
压榨榨汁机	Pressed_Juicery
时间紧迫	pressed_in_time
按下 4 次	按下 4 次
Pressto	Pressto

名称	ID
威望	威望
尊贵社区 CU	Prestige_Community_cu
尊享专属礼遇	Prestige_Exclusive
Presto	Presto
Presto！	Presto！
Preston	Preston
Pret a Manger	pret_a_manger
漂亮的芭蕾舞演员	Pretty_Ballerinas
漂亮女孩	Pretty_Girl
椒盐脆饼时间	Pretzel_Time
椒盐脆饼世界	Pretzel_World
椒盐脆饼机	椒盐脆饼机
Previlab	Previlab
Prezident	Prezident
Prezunic	Prezunic
价格	价格
价格攻击	价格攻击
价格削减器	Price_Chopper
价格联系人	价格_联系我们
Price Rite	Price_Rite

名称	ID
Price Rite 药房	价格_Rite_Pharmacy
价格鞋子	价格_鞋子
Pricele$$ Rent-A-Car	Pricele$$_Rent-A-Car
无价之鞋	Priceless_Shoes
价格线	价格线
PricewaterhouseCoopers	PricewaterhouseCoopers
骄傲	骄傲
普莱德酒店	Pride_Hotel
Prima	Prima
Prima Banka	Prima_Banka
PrimaPrix	PrimaPrix
Primark	Primark
小学农业合作社信贷协会	Primary_agri_Co-op_credit_S
普里马韦拉	普里马韦拉
Prime	Prime
主要银行	Prime_Bank
主要信用	Prime_Credi
Prime Finan	prime_financial_cu
黄金时段	Prime_Time
顶级酒吧	Prime_pubs

名称	ID
Prime 餐厅	Prime_restaur
PrimeStar 太阳能	PrimeStar_太阳能
PrimeTrust 财务	PrimeTrust_金融
Primera	Primera
Primerica 金融服务	Primerica_金融_服务
Primo	Primo
Primo Ciao Ciao	Primo_Ciao_Ciao
PrimoHoagies	PrimoHoagies
樱草学校	报春花_学校
Primsotsbank	Primsotsbank
普里姆斯酒店	Primus_Hotel
王子度假酒店	公主度假村_酒店
公主	公主
主体	主体
公国建筑协会	公国_建筑_社会
原则	原则
Prink	Prink
打印档案	打印_档案
打印 Hakase	打印_Hakase
印刷工业 CU	印刷行业_cu

名称	ID
Printo	Printo
Priorbank	Priorbank
优先一级 CU	Priority_one_cu
佛罗里达州的 PriorityOne CU	PriorityOne_cu_of_florida
PriorityOne 银行	PriorityOne_银行
Prisma	Prisma
Privatbank	Privatbank
Privatbanka	Privatbanka
私人银行	私人银行
私营企业理事会	私营行业理事会
职业高尔夫	Pro_Golf
专业图片	Pro_Image
专业光学	Pro_Optik
专业剪辑	专业剪辑
专业服务	专业服务
ProBank	ProBank
ProBuild	ProBuild
ProCredit 银行	ProCredit_银行
ProPark	ProPark
ProSpanek	ProSpanek

名称	ID
ProStore	ProStore
积极主动	积极主动
Probanka	Probanka
普罗维兹内斯银行	普罗维兹内斯银行
Prodalit'	Prodalit'
Prodia	Prodia
Prodotvary	Prodotvary
产品 Ermolino	产品_Ermolino
亲眼目睹者	亲眼目睹者
专业的	专业的
专业包装工和搬运工	Professial_Packers_&_Movers
专业人士	专业人士
Profi	Profi
光学概况	配置文件_Optik
轮胎中心简介	轮胎中心简介
Profilo	Profilo
利润讲述	利润讲述
Proforma	Proforma
渐进式保险	渐进式保险
专业医疗保健	专业医疗保健

名称	ID
PromTransBank	PromTransBank
Promedica FCU	Promedica_fcu
杰出的油漆	突出_Paints
Prominvest	Prominvest
Promitheftiki	Promitheftiki
促销市场	促销市场
普罗莫格罗	普罗莫格罗
Promsvyaz'Bank	Promsvyaz'Bank
Promyslovo-Finansovyi 银行	Promyslovo-Finansovyi_Bank
Pronti	Pronti
立刻	立刻
Pronto 意大利咖啡馆	pronto_cafe_italian
ProntoMatic	ProntoMatic
Prontowash	Prontowash
财产损失评估师	财产损失评估师
Proserv	Proserv
繁荣银行	繁荣_银行
Prosto apteka	Prosto_apteka
Prostor	Prostor
普罗蒂亚酒店	Protea_酒店

名称	ID
保护画家	保护画家
蛋白棒和厨房	Protein_bar_&_kitchen
普罗透斯	普罗透斯
Protid Urmaker	Protid_Urmaker
Proton 杂货	Proton_杂货
普罗图尔酒店	普罗图尔酒店
Prouds	Prouds
普罗维登斯健康	普罗维登斯_健康
公积金银行	普罗维登特银行
Provigo	Provigo
普罗沃比斯酒店	普罗沃比斯_酒店
挑衅	挑衅
Proxi	Proxi
Proxim	Proxim
保诚	保诚
保诚加州地产	保诚_加州_房地产
保诚保险公司	保诚保险公司
保诚地产	保诚_房地产
保诚唐恩地产	Prudential_Towne_Realty
Prvca poljoprivredna ljekarna	prvca_poljoprivredna_ljekarna

名称	ID
Pryvatbank	Pryvatbank
Prémaman	Prémaman
产前	产前
Prima	Prima
Prima Pek	Prima_Pek
Proxim	Proxim
普安药房	普安_药房
PubliBike	PubliBike
公开	公开
大众银行	大众银行
越南大众银行	大众银行_越南
公开投注	Public Bet
公共服务 CU	公共服务_cu
公共存储	公共存储
Publix	Publix
Publix Pix	Publix_Pix
Puccino 的	Puccino 的
帕奇兄弟披萨	Pudge_Brothers_Pi
克里斯托港	克里斯托港
普辉体检	普惠_体检

名称	ID
普吉岛	普吉岛
Pul's	Pul's
Pulipchae	Pulipchae
Pull&Bea	Pull&Bea
脉冲星	脉冲星
脉冲药房	脉冲药房
Pulte Homes	Pulte_Homes
Pulu	Pulu
Pum Plastiques	Pum_Plastiques
美洲狮	美洲狮
Pumb	Pumb
泵	泵
泵和储藏室	泵和储藏室
振作起来	Pump_It_Up
Pump N Pan	Pump_N_Pantr
南瓜补丁	南瓜补丁
PunThai 咖啡	PunThai_咖啡
浦那区中央合作银行	浦那地区中央合作银行
浦那人民合作银行	Pune_people_cop_bank
浦那城市合作银行	Pune_urban_Co-op_Bank

名称	ID
Pune Zilla Madhyavarti Sahakari	Pune_Zilla_Madhyavarti_Sahakari
旁遮普和马哈拉施特拉邦合作银行	Punjab_&_Maharashtra_co-op_Bank
旁遮普邦和信德银行	旁遮普_&_Sind_Bank
旁遮普格拉明银行	Punjab_Gramin_Bank
旁遮普烧烤	Punjab_Grill
旁遮普国家银行	旁遮普国民银行
Punkt1	Punkt1
Punnitse & Saästa	Punnitse_&_Saästa
蓬塔德切洛	Punta_de_Cielo
一击	一击
Punto Azul	Punto_Azul
Punto Blanco	Punto_Blanco
Punto Sma	Punto_Sma
Punto di	punto_di
PuntoFarma	PuntoFarma
Puppy Town 动物医院	Puppy_Town_Animal_Hospital
Pupule Himawari Yakkyoku	Pupule_Himawari_Yakkyoku
珀科顿	珀科顿
普渡大学 FCU	purdue_fcu
Purdys Chocolatier	Purdys_Chocolatier

名称	ID
Pure Barre	Pure_Barre
纯绿色气体	Pure_Green_Gas
纯粹的放纵体验	Pure_Indulgence
Puregold	Puregold
PuroClean	PuroClean
Pusat Pakaian Hari Hari	pusat_pakaian_Hari_Hari
Kime 服装中心	pusat_pakaian_Kime
莆田农商银行	莆田_农村_商业银行
普陀农村商业银行	普陀_农村_商业银行
Pustelli	Pustelli
Puzata Khata	Puzata_Khata
Pxmart	Pxmart
pyat'yu Pyat'	pyat'yu_pyat'
Pyatyorochka	Pyatyorochka
金字塔 CU	Pyramid_cu
Pátria Papir	Pátria_Papir
Pátria Takarek	Pátria_Takarék
Schumacher Pâtisserie	Pâtisserie_Schumacher
Pao de Acucar	páo_de_Acucar
Pao e Companhia	páo_e_companhia

名称	ID
科特迪瓦石油公司	科特迪瓦石油公司
Petrodis	Petrodis
Pío Pío	Pío_Pío
Portico Basico	Portico_Basico
Portico interiorismo	Portico_interiorismo
Q Burger	Q_Burger
Q 电影院	Q_Cinemas
Q 酒店	Q_Hotel
Q Lube	Q_Lube
Q Mobile	Q_Mobile
Q PLUS	Q_PLUS
Q PLUS 市场	q_plus_Market
Q Team	Q_Team
Q&M 牙科	Q&M_Dental
Q-PARK	Q-PARK
Q-PARK EV	Q-PARK_EV
Q-SERVICE	Q-SERVICE
Q-SERVICE 摩托	Q-SERVICE_MOTO
Q-SERVICE 高级版	Q-SERVICE_PREM
Q-sec Tintorerías	q-sec_tintorerías

名称	ID
Q1	Q1
Q8	Q8
Q8 很简单	q8_easy
QAPAQ	QAPAQ
Qapaq-Agente	Qapaq-Agente
卡塔尔国际伊斯兰银行	卡塔尔_国际_伊斯兰银行
卡塔尔伊斯兰银行	卡塔尔伊斯兰银行
QBANO	QBANO
QBD 书店	qbd_the_bookshop
QFC	QFC
QFC 燃料中心	QFC_Fuel_Center
QFC 药房	QFC_药房
QG Jeitinho Caseiro	qg_jeitinho_caseiro
QHotels	QHotels
千叶	千叶
齐鲁银行	QILU_BANK
齐鲁证券	QILU_SECURITIES
青木连锁酒店	青木连锁酒店
QML 病理学	qml_病理学
QNB FINANSBANK	QNB_FINANSBANK

名称	ID
QNB Kesawan	qnb_kesawan
QPET	QPET
QS	QS
QSP	QSP
QUAESTOR	QUAESTOR
QUALIPET	QUALIPET
品质旅馆	品质旅馆
质量提升	质量_PLUS
QUANJUDE	QUANJUDE
量子	量子
宿舍酒店	QUARTERS_酒店
差不多准备好了	QUASI_PRONTI
QUAZI	QUAZI
QUEBRAMAR	QUEBRAMAR
女王拳击	女王拳击
女王商店	女王店
女王公园	女王公园
QUERO-QUERO	QUERO-QUERO
很快	很快
快速选择	快速选择

名称	ID
QUICSILVER	QUICSILVER
QUIOSQUE	QUIOSQUE
测验	测验
QVC	QVC
卡斯尔合作社	Qasr_Co-Operative_Society
卡塔尔中央银行	卡塔尔_中央银行
卡塔尔国家银行	卡塔尔国家银行
Qazaq Oil	Qazaq_Oil
Qazaq banki	qazaq_Banki
Qazaqtelekom	Qazaqtelekom
Qdoba	Qdoba
Qeelin	Qeelin
钱大马	钱大马
钱慧	钱慧
钱元	Qian_Yuan
钱智业学校	Qian_zhi_Ye_School
千峰教育	千峰_教育
钱贵	钱贵
千佳 Culiangwang	千佳_Culiangwang
千家汇	千家汇

名称	ID
千里行	千里行
前途	前途
桥多天宫	桥多天宫
启臣	启臣
启东农村商业银行	启东农村商业银行
秦客栈	Qin_Inn
奇纳国际影城	Qina_国际_影院
江青南素邦才	清江_南_速邦彩
庆丰蒸饺店	庆丰_蒸_水饺_店
青岛解放	青岛_解放
青岛农村商业银行	青岛_农村_商业银行
青海农村信贷	青海_农村_信贷
清河川佳	清河川佳
青花娇沙果鱼	Qinghuajiao_Shaguoyu
庆客隆	庆客隆
庆陵	庆陵
青龙新	青龙新
青木酒店	青木_酒店
清远农村商业银行	清远_农村_商业银行
钦农银行	秦农银行

名称	ID
齐商银行	启尚银行
秋果酒店	Qiu_Guo_Hotel
七心田	七心田
齐家飞	Qk_Ka_Fei
观致汽车	观致汽车
Qstar	Qstar
Quadri foglio	Quadri_foglio
贵格会牛排和润滑油	Quaker_Steak_&_Lube
优质乳制品	质量_乳制品
品质酒店	品质_酒店
优质集市	质量超市
高品质 Q Mart	Quality_Q_Mart
品质套房	品质套房
Qualys Hotel	Qualys_Hotel
Quan Ann Ngon	Quan_An_Ngon
全国健康苏诗	Quan_Guo_jian_Kang_Su_Shi
全福源	全福源
全金城	全金城
量子信用合作社	量子信用联盟
权威	权威

名称	ID
夸尔斯	夸尔斯
采石场	采石场
Qubee	Qubee
Qubus	Qubus
Quce	Quce
Que Pollo	Que_Pollo
皇后区	皇后区
皇后护城河故居	女王护城河故居
昆士兰州乡村信用合作社	昆士兰州_乡村_信用联盟
昆士兰 X-ray	昆士兰_X-Ray
Quemay	Quemay
昆汀	昆汀
Querol	Querol
Quest	Quest
任务诊断	任务诊断
Quest FCU	quest_fcu
奎斯特石油	任务_石油
Quester	Quester
快点	快点
快速检查	Quick_Chek

名称	ID
快吃鸡	快吃鸡
快速加油	快速加油
快车道	快车道
快速租赁	快速租赁
快速婚姻	快速婚姻
快手超市	Quick_Mar
快宫	快速宫殿
Quick Quack 洗车	Quick_Quack_Car_Wash
快速租赁	快速租房
快舒吧	Quick_Schuh
快速服务	快速服务
快捷购物	快速购物
快速停止	快速停止
快速跟踪	快速追踪
Quicklee 的	Quicklee 的
汞	汞
快道	快道
Quik 便利店	快捷便利店
Quik Stop	Quik_Stop
QuikTrip	QuikTrip

名称	ID
Quilon 广播服务 (QRS)	Quilon_Radio_Service_ (QRS)
Quincy CU	quincy_cu
Quincy's	Quincy's
Quinovic	Quinovic
五次方	五次方
Quinta Essencia	Quinta_Essencia
quiosque Chopp Brahma	quiosque_Chopp_Brahma
Quix	Quix
Quiznos 订阅	Quiznos_Sub
曲靖市商业银行	曲靖城市商业银行
曲美家居	曲美_家居_家居
古兰经合作社	古兰经_合作社_社会
Qushiduo	Qushiduo
qwestdex	qwestdex
R Carvalho Supermercado	R_Carvalho_Supermercado
R S Brothers	R_S_兄弟
R Y B 教育机构	R_Y_B_教育机构
R&B	R&B
R-G FCU	R-G_FCU
R-kioski	R-kioski

名称	ID
R.P. Lumber	R.P._Lumber
R.a.d.e.f.	R.a.d.e.f.
RA 寿司	ra_sushi
兔子 reznictvi	rabbit_reznictVI
RABELO	RABELO
RABIKA 咖啡	RABIKA_COFFEE
RACV Auto	racv_Auto
RADO	RADO
莱佛士酒店及度假村	莱佛士酒店及度假村
RAFIDAIN 银行	RAFIDAIN_BANK
彩虹之旅	彩虹之旅
RAK 陶瓷	rak_Ceramics
RAKBANK	RAKBANK
拉拉尔萨	拉拉尔萨
RALLIART	RALLIART
RALPH LAUREN	RALPH_LAUREN
RAM	RAM
华美达	华美达
拉斯塔	拉斯塔
RATNADEEP	RATNADEEP

名称	ID
RAUTANET	RAUTANET
劳蒂亚	劳蒂亚
RAUTIA K-Maatalous	Rautia_k-Maatalous
肆虐	肆虐
旅行	旅行
雷蒙德	雷蒙德
RAZVOJNA BANKA VOJVODINE	RAZVOJNA_BANKA_VOJVODINE
RB Altdorf-Ergolding	rb_altdorf-Ergolding
RB Aschaffenburg	rb_aschaffenburg
RB Bad Windsheim	rb_bad_Windsheim
RB i Cham-Roding-Furth m Wald	rb_cham-roding-Furth_im_Wald
RB Deggendorf-Platling	rb_deggendorf-Plattling
RB Donau-Iller	rb_dona-iller
RB Estenfeld-Bergtheim	rb_estenfeld-bergtheim
RB Füssen-Pfronten-Nesselwang	rb_füssen-pfronten-nesselWang
RB Grosostheim-Obernburg	rb_grosostheim-Obernburg
RB Gräfenb。 -Forchh。 -Esch。 -Heroldsb	rb_gräfenB。 -Forchh。 -Esch。 -Heroldsb
RB Hallertau	rb_hallertau
RB Hersbruck	rb_Hersbruck
RB Höchberg	rb_Höchberg

名称	ID
RB I. Lkrs。Passau-Nord	rb_i._LKRS。 _Passau-Nord
RB Kaisersesch-Kaifenheim	rb_kaisersesch-Kaifenheim
RB Kemnather Land-Steinwald	rb_kemnather_land-Steinwald
RB Kitzinger Land	rb_kitzinger_Land
RB Kocher-Jagst	rb_kocher-Jagst
RB Mecklenburger Seenplatte	rb_mecklenburger_seenPlatte
RB Mittelrhein	rb_mittelRhein
RB 慕尼黑北部	rb_Munchen-Nord
RB Neumarkt I D Opf	rb_neumarkt_i_d_OPF
RB Neustadt-Vohenstraßt	rb_neustadt-vohenstraßt
RB Ochsenfurt	rb_Ochsenfurt
RB Pfaffenwinkel	rb_pfaffenWinkel
RB Rain am Lech	rb_rain_am_Lech
RB Roth-Schwabach	rb_roth-Schwabach
RB Stauden	rb_stauden
RB Sulzbach-Rosenberg	rb_Sulzbach-Rosenberg
RB Vilshofener Land	rb_vilshofener_Land
RB Weiden	rb_Weiden
RB Weilheim	rb_Weilheim
RB Weißenburg-Gunzenhausen	rb_weißenburg-gunzenhausen

名称	ID
RB Werratal-Landeck	rb_werratal-Landeck
RB östliche Südeifel	rb_östliche_südeifel
加拿大皇家银行 Centura Bank	RBC_centura_Bank
加拿大皇家银行	RBC_Royal_Bank
加拿大皇家银行财富管理	RBC_Wealth_Management
RBL 银行	RBL_bank
RBS	RBS
RCBC	RCBC
RCN	RCN
RD 石油公司	rd_Petrole
RD1	RD1
RDCC 银行	RDCC_BANK
RE/MAX	RE/MAX
真正的旅馆	REAL_INN
真实的生活保障	REAL_VIDA_SEGUROS
REAL-K	REAL-K
反叛运动	反叛运动
RECHEIO	RECHEIO
RECMAN	RECMAN
红色家族	RED_FAMILY

名称	ID
红太阳茶店	RED_SUN_TEA_SHOP
RED ZAC	RED_ZAC
REDDY Küchen	reddy_Küchen
REDHA AL-ANSARI 交易所	REDHA_AL-ANSARI_EXANGE
REDPACK	REDPACK
REEDS 珠宝商	Reeds_Jewelers
富豪宫殿酒店	REGAL_PALACE_HOTEL
御庭水疗度假村	Regalia_度假村及水疗中心
REGALIA 服务式公寓	REGALIA_服务式公寓
REGIO JÁTEK	REGIO_JÁTEK
REI	REI
REISS	REISS
罗莱夏朵精品酒店	RELAIS_&_CHATEAUX
放松模式	放松模式
中继	中继
RELO	RELO
移除	移除
REMOIL 市场	remoil_Marke
任仁乐	REN_REN_LE
雷诺	雷诺

名称	ID
雷诺租车	雷诺_租车
雷诺三星	雷诺_三星
RENERGIA	RENERGIA
里诺	里诺
回购店	回购店
REPSOL	REPSOL
共和国	共和国
预留	预留
休息汽车旅馆	REST_MOTEL
餐厅-酒吧	餐厅-酒吧
REV FCU	REV_FCU
REWE	REWE
REWE 即将上线	rewe_to_go
REY	REY
RGIS	RGIS
RHB 银行	RHB_BANK
REA	REA
戒指	戒指
戒指酒店	戒指酒店
RIU 酒店	RIU_Hotels

名称	ID
RIVELLI	RIVELLI
RM Garuda	rm_Garuda
RM Sari Bundo	rm_sari_bundo
RM Sari Minang	rm_sari_minang
RM Simpang Raya	rm_simpang_raya
RM 威廉姆斯	m_Williams
RMA Watanya	rma_watanya
RNKB	RNKB
路边烧烤炉	路屋烧烤
烤土豆	烤土豆
罗宾兹百货商店	ROBBINZ_百货商店
罗宾逊	罗宾逊
岩点	岩点
RODA	RODA
RODI 汽车服务	RODI_MOTOR_SERVICES
RODIER	RODIER
ROFU Kinderland	rofu_Kinderland
罗杰杜碧	ROGER_DUBUIS
劳力士	劳力士
滚筒	滚筒

名称	ID
劳斯莱斯	劳斯莱斯
罗曼的披萨	罗马披萨
罗曼蒂克	罗曼蒂克
ROOTT+CO	ROOTT+CO
ROSA	ROSA
罗斯代尔酒店	罗斯代尔酒店
ROSEWOOD A SENSE o	Rosewood_a_Sense_of_Place
ROSGOSSTRAKH 银行	ROSGOSSTRAKH_BANK
ROSKOSMOS 银行	ROSKOSMOS_BANK
罗斯曼	罗斯曼
ROSSOPOMODORO	ROSSOPOMODORO
桃红手枪	ROSE_PISTOL
ROTONDE	ROTONDE
ROY	ROY
罗伊·罗布森	ROY_ROBSON
皇家犬	ROYAL_CANIN
皇家农场	皇家农场
皇家超市	皇家超市
皇家郁金香	ROYAL_TULIP
ROaDy	ROaDy

名称	ID
RPM	RPM
RR 中心	RR_CENTER
RR Donnelley	rr_Donnelley
RRB-Bank	RRB-Bank
RSC 设备租赁	RSC_设备_租赁
RSEA 安全	rsea_safet
RSVP	RSVP
大润发	大润发
RTM 牙科	rtm_Dental
RTN FCU	RTN_FCU
RTV	RTV
RTV EURO AGD	RTV_EURO_AGD
RUCH	RUCH
RUFF	RUFF
RUMBO	RUMBO
RURALNOSTRA	RURALNOSTRA
RUSSLAVBANK	RUSSLAVBANK
RUSSTROYBANK	RUSSTROYBANK
RUSTA	RUSTA
RUSTICAE	RUSTICAE

名称	ID
RV Bank Rhein-Haardt	rv_bank_Rhein-Haardt
RVB Ebern	rvb_ebern
RVB Kronach-Ludwigsstadt	rvb_Kronach-LudwigsStadt
RVB Neuburg/Donau	rvb_neuburg/Donau
RW&CO。	RW&CO。
RWE	RWE
Raab Karcher	Raab_Karcher
Rabba	Rabba
拉巴尼	拉巴尼
荷兰合作银行	荷兰合作银行
Rabusch	Rabusch
RaceTrac	RaceTrac
赛道	赛道
Rack Room 鞋	Rack_Room _鞋子
机架	机架
Radabank	Radabank
拉德加斯特酒店	Radegast_Hotel
Radezh	Radezh
辐射的	辐射的
科拉西迪电台	Radio_Korasidi

名称	ID
电台租赁	电台租赁
Radioshack	Radioshack
雷迪森	雷迪森
丽笙酒店	Radisson_Blu
丽笙爱德华七世酒店	丽笙爱德华酒店
Radisson SAS 酒店及度假村	Radisson_sas_酒店及度假村
半径药房	Radius_Ph
拉杜加	拉杜加
拉弗蒂的	拉弗蒂的
莱佛士城	莱佛士城
莱佛士医疗集团	莱佛士医疗集团
拉菲基小额信贷银行	Rafiki_Microfinance_Bank
Ragazzo	Ragazzo
愤怒	愤怒
拉哈特·贝克斯	Rahat_Bakers
Rai Rai K	Rai_Rai_Ken
国际广播电台	国际广播电台
Raiffaizen Bank Aval	Raiffaizen_Bank_Aval
Raiffeisen	Raiffeisen
莱夫艾森银行	Raiffeisen_Bank

名称	ID
赖夫艾森波尔班克	Raiffeisen_Polbank
Raiffeisen Spar+Creditbank	Raiffeisen_Spar+Creditbank
Raiffeisen-Landesbank Steiermark	Raiffeisen-Landesbank_Steiermark
Raiffeisen-Landesbank 蒂罗尔	Raiffeisen-Landesbank_Tirol
Raffeisen-Vb Bad Staffelstein	Raiffeisen-Vb_Bad_Staffelstein
Raiffeisen-Vb Donawörth	Raiffeisen-Vb_Donawörth
Raiffeisen-Vb Fürth	Raiffeisen-Vb_Fürth
Raiffeisen-Vb Lichtenfels-Itzgrund	Raiffeisen-Vb_Lichtenfels-Itzgrund
Raiffeisen-Volksbank Ebersberg	Raiffeisen-Volksbank_Ebersberg
Raiffeisen-大众银行弗雷塞纳	Raiffeisen-Volksbank_Fesena
Raiffeisen-Volksbank Haßberge	Raiffeisen-Volksbank_Haßberge
Raiffeisen-Volksbank isen-Sempt	Raiffeisen-Volksbank_Isen-Sempt
赖夫艾森-大众银行里斯	Raiffeisen-Volksbank_ries
Raiffeisen-Volksbank Saale-Orla	Raiffeisen-Volksbank_Saale-Orla
Raiffeisenbank 奥格斯堡地西区	Raiffeisenbank_Augsburger_Land_West
Raiffeisenbank Bad Abbach-Saal	Raiffeisenbank_Bad_Abbach-Saal
布尔高 Raiffeisenbank	Raiffeisenbank_Burgau
Raiffeisenbank Donau-Heuberg	Raiffeisenbank_Dauna-Heuberg
Raiffeisenbank Ehingen-Hochsträße	Raiffeisenbank_Ehingen-Hochsträße
Raiffeisenbank Erkelenz	Raiffeisenbank_Erkelenz

名称	ID
Raiffeisenbank Grevenbroich	Raiffeisenbank_Grevenbroich
Raiffeisenbank Großenlüder	Raiffeisenbank_Großenluder
Raiffeisenbank 哈默尔堡	Raiffeisenbank_Hammelburg
海因斯贝格莱夫艾森银行	Raiffeisenbank_Heinsberg
Raiffeisenbank Isar-Loisachtal	Raiffeisenbank_Isar-Loisachtal
Raiffeisenbank Karlstadt-Gemunden	Raiffeisenbank_Karlstadt-Gemünden
肯普滕莱夫艾森银行	Raffeisenbank_Kempten
Raiffeisenbank Kirchweihtal	Raiffeisenbank_Kirchweihtal
莱夫艾森班克克雷斯凯尔海姆县	Raiffeisenbank_Kreis_Kelheim
Raiffeisenbank Krumbach	Raiffeisenbank_Krumbach
Raiffeisenbank Lohr am Main	raiffeisenbank_lohr_am_main
Raiffeisenbank Main-Spessart	Raiffeisenbank_Main-Spessart
莱夫艾森银行 Marktheidenfeld	Raiffeisenbank_Marktheidenf
Raiffeisenbank Oberallgäu-Sud	Raiffeisenbank_Oberallgäu-Sud
Raiffeisenbank 奥伯曼北部	Raiffeisenbank_Obermain_Nord
Raiffeisenbank Oberpfalz Süd	Raiffeisenbank_Oberpfalz_Süd
Raiffeisenbank Pfaffenhausen	Raiffeisenbank_Pfaffenhausen
Raiffeisenbank 莱茵-伯格	raiffeisenbank_Rhein-Berg
莱茵巴赫 Raiffeisenbank Voreifel	Raiffeisenbank_Rheinbach_Voreifel
Raiffeisenbank Risstal	Raiffeisenbank_Risstal

名称	ID
Raiffeisenbank Ronshausen-Marksuhl	Raiffeisenbank_Ronshausen-Marksuhl
罗斯多夫赖夫艾森银行	Raiffeisenbank_Rosdorf
Raiffeisenbank 罗图姆塔尔	Raiffeisenbank_Rottumtal
Raiffeisenbank Schwandorf-Nittenau	Raiffeisenbank_Schwandorf-Nittenau
赖夫艾森银行施特劳宾	赖夫艾森班克_施特劳宾
Raiffeisenbank Südstormarn Mölln	Raiffeisenbank_Südstormarn_Mölln
Raiffeisenbank Trostberg-Traunreut	Raiffeisenbank_Trostberg-Traunreut
Raiffeisenbank Tölzer Land	Raiffeisenbank_Tölzer_Land
Raiffeisenbank Volkach-Wiesentheid	Raiffeisenbank_Volkach-Wiesentheid
Raiffeisenbank Wesermarsch-Sud	Raiffeisenbank_Wesermarsch-Süd
Raiffeisenbank Westeifel	Raiffeisenbank_Westeifel
Allgauer Land 的 Raiffeisenbank	raiffeisenBank_im_allgauer_Land
卡尔县的 Raiffeisenbank	raiffeisenBank_im_kreis_calw
纳布塔尔的 Raiffeisenbank	raiffeisenBank_im_naabtal
高地的莱夫艾森班克	raiffeisenbank_im_Oberland
Stiftland 的 Raiffeisenbank	raiffeisenbank_im_Stiftland
Raiffeisenbank-Volksbank 米尔滕贝格	Raiffeisenbank-Volksbank_Miltenb
布尔根兰州莱夫艾森兰德斯银行	Raiffeisenlandesbank_Burgenland
Raiffeisenlandesbank Karnten	Raiffeisenlandesbank_Karnten
Nö-Wien Raiffeisenlandesbank	RaiffeisenlandesBank_nö-Wien

名称	ID
Raiffeisenlandesbank Oberösterreich	Raiffeisenlandesbank_Oberösterreich
福拉尔贝格州赖夫艾森兰德斯银行	Raiffeisenlandesbank_Forarlberg
萨尔茨堡莱菲森乐队	萨尔茨堡 Raiffeisenverband_
Raiffeisenvolksbank Gewerbebank	Raiffeisenvolksbank_Gewerbebank
铁路和工业 FCU	railroad__and__Industrial_FCU
彩虹	彩虹
彩虹百货	Rainbow_百货_商店
彩虹教育	彩虹_教育
彩虹帽	Rainbow_Hat
彩虹屋英语俱乐部	彩虹之家_英语_俱乐部
彩虹国际修复与清洁	Rainbow_International_修复_和_清洁
彩虹商店	彩虹商店
雨滴咖啡厅	Raindrops_Cafe
雷恩福德	雷恩福德
雨林咖啡厅	Rainforest Cafe
雨软	雨软
举起 Cane 的鸡手指	Raising_cane's_chicken_fingers
拉吉快车	Raj_Exp
拉贾	拉贾
拉贾斯坦邦 Marudhara Gramin 银行	拉贾斯坦邦_Marudhara_Gramin_Bank

名称	ID
拉贾斯坦邦合作银行	拉贾斯坦邦_州立合作社_银行
Rajdhani	Rajdhani
Rajkai Takarék	Rajkai_Takarek
拉杰曼迪尔大卖场	Rajmandir_大型超市
Rakamaz Takarék	Rakamaz_Takarék
Rakbankerd	Rakbankerd
Rakeru	Rakeru
Rakhat Lukum	Rakhat_Lukum
Raley's	Raley's
雷利药房	Raley's 药房
Ralf Ringer	Ralf_Ringer
拉力赛	拉力赛
Ralo 化妆品	Ralo_化妆品
拉尔夫的意大利冰	Ralph's_Italian_ices
Ralphs	Ralphs
Ralphs 燃料中心	Ralphs_Fuel_Center
拉尔夫斯药房	Ralphs_Phar
拉姆石油公司	Ram_Petrole
贾维斯华美达酒店	Ramada_Jarvis
Ramada-Treff	Ramada-Treff

名称	ID
罗摩衍那	罗摩衍那
Ramee	Ramee
拉面 Bankara	Ramen_Bankara
Rami Levi	Rami_Levi
Rammar	Rammar
横冲直撞	横冲直撞
Ramsey	Ramsey
拉姆斯托恩	拉姆斯托恩
跑了	跑了
Ranch 1	Ranch_1
兰德尔斯	兰德尔斯
任斯塔	任斯塔
户外游侠	游侠_户外
饶理工学院学院	rao_iit_Academy
快速自动	Rapid_Auto
快速贴合	Rapid_Fit
快速贴合和健康	Rapid_Fit_&_Well
快速 Pare-Brise	Rapid_Pare-Brise
Rapid's	Rapid_Robert's
Rapidito	Rapidito

名称	ID
Rapimueble	Rapimueble
Rapsodia	Rapsodia
Rarone	Rarone
Rasamas	Rasamas
rasMer	rasMer
Rasnerova pekarna	rasnerova_pekarna
Ratio	Ratio
Raumen Sugak	Raumen_Sugakiya
Rautakirja	Rautakirja
拉维交易所	Ravi Exchange
Ravinne 酒店及度假村	Ravinne_酒店_度假村
Rax	Rax
Raxevsky	Raxevsky
Ray Ban	Ray_Ban
雷·怀特	Ray_White
雷的美食广场	ray's Food_Place
Ray 的户外	Ray's 户外活动
Rayal Baby	Rayal_Baby
雷蒙德·詹姆斯金融服务	Raymond_James _金融_服务
Raymour & Flanigan Furniture	Raymour_&_Flanigan_Furniture

名称	ID
雷摩和弗兰尼根家具	RayMour_and_flannigan_Furniture
瑞途酒店及度假村	RayTour_酒店及度假村
Rayton-Lees Summit Comm	raytown-lees_summit_comm_cu
Raytsentr	Raytsentr
Raza Nostra Carnes y Brasas	raza_nostra_carnes_y_brasas
Raza Nostra Carnicerías	Raza_Nostra_Carnicerias
Razem	Razem
Razon's of Guagua	Razon's_of_Guagua
Razzle	Razzle
Razzoo's Cajun Caf	Razzoo's_cajun_cafe
重新存储	re_store
Re-Bath	Re-Bath
读男孩	Read_Boy
阅读电影院	Reading_Cinema
阅读英语	阅读_英语
就绪	就绪
Real Bank	Real_Bank
真正的加拿大超市	Real_Canadian_Superstor
房地产一号	Real_Estate_One
真正的辣椒粉	real_Paprika

名称	ID
RealTransfer	RealTransfer
真正打扫卫生	真的_打扫卫生
国际房地产高管协会	房地产_高管_国际
Rebel	Rebel
RechargeAccess	RechargeAccess
Recorcholis	Recorcholis
唱片库	唱片库
录制 Go	Record_Go
唱片小镇	Record Town
红色	红色
红苹果	红苹果
红地毯旅馆	Red_Carpet_Inn
红十字会	红十字会
Red Crown FCU	red_crown_fcu
Red D Mart	Red_D_Mart
红蜻蜓	Red_dragonfly
红热蓝色	红色_Hot_&_Blue
红莉亚鸡	Red_Lea_Chickens
红狮	Red_Lion
红龙虾	Red_Lobster

名称	ID
红芒果	Red_Mango
红色市场	红色市场
Red Oak	Red_Oak
Red One	red_One
红橙色	红色_橙色
红点租金	Red_Point_Rent
红丝带	红丝带
Red Robin	Red_Robin
红屋顶旅馆	红屋顶旅馆
Red Rooster	Red_Rooster
红点	Red_Spot
红场	红色正方形
繁文寡断	Red_Tape
红翼鞋	Red_Wing_Shoes
RedBrick 披萨	RedBrick_披萨
雷丹石油公司	Redan_Petrolem
Redbanc	Redbanc
雷迪快车	Reddy_Expr
Rede acaí.com	rede_açaí.com
Rede Econômia	rede_Econômia

名称	ID
Rede Leve Pizza pré-assada	rede_leve_pizza_pré-Assada
RedeMed	RedeMed
Redfarma	Redfarma
Redil	Redil
雷德蒙德	雷德蒙德
雷德纳的仓库市场	Redner's 仓库市场
红皮队	红皮队
红石美式烧烤	Redstone_American_Grill
redsun-iti	redsun-iti
Redtag	Redtag
红木	红木
锐步	锐步
Reece Plum	Reece_Plumb
礼德·史密斯	Reed_Smith
礁	礁
推荐房地产	Referral_Realty
炼油厂	炼油厂
炼油厂	炼油厂
改革与设计中心	Reform_&_Design_Center
Reformadísimo	Reformadísimo

名称	ID
改革之家	改革之家
Refosian	Refosian
加油	加油
帝王的	帝王的
Regal 电子消费品专卖店	Regal_消费电子产品专卖店
富豪电影院	Regal_Cinemas
富豪俱乐部	Regal_Club
Regal Nails	Regal_Nails
Regalitos	Regalitos
校务委员	校务委员
丽晶酒店及度假村	丽晶酒店及度假村
Reggie's	Reggie's
地区咖啡	地区_咖啡
澳大利亚地区银行	地区_澳大利亚_银行
区域 FCU	Regional_FCU
区域	区域
雷吉斯	雷吉斯
瑞吉斯沙龙	Regis_Salons
Regiustea	Regiustea
雷格斯	雷格斯

名称	ID
Rehion-Bank	Rehion-Bank
Rei do Mate	rei_do_mate
Rei dos Frangos	rei_dos_Frangos
Reid & Taylor	Reid_&_Taylor
Reifen-John	Reifen-John
Reiff	Reiff
Reisebank	Reisebank
Reiter Betten & Vorhánge	Reiter_Betten_&_Vorhánge
Reitmans	Reitmans
拒绝购物	拒绝_Shop
Relais Bleus	Relais_Bleus
放松背部	Relax_the_back
放松旅游	relax_turizem
Relaxo	Relaxo
信赖	信赖
信实资本	信实资本
信实数字	信赖_数字
信实足迹	信赖足迹
信实新鲜	Reliance_Fr
信实普通保险	信贷_一般_保险

名称	ID
信实珠宝	信仰_珠宝
信实购物中心	信心购物中心
信实市场	信赖市场
信实聪明	Reliance_Sm
信实证券	信贷_证券
信赖趋势	信赖趋势
信实网络世界	信赖_网络_世界
Reliant 社区 CU	reliant_Community_cu
Reliant FCU	reliant_fcu
Religare 健康保险	religare_健康_保险
依靠	依靠
Rema 1000	Rema_1000
Rema-Hotels	Rema-Hotels
补救措施智能人员配备	补救措施_智能_人员配备
Remicone	Remicone
Remke bigg's	remke_bigg's
Remzi Kitabevi	Remzi_Kitabevi
仁民同泰大药房	任敏同泰大药房
文艺复兴	文艺复兴
文艺复兴高管论坛	文艺复兴_高管_论坛

名称	ID
文艺复兴时期的斯特拉霍瓦尼	文艺复兴_Strakhovanie
华晨雷诺	雷诺_华晨
Rendez-Vous	Rendez-Vous
Rendichicas	Rendichicas
Renessans Credit	Renessans_Credit
仁和会计师事务所	仁和会计师事务所
伦纳	伦纳
Rennie's	Rennie's
里诺仓库	雷诺_仓库
租一辆自行车	租一辆自行车
租一辆车	租一辆车
租车 Kosmos	Rent_a_car_kosmos
租一个 Dent	Rent_a_dent
租房中心	租房中心
Rent-A-Wreck	Rent-A-Wreck
Rentamobile	Rentamobile
伦特莫尔克斯	伦特莫尔克斯
Rentway	Rentway
Renuar	Renuar
伦佐·科斯塔	Renzo_Costa

名称	ID
Reocar	Reocar
修理你的车辆	Repara_Tu_Vehiculo
Repco	Repco
Repco Bank	Repco_Bank
记者	记者
Repose Bakers	repose_Bakers
Repostar	Repostar
共和国银行和信托	共和国银行和信托银行
共和国服务	共和国_服务
救援 1122	救援_1122
救援 15	Rescue_15
Resene	Resene
印度储备银行	印度储备银行
原住客栈	Residence_Inn
SwissTouches 公寓酒店	Residences_Swisstouches
Reso-Garantiya	Reso-Garantiya
共振	共振
资源 One CU	resource_one_cu
尊重	尊重
休息舒适酒店	Rest_comfort_Hot

名称	ID
Restaumarché	Restaumarché
仙人掌餐厅	餐厅_仙人掌
Restaurant le Chesnoy S.a.	restaurant_le_chesnoy_s.a.
makro 餐厅	Restaurante_makro
Resthotel	Resthotel
Resto Ngalam	Resto_Ngalam
Resto Vanharte	Resto_Vanharte
Restoran Sederhana	Restoran_Sederhana
恢复硬件	恢复_硬件
还原	还原
Retitalia	Retitalia
Retoucherie de Manuela	Retoucherie_de_Manuela
Retravision	Retravision
Retz Butor	Retz_Butor
ReubenPower	ReubenPower
Revco 药店	Revco_Drug_Stores
Rever Pass	Rever_Pass
查看	查看
Revitex	Revitex
Revolta	Revolta

名称	ID
Revv	Revv
雷克斯	雷克斯
雷克萨尔	雷克萨尔
雷伊公司	Rey_Corporaci
雷耶斯烧烤	Reyes_Barback
雷森酒店	Rezen_Hotel
雷森度假酒店	Rezen_Retreat
Rezen 精选	Rezen_Select
Rhapsody's	Rhapsody's
拉萨	拉萨
Rheingauer Vb	Rheingauer_Vb
莱茵金	莱茵金
Rhino 现金和随身携带	Rhino_Cash_&_Carry
罗德岛 CU	Rhode_Island_cu
Rhodes 101 站	Rhodes_101_Stop
Rhomberg	Rhomberg
隆布斯国际酒店	Rhombus_国际_酒店
Rhön-Rennsteig-Sparkasse	Rhön-Rennsteig-Sparkasse
Ri Happ	Ri_Happy
Ria	Ria

名称	ID
Ria 汇款	里亚_汇款_转账
Riachuelo	Riachuelo
里亚德莫加多尔酒店	Riad_Mogador
rib City	Rib_City
Ribola	Ribola
排骨和汉堡	Ribs_&_Burgers
哥斯达黎加	哥斯达黎加
Ricami Veronica	Ricami_Veronica
里卡多·埃莱特罗	Ricardo_Eletro
饭碗	Rice_Bowl
Rich Health Check	Rich_Health_Check
理查兹	理查兹
里奇邦德	里奇邦德
Richeese 工厂	Richeese_Factor
更丰富的声音	更丰富的声音
里士满酒店	里士满酒店
里希特	里希特
日春茶	Richun_Tea
Ricker	Ricker
Ricky Sarkany	Ricky_Sarkany

名称	ID
Ricky's	Ricky's
瑞可乐园	Rico_Land
Rico's	Rico's
理光	理光
Ricoy	Ricoy
Riddle 的珠宝	Riddle's_Jewelry
在线拼车	拼车_Online
Ridgeway	Ridgeway
里奇伍德储蓄银行	Ridgewood_Savings_Bank
Rieker	Rieker
Rifarm	Rifarm
价格合适的瓷砖	合适的价格_瓷砖
就在家里	Right_at_Home
里格拉	里格拉
Rigoletto	Rigoletto
Rich	Rich
Ris Bilglass	Riis_Bilglass
Rikushet	Rikushet
Rileys	Rileys
里米	里米

名称	ID
瑞米大卖场	Rimi_大卖场
瑞米超市	里米_超市
Rimova	Rimova
Rincon Norteño	Rincon_Norteño
林戈	林戈
Ringpai 宠物医院	林派_宠物医院
里约	里约
里奥格兰德州立大学	rio_Grande_cu
防暴艺术与手工艺	Riot_Art_&_Craft
Rip Curl	Rip_Curl
瑞普·格里芬	Rip_Griffin
Ripley	Ripley
Rira Oil	Rira_Oil
Ris	Ris
崛起	崛起
瑞星酒店	Rising_Hotel
瑞星证券	上涨_证券
意大利调味饭混合物	Risotto_Mix
Ristoragip	Ristoragip
Risøe	Risøe

名称	ID
丽塔的	丽塔的
Rite Aid	Rite_Aid
Ritu kumar	ritu_Kumar
仪式	仪式
丽兹相机和影像	Ritz_Camera_&_Image
丽思卡尔顿	丽思卡尔顿
丽思卡尔顿俱乐部	丽思卡尔顿俱乐部
Riv'era	Riv'era
Rive Gauche	Rive_Gauche
河城银行	河城银行
河岛	河岛
河流地区 CU	River_region_cu
加州河谷	River_valley_cu
河谷社区 FCU	River_Valley_Community_FCU
伍兹河	里弗_伍兹
Riverland FCU	Riverland_FCU
河流	河流
河边 FCU	Riverside_FCU
里弗斯通	里弗斯通
Rivertown 社区 FCU	Rivertown_community_FCU

名称	ID
河景 CU Inc	Riverview_cu__inc
Riverways FCU	Riverways_fcu
里沃利酒店	里沃利_酒店
里文	里文
Rix	Rix
Rixos	Rixos
Rizap	Rizap
Rlc	Rlc
美国之路	美国之路
公路游侠	Road_Ranger
公路跑者体育	Road_Runner_Sports
路厨	路厨
Roadrunner	Roadrunner _市场
Roadway 快递公司	Roadway_Express_
Roady 的卡车停靠站	Roady's_truck_stops
烤牛肉 Hoshi	烤牛肉_Hoshi
Robbialac	Robbialac
Robeks	Robeks
罗贝尔	罗贝尔
Robert Gasparik-Masovyroba	Robert_Gasparik-Masovyroba

名称	ID
罗伯特·哈夫	罗伯特_哈夫
罗伯特·哈里斯咖啡厅	Robert_Harris Café
罗伯特的咖啡	罗伯特的咖啡
罗伯托·维里诺	Roberto_Verino
Roberto's Taco Shop	Roberto's_taco_Shop
罗伯茨石油	Roberts_Oil
罗宾油	Robin_Oil
罗宾·斯多宾	Robin_Sdobin
罗宾汉	罗宾汉
Robins FCU	robins_FCU
罗宾斯金融大学	robins_financial_c
罗宾斯厨房	罗宾斯厨房
罗宾逊家电	Robinsons_家电
罗宾逊银行	罗宾逊银行
罗宾逊百货公司	罗宾逊百货商店
罗宾逊超市	罗宾逊超市
Roborobo	Roborobo
机器人	机器人
Roby's	Roby's
洛克酒店	Roc_Hotels

名称	ID
罗氏诊断	罗氏诊断
罗切斯特	罗切斯特
Rochester 大码高大衣	Rochester_Big_&_Tall_Clothing
Rock & Ribs	Rock_&_Ribs
岩底	Bottom
火箭鸡	Rocket_Chicke
火箭城 FCU	Rocket_city_FCU
Rockfeller	Rockfeller
罗克福德·贝尔 CU	Rockford_bell_cu
Rockin'Jonny's Diner	Rockin'_johnny's_diner
Rockland 员工 FCU	Rockland_员工_FCU
罗克兰信托基金	罗克兰信托基金
Rockler 木工和五金	Rockler_woodworking_and_H
摇滚人	摇滚人
罗克波特	罗克波特
落基山巧克力工厂	洛基山巧克力工厂
落基山执法部门 FCU	Rocky_Mountain_Law_eforcement_F
洛基洛可可披萨和意大利面	Rocky_rococo_Pizza_and_Pasta
洛可可	洛可可
Rocomamas	Rocomamas

名称	ID
RodOil	RodOil
Rodd & Gunn	Rodd_&_Gunn
罗德威旅馆	Rodeway_Inn
罗迪	罗迪
罗迪拉	罗迪拉
罗迪齐奥	罗迪齐奥
Rodnaya Storona	Rodnaya_Storona
Rodnik Zdorov'ya	rodnik_zdorov'ya
罗多多火锅	Rododo_Hot_Pot
Rodovante	Rodovante
Rodovid Bank	Rodovid_Bank
Rogan 的鞋子	Rogan's_shoes
罗杰大卫	Roger_David
罗杰斯	罗杰斯
Rohan	Rohan
Rokn Zahrat Al Rawdah Pharamcy	Rokn_Zahrat_Al_Rawdah_Pharamcy
Roky's	Roky's
罗拉丁	罗拉丁
Roldão Atacadista	Roldão_Atacadista
Roll'd	Roll'd

名称	ID
Rollerz	Rollerz
罗林斯卡车租赁	罗林斯_卡车_出租
Rolo Banca 1473	Rolo_Banca_1473
Roly Poly	Roly_Poly
罗马	罗马
罗马披萨	Roma_Pizza
罗马的	罗马的
罗马原著	罗曼_原创作品
罗马纳市场	罗马市场
罗马尼亚-租车	罗马尼亚-租车
罗曼乔伊	罗曼乔伊
Romano's Macaroni Grill	Romano's_macaroni_Grill
罗曼蒂	罗曼蒂
Romantix	Romantix
Romashka	Romashka
罗梅拉	罗梅拉
Romp	Romp
Rompetro	Rompetro
罗马石油快车	Rompetrol_Expr
Romstal	Romstal

名称	ID
罗娜	罗娜
荣菜	荣_菜
荣记火锅	Rong_ji_Hot_Pot
荣百云酒店	荣百云_酒店
荣成老马	荣成老马
荣富佳岩	荣富佳岩
荣和村镇银行	荣和村镇银行
荣华药房	荣华_药店
Ronikon	Ronikon
roodstop	roodstop
室友	Room_Mate
罗曼科技	Roman_Technol
待去的房间	Rooms_To_Go
房间商店	房间商店
雄鸡	雄鸡
Roosters 男士美容中心	Roosters_men's_Grooming_Center
雄鸡之翼	雄鸡_翅膀
Rooter-Man	Rooter-Man
根	根
Roots 学校系统	根系_学校_系统

名称	ID
Ropharma	Ropharma
罗斯	罗斯
RosEvroBank	RosEvroBank
Rosa Cha	Rosa_Cha
罗莎的咖啡厅	Rosa's_cafe
Rosati's Pizza	Rosati's Pizza
罗斯班克	罗斯班克
玫瑰之家	玫瑰屋
玫瑰屋茶馆	玫瑰屋茶屋
玫瑰药房	玫瑰药房
罗森伯格	罗森伯格
罗森博格面包店	罗森博格_面包店
Rosenergobank	Rosenergobank
Rosense	Rosense
罗森塔尔	罗森塔尔
仅限玫瑰	仅限玫瑰
玫瑰	玫瑰
Rosgosstrakh	Rosgosstrakh
Roshan Bag 购物中心	Roshan_Bag_Mall
Roshdy	Roshdy

名称	ID
罗辛卡	罗辛卡
Rosinterbank	Rosinterbank
俄罗斯石油公司'	俄罗斯石油公司'
Rosneftegaz	Rosneftegaz
罗斯诺	罗斯诺
Rospechat'	Rospechat'
Ross 低价连衣裙	Ross_Dress_For_Less
罗斯-西蒙斯	罗斯-西蒙斯
Rossel'khozbank	Rossel'khozbank
Rossetti Deportes	Rossetti_Deportes
罗西尼	罗西尼
罗斯曼专卖店	Rossmann_Specialty_Store
罗索	罗索
Rossy	Rossy
罗斯特银行	Rost_Bank
Rostelekom	Rostelekom
Rosticerias Santo Gallo	Rosticerías_Santo_Gallo
肯德基 Rostiks	rostiks_KFC
Rostneft'	Rostneft'
罗塔纳	罗塔纳

名称	ID
Rotelli	Rotelli
罗斯	罗斯
Roti'O	Roti'O
Rotiboy	Rotiboy
Roto Rooter 管道和排水管服务	Roto_Rooter_Plumbing_&_Drain_Services
Rottaler Rb	Rottaler_Rb
Rottaler Volksbank-Raiffeisenbank	Rottaler_Volksbank-Raiffeisenbank
烂罗比	Rotten_Robbie
Rotter & Krauss	Rotter_&_Krauss
粗糙而甜美	粗糙&_Sweet
圆桌披萨	圆桌_披萨
劳斯市场	Rouses Market
流浪者	流浪者
罗兰兹药房	罗兰兹药房
Roxy	Roxy
罗伊·罗杰斯	Roy_Rogers
Roy's	Roy's
皇家联盟同事	Royal_Alliance_同事
皇家银行	皇家银行
加拿大皇家银行	加拿大皇家银行

名称	ID
皇家收藏	皇家系列
皇家美味面包店	皇家_美味_面包店
皇家恩菲尔德	皇家恩菲尔德
皇家集团酒店及汽车旅馆	皇家集团_酒店_汽车旅馆
Royal Hali	Royal_Hali
皇家家居中心	Royal_Homecenter
皇家 LePage	Royal_ LePage
皇家汽油	皇家石油公司
皇家体育馆	皇家体育馆
皇家茶	皇家茶
特许权使用费	特许权使用费
Roza Vetrov	Roza_Vetrov
Rubaiyat	Rubaiyat
Rubi 鞋子	Rubi_Shoes
魔方心理学家	Rubik_Psicologogogos
Rubik's Cube	Rubik's_cube
Rubio 的新鲜墨西哥烧烤	Rubio's _fresh_mexican_Gril
Rubis	Rubis
Rubl'Bum	Rubl'_Bum
Rublyovskiy	Rublyovskiy

名称	ID
露比酒店	Ruby_Hotels
红宝石星期二	Ruby_Tusday
Ruby's Diner	Ruby's_diner
Ruday	Ruday
如东农村商业银行	如东农村商业银行
Rudy's	Rudy's
鲁迪的德克萨斯州 Bar-B-Q	Rudy's_texas_bar-b-q
Ruedi Rüssel	Ruedi_Rüssel
Ruff 服装店	Ruff_Clothing_Store
Rufo's	Rufo's
RugDoctor	RugDoctor
如皋包商农村银行	如皋_包商_农村银行
坚固的熊	Rugged_Bear
林睿	Rui_Lin
瑞安农村商业银行	瑞安农村商业银行
瑞鹏宠物医院	瑞鹏宠物医院
瑞琪	瑞琪
瑞赛龙租车	瑞赛龙_租车
瑞源教育	瑞源_教育
Ruler	Ruler

名称	ID
伦比岛烧烤	Rumbi Island_Grill
Rumpold	Rumpold
在云上奔跑	Run_On_Clouds
跑步者学院	跑步者_学术界
跑步者积分	跑步者积分
RunnersWorld	RunnersWorld
跑步农场和舰队	Runnings_Farm_&_Fleet
Runza	Runza
卢比合作银行	Rupee_co_op_Bank
福建南安农村商业银行	福建南安农村通信银行
张家港农村商业银行	Rural_comm_bank_of _张家港
石狮农村商业银行	石狮市农村商业银行
盐城农村商业银行	盐城农村商业银行
永康农村商业银行	永康农村商业银行
农村信用合作社	农村_信用_合作社
河北省农村信用社	河北农村信用合作社
Rural Kutxa	Rural_Kutxa
Ruscobank	Ruscobank
Rusfinans Bank	俄罗斯金融银行
RussNeft'	RussNeft'

名称	ID
Russell & Bromley	Russell_&_Bromley
罗素和杰夫考特房地产经纪人	Russell_&_Jeffcoat_Realtors
Russell's Shop 4	Russell's_shop_4_you
罗素	罗素
俄罗斯地区开发银行	俄罗斯_地区_发展_银行
Russkaya Kukhnya	russkaya_Kukhnya
Russkiy Appetit	russkiy_Appetit
俄罗斯标准版	Russkiy_Standar
Russkiy khol	russky_kholod
Russkoe Zoloto	russkoe_Zoloto
鲁斯坦的超市	鲁斯坦的超市
Rustica	Rustica
Rusticae Metröpoli	Rusticae_Metröpoli
露丝的克里斯牛排馆	Ruth's_chris_Steak_House
Rutter's	Rutter's
如轩	如轩
如意芳	如意芳
Rx Plus	Rx_Plus
瑞安霍姆斯	Ryan_Homes
Ryan's	Ryan's

名称	ID
莱德卡车租赁和租赁	Ryder_truck_Rental_and_Leasing
Rydges	Rydges
Ryf Coiffeur	Ryf_Coiffeur
莱兰德家园	Ryland_Homes
Rylko	Rylko
莱曼	莱曼
文具师莱曼	Ryman_the_Stationer
良品酒店	良品酒店
自然保护区	RESERVE_NATURELLE
Radi Pekség	Radi_Pekség
大众电台	Radio_Popular
Ráj drogerie	ráj_drogerie
Ráde Bakeri	Ráde_Bakeri
Révalis Ever	Révalis_Ever
吕尔	吕尔
Rüsselsheimer Vb	Rüsselsheimer_Vb
S & S Food Store	S_&_S_Food_Store
S minimart	s_minimart
S&D	S&D
S&G	S&G

名称	ID
S&N 酒店	S&N_Hotel
标准普尔	标准普尔
S&R 会员购物	S&R_会员资格_购物
科技银行	S&T_Bank
S 市场	S 市场
S-Mart	S-Mart
S-Oil	S-Oil
S-Pankki	S-Pankki
S·奥利弗	S._Oliver
上午 73	S.A.M._73
S.T. DUPONT	S.T._DUPONT
S.Wear	S.Wear
SA NOSTRA	SA_NOSTRA
萨博	萨博
SABA 服装店	Saba_服装店
SABB	SABB
SABECO	SABECO
萨布兰卡	萨布兰卡
萨布丽娜	萨布丽娜
SACOOR	SACOOR

名称	ID
SADA	SADA
马鞍	马鞍
SADORENT	SADORENT
安全杯	SAFE_CU
SAFE FCU	SAFE_FCU
安全保险	安全保险
SAGARDI	SAGARDI
下垂	下垂
SAHAM 保险	Saham_Assur
撒哈拉 4X4	SAHARA_4X4
上汽的	上汽的
上汽通用汽车	SAIC_GM
SAIC LDV	SAIC_LDV
上汽大通	SAIC_MAXUS
上汽名爵	SAIC_MG
上汽荣威	上汽_荣威
圣戈班	圣戈班
荣寿司	日本酒寿司
SAKUB	SAKUB
火蜥蜴	火蜥蜴

名称	ID
萨拉瓦特	萨拉瓦特
桑巴舞	桑巴舞
SAMBA Moda	samba_moda
相同的	相同的
SAMPO	SAMPO
相同	相同
旧金山德阿西斯	SAN_FRANCISCO_DE_ASIS
三静	SAN_JING
圣玛丽娜	SAN_MARINA
圣罗克	SAN_ROQUE
SAN	SAN_
萨那	萨那
桑顿酒店	桑顿酒店
萨尼塔	萨尼塔
SANParks	SANParks
三丽欧	三丽欧
圣克鲁斯	SANTA_CRUZ
SANTABRASA	SANTABRASA
圣托里尼	圣托里尼
SAO	SAO

名称	ID
SAPELI	SAPELI
SAPIX	SAPIX
莎拉劳伦斯	莎拉·劳伦斯
SARAVANA BHAVAN	SARAVANA_BHAVAN
萨罗夫商业银行	萨罗夫商业银行
SASAM	SASAM
SASCH	SASCH
SASCO	SASCO
SASOL	SASOL
SATS	SATS
土星	土星
香肠沙龙	香肠_沙龙
保存毒品	SAVE_DRUG
SAXOO 伦敦	Saxoo_London
SB 酒店	SB_Hotels
SB-mobel Boss	sb-mobel_boss
SBARRO	SBARRO
俄罗斯联邦储蓄银行	俄罗斯联邦储蓄银行
SBI 证券	SBI_证券
SBM	SBM

名称	ID
SBS	SBS
SBT La Sabateria	sbt_la_sabateria
SBV	SBV
直播扫描	SCAN_LIVIN
斯堪尼亚	斯堪尼亚
SCANTEAK	SCANTEAK
SCANquilt	SCANquilt
SCAPA	SCAPA
SCAPINO	SCAPINO
Scarpe&Scarpe	Scarpe&Scarpe
SCB	SCB
渣打银行人寿保险	SCB_人寿保险
SCDCC 银行	SCDCC_BANK
SCE FCU	SCE_FCU
场景	场景
SCG	SCG
SCG 快车	SCG_Expres
SCHAFER	SCHAFER
SCHLECKER	SCHLECKER
施密特	施密特

名称	ID
SCHOENENREUS	SCHOENENREUS
学者酒店	学者_酒店
SCHOON	SCHOON
科学药学	科学药学
滑板车	滑板车
得分	得分
SCORE 导师 Bozeman	Score_mentors_Bozeman
SCORE 商店	Score_shop
斯科雷特	斯科雷特
SCOTMiD	SCOTMiD
斯科特的	斯科特的
斯科特的燃料中心	Scott's Fuel_Center
斯科特药房	斯科特药房
苏格兰房子	SCOTTISH_HOU
SDM-BANK	SDM-BANK
SE&PAS	SE&PAS
SE-ED 图书中心	se-ed_book_center
海鸥	海鸥
海边	海边
四季酒店	SEASONS_酒店

名称	ID
座位	座位
SEB	SEB
SECURITEST	SECURITEST
见 FAH	SEE_FAH
精工	精工
SELCO 社区 CU	selco_community_cu
SELEXION	SELEXION
SELGROS	SELGROS
Selgros Cash&Carry	Selgros_Cash&Carr
参议员	参议员
SENDO	SENDO
森亨	森亨
SEO	SEO
丝芙兰	丝芙兰
系列	系列
塞尔吉·布兰科	谢尔盖·布兰科
塞尔吉奥的	塞尔吉奥的
SERINCE CHIG KOFTECISI	SERINCE_CIG_KÖFTECISI
SERMSUK	SERMSUK
伺服专业版	伺服专业版

名称	ID
SES	SES
SESI Farmácia	sesi_farmácia
套装连锁酒店	SET_CHAIN_HOTEL
SETTE	SETTE
SEUR	SEUR
七茶	七_茶
七辆巴士	七辆巴士
SEVENOaKS 声音与视觉	SEVENOaKS_SOUND_&_Vision
SF	SF
顺丰速运	顺丰快递
SGB Spoldzielcza Grupa Bankowa	sgb_spoldzielcza_grupa_bankowa
SGBE	SGBE
SGBCI	SGBCI
SGBG	SGBG
SGBS	SGBS
SGC	SGC
SGM	SGM
SGMW	SGMW
SGO 设计师玻璃	SGO_designer_Glass
SH 酒店	SH_Hoteles

名称	ID
SHACMAN	SHACMAN
山东航空	山东航空
尚衡	尚恒
上海商业银行	上海商业银行
上海敏	上海_分钟
上海证券	上海证券
山西证券	山西证券
SHANYON SPOR	SANYON_SPORTS
少冰	少冰
分享茶	分享_TEA
SHAZAM	SHAZAM
SHB	SHB
盛京银行	盛京银行
申银万国证券	申银万国证券
深航精品店	深圳航空精品店
深航国际	深圳航空_国际
宣威-威廉姆斯	宣威-威廉姆斯
宣伟-威廉姆斯地板覆盖物	Sherwin-Williams_Floor
世茂广场	世茂广场
SHINDOM 旅馆	SHINDOM_INN

名称	ID
SHINFANG 123	SHINFANG_123
新韩银行	新韩银行
闪亮的茶	闪亮的茶
船舶	船舶
SHIRBLE DEPARTMENT	SHIRBLE_DEFARTMENT_STORE
鞋城	鞋城
鞋市	SHOE_MART
鞋子	鞋子
商店	商店
购物 N 保存	SHOP_N_SAVE
选一个	SHOP_ONE
商店狗	商店狗
购物者停下来	SHOPPERS_STOP
肩膀	肩膀
演出时间	SHOW_TIME
该死	该死
SHUBH 珠宝商	shubh_Jewelers
书友	书友
四川汇关	SI_CHUAN_HUI_GUAN
SI 市场	SI_Marke

名称	ID
暹罗商业银行	暹罗商业银行
四川航空	四川航空公司
SICOOB	SICOOB
SIGMA Bakeries	Sigma_Bakeries
SIGNARAMA	SIGNARAMA
招牌酒店	特色酒店
真丝和羊绒	真丝_&_羊绒
丝绸的	丝绸的
SILMAYASEMA	SILMAYASEMA
西尔文	西尔文
西尔弗顿散热器	SILVERTON_散热器
SIMAL	SIMAL
西蒙斯	西蒙斯
SIMPATIA CRAI	SIMPATIA_CRAI
SIMPLE	SIMPLE
简单地	简单地
只是汉堡	SIMPLY_BURG
简单基本	简单_基本
SIMPLY	简单城市
只是市场	简单_市场

名称	ID
SINOIL	SINOIL
中国重汽	中国重汽
SIP Abacus	SIP_Abacus
先生	先生
SISA	SISA
希思黎	希思黎
SISLI OPTIK	SISLI_OPTIK
姐妹	姐妹
SIU CU	SIU_CU
SIXDO	SIXDO
SIXT	SIXT
SJC	SJC
SK	SK
SKB 银行	skb_Banka
SKB-Bank	SKB-Bank
SKECHERS	SKECHERS
斯凯奇工厂直销店	斯凯彻斯工厂_奥特莱斯
SKEIDAR	SKEIDAR
草图	草图
素描师	素描师

名称	ID
技能语言	Skill_idiomas
斯柯达	斯柯达
SKOK Arka	skok_arka
SKOK Jaworzno	skok_jaworzno
SKOK Jowisz	skok_jowisz
SKOK Kujawiak	skok_kujawiak
SKOK 耐克	skok_Nike
SKOK Piast	skok_piast
SKOK Rafineria	skok_rafineria
SKOK Silesia-Centrum	skok_Silesia-Centrum
SKOK Skarbiec	skok_skarbiec
SKOK Slask	skok_slask
SKOK Wisla	skok_wisla
SKOK Wybrzeze	skok_wybrzeze
SKOK Ziemi Rybnickiej	skok_ziemi_rybnickiej
SOKK 我是。E. Kwiatkowskiego	skok_im。 _E._Kwiatkowskiego
SOKK 我是。Pwostancow Slaskich	skok_im。 _Pwostancow_Slaskich
SOKK 我是。Unii Lubelskiej	skok_im。 _Unii_Lubelskiej
SOKK 我是。Zygmunta Chmielewskiego	skok_im。 _zygmunta_Chmielewskiego
SOKK im. kard。Stefana Wyszynskiego	skok_im。 _kard。 _Stefana_Wyszynskiego

名称	ID
SKOKANONEN	SKOKANONEN
SKP	SKP
SKRIBO	SKRIBO
SLO	SLO
SM 设备中心	SM_Appliance_Cent
SM 百货商店	SM_百货商店
SM 大卖场	SM_大卖场
SM 超级购物中心	SM_超级购物中心
SM 超市	SM_Sump
SMATCH	SMATCH
SMBC	SMBC
三井住友银行日兴证券	三井住友银行_日兴证券
SMC	SMC
SMC 环球证券	SMC_环球_证券
中小企业银行	SME_bank
泰国中小企业银行	泰国中小企业银行
SMOOKE.PL	SMOOKE.PL
SMP 银行	SMP_Bank
SMYCKA	SMYCKA
SNG	SNG

名称	ID
SNS 银行	SNS_Bank
太简单了	太简单了
太棒了	太棒了
所以！咖啡	所以！ _咖啡
SOCAR	SOCAR
法国兴业银行	SOCIETE_GENERALE_DE_BANQUE
黑山兴业银行	SOCIETE_GENERALE_MONEGEN
塞尔维亚兴业银行	SOCIETE_GENERALE_SRBIJA
国家保险公司	SOCIETE_NATIONALE_D'ASSUR
索菲特	索菲特
SOHO 酒吧	SOHO_BAR
SOKIMEX	SOKIMEX
索科洛夫	索科洛夫
SOKOS	SOKOS
索科斯酒店	SOKOS_HOTELS
SOLARIA	SOLARIA
索拉托	索拉托
个体经营者	独家交易者
SOLMAR	SOLMAR
SOMAP	SOMAP

名称	ID
SOMAREC	SOMAREC
SOMEPI	SOMEPI
萨默塞特服务公寓	萨默塞特服务式公寓
声纳	声纳
SONIC	SONIC
索诺拉	索诺拉
索尼	索尼
东吴证券	SOOCHOW_SECURITIES
SOPAS Y POSTRES	SOPAS_Y_POSTRES
索菲亚	索菲亚
SOPOR	SOPOR
SORELLE SR RAMONDA	SORELLE_SR_RAMONDA
SOTC	SOTC
索特卡	索特卡
南方美女	南方美女
南方记忆	南方记忆
南街汉堡	SOUTH_ST_BURGER
东南保险	东南保险
东南人寿保险	东南人寿保险
西南证券	西南证券

名称	ID
联盟号	联盟号
SP	SP
SP 西西里能源	sp_energia_siciliana
SP+ 停车场	SP+_parking
太空世界	太空世界
SPAL	SPAL
SPAQ	SPAQ
SPAR	SPAR
SPAR 杂货店	SPAR_GROCERY_ST
SPAR 快递	Spar_Expr
SPARBANKEN FINN	SPARBANKEN_FINN
SPARBANKEN 厄勒海峡	SPARBANKEN_ÖRESUND
马其顿储蓄银行	Sparkasse_bank_makedonja
斯巴达商店	SPARTAN_STORES
SPB	SPB
SPC	SPC
SPC加油站	SPC_Petrol_Station
SPCA	SPCA
专业运动器材	专业运动器材
速度	速度

名称	ID
SPETSSET 的 STROYBANK	SPETSSET 的 STROYBANK
狮身人面像	狮身人面像
SPI 电影院	SPI_电影院
辣奶奶	SPICY_奶奶
辛辣 VE 中国菜	Spicy_ve_Chinese
间谍	间谍
SPITZ	SPITZ
SPLITSKA BANKA	SPLITSKA_BANKA
运动视觉	运动视觉
运动区	运动专区
SPORTISIMO	SPORTISIMO
体育链接	体育链接
春秋航空	春季航空
斯普林菲尔	斯普林菲尔
SPRINO 茶	SPRINO_TEA
SPX	SPX
挤压棒	挤压棒
SRS 旅行社	srs_Trav
SRS 电影院	SRS_电影院
SS3C	SS3C

名称	ID
SSAW 酒店	ssaw_Hotel
SSJD 牛排自助餐	SSJD_STEAK_BUFFET
ssk-supersKunk	ssk-supersKunk
SSM 医疗保健	SSM_医疗保健
SSS SEADECHASEADER	SSS_SUDEANCHEASE
SST&C	SST&C
ST。REGIS	ST。 _REGIS
STA旅行	sta_Travel
体育场	体育场
STANDA	STANDA
星际集市	Star_Bazaar
明星健身房	STAR_GYM
星际洗礼	STAR_WASH
星汉堡	星汉堡
STARCAR	STARCAR
明显	明显
STARKKI	STARKKI
STAROIL	STAROIL
STARWASH	STARWASH
星程酒店	STARWAY_HOTEL

名称	ID
星程尊享版	Starway_Premi
住酒店	STAY_HOTELS
STB 银行	STB_BANK
STC	STC
蒸过的菜肴	蒸过的菜肴
STEFANEL	STEFANEL
STERNS	STERNS
STESEN MINYAK	STESEN_MINYAK
STF	STF
STJÄRNURMAKARNA	STJÄRNURMAKARNA
STM 摩托车	STM_motocicletas
石头	石头
STONEFLY	STONEFLY
停止并清洗	停止并清洗
STORNAWAY	STORNAWAY
食物的故事	食物的故事
紧张	紧张
中风	中风
结构管	结构管
STS 轮胎和汽车中心	STS_tire_&_auto_Centers

名称	ID
斯图尔特·韦茨曼	STUART_WEITZMAN
学生	学生
STUSID BANK	STUSID_BANK
斯巴鲁	斯巴鲁
地铁	地铁
SUGUO	SUGUO
西装选择	西装_选择
西装和西装	西装_&_西装
SUKIYA	SUKIYA
SUM	SUM
SUMGO 茶	SUMGO_TEA
阳光和沙滩运动	SUN_SAND_SPORTS
太阳很远	SUN_FAR
日月酒店	SUN_MOON_HOTEL
日月旅馆	SUN_MOON_INN
太阳商店	SUN_STORE
SUN1	SUN1
SUNINEN	SUNINEN
晴朗的天空	SUNNY_SKY
SUNPHARMA	SUNPHARMA

名称	ID
日落套装	日落套装
阳光厨房	阳光厨房
SUOMALAINEN KIRJAKAUPPA	SUOMALAINEN_KIRJAKAUPPA
超级剪裁	超级剪裁
超级旧金山德阿西斯	SUPER_SAN FRANCISCO_DE_ASIS
超级明星海鲜餐厅	超级明星海鲜餐厅
超级动物园	超级动物园
超级运动	超级体育
超级药房	超级药房
SUPERALBA	SUPERALBA
超级蜜蜂	超级蜜蜂
超级核心	超级核心
SUPERCOR Stop&G	Supercor_stop&Go
SUPERCOR 快车	Supercor_exprés
SUPERMERCADOS SANCHEZ ROMERO	SUPERMERCADOS_SANCHEZ_ROMERO
SUPERMERCATS 适用	Supermercats_aprop
超级晶石	超级晶石
超级商店	超级商店
苏利亚	苏利亚
SURTIMAX	SURTIMAX

名称	ID
SUSCO	SUSCO
寿司明泰	SUSHI_MENTAI
寿司外卖	SUSHI_TAKEOUT
SUSHILOKO	SUSHILOKO
苏文	苏文
铃木	铃木
铃木咖啡公司	Suzuki_cafe_公司
斯文森	斯文森
SVG Fuel	SVG_fuel
SVIJET OBUCE	SVIJET_OBUCE
SWA Rad	swa_rad
SWARZEDZ	SWARZEDZ
发誓	发誓
瑞兴面包店	SWEE_HENG_BA
SWEETTALK	SWEETTALK
SWJ	SWJ
SWM	SWM
SYM	SYM
SYN MUN KONG 保险	SYN_MUN_KONG_INSURAN
同义词	同义词

名称	ID
SYSTEMBOLAGET	SYSTEMBOLAGET
SZUZ SZOP	SZUZ_SZOP
SaSa	SaSa
Saad Al-Abdullah 城市合作社协会	Saad_al-Abdullah_city_coop_Society
Saalesparkasse	Saalesparkasse
萨瓦德尔	萨瓦德尔
SabadellAtl安蒂科	SabadellAtl安蒂科
SabadellCam	SabadellCam
SabadellGallego	SabadellGallego
SabadellGuipuzcoano	SabadellGuipuzcoano
SabadellHerrero	SabadellHerrero
SabadellSolbank	SabadellSolbank
SabadellUrquijo	SabadellUrquijo
Sabah Al-Salem 合作社	Sabah_al-Salem_Co-op_Society
Sabarmati Gas	Sabarmati_Gas
Sabin Medicina Diagnóstica	Sabin_Medicina_Diagnóstica
萨宾娜	萨宾娜
Sabine FCU	sabine_fcu
萨布卡集市	Sabka_Bazaar
黑貂糖果	Sable_Swe

名称	ID
Sabon	Sabon
Saboten	Saboten
Sabri Ozel	Sabri_Ozel
SabroPollo	SabroPollo
Sabyasachi	Sabyasachi
Sac d'Anvers	Sac_d'Anvers
SacDuCo	SacDuCo
Saco Valley CU	saco_valley_cu
Sacombank	Sacombank
萨库尔兄弟	Sacoor_Brothers
萨达什	萨达什
SafePark	SafePark
Safeco 保险	安全生态保险
保障	保障
保护自助存储	Sefud_Self_Storag
Safelite AutoGlass	Safelite_ AutoGlass
SafetyKleen	SafetyKleen
安全之路	安全之路
西夫威药房	Safeway_药房
藏红花	藏红花

名称	ID
Safra	Safra
佐贺化妆品	Saga_化妆品
Saga Falabella	Saga_Falabella
相模信用金库	相模信用金库
萨加尔	萨加尔
Sagar Ratna	Sagar_Ratna
佐川急便	佐川快车
Sage 软件	Sage_软件
Sahakari Bhandar	Sahakari_Bhandar
Sahan	Sahan
撒哈拉	撒哈拉
撒哈拉之星	撒哈拉之星
萨赫勒	萨赫勒
Sahel Mart	Sahel_Mart
萨赫勒-撒哈拉银行	萨赫勒-撒哈拉银行
Sahyadri 医院	Sahyadri_医院
Saib	Saib
上汽红岩	上汽_红岩
上汽斯柯达	上汽_斯柯达
上汽大众	上汽_大众

名称	ID
西贡石油	Saigon_Petro
西贡银行	西贡银行
西京信用金库	西京信用金库
塞恩斯伯里的	塞恩斯伯里的
塞恩斯伯里本地人	Sainsbury's_Local
Saint Honore 蛋糕店	Saint_Honore_Cake_Shop
圣马克鲁	圣马克卢
圣特罗佩	圣特罗佩
Saint-Algue 发型	Saint-Algue_Coiffure
埼玉县信用金库	埼玉县信用金库
Saizeriya	Saizeriya
Sajiao Chilli 餐厅	三角_辣椒餐厅
Sajövölgye Takarék	Sajövölgye_Takarék
酒井咖啡	酒井_咖啡
清酒之杉	清酒之杉
Sakhaneftegazsbyt	Sakhaneftegazsbyt
萨克斯第五大道	萨克斯_第五大道
Saks 第五大道 5 号旁边	saks_fifth_Avenue_off_5th
Saksiam Panit 租赁	Saksiam_Panit_Leasing
萨昆达能源	萨昆达_能源

名称	ID
樱花	樱花
樱花日语	樱花_日语
樱花厨房生活	樱花_厨房_生活
Sakura Suisan	Sakura_Suisan
沙拉创作	沙拉_创作
沙拉旋转器	Salad_Spinners
沙拉故事	Salad_Story
Saladworks	Saladworks
萨拉芬	萨拉芬
萨拉马	萨拉马
Salarpuria Sattva	Salarpuria_Sattva
Salcobrand	Salcobrand
加那利群岛余额	Saldos_Canarias
出售	出售
Saleen	Saleen
塞勒姆区中央合作银行	Salem_district_Central_Co-op_Ban
塞勒姆五号银行	Salem_Five_Bank
Salfer	Salfer
萨利纳斯	萨利纳斯
索尔兹伯里	索尔兹伯里故居

名称	ID
Salitur	Salitur
Sally Beauty	Sally_Beauty_Suppl
Salmiya 合作社	Salmiya_合作社_协会
所罗门	所罗门
以沙龙为中心	以沙龙为中心
沙龙 Chasov 3-15	Salon_Chasov_3-15
沙龙 Tony	Salon_Tony
莎莎舞	莎莎舞
Salsa's	Salsa's
Salsarita 的新鲜小酒馆	Salsarita's _fresh_cantina
Saltgrass 牛排馆	Saltgrass_Steak_House
救世军	救世_军队
萨尔瓦多	萨尔瓦多
萨尔瓦多·菲拉格慕	Salvatore_Ferragamo
萨尔瓦多的老式比萨店	Salvatore's_old_fashioned_Pizzeria
Salvos 商店	Salvos_Stores
Salzlandsparkasse	Salzlandsparkasse
Sam 0-13	Sam_0-13
Sam Ash 音乐商店	Sam_Ash_Music_Store
Sam Goody	Sam_Goody

名称	ID
山姆 McCauley	山姆_ McCauley
山姆的鸡	Sam's Chicken
山姆俱乐部	Sam's Club
山姆俱乐部加油站	Sam's Club_gas
山姆食品店	Sam's Food_Stores
Sam's Mart	Sam's_mart
山姆的披萨	Sam's Pizza
山姆的南方餐厅	Sam's_southern_eatery
山姆的仓库	山姆的仓库
Samaranefteprodukt	Samaranefteprodukt
当天性病检测	当天_STD_测试
萨米尔油	Samir_Oil
Samkaup	Samkaup
Samoborska banka	Samoborska_Banka
Samokhval	Samokhval
萨摩斯卡	萨摩斯卡
桑皮	桑皮
Sampiyon Kokoreq	Sampiyon_Kokoreq
Samson-Farma	Samson-Farma
Samsonite	Samsonite

名称	ID
三星	三星
塞缪尔·曼奇诺的	Samuel_Mancino's
SamunLisa	SamunLisa
Samura	Samura
Samurai Sam's	Samurai_Sam's
圣安东尼奥	圣安东尼奥
圣安东尼奥咖啡厅	San_Antonio_Café
圣卡罗	San_Carlo
San Churro	San_Churro
圣地亚哥县信用合作社	圣地亚哥县信用联盟
旧金山咖啡	旧金山咖啡
旧金山消防队	旧金山_fire_cu
旧金山汤业公司	旧金山汤公司
圣路易斯	San_Luiz
San Ma	San_Ma
三妙超市有限公司	San_Miu_超市_有限公司
圣巴勃罗药房	San_pablo_farmacia
三珠山银行	San_ju_san_Bank
Sana Safinaz	Sana_Safinaz
萨纳贝尔·萨拉姆糖果	Sanabel_Al_Salam_Sweets

名称	ID
Sanad 保险	萨纳德保险
萨纳尔	萨纳尔
Sanborns	Sanborns
Sandalyeci	Sandalyeci
桑德拉的	桑德拉的
桑德斯	桑德斯
桑德勒训练	桑德勒训练
睡魔	睡魔
桑多斯和马康福特酒店	Sandos_&_Marconfort_Hotels
三明治大亨	三明治_Baron
三明治点	三明治_Spot
桑迪斯普林银行	桑迪斯普林_班克
Sandys	Sandys
Sanesco	Sanesco
Sangeetha Mobiles	Sangeetha_Mobiles
Sangem 国际酒店及度假村	Sangem_Int'l_Hotels_&_resorts
Sangi Stil'	Sangi_Stil'
Sangkaya	Sangkaya
桑利合作银行	sangli_co-op_Bank
Sanitas	Sanitas

名称	ID
理智	理智
三江购物	三江_购物
三条信用金库	三条信用金库
Sankalp	Sankalp
圣盖勒州立银行	Sankt_Gallery_KantonalBank
三龙加油站	三龙加油站
三门农村商业银行	三门农村商业银行
Sannegardens 披萨店	Sannegardens_Pizzeria
三品旺	三品旺
Sanpomen	Sanpomen
三仁兴古头王火锅	三人行_谷头王_火锅
三星堂商店	Sanseido_Shoten
圣埃琳娜	圣埃琳娜
圣诞老人	圣诞老人
圣塔芭芭拉银行和信托基金	Santa_Barbara_Bank_&_Trust
圣伊莎贝尔	Santa_Isabel
圣诞老人萝拉	Santa_Lolla
圣卢西亚	Santa_Luzia
圣达菲牛排	Santa_fe_Steak
圣塔格洛里亚	圣塔格洛里亚

名称	ID
桑塔纳	桑塔纳
桑坦德	桑坦德
桑坦德消费者银行	桑坦德消费者银行
西班牙桑坦德中央银行	Santander_Central_Hispano
桑坦德托塔	Santander_Totta
圣达菲市场	Sante_Fe_Market
圣地亚哥的	圣地亚哥的
Santoni	Santoni
Santory	Santory
三万昌	三万昌
三香人家	三香人家
三一重工	三一重工
萨普纳书屋	sapna_book_House
札幌拉面 Dosanko	札幌_拉面_道参香
Saraiva	Saraiva
萨兰斯克 Pechat'	Saransk_Pechat'
萨拉尔	萨拉尔
萨拉斯	萨拉斯
萨拉斯能源	Saras_Energia
萨拉斯瓦特银行	Saraswat_Bank

名称	ID
Saray Hali	Saray_Hali
Saray Mobilya	Saray_Mobilya
Saray Muhallebicisi	Saray_Muhallebicisi
Saray Pen	Saray_Pen
Sarcletti	Sarcletti
萨甘电子	Sargam_电子产品
Sariyer Börekçisi	Sariyer_Börekçisi
Sarku 日本	Sarku_Japan
Sarni	Sarni
Saroil	Saroil
萨罗瓦尔酒店及度假村	萨罗瓦尔_酒店及度假村
Sarpino's Pizzeria	Sarpino's Pizzeria
Sarva Haryana Gramin 银行	Sarva_Haryana_Gramin_Bank
Sarva U.P.Gramin 银行	Sarva_U.P.Gramin_Bank
Sasco 餐厅	Sasco 餐厅
Sass & Bide	Sass_&_Bide
Sastasundar 健康与幸福	SastaSundar_Health_&_Happiness
萨塔拉区中央合作银行	Satara_district_Central_Co-op_B
沙爹王	Satay_King
Sate Khas Senayan	Sate_Khas_Senayan

名称	ID
Satinni	Satinni
佐藤町	佐藤町
Satoriz	Satoriz
萨特拉	萨特拉
萨特拉食品	Satra_Foods
Satra Mart	Satra_Mart
萨蒂亚集团	Satya_Group
萨蒂亚姆	萨蒂亚姆
沙特阿拉伯航空	沙特_阿拉伯航空公司
沙特信贷和储蓄银行	Saudi_credit_&_Saving_Bank
沙特燃料	Saudi_Fuel
沙特黎巴嫩银行	沙特_黎巴嫩_银行
沙特国家银行	沙特国民银行
索尔斯	索尔斯
Saurina	Saurina
Sav-A-Stop	Sav-A-Stop
Sav-Mor Foods	Sav-Mor_Foods
Sav-O-Mat	Sav-O-Mat
Sav-On 药房	Sav-on_Pharmacy
萨瓦酒店及度假村	Sava_酒店及度假村

名称	ID
Savage	Savage
Savaria Takarék	Savaria_Takarék
Save（保存）	Save（保存）
拯救 Mart	Save_Mart
保存更多	保存_更多
Save-A-Lot	Save-A-Lot
SaveMart	SaveMart
Savegnago	Savegnago
Savells	Savells
Savells Fairdeal	Savells_Fairdeal
Savemore	Savemore
储户	储户
萨维尔	萨维尔
品尝咖啡	Savor_coffee
Savory	Savory
萨沃伊	萨沃伊
泽屋信用金库	Sawayaka_Shinkin_Bank
Sayers	Sayers
Saypa	Saypa
Saúde e Bem Estar	saude_e_bem_estar

名称	ID
Sberegayka	Sberegayka
SC	SC
Scala	Scala
斯卡利	斯卡利
ScanBurger	ScanBurger
Scandic	Scandic
斯堪的纳维亚人	斯堪的纳维亚人
Scappino	Scappino
Scartezini	Scartezini
风景社区 CU	Scenic_Community_cu
Schaffhauser Kantonalbank	Schaffhauser_KantonalBank
沙弗拉特	沙弗拉特
Schakolad 巧克力工厂	Schakolad_Chocolate_Factory
Schanzenbackerei	Schanzenbackerei
Scheer & Foppen	Scheer_&_Foppen
Schewels	Schewels
辛德勒外语学校	辛德勒_外语_学校
迅达集团	Schindler_Group
施莱珀	施莱珀
Schlemmermeyer	Schlemmermeyer

名称	ID
Schlotzsky's	Schlotzsky's
SchneiderElectric	SchneiderElectric
施尼茨	施尼茨
施尼策尔豪斯	施尼策尔豪斯
Schnitzlplatzl	Schnitzlplatzl
Schnucks	Schnucks
Schoellerbank	Schoellerbank
学者教育	学者_教育
学者精品酒店	学者_精品_酒店
Scholl	Scholl
学校食品	学校_食品
学校系统 FCU	School_systems_FCU
SchoolFirst 联邦信用合作社	SchoolFirst_Federal_Credit_Union
Schoop's 汉堡包	Schoop's 汉堡包
Schrader	Schrader
舒伯特	舒伯特
Schuck 的汽车补给	Schuck's auto_Supply
Schuhhaus Klauser	Schuhhaus_Klauser
Schuhhaus Tiefenbacher	Schuhhaus_Tiefenbacher
Schuhhof	Schuhhof

名称	ID
舒马赫家园	Schumacher_Homes
Schutz	Schutz
Schuurman Schoenen	Schuurman_Schoenen
Schwan's	Schwan's
Schwyzer Kantonalbank	Schwyzer_Kantonalbank
Schwébisch Hall	Schwebisch_Hall
Schäfer's	Schäfer's
Schöps	Schöps
Scient FCU	Scient_fcu
嫩芽	嫩芽
Scipioni	Scipioni
Informatique	独家新闻_信息学
Scooter 的咖啡屋	Scooter's CoffeeHouse
踏板车披萨	Scooters_Pizza
分数超市	分数_超市
Scores 烤肉店	Scores_Rotisserie
苏格兰人	苏格兰人
苏格兰威士忌和苏打水	苏格兰威士忌_&_苏打水
Scotchman	Scotchman
丰业银行	丰业银行

名称	ID
丰业银行经纪人	丰业银行经纪人
苏格兰受到表彰	苏格兰_推荐
Scotta 1985	Scotta_1985
Scottage	Scottage
苏格兰燃料	苏格兰_燃料
苏格兰高地	苏格兰高地
苏格兰水电公司	苏格兰水力发电公司
苏格兰旅馆	Scottish_Inn
苏格兰和纽卡斯尔	苏格兰_和_纽卡斯尔
Scotts LawnService	Scotts_ LawnService
争夺者	争夺者
ScreenMobile	ScreenMobile
螺丝固定	螺丝固定
Scrivens 配镜师	抄写员_配镜师
磨砂膏及其他	Scrubs_&_Beyond
Se ABank	Se ABank
见 RGe NT Major	参见 RGe nt_major
Sea Air FCU	sea_air_FCU
海百合酒店	Sea_Lily_Hotel
海油	Sea_Oil

名称	ID
海凤凰酒店	海凤凰酒店
Seaboard FCU	Seaboard_FCU
海岸国家银行	海岸_国民银行
海鲜城超市	海鲜城_超市
海罩	海罩
西兰证券	海陆证券
海港餐厅	海港餐厅
西尔斯	西尔斯
西尔斯汽车中心	搜索自动中心
西尔斯家居服务	搜索_主页_服务
西尔斯 Rent-A-Car	搜索_租车
海边酒店	海边_酒店
赛季	赛季
Seasonova	Seasonova
第 52 季	赛季_52
四季角市场	季节_角落_市场
四季 FCU	seasons_fcu
季节表	季节表
海边酒店	海边_酒店
西雅图咖啡公司	西雅图_咖啡公司

名称	ID
西雅图最佳	西雅图_最佳
Sebago	Sebago
塞巴斯蒂库克谷 FCU	sebastiCook_Valley_FCU
Seca	Seca
第二杯	第二杯
秘方食谱	秘密食谱
Secrets 嘘	Secrets_嘘
SecurCare 自助存储	SecurCare_Self_Storage
安全停车	安全停车
安全优势 FCU	secured_Advantage_FC
塞科利达安全服务	Securitas_安全_服务
美国证券	证券_美国
安全 BanDon	安全_ BanDon
证券银行	证券_银行
证券金融	安全_金融
安全第一国民银行	安全第一国民银行
安全自助存储	安全_自助存储
安全局联邦信用合作社	安全服务_联邦信用合作社
sed'moy Continent	sed'moy_kontinent
世田野的	世田野的

名称	ID
看看's CANDIES	see's_candies
种子	种子
跷跷板	跷跷板
Sefalana	Sefalana
塞加弗雷多	塞加弗雷多
Segafredo Zanetti Es	Segafredo_Zanetti_espresso
塞贡多·穆埃勒	Segundo_Muelle
赛格威 Ninebot	Segway_Ninebot
Sehne Backwaren	Sehne_Backwaren
Sei Tu	Sei_Tu
Sei-Ren	Sei-Ren
西武信用金库	西武信用金库
Seif	Seif
Seijo	Seijo
Seishin 信用金库	Seishin_Shinkin_Bank
Seiya	Seiya
Sekai-no-Yamachan	Sekai-no-Yamachan
Sekerbank	Sekerbank
SelFish	SelFish
Sela	Sela

名称	ID
Select	Select
选择并保存	选择_并_保存
选择旅馆	Select_Inn
选择物理疗法	选择_物理_疗法
选择注册表	选择_注册表
选择人员配备	选择_人员配备
Selected	Selected
瑞士精选	精选_Suisse
Selene	Selene
Seletti	Seletti
Selexyz	Selexyz
Selfix	Selfix
Selver	Selver
Sem'Dney	Sem'_Dney
Sem'Pyatnits	Sem'_Pyatnits
Sem'Sot	Sem'Sot
Sem'ia	Sem'ia
Sem-Angles	Sem-Angles
SemaCharge	SemaCharge
Sembonia	Sembonia

名称	ID
Semerkand	Semerkand
Semeur	Semeur
Semeynaya Kopilka	Semeynaya_Kopilka
Semeynaya apteka	semeynaya_apteka
Semeyniy	Semeyniy
Semeyniy doctor	Semeyniy_doktor
半化学	Semi_Chem
Semiacs	Semiacs
Semishagoff	Semishagoff
塞莫宁房地产经纪人	Semonin_Realtors
Sen-Ryo	Sen-Ryo
森博度假村	森波度假村
森科	森科
Sendas	Sendas
森迪克的食品市场	Sendik's Food_Market
Sendo Ichiba	Sendo_Ichiba
高级军团	Senior_Corps
Senseio 书店	Senseio_Bookstore
森西布鲁	森西布鲁
圣淘沙酒店	圣淘沙酒店

名称	ID
圣淘沙油	圣淘沙石油公司
Sentra Ponsel	Sentra_Ponsel
哨兵食品市场	Sentry_Food_Mart
哨兵食品	哨兵_食物
哨兵保险	哨兵_保险
Seorae	Seorae
首尔熟食店	首尔_熟食店
首尔花园	首尔_花园
Seplife	Seplife
Seppala	Seppala
Seppayla	Seppayla
七匹狼	七匹狼
SerFloor	SerFloor
Sercotel	Sercotel
Serdechko	Serdechko
塞雷纳酒店	塞雷纳酒店
Sergaz	Sergaz
塞尔吉奥·K.	Sergio_K。
Serhs 酒店	Serhs_Hotels
严肃的咖啡	严肃的咖啡

名称	ID
Serra da Estrela	serra_da_Estrela
Servatur	Servatur
ServiCompras	ServiCompras
ServiEstado	ServiEstado
塞维班卡	塞维班卡
服务 1st FCU	service_1st_FCU
服务 CU	服务_cu
服务之王碰撞	服务之王碰撞
ServiceMaster 干净	ServiceMaster_Clean
服务	服务
Servicio Andaluz de Salud	servicio_andaluz_de_Salud
Servifácil	Servifácil
瑟维集团	瑟维集团
Servis	Servis
Servus 信用合作社	Servus_Credit_U
性爱时装	Ses_Fashions
芝麻街英语	芝麻街_英语
会话	会话
Sete Estrelas	Sete_Estrelas
Seth M.R. Jaipuria 学校	Seth_M._R._Jaipuria_Schools

名称	ID
濑户信用金库	濑户信用金库
设定点	设定点
塞瓦美女	Seva_Beaut
Seven	Seven
七山	七山
七个披萨	七_Pizza
基督复临安息日会	基督复临安息日会基督复临安息日会
Sever-Metropol'	Sever-Metropol'
Severa	Severa
Sevt	Sevt
性感牛仔裤	性感_牛仔裤
性感茶	Sex_Tea
Seyhan expres	seyhan_expres
塞汉拉尔市场	塞汉拉尔_市场
Seyidoglu	Seyidoglu
Señor Limon	Señor_Limon
Sf Best	Sf_Best
Sfera	Sfera
Sferis	Sferis
Sfic	Sfic

名称	ID
Sfree	Sfree
Sgomon'	Sgomon'
陕汽集团通信车	陕西_汽车_Grp_Comm_Vehicle
陕西光明眼镜	陕西_Bright_Glasses
陕西十三	陕西_十三
Shabu Shabu Nang Nai	Shabu_Shabu_Nang_Nai
涮火锅 Onyasai	Shabu-Shabu_Onyasai
沙布里	沙布里
Shabushi	Shabushi
Shag Navstrechu	Shag_Navstrechu
Shaguoju	Shaguoju
Shahnaz Husain	Shahnaz_Husain
Shake Shack	Shake_Shack
Shake's 冰冻蛋奶冻	Shake's_frozen_Custard
Shakespeare & Co	莎士比亚_&_Co
Shakey's	Shakey's
Shakh	Shakh
嘉康利分销商	Shaklee_分销商
Shama	Shama
Shama Luxe	ShamaLuxe

名称	ID
沙米尔	沙米尔
沙米尔银行	Shamil_Bank
洗发水专家	洗发水_专家
Shamrao Vithal 合作银行	shamrao_vithal_co-op_Bank
山山	Shan_Shan
Shana	Shana
山东农村信用社	山东_农村信用联盟
Shane 英语学校	Shane_English_
Shane's Rib Shack	Shane's_rib_shack
Shaneleasing 信用合作社	Shaneleasing_Credit_Union
林尚玉	尚玉林
上海航空旅行酒店	上海_航空_旅游_酒店
上海商业储蓄银行	上海_商业和储蓄银行
上海电影艺术中心	上海_电影_艺术_中心
上海欢乐城	上海_Gala_City
上海客栈	上海_客栈
上海浦东发展银行	上海_浦东发展银行
上海农村商业银行	上海_农村_商业银行
上海学校用品超市	上海_学校_用品_超市
商丘华商银行	商丘_中国_商业银行

名称	ID
香格里拉	香格里拉
尚尚前	尚尚前
上虞农商银行	上虞农村商业银行
山水S酒店	山水_s_Hotel
Shanti Juniors	Shanti_Juniors
Shanton	Shanton
山西银行	山西银行
山西大运	山西_大运
山西介休农村商业银行	山西_介休_农村_商业银行
山西灵石农村商业银行	山西_灵石_农村通信_银行
山西平遥农村商业银行	山西_平遥_农村通信_银行
山西省农村信用社b	山西_农村_信用_合作社
山西闻喜农村商业银行	山西_文溪_农村_商业银行
山西榆次农村商业银行	山西_育慈_农村_商业银行
山西蔚县农村商业银行	山西_玉县_农村_商业_银行
ShaoMeng 酒店	ShaoMeng_酒店
Sharaf DG	sharaf_dg
沙拉夫交易所	Sharaf_Exchan
ShareTea	ShareTea
Sharefax	sharefax_cu

名称	ID
沙雷汉	沙雷汉
Shari's	Shari's
Sharin	Sharin
沙迦市中心有限责任公司	Sharjah_city_centre_LLC
沙迦合作社	Sharjah_合作社_社会
沙迦伊斯兰银行	Sharjah_Islamic_Bank
鲨鱼咬吐司	Shark_Bites_Toast
Sharky's Woodfied 墨西哥烧烤	Sharky's_woodfired_Mexican_Grill
夏尔马运输	Sharma_Transport
夏普	夏普
沙图拉	沙图拉
Shaukat Khanum	Shaukat_Khanum
剃须刀商店	剃须刀_Shop
Shaw's	Shaw's
Shaw's Osco	Shaw's_osco
Shaw's 超市	Shaw's 超市
Shawaya House	Shawaya House
Shawerma El Reem	Shawerma_El_Reem
她是	她是
剪毛	剪毛

名称	ID
Sheetz	Sheetz
射洪农商银行	射洪农村商业银行
谢赫·巴拉德	Sheik_El_Balad
ShelfGenie	ShelfGenie
Shell	Shell
壳牌酒店	Shell_Hotel
壳牌充能	Shell_Recharge
外壳选择	Shell_Select
壳牌商店	Shell_Shop
壳牌快递	壳牌快递
住房保险	庇护所_保险
申福德未来派学校	Shemford 未来派学校
Shemrock	Shemrock
ShenPu 宠物医院	ShenPu_Pet_Hospital
盛雄	Seng_Siong
盛 XiFu 帽专卖店	Sheng_ _Hat_Stor XiFu e
盛源超级中心	圣元_超级中心
盛和道	盛和道
申贵派	申贵派
圣武牛肉	盛武_牛肉

名称	ID
嵊州农村商业银行	嵊州_农村_商业银行
神木农商银行	神木_农村_商业银行
神农大酒店	神农_酒店
申顿医疗集团	Shenton_Medical_Group
申万宏源证券	申万_宏源证券
沈阳农村商业银行	沈阳_农村_商业银行
深圳发展银行	深圳发展银行
深圳农村商业银行	深圳_农村_商业银行
深航假期	深圳航空_假日
喜来登	喜来登
夏尔巴人的食物	夏尔巴_Alimention
Sherrif's	Sherrif's
雪莉	雪莉
世尚科学商店	Shi_shang_Science_
Shiatzy Chen	Shiatzy_Chen
柴信用金库	芝信用金库
七林	七林
Shiekh Shoes	Shiekh_Shoes
盾牌	盾牌
志贺中央信用金库	滋贺中央信用金库

名称	ID
ShihShen	ShihShen
士林台湾街头小吃	士林_台湾_街头_零食
世纪园酒店	世纪园_酒店
希拉夫	希拉夫
Shilo Inn	Shilo_Inn
Shimachu HOME'S	Shimachu_Home'S
岛田挂川信用金库	Shimada_Kakegawa_Shinkin_Bank
岛村	岛村
岛波信用金库	岛波信用金库
岛根中央信用金库	岛根县_中央_信用金库
世茂影院	世茂影院
新光银行	新光银行
闪耀之城	Shine_City
Shingle Inn	Shingle_Inn
越南新韩银行	新韩银行_越南
闪亮的房子	Shining_House
闪亮的味道	Shining_Taste
东云信用金库	Shinonom_Shinkin_Bank
Shinsegye	Shinsegye
Shinshifuku No Futata	Shinshifuku_No_Futata

名称	ID
新和超市	新和超市
Shiogensui	Shiogensui
Shipley Do-Nuts	Shipley_Do-Nut
希尔	希尔
白熊咖喱	Shirokuma_Curry
衬衫和领带	衬衫_&_领带
Shivajinagar Sahakari Bank	Shivajinagar_Sahakari_Bank
Shivajirao Bhosale Sahakari Bank	Shivajirao_Bhosale_Sahakari_Bank
Shivajirao Bhosle Bank	Shivajirao_Bhosle_Bank
世维关	世维关
Shiweiji	Shiweiji
Shizu	Shizu
静冈烧津信用金库	静冈_烧津_信用金库
静铁店	静铁店
香城餐厅	Shing_Town_Restaurant
Shoan	Shoan
鞋业嘉年华	鞋履嘉年华
鞋类诊所	鞋类诊所
鞋业公司	鞋业公司
鞋业部	鞋部

名称	ID
鞋子折扣	鞋子_折扣
鞋子感觉	Shoe_Sens
鞋展	Shoe_Show
鞋工作室	鞋类工作室
鞋区	Shoe_Zone
ShoeBeDo	ShoeBeDo
Shoeday	Shoeday
Shoeless Joe 的	Shoeless_Joe's
鞋子和袜子	鞋子_&_Sox
Shokoladnitsa	Shokoladnitsa
Sholodge	Sholodge
湘南信用金库	湘南信用金库
Shoney's	Shoney's
Shoney's Inn	Shoney's_inn
Shooster	Shooster
购物和驾车	Shop_&_Drive
Shop & Go	Shop_&_Go
Shop 'n Save	shop_'n_Save
选购快递	购物快递
购物 N'Save Food	Shop_n_Save_Food

名称	ID
Shop-N-Go	Shop-N-Go
Shopi	Shopi
购物	购物
Shopko	Shopko
Shopko 光学	Shopko_光学
购物者们	购物者们
购物者药店	Shoppers_Drug_Mart
购物者药房	购物者_药房
Shoprite	Shoprite
Shoprite 你保存	Shoprite_U_Save
Shopretho	Shopretho
Shorepower	Shorepower
Shorewest 房地产经纪人	Shorewest_Realtors
短停	Short_Stop
Shota Muni Sushi&Grill	Shota_muni_Sushi&Grill
寿航超市	首航超市
寿欣芳	寿欣芳
Shouchailu 烤鸭	Shouchailu_Roast_Duck
寿康永乐	寿康永乐
首汽租车	首汽_租车

名称	ID
Shoushia	Shoushia
寿县农村商业银行	寿县_农村_商业银行
昭和公园	Showa_Park
昭和壳牌	Showa_Shell
展示柜	展示柜
Showcolate	Showcolate
Showmars	Showmars
Shree Swaminarayan Mandir	Shree_Swaminarayan_Mandir
Shri Kannan 百货商店	Shri_kannan_百货店
Shri Rajkot 地区合作银行	SHRI_rajkot_district_district_
虾篮	虾篮
Shrimpy 餐厅	Shrimpy_Resta
Shriners 儿童医院	Shriners_Hospital_for Chir
Shriram 城市联盟财务	Shriram_City_Union_Finan
顺阳教育	Shsun_教育
蜀大侠火锅	Shu_Daxia_Hot_Pot
叶爽	双叶
Shufersal	Shufersal
曙光国际	曙光国际
淑煌	淑煌

名称	ID
舒江春	舒江春
书久香	书久香
顺成烘焙坊	Shun_Chen_Bakery
牛顺峰飞飞	Shun_Feng_Fei_Niu
刘顺扣	Shun_Kou_Liu
王顺志	王顺智
顺德农村商业银行	顺德农村商业银行
顺和	顺和
顺六儿	顺六儿
Shurgard 自助仓储	Shurgard_Self-Storage
Shutiq	Shutiq
蜀阳农村商业银行	蜀阳农村商业银行
Shuyishaoxiancao	Shuyishaoxiancao
淑子香	淑子香
Shveyniy Mir	Shveyniy_Mir
si.mobil	si.mobil
Sia 燃料	Sia_Fuel
Siam Gas	Siam_Gas
Siauliu Bankas	Siauliu_Bankas
西伯利亚的	西伯利亚的

名称	ID
西布霍洛德	西布霍洛德
Sibneft'	Sibneft'
Sibylla	Sibylla
Sical	Sical
四川银行	四川_银行
四川现代	四川_现代
四川豆腐布丁	四川_豆腐_布丁
自驾游西西里岛	西西里岛_by_Car
Sico	Sico
Sicredi	Sicredi
Sidorme	Sidorme
西门子	西门子
Siemes Schuhcenter	Siemes_Schuhcenter
Siera	Siera
齿状山脉	齿状山脉
塞拉咖啡	Sierra_Coffee
马德雷山脉	Sierra_Madre
塞拉太平洋 FCU	Sierra_Pacific_FCU
SierraPark	SierraPark
Sieu Thi Dien May-Noi That Cho Lon	Sieu_Thi_Dien_May-Noi_That_Cho_Lon

名称	ID
Siglo Dental	Siglo_Dental
西格玛涂料	Sigma_Paints
标志 1	Sign_1
信号图形	信号_图形
签名	签名
招牌旅馆	Signature_Inn
招牌厨房	招牌厨房
Signo Aquários	Signo_Aquários
明天之前的迹象	Signs_By_Tomor
现在的迹象	Sign_Now
Siko koupelny & kuchyne	siko_koupelny_&_kuchyne
西康集团有限公司	西康集团有限公司
SiliGom	SiliGom
硅谷银行	N.A。
硅谷培训	硅谷训练
Silk 油	Silk_oil
丝丽酒店	丝绸_酒店
丝绸银行	丝绸银行
Silmet	Silmet
Silpo	Silpo

名称	ID
Silver Diner	Silver_Diner
白发美人	银狐
银矿订阅者	Silver_Mine_Subs
银州立学校 CU	银州立学校_cu
西尔维拉多	西尔维拉多
西尔维娅咖啡	西尔维亚咖啡
Simcare	Simcare
Simit Dünyasi	Simit_dünyasi
Simit Sarayi	Simit_Sarayi
Simitci Dünyasi	Simitci_dünyasi
西蒙斯银行	Simmons_Bank
西蒙斯第一银行	Simmons_First_Bank
西蒙购物中心	Simon_Malls
SimonMed	SimonMed
简约胶囊酒店	Simple_Capsule_Hotel
简易超市	Simple_Mart
Simple Simon's	Simple_Simon's_Pizza
简直是亚洲	简单_亚洲
简直时尚	简约时尚
只需自助存储	简单_自助存储

名称	ID
简单生活	简单生活
Simplylife 面包店咖啡	SimplyLife_Bakery_C
Sin más Piojitos	Sin_Más_Piojitos
Sin Tabu 礼品和附件	sin_tabu_regalos_y_complementos
信纳泰保险	信纳泰_保险
Sincap Marketler Zincir	Sincap_Marketler_Zinciri
真诚	真诚
辛克莱	辛克莱
信德银行	Sindh_Bank
SingTel	SingTel
歌手	歌手
Sing River FCU	Singing_river_fcu
单弓	单弓
单身贵族	单曲_贵族
Singlenoble	Singlenoble
西诺	西诺
中德 Bausparkasse	中德_Bausparkasse
Sinoadl	Sinoadl
中化集团	中化集团
国金证券	国金证券

名称	ID
Sinomax	Sinomax
Sinooperi	Sinooperi
中石化	中石化
中石化埃索	Sinopec_Esso
Sinsay	Sinsay
信义地产	Sinyi_Realty
Sinequanone	Sinequanone
Sioumazang 烤肉	Sioumazang_烤肉
苏谷社区 CU	Sioux_Valley_community_cu
披萨爵士	先生_Pizza
Sir Speedy 印刷和营销服务	sir_speedy_printing_and_marketing_an
Sirenis 酒店及度假村	Sirenis_酒店_度假村
警笛	警笛
沙朗寨子	Sarloin_Stockade
Sirma	Sirma
魏姐	Wei姐姐
姐妹积分	Sisters_Poin
巴黎圣母院修女会	巴黎圣母院的姐妹们
圣约瑟夫修女会	圣约瑟夫姐妹们
圣弗朗西斯修女会健康服务部	Sisters_of_st._Francis_Health_Services

名称	ID
Sita Murt	Sita_Murt
景点和景观	Sites_&_Paysages
Siti Obuv'	Siti_Obuv'
Sitilab	Sitilab
Sitis	Sitis
Sivagangai Dist 中央合作银行	SivaGangai_dist_central_Co-op_Bank
六面旗帜	六面旗帜
思燕丽	思燕丽
更大尺寸	更大尺寸
尺寸不限	尺寸_不限
Sizzler	Sizzler
Skarlet	Skarlet
Skelleftebránslen	Skelleftebránslen
技能法杖	技能_法杖
皮肤补品	皮肤补品
Skiny	Skiny
船长	船长
船长炸鱼和薯条	船长_Fish_&_Chips
斯基普顿建筑协会	斯基普顿建筑协会
滑板套装	滑板套装

名称	ID
斯凯恩银行	Skjern_Bank
Skjold Burne	Skjold_Burne
奴隶炎	奴隶炎
滑板车	滑板车
斯科林根	斯科林根
Skousen	Skousen
斯科沃罗德卡	斯科沃罗德卡
天空	天空
天空珠宝	天空珠宝
空中公园安全	空中公园_安全
SkyNet	SkyNet
天鸟酒店	Skybird_Hotel
斯凯银行	Skye_Bank
Skyedu	Skyedu
云雀	云雀
天际线辣椒	天际线_Chili
Skyward	skyward_cu
斯拉多米尔	斯拉多米尔
斯拉塔	斯拉塔
Slatinska banka	Slatinska_Banka

名称	ID
Slatka kuca	slatka_kuca
Slavneft'	Slavneft'
Slavnyeft'	Slavnyeft'
Slazenger	Slazenger
睡眠之乡	睡眠国家
睡个好觉	睡眠高度
睡眠旅馆	Sleep_Inn
睡眠用品店	Sleep_Outfitters
睡眠训练床垫中心（Sleep Train）	睡眠训练床垫中心（Sleep_Train）
睡眠大师	睡眠大师
Sleepy's	Sleepy's
Slice N'Dice	Slice_N'_Dice
意大利一片	意大利片段
苗条的鸡	Slim_Chickens
瘦身食物	Slim_Food
SloWatch	SloWatch
Sloneczko	Sloneczko
斯洛伐克的	斯洛伐克的
斯洛文尼亚体育馆	Slovenska_Sporitelna
斯洛夫纳夫特	斯洛夫纳夫特

名称	ID
Slug & Lettuce	Slug_&_Lettuce
Slemberland 家具	Slumberland_家具
Sma	Sma
Smaaash	Smaaash
小型	小型
小型企业管理局	小型企业管理
小型优雅酒店	小型_优雅_酒店
小型豪华酒店	小型豪华酒店
希尔顿小酒店	Small_Xierdun_Hotel
小北	小北
SmallCakes	SmallCakes
Smar 儿童艺术中心	智能艺术中心_儿童版
SmarTone	SmarTone
Smart & Fin	Smart_&_Final
聪明的大脑	智能大脑
智能贴合	智能贴合
智能修复	智能修复
智能服务	智能服务
智能套装	智能套装
智能超级市场	智能超级市场

名称	ID
智能超级市场	Smart_Supermercad
智能实验室	智能实验室
Smartstyle 家庭发廊	Smartstyle_Family_美发沙龙
Smartwin 国际英语	Smartwin_国际版_英语
Smashburg	Smashburg
Smesco 玛特	Smesco_Mart
Smeshnye Tseny	Smeshnye_Tseny
Smiggle	Smiggle
微笑	微笑
微笑直达俱乐部	Smile_Direct_
微笑气体	微笑气体
微笑加油站	微笑加油站
微笑店	微笑店
微笑商店	Smiles Stores
史密斯的	史密斯的
史密斯的燃料中心	史密斯燃料中心
史密斯市场	史密斯市场
Smith 的 Marketplace	史密斯市场_燃料中心
Smith's Marketp	史密斯_市场_药房
史密斯药房	史密斯药房

名称	ID
史密斯菲尔德的 Chicken 'n Bar-B-Q	Smithfield's_chicken_'n_bar-b-q
史密斯城	史密斯城
Smitty's	Smitty's
Smog Busters	Smog_Busters
Smokemart & Giftb	Smokemart_&_Giftbox
吸烟者快车	吸烟者_快递
烟熏骨头	Smokey_Bon
Smokin's	Smokin_Joe's
斯摩棱斯克-法尔马齐亚	斯摩棱斯克-法尔马齐亚
Smoliar	Smoliar
斯莫伦斯基	斯莫伦斯基
冰沙工厂	冰沙工厂
冰沙屋	Smoothie_Hou
冰沙之王	冰沙之王
冰沙兑换	冰沙_Xchange
零食攻击	零食攻击
快照驱动器	快照盘
快照健身	Snap_Fitn
快餐厨房	Snap_Kitch
Snap-on	Snap-on

名称	ID
快照盘	快照盘
活泼的番茄披萨	快餐_番茄_披萨
Sneaky Pete's	Sneaky_Pete's
Snelling 人员配备服务	Snelling_人员配备_服务
斯内任斯基伊	斯内任斯基伊
Snezhnaya Koroleva	Snezhnaya_Koroleva
Snezhniy Gorodok	Snezhniy_Gorodok
狙击	狙击
Snoekies	Snoekies
Snoras	Snoras
白雪公主	白雪公主
Snowhite 干洗店	Snowhite_Dry_Cle
太别致了	太时髦了
太棒了	太棒了
太甜了	太甜了
so.bio	so.bio
SoCoo'c	SoCoo'c
沉浸在奢华中	沉浸在奢华中
Sobeys	Sobeys
Sobinbank	Sobinbank

名称	ID
Soccx	Soccx
社交	社交
Socie	Socie
Sociedad Textil Lonia	Sociedad_Textil_Lonia
阿尔巴尼亚兴业银行	Societe_Generale_Albania
法国兴业银行ExpressBank	兴业银行_Generale_ExpressBank
法国兴业银行	法国兴业银行
科特迪瓦银行	Sociéte_Ivoirienne_de_Banque
Socremo	Socremo
苏打	苏打
苏打块	Soda_Block
索迪斯	索迪斯
Sodié Doces	Sodiè_Doces
沙发	沙发
Sofeco	Sofeco
索菲亚	索菲亚
索非亚市中心电影院	索非亚市中心电影院
SoftBank	SoftBank
SoftMoc	SoftMoc
柔和的色调	柔和的色调

名称	ID
软件等	软件_等
Sofuren	Sofuren
Soglasie	Soglasie
Soglia	Soglia
苏豪区	苏豪区
土壤	土壤
Soiuz	Soiuz
Sok	Sok
索尔	索尔
索尔酒店	Sol_Hotels
Sol Melia	Sol_Melia
索拉沙龙工作室	索拉沙龙工作室
索拉马莱汽车	Solamalai_汽车
Solapur Janata Sahakari Bank	Solapur_Janata_Sahakari_Bank
Solapur Janta Sahakari Bank	Solapur_Janta_Sahakari_Bank
太阳能	太阳能
Solarcheck	Solarcheck
Soleil Sucré	Soleil_Sucré
固体	固体
团结社区 FCU	solidarity_Community_FCU

名称	ID
Solidarnost'	Solidarnost'
Solike	Solike
索林卡	索林卡
Solnyshko	Solnyshko
Solo	Solo
单人挤压	Solo_Squees
Soloptical	Soloptical
Solorunners	Solorunners
Solstice 太阳镜精品店	Solstice_太阳镜_精品店
解决方案美食	解决方案_Cuisines
阳光酒店	Soluxe_Hotel
Soma Intimates	Soma_Intimates
萨默菲尔德	萨默菲尔德
有点可疑	Something_Fishy
某个地方	某个地方
Somogy Takarék	Somogy_Takarék
Somwang Ngern Sang Dai	Somwang_Ngern_Sang_Dai
索纳埃·塞拉	Sonae_Seira
索纳利卡	索纳利卡
Sonatrach	Sonatrach

名称	ID
索纳克斯	索纳克斯
Sonda	Sonda
索内拉滑雪道	Sonera_piste
索内里银行	Soneri_Bank
索尼斯塔 ES Suites	sonesta_es_Suites
索尼斯塔酒店及度假村	Sonesta_酒店及度假村
索尼斯塔精选	Sonesta_Select
索尼斯塔简约套房酒店	Sonesta Simply_Suites
松鹤楼	松鹤楼
宋华江	宋华江
松雷	松雷
松林菜	松林_料理
索尼克	索尼克
Sonitrol	Sonitrol
Sonny's Real Pit 烧烤	Sonny's_real_pit_BBQ
索诺尔	索诺尔
索诺玛银行	Sonoma_Bank
索丽莎·布兰卡	索里萨_布兰卡
Soobe	Soobe
很快 Yi	Soon_Yi

名称	ID
Sooper CU	sooper_cu
Sootang Hobby	Sootang_Hobby
索菲·帕里斯	Sophie_Pari
索普隆银行	Sopron_Bank
Sorabol	Sorabol
空知工商信用合作社	Sorachi_商业_&_行业_信用_合作社
空知信用金库	Sorachi_Shinkin_Bank
Sorat	Sorat
Soreli-Isla	Soreli-Isla
Sorepro	Sorepro
索里亚娜	索里亚娜
索里亚纳快车	Soriana Express
索里亚娜·梅尔卡多	Soriana_Mercado
Soriana Super	Soriana_Super
索罗拉	索罗拉
Sorvete Itália	Sorvete_Italia
Sorvetes FriSabor	Sorvetes_ FriSabor
索塞迪	索塞迪
Soshakan FC	soshakan_fc
Sota Caballo Rey	Sota_Caballo_Rey

名称	ID
Sotetsu Rozen	Sotetsu_Rozen
Sotoviy Mir	Sotoviy_Mir
Sotsial'naya apteka	sotsial'naya_apteka
Sotsinvetbank	Sotsinvetbank
Soueast	Soueast
灵魂起源	灵魂起源
灵魂帕丁森化学家	Soul_Pattinson_Chemist
SoulCycle	SoulCycle
Soulman's Bar-B-Que	Soulman's_bar-b-que
声音 CU	Sound_cu
汤沙拉	Souper_Salad
种植园	种植园
Sour Sally	Sour_Sally
来源 East	来源_East
来源：伦敦	来源_伦敦
来源 West	来源_West
南区联赛	South_Division_cu
南印度银行	南印度银行
南马拉巴尔格拉明银行	South_Malabar_Gramin_Bank
南地铁 FCU	South_Metro_FCU

名称	ID
南星药业	南星药物
南宝贝	南宝贝
索斯科	索斯科
南方面包店	南方面包店
南方银行	Southern_Bank
南方咖啡	南方咖啡
南方炸鸡	Southern_fried_Chicken
南湖加州大学	Southern_Lakes_cu
南方安全 FCU	Southern_security_FCU
南方太阳	Southern_Sun
南方海啸	南方海啸
南国 FCU	Southland_FCU
南州银行	南州银行
南方信托银行	南方信托银行
西南 66 CU	southwest_66_cu
西南 FCU	southwest_fcu
索瓦拉基小屋	Souvlaki_Hut
苏维特斯基	苏维特斯基
Sovkombank	Sovkombank
Sox World	Sox_World

名称	ID
Soykan	Soykan
soyuzpechat'	soyuzpechat'
Sozvezdie Krasoty	Sozvezdie_Krasoty
空间	空间
太空城 CU	space_city_cu
太空风水	空间_风水_风水
太空世界声音花园	太空世界_声佳园
Spacio 1	Spacio_1
意大利面仓库	Spaghetti_Warehote
Spangles	Spangles
Spar Gourm	Spar_Gourm
Spar Kjöp	Spar_Kjöp
Spar Nord Bank	Spar_Nord_Bank
Spar-und Leihkasse zu Bredstedt AG	spar_und_leihkasse_zu_bredstedt_ag
Spar-und Creditbank	spar-und_CreditBank
Sparbank	Sparbank
Sparbank 背心	Sparbank_Vest
斯巴达银行	Sparda_Bank
Spare Rib E	Spare_rib_Express
Sparebank 1	Sparebank_1

名称	ID
Sparebanken Sogn 和 Fjordane	sparebanken_sogn_og_fjordane
南备用银行	Sparebanken_Sour
备用银行背心	Sparebanken_Vest
Sparebanken Eost	Sparebanken_Est
Sparisjodurinn	Sparisjodurinn
Spark	Spark
Sparkasse	Sparkasse
亚琛Sparkasse	Sparkasse_Aachen
Sparkasse Aichach-Schrobenhausen	Sparkasse_Aichach-Schrobenhausen
Sparkasse Allgau	Sparkasse_Allgau
Sparkasse Altenburger 乐园	Sparkasse_Altenburger_Land
Sparkasse Altmark West	Sparkasse_Altmark_West
Sparkasse Altötting-Mühldorf	Sparkasse_Altötting-Mühldorf
Sparkasse Amberg-Sulzbach	Sparkasse_Amberg-Sulzbach
Sparkasse Ansbach	Sparkasse_Ansbac
Sparkasse Arnsberg-Sundern	Sparkasse_Arnsberg-Sundern
Sparkasse Arnstadt-Ilmenau	Sparkasse_arnstadt-Ilmenau
Sparkasse Aschaffenburg-Alzenau	Sparkasse_Aschaffenburg-Alzenau
Sparkass Attendorn-Lennestadt-Kir e。	Sparkasse_Attendorn-Lennestadt-Ki
Sparkasse Aurich-Norden	Sparkasse_Aurich-Norden

名称	ID
Sparkasse Bad Hersfeld-Rotenburg	Sparkasse_Bad_Hersfeld-Rotenburg
Sparkasse 巴特基辛根	Sparkasse_Bad_Kissingen
Sparkasse Bad Neustadt a. d. Saale	sparkasse_bad_neustadt_a._d._saale
Sparkasse Bad Tölz-Wolfratshausen	Sparkasse_Bad_Tölz_-_Wolfratshausen
Sparkasse 巴登-巴登嘉格瑙	Sparkasse_Baden-Baden_Gaggenau
班贝格 Sparkasse	斯巴卡斯_班贝格
Sparkasse Barnim	Sparkasse_Barnim
巴腾堡储蓄银行	Sparkasse_Batten
拜罗伊特Sparkasse	Sparkasse_Bayreuth
Sparkasse Beckum-Wadersloh	Sparkasse_Beckum-Wadersloh
本斯海姆Sparkasse	Sparkasse_本斯海姆
Sparkasse Berchtesgadener Land	Sparkasse_Berchtesgadener_Land
Sparkasse Bergkamen-Bönen	Sparkasse_Bergkamen-Bönen
Sparkasse Bielefeld	Sparkasse_Bielefeld
波鸿Sparkasse	Sparkasse_波鸿
Sparkasse Bodensee	Sparkasse_Bodensee
Sparkasse Bonndorf-Stühlingen	Sparkasse_Bonndorf-Stühlingen
Sparkasse Bottrop	Sparkasse_Bottrop
不来梅哈芬Sparkasse	Sparkasse_不来梅哈芬
Sparkasse Burbach-Neunkirche	Sparkasse_Burbach-Neunkirch

名称	ID
Sparkasse Burgenlandkreis	Sparkasse_布尔根兰德县
Sparkasse Bühl	Sparkasse_Bühl
Sparkasse Celle	Sparkasse_Celle
Sparkasse Chemnitz	Sparkasse_Chemnitz
Sparkasse Coburg-Lichtenfels	Sparkasse_Coburg_-_Lichtenfels
Sparkasse Dachau	Sparkasse_达豪
达姆施塔特Sparkasse	Sparkasse_Darmstadt
Sparkasse Degendorf	Sparkasse_Deggend
Sparkasse Dieburg	Sparkasse_Diebur
迪伦堡 Sparkasse	Sparkasse_Dillen
Sparkass Dinslaken-Voerde-H e unxe	Sparkasse_dinslaken-Voerde-Hünxe
Sparkasse Donawörth	Sparkasse_Donawörth
Sparkasse Donnersberg	Sparkasse_Donnersberg
多特蒙德储蓄银行	Sparkasse_Dot
Sparkasse Duderstadt	Sparkasse_杜德施塔特
杜伊斯堡Sparkasse	Sparkasse_Duisburg
Sparkasse Düren	Sparkasse_Düren
Sparkasse Eichstátt	Sparkasse_Eichstátt
Sparkasse Einbeck	Sparkasse_Einbeck
Sparkasse Elbe-Elster	Sparkasse_Elbe-Elster

名称	ID
Sparkasse Elmshorn	Sparkasse_Elmshorn
Sparkasse Emden	Sparkasse_Emde
Sparkasse Emslan	Sparkasse_emslan
Sparkasse Engen-Gottmadingen	Sparkasse_Engen-Gottmadingen
Sparkasse Ennepetal-Breckerfeld	Sparkasse_Ennepetal-Breckerfeld
Sparkasse Erding-Dorfen	Sparkasse_erding-Dorfen
Sparkasse 埃尔兰根	Sparkasse_El
Sparkasse Erwitte-Anröchte	Sparkasse_Erwitte-Anröchte
Sparkasse Essen	Sparkasse_esen
Sparkasse Fichtelgebirge	Sparkasse_Fichtelgebirge
Sparkasse Finnentrop	Sparkasse_Finnentrop
Sparkasse Forchheim	Sparkasse_Forchhei
Sparkasse Freiburg-Nördl。布赖斯	Sparkasse_Freiburg-Nördl。 _Breisgau
Sparkasse Freising	Sparkass_Freising
Sparkasse Freyung-Grafenau	Sparkasse_Freyung-Grafenau
Sparkasse Fröndenberg	Sparkasse_Fröndenberg
Sparkasse Fulda	Sparkasse_Fulda
Fürstenfeldbruck Sparkasse	Sparkasse_Fürstenfeldbruck
Sparkasse Fürth	Sparkasse_Fürth
盖尔森基兴Sparkasse	Sparkasse_盖尔森基兴

名称	ID
Gengenbach Sparkasse	Sparkasse_Gengenbac
Sparkasse Gera-Greiz	Sparkasse_Gera-Greiz
Sparkasse Germersheim-Kandel	Sparkasse_Germersheim-Kandel
Sparkasse Geseke	Sparkasse_Geseke
Sparkasse Gevelsberg-Wetter	Sparkasse_Gevelsberg-Wetter
Sparkasse Gießen	Sparkasse_Gießen
Sparkasse Gifhorn-Wolfsbur	Sparkasse_Gifhorn-Wolfsb
Sparkasse Goslar/Harz	Sparkasse_Goslar/Harz
Sparkasse Gronau	Sparkasse_Gronau
Sparkasse Grunberg	Sparkasse_Grunberg
Sparkasse Gummersbach-Bergneustadt	Sparkasse_Gummersbach-Bergneustad
Sparkasse Göppingen	Sparkasse_Göppinge
哥廷根 Sparkasse	Sparkasse_哥廷根
Sparkasse Günzburg-Krumbach	Sparkasse_Günzburg-Krumbach
Sparkasse Gütersloh-Rietberg	Sparkasse_Gütersloh-Rietberg
Sparkasse HagenHerdecke	Sparkasse_ HagenHerdecke
Sparkasse Hameln-Weserbergland	Sparkasse_Hameln-Weserbergland
Sparkasse Hamm	Sparkasse_Hamm
Sparkasse Hanau	Sparkasse_Hanau
Sparkasse Hanauerland	Sparkasse_Hanauerland

名称	ID
汉诺威Sparkasse	Sparkasse_Han
Sparkasse Harburg-Buxtehude	Sparkasse_Harburg-Buxtehude
Sparkasse Haslach-Zell	Sparkasse_Haslach-Zell
Sparkasse Hattingen	Sparkasse_Hattinge
Sparkasse Hegau-Bodensee	Sparkasse_Hegau-Bodensee
Sparkasse Heidelberg	Sparkasse_Heidelber
Sparkasse Hennstedt-Wesselburen	Sparkasse_Hennstedt-Wesselburen
Sparkasse Herford	Sparkass_Herford
Sparkasse Hilden-Ratingen-Velbert	Sparkasse_Hilden-Ratingen-Velb
Sparkasse Hildesheim Goslar Peine	Sparkasse_Hildesheim_Goslar_Peine
Sparkasse Hochfranken	Sparkasse_Hochfranke
Sparkasse Hochrhein	Sparkasse_Hochrhein
Sparkasse Hochsauerland	Sparkasse_Hochsauerland
Sparkasse Hochschwarzwald	Sparkasse_Hochschwarzwald
Sparkasse Hohenlohekreis	Sparkasse_Hohenlohekreis
Sparkasse Hohenwestedt	Sparkasse_Hohenwestedt
Sparkasse Holstein	Sparkasse_Holste
Sparkasse Höxter	Sparkasse_Höxter
Sparkasse Ingolstadt Eichstátt	Sparkasse_Ingolstadt_Eichstátt
Sparkasse Jena-Saale-Holzland	Sparkasse_Jena-Saale-Holzland

名称	ID
Sparkasse Jerichower Land	Sparkasse_Jerichower_Land
Sparkasse Karlsruhe	Sparkasse_Karlsruhe
Sparkasse 卡尔斯鲁厄埃特林根	Sparkasse_Karlsruhe_Etlingen
Sparkasse Kierspe-Meinerzhagen	Sparkasse_kierspe-Meinerzhagen
Sparkasse Kleve	Sparkasse_Kleve
Sparkasse Koblenz	Sparkasse_Koblenz
Sparkasse Kraichgau	Sparkasse_Kraichga
Sparkasse Krefeld	Sparkasse_Krefel
Sparkasse Kulmbach-Kronach	Sparkasse_Kulmbach-Kronach
Sparkasse KölnBonn	Sparkasse_KölnBon
Sparkasse Landsberg-Diesen	Sparkasse_Landsberg-Diese
兰茨胡特 Sparkasse	Sparkasse_兰茨胡特
Sparkasse Langen-Seligenstadt	Sparkasse_Langen-Seligenstadt
Sparkasse Laubach-Hungen	Sparkasse_Laubach-Hunge
Sparkasse LeerWittmund	Sparkasse_ LeerWittmund
莱比锡Sparkasse	莱比锡储蓄银行
Sparkasse Lemgo	Sparkasse_Lemgo
Sparkasse 勒沃库森	Sparkasse_Leverkus
Sparkasse Lippstadt	Sparkasse_Lippstadt
Sparkasse Lörrach-Rheinfelden	Sparkasse_Lörrach-Rheinfelden

名称	ID
Sparkasse Lüdenscheid	Sparkasse_Lüdenschei
吕讷堡Sparkasse	Sparkasse_Lüneburg
Sparkasse Lunen	Sparkasse_Lunen
Sparkasse Mainfranken 维尔茨堡	Sparkasse_Mainfranken_Würzburg
Sparkasse Mainz	Sparkasse_Mainz
Sparkasse Mansfeld-Sudharz	Sparkasse_Mansfeld-Südharz
Sparkasse Marburg-Biedenkopf	Sparkasse_Marburg-Biedenkopf
Sparkasse Markgräflerland	Sparkasse_Markgräflerland
Sparkasse Mecklenburg-Nordwest	Sparkasse_梅克伦堡-西北
Sparkasse 梅克伦堡-什未林	Sparkasse_Mecklenburg-Schwerin
Sparkasse Mecklenburg-Strelitz	Sparkasse_Mecklenburg-Strelitz
Sparkasse Meißen	Sparkasse_Meißen
Sparkass Memmingen-Lindau-Mindel e。	Sparkasse_Memmingen-Lindau-Mindel。
Sparkasse Merzig-Wadern	Sparkasse_Merzig-Wadern
Sparkasse Meschede	Sparkasse_Meschede
Sparkasse Miltenberg-Obernburg	Sparkasse_米尔滕伯格-奥伯恩堡
Sparkasse Minden-Lubbecke	Sparkasse_Minden-Lubbecke
Sparkasse Mittelfranken-Sud	Sparkasse_Mittelfranken-Sud
Sparkasse Mittelholstein AG	Sparkasse_mittelholstein_ag
Sparkasse Mittelmosel-Eifel	Sparkasse_Mittelmosel-Eifel

名称	ID
Sparkasse 中萨克森银行	Sparkasse_中萨克森州
Sparkasse Mittelthüringen	Sparkasse_Mittelthüringen
Sparkasse Muldent	Sparkasse_Mulden
Sparkasse Markisch-Oderland	Sparkasse_Markisch-Oderland
Sparkasse 鲁尔河畔米尔海姆	Sparkasse_mülheim_an_der_Ruhr
Sparkasse Munden	储蓄银行_明登
明斯特兰东部Sparkasse	Sparkasse_Münsterland_O
Sparkasse Neckartal-Odenwald	Sparkasse_Necartal-Odenwald
Sparkasse Neu-Ulm-Illertissen	Sparkasse_Neu-Ulm_-_Illertissen
新勃兰登堡 Sparkasse-Demm	Sparkasse_Bembrandenburg_-_Demmin
Sparkasse Neuburg-Rain	Sparkasse_Neuburg-Ra
Sparkasse Neumarkt-Parsber	Sparkasse_Neumarkt-Parsb
Sparkasse Neunkirchen	Sparkasse_Neunkirch
Sparkasse Neuss	Sparkasse_Neuss
Sparkasse Neustadt a. d. Aisch	Sparkasse_neustadt_a._d._aisch
Sparkasse Neuwied	Sparkasse_Neuwied
Sparkasse Niederbayern-Mitte	Sparkasse_Niederbayern-Mitte
Sparkasse Niederlausitz	Sparkasse_Niederlausitz
Sparkasse Nieburg	Sparkasse_Niebur
Sparkasse Nördlingen	Sparkasse_Nördlinge

名称	ID
纽伦堡储蓄银行	纽伦堡储蓄银行
Sparkasse Oberhesse	Sparkasse_Oberhessen
Sparkasse 高地	Sparkasse_Oberl
Sparkasse Oberpfalz Nord	Sparkasse_Oberpfalz_Nord
Sparkasse Odenwaldkreis	Sparkasse_Odenwaldkreis
Sparkasse Oder-Spree	Sparkasse_Oder-Spree
Sparkasse 奥芬堡/奥尔特瑙	Sparkasse_Offenburg/Orten
Sparkasse Olpe-Drolshagen-Wenden	Sparkasse_Olpe-Drolshagen-Wenden
Sparkasse Osnabrück	Sparkasse_Osnabrück
Sparkasse Osterode am Harz	sparkasse_osterode_am_Harz
Sparkasse Ostprignitz-Ruppin	Sparkasse_Ostprignitz-Ruppin
Sparkasse Ostunterfranken	Sparkasse_Ostunterfranken
Sparkasse 帕德博恩-德特莫尔德	Sparkasse_帕德博恩-德特莫尔德
Sparkasse Parchim-Lubz	Sparkasse_Parchim-Lubz
Sparkasse Passau	Sparkasse_Passa
Sparkasse Pfaffenhofen	Sparkasse_Pfaffenhofen
Sparkasse Partheim	Sparkasse_porforzheim_Calw
Sparkasse Pfullendorf-Meßkirch	Sparkasse_Pfullendorf-Meßkirch
Sparkasse Prignitz	Sparkasse_Prignitz
Sparkasse Radevormwald-Hückeswagen	Sparkasse_radevormwald-Hückeswagen

名称	ID
Sparkasse Rastatt-Gernsbach	Sparkasse_rastatt-Gernsbach
Sparkasse Regen-Viechtach	Sparkasse_Regen-Viechtach
Sparkasse Regensburg	Sparkasse_雷根斯堡
Sparkasse Rhein Neckar 北部	Sparkasse_Rhein_Neckar_Nord
Sparkasse Rhein-Haardt	Sparkasse_Rhein-Haardt
Sparkasse Rhein-Maas	Sparkasse_Rhein-Maas
Sparkasse Rhein-Nahe	Sparkasse_Rhein-Nah
Sparkasse Rheine	Sparkasse_Reine
Sparkasse Rietberg	Sparkasse_Rietberg
Sparkasse Rosenheim-Bad Aiblin	Sparkasse_Rosenheim-Bad_Aibling
Sparkasse Rotenburg-Bremervorde	Sparkasse_Rotenburg-Bremervorde
Sparkasse Rothenburg der Tauber	Sparkasse_rothenburg_ob_der_Tauber
Sparkasse Rottal-Inn	Sparkasse_Rottal-Inn
Sparkasse Saarbrücken	Sparkasse_Saarbrücken
Sparkasse Salem-Heiligenberg	Sparkasse_Salem-Heiligenberg
Schaumburg Sparkasse	Sparkasse_Schaumburg
Sparkasse Scheeßel	Sparkasse_Scheeßel
Sparkasse Schopfheim-Zell	Sparkasse_Schopfheim-Zell
Sparkasse Schwarzwald-Baar	Sparkasse_Schwarzwald-Baar
Sparkasse Schweinfurt	Sparkasse_Schweinfurt

名称	ID
Sparkasse Schwerte	Sparkasse_Schwerte
Sparkasse Siegen	Sparkasse_Siegen
Sparkasse Singen-Radolfzell	Sparkasse_Singen_-_Radolfzell
Sparkasse Soest	Sparkasse_Soest
Sparkasse Sonneberg	Sparkasse_Sonneberg
Sparkasse Spree-Neiße	Sparkasse_Spree-Neiße
Sparkasse St. Blasien	Sparkasse_St._Blasien
Sparkasse Stade-Altes Land	Sparkasse_Stade-Altes_Land
Sparkasse Starkenburg	Sparkasse_Starkenbur
Sparkasse Staufen-Breisach	Sparkasse_Staufen-Breisach
Sparkasse Stockach	Sparkasse_Stocach
Sparkasse Südholstein	Sparkasse_Südholstein
Sparkasse Südliche Weinstraße	Sparkasse_Südliche_Weinstraße
Sparkasse Südwestpfalz	Sparkasse_Südwestpfalz
Sparkasse 陶伯弗兰肯	Sparkasse_Tauberfrank
Sparkasse Trier	Sparkasse_Trier
Sparkasse Uckermark	Sparkass_Uckermark
Sparkasse Uecker-Randow	Sparkasse_Uecker-Randow
Sparkasse Uelzen Lüchow-Dannenberg	Sparkasse_Uelzen_Lüchow-Dannenberg
Sparkasse Ulm	Sparkasse_Ulm

名称	ID
Sparkasse UnnaKamen	Sparkasse_ UnnaKamen
Sparkasse Unstrut-Hainich	Sparkasse_Unstrut-Hainich
Sparkasse Vest 雷克林豪森	Sparkasse_Vest_Reckinghausen
Sparkasse Vogtland	Sparkasse_Vogtlan
Sparkasse Vorderpfalz	Sparkasse_Vorderpfalz
Sparkasse Vorpommern	Sparkasse_Vorpommern
Sparkasse Waldeck-Frankenberg	Sparkasse_Waldeck-Frankenberg
Sparkasse Werl	Sparkasse_Werl
Sparkasse Werra-Meißner	Sparkasse_Werra-Meißner
Sparkasse Weserbergland	Sparkasse_Weserberglan
Sparkasse Westerwald-Sieg	Sparkasse_Westerwald-Sieg
Sparkasse Westholstein	Sparkasse_Westholstein
Sparkasse Westmunsterland	Sparkasse_威斯敏斯特兰
Sparkasse Wetzlar	Sparkasse_Wetzlar
Sparkasse Wiesental	Sparkasse_Wiesental
威廉港储蓄银行	储蓄银行_威廉港
Sparkasse Witten	Sparkass_Witten
Sparkasse Wittenb	Sparkasse_Witten
Sparkasse Wittgenstein	Sparkasse_Wittgenstein
Sparkasse Wolfach	Sparkasse_Wolfach

名称	ID
Sparkasse Worms-Alzey-Ried	Sparkasse_Worms-Alzey-Ried
Sparkasse Zollernalb	Sparkasse_Zollernalb
Sparkasse Zwickau	Sparkasse_Zwickau
Sparkasse am Niederrhein	Sparkasse_am_niederRhein
Sparkasse an der Lippe	Sparkasse_an_der_Lippe
Sparkasse d. 洪堡。Gemeinden	Sparkasse_d._Homburg。 _Gemeinden
Sparkasse der Stadt Iserlohn	Sparkasse_der_stadt_iserlohn
Sparkasse der Stadt Straelen	Sparkasse_der_stadt_stadt_Strael
Sparkasse im Landkreis Cham	sparkasse_im_landkreis_Cham
施万多夫县的储蓄银行	sparkasse_im_landkreis_Schwandorf
吕贝克股份公司的 Sparkasse	Sparkasse_zu_lübeck_ag
SparkleWash	SparkleWash
斯帕克波特	斯帕克波特
Sparky's	Sparky's
Sparkz	Sparkz
斯巴达克	斯巴达克
斯巴达人	斯巴达人
Speakwell 技能学院	Speakwell_Skills
特别现金大奖	Special_Cash_Sw
特别顾问	特别顾问

名称	ID
专业	专业
Specsavers	Specsavers
奇观小屋	Spectacle_H
Spectra	Spectra
Spectrum	Spectrum
频谱移动	频谱_移动
Spedini Trattoria Express	Spedini_Trattoria_Expressa
SpeeDee	SpeeDee
极速汉堡	Speed_Burg
极速车	Speed_Car
极速女王	速度_Queen
极速兔子披萨	Speed_Rabbit_Pizz
SpeedMate	SpeedMate
Speedex	Speedex
摩托车赛	摩托车赛
赛车场石油公司	赛车场_石油
快速	快速
快速汽车服务	Speed_Auto_Serv
快速配送快递	快速配送_快递
快捷玻璃	Speed_Glas

名称	ID
快速标志	Speed_Sign
Speedywash	Speedywash
Spejo'S	Spejo'S
Spektr	Spektr
Spencer's	Spencer's
斯宾塞的零售	Spencer's 零售
免花钱鞋	少花钱鞋子
Spf	Spf
Spherion	Spherion
Spice	Spice
SpiceIce	SpiceIce
辛辣	辛辣
Spicy Pickle	Spicy_Pickle
Spiele Max	Spiele_Max
菠菜	菠菜
Spinelli 咖啡公司	尖晶石咖啡公司
Spinney	Spinney
Spinneys 酒	Spinneys_Liqu
Spinx	Spinx
Spire CU	Spire_cu

名称	ID
精神万圣节	Spirit_万圣节
Spirit 儿童英语	Spirit_Kids_英语
Spk。Mark。Sauerland Hemer-Menden	Spk。 _mark。 _Sauerland_Hemer-Menden
Spk。Oberlausitz-Niederschlesien	Spk。 _Oberlausitz-Niederschlesien
Spk。Schwébisch Hall-克赖尔斯海姆	Spk。 _Schwébisch_Hall_-_克赖尔斯海姆
飞溅	飞溅
Splash 婴儿水疗中心	Splash_Baby_Spa
Spok	Spok
Spolem	Spolem
斯波莱托	斯波莱托
Sport 2000	Sport_2000
Sport 24	Sport_24
运动木屋	运动小木屋
运动剪辑	运动剪辑
体育俱乐部	体育俱乐部
体育奥特莱斯	体育奥特莱斯
Sport Pl	运动版_Plus
街头运动	Sport_in_Street
SportScheck	SportScheck
SportXS	SportXS

名称	ID
Sportchek	Sportchek
Sportful Garden	Sportful_Garden_Restaur
Sportigan	Sportigan
Sportlandiya	Sportlandiya
Sportlife	Sportlife
Sportmaster	Sportmaster
体育管理局	体育权威
体育专家	体育_专家
体育迷-阁楼	Sports_Fan-Attic
体育狂热	Sports_Fe
体育 Kamui	Sports_Kamui
体育力量	运动_力量
体育站	体育站
体育多多	Sports_Toto
体育世界	体育_世界
SportsPower	SportsPower
体育公司	体育公司
运动工艺	运动工艺
Sportsdire	Sportsdire
女运动员	女运动员

名称	ID
运动员仓库	运动员_仓库
Sportson	Sportson
Spot 实例	Spot 实例
聚光灯	聚光灯
Spreewaldbank	Spreewaldbank
Sprey	Sprey
Sprider 商店	Sprider_Stores
春季银行	春季银行
春季教育集团	春季教育集团
春季酒店	春季酒店
春步	弹簧_Step
春天的时候	春天_时间
春绿	春绿
SpringHill 万豪套房	SpringHill_Suites_by_marriott
春叶金融服务	Springleaf_金融_服务
Sprint	Sprint
Sprint M	Sprint_Mart
短跑运动员	短跑运动员
Sprouts 农贸市场	豆芽_农贸市场
Spudbar	Spudbar

名称	ID
马刺	马刺
Spur Steak 牧场	Spur_Steak_Ranches
Spur 牛排和烧烤	Spur_Steak_and_Grill
Spyder Rent	Spyder_Rent
Spykar	Spykar
Square D 公司	Square_D_Company
Frango da Guia 先生	sr._frango_da_Guia
Sree Annapoorna	Sree_Annapoorna
Sree Kumaran Thangamaligai	Sree_Kumaran_Thangamaligai
Sri Chaitanya 学校	Sri_Chaitanya_Schools
斯里克里希纳糖果	Sri_Krishna_Sweets
Srisawad Power 1979	Srisawad_Power_1979
SsangYong	SsangYong
St Elmo's 柴火披萨	st_elmo's_woodfired_Pizza
圣乔治	St_George
圣劳伦斯 FCU	st_Lawrence_FCU
圣路易斯酒吧和烧烤店	St_Louis_Bar_&Grill
圣路易斯面包公司	St_Louis_Bread_Co
St Paul FCU	st_paul_fcu
圣皮埃尔寿司	st_Pierre'sushi

名称	ID
圣文森特德保罗	St_Vincent_de_Paul
St Wendeler Vb	St_Wendeler_Vb
圣休伯特	圣休伯特
圣加勒坎托纳尔银行	St._Galler_KantonalBank
圣文森特德保罗	St._Vincent_de_Paul
Stach	Stach
Stadt-u。kr.spk Erlangan Höchstadt	stadt-_u._kr.spk_Erlangen_Höchstadt
Moosburg Stadt-and Kreissparkasse	Stadt-_und_kreissparkasse_Moosburg
Stadt-Sparkasse Haan（莱茵尔。）	Stadt-Sparkasse_Haan_（Rheinl。）
朗根费尔德Stadt-Sparkasse	Stadt-Sparkasse_朗根费尔德
索林根市储蓄银行	Stadt-Sparkasse_索林根
汉堡 StadTrad	Stadtrad_Hamburg
Stadtbackerei Gatenbröcker	Stadtbackerei_Gatenbröcker
Stadtbackerei Schaller	Stadtbackerei_Schaller
奥格斯堡城市公园	奥格斯堡城市公园
Stadtsparkasse Bad Honnef	Stadtsparkasse_Bad_Honnef
Stadtsparkasse Bad Oeynhausen	Stadtsparkasse_Bad_Oeynhausen
Stadtsparkasse Bad Pyrmont	Stadtsparkasse_Bad_Pyrmont
巴特萨克萨市公园大楼	Stadtsparkasse_Bad_Sachsa
Stadtsparkasse Barsinghausen	Stadtsparkasse_Barsinghausen

名称	ID
Stadtsparkasse Blomberg/Lippe	Stadtsparkasse_Blomberg/Lippe
Stadtsparkasse Bocholt	Stadtsparkasse_Bocholt
Borken Stadtsparkasse（黑森）	Stadtsparkasse_Borken_（黑森）
布格多夫城市公园大厅	Stadtsparkasse_Burgdorf
库克斯港城市公园	Stadtparkasse_Cuxhaven
德尔布吕克城市公园	Stadtsparkasse_Delbrück
Dessau Stadtsparkasse	Stadtparkasse_Dessau
杜塞尔多夫城市公园	杜塞尔多夫城市公园
费尔斯伯格城市公园	Stadtsparkasse_Felsberg
弗洛伊登贝格城市公园	Stadtsparkasse_Freudenberg
格拉德贝克Stadtsparkasse	格拉德贝克城市公园
Stadtsparkasse Grebenstein	Stadtparkasse_Grebenstein
Stadtsparkasse Haltern am See	Stadtsparkasse_Haltern_am_see
Stadtsparkasse Herdecke	Stadtsparkasse_Herdecke
希尔兴巴赫城市公园大厅	Stadtparkasse_Hilchenbach
凯撒斯劳滕城市公园	Stadtsparkasse_Kaiserslautern
Stadtsparkasse Lengerich	Stadtsparkasse_Lengerich
马格德堡城市公园	Stadtsparkasse_Magdeburg
门兴格拉德巴赫城市公园	Stadtparkasse_Mönchengladbach
慕尼黑城市公园	慕尼黑城市公园_慕尼黑

名称	ID
奥伯豪森市立公园	奥伯豪森城市公园
Stadtsparkasse Porta Westfalic	Stadtsparkasse_Porta_Westfalica
Stadtsparkasse Rahden	Stadtsparkasse_Rahden
Stadtsparkasse Remscheid	Stadtsparkasse_Remscheid
施马伦贝格城市公园	Stadtparkasse_Schmallenberg
施瓦尔姆施塔特城市公园大楼	Stadtsparkasse_Schwalmstadt
Stadtsparkasse Schwedt	Stadtsparkasse_Schwedt
Stadtsparkasse Schwelm	Stadtsparkasse_Schwelm
Stadtsparkasse Sprockhövel	Stadtsparkasse_Sprockhövel
费尔斯莫尔德Stadtsparkasse	Stadtparkasse_Versmol
弗尔克林根城市公园大楼	Stadtparkasse_Völklingen
Stadtsparkasse Wedel	Stadtsparkasse_Wedel
Stadtsparkasse Wermelskirchen	Stadtsparkasse_Wermelskirchen
Stadtsparkasse Werne	Stadtsparkasse_Werne
Stadtsparkasse 天气（鲁尔）	Stadtsparkasse_Wetter_ (Ruhr)
Stadtsparkasse Wunstorf	Stadtsparkasse_Wunstorf
伍珀塔尔Stadtsparkasse	Stadtsparkasse_Wuppertal
斯塔福德郡	斯塔福德郡
Stakis	Stakis
斯坦比克银行	Stanbic_Bank

名称	ID
Stanbic IBTC 银行	Stanbic_ibtc_Bank
Standaard Bookhandel	Standaard_Bookhandel
Standard	Standard
标准银行	标准银行
标准银行和信托	标准银行和信托
渣打银行	渣打银行
标准联邦银行	标准联邦银行
莱索托标准银行	莱索托标准银行
标准停车	标准_停车
标准管道和供应	标准_管道_和_供应
标准寄存器	标准_注册
斯坦福 FCU	Stanford_FCU
Stangengrüner Mühlenbackerei	Stangengrüner_Mühlenbackerei
斯坦利	斯坦利
Stanley Steemer	Stanley_Steemer
Stanleybet	Stanleybet
订书钉	订书钉
斯台普斯商业仓库	Staples_Busines_Dep
明星	明星
星际冲锋	Star_Charge

名称	ID
明星影院	Star_Cinema
Star Health 与联合保险	Star_Health_&_Allied_Insurance
Star Inn	Star_Inn
明星市场	明星市场
星际超市	Star_Mart
星形指甲	Star_Nails
星油	Star_Oil
星际停止	Star_Stop
星月时尚旅馆	Star-Moon_Fashion_Inn
Star-Osco	Star-Osco
StarHub	StarHub
Stara Mydlarnia	Stara_Mydlarnia
星巴克	星巴克
Starfleur	Starfleur
星际酒店	星际酒店
Stariy Lekar'	Stariy_Lekar'
小明星	小明星
Starpet	Starpet
繁星点点的街道	Starry_Street
Stars Drive-in	Stars_drive-in

名称	ID
星际酒店	Stars_酒店
Starschnitt	Starschnitt
创业咖啡	初创公司_Coffee
Starvin Marvin	Starvin_Marvin
喜达屋	喜达屋
州和主要	State_&_Main
国家银行	州立银行
印度国家银行	印度国家银行
国务院 FCU	国务院_FCU
国家雇员 CU	州雇员_cu
州雇员信用合作社	州雇员信用联盟
州立农场	State_Farm
国家生活	状态_生活
道富银行	State_Street_Bank
Stater Bros	Stater_Bros
全州 FCU	全州_FCU
一号站咖啡厅	Station_1_Cafe
Kopitiam 站	车站_Kopitiam
车站服务 Sonic	电台服务_Sonic
蒙特利尔车站	Stationnement_de_Montréal

名称	ID
挪威国家石油公司	挪威国家石油公司
挪威国家石油公司 1-2-3	挪威国家石油公司_1-2-3
Staut	Staut
Stax	Stax
Stax 家庭娱乐	Stax_Home_娱乐
Stay N Save	Stay_N_Save
斯泰布里奇套房酒店	Staybridge Suites
Stayeasy	Stayeasy
斯特德和辛普森	Stead_&_Simpson
逃出牛排	Steak_escape
牛排馆比利小子	Steak_House_Billy_The_Ki
牛排肺坚果	Steak_Lung_Nuat
牛排 Miya	Steak_Miya
牛排 n Shake	Steak_n_Shake
SteakOut	SteakOut
蒸汽学院	蒸汽学院
蒸汽的	蒸汽的
转向	转向
Steigenberger	Steigenberger
Steimatzky	Steimatzky

名称	ID
Steiskal	Steiskal
Stella Empotrados	Stella_Empotrados
斯特拉 McCartney	Stella_ McCartney
恒星国际影城	恒星国际_Cineplex
Stels	Stels
Stemma	Stemma
Stender	Stender
Stenders	Stenders
阶梯地板拼花地板	阶梯_地板_实木复合地板
踏板运动	Step_Sport
循序渐进	Step_by_step
Steph & Andy's Pizza	Steph_&_andy's pizza_Plus
垫脚石书店	stepping_stone_bookstore
步数	步数
Steptoe & Jo	Steptoe_&_John
星际世纪	星际世纪
明星电影院	明星电影院
Ster-Kinek	Ster-Kinek
斯特伦	斯特伦
立体声工作室	立体声工作室

名称	ID
斯特林银行	Sterling_Bank
斯特林生物	Sterling_Bio
斯特林国家银行	英镑国民银行
斯特林光学	英镑_光学
英镑储蓄银行	Sterling_Savings_Bank
斯特恩	斯特恩
Sternenback	Sternenback
史蒂夫·马登	Steve_Madden
史蒂文斯	史蒂文斯
Stevi B 的披萨自助餐	Stevi_b's_pizza_Buffet
管家银行	Steward_Bank
斯图尔特的	斯图尔特的
Steart's Drive-	Stewart's_drive-in
斯图尔特的商店	Stewart's Shops
Sticks N Sus	Sticks_N_Sushi
小偷	Sticky_Finger
Stiefelkönig	Stiefelkönig
斯蒂菲尔金融	Stifel_Financ
Stihl Shop	Stihl_Shop
Stinges Landbackerei	Stinges_Landbackerei

名称	ID
Stingpa	Stingpa
Stinker	Stinker
斯汀森	斯汀森
疯狂地搅拌	Stir_Crazy
斯特林体育	斯特林_体育
缝合	缝合
申通快递	Sto_Expr
斯德哥尔摩停车场	斯德哥尔摩_Parkering
斯托克兰	斯托克兰
Stof 2000	Stof_2000
股票中心	库存中心
Stock-Tsentr	Stock-Tsentr
Stokrotka	Stokrotka
Stoletnik	Stoletnik
Stolichki	Stolichki
Stolichniy	Stolichniy
Stolichnye Apteki	Stolichnye_Apteki
Stolle	Stolle
Stolovaya s'em Slona	stolovaya_s'em_slona
Stolplit	Stolplit

名称	ID
Ya 的口腔科医生	Somatolog_i_ya
石烧肉	石头烧肉
石器时代	石器时代
石头	石头
停下来购物	Stop_&_Shop
停止购物	Stop_Shop
停止旅行	停止旅行
停下来	Stop_in
Stop-N-Go	Stop-N-Go
Stop-N-Save	Stop-N-Save
Stora coop	Stora_coop
存储快递	存储_Express
存储之王	Storage_King
西部储藏室	Storage_West
二十一号商店	Store_Twenty_One
斯托尔姆伯格	斯托尔姆伯格
StoryWay	StoryWay
Stouffer	Stouffer
斯特拉达	斯特拉达
斯特拉迪瓦里斯	斯特拉迪瓦里斯

名称	ID
Strakhovaya Gruppa Sogaz	Strakhovaya_Gruppa_Sogaz
沙滩包	沙滩包
陌生人火锅餐厅	Stranger_Hot_Pot_Restauran
斯特拉斯堡儿童	斯特拉斯堡_孩子
斯特拉特福德豪斯旅馆	斯特拉特福德豪斯旅馆
斯特拉斯菲尔德汽车收音机	斯特拉斯菲尔德_汽车电台
斯特拉托斯披萨店	Stratos_Pizzeria
草帽披萨	草帽披萨
草莓甜筒	Strawberry_Cones
街角	街角
街头狂热	街头狂热
第一街	Street_One
StreetHouse	StreetHouse
纽约街头	纽约街头
Stretch-n Grow	Stretch-n Grow
Stride Rite	Stride_Rite
打击保龄球	Strike_Bowlin
条纹	条纹
Stroh	Stroh
Stroili Oro	Stroili_Oro

名称	ID
StrollerFit 专家	StrollerFit_专家
斯特劳德和史云顿建筑协会	Stroud_&_Swindon_Building_Society
Stuckey's	Stuckey's
学生航班	学生机票
Studio	Studio
工作室 6	Studio_6
工作室 A	Studio_A
卡拉特工作室	Studio_Caratt
工作室 F	Studio_F
Studio11	Studio11
StudioPl	StudioPl
Studium Bosküchen	Studium_Bosküchen
Studiya manikyura Leny Leninoy	studiya_manikyura_leny_leninoy
Janak 的研究	Study_by_Janak
Stuttafords	Stuttafords
美国风格	风格_美国
时尚咖啡厅	风格_咖啡厅
少花钱的款式	Styles_for_less
Stylo	Stylo
Stádtische Sparkasse Offenbach	Stádtische_Sparkasse_Offenbach

名称	ID
Stádtische Sparkasse zu Schwelm	stádtische_sparkasse_zu_schwelm
SuNing	SuNing
SuShi 牛肉面	SuShi_Beef_Noodle
二号分站	变电站_II
Subai.Yun 酒店	Subai.Yun_Hotel
Subhiksha	Subhiksha
Subhiksha 药房	Subhiksha_药房
Submarina 加州潜艇	加州潜艇潜艇
郊区诊断	郊区诊断
郊区长住酒店	郊区_延长_住宿
郊区丙烷	郊区_丙烷
郊区	郊区
Suco Bagaço	Suco_Bagaço
南方快车	SUD_Exp
Sudameris 银行	Sudameris_Bank
Sudar'	Sudar'
突发服务	突发服务
SuddenLink 通讯	SuddenLink_通讯
sudostroitel'niy Bank	sudostroitel'niy_Bank
苏伊士运河银行	Suez_Canal_Bank

名称	ID
萨福克 FCU	suffolk_fcu
Sufood	Sufood
菅木屋	菅木屋
巢鸭信用金库	巢鸭信用金库
糖	糖
糖包	糖包
糖和香料	Sugar_n_Spice
SugarClub	SugarClub
甘蔗妈妈	Sugarcane_Mama
舒格克里克专卖店	舒格克里克_Stores
杉乃屋本阵	Suginoya_Honjin
Sugiyama Chozai Yakkyok	Sugiyama_chozai_Yakkyoku
苏杭时光超市	Suhang_Time_Mart
Suhyub Bank	suhyub_Bank
Suidie Music 的咖啡	suidie_music's Coffee
遂宁市商业银行	遂宁城市商业银行
诺富特套房	套房_诺富特
套房酒店	套房酒店
Suitemark	Suitemark
Paxton Suiton Plus 酒店	Paxton_Plus_By_Paxton_Hotel

名称	ID
很适合你	西装_你
遂溪农村商业银行	遂溪农村商业银行
Sukh Sagar	Sukh_Sagar
Sukishi 韩国木炭烧烤	Sukishi_Korean_Charcoal_Grill
Sukishi 东京自助餐	寿喜_东京_自助餐
Sulland	Sulland
沙利文·尼古拉德斯病理学	Sullivan_Nicolaides _病理学
沙利文的牛排馆	Sullivan's SteakHouse
Sulpak	Sulpak
苏丹	苏丹
Sultanahmet Koftecisi	Sultanahmet_Koftecisi
须磨	须磨
苏门答腊	苏门答腊
炭火串烧天狗酒场	Sumibi_kushiyaki_Tengu_Sakaba
炭火烤肉七轮坊	Sumibi_烤肉_七轮坊
炭火烧肉屋酒井	炭火烧肉屋_Sakai
炭烧餐厅 Sawayaka	Sumiyaki_餐厅_Sawayaka
峰会银行	Summit_Bank
萨米特 CU	Summit_cu
相扑	相扑

名称	ID
相扑烧烤	Sumo_bbq
相扑沙拉	Sumo_Salad
SumoSam	SumoSam
太阳	太阳
阳光与沙滩	Sun_&_Sand
孙东宝	孙东宝
Sun East FCU	sun_east_fcu
太阳花	太阳花
太阳国际	太阳国际
《太阳人生》	Sun_Life
永明光大人寿光大人寿	太阳生活_Everbright_Life
永明金融	Sun_Life_财经
太阳贷款公司	Sun_Loan_Company
太阳快乐	Sun_Merry
太阳国民银行	太阳国民银行
防晒油	Sun_Oil
太阳生物	Sun_Ogarnism
太阳药业	太阳药业
太阳猪	Sun_Pig
太阳学校	Sun_School

名称	ID
孙尚乔夫	孙尚乔夫
新滩市	Sun_tan_City
孙氏炸饺子	Sun's fried_Dumpling
太阳有机主义	太阳有机主义
SunAmerica	SunAmerica
SunStop 门店	SunStop_Stores
SunWest 教育 CU	SunWest_Educatial_cu
SunWest FCU	SunWest_FCU
太阳带	太阳带
Sunbelt 积分	Sunbelt_Cred
申博外国语学校	Sunbo_外语_学校
太阳船教育	Sunboat_教育
阳光海岸电影公司	Suncoast_Motion_Picture_Co
Suncorp	Suncorp
圣丹斯电影节康复	圣丹斯_康复
Sundara	Sundara
Sundi	Sundi
Sundowner 国际	Sundowner_国际
Sunesy	Sunesy
太阳镜小屋	太阳镜_Hut

名称	ID
Sunglo	Sunglo
太阳酒店	太阳酒店
苏宁中信	苏宁_Citicall
苏宁广场	苏宁广场
苏宁店	苏宁_Shop
Sunkost	Sunkost
阳光	阳光
海军陆战队	海军陆战队
Sunmark CU	sunmark_cu
太阳快乐	太阳快乐
阳光明媚	阳光明媚
阳光电器	阳光明媚的家电
阳光银行	Sunny_Bank
Sunny 部门	阳光部门
阳光之王幼儿园	Sunny_kings_幼儿园
Sunoco	Sunoco
Sunoco 加拿大	Sunoco_Canada
Sunpet	Sunpet
Sunpet 市场	Sunpet_Marke
太阳广场	太阳广场

名称	ID
Sunra	Sunra
日出	日出
日出海滩酒店	日出海滩酒店
Sunrise 老年生活	Sunrise_Senior_Living
日出之旅	日出之旅
太阳河茶	Sunriver_Tea
日落大道	日落大道
Sunshine	Sunshine
阳光保险	Sunshine_保险
阳光超市	阳光_超市
日星酒店	Sunstar_酒店
太阳石广场酒店1	Sunstone_Plaza_Hote1
三和角川	Sunwah_Kadokawa
Sunya Food	Sunya_Food
纽约州立大学	纽约州立大学
苏纽基	苏纽基
Sunèlia	Sunèlia
索兴精选酒店	索兴_精选_酒店
索兴宾馆	索兴_酒店
快速	Supa_Qu

名称	ID
Supa Valu	SupaValu
Supanews	Supanews
超级	超级
超级 1 食物	Super_1_Foods
Super 1 食品和折扣药房	超级_1_Foods_&_Discount_Pharmacy
Super 10	Super_10
Super 7	Super_7
Super 8	Super_8
Super 8 尊尚酒店	Super_8_酒店_Premier
超级阿尔卑斯山	超级阿尔卑斯山
超级 BM	super_bm
超级酒窖	Super_Bodega
好莱坞超级博蒂卡斯	超级博蒂卡斯_好莱坞
Super C	Super_C
超级 CERKA	Super_cerka
超级护理药房	超级护理药房
超级鸡	超级鸡肉
超级城市	超级城市
超级美元	超级美元
超级毒品 Asahi	Super_Drug_Asahi

名称	ID
超级精英	超级精英
超级气体	超级气体
Super Gaz	Super_Gaz
超级 Gb	Super_Gb
超级普通韩国餐厅	Super_General_Korean _餐厅
超级ISSSTE	super_issste
超级印度	Super_Indo
Super Kompras	Super_Kompras
超级酒	超级烈酒
超级市场	超级市场
超级松本	超级_松本
超级马法托	Super_Muff
Super Nosso	Super_Nosso
超级一食品	Super_One_Foo
超级伙伴 Gb	超级合作伙伴_Gb
Super Poupadinha	Super_Poupadinha
超级力量	超级力量
超级价格	Super_Preci
超级春天	Super_Primavera
超级索科	super_soco

名称	ID
超级沙拉	Super_Salads
超级保存	超级保存
超级省钱方便	超级保存便利
超级商店	超级商店
超级明星	超级明星
超级待办事项	Super_Todo
Super U	Super_U
超级价值	超级价值
Super Vea	Super_Vea
超级清洗	超级清洗
超级炸弹	super_bompreço
Super de Boer	Super_de_Boer
超级北方	Super_del_Nort
Super-letto	Super-letto
SuperAmerica	SuperAmerica
SuperBioMarkt	SuperBioMarkt
SuperBodega Acuenta	SuperBodega_aCuenta
SuperCheap	SuperCheap
SuperCoups	SuperCoups
SuperFarma	SuperFarma

名称	ID
SuperFarmacias ezquivar	SuperFarmacias_ezquivar
SuperGlass 挡风玻璃维修	SuperGlass_挡风玻璃_维修
SuperKids	SuperKids
SuperMag	SuperMag
SuperPumper	SuperPumper
SuperSol	SuperSol
SuperStep	SuperStep
SuperStroy	SuperStroy
SuperValu	SuperValu
超级大战	超级大战
Superavui	Superavui
超级最佳	超级最佳
超级布鲁格森	超级布鲁格森
超级便宜的汽车	Supercheap_Auto
Supercoop	Supercoop
超级剪辑	超级剪辑
超级毒品	超级毒品
超级干燥	超级干燥
Superestalvi	Superestalvi
超级新鲜	超级新鲜

名称	ID
超级新鲜药房	超级新鲜药_药房
超级巨星	超级巨星
卓越银行	Superior_Ban
Superior	superior_cu_inc
卓越之选 CU	Superior_choice_cu
卓越的货车和机动性	卓越的货车和出行性
Superm	Superm
超人	超人
Supermac 的家庭餐厅	Supermac's_Family_Re
Supermac 的新鲜快车	Supermac's_fresh_E
超市	超市
三德超市	超市_三德
超级市场	超级市场
Supermercado Dani	Supermercado_Dani
Supermercado Koch	Supermercado_Koch
Supermercado LM	Supermercado_LM
圣维森特超级市场	Supermercado_Sao_Vicente
Supermercados Alvorada	Supermercados_Alvorada
超级市场大道	Supermercados_Avenida
超级市场 Bh	Supermercados_Bh

名称	ID
超级市场冠军	Supermercados_Campeão
Supermercados Cidade Canção	Supermercados_Cidade_Cancao
超级市场 Compre Bem	Supermercados_Compre_Bem
Supermercados El Jamon	Supermercados_El_Jamon
Hiber 超级市场	Supermercados_Hiber
超级市场 Imperatriz	Supermercados_Imperatriz
Supermercados Lola	Supermercados_Lola
超级市场模具	Supermercados_Moldes
世界超级市场	Supermercados_Mundial
超级市场 Todo Todo Todo Todo	Supermercados_Todo_Todo
马德里超级市场	Supermercados_de_Madrid
超级商人 Veritas	超级商场_华睿泰
超级猴子	超级猴子
超级内托	超级内托
Superpetz	Superpetz
超级制药	超级制药
Superpão	Superpão
超级奎因	超级奎因
Supersano	Supersano
超级巨星	超级巨星

名称	ID
超级阻止	超级阻止
超级科技	超级科技
超级动作	超级动作
超值	超值
Superviele Banco	Supervielle_Banco
Supol	Supol
Supper Rissul	Supper_Rissul
Supplyco	Supplyco
见上文	见上文
Supre	Supre
Supré	Supré
Surla Table	Sur_La_Table
Surat Kargo	Surat_Kargo
Surcouf	Surcouf
SureBank	SureBank
苏雷莫夫	苏雷莫夫
Surestay 酒店	Surestay_Hotel
Surestay Plu	Surestay_Plus_Hotel
冲浪城挤压	冲浪城市挤压
Surf Dive 'n 'Ski	Surf_dive_'n'_ski

名称	ID
冲浪救生	Surf_Life_Saving
Surgutneftegasbank	Surgutneftegasbank
Surgutneftegaz	Surgutneftegaz
剩余仓库	盈余_仓库
寿司快递	寿司_Express
寿司酒店	寿司_酒店
寿司 Itto	Sushi_Itto
寿司之王	Sushi_King
寿司大师	Sushi_Master
寿司卷	Sushi_Roll
寿司店	寿司店
寿司寿司	寿司_寿司
寿司亭	Sushi_Tei
寿司火车	Sushi_Train
寿司炒锅	Sushi_Wok
山寿司	Sushi_Yama
寿司 Zanmai	Sushi_Zanmai
寿司泰拉	寿司泰拉
SushiClub	SushiClub
SushiVosla	SushiVosla

名称	ID
Sushiaki	Sushiaki
寿司郎	寿司郎
寿司店	寿司店
寿司屋	寿司屋
萨斯奎哈纳银行	Susquehanna_Bank
苏桑	苏桑
Sutis	Sutis
Sutter Health	Sutter_Health
Sutterlüty	Sutterlüty
萨顿集团地产	Sutton_Group_Realty
萨顿广场	Sutton_Place
苏瓦纳萨哈卡里银行	Suvarna_Sahakari_Bank
Suwa 信用金库	Suwa_Shinkin_Bank
Suwannee Swifty	Suwannee_Swifty_Store
Suzanne Grae	Suzanne_Grae
苏州农村商业银行	苏州_农村_商业银行
Suzy Shier	Suzy_Shier
Sv。迈克尔森巧克力	Sv。 _Michelsen_Chokolade
Svejk 餐厅	svejk_restauran
Svensk Passaservice	Svensk_Kassaservice

名称	ID
Svetofor	Svetofor
Svg	Svg
Svilanit	Svilanit
Svitanok	Svitanok
Svoya Companiya	Svoya_Companiya
Svyaz'-Bank	Svyaz'-Bank
斯维亚兹诺伊	斯维亚兹诺伊
Svyaznoy Bank	Svyaznoy_Bank
Swagat	Swagat
燕子	燕子
天鹅面包店	天鹅面包店
Swaraj	Swaraj
施华洛世奇	施华洛世奇
斯威士兰银行	斯威士兰银行
斯威士兰建筑协会	斯威士兰_建筑学会
瑞典银行	瑞典银行
瑞典酒店	瑞典_酒店
甜蜜的	甜蜜的
甜蜜工厂	甜蜜工厂
甜番茄	甜番茄

名称	ID
Sweet7	Sweet7
甜蜜贝超市	Sweetbay_超市
甜绿	甜绿
Swensen's	Swensen's
Swiat Ksiazki	Swiat_Ksiazki
Swiat Spinek	Swiat_Spinek
Swiat Zdrowia	Swiat_Zdrowia
斯威夫蒂	斯威夫蒂
窸窸作响	窸窸作响
Swisher 卫生	Swisher_Hygies
瑞士阿拉伯人	瑞士人_阿拉伯人
瑞士肉店	Swiss_Butchery
瑞士城堡面包店	瑞士城堡面包店
瑞士木屋	Swiss_Chalet
瑞士豪华酒店	瑞士豪华酒店
瑞士国际酒店	瑞士_国际_酒店
瑞士领先酒店	瑞士领先酒店
瑞士优质酒店	瑞士品质酒店
瑞士时间	swiss_Time
瑞士贝尔酒店	瑞士贝尔酒店

名称	ID
瑞士酒店及度假村	瑞士酒店_酒店_度假村
Swissotel Ltd	Swissotel_Ltd
SwissTouches 酒店及度假村	Swisstouches _ 酒店及度假村
Switch	Switch
昏昏欲睡	昏昏欲睡
Syabuyo	Syabuyo
Sydbank	Sydbank
西尔万学习中心	希尔万_学习中心
西尔维·蒂里兹	Sylvie_Thiriez
研讨会咖啡厅	研讨会_咖啡厅
协同作用	协同作用
Synevo	Synevo
Synne	Synne
剧情简介	剧情简介
Synovus	Synovus
叙利亚黎巴嫩商业银行	叙利亚_黎巴嫩_商业银行
系统停车	系统_停车
Syun Hotel	Syun_Hotel
SZ-bike	SZ-bike
Szabolcs Takarék	Szabolcs_Takarék

名称	ID
Szatmari	Szatmari
Szatmar-Beregi Takarekszövetkezet	Szatmar-Beregi_Takarekszövetkezet
Szegvár 是 Vidéke Takarekszöv。	szegvár_és_vidéke_takarekszöv。
Szigetvári Takarék	Szigetvári_Takarék
SzimPatika	SzimPatika
SAIFA	SAIFA
圣布拉兹	Sao_Braz
Saestumarket	Saestumarket
Saystöpankki	Saystöpankki
Sergio Franco Medicina Diagnóstica	Sergio_Franco_Medicina_Diagnóstica
Sostrene Grene	Sèstrene_Grene
Super Aki	Super_Aki
西南银行	西南银行
Süvari	Süvari
T 银行	T_Bank
T C S	T_C_S
T 代表电信	T_For_Teleco
T 星	T_STAR
T 型酒吧	T_bar
T&T 商店	T&T_SHOP

名称	ID
T&T 超市	T&T_超市
T'Station	T'Station
T-Bank	T-Bank
T-Copia	T-Copia
T-JOIN	T-JOIN
T 市场	T 市场
T-Mobile	T-Mobile
T 型零件	T 型零件
T-Punkt	T-Punkt
霸王龙	霸王龙
T-tankstelle	T-tankstelle
T-Up 加油站	T-Up_Gas Station
T-box	T-box
T-cat	T-cat
T.G.C.	T.G.C.
T.I.M.E	T.I.M.E
T.J. Cinnamons	T.J._Cinnamons
T.J. Maxx	T.J._Maxx
T.O.P.24 h	t.o.p.24_h
T11 食品市场	T11_Food_Market

名称	ID
T2	T2
TA 快车	ta_Expr
TAB	TAB
TABLAPIZZA	TABLAPIZZA
塔可	塔可
太兴	TAI_HING
TAI JUM Shabu Shabu	tai_jum_Shabu_shabu
太极药房	太极_药房
大班	大班
台湾传统食品	台湾_传统食品
太延	太延
Tally WEi JL	TALLY_ JL WEi
TAMOIL	TAMOIL
唐宫海鲜餐厅	唐宫海鲜餐厅
唐家菜	唐家菜
坦克	坦克
陶然居	TAO_RAN_JU
塔皮奥拉	塔皮奥拉
TARANKO	TARANKO
TARGOBANK	TARGOBANK

名称	ID
TARJOUSTALO	TARJOUSTALO
TARMO LAYHIKAUPPIAS	TARMO_LAYHIKAUPPIAS
TAS	TAS
TAS 系列	TAS_Collec
味道	味道
TASTy	TASTy
塔塔	塔塔
TATA AIG	TATA_AIG
塔塔资本	塔塔资本
塔塔 Docomo	tata_docomo
塔塔钢铁	Tata_Steel
TATLISES CIG KOFTE	TATLISES_CIG_KOFTE
TATUUM	TATUUM
金牛座	金牛座
TAVRIIa V	TAVRIIa_V
税	税
TAYWIN	TAYWIN
待定	待定
TBK 银行	tbk_bank
TBS	TBS

名称	ID
TCBY	TCBY
TCC 鉴赏家协奏曲	TCC_THE_CONNOUSEUR_CONCERTO
TCE 铲球	TCE_Tackles
TCF	TCF
TCG	TCG
TCT	TCT
TCT NANO	TCT_NANO
道明银行	道明银行
道明加拿大信托	TD_Canada_Trust
TDM	TDM
TDS	TDS
TE 24/24	TE_24/24
茶的旋律	TEA_MELODY
TEA-TOP	TEA-TOP
SPORTIA 车队	TEAM_SPORTIA
茶店	茶店
TEB	TEB
TEBOIL	TEBOIL
TEBOIL 快车	TEBOIL_EXPRESS
TECATE SIX	TECATE_SIX

名称	ID
TECNO	TECNO
TECNOTALLER	TECNOTALLER
泰德·贝克	TED_BAKER
泰德·拉皮德斯	TED_LAPIDUS
TEDi	TEDi
TEISHOKU 8	TEISHOKU_8
TEKBIR	TEKBIR
TEKNIKMAGASINET	TEKNIKMAGASINET
TEKNISET	TEKNISET
特克森	特克森
TELA	TELA
TELE2	TELE2
TELKOM	TELKOM
TELMEX	TELMEX
TELo知识产权	TELo知识产权
TEMPO Kondela nabyt	tempo_kondela_Nabytok
TEMPUR	TEMPUR
诱惑	诱惑
TEN & CO	TEN_&_CO
十件内衣	TEN_LIGERY

名称	ID
十天	十天
TENTACION	TENTACION
TENTISSIMO	TENTISSIMO
TEQUENDAMA	TEQUENDAMA
泰拉健康俱乐部	TERA_健康俱乐部
TERGAN	TERGAN
泰拉诺瓦	泰拉诺瓦
TERRE BLEUE	TERRE_BLEUE
乐购额外	tesco_Extra
TESTA ROSSA	TEST_ROSSA
TETA drogerie	teta_drogerie
TETCO	TETCO
TEVAH	TEVAH
TEVOS Drogerie	tevos_drogerie
TEVOS Drogérie	tevos_drogérie
德士古	德士古
德州鸡	德州鸡肉
德州长角牛	TEXAS_LONGHORN
TEXTO	TEXTO
TEZ 之旅	TEZ_TOUR

名称	ID
天丰证券	TF_证券
TG FONE	TG_FONE
TGB 咖啡厅 N Bakery	tgb_cafe_n_Bakery
TGB 好汉堡	tgb_the_good_Burger
TGI Friday's	TGI_Friday's
TGV 电影院	TGV_Cinemas
泰国快递	泰国快递
泰国帕塔纳保险	泰国_PATTANA_保险
泰国人租车	泰国租车
泰国小镇美食	泰国小镇美食
泰斯里保险	泰国人_保险
谢	谢
THASIAM	THASIAM
主教的武器	THE_BISHOPS_ARMS
车身修理厂	THE_BODY_SHOP
江氏兄弟	姜氏兄弟
咖啡馆	THE_CAFFE
咖啡俱乐部	咖啡俱乐部
中国进出口银行	中国进出口银行
奶奶的	奶奶的

名称	ID
大吉林药店	THE_GREAT_GREAT_JILIN_MEDICINE_STORE
医疗保健特许经营权	医疗保健特许经营权
豪华精选酒店	奢华系列酒店
移动商店	THE_MOBILE_STOR
太平洋电影院	THE_PACIFIC_CINEPLEX
意大利面屋	THE_SPAGHETTI_HOUSE
工会繁荣保险	工会繁荣保险
VIRIYAH 保险	VIRIYAH_INSURANCE
温斯顿小酒馆	温斯顿小酒馆
THEFACESHOP	THEFACESHOP
理论	理论
热水瓶	热水瓶
这些晚会	thes_Gala
THI 系列	这个集合
托马斯·蒙兹	THOMAS_MUNZ
THON	THON
THREEPPY	THREEPPY
田木蓝鸟咖啡馆	TIAN_MU_BLUE_BIRD_CAFE
天水雅居	TIAN_SHUI_YA_JU
天元龙商场	TIAN_YUAN_LONG_EMPORIUM

名称	ID
天安人寿	天安生活
TID LAMOND	TID_LAMOND
TIEN21	TIEN21
TIFFI	TIFFI
老虎	老虎
TIKI	TIKI
TILEAFRIC	TILEAFRIC
TIMANTTISET	TIMANTTISET
超时	TIME_OUT
时空旅行	时光旅行
时间趋势	时间_趋势
锡罐药店	TIN_TIN_DRUGSTORE
蒂诺·冈萨雷斯	TINO_GONZALEZ
TIPSA	TIPSA
地拉那银行	TIRANA_BANK
TISAK	TISAK
太钢银行	Tisco_bank
TITONI	TITONI
TIZA	TIZA
TJ HUGHES	TJ_HUGHES

名称	ID
TJX	TJX
TK Maxx	tk_maxX
TKK	TKK
TKNG	TKNG
TLCU 金融	TLCU_Financ
TLG 鸡肉	TLG_CHICKEN
TM	TM
TM Lewin	tm_lewin
TM 超市	TM_Sump
TMFB	TMFB
TMH 联邦信用合作社	TMH_Federal_Credit_Union
TMall 维护	TMall_维护
田纳西烧烤	tn_Grill
TN Mart	TN_mart
TNG	TNG
TNT	TNT
TNT Courier	TN_Courier
吐司盒	TOAST_BOX
烟草压榨机	TOBACCO_PRESS
东武公园	TOBU_PARK

名称	ID
TOD'S	TOD'S
今日旅馆	Today_inns
TODO MODA	TODO_MODA
TODOMODA	TODOMODA
多哥	多哥
多哥的	多哥的
TOK	TOK
TOKMANNI	TOKMANNI
TOM & CO	TOM_&_CO
TOM N TOMS	TOM_N_TOMS
汤姆裁缝	TOM_TAILER
TOMBIS	TOMBIS
汤米	汤米
TOMMY HILFIGER	TOMMY_HILFIGER
TOMONY	TOMONY
TONI&GUY	TONI&GUY
补品	补品
TONY ROMA 的	TONY_ROMA'S
TOP CLEAN ruhatisztító	top_clean_ruhatisztító
上头发	上头发

名称	ID
最高机密	最高机密
顶级服务团队	顶级服务团队
最热门的时间	TOP_TIME
顶级运动	顶级运动
黄玉杂货店	topaz_Grocely_store
顶级汽车	顶级汽车
顶级螃蟹	顶级螃蟹
TOPPRES	TOPPRES
上衣	上衣
TOPS 友好市场	热门_友好_市场
TOPs	TOPs
TORFS	TORFS
TORGA OPTICAL	TORGA_OTICAL
HIPOCAMPO LOCAS LOCAS	TORTAS_LOCAS_HIPOCAMPO
TORY BURCH	TORY_BURCH
小孩	小孩
TOTAL 洗车	Total_car _wash
TOTAL PARCO	TOTAL_PARCO
TOTALERG	TOTALERG
TOTALGAZ	TOTALGAZ

名称	ID
TOTO	TOTO
触摸	触摸
强硬的牛仔史密斯	Tough_JeanSmit
旅游线快车	TOURLINE_EX
TOUS	TOUS
TOUS les JOURS	TOUS_les_Jours
玩具星球	玩具星球
TOY.RU	TOY.RU
丰田	丰田
丰田 Rent-A-Car	丰田_租车
丰田当然	丰田_当然
TP 汽油市场	TP_petrol_Market
TPBank	TPBank
TPC	TPC
TPI	TPI
TPL	TPL
TPTEA	TPTEA
Trac-Astra 租车	trac-astra_Rent_a_Car
贸易商们	贸易商们
TRAMASmas	TRAMASmas

名称	ID
TRANSERVI	TRANSERVI
TRASSA	TRASSA
旅行是啊	TRAVEL_YA
旅行者	旅行者
TRAVESURAS	TRAVESURAS
TREF Sport	Tref_sport
TREYS	TREYS
TRGOSTIL	TRGOSTIL
TRADGONE KRK	TRGOVINA_KRK
TRH 酒店	TRH_Hoteles
TRIADA	TRIADA
TRIBO	TRIBO
TRIKA	TRIKA
TRIKO MISIRLI	TRIKO_MISIRLI
旅行旅馆	TRIP_INN
胜利	胜利
巨魔	巨魔
TRONY	TRONY
TRU TRUSSARDI	TRU_TRUSSARDI
卡车	卡车

名称	ID
TRUEFITT & Hill	TrueFitt_&_Hill
TRUFFAUT	TRUFFAUT
信任 FCU	TRUST_FCU
TRYBA	TRYBA
宝藏	宝藏
TS 俱乐部	TS_Club
TSB	TSB
TSB 银行	TSB_Bank
TSINGTUI TE PHANG	TSHING_TUI_TE_PHANG
TSIFROGRAD	TSIFROGRAD
TSL	TSL
TSS 摄影	TSS_摄影
TSURUHA	TSURUHA
TSUTAYA	TSUTAYA
TSUTAYA 书店	TSUTAYA_BOOKSTORE
TT bagatt	tt_Bagatt
TT&T	TT&T
TTB	TTB
TTCU 联邦信用合作社	TTCU_Federal_Credit_Union
TTK 快车	ttk_Expres

名称	ID
TTL	TTL
TTNET	TTNET
TUACAR	TUACAR
TUBAO	TUBAO
都铎王朝	都铎王朝
TUI	TUI
TUI ReiseCenter	TUI_ ReiseCenter
TUI Viagens	TUI_Viagens
TUMELERO	TUMELERO
董芳美	董芳美
TUZZI	TUZZI
电视销售和房屋	TV_sales_&_Home
TVOE	TVOE
TVR	TVR
电视	电视
TWG Tea	TWG_tea
曲折的	曲折的
TWILFIT	TWILFIT
扭曲	扭曲
旋风牛仔裤	TWISTER_牛仔裤

名称	ID
两个男人和一辆卡车	TWO_MEN_AND_A_TRUCK
TXB	TXB
TYREPLUS	TYREPLUS
大众银行	大众银行
大湾	Ta_Wan
Tabak Trafik	Tabak_Trafik
Tabbouli	Tabbouli
圣玛丽亚酒馆	Taberna_Santamaría
塔比	塔比
Tabio	Tabio
桌子桌子	Table_Table
法兰西餐桌和旅馆	Tables_&_auberges_de_France
Tabletka	Tabletka
Tabung Haji	Tabung_Haji
Tacham	Tacham
Tachiya	Tachiya
塔西油	Taci_Oil
TackLE	Tackle_Ber
塔可酒吧	Taco_Bar
Taco Bell	Taco_Bell

名称	ID
Taco Bill	Taco_Bill
Taco Bueno	Taco_Bueno
Taco Cabana	Taco_Cabana
Taco H	Taco_H
塔可旅馆	Taco_Inn
Taco John's	Taco_John's
Taco Mac	Taco_Mac
Taco Maker	Taco_Maker
Taco Mayo	Taco_Mayo
Taco Palenque	Taco_Palenque
Taco Tico	Taco_Tico
Taco del Mar	taco_del_mar
TacoTime	TacoTime
炸玉米饼	炸玉米饼
塔迪姆披萨	Tadim_Pizza
Tadychs Econofoods	Tadychs_Econofoods
Tafe	Tafe
Tag	Tag
泰格豪雅	Tag_Heuer
TagDisruptor	TagDisruptor

名称	ID
塔霍广场	塔霍广场
塔塔卡莱景点	Thakale_Spot
Tahweel Al Rajhi	Tahweel_Al_Rajhi
太昌饼家	太清_面包店
大钟	Tai_Chung
Tai Loy	Tai_Loy
泰·欧西	Tai_Oursea
Tai Phat Gioi May Tap	Tai_Phat_The_Gioi_May_Tap
Tai Shih Fu	Tai_Shih_Fu
泰安银行	泰安银行
泰安保险	泰安_保险
太仓农村商业银行	太仓_农村_商业银行
台青地产	台青地产
台中银行	台中_银行
泰康人寿	泰康生活
Tailg	Tailg
台北富邦银行	台北_富邦银行
台北之星银行	台北_星银行
Tairyo	Tairyo
Tairyoudonya	Tairyoudonya

名称	ID
泰山农村商业银行	台山_农村_商业银行
台新银行	台新银行
Taishu Shokudo Yasubee	Taishu_Shokudo_Yasubee
太书溪	太书溪
台湾 Big Egg	台湾_Big_Egg
台湾商业银行	台湾_商业银行
台湾合作银行	台湾_合作银行
台湾美味	台湾_好吃
台湾火灾和海上保险	台湾_火灾_海上_保险
台湾生鲜超市	台湾_生鲜超市
台湾本垒打	Taiwan_Home_Run
台湾人寿	台湾_生活
台湾移动	台湾_手机
台湾停车	台湾_停车
台湾糖业加油站	台湾_糖业加油站
台湾苏格股份有限公司	台湾_Suger_股份有限公司
台湾 Youni 加油站	台湾_Youni_加油站
泰兴	泰兴
泰兴农村商业银行	泰兴农村商业银行
Taiyo Sekiyu	Taiyo_Sekiyu

名称	ID
太原农村商业银行	太原_农村_商业银行
台州农村商业银行	泰州_农村_商业银行
泰姬陵	泰姬陵
但马信用金库	但马信用金库
Taka-Q	Taka-Q
高仓町咖啡	高仓町咖啡
高松信用金库	高松信用金库
高锅信用金库	Takanabe_Shinkin_Bank
高冈信用金库	高冈信用金库
Takara 标准	takara_standard_
宝岛	宝岛
takarekPont	takarekPont
高崎信用金库	高崎信用金库
高岛屋	高岛屋
高岛	高岛
换5次机油	Take_5_oil_Change
记笔记	记笔记
起飞	起飞
Takeroom	Takeroom
泷野川信用金库	泷之川_信用金库

名称	ID
Takko 时尚	Takko_时尚
Tako-Away	Tako-Away
章鱼烧屋	章鱼烧屋
Taku	Taku
Talbots	Talbots
天赋树	天赋树
Talisman	Talisman
会说法语	会说话_法语
塔拉哈西-莱昂 FCU	TallaHassee-leon_FCU
Talleres Verdes	Talleres_Verdes
塔尔默银行和信托基金	Talmer_bank_and_Trust
塔鲁克总部医院	Taluk_Head_Head_Quart
Talwalkars	Talwalkars
TamJai 云南密县	TamJai_云南_密县
TamJia SamGor	TamJia_SamGor
多摩信用金库	Tama_Shinkin_Bank
Tamaris	Tamaris
玉岛信用金库	玉岛信用金库
Tamayaki	Tamayaki
坦博	坦博

名称	ID
Tambovfarmatsiya	Tambovfarmatsiya
泰米尔纳德商业银行	泰米尔纳德邦_商业银行
塔米米市场	Tamimi_Markets
Tamoya	Tamoya
坦帕湾 FCU	tampa_bay_fcu
Tan-Yo 信用金库	Tan-Yo_Shinkin_Bank
Tandem	Tandem
坦迪	坦迪
坦迪电子	Tandy_电子
坦迪皮革工厂	坦迪_皮革工厂
塔内拉	塔内拉
唐宫	唐宫
丹吉尔	丹吉尔
丹吉尔奥特莱斯	丹吉尔奥特莱斯
Tangerina	Tangerina
柑橘	柑橘
唐九超市	唐九超市
唐拉雅秀酒店及度假村	Tangla_酒店_度假村
唐年	唐年
探戈舞	探戈舞

名称	ID
唐山市商业银行	唐山城市商业银行
唐山百货大楼	唐山_百货大楼
唐山农商银行	唐山_农村_商业银行
塔尼亚	塔尼亚
Tanishq	Tanishq
Tank & Rast	坦克_&_Rast
坦卡	坦卡
Tankvart	Tankvart
檀露烧烤	天路_烧烤
Tansas	Tansas
坦西安	坦西安
坦塔	坦塔
诱惑者	诱惑者
Tantammen Hompo	Tantammen_Hompo
Tantitoni	Tantitoni
Tanuki	Tanuki
Tanyaxue	Tanyaxue
Tanyu	Tanyu
坦桑尼亚邮政银行	坦桑尼亚_邮政银行
陶香	陶香

名称	ID
Taode Shaguo	Taode_Shaguo
陶翔	陶翔
桃园观塘宝	桃园_冠堂宝
Tapa King	Tapa_King
Tape a l'oeil	tape_à_l'oeil
Tapelia	Tapelia
希尔顿挂毯系列	Tapestry_Collection_by_hilton [酒店] _18
希尔顿挂毯系列	Tapestry_Collection_by_hilton [酒店] _81
Taps 啤酒厂和餐厅	Taps_Brewery_&_Restauran
Taqa	Taqa
Taqueria Arandas	Taqueria_Arandas
Taras Bul'Ba	Taras_Bul'Ba
Tarbell	房地产经纪人
目标	目标
Taro Boba	Taro_Boba
Tarocash	Tarocash
Tarragona	Tarragona
塔兰特县 CU	Tarrant_countys_cu
塔丁和巧克力	tartine_et_Chocolat
田崎	田崎

名称	ID
塔斯科姆银行	塔斯科姆银行
Taset	Taset
塔希尔披萨	塔希尔_披萨
Tastee-Freez	Tastee-Freez
塔塔咨询服务	Tata_咨询_服务
Tatbak	Tatbak
塔特利	塔特利
Tatfondbank	Tatfondbank
塔蒂	塔蒂
Tatlibak	Tatlibak
Tatnafta	Tatnafta
Tatneft	Tatneft
Tatnefteprodu	Tatnefteprodu
塔特拉银行	Tatra_Banka
Tatsotsbank	Tatsotsbank
Tattekhmedfarm	Tattekhmedfarm
Tatts	Tatts
Tatuang	Tatuang
汤顿联邦信用合作社	Taunton_Federal_Credit_Union
Taunus Sparkasse	Taunus_sparkasse

名称	ID
塔维利	塔维利
Tavricheskiy	Tavricheskiy
Tavrovskie Myasnye Lavki	Tavrovskie_myasnye_Lavki
Tavuk dünyasi	tavuk_dünyasi
泰之家	Tay_House
泰勒租赁	Taylor_Rental
Taymer	Taymer
Taziki 的地中海咖啡厅	Taziki's 地中海咖啡厅
Taq	Taq
Tchap-Tchap Petro	Tchap-Tchap_Petro
Tchibo	Tchibo
Tchip Coiffure	Tchip_Coiffure
Tché Farmácias	Tchê_Farmácias
道明汽车	Td_Cars
Tdc	Tdc
茶和魔法之手	Tea_&_Magic_Hand
茶钱	茶_Bucks
Tea Daye	Tea_Daye
茶帖	Tea_Post
茶风暴	Tea_Storm

名称	ID
Tea's	Tea's
Tea-juan	Tea-juan
Tea12	Tea12
TeaWood	TeaWood
Tian Bouilli 老师	Teacher_Tian_Bouilli
教师信用合作社	教师_Credit_Union
教师 FCU	teachers_FCU
教师互助银行	Teachers_Mutual_Bank
Tealive	Tealive
TeamLogic IT	TeamLogic_IT
Teaplus	Teaplus
Teavana	Teavana
Teba Bank	Teba_Bank
德邦证券	泰邦证券
Tech CU	tech_cu
Tech Mahindra	Tech_Mahindra
Tech-Net	Tech-Net
科技商业银行	科技商业银行
科技世界	科技世界
Technomarket	Technomarket

名称	ID
科技园	科技园
科技城	科技城
Teckon Ciel	Teckon_Ciel
Tecno 清洗系统	泰克诺清洗系统
Tecnocasa	Tecnocasa
东元电气	东元电气
东元市场	Teco_Market
Tectra	Tectra
泰德·博丁	Ted_Bodin
Ted 的相机	Ted's 的相机
泰德的蒙大拿州烧烤	Ted's_montana_Grill
Tedcastles	Tedcastles
Tedeschi Food Shop	Tedeschi_Food_Shop
Tedu	Tedu
TeeGschwendner	TeeGschwendner
Tefal	Tefal
Tegler	Tegler
Tegut	Tegut
塔里克广场	Teh_Tarik_Place
Teh-Tai	Teh-Tai

名称	ID
Technomanija	Technomanija
Teishokuya Hyakusai Shun	Teishokuya_Hyakusai_Shun
TekhnoLux	TekhnoLux
TekhnoMir	TekhnoMir
Tekhnobank	Tekhnobank
泰赫诺波利斯	泰赫诺波利斯
Tekhnoshok	Tekhnoshok
Tekhnosila	Tekhnosila
Tekin Acar	Tekin_Acar
铁基镇	Tekkie_Town
Teknosa	Teknosa
Tekpa	Tekpa
纺织银行	纺织银行
Telangana Grameena 银行	Telangana_Grameena_Bank
电信社区 CU	telco_community_cu
Telco Plus	telco_plus_cu
告诉	告诉
TeleChoice	TeleChoice
TeleShop	TeleShop
可传输	可传输

名称	ID
Telecel	Telecel
数字电信	电信_数字
Telecor	Telecor
电话.Ru	电话.Ru
Telefoonkopen.Nl	Telefoonkopen.Nl
Telehuset	Telehuset
电信	电信
德国电信	德国电信_德国
斯洛文尼亚电信	斯洛文尼亚电信
Telemach	Telemach
Telemaks	Telemaks
Telenor	Telenor
Telenorbutikken	Telenorbutikken
Telesindo 商店	Telesindo_Shop
Teletorium	Teletorium
Telewiz	Telewiz
泰尔福	泰尔福
Telhanorte	Telhanorte
Telhio CU	telhio_cu
Telia	Telia

名称	ID
TeliaSonera	TeliaSonera
Telis Kikeris	Telis_kikeris
电信直播	电信_Direct
澳洲电信	澳洲电信
Telus	Telus
Temeiyigou 超市	Temeiyigou_超市
Tempero Manero	Tempero_Manero
暴风雨	暴风雨
Tempo	Tempo
诱惑	诱惑
天妇罗新宿纲八	天妇罗_新宿_Tsunahachi
Ten	Ten
Ten Ren	Ten_Ren
十十	Ten_Ten
Ten-Ichi	Ten-Ichi
TenFu's TEA	TenFu's_tea
TenRensTeaHongKong	TenRensTeaHongKong
Tenaga Nasional Berha	Tenaga_Nasional_Berhad
Tenda Atacado	Tenda_Atacado
Tenda R	Tenda_R

名称	ID
潮流	潮流
Tendido 18	Tendido_18
天妇罗天妇罗本铺参天	Tendon_tempura_Honpo_Santen
Tendto CU	Tendto_cu
Tenet 医疗保健	宗旨_医疗保健
腾茶	腾茶
Tenho	Tenho
天神屋	天神屋
天纳克	天纳克
田纳西河谷管理局	田纳西河谷管理局
网球一号	Tennis_One
Tentazioni	Tentazioni
Tepe Home	Tepe_Home
Tepoznieves	Tepoznieves
Teremok	Teremok
Terex Vectra	Terex_Vectra
照烧男孩	Teriyaki_Boy
照烧体验	照烧_体验
Terkenlis	Terkenlis
终端	终端

名称	ID
Terminix	Terminix
Termo	Termo
Terno	Terno
Terpel	Terpel
泰拉银行	Terra_Bank
泰拉工业公司	terra_Industries
可怕的 Herbst	Terrible_Herbst
特里·怀特化学家	特里_怀特_化学家
TerryWhite Chemmart	TerryWhite_Chemmart
Teruteru	Teruteru
Tervolina	Tervolina
乐购	乐购
乐购快递	乐购快递
乐购店内药房	乐购_店内_药房
爱尔兰乐购	乐购_爱尔兰
乐购地铁	Tesco_Metro
乐购加油站	乐购加油站
乐购超市	Tesco_Superstore
tescoma	tescoma
Tesla	Tesla

名称	ID
特斯拉电动汽车	特斯拉_EV
Tesoro	Tesoro
Teva Naot	Teva_Naot
德克萨斯湾 CU	Texas_bay_cu
德克萨斯州汽车所有权和发薪日贷款	公司
德州农业局保险	得克萨斯州_农场_局_保险
德州健康资源	德克萨斯州_健康_资源
德州土地和牛排馆	Texas_Land_&_Cattle_Steak_House
德州土地和牛排馆	Texas_Land_&_Cattle_Steakhouse
德州肋骨	Texas_Ribs
德州路屋	Texas_roadhouse
德克萨斯州立银行	德克萨斯州立银行
德克萨斯州低成本保险	德克萨斯州_低成本_保险
德州牛排馆和沙龙	Texas_Steakhouse_&_Saloon
巴西德克萨斯州	德克萨斯州_de_Brazil
Texim Bank	Texim_Bank
纺织品市场	纺织品市场
纹理	纹理
纹理宝贝	Testura_Baby
TexxEnergy	TexxEnergy

名称	ID
Tezenis	Tezenis
标签	标签
泰国航空	泰国航空
泰国信贷零售银行	泰国_信贷_零售_银行
泰国人寿保险	泰国人寿保险
ThaiNamthip	ThaiNamthip
Thaizone	Thaizone
Thalexim	Thalexim
Thalia	Thalia
泰晤士河	泰晤士河
Thanachart 银行	Thanachart_Bank
塔那区中央合作银行	Thane_district_central_Co-op
Thane Janata Sahakari 银行	Thane_Janata_Sahakari_Bank
谢谢酒店 Plus	谢谢 Hotel_Plus
Thann	Thann
坦森	坦森
第二乐章	第二动作
AFL Store	the_afl_Store
地址 Hotels + 度假村	地址_酒店_+_度假村
Ajara Urban Bank	Ajara_Urban_Bank

名称	ID
胡同	The_Alley
另类董事会	_另类董事会
动物之王	动物之王
艺术机构	艺术机构
Arvind Store	The_Arvind_Store
运动员的脚	运动员的脚
婴儿工厂	宝贝工厂
贝克小屋	面包师的小屋
The Baker's	the_Baker's_doze
该银行	The_Bank
东亚银行	东亚银行
喀山银行	喀山银行
韩国银行	韩国银行
密苏里银行	密苏里银行
酒吧法	_Bar_Method
Bar 证券交易所	The_Bar_Stock_Exchan
理发公司	理发师公司
巴雷守则	_Barre_Code
美容盒	The_Beauty_Box
美容角落	美容角落

名称	ID
床品商店	The_Bed_Shop
Bed Store	The_Bed_Store
啤酒咖啡厅	The_Beer_Café
啤酒公司	The_Beer_Company
啤酒商店	The_Beer_Store
北京游行	北京大游行
Big Biscuit	The_Big_Biscuit
大快车	The_Big_Expr
蓝军	The_Blues
植物学家	The_Botanist
The Bottle-O	The_Bottle-O
Brahouse	The_Brahous
面包篮	The_bread_Bask
的 BrickKicker	The_ BrickKicker
汉堡的牧师	The_Burger's _Priest
Butcher Shop & Grill	The_Butcher_Shop_&_Grill
CRAZY Store	The_crazy_Store
首都	The Capital
Capital Grile	Capital_Grille
凯雷集团	凯雷集团

名称	ID
芝士蛋糕工厂	芝士蛋糕工厂
芝士蛋糕店	The_Cheesecake_Shop
钦奈手机	The_Chennai_Mobiles
鸡肉饭店	The_Chicken_Rice_Shop
儿童乐园奥特莱斯	儿童乐园奥特莱斯
巧克力房	The_Chocolate_Roo
清洁局	清洁管理局
克利夫兰诊所	克利夫兰诊所
合作银行	the_coperative_Bank
咖啡豆和茶叶	The_Coffee_Bean_&_Tea_Leaf
咖啡屋	The_Coffee_
咖啡店	The_Coffee_
康拉多的咖啡屋	Conrado's Coffee_House
集装箱商店	_Container_Store
Cosmos 合作银行	the_cosmos_co-op_Bank
乡村商店	乡村商店
County FCU	the_county_fcu
创意小组	_Creative_Group
纸杯蛋糕面包店	纸杯蛋糕_面包店
德干商人合作银行	the_deccan_Merchants_Co-op_Bank

名称	ID
牙医的选择	牙医的选择
餐厅	The_dining_Room
探索商店	Discovery_Store
The EGG & I	the_egg_&_i
教育工作者	教育工作者
Egg & I 餐厅	The_Egg_&_I_Restaurants
皇帝	_Emperor
《艺人》	_Entertainer
企业家的来源	企业家的来源
行政人员	_Executive
Face Shop	The_Face_Shop
五十年代	五十年代
最好的 FCU	the_finest_FCU
第一银行	The_First_Bank
法国面包师	The_French_Baker
法式面包	The_French_Loaf
Fresh Grocer	The_Fresh_Grocer
生鲜市场	新鲜市场
自由之丘花园	花园_自由之丘
盖特威酒店	The_Gateway_Hotel

名称	ID
仿制药房	仿制药_药房
戈达德学校	The_Goddard_School
好家伙	The Good_Guys
伟大的美国百吉饼	The_Great_American_Bagel
成长教练	成长教练
Guns & Roses Pub-Grille	The_Guns_&_Roses_Pub-Grille
Gutter Guys	The_Gutter_Guys
Habit 汉堡烧烤	The_Habit_Burger_Grill
Halal Guys	The_Halal_Guys
哈特福德	_Hartford
家得宝	家得宝
Hosteller	The_Hosteller
酒店系列	酒店系列
五号酒店	The_Hotel_V
豪斯咖啡厅	House_Cafe
高尔夫之家	高尔夫之家
Hub	The_Hub
亨廷顿国家银行	亨廷顿国民银行
帝国酒店	帝国酒店
爱尔兰之家	The_Irish_House

名称	ID
意大利咖啡公司	意大利咖啡公司
联合脊骨疗法	关节脊骨疗法
茶之王站	茶之王站
Kanchipuram 中央合作银行	the_Kanchipuram_Central_Co-op_Bank
Kangra 合作银行	the_kangra_co-operative_Bank
卡拉德城市合作银行	The_Karad_urban_Urban_Co-operativ
Kase	The_Kase
Keg Steakhouse & Bar	The_Keg_Steakhouse_&_Bar
Kickin'Crab	The_Kickin'_Crab
Kooples	The_Kooples
The Lalit	The_Lalit
点头之地	点头之地
洗衣篮	_Laundry_Basket
领先酒店	领先酒店
学习经历	学习体验
照明仓库	照明仓库
链接	The_Link
小健身房	The_Little_Gym
她需要的小东西	她需要的_Little_Things_Things
豪华精选	_Luxury_系列

名称	ID
豪华精选（阿拉贝拉）	_Luxury_Collection_（Arabella）
疯狂屠夫	The_Mad_Butcher
《女佣》	The_Maids
购物中心生活商店	购物中心_Lifestore
曼哈顿鱼市场	曼哈顿鱼市场
熔炉	The Melting_Pot
磨坊	The_Mill
磨坊	The_Mill
奈尼塔尔银行	The_Nainital_Bank
国家银行	国民银行
国家合作银行	National_Co-operative_Bank
大自然保护协会	自然保护协会
新店	The_New_Shop
北面	The_North_Fac
诺丁汉	_诺丁汉
俄亥俄州教育大学	the_Ohio_Educatial_cu
老房子	The Old_House
Old Spaghetti House	The_Old_Spaghetti_House
光学超市	The_Optical_Superstore
原版 Buscemis	The_Original_Buscemis

名称	ID
原始的意大利馅饼	The_Original_Italian_Pie
披萨公司	披萨公司
PRO SHOP	the_pro_shop
煎饼店	The_pancake_Parlour
储藏室	_Pantry
公园	The_Park
宾夕法尼亚交通公司	Penn_Traffic_Company
中国人民银行	中国人民银行
香水连接	香水连接
香水店	_Perfume_Shop
电话画廊	_Phone_Gallery
电话屋	电话屋
泡菜桶	Pickle_Barrel
画面人物	Picture_People
披萨之王	The_Pizza_King
首席财务集团	Principal_Financial_Group
Qube 酒店	The_Qube_Hotel
The Raj	The_Raj
山脉	_Range
Read Shop	The_Read_Shop

名称	ID
真实的中国	The_Real_China
拒绝商店	The_Reject_Shop
储备网络	_Reserves_Network
合适的那个	The Right_One
皇家索尼斯塔	皇家索尼斯塔
滑板车商店	the_scooter_store
沙特投资银行	沙特投资银行
塞贝尔	The_Sebel
鞋类清仓店	The_Shoe_Clearance_Stor
Shoko Chukin Bank	The_Shoko_Chukin_Bank
奇异厨房	The_Singular_Kitchen
Select Comfort 推出的睡眠号码商店	sleep_number_store_by_select_Comfor
Small Tadpole 高端钢琴	The_Small_Tadpole_Hidge_End
汤勺	The_Soup_Spoon
来源	_Source
Sting	The_Sting
该商店	The_Store
苏丹中心 TSC	the_Sultan_center_TSC
Summit FCU	the_Summit_FCU
阳光屋	The_Sunshine_House

名称	ID
Tea Corner	Tea_Corner
田纳西州大学	田纳西州_cu
Thickshake	Thechshake_Factory
Tosti Club	The_Tosti_Club
UPS 商店	the_UPS_store
联合银行	联合银行
联合保险公司	美国联合保险公司
Vaish 合作社 Adarsh Bank	The_Vaish_Co-operative_Adarsh_Bank
维生素专柜	The_Vitamin_Shoppe
行走公司	Walking_Company
仓库	_仓库
Washpoint	洗手点
西布朗	The_West_Brom
白人公司	白人公司
温莎之结	The_Windsor_Knot
炒锅	The_Wok
作品	_Works
黄辣椒	The_Yellow_Chilli
TheNewMotion	TheNewMotion
ThePluginCompany	ThePluginCompany

名称	ID
Theeb	Theeb
thegioididong	thegioididong
蒂哈斯	蒂哈斯
Thelins	Thelins
主题	主题
Theobroma	Theobroma
理论	理论
Theos 食物	Theos_Food
治疗助理	治疗员工
赛默飞世尔科技	Thermo_Fisher_
热气体	热气体
Thiele	Thiele
Thierry 21	Thierry_21
记住的事情	记住的事情
Thinku 教育	Thinku_教育
第三次联邦储蓄和贷款	Third_Federal_Savings_&_Loan
Thiros	Thiros
Thirsty Camel	Thirsty_Camel
口渴的酒	口渴_酒
Thiruvananthapuram Dist 合作银行	Thiruvananthapuram_dist_co-op_Bank

名称	ID
这种时尚	这_Fashion
本周	本周
蓟	蓟
Thoben	Thoben
Thomas Cook 旅行社	Thomas_Cook_Reiseburo
托马斯·库克旅游商店	Thomas_Cook_Travel_Shop
Thomas Dux 杂货店	Thomas_Dux_Grocer
托马斯·菲利普斯	Thomas_Philipps
托马斯维尔	托马斯维尔
丁字匠	丁字匠
桑顿石油	Thornton_Oil
Thorntons	Thorntons
三	三
三个贵州人	Three_guizhouren
加拿大 Thriftlodge	ThriftLodge_加拿大
Thriftway	Thriftway
节俭	节俭
节俭食品	Thrifty_Foods
节俭旅馆	Thrifty_Inn
节俭租车	Thrifty_Rental_Car

名称	ID
节俭链接	节俭链接
Thrivent 金融银行	繁荣_金融_银行
Thrivent Financial for	路德教会的 Thrivent_Financial_for_F
Throtleman	Throtleman
推力	推力
Thuisin	Thuisin
图恩	图恩
雷云订阅	雷云_订阅
Thurgauer Kantonalbank	Thurgauer_KantonalBank
百里香	百里香
甲状腺保健	甲状腺保健
ThyssenKrupp 电梯	ThyssenKrupp_电梯
Tia	Tia
蒂亚租车	Tia_Car_Rent
Tia's Tex-Mex	Tia's_tex-Mex
田可龙	田可龙
田田玉刚	田田玉刚
TianLv 寿司	TianLv_寿司
天道教育	天道_教育
天地人和连锁酒店	天地人河_连锁酒店

名称	ID
天鹅莲	天鹅莲
天方茶	天芳_茶
天河	天河
天惠超市	天汇_超市
天津百娇园	天津_百角_园
天津滨海农村商业银行	天津_滨海_农村通信_银行
天津津南村镇银行	天津_津南_村镇银行
天津农村商业银行	天津_农村_商业银行
天猫小店	天猫_小店
天目	天目
天天	天天
天王	天王
天薇	天薇
Tiany	Tiany
天一小馆餐厅	天一小馆_餐厅
天元	天元
天元证券	天元证券
头饰头饰	Tiara_Tiara
Tibb 的 Frankie	Tibb's_Frankie
票证	票证

名称	ID
Ticketek	Ticketek
潮汐干洗店	Tide_Dry_Cleaners
Tie Ge Bi	Tie_Ge_Bi
TieRack	TieRack
铁木真	铁木真
店铺 IMSS	tienda_imss
TiendaCulturista.com	TiendaCulturista.com
Tiendanimal	Tiendanimal
Tiendas 3B	Tiendas_3B
Tiendas Lores	Tiendas_Lores
Tiendas Mass	Tiendas_Mass
Tiendas Super Joel	Tiendas_Super_Joel
Tiendas TX2	天达斯_ TX2
Tienyow 酒店	Tienyow_Hotels
TierOne 银行	TierOne_银行
铁通电信	铁通_电信
Tiff 的零食	Tiff's_Treats
蒂芙尼	蒂芙尼
Tiffany & Co	Tiffany_&_Co
Tiffosi	Tiffosi

名称	ID
Tiffosi Kids	Tiffosi_Kids
Tifon	Tifon
老虎家族	Tiger_Family
老虎玛特	Tiger_Mart
老虎轮和轮胎	Tiger_Wheel_&_Tyre
老虎炒锅	Tiger_Wok
瑞典之虎	瑞典之虎
老虎的牛奶	Tiger's_Milk
TigerAttitude 特许糕点银行	TigerAttitude_Chartered_Pastry_Bank
TigerDirect	TigerDirect
TigerMart	TigerMart
TigerSugar	TigerSugar
Tigros	Tigros
蒂华纳平底鞋	Tijuana_Flats
Tike	Tike
Tiki-Ming	Tiki-Ming
Til Henne	Til_Henne
瓷砖巨人	瓷砖_巨人
瓷砖仓库	瓷砖_仓库
Tilted Kilt Pub & Eatery	Tilted_Kilt_Pub_&_Eatery

名称	ID
Tim	Tim
Tim Ho Wan	Tim_Ho_Wan
蒂姆·霍顿斯	Tim_Hortons
Timber Lodge 牛排馆	Timber_Lodge_Steakhouse
Timbercity	Timbercity
林地	林地
天伯兰银行	Timberland_Bank
天伯兰工厂店	Timberland_工厂_商店
Timberline FCU	timberline_FCU
Time	Time
津巴布韦时间银行	津巴布韦时光银行
超时运动	Time_Out_Sport
时代华纳有线电视	Time_Warner_Cable
计时器库	Timer_Bank
时间方面	时间方面
天美时	天美时
时区	时区
Timhotel	Timhotel
Timothy's	Timothy's
Timpson	Timpson

名称	ID
Tims & Parker	Tims_&_Parker
Tin'koff 银行	Tin'koff_Bank
TinQ	TinQ
TinderFit	TinderFit
Ting	Ting
婷婷	Ting_Ting
听道教育	听道_教育
Tinker FCU	tinker_fcu
Tinospizza	Tinospizza
Tint World	色彩世界
Tintorerías Electrolux ultima gen	tintorerías_Electrolux_Ultima_Gen。
丁托列托	丁托列托
小贴士	Tip_Top
Tip Top Top	Tip_Top_Tailors
Tip y Tap	tip_y_tap
小贴士和脚趾美甲沙龙	Tips_&_Toes_Toes_Nail_Salon
轮胎仓库	轮胎仓库_仓库
轮胎王国	轮胎王国
轮胎专业人士	Tire_Pros
Tire Rama	Tire_Rama

名称	ID
Tirecraft	Tirecraft
轮胎升级版	Tires_Plus
泰雷克斯	泰雷克斯
天梭	天梭
Tiszántuli Takarék	Tiszántuli_Takarék
泰坦	泰坦
泰坦之眼+	Titan_Eye+
标题现金	Title_Cash
TitleBucks	TitleBucks
Titlemax	Titlemax
Tittiot	Tittiot
蒂沃利	蒂沃利
Tizza	Tizza
Tk Kredyt	Tk_Kredyt
Tkk 炸鸡	Tkk_fried_Chicken
Tnsc 银行	Tnsc_Bank
致霍里亚蒂科	To_Horiatiko
Toarmina 披萨	Toarmina_Pizz
吐司和森林	吐司和森林
烟草连接	烟草连接

名称	ID
烟草奥特莱斯加杂货	Tobacco_Outlet_Plus_Grocery
烟草路奥特莱斯	Tobacco_Road_Outlet
烟草 S 压榨机	tobacco_s_press
烟草店	Tobacco_Shop
烟草站	烟草站
托比	托比
Toby Carvery	Toby_Carvery
Toco Toco	Toco_Toco
今日门店	Today_Stores
今天的美味商店	今天的_Delicious_stores
今天的教育	今天的教育
今日特辑	Today's_Extra
Todeschini	Todeschini
托迪斯	托迪斯
TodoDia	TodoDia
Togas	Togas
Togg	Togg
Togliattikhimbank	Togliattikhimbank
Tohapi	Tohapi
Toi & Moi	Toi_&_Moi

名称	ID
Tok&Stok	Tok&Stok
Tokai-Rokin	Tokai-Rokin
Toki Tok	Toki_Tok
Tokic	Tokic
Tokio Marine Newa 保险	Tokio_Marine_Newa_Insurance
Tokiya	Tokiya
东京 Gunung Agung	Toko_Gunung_Agung
Toks	Toks
德田银行	德田银行
Tokujuan	Tokujuan
Tokuyo	Tokuyo
东京 Aburagumi 总本店	Tokyo_Aburagumi_Sohonten
东京湾信用金库	Tokyo_Bay_Shinkin_Bank
东京城市信用金库	Tokyo_cityo_Shinkin_Bank
东京熟食店	Tokyo_Deli
东京快车	东京快车
东京东信用金库	东京_东信用金库
东京内饰 Kagu	Tokyo_Interior_Kagu
东京 Joe's	Tokyo_Joe's
东京梅甘娜	Tokyo_Megane

名称	ID
东京信用金库	Tokyo_Shinkin_Bank
东京 Sundubu	Tokyo_Sundubu
东京轮胎	东京_轮胎
东京东京	东京_东京
东京市	东京市
东急Hands	Tokyu Hands
东急体育绿洲	东京_体育_Oasis
托莱多	托莱多
Toli	Toli
托尔兄弟	Toll_Brother
汤姆·福特	Tom_Ford
Tom N Toms 咖啡	Tom_N_Toms_Coffee
汤姆·图姆	Tom_Thumb
Tom's Burgers	Tom's_Burgers
Tomakomai 信用金库	Tomakomai_Shinkin_Bank
番茄和洋葱	番茄_&_洋葱
番茄早午餐和咖啡厅	番茄_早午餐_&_Café
番茄厨房咖啡厅	番茄_厨房_咖啡厅
Tometeba	Tometeba
汤姆林森房地产	汤姆林森_房地产

名称	ID
汤米·巴哈马	Tommy_Bahama
Tommy Mel's	Tommy_Mel's
Tomod's	Tomod's
Tomolo	Tomolo
明天让生活更美好	Tomorrow_Advancing_Life
汤普金斯信托公司	汤普金斯信托公司
唐乐	Tong_Lok
Tongass FCU	Tongass_fcu
同程通美	同程通美
同德莱火锅	Tongdelai _Hot_Pot
同方教育	同方_教育
同富怡牙科	同富怡_口腔医疗
同汉春堂	同汉春堂
东港同和村镇银行	同和村_东港银行
通华	通华
铜陵农村商业银行	铜陵农村商业银行
铜陵万江农村商业银行	铜陵_万江_农村通信_银行
同庆楼	同庆楼
桐乡农村商业银行	桐乡_农村_商业银行
同信证券	同信证券

名称	ID
托尼亚·妮可	Tonia_Nicole
托宁	托宁
炸猪排	炸猪排
炸猪排滨胜店	炸猪排_Hamakatsu
炸猪排 Katsuki	炸猪排_Katsuki
远野信用金库	Tono_Shinkin_Bank
Tontaro	Tontaro
TonusClub	TonusClub
托尼	托尼
Tony & Guy	Tony_&_Guy
Tony Bianco	Tony_Bianco
托尼旅馆	Tony_Inn
托尼工作室	托尼工作室
Tony's 轮胎服务	Tony's tyre_service
TooJay的	TooJay的
Toocha	Toocha
拿走了 Lae Dee	Took_Lae_Dee
工具站	工具站
Toom	Toom
Toom Baumarkt	Toom_Baumarkt

名称	ID
Toot'n Totum	toot'n_totum
Tootsies 鞋市	Tootsie_Shoe_Market
页首	页首
十大假日公园	十大_假日公园
Top CD	top_CD
热门相机	顶部摄像机
顶级车库	顶部_Garage
顶级健康医疗	热门_健康_医疗
顶级国际酒店	Top_国际_酒店
顶级果汁	顶级果汁
Top Layn	Top_Layn
头条新闻	Top_Line
Top Lantas	Top_Lantas
热门市场	热门市场
Top N town	Top_N_Town
热门商店	Top_Shop
Top Star 酒店	Top_Star_Hotel
前十名	前十名
超市	超市
顶级油	顶级油

名称	ID
TopAtlAntico	TopAtlAntico
TopLine FCU	TopLine_FCU
TopMark FCU	TopMark_FCU
TopShop 便利店	TopShop_便利店
黄玉	黄玉
Topek pekarstvi	topek_pekarstVI
顶级高尔夫	顶级高尔夫
主题	主题
Topman	Topman
Topne	Topne
Topone	Topone
Topper 的披萨	topper's_Pizz
Toppers 披萨	Toppers_Piz
每日最佳	上衣_Daily
泰国上衣	上衣_泰国
顶级市场	热门市场
超级上衣	上衣_Super
Topsh	Topsh
Topip	Topip
虎河豚亭	虎河豚亭

名称	ID
Torex	Torex
Torgovaya 公司人	torgovaya_kompaniya_man
Torgovaya ploshchad'	torgovaya_ploshchad'
Tori Tetsu	Tori_Tetsu
鸟之乔治	鸟之乔治
Toriaezu Gohei	Toriaezu_Gohei
Torian	Torian
Torigen	Torigen
Torinosuke	Torinosuke
鸟屋惠比寿	鸟屋_惠比寿
Torq	Torq
托伦特制药	托伦特_制药
Torrid	Torrid
Torshov Bilrekvisita	Torshov_Bilrekvisita
托斯卡尼的	托斯卡尼的
东芝	东芝
Tostao'	Tostao'
全面健身	总健身
总市场	总市场
超级商店总数	Total_superst

名称	ID
Total Wine 及更多	Total_Wine_&_More
道达尔-中化集团	道达尔-中化集团
TotalEnergies	TotalEnergies[汽油站]
TotalEnergies	TotalEnergies[ev_charging_station]
TotalEnergies 访问	TotalEnergies_access
Totalfina	Totalfina
体育总数	体育总数
Totomaru	Totomaru
Totoyamichi	Totoyamichi
托塔	托塔
Tottus	Tottus
Touche de Lis	Touche_de_lis
Tourigalo	Tourigalo
旅行玻璃	旅行玻璃杯
Tout compte fait	Tout_compte_Fait
塔式清洁系统	塔式清洗系统
Tower Health 紧急护理	塔_健康_紧急_护理
城镇与乡村 FCU	town__and__country_fcu
城镇与乡村	城镇和国家
城镇 Health	城镇健康

名称	ID
城镇储藏室	Town_Pant
城镇泵	Town Pump
小镇之星商店	Town_Star_Star_St
城镇谈话家具商	Town_Talk_Firhiners
TownePlace 万豪套房	TownePlace_Suites_by_marriott
煤气公司	煤气公司
联排别墅旅馆	Townhome_Inn
托沃	托沃
Towpath CU	towpath_cu
富山信用金库	富山_信用金库
东洋证券	东洋证券
丰桥信用金库	丰桥信用金库
丰川信用金库	丰川_信用金库
丰田信用金库	丰田信用金库
Toys “R” Us	Toys “R” Us
Toys2Play	Toys2Play
玩具世界	玩具世界
玩具店	Toyzz_Shop
Tozlu	Tozlu
Tplancha	Tplancha

名称	ID
茶天香	Tra_Tien_Huong
田径运动	Track_&_Field
田径和步道	Track_&_Trail
Trackon	Trackon
拖拉机供应	拖拉机_供应
贸易援助	贸易援助
贸易中心	贸易中心
商业秘密	商业秘密
交易之家	交易之家
贸易通	贸易通
商标集	商标集合
FCU 商标	商标_FCU
交易者 Joe's	Trader_Joe's
Trader Vics 餐厅	交易员_Vics_Restaurant
Tradex	Tradex
传统	传统
Trafika	Trafika
Trafika 3DVA	trafika_3dva
Tragus	Tragus
TrailNorth FCU	TrailNorth_FCU

名称	ID
拖拉机	拖拉机
蹦床	蹦床
宁静下来	宁静下来
跨可用区	跨可用区
TransKreditBank	TransKreditBank
泛美航空	泛美航空
泛美西方人寿	泛美_西方_生活
Transamerica	Transamerica
快速	快速
Transkapitalbank	Transkapitalbank
跨国银行	跨国银行
TransportScotland	TransportScotland
运输 FCU	交通_FCU
TransportforLondon	TransportforLondon
Trappers 交易	捕兽者_交易
Trast	Trast
Trast-Capital	Trast-Capital
旅行旅馆	Travel_Inn
TravelCenters 美国的	TravelCenters_of_America
旅行者咖啡	旅行者_咖啡

名称	ID
Travex	Travex
旅行者旅馆	Travelers_Inn
与酒店同行	与酒店同行
Travelink Viajes	Travellink_Viajes
旅行住宿	旅行住宿
Travelodge 酒店	Travelodge_酒店
温德姆旅屋酒店	温德姆 Travelodge_by_wyn
旅行网	旅行网
《旅行世界》	《旅行世界》
Travis CU	travis_cu
Traypl	Traypl
金银岛	宝岛
零食	零食
Trebol	Trebol
树英语	树_英语
Tree House 教育和配件	Tree_House _教育_&_配件
生命之树	生命之树
Treff	Treff
Treff-Hotels	Treff-Hotels
迷航石油公司	Trek_Petrolem

名称	ID
Trek Turizm	Trek_Turizm
Trekpleister	Trekpleister
Trellidor	Trellidor
特里蒙特信用合作社	特里蒙特信用联盟
特伦迪亚诺	特伦迪亚诺
Trendwest 度假村	TrendWest_度假村
Trenery	Trenery
特伦太尔	特伦太尔
Tresera	Tresera
侵入	侵入
Trgopromet	Trgopromet
Tri Boro FCU	tri_boro_fcu
三城国家银行	三城国民银行
三环马达	三环电机
试用	试用
Trial-Sport	Trial-Sport
Tribhovandas Bhimji Zaveri	Tribhovandas_Bhimji_Zaveri
Triburbana	Triburbana
致敬作品集	致敬_作品集
Tricana	Tricana

名称	ID
针织	针织
三叉戟	三叉戟
Trifarma	Trifarma
三位一体	三位一体
Trinity He	Trinity_Health
Triodos Bank	Triodos_Bank
Tripes Hotpot	Tripes_Hotpot
Triple O	Triple_O
Tripro	Tripro
Triton 眼镜	Triton_Eyewear
凯旋国际	胜利_国际
Trius FCU	trius_fcu
Triveni 超市	Triveni_超市
Troefmarkt	Troefmarkt
TrofaSaude	TrofaSaude
Tromba Rija	Tromba_Rija
Trona Valley 社区 FCU	trona_valley_community_FCU
特隆赫姆停车场	trondheim_parkering
特罗珀哈根动物园	Tropehagen_Zoo
热带	热带

名称	ID
热带小屋	Tropical_Hut
热带冰沙咖啡厅	Tropical_Smoothie_Cafe
热带生活	Tropicana_Life
真实价值	真值
Tru by Hilton	Tru_by_Hilton
TruGreen	TruGreen
TruPartner CU	TruPartner_CU
特鲁科	特鲁科
真正的 North FCU	true_North_fcu
真正的宗教	真正的宗教
真实价值	真实价值
真正价值的硬件	真正超值的硬件
真正超值的房屋	真正超值的房屋
True2Form	True2Form
真彩色	真彩色
TrueCore FCU	TrueCore_FCU
TrueWin	TrueWin
值得信赖的银行	Truist_Bank
Truliant FCU	Truliant_fcu
真的 Nolen	Truly_Nolen

名称	ID
Trumpchi	Trumpchi
中阮咖啡	Trung_Nguyen_Coff
特鲁萨迪	特鲁萨迪
Trust-Mart	Trust-Mart
TrustBank	TrustBank
Trustco 银行	Trustco_Bank
Trustmark 国家银行	Trustmark_国民银行
特鲁弗	特鲁弗
Truworth	Truworth
Truworths	Truworths_
温德姆 Tryp	Tryp_by_Wyndh
tsifrovik	tsifrovik
Taskiris Mallas	Tsakiris_Mallas
Tsarskoe zoloto	tsarskoe_zoloto
Tselitel'	Tselitel'
Tsenopad	Tsenopad
Tsentr Doktora Bubnovskogo	Tsentr_doktora_Bubnovskogo
Tsentr Yuvelir	Tsentr_Yuvelir
tsentr-Invest	tsentr-Invest
TsentrObuv'	TsentrObuv'

名称	ID
TsentroKredit	TsentroKredit
Tsentrtorg	Tsentrtorg
Tsesna Bank	Tsesna_Bank
Tsimus	Tsimus
清达大脑发育	青达_大脑_发展
Tsogo Sun 酒店	Tsogo_Sun_Hotels
Tsubame Grill	tsubame_Grill
翠华餐厅	Tsui_Wah_Restaurant
Tsukemen TETSU	Tsukemen_tetsu
鹤信用合作社	Tsuru_Credit_合作社
鹤丸乌冬面	鹤丸_乌冬面
鹤屋	鹤屋
茨维特·迪瓦诺夫	Tsvet_Divanov
茨维特纳亚	茨维特纳亚
TuTuanna	TuTuanna
Tubby's	Tubby's
地铁站	Tube_Station
Tucci	Tucci
Tuco	Tuco
图森 Old Pueblo CU	Tucson_old_pueblo_cu

名称	ID
图达利	图达利
都铎	都铎
都铎的饼干世界	Tudor's Biscuit_World
星期二早上	星期二_早上
Tuffy	Tuffy
Tugba Kuruyemis	Tugba_Kuruyemis
Tuhu	Tuhu
途易汽车	Tui_Cars
嘟嘟车亚洲街头小吃	嘟嘟车_亚洲_街头_美食
Tuke	Tuke
Tulane-Loyola FCU	Tulane-loyola_FCU
郁金香旅馆	郁金香旅馆
Tully's	Tully's
塔尔萨联邦 CU	Tulsa_federal_cu
滚筒烘干	滚筒_Dry
Tumbleweed Tex Mex	Tumbleweed_Tex_Mex_Grill
Tumi	Tumi
Tune Up	Tune_Up
Tunex	Tunex
突尼斯卡塔尔银行	突尼斯_卡塔尔银行

名称	ID
突尼斯人	突尼斯人
突尼斯租赁	突尼斯_租赁
突尼斯电信	突尼斯_Telecom
屯江拉面	屯江_拉面
Tuo	Tuo
Tuodi	Tuodi
拓江	拓江
特百惠	特百惠
特百惠顾问	特百惠顾问
TupropiaagenciaadeViajes	TupropiaagenciaadeViajes
图兰朵	图兰朵
涡轮增压	涡轮增压
Turgyn y kurylys zhinak banki	turgyn_uy_kurylys_zhinak_banki
Turicum	Turicum
都灵	都灵
Turiscar	Turiscar
土耳其进出口银行	Turk_Eximbank
Turkcel	Turkcel
土耳其山	Turkey_Hill
土耳其银行	土耳其银行

名称	ID
土耳其烧烤餐厅	土耳其烧烤餐厅
土耳其石油	Turkish_Oil
土耳其石油市场	土耳其石油市场
Turkiye Finans Katilim Bankasi	土耳其_金融_Katilim_Bankasi
Turkiye Is Bankasi	Turkiye_Is_Bankasi
特克兰银行	特克兰银行
Turmöl	Turmöl
Turn'n Tender	Turn'n_Tender
特纳建筑	Turner_施工
海龟湾	Turtle_Bay
Tus	Tus
Tus油	Tus_Oil
Tuskys	Tuskys
导师时间托儿服务	Tutor_Time_Child_Care
辅导俱乐部	Tutoring_Club
Tutti Frutti 冷冻酸奶	Tutti_Frutti_Frutti_Frozen_酸奶
Tutti 披萨	Tutti_Pizza
TuttoBene	TuttoBene
Tuzapato	Tuzapato
土尊	土尊

名称	ID
Tver'nefteprodukt	Tver'nefteprodukt
Tverskoy Kupets	Tverskoy_Kupets
Tvoy Doktor	Tvoy_doktor
十二个纸杯蛋糕	十二_纸杯蛋糕
二十四七	二十_四_七
每日两次	每天两次
双重优惠披萨	Twice_The_Deal_Pizz
两倍漂亮	Twice_as_nice
双酒	Twin_Liquors
双峰	Twin_Peaks
Twinner	Twinner
Two Haugnli	Two_Haugnli
Two Peck	Two_Peck
二人组咖啡	二人组_咖啡
错字	错字
轮胎升级版	Tyre_Plus
Tyre-King	Tyre-King
Tyremart	Tyremart
Tyrepower	Tyrepower
轮胎及更多	轮胎_&_更多

名称	ID
Tzomet Sfarim	Tzomet_Sfarim
Tío Pepe	Tío_Pepe
Töjeksperten	Töjeksperten
Türk Hava Yollari	Türk_Hava_Yollari
土耳其石油公司	Türk_Petrol
土耳其电信	Türk_Telekom
TürKoil	TürKoil
TürKoil 市场	土耳其石油市场
Türkiye Petrolleri	Türkiye_Petrolleri
Tüzépker	Tüzépker
U	U
U Clean	U_Clean
U 快车	U_Expres
U Mobile	U_Mobile
U Palycha	U_Palycha
U Plus 酒店	U_Plus_Hotel
你省租车	U_Save_Auto_Rental
U Shop	U_Shop
U Trip 酒店	U_Trip_Hotel
U 位置	u_location

名称	ID
U-Ding 成都冒菜	U-Ding_成都_冒彩
U-FO	U-FO
U-Gas-Um	U-Gas-Um
U 型拖车	U 型拖车
U-Pump-It	U-Pump-It
U 型选择	U 型选择
U-Store-It	U-Store-It
U-wink	U-wink
U.N.D.C. 咖啡	U.N.D.C_COFFEE
U.P. State CU	u.p._state_cu
美国退伍军人事务部	美国退伍军人事务部
美国草坪	Us_Lawns
美国马球	Us._Polo
美国马球协会	U.S_Polo_Assn。
美国信托	Us_Trust
美国披萨	美国披萨
阿联酋交易所	阿联酋交易所
UBA	UBA
UBCI	UBCI
UBI Banca	ubi_banca

名称	ID
UBL Omni	ubl_omni
UBRR	UBRR
UBS	UBS
UBS 标签	UBS_AG
瑞银私人银行	UBS_BANCA_PRIVADA
USBUS SA	UBS_SA
UBuild它	UBuild它
加州大学快车	UC_Expr
UCC Cafe Mercado	ucc_cafe_mercado
UCI	UCI
UCI Kinowelt	UCI_kinowelt
UCZ	UCZ
UD 卡车	UD 卡车
UDACO	UDACO
UDOMAGG	UDOMAGG
乌冬面	乌冬面
UD	UD
使用	使用
不明飞行物汉堡	UFO_BURGER
UFirst FCU	UFirst_FCU

名称	ID
UGC	UGC
UGO	UGO
UH 便利店	uh_便利店
UIB	UIB
英国	英国
UKREXIMBANK	UKREXIMBANK
超	超
超级烈酒	超级烈酒
UMAY PLUS	UMAY_PLUS
UMB 银行	umb_bank
梅子	梅子
JUN JOUR AILLEURS	UN_JOUR_AILLEURS
CU 叔叔	UNCLE_CU
贝纳通的底色	贝纳通的底色
地下旅程	地下_by_Journeys
UNES	UNES
UNIBANCO	UNIBANCO
独角兽	独角兽
UNIPOL BANCA	UNIPOL_BANCA
团结起来	团结起来

名称	ID
联合之箭	联合之箭
美联航地毯和床	美联航_地毯_和_床
联合鸡	UNITED_CHICK
贝纳通的统一色彩	贝纳通的 UNITED_COLORS_OF
UNIVERCEL	UNIVERCEL
UNIVEREXPORT	UNIVEREXPORT
环球银行	环球银行
UNIVERSAL BANKA AD.D.BEOGRAD	UNIVERSAL_BANKA_A.D.BEOGRAD
UNIX	UNIX
UNIbank	UNIbank
UNKOIL	UNKOIL
UNO FCU	UNO_FCU
联合国人道主义事务协调厅	联合国人道主义事务协调厅
UNS	UNS
UNS 语言	uns_idiomas
UPG	UPG
UPMC 儿童社区儿科	upmc_儿童社区_儿科
UPS 接入点	UPS_接入点
UPS 投递箱	ups_drop_box
UPS 货运服务中心	UPS_Freight_服务中心

名称	ID
UPTEC	UPTEC
上线时间	上线时间
UPaisa	UPaisa
昆士兰大学手机	uq_mobile
URALSIB	URALSIB
都市丛林运动鞋	URBAN_JUNGLE_运动鞋
URBIS PARK	URBIS_PARK
URENDE	URENDE
URH Co Hoteliers	urh_co_Hoteliers
乌鲁木齐市商业银行	乌鲁木齐城市商业银行
URemit	URemit
美国航空	美国航空公司
美国银行	美国银行
美国蜂窝电话	_蜂窝网络
美国冷库	us_cold_Storage
美国社区 CU	us_community_cu
美国出行	USA_移动
美国石油公司	美国_石油
USAgencies	USAgencies
USAlliance FCU	USAlliance_FCU

名称	ID
南加州大学	南加州大学
美国公民及移民局	美国公民及移民局
美国邮政局	美国邮政局
USU 信用合作社	USU_Credit_Uni
UTB	UTB
UTELS 酒店	utels_Hotel
UTERQUE	UTERQUE
乌托波利斯	乌托波利斯
UTStechnologiesltd	UTStechnologiesltd
弗大社区 CU	uva_Community_cu
UVEPAN	UVEPAN
UVK MULTICINES	UVK_MULTICINES
UWOOD	UWOOD
UZ Games	uz_Games
UZZI	UZZI
Ua 电影院	Ua_电影院
Uai farma	uai_farma
Ubc-Cof	Ubc-Cof
Ubskin	Ubskin
Ucar	Ucar

名称	ID
Ucla Health	加州大学洛杉矶分校_健康
Uco 银行	Uco_Bank
udarling 咖啡	udarling_Coff
Udgam Vikas Sahakari Bank	Udgam_Vikas_Sahakari_Bank
上田信用金库	上田信用金库
ueee	ueee
Ufa-Pechat'	Ufa-Pechat'
Ufimneft'	Ufimneft'
Ufone	Ufone
Uglemetbank	Uglemetbank
羽后信用金库	Ugo_Shinkin_Bank
Ugrant	Ugrant
乌古尔中心	Ugur_Center
Ugur Derin Dondurucu	Ugur_Derin_Dondurucu
Ujjivan 小型金融银行	Ujjivan_Small_Finance_Bank
Ukr-Petrol	Ukr-Petrol
乌克兰北美自由贸易协定	乌克兰_北美自由贸易协定
乌克兰商业银行	Ukrainskyi_Biznes_Bank
Ukrainskyi Kapital	Ukrainskyi_Kapital
Ukrbudinvestb	Ukrbudinvestb

名称	ID
Ukrhazbank	Ukrhazbank
Ukrhazprombank	Ukrhazprombank
Ukrinbank	Ukrinbank
Ukrnafta	Ukrnafta
Ukrop 的	Ukrop 的
Ukrsybbank	Ukrsybbank
Ukrtatnafta	Ukrtatnafta
Ul'yanovskfarmatsiya	Ul'yanovskfarmatsiya
乌拉·波普肯	Ulla_Popken
Ullrich	Ullrich
Ulmart	Ulmart
Ulmer Volksbank	Ulmer_Volksbank
阿尔斯特银行	阿尔斯特银行
Ulta	Ulta
终极电子	Ultimate_电子产品
终极人员配备	终极人员配备
超钻石	超钻石
UltraTune	UltraTune
超级法玛最受欢迎	Ultrafarma_最受欢迎
奥特玛	奥特玛

名称	ID
Ulusoy	Ulusoy
Ulybka Radugi	Ulybka_Radugi
Ulysse Nardin	Ulysse_Nardin
Uma	Uma
Umaisushikan	Umaisushikan
鲜味汉堡	Umami_Burgers
鲜味火锅	Umami_Hot_Pot
Umaya	Umaya
Umbrale	Umbrale
遮阳伞	遮阳伞
Umbro	Umbro
海寿司	海寿司
嗯牛奶	嗯_牛奶
乌姆尼亚银行	Umnia_Bank
Umoja	Umoja
Umpqua Bank	Umpqua_Bank
乌纳酒店	Una_酒店
Una Mas	Una_Mas
Unal Kuruyemis	Unal_Kuruyemis
宇纳托托	宇纳托托

名称	ID
不知所措	不知所措
鲍勃叔叔的自助仓库	Uncle_bob'self_Storage
藤屋叔叔	Uncle_Fujiya
K 叔叔	Uncle_K。
秋叔柠檬茶	邱叔叔柠檬茶
叔叔餐厅	叔叔餐厅
茶叔叔	Uncle_Tea
安德玛	under_Armour
安德玛品牌屋	Under_Armour_Brand_H
安德玛工厂大楼	Under_Armour_Factory_
地铁站	地下站
Undiz	Undiz
Unex Bank	unex_Bank
Uni Credit	Uni_Credito
Uni 租房	Uni_Rent
Uni-Mart	Uni-Mart
UniCredit Banca	UniCredit_Banca
UniCredit 罗马银行	UniCredit_Banca_di_Roma
UniCredit 银行	UniCredit_银行
UniEuro	UniEuro

名称	ID
UniGroup	UniGroup
UniPost	UniPost
UniWyo FCU	UniWyo_FCU
uniarts	uniarts
unicaja	unicaja
Unicare药房	Unicare药房
Unicasa 超级市场	unicasa_Supermercados
unichem	unichem
Unico	Unico
联合信贷银行 Bulbank	Unicredit_Bulbank
罗马尼亚联合信贷银行	联合信贷银行_罗马尼亚
Unicredit Tiriac Bank	Unicredit_Tiriac_Bank
Unicambio	Unicambio
Unidas	Unidas
Unifarma	Unifarma
Uniglass P	Uniglass_Pl
unikombank	unikombank
Unimarc	Unimarc
Unimarkt	Unimarkt
unioil	unioil

名称	ID
联合银行	联合银行
联合银行和信托	联盟银行和信托
台湾联合银行	台湾联合银行
加州联合银行	加州联合银行
印度联合银行	印度联合银行
联合之家	联盟之家
工会人寿	工会_生活
联合国民银行	联盟国民银行
联合太平洋铁路	联合太平洋铁路
联合种植者银行	联盟种植者银行
联合储蓄银行	工会储蓄银行
unipet	unipet
uniprix	uniprix
Uniqa Asigurari	Uniqa_Asigurari
优衣库	优衣库
唯一	唯一
独特的气体	unique_Gas
男女皆宜的发型	男女皆宜的发型
Unistre	Unistre
Unisys	Unisys

名称	ID
Unitech	Unitech
联合阿马拉银行	United_Amara_Bank
阿拉伯联合银行	阿拉伯联合银行
美联箭队绿色标签放松身心	United_arrows_green_label_Label_L
联合银行	联合银行
联合银行有限公司	联合银行有限公司
非洲联合银行	非洲联合银行
联合品牌	美联航品牌
保加利亚联合银行	United_Bulgarian_Bank
曼联足球俱乐部	United_cu
美联航支票兑现	美联航支票兑现
联合化学家	United_Chemist
联合椰子种植者银行	United_Coconut_Planters_Bank
联合社区银行	联合社区银行
联合公司	美联航公司
美国消费者联合会 CU	United_cusmers_cu
联合奶农联合会	联合_Dairy_Farmers
联合股票 CU	United_equity_cu
联合快递	美联航快车
联合联邦信用合作社	美联航_联邦_信用合作社

名称	ID
联合食品和商业工人工会	United_Food_&_Commers_Workers_Union
联合杂货店	美联航_杂货店_直销店
海湾联合银行	United_Gulf_Bank
联合遗产 CU	United_Heritage_cu
印度联合保险	美国_印度_保险
联合保险公司	美联航保险公司
联合卫理公会	联合卫理公会教堂
美联航移动	美联航_移动
联合摩托车	美联航_摩托车
联合天然食品	美联航_天然食品
联合石油	美联航_石油
大华银行	美联航海外银行
五旬节联合教会	United_Pentecostal_Church
联合石油公司	美联航_石油公司
联合药房	美联航_药房
美联航租赁	美联航_租赁
美联航租赁航空设备	美联航_租赁_空中设备
美联航租赁公司战壕安全	美联航_租赁_Trench_Safety
美联航运解决方案	美联航_配送_解决方案
美联航网站服务	美联航_网站_服务

名称	ID
美国海军	美国_美国_海军
美国无缝	美国_美国_Seamless
美国钢铁工人联合会	美国钢铁工人联合会
联合自卫工作室	United_Studios_of_self_Def
联合超市	美联航_超市
联合电信财经 FCU	United_Teletech_Financial_FCU
美联航视频	曼联_视频
Unitop 快车	Unitop_Exp
Unitus 社区 CU	unitus_Community_cu
联合银行	Unity_Bank
联合银行有限公司	Unity_Bank_Limite
天主教团结会 FCU	Unity_catholic_FCU
联合医疗保健	Unity_医疗保健
Unity One	unity_one_cu
通用型 1 CU	Universal_1_cu
环球大厦酒店	环球影城之家酒店
通用涂料	通用涂料
环球旅行者	环球旅行者
universam Udachnykh Pokupok	Universam_Udachnykh_Pokupok
宇宙	宇宙

名称	ID
大学和社区 FCU	University__and__community_FCU
缅因大学 CU	University_cu_of_me
大学 FCU	University_FCU
犹他州FCU大学	University_FCU-Uat
大学教育资助委员会	大学_补助金_佣金
夏威夷大学 FCU	夏威夷大学_FCU
路易斯安那大学 FCU	路易斯安那大学_FCU
马里兰大学医学系统	马里兰大学医疗系统
匹兹堡大学医学中心	匹兹堡大学医学中心
Union Cine Cine C	Union_Cine_Ciudad
宇野	宇野
Uno Momento	Uno_Momento
Uno Pizzeria & Grill	Uno_Pizzeria_&_Grill
Uno-X	Uno-X
我们的大众银行圣温德勒乐园	Unsere_Volksbank_St._Wendeler_Land
Untsiya	Untsiya
Uny Oil	Uny_Oil
Uo-Show	Uo-Show
鱼鹿寿司	鱼鹿寿司
Uoroku	Uoroku

名称	ID
鱼仙水产	Uosen_Suisan
Upandin	Upandin
Upi	Upi
Ups	Ups
楼上	楼上
Upstalsboom	Upstalsboom
Ur & Penn	Ur_&_Penn
Ural FD	ural_fd
Razvit 中乌拉尔斯基银行重建	ural'skiy_bank_rekonstrukt_i_razvit
乌拉尔私人银行	乌拉尔私人银行
Uralsib Strakhovanie	uralsib_Strakhovanie
乌拉尔跨境银行	乌拉尔跨境银行
Urba Hotel	Urba_Hotel
都市	都市
都市活跃	都市活跃
城市行为	都市行为
都市汉堡	都市_Burger
城市信用合作社	城市_信用_合作社
城市花园酒店	都市花园酒店
都市孩子	都市_Kids

名称	ID
都市服装商	Urban_Outfitters
城市星球	都市星球
城市初级健康中心	城市初级健康中心
城市搬迁	城市搬迁
都市复兴	Urban_Revivo
都市鞋底	urban_Sole
城市站	都市站
UrbanClean	UrbanClean
Urbis	Urbis
Urner Kantonalbank	Urner_KantonalBank
UsedomRad	UsedomRad
UsetheSun	UsetheSun
乌莎	乌莎
Usimle	Usimle
Uspet	Uspet
Usta Dönerci	Usta_Dönerci
犹他州第一FCU	Utah_first_FCU
Utile	Utile
Utkal Grameen 银行	Utkal_Grameen_Bank
乌特科诺斯	乌特科诺斯

名称	ID
Utn-Vostok	Utn-Vostok
乌托邦	乌托邦
Uttar Banga Kshetriya Gramin Bank	Uttar_Banga_Kshetriya_Gramin_Bank
北比哈尔邦格拉明银行	Uttar_Bihar_Gramin_Bank
北方邦合作银行	Uttar_pradesh_Co-operative_Bank
北阿坎德邦格拉明银行	Uttarakhand_Gramin_Bank
Uysal	Uysal
Uyum	Uyum
Uyuterra	Uyuterra
V 2 Shagakh	V_2_Shagakh
V Mart	V_Mart
V 类药物	v_drug
V 星	v_star
V&D	V&D
v'Lille	v'Lille
V-LAZER	V-LAZER
V-Markt	V-Markt
V 型变速箱	V 型变速箱
V-und RB Saale-Unstrut	v-und_rb_saale-unstrut
V.SHOP	V.SHOP

名称	ID
V2K	V2K
V8 酒店	V8_HOTEL
VAB 银行	vab_bank
VAHALA maso-uzeniny	vahala_maso-uzeniny
VAKA	VAKA
萨拉热窝的VAKUFSKA BANKA	VAKUFSKA_BANKA_DD_SAREVOO
VALAMAR	VALAMAR
VALCARCE	VALCARCE
VALECUATRO	VALECUATRO
华伦天奴	华伦天奴
巴利亚多利德	巴利亚多利德
瓦莱韦尔德	瓦莱韦尔德
瓦尔特拉	瓦尔特拉
VAMUN	VAMUN
范克里夫	VAN_CLIFF
VAN DER VALK	VAN_DER_VALK
VAN HEUSEN	VAN_HEUSEN
VAN SCHAIK	VAN_SCHAIK
面包车的部门。商店	VAN'S_DEPT。 _STORE
香草	香草

名称	ID
虚荣	虚荣
货车从墙上掉下来	VANS_OFF_THE_WALL
VAPAA VALINTA	VAPAA_VALINTA
VAPIANO	VAPIANO
VARAN	VARAN
VARTEKS	VARTEKS
VARUS	VARUS
大桶酒	VATS_LIQUI
沃克斯豪尔	沃克斯豪尔
VB Backnang	vb_backnang
VB Bad Driburg-Brakel-Steinheim	vb_bad_driburg-Brakel-Steinheim
VB Bad Saulgau	vb_bad_Saulgau
VB Baden-Baden Rastatt	vb_baden-Baden_Baden_Rastatt
BB Bitburg	vb_bitburg
VB Bramgau im Osnabrücker Land	vb_bramgau_im_osnabrücker_Land
VB Breisgau-Sud	vb_breisgau-süd
VB Büren and Salzkotten	vb_büren_und_salzkotten
VB Chemnitz	vb_Chemnitz
VB Darmstadt-Kreis Bergstraße	vb_darmstadt-kreis_bergstraße
VB Dill-Vb 和 Rb	vb_dill-vb_und_rb

名称	ID
VB Erkelenz-Hückelhoven-Wegberg	vb_erkelenz-Hückelhoven-Wegberg
VB Ettlingen	vb_Ettlingen
VB 弗赖堡	vb_Freiburg
VB Glan-Munchweiler	vb_glan-Munchweiler
VB Gronau-Ahaus	vb_Gronau-ahaus
VB Herrenberg-Nagold-Rottenburg	vb_Herrenberg-Nagold-Rottenburg
VB Heuchelheim	vb_HeuchelHeim
VB Hildesheim-Lehrte-Pattensen	vb_Hildesheim-Lehrte-Pattensen
VB Hohenlohe	vb_hohenlohe
VB Kaiserslautern-Nordwestpfalz	vb_Kaiserslautern-nordwestpfalz
VB Kirnau	vb_kirnau
VB Kraichgau	vb_kraichGau
VB Lauterbach-Schlitz	vb_Lauterbach-Schlitz
VB Lauterecken	vb_Lauterecken
VB 路德维希堡	vb_Ludwigsburg
VB Magdeburg	vb_Magdeburg
VB Main-Taunus	vb_main-taunus
VB Meßkirch RB	vb_meßkirch_rb
VB Mittleres Erzgebirge	vb_mittleres_erzgebirge
VB Möckmühl-Neuenstadt	vb_meckmühl-Neuenstadt

名称	ID
VB Nordoberpfalz	vb_nordoberpfalz
VB Nordschwarzwald	vb_nordschwarzwald
VB Osterburg-Luchow-Dannenberg	vb_osterburg-Lüchow-dannenberg
VB Ostlippe	vb_ostLippe
VB 普福尔斯海姆	VB_Portheim
VB Rb Eichstátt	vb_rb_eichstátt
VB Reiste-Eslohe	vb_reste-eslohe
B Rhein-Lahn	vb_Rhein-lahn
VB Riesa	vb_riesa
VB Rion Leonberg	vb_rion_Leonberg
B Rottweil	vb_rottweil
B Saaletal	vb_saalEtal
VB Saar-West	vb_saar-West
VB Saarlouis	vb_saarLouis
VB Schmallenberg	vb_schmallenberg
VB Schwábisch Gmünd	vb_schwébisch_gmünd
B Sulmtal	vb_sulmtal
VB Tubingen	vb_Tubingen
B Untere Saar	vb_untere_Saar
VB Wilferdingen-Keltern	vb_Wilferdingen-Keltern

名称	ID
VB Worms-Wonnegau	vb_worms-wonnegau
VB 和 Rb Eisenach	vb_und_rb_Eisenach
VB-RB Deggingen	vb-rb_deggingen
VB-RB Dingolfing	vb-rb_dingolfing
VB-RB Laupheim-Illertal	vb-rb_laupheim-illertal
VB-RB Meißen Großenhain	vb-rb_meißen_GroßenHain
VB-RB Rhön-Grabfeld	vb-rb_rhön-Grabfeld
VB-RB Riedlingen	vb-rb_Riedlingen
VBRB 曼格法尔塔尔-罗森海姆	vbrb_mangfalltal-Rosenheim
VBSP	VBSP
位于内特兰的 VBU Volksbank	vbu_volksbank_im_unterland
VCA	VCA
VCA 动物医院	VCA_Animal_Hospitals
VCub	VCub
VDB	VDB
VDK SPARBANK	VDK_SPARBANK
VDV	VDV
VELO 和氧气	velo_&_Oxyge
VENETI	VENETI
威尼托银行	VENETO_BANCA

名称	ID
威尼托银行	VENETO_BANKA
威尼斯	威尼斯
VENTOLiNi	VENTOLiNi
VEPSÁLAINEN	VEPSÁLAINEN
VERIMARK	VERIMARK
VERNER	VERNER
VERO MODA	VERO_MODA
范思哲	范思哲
范思哲系列	范思哲系列
背心之家	背心之家
背心	背心
VF Outlets	VF_Outlets
VHA	VHA
vi-butikerna	vi-butikerna
VIAGGIO	VIAGGIO
VIAJAR Y AYUDAR	VIAJAR_Y_AYUDAR
VIAJES El Corte Inglés	viajes_el_Corte_inglés
VIAJES SERCOM	VIAJES_SERCOM
VIB	VIB
VIBRACauto	VIBRACauto

名称	ID
VIC	VIC
维多利亚州市场	Vic_Marke
VICKO	VICKO
VICKY FORM	VICKY_FORM
维克多·雨果	VICTOR_HUGO
维多利亚	维多利亚
视频蓝色	视频_蓝色
维也纳之家	维也纳之家
VIESHOW 电影院	Vieshow_Cinemas
乡村酒店	乡村酒店
VINAMILK	VINAMILK
VINASIAM 银行	Vinasiam_Bank
芬奇公园	VINCI_PARK
VINCOM	VINCOM
贵宾	贵宾
贵宾系列	VIP_COLLECT
VIPnet	VIPnet
视觉实验室	视觉实验室
VISTA ALEGRE	VISTA_ALEGRE
VISTAOPTICA	VISTAOPTICA

名称	ID
维斯瓦河	维斯瓦河
VITA 独家	VITA_独家
VITAFARM	VITAFARM
重要的牙齿	VITAL_DENTE
活力白	VITAL_WHITE
VITALDENT	VITALDENT
VITO	VITO
VITRUM	VITRUM
VIVA	VIVA
VIVA CURRY	VIVA_CURRY
VIVACOM	VIVACOM
VIVARA	VIVARA
VIVECO	VIVECO
VIVI BRIDE 照片设计和连衣裙	VIVI_BRIDE_PHOTO_DESIGN_AND_DRESS
VKB-Bank	VKB-Bank
VLCC	VLCC
VLENS	VLENS
VO5	VO5
VOBIS	VOBIS
VOG 发型	VOG_Coiffure

名称	ID
VOI & NOI	VOI_&_NOI
VOLI	VOLI
塞尔维亚大众银行	VOLKSBANK_SRBIJA
伏特	伏特
沃尔沃	沃尔沃
沃尔沃卡车中心	沃尔沃卡车中心
VOX Meble	vox_meble
VOY A VERTE	VOY_A_VERTE
赛车燃料副总裁	VP_Racing_Fuels
VPBank	VPBank
VPK 城市合作社信贷协会	vpk_urban_Co-op_credit_Societ
VPharma	VPharma
VR	VR
班贝格虚拟现实银行	VR_bank_Bamber
VR 银行比登科普夫-格拉登巴赫	vr_bank_biedenkopf-Gladenbach
VR Bank Dinkelsbühl	vr_bank_dinkelsBühl
弗伦斯堡-石勒苏益格VR银行	vr_bank_Flensburg-石勒苏益格
虚拟现实银行 HessenLand	VR_bank_ HessenLand
VR 银行霍夫	vr_bank_hof
VR Bank Kaufbeuren-Ostallgau	vr_bank_kaufbeuren-ostallgau

名称	ID
基钦根虚拟现实银行	vr_bank_kitzingen
Lausitz VR 银行	vr_bank_Lausitz
VR Bank Main-Kinzig-B üdingen	vr_bank_main-kinzig-Büdingen
VR Bank Mittelhaardt	vr_bank_mittelHaardt
VR 银行慕尼黑乐园	VR_bank_München_Land
VR Bank Niebüll	vr_bank_niebüll
纽伦堡虚拟现实银行	VR_Bank_Nürb
VR Bank Ostholstein Nord-Plön	vr_bank_ostholstein_nord-Plön
VR Bank Pinneberg	VR_bank_Pinneberg
莱茵-摩泽尔虚拟现实银行	vr_bank_Rhein-Mosel
莱茵-内卡虚拟现实银行	vr_bank_Rhein-neckar
Rosenheim-Chiemsee VR 银行	vr_bank_rosenheim-Chiemsee
VR Bank Saarpfalz	vr_bank_saarpfalz
VR Bank Schwébisch Hall-Crailh	vr_bank_schwébisch_Hall-Crailham
VR Bank Südliche Weinstraße-Wasgau	vr_bank_Südliche_weinstraße-Wasgau
VR Bank Südpfalz	vr_bank_südpFalz
Südthüringen VR 银行	vr_bank_Südthüringen
VR Bank Südwestpfalz	vr_bank_südwestpfalz
VR Bank Untertaunus	vr_bank_untertaunus
Westhüringen VR 银行	vr_bank_Westhüringen

名称	ID
VR Bank im Enzkreis	vr_bank_im_enzkreis
VR Genossenschaftsbank Fulda	vr_genossenschaftsBank_Fulda
VR Meine Raiffeisenbank	vr_meine_raiffeisenBank
VR-Bank Aalen	vr-bank_Aalen
VR-Bank Alb	vr-bank_alb
VR-Bank 阿尔滕堡乐园	VR-bank_Altenburger_Land
VR-Bank Asperg-Markgröningen	vr-bank_asperg-Markgröningen
VR-Bank B.Gladbach-Overath-Rösrath	vr-bank_b.gladbach-overath-Rösrath
VR-Bank Bad Hersfeld-Rotenburg	vr-bank_bad_Hersfeld-Rotenburg
VR-Bank Bad Kissingen-Bad Brückenau	vr-bank_bad_kissingen-bad_brückenau
VR-Bank 巴特萨尔聪根施马尔卡尔登	vr-bank_bad_salzungen_Schmalkalden
VR-Bank 拜罗伊特	VR-bank_Bayreuth
VR-Bank 波恩	VR-bank_Bonn
VR-Bank Coburg	VR-bank_coburg
VR-Bank Elwangen	VR-bank_Ellwangen
VR-Bank Erl。 -Höchst。 -Herzogenaurach	vr-bank_erl。 -Höchst。 -Herzogenaurach
VR-Bank Fichtelgebirge	vr-bank_fichtelgebirge
VR-Bank Flaming	VR-Bank_Flaming
VR-Bank Fürstenfeldbruck	vr-bank_fürstenfeldBruck
VR-Bank Fürstenwalde Seelow Wriezen	vr-bank_fürstenwalde_seelow_Wriezen

名称	ID
VR-Bank Gerolzhofen	vr-bank_Gerolzhofen
VR-Bank Hunsrück-Mosel	vr-bank_hunsrück-Mosel
VR-Bank Landau	vr-bank_Landau
VR-Bank Lech-Zusam	vr-bank_lech-zusam
VR-Bank Lkr。加米施-帕滕基兴	vr-bank_LKR。 _加米施-帕滕基兴
VR-Bank Memmingen	vr-bank_memmingen
VR-Bank 米特尔弗兰肯西部	vr-bank_mittelfranken_West
VR-Bank 中萨克森州	VR-bank_Mittelsachsen
VR-Bank Neu-Ulm/Weißenhorn	vr-bank_neu-ulm/Weißenhorn
VR-Bank Neuwied-Linz	vr-bank_neuwied-Linz
VR-Bank Niederschlesien	vr-bank_niederschlesien
VR-Bank Nordeifel	vr-bank_NordeiFel
VR-Bank Nordrhön	vr-bank_nordrhön
VR-Bank Oberbayern Sundost	vr-bank_Oberbayern_Südost
VR-Bank Passau	VR-bank_Passau
VR-Bank Raiffeisenbank 拜仁中部	vr-bank_raiffeisenbank_bayern_mitte
VR-Bank Rhein-Erft	vr-bank_Rhein-erft
VR-Bank Rhein-Sieg	vr-bank_Rhein-Sieg
VR-Bank Rhön-Grabfeld	vr-bank_rhön-Grabfeld
VR-Bank Rothenburg o. D. Tbr	vr-bank_rothenburg_o._d._TBR。

名称	ID
VR-Bank Rottal-Inn	vr-bank_rottal-Inn
VR-Bank Schweinfurt	VR-bank_Schweinfurt
VR-Bank Stromberg-Necka	vr-bank_stromberg-neckar
VR-Bank Taufkirchen-Dorfen	vr-bank_taufkirchen-Dorfen
VR-Bank Uckermark-Randow	vr-bank_uckermark-randow
VR-Bank Uffenheim-Neustadt	vr-bank_uffenheim-Neustadt
VR-Bank Vilsbiburg	VR-bank_vilsbiburg
VR-Bank Werra-Meißner	vr-bank_werra-Meißner
VR-Bank Westmunsterland	VR-bank_Westmünsterland
VR-Bank Westpfalz	vr-bank_Westpfalz
伦茨堡县的 VR-Bank	vr-bank_im_kreis_rendsburg
位于米特尔巴登的 VR-Bank	vr-bank_in_mittelbaden
VRB	VRB
VRK	VRK
VRNnextbike	VRNnextbike
VSECU	VSECU
VSK	VSK
VTB	VTB
VTB 24	VTB_24
VTK	VTK

名称	ID
VTNZ	VTNZ
VU	VU
VULCO	VULCO
VUS	VUS
Vuz-Bank	Vuz-Bank
VVB Dillingen Dudweiler Sulzb。 /Saar	vvb_dillingen_dudweiler_sulzB。 /Saar
大众普通座椅	vw_common_Seat
Vaango	Vaango
Vabi Sabi	Vabi_Sabi
Vaca Nostra 餐厅	Vac_Nostra_Restaurante
Vacances Bleues	Vacances_Bleues
假日配饰	假期_配饰
江诗丹顿	江诗丹顿
Vagabond Inn	Vagabond_Inn
Vaillant	Vaillant
Vait	Vait
Vakifbank	Vakifbank
Vakko	Vakko
淡水河谷	Val_Verde
情人节	情人节

名称	ID
瓦伦丁酒店	情人节酒店
瓦莱罗	瓦莱罗
瓦莱罗街角商店	Valero_Corner_Store
瓦莱塔	瓦莱塔
英勇银行	Valiant_Bank
Valiexchimp	Valiexchimp
山谷银行	Valley_Bank
Valley Isle 社区 FCU	valley_isle_community_FCU
Valley Oak	valley_Oak_cu
ValleyStar CU	ValleyStar_CU
Valleygirl	Valleygirl
Valmont tabak	Valmont_Tabák
Valpak	Valpak
超值之王	Valu_King
Valu-mart	Valu-mart
ValuDollar	ValuDollar
超值城市家具	Value_城市_家具
超值地点	价值_位置
超值村	Value_Vil
Vale-Rite	Vale-Rite

名称	ID
Valvoline 即时换油	Valvoline_Instant_Oil_Change
Vamin	Vamin
Van Cleef & Arpels	Van_Cleef_&_Arpels
Van Graaf	Van_Graaf
范海尔登药房	Van_Heerden_Pharmacy
Van Houtte	Van_Houtte
Van Raak	Van_Raak
Van Reeth	Van_Reeth
vanGO	vanGO
Vancity 信用合作社	Vancity_Credit_Un
温哥华海岸健康	温哥华_海岸_健康
温哥华之翼	温哥华之翼
Vanden Borre	Vanden_Borre
先锋清洁	先锋_清洁
万豪证券	万豪证券
名利场酒店	Vanity_Hotels
货车	货车
华帝药房	Vantage_药房
Vantage West CU	vantage_west_cu
Varan Kargo	Varan_Kargo

名称	ID
Varsovienne	Varsovienne
瓦萨健身	瓦萨_健身
瓦萨比	瓦萨比
Vasan 眼部护理	Vasan_Eye_Care
巴斯卡拉	巴斯卡拉
Vash Doktor	Vash_doktor
Vasha apteka	vasha_apteka
Vashe zdorov'ie	vashe_zdorov'ie
Vashe Zoloto	Vashe_Zoloto
Vasvári Takarek	Vasvari_Takarék
Vatan 电脑	Vatan_computer
梵蒂冈酒店	梵蒂冈酒店
Vattenfall InCharge	Vattenfall _ InCharge
Vaz	Vaz
Vea Cencosud	Vea_Cencosud
贝西诺	贝西诺
Vector	Vector
Vedes	Vedes
Vee 的视频	Vee's 视频
Vef Banka	Vef_Banka

名称	ID
纯素生有机食品	Vegan_Raw_Organic
Veikon Kone	Veikon_Kone
Vel Pitar	Vel_Pitar
Velikolukskiy myasokombinat	velikolukskiy_myasokombinat
Vellore Dist 中央合作银行	vellore_dist_central_Co-op_Bank
Velly	Velly
Velyka Kyshenia	Velyka_Kyshenia
Venezolano de Crédito	venezolano_de_Crédito
Venizi	Venizi
Venizia Ice	Venizia_Ice
风险投资银行	风险投资银行
冒险旅馆	Venture_Inn
Ventus	Ventus
维纳斯国际酒店	维纳斯_国际_酒店
维纳斯皇家酒店	Venus_royal_Hotel
威立雅环境服务	Veolia_环境_服务
Ver	Ver
Ver。VB Regionalverband Saarbrücken	Ver。 _vb_regionalverband_saarbrücken
维拉	维拉
维拉·布拉德利	Vera_Bradley

名称	ID
维拉披萨	Vera_Piz
verb.sparkasse Goch-Kevelaer-Weeze	verb.sparkasse_goch-kevelaer-Weeze
Verbands-Sparkasse Wesel	Verbands-Sparkasse_Wesel
VerbundSparkasse Emsdetten-Ochtrup	VerbundSparkasse_Emsdetten-Ochtrup
Verdemar	Verdemar
Vereinigte RB Burgstädt	vereinigte_rb_burgstädt
Vereinigte Sparkasse im Mark。Kreis	vereinigte_sparkasse_im_mark。 _Kreis
Eschenbach 联合储蓄银行埃申巴赫	Vereinigte_Sparkassen_Eschenbach
Vereinigte Sparkassen 贡岑豪森	Vereinigte_Sparkassen_Gunzenhausen
联合大众银行	Vereinigte_Volksbank
林堡联合大众银行	Vereinigte_Volksbank_Limburg
联合大众银行 Maingau	Vereinigte_Volksbank_Maingau
明斯特联合大众银行	Vereinigte_Volksbank_Münster
联合大众银行 Raiffeisenbank	Vereinigte_Volksbank_Raiffeisenbank
Vergölst	Vergölst
Verhage	Verhage
Veritas FCU	Veritas_FCU
Verity CU	verity_cu
威瑞森无线	Verizon_Wireless
维尔卡乳业	Verka_Dairy

名称	ID
Verkkokauppa	Verkkokauppa
佛蒙特州 FCU	Vermont_FCU
Vern Fonk	Vern_Fonk
Vern's	Vern's
Verniy	Verniy
Veronika Maine	Veronika_Maine
Veropoulos Spar	Veropoulos_Spar
Versato	Versato
对比银行	vsus_Bank
非常泰式的面条	Very_Thai_Noods
Veshch'	Veshch'
灶神星	灶神星
Vestel	Vestel
Vestino	Vestino
Vestjysk 银行	Vestjysk_Bank
兽医事务	兽医事务
Veta 英语口语课程	Veta_英语口语_课程
Vetcoclinics	Vetcoclinics
外国战争的退伍军人	外国战争的退伍军人
Vi	Vi

名称	ID
途经巴西	Via_Brasil
通过 CU	via_cu
通过 Uno	Via_Uno
威尼托大街	Via_Veneto
Viada	Viada
Viaero 无线	Viaero_无线
Viajes Crisol	Viajes_Crisol
厄瓜多尔之旅	维亚杰斯_厄瓜多尔
Viajes Iltrida	Viajes_Iltrida
Viajes Incavisa	Viajes_Incavisa
Viajes Levante Groupo	Viajes_Levante_Grupo
Viajes Marper Tours	Viajes_Marper_Tours
Viajes Tabora 之旅	Viajes_Tabora_Tours
维亚耶斯大学	Viajes_大学
Viamo	Viamo
Viana	Viana
Vianor	Vianor
Viatges Caprabo	Viatges_Caprabo
氛围	氛围
Vibgyor 学校集团	vibgyor_group_of_schools

名称	ID
Vibo	Vibo
Vicco	Vicco
反之亦然	Vice_Vers
维多利亚	维多利亚
维多利亚高尔夫	维多利亚高尔夫
维多利亚站	维多利亚站
维多利亚的秘密	维多利亚的秘密
胜利	胜利
胜利教育	胜利_教育
胜利餐厅	胜利餐厅
Victra	Victra
Vida Farmácias	Vida_Farmácias
维达尔	维达尔
视频书房	视频_Den
视频小镇	视频城镇
Videocon	Videocon
Videoland	Videoland
Vidharbha Konkan Gramin 银行	Vidharbha_Konkan_Gramin_Bank
Vidya Sahakari Bank	Vidya_Sahakari_Bank
维也纳	维也纳

名称	ID
维也纳经典酒店	维也纳_经典酒店
维也纳酒店	维也纳_酒店
维也纳国际	维也纳_国际
维也纳国际酒店	维也纳_国际_酒店
Vinthonga	Vinthonga
越南资本银行	越南资本银行
越南菜	越南菜
越南 ABank	越南 ABank
VietBank	VietBank
越南外贸股份商业银行	越南外贸股份商业银行
VietinBank	VietinBank
Viettel	Viettel
越南人	越南人
Vifille	Vifille
维尼昂学校	Vignan_Schools
Vigoss 牛仔裤	Vigoss_Jeans
Vijay 诊断	Vijay_诊断
维杰·塞勒斯	Vijay_Sales
维贾亚南德旅行社	Vijayanand_Travels
维克酒店集团	Vik_酒店集团

名称	ID
维京人	维京人
Viko	Viko
维克托里亚	维克托里亚
维克托斯	维克托斯
维拉	维拉
Vila Galé	Vila_Galé
Viled	Viled
Vilgud	Vilgud
Vilka-Lozhka	Vilka-Lozhka
别墅市场	别墅市场
比萨别墅	别墅_Pizza
德尔玛别墅	Villa_del_Mar
Villacastín Med. 美容和牙科	Villacastin_Med。 _estética_y_dental
乡村电影院	乡村电影院
乡村旅馆	Village_Inn
乡村储藏室	乡村_储藏室
村庄酒店	Villages_Hotel
维尔福	维尔福
Villeroy & Boch	Villeroy_&_Boch
维卢普拉姆区中央合作银行	Villupuram_dist_central_Co-op_Bank

名称	ID
Vimos	Vimos
VinFast	VinFast
VinFast 电动车	VinFast_EV
VinFast-雪佛兰	VinFast-雪佛兰
VinMec	VinMec
VinSchool	VinSchool
VinWonders	VinWonders
Vina Giay	Vina_Giay
Vina Kubani	Vina_Kubani
Vinaphone	Vinaphone
Vinarstvi U Kaplicky	Vinarstvi_U_Kaplicky
Vinayagar 发起人和建设者	Vinayagar_Promoters_&_Builders
Vincci	Vincci
Vincci 服装	Vincci _服装
Vincom 店屋	Vincom_Shophous
Vindija	Vindija
Vinhomes	Vinhomes
Vinmonopolet	Vinmonopolet
Vino & Destilaty	Vino_&_Destilaty
Vino Vino	Vino_Vino

名称	ID
Vinos Yamazaki	Vinos_Yamazaki
Vinoteka Vinofol	Vinoteka_Vinofol
珍珠酒庄	珍珠酒庄
珍珠珍珠高尔夫	Vinpearl_Golf
珍珠岛野生动物园	Vinpearl_Safar
复古	复古
复古酒窖	复古酒窖
Vinzenzmurr	Vinzenzmurr
Vipoil	Vipoil
Vips	Vips
维珍活跃	处女活跃
维珍藏品	处女系列
维珍货币	Virgin_Mon
维珍手机	维珍手机版
弗吉尼亚广播公司	Virginia_abc
virsi-A	virsi-A
vis-a-vi	vis-a-vi
相对于	相对于
Vishal Mega Mart	Vishal_Mega_Mart
可见的变化	可见的变化

名称	ID
Visiolab	Visiolab
愿景银行	Vision_Banco
愿景快车	愿景快报
愿景金融 FCU	Vision_Financial_FCU
Vision+	Vision_PI
视觉商店	Vision_Stor
视觉快车	视觉快车
愿景 FCU	Visions_FCU
愿景工厂	愿景工厂
Visique 验光师	Visique_验光师
来访的天使	来访_Angels
Vissan	Vissan
维斯塔光学	远景光学
Vistar 汽车玻璃	Vistar_Auto_Glass
维塔	维塔
Vita Green	VitaGreen
Vita Liberté	Vita_Liberté
Vita 专卖店	Vita_Specialty_stor
Vitacare	Vitacare
Vitafoam	Vitafoam

名称	ID
Vitaland	Vitaland
Vitalant	Vitalant
维塔利亚改革之家	Vitalia Reformhaus
Vitalys	Vitalys
Vitalyur	Vitalyur
Vitamed	Vitamed
维生素世界	Vitamin_World
维生素商店	维生素商店
维他命	维他命
Vitimex	Vitimex
维特拉	维特拉
Vitsec	Vitsec
Vitusapotek	Vitusapotek
Viva Chickin	Viva_Chicken
Viva 酒店及度假村	Viva_酒店及度假村
Viva la Vie	viva_la_vie
VivaFit	VivaFit
Vivahotel	Vivahotel
Vival	Vival
Vivanda	Vivanda

名称	ID
Vivanta	Vivanta
Vive Futura	Vive_Futura
Vivechrom	Vivechrom
Viveks-无限商店	Viveks-The_Unlimited_Shop
Vivenda do Camarao	vivenda_do_Camarão
Vives	Vives
Vivianda	Vivianda
薇薇安·韦斯特伍德	Vivienne_Westwood
Vivikes	Vivikes
Vivo	Vivo
Vivo 美式披萨和帕尼尼	Vivo_American_Pizza_&_Panini
Viyella	Viyella
Viz Nafta	Viz_Nafta
VkusVill	VkusVill
Vkusnolubov	Vkusnolubov
Vladimir-Opton	Vladimir-Opton
Vladprombank	Vladprombank
弗莱科娃	弗莱科娃
Vneshprombank	Vneshprombank
Vnp	Vnp

名称	ID
Vocelli 披萨	Vocelli_Pizz
Voco	Voco
Vodacare	Vodacare
Vodacom	Vodacom
Vodacom 4u Stores	vodacom_4u_Stores
Vodashop	Vodashop
Voegtlin-Meyer	Voegtlin-Meyer
Voglia	Voglia
Vojvodanska banka	vojvodanska_Banka
Volapié	Volapié
Volaria 旅行	Volaria Travel
Volcom	Volcom
Volg	Volg
伏尔加石油公司	伏尔加石油公司
Volganefteproduc	Volganefteproduc
伏尔加农场	伏尔加农场
Volks	Volks
Volks-und Raiffeisenbank 普里格尼茨	Volks-und_raiffeisenBank_Prignitz
Volks-und Raiffeisenbank Saarpfalz	volks-und_raiffeisenBank_saarpfalz
Volks-und Raiffeisenbank 维斯马	Volks-und_raiffeisenBank_Wismar

名称	ID
大众银行	大众银行
大众银行 Achern	Volksbank_Achern
阿尔伯斯塔特大众银行	Volksbank_Albstadt
Volksbank Allgäu-West	Volksbank_Allgäu-West
大众银行 Alzey	Volksbank_Alzey
大众银行 Alzey-Worms	Volksbank_Alzey-Worms
Volksbank Bad Oeynhausen-Herford	Volksbank_Bad_Oeynhausen-Herford
Volksbank Bad Salzuflen	Volksbank_Bad_Salzuflen
巴林根大众银行	大众银行_巴林根
大众银行 Beckum	大众银行_Beckum
大众银行 Bielefeld-Gütersloh	Volksbank_Bielefeld-Gütersloh
大众银行 Bigge-Lenne	Volksbank_Bigge-Lenne
波鸿大众银行 Witten	Volksbank_Bochum_Witten
大众银行 Bonn Rhein-Sieg	Volksbank_Bonn_Rhein-Sieg
大众银行 BraWo	大众银行_ BraWo
大众银行 Brackenheim-Güglingen	Volksbank_Brackenheim-Güglingen
大众银行 Braunschweig Wolfsburg	Volksbank_Braunschweig_Wolfsburg
大众银行 Breisgau Nord	大众银行_Breisgau_Nord
不来梅北部大众银行	大众银行_不来梅北部
大众银行 Bremerhaven-Cuxland	Volksbank_Bremerhaven-Cuxland

名称	ID
布伦茨塔尔大众银行	大众银行_Brenztal
大众银行 Brilon-Büren-Salzkotten	Volksbank_Brilon-Büren-Salzkotten
Bulksbank Bruchsal-Bretten	Volksbank_Bruchsal-Bretten
大众银行 Bruhrain-Kraich-Hardt	Volksbank_Bruhrain-Kraich-Hardt
大众银行 Börde-Bernburg	Volksbank_Berde-Bernburg
Bulksbank Bühl	大众银行_Bühl
大众银行 Darmstadt-Südhessen	Volksbank_Darmstadt-Südhessen
大众银行 Delbrück-Hövelhof	Volksbank_Delbrück-Hövelhof
德绍-安哈尔特大众银行	Volksbank_dessau-Anhalt
大众银行 Donau-Neckar	大众银行_Donae-Neckar
大众银行 Dornstetten	Volksbank_Dornstetten
Dreieich Volksbank	Volksbank_Dreieich
Volksbank Dreilandereck	Volksbank_Dreilandereck
德累斯顿-包岑大众银行	大众银行_Dresden-Bautzen
杜伦大众银行	Volksbank_Düren
杜塞尔多夫诺伊斯大众银行	大众银行_杜塞尔多夫_诺伊斯
埃宾根大众银行	大众银行_埃宾根
大众银行 Eichsfeld-Northeim	Volksbank_Eichsfeld-Northeim
埃菲尔大众银行	Volksbank_Eifel
大众银行 Eifel Mitte	Volksbank_Eifel_Mitte

名称	ID
Volksbank Einbeck	Volksbank_Einbeck
大众银行 Elmshorn	Volksbank_Elmshorn
大众银行 Elsterland	Volksbank_Elsterland
大众银行 Emmerich-Rees	大众银行_Emmerich-Rees
Volksbank Erft	Volksbank_Erft
Volksbank Ermstal-Alb	Volksbank_Ermstal-Alb
埃斯林根大众银行	大众银行_埃斯林根
奥伊斯基兴大众银行	大众银行_奥伊斯基兴
大众银行 Eutin Raiffeisenbank	大众银行_Eutin_Raiffeisenbank
福希海姆大众银行	Volksbank_Forchheim
弗兰肯大众银行	大众银行_Franken
大众银行 gmhütte-Hagen-Bissendorf	Volksbank_gmhütte-Hagen-Bissendorf
Volksbank Geest	Volksbank_Geest
大众银行 Gera-Jena-Rudolstadt	Volksbank_Gera-Jena-Rudolstadt
Volksbank Grafschaft Hoya	Volksbank_Grafschaft_Hoya
格平根大众银行	大众银行_格平根
哥廷根大众银行	大众银行_哥廷根
居特斯洛大众银行	Volksbank_Gütersloh
大众银行哈雷（萨勒）	Volksbank_Halle_ (Sale)
哈默尔恩-斯塔德哈根大众银行	Volksbank_Hameln-Stadthagen

名称	ID
大众银行哈姆	Volksbank_Hamm
大众银行 Hegau	大众银行_Hegau
海尔布隆大众银行	大众银行_海尔布隆
Volksbank Hellweg	Volksbank_Hellweg
赫尔姆施泰特大众银行	Volksbank_Helmstedt
赫伦贝格-罗滕堡大众银行	Volksbank_Herrenberg-Rottenburg
希尔德斯海姆大众银行	大众银行_希尔德斯海姆
大众银行 Hildesheimer Börde	Volksbank_Hildesheimer_Börde
Hochrhein 大众银行	Volksbank_Hochrhein
霍亨索伦大众银行	Volksbank_Hohenzollern
霍亨索伦-巴林根大众银行	Volksbank_Hohensollern-Balingen
Horb-Freudenstadt Volksbank	Volksbank_Horb-Freudenstadt
Hunsrück-Nahe 大众银行	大众银行_Hunsrück-Nahe
大众银行 Jever	Volksbank_Jever
卡尔斯鲁厄大众银行	大众银行_卡尔斯鲁厄
大众银行 Kinzigtal	Volksbank_Kinzigtal
Volksbank Kirchheim-Nürtingen	Volksbank_Kirchheim-Nürtingen
大众银行 Kleverland	Volksbank_Kleverland
大众银行 Koblenz Mittelrhein	Volksbank_Koblenz_Mittelrhein
康斯坦茨大众银行	大众银行_康斯坦茨

名称	ID
大众银行 Kraichgau Wiesloch-Sinsh。	大众银行_Kraichgau_Wiesloch-Sinsh。
克雷费尔德大众银行	Volksbank_Krefeld
大众银行 Kur-und Rheinpfalz	Volksbank_kur-und_Rheinpfalz
Volksbank Kurpfalz H + G BANK	volksbank_kurpfalz_h_+_g_Bank
大众银行 Karnten	Volksbank_Karnten
Lahr 大众银行	Volksbank_Lahr
劳普海姆大众银行	大众银行_劳普海姆
莱比锡大众银行	大众银行_莱比锡
林根大众银行	大众银行_林根
路德维希堡大众银行	大众银行_路德维希堡
大众银行 Löbau-Zittau	Volksbank_Löbau-Zittau
Lubbecker Lubbecker Land	Volksbank_Lubbecker_Land
吕贝克大众银行	大众银行_吕贝克
吕讷堡海德大众银行	Volksbank_Luneburger_Heide
大众银行 Main-Tauber	Volksbank_Main-Tauber
大众银行 Marl-Recklinghausen	Volksbank_Marl-Recklinghausen
大众银行 Metzingen-Bad Urach	Volksbank_Metzingen-Bad_Urach
大众银行 Mindener Land	Volksbank_Mindener_Land
Volksbank Mittelhessen	Volksbank_Mittelhessen
莫斯巴赫大众银行	Volksbank_Mosbach

名称	ID
门兴格拉德巴赫大众银行	大众银行_门兴格拉德巴赫
明辛根大众银行	大众银行_明辛根
明斯特大众银行	大众银行_明斯特
大众银行 Nahetal	Volksbank_Nahetal
内卡尔塔尔大众银行	Volksbank_Neckartal
诺因基兴大众银行	大众银行_诺因基兴
下莱茵大众银行	Volksbank_Niederrhein
Niederösterreich Volksbank	Volksbank_Niederösterreich
Nordharz 大众银行	大众银行_Nordharz
Nordheide 大众银行	大众银行_诺德海德
北明斯特兰大众银行	大众银行_北明斯特兰
奥伯伯格大众银行	大众银行_Oberberg
大众银行 Oberösterreich	Volksbank_Oberösterreich
奥登瓦尔德大众银行	Volksbank_Odenwald
奥芬堡大众银行	大众银行_奥芬堡
奥尔登堡大众银行	Volksbank_Oldenburg
奥斯纳布吕克大众银行	Volksbank_Osnabrück
大众银行帕德博恩-霍克斯特-德特莫尔德	大众银行_帕德博恩-霍克斯特-德特莫尔德
大众银行 Peine	Volksbank_Peine
大众银行 Pinneberg-Elmshorn	Volksbank_Pinneberg-Elmshorn

名称	ID
大众银行 Raiffeisenbank Dachau	Volksbank_Raiffeisenbank_Dachau
大众银行 Raiffeisenbank Itzehoe	Volksbank_Raiffeisenbank_Itzehoe
维尔茨堡大众银行 Raiffeisenbank	Volksbank_Raiffeisenbank_维尔茨堡
Volksbank Rems	Volksbank_Rems
大众银行雷姆沙伊德-索林根	大众银行_雷姆沙伊德-索林根
罗伊特林根大众银行	大众银行_罗伊特林根
大众银行 Rhein-Lippe	大众银行_Rhein-Lippe
大众银行 uck Rhein-Nahe-Hunsr	大众银行_Rhein-Nahe-Hunsrück
莱茵-鲁尔大众银行	大众银行_莱茵-鲁尔
莱茵-韦拉大众银行	Volksbank_Rhein-Wehra
莱茵-伍珀大众银行	Volksbank_Rhein-Wupper
大众银行 RheinAhrEifel	大众银行_ RheinAhrEifel
里特伯格大众银行	Volksbank_Rietberg
大众银行 Ruhr Mitte	Volksbank_Ruhr_Mitte
萨尔普法尔茨大众银行	Volksbank_Saarpfalz
萨尔茨堡大众银行	大众银行_萨尔茨堡
大众银行 Sauerland	Volksbank_Sauerland
大众银行 Schwarzwald-Neckar	Volksbank_Schwarzwald-Neckar
大众银行 Siegerland	Volksbank_Siegerland
大众银行 Solling	Volksbank_Solling

名称	ID
大众银行 Steiermark	Volksbank_Steiermark
大众银行 Stormarn	大众银行_Stormarn
斯图加特大众银行	大众银行_斯图加特
苏林根大众银行	大众银行_苏林根
南埃姆斯兰大众银行	大众银行_Süd-Emsland
南海德大众银行	Volksbank_Südheide
大众银行 Tecklenburger Land	Volksbank_Tecklenburger_Land
蒂罗尔大众银行	Volksbank_Tirol
特里尔大众银行	Volksbank_Trier
大众银行 Uelzen-Salzwedel	Volksbank_Uelzen-Salzwedel
Volksbank Ulm-Biberach	Volksbank_Ulm-Biberach
Volksbank Unna	Volksbank_Unna
世界大众银行	Volksbank_Verden
Vogtland 大众银行	Volksbank_Vogtland
福拉尔贝格州大众银行	大众银行_福拉尔贝格州
大众银行 Warburger Land	Volksbank_Warburger_Land
大众银行 Wenden-Drolshagen	Volksbank_Wenden-Drolshagen
大众银行 Weserbergland	Volksbank_Weserbergland
大众银行 Westliche Saar plus	大众银行_Westliche_saar_Plus
维也纳大众银行	Volksbank_Wien

名称	ID
Volksbank Wittingen-Klotze	Volksbank_Wittingen-Klötze
Volksbank Wolfenbüttel-Salzgitter	Volksbank_Wolfenbuttel-Salzgittel
尼尔斯河畔大众银行	Volksbank_an_der_Niers
哈尔茨大众银行	Volksbank_im_Harz
大众银行 im Markischen Kreis	volksbank_im_markischen_kreis
绍姆堡的大众银行	VolksBank_in_in_Schaumburg
大众银行-赖夫艾森银行安贝格	Volksbank-Raiffeisenbank_Amberg
Volksbank-Raiffeisenbank Döbeln	Volksbank-Raiffeisenbank_Döbeln
大众银行 Raiffeisenbanken	Volksbanken_Raiffeisenbanken
大众汽车	大众汽车
沃兰	沃兰
Volt 信息科学	Volt_信息_科学
沃尔塔	沃尔塔
沃尔塔斯	沃尔塔斯
音量增强	音量_Plus
美国志愿者协会	美国志愿者_of_America
Vomar	Vomar
冯·莫尔	Von_Maur
Vons	Vons
Vooruit	Vooruit

名称	ID
Vopak	Vopak
Vostochniy Bazar	Vostochniy_Bazar
Vostochniy 银行	Vostochniy_Bank
VotOn是的	VotOn是的
Vovo Telo	Vovo_Telo
航程保障	Voyage_Care
Vozrozhdenie	Vozrozhdenie
Vp-Market	Vp-Market
Vseobecna Uverova Banka	Vseobecna_Uverova_Banka
Vue	Vue
Vue 电影院	Vue_电影院
Vunderkind	Vunderkind
Vuvuzela 啤酒俱乐部	Vuvuzela_Beer_Club
Vx 酒店	Vx_Hotel
Vyatka-Bank	Vyatka-Bank
维卢克酒店	Vyluk_Hotel
Vysshaya Liga	Vysshaya_Liga
VELOMET'	VELOMET'
Varisilma	Varisilma
Vårdapoteket	Vårdapoteket

名称	ID
Vèrsgo	Vèrsgo
贝莱斯	贝莱斯
Vélhop	Vélhop
Vêti	Vêti
Vögele	Vögele
W	W
W Boyes & Co	W_Boys_&_Co
W 系列	W_系列
W 酒店	W_HOTELS
W 咖啡厅	W_Café
W for Woman	w_for_Woman
W.B. Samson	W._B._Samson
W. KRUK	W._KRUK
W. Lane	W._Lane
W.Steak	W.Steak
华夫饼艺术	WAFFLE_ART
沃尔特	沃尔特
华尔兹	华尔兹
万达广场	万达广场
万达王国	万达王国

名称	ID
万和春	万和春
想喝一杯	想喝一杯
WARAPORN SALAPAO	WARPORN_SALAPAO
华纳酒店	华纳酒店
华纳村	华纳村
WASBRUK	WASBRUK
WASGAU	WASGAU
洗涤盒	WASH_BOX
洗	WASH_
WATAMI	WATAMI
WATEI	WATEI
屈臣氏	屈臣氏
韦恩的咖啡	WAYNE'S COFFEE
WCU	WCU
我们	我们
WEGO	WEGO
WELLCARE 健康护理用品	Wellcare_Health_care_
WELTON	WELTON
WEN TANG 吴江	WEN_TANG_WU_JIANG
WENICE	WENICE

名称	ID
WESCO 发行版	wesco_Distrib
西部联盟	西部联盟
西联汇款	WESTERN_UNION
威斯汀	威斯汀
西边	西边
潮湿的市场	潮湿市场
WEY	WEY
WG Grinders	WG_Grinders
口哨	口哨
白色小巷	白色胡同
白天鹅酒店	白天鹅酒店
WHM 水煮饺子馆	whm_boiled_Dumplings_Hall
WHSmith	WHSmith
为什么还有 1/2	WHY_AND_1/2
WIC国家机构	WIC_State_agency
WIKY hracky	wiky_Hracky
威廉姆斯-索诺玛	威廉姆斯-索诺玛
WILLSOOR	WILLSOOR
WILLYS	WILLYS
WIN 房屋检查	win_Home_insect

名称	ID
WIN-SENT 中心服务	Winsent_cent_服务
WINDSMOOR	WINDSMOOR
永隆银行	WING_LUNG_BANK
边锋	边锋
WINXO	WINXO
WIPARK	WIPARK
巫师	巫师
巫师	巫师
WMF	WMF
WO JA NI CHR	WO_JA_NI_CHR
WOG	WOG
炒锅	炒锅
大吃一惊	WOK_UP
WOM	WOM
美好生活餐厅	精彩生活餐厅
WONGPANIT	WONGPANIT
哈哈哈哈哈	哈哈哈哈哈
木制面包店	木制面包店
伍尔沃斯食物	Woolworths_
WOOQOD	WOOQOD

名称	ID
工作室	工作室
体育世界	体育世界
世界酒店	世界酒店
哇 Cafe & Wingery	WOW_cafe_&_wingery
WOW 颜色	WOW_color
把它包起来 café	wrap_it_up_Caffé
WS Bakers	ws_bakers
WSECU	WSECU
WSFS 银行	wsfs_bank
WSI	WSI
WSK 银行	wsk_bank
WSP	WSP
WTG Fuels	WTG_Fuels
五粮液	五粮液
五明火锅	WUMING_HOT_POT
WYNSORS 的鞋子世界	WYNSORS_WORLD_OF SHOES
Wa Lai Da	Wa_Lai_Da
WaBa 烧烤	WaBa_Grill
WaHot	WaHot
Wachovia	Wachovia

名称	ID
韦科	韦科
韦科产权公司	Waco_Title_Company
华歌尔	华歌尔
Waddell & Reed	Waddell_&_Reed
Wafa 保证	Wafa_保险
Wafa Immobilier	Wafa_immobilier
WafaCash	WafaCash
Wafasalaf	Wafasalaf
Waffle Bant	Waffle_Bant
华夫饼屋	华夫饼屋
华夫洛	华夫洛
Wagas	Wagas
瓦格纳	瓦格纳
Wagner Savoy	Wagner_Savoy
哇智华	Wah_ji_Wah
Wahaca	Wahaca
娃哈哈酒店	Wahaha_Hotel
娃哈哈茶	Wahaha_Tea
Wahoo's Fish Taco	wahoo's_fish_taco
位元堂	Wai_Yuen_Tong

名称	ID
怀托摩	怀托摩
怀特罗斯	怀特罗斯
Wakaberry	Wakaberry
醒来早午餐	醒来_早午餐
稚内信用金库	稚内_信用金库
Wako	Wako
Waku	Waku
Waldbaum's	Waldbaum's
Waldecker Bank	Waldecker_Bank
Waldo Sensini	Waldo_Sensini
Waldo's	Waldo's
华尔道夫酒店及度假村	Waldorf_Astoria_酒店及度假村
沃尔格林	沃尔格林
沃尔和合伙人	公司
WallStreet	WallStreet
华莱士毕晓普	华莱士_毕晓普
Walliser Kantonalbank	Walliser_KantonalBank
沃尔玛（Walmart）	沃尔玛（Walmart）
沃尔玛燃料	Walmart_Fuel
沃尔玛邻里市场	沃尔玛邻里市场

名称	ID
Walmsley's	Walmsley's
Walon	Walon
沃尔特迪斯尼世界度假区	沃尔特_迪士尼_世界_度假区
沃尔顿斯	沃尔顿斯
Wam	Wam
万瑞	Wan_Sui
WanDa 电影院	WanDa_电影院
瓦奈爱情酒店	Wanai_Love_Hotel
Wanama	Wanama
金万达	Wanda_Jin
万达时刻	万达时刻
万达瑞恩	WandaReign
万达主题公园	万达主题公园
WandaVista	WandaVista
万德林	万德林
Wandro	Wandro
Wang	Wang
王飞佳	王飞佳
Wang 牛排	王_牛排
王兴吉	王兴吉

名称	ID
王琪	王琪
王府井	王府井
王氏厨房	王氏厨房
王顺阁	王顺阁
万佳 Culiangwang	万佳_Culiangwang
万家汇	万家汇
万家乐	万家乐
万联证券	万联证券
万龙洲海仙	万龙舟_海县
Wann Yueh	Wann_Yueh
万宝茶店	Wanpo_Tea_Shop
万载农村商业银行	湾仔_农村_商业银行
Warba Bank	Warba_Bank
Warda	Warda
Wareco	Wareco
数据仓库	数据仓库
一号仓库	仓库_一
仓库文具	仓库_文具
Warid	Warid
Waroeng 牛排和奶昔	Waroeng_Steak_&_Shake

名称	ID
Wartburg-Sparkasse	Wartburg-Sparkasse
Warung Leko	Warung_Leko
Warung Tekko	Warung_Tekko
华威国际酒店	华威国际酒店
瓦萨奇峰 FCU	wasatch_peaks_fcu
清洗并折叠	Wash_&_Fold
Wash N'Run	Wash_N'_Run
Wash'n Dry	Wash'N_Dry
WashEasy	WashEasy
华盛顿联邦储蓄银行	华盛顿联邦储蓄银行
华盛顿信托银行	华盛顿信托银行
Washintown	Washintown
沃什玛特	沃什玛特
和食 Resutoran Tengu	Washoku_Resutoran_Tengu
瓦西姆酒店	Wassim_Hotel
Wassinn	Wassinn
废物管理	废物管理
Wasu	Wasu
Watanya	Watanya
W atashi-no-heya	W atashi-no-heya

名称	ID
手表作品	手表作品
Wateen	Wateen
水滴茶馆	Water_Drop_Teahouse
水上天堂	水上天堂
Water To Go	Water_To_Go
WaterField	WaterField
水域	水域
沃特兰证券	沃特兰证券
水石	水石
屈臣氏专卖店	屈臣氏专卖店
瓦特·曼恩	Watt_Mann
波次	波次
Wave FCU	Wave_fcu
Wawa	Wawa
Wawa Cha	Wawa_Cha
wawajiao ganguonianda	wawajiao_ganguoniandai
Wawee 咖啡	哇哇咖啡
Wayback 汉堡	Wayback_Burgers
Wayne-Dalton Genie 销售中心	Wayne-Dalton_Genie_Genie_销售中心
Wcf	Wcf

名称	ID
我们佛罗里达财经	我们_佛罗里达_财经
我们是人民	我们_The_People
我们是 Rolling 椒盐脆饼公司	我们在 rolling_pretzel_company
WeWork	WeWork
韦伯能源燃料	韦伯_能源_燃料
韦伯斯	韦伯斯
Webhallen	Webhallen
韦伯斯特银行	韦伯斯特银行
Wedge 药房	楔形药房
Wedome	Wedome
杂草人	Weed_Man
工作日	工作日
每周	每周
Wegagen Bank	Wegagen_Bank
Wegmans	Wegmans
韦格曼药房	韦格曼斯_药房
Wegymer	Wegymer
魏精华酒店	Wei_Essence_Hotel
Wei Zhi Jue	Wei_zhi_Jue
潍柴汽车	潍柴_汽车

名称	ID
Weichert 房地产经纪人	Weichert_Realtors
味道江湖火锅	微道_江湖_火锅
Weier	Weier
潍坊农村商业银行	潍坊_农村_商业银行
Weigel's	Weigel's
体重观察者	体重_Watchers
威海市商业银行	威海城市商业银行
Weil	Weil
魏斯市场	Weis_Markets
Weiselect	Weiselect
Weishuwu	Weishuwu
魏子富	魏子富
欢迎使用	欢迎使用
欢迎光临 FCU	Welcome_fcu
欢迎玛特	欢迎_Mart
欢迎瑞士酒店	欢迎光临_瑞士酒店
Weldrick	Weldrick
好吧	好吧
WellPoint-Blue Shield Cross/Blue	WellPoint-蓝十字/蓝盾
WellStar Health Sy	WellStar_健康_系统

名称	ID
韦尔布里奇	韦尔布里奇
惠康	惠康
威尔迪斯健身	运动健身
Welle Dental	Wel_Dental
韦尔斯利旅馆	Wellesley_Inn
永远健康医疗保险	健康_Forever_Medicare
韦尔斯	韦尔斯
富国银行	Wells_Fargo
Weltbild	Weltbild
Weltmeister	Weltmeister
Wema Bank	Wema_Bank
Wempe	Wempe
Wendy's	Wendy's
温迪的牛奶棒	Wendy's milk_Bar
文峰	文峰
文峰殿奇	Wenfeng_dianqi
文峰大世界	文峰_伟大_世界
文峰广场	文峰_广昌
Wenhoyo 太棒了	wenhoyo_Super
文雄	文雄

名称	ID
温岭农村商业银行	温岭农村商业银行
温氏生鲜连锁店	Wenshi Fresh_Chain
文兴连锁酒店	文兴_连锁酒店
Wenzel's	Wenzel's
Wereldwinkel	Wereldwinkel
Werkmarkt	Werkmarkt
WesBanco	WesBanco
Wesbank	Wesbank
Wesco	Wesco
微视教育	wesee_教育
Weser-Elbe Sparkasse	Weser-Elbe_Sparkasse
Wessels + Müller AG	wessels_+_müller_ag
Wessels+Müller	Wessels+Müller
西孟加拉邦合作银行	西孟加拉邦州立合作银行
西海岸视频	西海岸_视频
西部社区 CU	West_Community_cu
西康沃尔 Pasty Co.	West_Cornwall_Pasty_Co.
West Elm	West_Elm
西部海军	West_Marine
西部卫生	West_Wilant

名称	ID
西弗吉尼亚州中部 FCU	西弗吉尼亚州_中部_FCU
西区超市	西区_超市
West-Aircomm FCU	West-aircomm_FCU
WestLB	WestLB
Westaff	Westaff
西美银行	西美银行
Westar 石油公司	Westar_Oil_Company
西方和南方人寿保险	西方人寿和南方人寿保险
西部汽车	Western_Auto
西方银行	西部银行
西式牛肉	西式_牛肉
西方牙科	Western_Dental
西方诊断病理学	西方_诊断_病理
西方家庭	Western_Family
西方遗产 CU	Western_Heritage_cu
西落基山脉 FCU	Western_Rockies_FCU
西部证券	西部证券
西式炙手可热牛排及更多	Western_Sizzlin_Steak_&_More
Western Vista	Western_Vista_FCU
Westerra CU	Westerra_cu

名称	ID
韦斯特瓦尔德银行	Westerwald_Bank
威斯特法伦	威斯特法伦
Westfalika	Westfalika
韦斯特菲尔德	韦斯特菲尔德
韦斯特兰	韦斯特兰
西太平洋银行	西太平洋银行
西边	西边
西区石油公司	Westside_Petro
湿封	Wet_Seal
Wetherlys	Wetherlys
韦瑟斯彭酒店	Wetherspoon_酒店
Wetzel's Pretzels	Wetzel's_pretzels
Weyerhaeuser	Weyerhaeuser
Weylandt's	Weylandt's
Whataburger	Whataburger
小麦第一屠夫歌手	Wheat_First_Butcher_Singer
Wheatland FCU	Wheatland_FCU
惠顿全球搬家	惠顿_全球_移动
摩天轮乐趣租赁	Wheel_Fun 租赁
车轮工厂	车轮_Works

名称	ID
Wheelhous	驾驶室_cu
哪个 Wich	哪个_Which
惠德比咖啡	惠德比_咖啡
涡流	涡流
Whit's Frozen 卡士达	Whit's_frozen_Custard
Whitcoulls	Whitcoulls
白色和米色	白色_&_米色
White Barn Candle	White_Barn_Candle_C
白色城堡	白色城堡
白饺子	White_gyoza
白母鸡储藏室	白鸡_储藏室
白宫-黑市	白宫黑市
白点	白点
白色的东西	白色_Stuff
白木屋	White_Wood_Hou
Whiting	Whiting
惠特尼银行	惠特尼银行
切尔西的惠塔德	Whittard_of_Chelsea
惠茨烧烤	Whitts_Babb
惠特沃斯化学家	惠特沃斯_化学家

名称	ID
谁是茶	Who's_tea
全食	全食品
批发俱乐部	批发俱乐部
Whwh-We Hotel We Home	Whwh-We_Hotel_We_Home
Wi-Tribe	Wi-Tribe
Wi LLDi	Wi LLDi
WiZink	WiZink
Wibra	Wibra
威奇托联邦信用合作社	威奇托_联邦信用合作社
Wickes	Wickes
Wicks'n'Stic	Wicks'n'Stic
Wielkopolska SKOK	wielkopolska_skok
Wiener Feinbackerei Heberer	Wiener_Feinbackerei_Heberer
Wienerschnitzel	Wienerschnitzel
维纳瓦尔德	维纳瓦尔德
Wienerwald-Tourast	Wienerwald-Tourast
威斯巴登大众银行	Wiesbadener_Volksbank
维森霍夫	维森霍夫
Wilco	Wilco
Wilco Farm	Wilco_Farm

名称	ID
野豆咖啡厅	Wild_bean_Cafe
野鸟中心	Wild_Bird_Centers
野鸟无限	野鸟_无限
通配符和礼物	Wild_Cards_&_Gifts
狂野配对	百搭配对
狂野的南方	狂野南方
狂野之翼	Wild_Wing
野翼咖啡厅	野翼咖啡厅
野莓	野莓
野兽争霸	野兽争霸
威尔芬格	威尔芬格
威尔金森	威尔金森
Wilko	Wilko
Will's	Will's
威尔的健身	Will's_Fitn
威拉德电池	威拉德_电池
威廉·约翰的披萨	威廉·约翰的披萨
威廉·佩恩	威廉·佩恩
威廉姆斯	威廉姆斯
威廉姆斯咖啡酒吧	威廉姆斯咖啡酒吧

名称	ID
威廉姆斯新鲜咖啡厅	Williams_Fresh_
威廉姆斯 P Chemists	Williams_P_Chemists
Willisch	Willisch
Wills 生活方式	Wills_生活方式
威尔斯塔夫全球	Willstaff_W
Willys Hemma	Willys_Hemma
威尔明顿信托基金	Wilmingoton_Trust
威尔希尔州立银行	威尔希尔州立银行
威尔逊通讯有限公司	Wilson_Communications_
威尔逊农场	Wilson_Farms
威尔逊停车场	威尔逊_停车
威尔逊世界酒店	Wilson_World_Hotel
Wilsons	Wilsons
威尔逊皮革	威尔逊_皮革
温姆早午餐	温姆早午餐
Wimpy	Wimpy
WinCo 食物	WinCo_食物
WinEco	WinEco
WinMart+	WinMart+
WinSouth CU	WinSouth_CU

名称	ID
WinTrust 银行	WinTrust_银行
Winchell's	Winchell's
Wind	Wind
橱窗教育	Window_Education
窗户帮派	window_Gang
窗户精灵	Window_Genie
温莎	温莎
温莎胶合板	温莎_胶合板
温莎史密斯	Windsor_Smith
Windstar	Windstar
Windstream	Windstream
葡萄酒连接	葡萄酒连接
酒庄	酒庄
永亨银行	Wing_Hang_Bank
Wing Nutz	Wing_Nutz
Wing On	Wing_On
翼租一辆车	Wing_Rent_A_Car
荣华	Wing_Wah
翼区	Wing_Zone
Wing 的军队	Wing's_army

名称	ID
温德姆集团温盖特酒店	温德姆温盖特酒店
翅膀	翅膀
翅膀等	翅膀_等
翅膀待发	Wings_To_Go
挡翼	挡翼
温豪斯	温豪斯
眨眼	眨眼
Winn-Dixie	Winn-Dixie
Winn-Dixie 酒类商店	Winn-Dixie_Liquor_Store
赢家的	赢家的
获胜者	获胜者
Winsa	Winsa
赢泰 Autoglass	Wintec_Autoglas
冬天的	冬天的
无线升级版	无线_Plus
无线玩具	Wireless_Toyz
无线区域	Wireless_Zone
Wise Up	Wise_Up
许愿	许愿
Wispol 德罗杰里亚	Wispol_Drogeria

名称	ID
Wissol	Wissol
巫术	巫术
Witchery Kids	Witchery_Kids
Witchery Man	Witchery_Man
Witt-Weiden	Witt-Weiden
Witteveen	Witteveen
Wittner	Wittner
Wo Ai Wo Jia	Wo_Ai_Wo_Jia
wochongwoai	wochongwoai
Wockhardt	Wockhardt
沃尔曼·布罗克	Woermann_Brock
Wojcik	Wojcik
Wojooh	Wojooh
炒锅盒	Wok_Box
盒装炒锅	wok_in_a_box
Wok to Wal	wok_to_walk
沃尔赞卡	沃尔赞卡
Wolf 相机和图像	Wolf_Camera_&_Image
沃尔夫	沃尔夫
沃尔夫电影院	Wolff_Cinema

名称	ID
沃尔夫冈·帕克快车	沃尔夫冈_Puck_Express
沃尔福德	沃尔福德
沃尔斯多夫	沃尔斯多夫
女性健康精品店	女士健康精品店
Wonder GOO	Wonder_goo
仙境度假村	仙境_度假村
黄	黄
黄氏炒锅	Wong's_wok
wonitrol	wonitrol
哇	哇
哇！ mart	哇！ mart
WooTea	WooTea
伍德城堡	伍德城堡
木工艺	木工艺
Woodfin	Woodfin
伍德福里斯特国家银行	伍德福里斯特国民银行
Woodie's	Woodie's
林地	林地
世界伍德曼代表	世界木人代表
Woodoo	Woodoo

名称	ID
啄木鸟	啄木鸟
伍兹咖啡	伍兹_咖啡
伍德斯普林套房酒店	伍德斯普林套房酒店
伍德斯通	伍德斯通
伍迪	伍迪
Woody's	Woody's
Woody's BarBQ	woody's_barbq
Woof Gang 面包店	Woof_Gang_Bakery
WoolStreet	WoolStreet
伍尔维奇	伍尔维奇
伍尔沃斯	伍尔沃斯
伍尔沃斯酒	Woolworths_
Woolworth MetroGo	Woolworths_ MetroGo
Woolworths 的食物站	Woolworths_food_
友利银行	Woori_Bank
Woposo	Woposo
工作用地	Work_Land
工作世界	工作世界
工人 CU	workers_cu
锻炼世界	World_World

名称	ID
世界认可	世界_接受
世界级	世界一流
世界英语	世界_英语
世界金融公司	World_Finance_公司
世界金融集团	世界金融集团
世界天然气	世界天然气
世界健身房	世界健身房
世界健身房健身中心	世界_Gym_Fitness_Center
世界储蓄和贷款协会	世界储蓄与贷款协会
世界储蓄银行	世界储蓄银行
世界网球	世界_网球
啤酒世界	啤酒世界
泰坦世界	泰坦世界
WorldCall	WorldCall
WorldClass	WorldClass
Worten	Worten
Worten 移动版	Worten_Mobile
哇	哇
哇烤肉串	哇_Kebab
哇！移动精品店	哇！ _手机精品店

名称	ID
哇！Momos	哇！ _Momos
哇曲	哇曲
牧马者	牧马者
Wrap & Roll	Wrap_&_Roll
把它包起来	Wrap_It_Up
雷恩·基奇	Wren_Kitch
Wren 厨房和浴室	Wren_厨房_和_浴室
莱特线代表	Wright_Line_代表
Wt 定制器设计	Wt_Customizer_Design
吴芳寨	吴芳寨
吴家冰红茶	Wu_jia_Ice_Black_Tea
五道香	五道香
五东风	五东风
武斗堂	武斗堂
舞钢包商农村银行	舞钢_包商_农村银行
武汉农村商业银行	武汉_农村_商业银行
芜湖扬子农村商业银行	芜湖_扬子_农村_商业银行
wika Times	wuika_Times
吴江农村商业银行	吴江_农村_商业银行
五里关火锅	五里关_火锅

名称	ID
Wumart	Wumart
内衣裤	内衣裤
Wurst-Basar	Wurst-Basar
武尚	武尚
武胜羊肉汤	Wusheng_Mutton_Soup
无锡农商银行	无锡_农村_商业银行
五原	五原
五月	五月
五洲国际	五洲_国际
WyHy FCU	WyHy_FCU
Wycon	Wycon
WyjatkowyPrezent.pl	WyjatkowyPrezent.pl
温德姆	温德姆
Wz 药品超市	wz_Drugs_super
伍尔特	伍尔特
Wüstenrot	Wüstenrot
X Oil	X_Oil
X 国家	x_nation
X-ACT	X-ACT
X-KOM.PL	X-KOM.PL

名称	ID
极限商店	X-TREME_STORES
XBLC	XBLC
XCEL FCU	XCEL_FCU
西安酒店	西安_酒店
西安餐厅	西安_餐厅
厦门国际银行	厦门国际银行
湘财证券	香财证券
小豆面	小豆_面条
小桥寿司	小巧_寿司
西贝餐厅	西贝_餐厅
西单百货	西单百货
协恒	协恒
XIMI Vogue	ximi_Vogue
XIN BAI LU	XIN_BAI_LU
信达信百货	XIN_DA_XIN_百货大楼
辛沐阳	Xin_muyang
XINRONGJI	XINRONGJI
XJ 海鲜	xj_Seafood
XL	XL
XL 中心	XL_Center

名称	ID
XL 超大号	XL_EXTRA_LARGE
XL 零件	XL_Parts
XL STOP & SHOP	XL_STOP_&_SHOP
xl-bygg	xl-bygg
XPO 物流	世博会物流
快速停止	XPRESS_STOP
XUSS	XUSS
XXL	XXL
XXXLutz	XXXLutz
Xana 酒店	Xana_Hotelle
Xana 精简版	Xana_Lite
Xanaka	Xanaka
徐工集团	徐工集团
Xenos	Xenos
Xes 工作室	X_Studio
Xfinity	Xfinity
Xhyee 酒店	Xhye_Hotels
西和雅园	xi_he_ya_ya_yuan
西山贡院	Xi_shan_Gong_Yuan
XiLong 酒店	XiLong_酒店

名称	ID
夏丽	Xia_Li
Xiachixiashuan	Xiachixiashuan
下关茶	夏冠_茶
厦门银行	厦门_银行
厦门农商银行	厦门_农村_商业银行
咸丰水果	鲜丰_水果
向鄂青	向艾青
向派芳	香排芳
向村微道	xiang_cun_wei_dao
香草云南原味火锅	香草_云南_原味火锅
湘江新区农村商业银行	湘江_新区_农村通信_银行
向杰芳	向杰芳
响水农村商业银行	响水_农村_商业银行
向天峡	向天峡
湘乡农村商业银行	湘乡_农村_商业银行
向香乐	向香乐
祥源农商银行	祥源_农村_商业银行
Xianhezuangluweihuogu	Xianhezuangluweihuogu
仙居农村商业银行	仙居_农村_商业银行
仙来仙池	仙来仙池

名称	ID
小彩园	Xiao_Cai_Yu
小胡鸭	小虎鸭
小刘汤饺	Xiao_Liu_soup_Dupling
谢晓	Xiao_Xie
小张烤鱼	Xiao_Zhang_Grilled_Fish
小雕理塘	Xiaodiao_Litang
牛晓芳	牛晓芳
小关茶	小关_茶
小亨水饺	小亨_水饺
小辉阁火锅	Xiaohuige _火锅
小娇天	小娇天
小龙坎	小龙坎
小梦华	小梦华
小米公司	小米公司
小米米	小米米
Xiaomizhijia	Xiaomizhijia
小鹏汽车	小鹏_汽车
萧山农商银行	萧山_农村_商业银行
晓松图书馆	小松_图书馆
晓武	晓武

名称	ID
小香生鲜	小香_Fresh
晓阳	晓阳
Xicuizl	Xicuizl
西定	西定
谢佳伟道	谢佳伟道
协都汇蟹屋	Xieduhui_Crab_House
谢江军	谢江军
谢王府	谢王府
谢协农	谢协农
协宇达茶	协御达_茶
西峰包商乾隆村镇银行	西丰_包商_乾隆_农村银行
西湖之春	Xihu_Spring
西佳德夏仁水娇	喜佳德_夏人_水娇
新车汽车服务	Xin_Che_Auto_Service
新大兴爱家超市	信达兴爱家超市
新南华餐厅	新南华餐厅
欣旺咖啡厅	Xin_Wang_Cafe
XinFeng XiaoChi	XinFeng_XiaoChi
XinHua 书店	XinHua_书店
XinXiangYue	XinXiangYue

名称	ID
新安	新安
新宝成海鲜	新宝城_海鲜_食品
新昌农商银行	新昌农村商业银行
新地广场	新地广场
新都桂城乡镇银行	信都_贵城_农村银行
幸福堂	唐兴富
兴华堂著名小吃	Xing_Hua_Tang_Famous_Snaceks
星波连锁酒店	星波_连锁酒店
杏花楼	杏花楼
兴隆	兴隆
兴隆国际	兴隆_国际
兴美乐	兴美乐
星沙农商银行	星沙_农村_商业银行
星星国际影城	Xingx_国际_影院
兴业银行	兴业银行
新海富	新海富
新华百货	新华百货大楼
新会农商银行	新会_农村_商业银行
西宁农村商业银行	西宁农村商业银行
新疆银行	新疆银行

名称	ID
新疆汇和银行	新疆_汇和银行
辛集市	辛集市
Xinladao	Xinladao
欣蕾	欣蕾
新民农村商业银行	新民_农村_商业银行
新桥口腔科	Xinqiao_口腔科
新胜波	新胜波
信德汇	信德汇
新天地超市	新天地_超市
新旺	新旺
新向辉	新向辉
新兴药房	新兴_药店
欣怡华	欣怡华
新余农村商业银行	新余农村商业银行
信源国际	欣源_国际
欣源精智	欣源精智
新宇楼	新宇楼
辛兆县	辛兆县
雄丰超市	雄丰超市
西元堂	西元堂

名称	ID
喜悦国际影城	喜悦_国际_影城
期待	期待
Xpeng	Xpeng
探索 FCU	xplore_fcu
Xpress 燃料	快递_燃料
Xpress Lube	Xpress_Lube
随心所欲的钱	Xpress_Mone
随心所欲超级	极速快递
Xspark 教育	Xspark_教育
Xtra Mart	Xtra_Mart
极限	极限
宣城万南农村商业银行	宣城_万南_农村通信_银行
学达教育	学达_教育
雪儿思	雪儿思
Xueshisanba	Xueshisanba
旬阳农村商业银行	旬阳农村商业银行
徐州农村商业银行	徐州_农村_商业银行
徐州铜山农村商业银行	徐州_铜山_农村通信_银行
Xynos	Xynos
你是旅行吗？	y_si_viajas？

名称	ID
是！ 移动设备	是！ 移动设备
Y's Road	Y's_road
Y-伦敦	Y-伦敦
Y. YAMADA	Y._YAMADA
YA KUN KAYA 吐司	YA_KUN_KAYA_TOAST
YALI SPOR	YALI_SPOR
山崎	山崎
YAN YU CHA	YAN_YU_CHA
杨芳胜利羊肉火锅	YANGFANG_SENGLI_MUTTON_HOTPOT
YANNICK	YANNICK
燕莎友谊购物城	YANSHA_YOUYI_SHOPPING_CITY
YAODE	YAODE
八佰伴	八佰伴
YARGICI	YARGICI
矢田	矢田
YATEEM OPTICIAN	YATEEM_OPTICIAN
YATOUT	YATOUT
YATSUDOKI	YATSUDOKI
YAVA	YAVA
YBS	YBS

名称	ID
是的	是的
YELMO	YELMO
YEMA AUTO	YEMA_AUTO
YENTAFO KRUENG SONGE	YENTAFO_KRUENG_SONGE
是的	是的
是的 EKMEK	YEPAS_EKMEK
是	是
是的链	是的链
YESFEL 酒店	YESFEL_HOTEL
YH.TANG	YH.TANG
YHA	YHA
一芳台湾水果茶	YIFANG_TAIWAN_FRUIT_TEA
YIPPIN	YIPPIN
YISHION	YISHION
YITEL	YITEL
宜兴顺	宜兴顺
YK 旅馆	YK_inns
YKK AP 陈列室	ykk_ap_Showroom
YKKO	YKKO
YKM	YKM

名称	ID
基督教青年会	基督教青年会
YODY	YODY
YOGER PRES	OGER_PRESSO
横滨	横滨
尤兰达	尤兰达
YON HO	YON_HO
YONCA	YONCA
永和之王	YONGHE_KING
永辉超市	永辉_超市
我是 YOSHI	YOSHI_ME_ME
YOTA	YOTA
YOTELQQ	YOTELQQ
优佳四川菜	YOUJIA_SCHUAN_DISH
年轻人眼镜	年轻人_眼镜
YOUGOR	YOUGOR
你的脸	你的脸
YRC	YRC
YSL	YSL
是的	是的
工作室	工作室

名称	ID
YTS	YTS
YU KEE	YU_KEE
余香啦宝宝	YU_XIANG_LA_PO_PO
元	元
元嘉经典奶茶	YUAN_CHIA_CLASSICAL_MILK_TEA
元正	元正
YUDO	YUDO
云波大酒店	云波酒店
云龙之星	云龙之星
YURTBAY SERAMIK	YURTBAY_SERAMIK
御心四川菜	御心_四川_菜
玉渊潭酒店	玉渊潭_酒店
YVONNE 系列	YVONNE_COLLECT
基督教女青年会	基督教女青年会
YZB	YZB
雅安市商业银行	雅安城市商业银行
Yadea	Yadea
亚朵S酒店	Yaduo_S_Hotel
Yagmour	Yagmour
Yagmur Mobilya	Yagmur_Mobilya

名称	ID
Yakimix	Yakimix
烤肉五园	Shakku_Goen
烤肉烤肉烧烤	烤肉_Griglia
Yakink no Gyuta	Shakku_No_Gyuta
烤肉末广馆	烤肉_Suehiro_Kan
烤肉之和民	Shakku_no_watami
烧肉屋酒井	烤肉屋_Sakai
炒面工厂	炒面_工厂
烤鸡肉串日高	烤鸡肉串_日高
Yakitoriya	Yakitoriya
Yakiyan	Yakiyan
Yama 铁板烧餐厅	Yama_铁板烧_餐厅
山小屋	山小屋
山小屋拉面	山小屋_拉面
Yamaha	Yamaha
山形屋店	山形屋店
亚马尔	亚马尔
YamaMay	YamaMay
山中屋	山中屋
山梨县民信用合作社	山梨县_公民信用_合作社

名称	ID
山梨信用金库	山梨县_信用金库
山冈屋	山冈屋
大和信用金库	大和信用金库
雅玛多运输	雅玛多运输
Yame	Yame
亚美酒店	亚美_酒店
严平	严平
延边农村商业银行	延边_农村_商业银行
盐城包商村镇银行	盐城_包商_村镇银行
燕当	燕当
郭阳	郭阳
杨国富	杨国富
杨继龙富	杨继龙富
杨氏炸饺子	Yang's Fried-Dumplin
杨记牛肉面	杨记_牛肉_面条
阳泉市商业银行	阳泉城市商业银行
阳香豆腐皮冲洗牛肚皮	阳香_豆腐皮_冲洗_牛肚_冲洗_牛肚皮
阳阳商业	阳阳_商业
杨宇兴	杨宇兴
扬州农村商业银行	扬州_农村_商业银行

名称	ID
Yanjiyou	Yanjiyou
洋基蜡烛	Yankeee_Can
延年酒店	延年_酒店
矢野梅甘娜	Yano_Megane
烟台银行	烟台银行
烟台青年外国语学校	烟台青年外语学校
瑶瑶 suancaiyu	yaoyao_suancaiyu
Yapi Credi Bankasi	Yapi_Kredi_Bankasi
Yaposha	Yaposha
Yarche	Yarche
院子里的房子	Yard_House
码数	码数
Yarobltour	Yarobltour
雅罗斯拉夫斯基·布罗伊勒	Yaroslavskiy_broyler
Yart Inn	Yart_Inn
亚斯岛	亚斯岛
Yashma Zoloto	Yashma_Zoloto
雅斯特国际酒店	Yaste_国际_酒店
Yatas	Yatas
Yatra	Yatra

名称	ID
Yatran	Yatran
Yayan	Yayan
弥生	弥生
Ybbtb	Ybbtb
Yebeng	Yebeng
惠比寿酒吧	Yebisu_Bar
Yebrf	Yebrf
Yee Ping	Yee_Ping_Go
Yeh Wah Intl Edu School	Yeh_Wah_Intl_Edu_School
Yeinot Bitan	Yeinot_Bitan
Yekaitai	Yekaitai
黄色	黄色
黄色出租车	黄色_出租车
黄色货运系统公司	黄色_货运_System_Inc
黄河农村商业银行	黄河农村商业银行
Yellowhire 租赁护理机构	Yellowhire_Rental_Care_Agency
耶洛科纳	耶洛科纳
Yenny	Yenny
Yershari	Yershari
是的银行	是的_银行

名称	ID
是的化妆品	是的_化妆品
可以！Vi Leker	是的！ _Vi_Leker
是的-Tea	是的-Tea
耶西尔·昆杜拉	yesil_kundura
耶斯特荣誉酒店	Yeste_Honor_Hotel
耶斯特酒店	Yeste_Hotel
Yestore	Yestore
是啊	是啊
Yetel 高级版	Yetel_Premiu
Yevrobank	Yevrobank
Yevroprombank	Yevroprombank
Yi Fan 寿司	Yi_fan_Sushi
易记香雅居	Yi_ji_Xiang_Ya_ju
Yi Mai	Yi_Mai
易品黄牛	Yi_Pin_Huang_Niu
易香朱春	朱义祥春
宜宾城市商业银行	宜宾城市商业银行
伊川农村商业银行	宜川_农村_商业银行
益丰药业	怡丰_药店
宜华百货	怡华_百货_门店

名称	ID
宜卡租车	怡卡_租车
怡康	怡康
Yimayila	Yimayila
Ying Ua 地面停车场	Ying_Ua_Ground_Park
英达证券	英达证券
英达泰和人寿保险	英达_太和_人寿保险
营口沿海银行	营口沿海银行
营口荣盛农村商业银行	营口_荣盛_农村通信_银行
营口农村商业银行	营口_农村_商业银行
颍泉农村商业银行	颍泉_农村_商业银行
尹双杰	尹双杰
银泰证券	银泰证券
银溪	银溪
银香小关	尹香_小关
鄞州银行	银州银行
和颐至阁酒店	Yitel_zhi_Ge
益田	益田
义乌农村商业银行	义乌_农村_商业银行
哟！寿司	哟！ _寿司
Yocar	Yocar

名称	ID
Yoer	Yoer
Yogen Fruz	Yogen_Fruz
瑜伽之家	Yogi_House
酸奶企鹅	酸奶_企鹅
酸奶乐园	酸奶乐园
瑜伽鸡	Yogya_Chick
瑜伽瑜伽	瑜伽瑜伽
你！Roll's & Temaki	你！ _roll's_&_temaki
Yoim 人参咖啡	Yoim_ginseng_Coffee
横滨信用金库	横滨_信用金库
横滨屋	横滨屋
横纲	横纲
Yole	Yole
Yolki-Palki	Yolki-Palki
友马银行	Yoma_Bank
Yomart	Yomart
优美连锁酒店	Yome_Chain_Hotel
洋面屋五右卫门	yomen-ya_goemon
米八町	米八町
永和斗江	Yong_he_dou_jiang

名称	ID
Yong He Dou 姜大王大	Yong_He_Dou_jiang_da_wang
程永胜	程永胜
永吉	永吉
Yongki Komaladi	Yongki_Komaladi
约克便利店	约克便利店
约克酒店	约克酒店
约克郡银行	约克郡银行
约克郡建筑协会	约克郡建筑协会
吉野家	吉野家
Yoshu Shonin	Yoshu_Shonin
大冢四也	Yotsuya_Otsuka
您	您
友豪	友豪
Youjia suancaiyu	Youjia_suancaiyu
Young 迷你酒店	Young_mini_Hot
年轻的伦勃朗	Young_Rembrandts
杨氏的	杨氏的
扬斯啤酒厂	扬斯啤酒厂
年轻人	年轻人
你的机会	你的机会

名称	ID
您的独立杂货商	你的独立杂货商
你的馅饼	Your_Pie
青年餐厅	青年餐厅
青年阳光酒店	Youth_Sunshine_Hotel
Youus 牙科	Youus_牙科
友轩酒店	优轩_酒店
Ypf	Ypf
Ysatis	Ysatis
余可香	Yu_ke_Xiang
元世林周品	袁世林周平
袁氏传川香	袁氏_川川香
Yuangi 早午餐	Yuangi_Brunch
袁继云娇	袁继云娇
源泉瑜伽	元泉_瑜伽
元苏	元苏
元大银行	元大银行
元大证券	元大证券
榆次融信乡村银行	Yuci_Rongxin_乡村银行
余蝶芳	余蝶芳
裕华中国产品	Yue_Hwa_中文_产品

名称	ID
悦江春中式料理	悦江春中国菜
岳平	岳平
跃进	跃进
乐清农村商业银行	乐清_农村_商业银行
跃兴	跃兴
北京悦友酒店	悦游_酒店
岳振晓初	岳振晓初
Yug Farma	Yug_Farma
Yug-Investb	Yug-Investb
余杭农村商业银行	余杭农村商业银行
玉环农商银行	玉环_农村_商业银行
尤卡迪村庄	Yukadi_Villages
结城信用金库	Yuki_Shinkin_Bank
育空地区	育空地区
优酷烤鱼	Yuku_Grilled_Fish
圣诞宝贝	Yule_Baby
玉林茶	玉林_茶
裕隆汽车	裕隆汽车
Yum Saap	Yum_Saap
Yum Yum Donut Shop	Yum_Yum_Donut_Shop

名称	ID
玉苗教育	Yumiao_教育
云府烧鹅仔餐厅	Yun_fu_shao_e_Zai_Restai_r
云和叶波酒店	Yun_He_Ye_Bo_Hotel
云晨意大利餐厅	Yun-Chen _意大利餐厅
云达	云达
永青地产	Yung_Ching_Realty
尤尼亚斯特鲁姆银行	Yuniastrum_Bank
Yunichel	Yunichel
云南红塔银行	云南_红塔银行
云南省农村信用社	云南农村信用合作社
尤努斯市场	尤努斯_市场
玉品轩	玉品轩
对	对
余勤尚品	余勤尚品
Yuropet	Yuropet
Yurtici Kargo	Yurtici_Kargo
雨润酒店	雨润_酒店
Yuweixiao	Yuweixiao
玉溪市商业银行	玉溪城市商业银行
宇翔厨房	宇翔_厨房

名称	ID
Yuyuto	Yuyuto
Yuzhniy Dvor	Yuzhniy_Dvor
伊夫·罗彻	Yves_Rocher
伊夫·圣洛朗	伊夫·圣洛朗
Yx	Yx
Yázigi	Yázigi
Z	Z
Z 捆绑包	Z_Bundle
Z 能量	Z_Energy
Z Gallerie	Z_Gallerie
Z PIECA RODEM	Z_PIECA_RODEM
Z-Emos	Z-Emos
扎迪格和伏尔泰	ZADIG_&_VOLTAIRE
ZAFIRO 之旅	ZAFIRO_TOUR
ZAILAND	ZAILAND
赞比特	赞比特
ZAPATERIAS 3HERMANOS	ZAPATERIAS_3HERMANOS
扎拉	扎拉
扎拉回家	ZARA_HOME
ZB 银行	ZB_bank

名称	ID
ZEB	ZEB
ZEEKR	ZEEKR
ZEEMAN	ZEEMAN
玉米醇溶酶	玉米醇溶酶
ZELO	ZELO
ZEMA	ZEMA
ZEMAN maso-uzeniny	Zeman_maso-uzeniny
禅	禅
ZENCAR	ZENCAR
天顶	天顶
ZESCO	ZESCO
ZFY	ZFY
兆义	兆义
正宇	正宇
郑州日产	郑州_日产
振华	振华
振华电器	振华电器
振华超市	振华超市
浙商证券	浙商证券
志伟关	志伟关

名称	ID
中百电子	中百电子
中百百货	中百百货
钟敏百汇	钟敏百汇
诸葛高宇	ZHUGE_KAOYU
诸暨市	诸暨市
朱莲	朱莲
竹叶青	竹叶青
ZHZHUK	ZHZHUK
锡洛尼市场	ZIELONY_MARKET
ZIKO	ZIKO
ZIPPY KIDSTORE	ZIPPY_KIDSTORE
ZIZ	ZIZ
ZMAX 酒店	ZMAX_HOTELS
ZOJIRUSHI	ZOJIRUSHI
动物园城	ZOO_CITY
动物园咖啡	ZOO_COFFEE
变焦飞行	变焦飞行
人畜共患的	人畜共患的
众泰	众泰
ZT 酒店	zt_Hotels

名称	ID
ZTBL	ZTBL
ZTL 酒店	ZTL_HOTEL
ZTP	ZTP
ZTTMH chicken chaffy dish city	zttmh_chicken_chicken_chaffy_dish_city
ZUGATTI	ZUGATTI
苏黎世	苏黎世
ZUS 咖啡	Zus_coffe
ZVEROKRUH	ZVEROKRUH
心甘情愿	心甘情愿
ZX	ZX
ZXMCO	ZXMCO
ZYH Moon	zyh_moon
Zaatar W Zeit	Zaatar_W_Zeit
Zabka	Zabka
Zachodnia SKOK	zachodnia_skok
Zachodniopomorska SKOK	zachodniopomorska_skok
扎法里	扎法里
Zagrebacka banka	萨格勒巴卡_银行
Zain	Zain
杂货店	Zakka_Mart

名称	ID
Zales Jewelers	销售_珠宝商
Zales 珠宝商奥特莱斯	销售_珠宝商_奥特莱斯
赞比亚国家商业银行	赞比亚_国民商业银行
赞布雷罗	赞布雷罗
扎梅尔斯	扎梅尔斯
Zamnet	Zamnet
Zamtel	Zamtel
Zanaco	Zanaco
枣庄银行	枣庄银行
Zaozishu	Zaozishu
Zapa	Zapa
Zapovednik	Zapovednik
Zapsikombank	Zapsikombank
Zarco	Zarco
Zarina	Zarina
Zark's Burgers	Zark's_Burgers
Zarraffa 的咖啡	Zarraffa's coffee
Zavanna	Zavanna
Zaxby's	Zaxby's
Zazhigalka	Zazhigalka

名称	ID
zdorov'e	zdorov'e
兹多罗维戈罗德	Zdoroviy_Gorod
Zdoroviy Malysh	Zdoroviy_Malysh
Zdorovye Lyudi	Zdorovye_Lyudi
Zdrava	Zdrava
兹德拉夫尼克	兹德拉夫尼克
兹德拉武什卡	兹德拉武什卡
Zdrofit	Zdrofit
斑马	斑马
Zebro's	Zebro's
Zeenat 实验室	Zeenat_实验室
杰尼亚	杰尼亚
Zehrs	Zehrs
Zein El Atet	Zein_El_Atet
Zeki	Zeki
Zeki Triko	Zeki_Triko
Zelenaya apteka	Zelenaya_apteka
泽莱尼名单	Zeleniy_list
Zellers	Zellers
泽勒斯药房	Zellers_药房

名称	ID
Zemelnyi Kapital	Zemelnyi_Capital
Zempin 280Yen 居酒屋 Nipachi	Zempin_280Yen_Isakaya_Nipachi
泽姆斯基银行	Zemsky_bank
Zen Pirlanta	Zen_Pirlanta
禅宗服装店	Zen_clothing_store
Zenden	Zenden
Zenex	Zenex
泽尼特	泽尼特
真力时银行	Zenith_Bank
Zenj 交易所	禅宗交易所
零	零
零空间	Zero_Space
Zero's Subs	zero's_subs
Zerodha	Zerodha
泽塔	泽塔
宙斯	宙斯
Zeynel Cilli	Zeynel_Cilli
张胜吉	张胜吉
张贵餐厅	张贵_餐厅
张家口市商业银行	张家口_城市_商业银行

名称	ID
张老坎	张老坎
章丘齐鲁村镇银行	章丘_齐鲁_村镇银行
张忠景中医中心	张忠静_中医_CEN
詹娜	詹娜
Zhar-Pitsta	Zhar-Pitsta
浙江稠州商业银行	浙江_稠州_商业银行
浙江金华成泰商业银行	浙江_金华_成泰通讯_银行
浙江民泰商业银行	浙江民泰商业银行
浙江泰隆商业银行	浙江_泰隆_商业银行
Zhelaem zdorov'ya	zhelaem_zdorov'ya
震宇五金	震宇五金
ZhenQuo 咖啡厅	ZhenQuo_Cafe
振德利	振德利
真定鸡	振鼎_鸡
郑义伟	郑义伟
镇江农村商业银行	镇江_农村_商业银行
真美假日酒店	真美_假日客栈
振罗	振罗
真三关	真三关
浙商保险	浙商_保险

名称	ID
智草堂	Zhi_cao_Tang
智汇学校	智汇_学校
智家便利店	智家_便利店
Zhili-Byli	Zhili-Byli
志诺	志诺
日维卡	日维卡
智尊汽车服务	智尊_汽车服务
钟敏魏石	钟敏_Wei_Shi
钟友 Health	钟_You_Health
ZhongAn 旅馆	ZhongAn_旅馆
ZhongBai 湖北	ZhongBai_湖北
ZhongBai 山东	ZhongBai_山东
中百超市	中百_超市
忠诚	忠诚
中诚证券	中成证券
中海大厦	中海大厦
中海国际中心	中海国际中心
中海广场	中海广场
中南商业	中南_商业
中山农商银行	中山_农村_商业银行

名称	ID
中山证券	中山证券
中天健身	中天健身
忠伟	忠伟
钟宣恩佳	钟宣恩佳
中音音乐	钟音_音乐
中宇电信	中宇_电信
中原银行	中原银行
中州客栈	中州客栈
中州国际大酒店	中州_国际_酒店
中州怡和大酒店	中州_怡和_酒店
Zhotels	Zhotels
周刘福	周刘富
周黑雅	周黑雅
周泉州岛	周泉州岛
周世雄	周世雄
朱叶少柔	Zhu_Ye_shao_rou
朱元存	朱元存
zhuangyuan100	zhuangyuan100
珠海华润银行	珠海_中国_资源银行
珠海农村商业银行	珠海_农村_商业银行

名称	ID
诸暨农村商业银行	诸暨农村商业银行
竹露茶	Zhulu_Tea
株洲农村商业银行	株洲农村商业银行
紫光元	Zi_Guang_Yuan
ZiZi 烤鱼	ZiZi_Grilled_Fish
Zia CU	zia_cu
齐巴特	齐巴特
Zielpunkt	Zielpunkt
Ziengs	Ziengs
齐拉	齐拉
Ziggi 的咖啡	Ziggi's coffee
自贡商业银行	自贡商业银行
紫金农商银行	紫金农村商业银行
Zila Sahkari Bank	Zila_Sahkari_Bank
辛伯格	辛伯格
Zinfu 牙科	Zinfu_Dental
Zintia	Zintia
锡安银行	犹太复国银行
锡安第一国民银行	锡安斯第一国民银行
Zip	Zip

名称	ID
Zip 药物 Yakkyoku	Zip_Drug_Yakkyoku
Zip 的免下车	Zip's_drive-in
ZiPcar	ZiPcar
芝宝	芝宝
Zippy's	Zippy's
Ziraat Bankasi	Ziraat_Bankasi
ZiraatBank	ZiraatBank
Zirci Takarék	Zirci_Takarék
齐托	齐托
Zivame	Zivame
子厦门	子厦门
Ziyafet chigköfte	ziyafet_cigköfte
子有刚	子有刚
Zizzi	Zizzi
Zizzi 意大利语	Zizzi_意大利语
Zking 财产和意外伤害保险	Zking_财产与意外伤害_保险
Zlatarna Celje	Zlatarna_Celje
Zleep 酒店	Zleep_Hotels
Zocai	Zocai
黄道十二宫	黄道十二宫

名称	ID
佐斯厨房	Zoes_厨房
佐夫	佐夫
Zolotaya Mechta	Zolotaya_Mechta
Zolotaya Rybka	Zolotaya_Rybka
Zolotaya strekoza	zolotaya_strekoza
Zolotoy Hepard	zolotoy_Hepard
Zomoz	Zomoz
Zon Ria	Zon_Ria
南区	Zona_Sul
光学区	光学区
宗	宗
宗阳农商银行	宗阳_农村_商业银行
Zooba	Zooba
Zoomcar	Zoomcar
佐拉	佐拉
Zostel	Zostel
Zoup！	Zoup！
卓娅	卓娅
中兴	中兴
中通快递	Zto Express

名称	ID
Zu	Zu
Zudio	Zudio
Zuger Kantonalbank	Zuger_KantonalBank
Zumiez	Zumiez
Zun-zu 配饰	zun-zu_accessoriOS
尊品牛排	zunpin_BeefSteak
Zuoyou 沙发	zuoyou_Sofa
Zuperia	Zuperia
祖瓦	祖瓦
Zyng 亚洲烧烤	Zyng_Asian_Grill
佐卡洛	佐卡洛
Zürcher Kantonalbank	Zürcher_KantonalBank
a la Campagne	a_la_campagne
a+ Medicina Diagnóstica	a+_medicina_diagnóstica
a. Apotheka	a._apotheka
a.testoni	a.testoni
a8 酒店	a8_hotel
aHerreros	aHerreros
Zoria	Zoria
abQ +	abq_+

名称	ID
abba 酒店	abba_hotels
美国广播公司乡村音乐	ABC_Country
abra	abra
abreu	abreu
口音	口音
会计负责人	会计负责人
宏碁公司	宏碁公司
achega	achega
active	active
广告部件	广告部件
阿德吉奥城市公寓式酒店	adagio_city_公寓式酒店
adbb autour de bebé	adbb_autour_de_bebé
阿迪达斯	阿迪达斯
阿迪纳公寓式酒店	adina_公寓酒店
Aerie	Aerie
apfidea	apfidea
agnès b.	agnès_b。
啊兰	啊兰
ahli 联合银行	ahli_united_bank
ahlibank	ahlibank

名称	ID
aimo 公园	aimo_park
AiraHom	AiraHom
akf 银行	akf_bank
活性银行	aktif_bank
al hilal bank	al_hilal_bank
al-ikhsan	al-ikhsan
aldik	aldik
aleksi 13	aleksi_13
alhomaidhi	alhomaidhi
alinma bank	alinma_bank
allsun	allsun
Alma DELIKATES	alma_delikatesy
按字母顺序	按字母顺序
alphega lekarna	alphega_lékarna
alpie	alpie
alpina	alpina
alsuper	alsuper
altice	altice
alves bandeira	alves_bandeira
alvo	alvo

名称	ID
阿玛里斯酒店	amaris_hotel
亚马逊	亚马逊
ambra	ambra
americanos	americanos
amic 能源	amic_energy
ampm	ampm
anb	anb
以及	以及
安德里亚	安德里亚
安德烈	安德烈
安德烈斯快车	Andrés_Exprés
动物	动物
aniusao	aniusao
antarte	antarte
antoil	antoil
api	api
阿波罗站	阿波罗站
撇号	撇号
apoteksgruppen	apoteksgruppen
appel's	appel's

名称	ID
艾普瑞利亚	艾普瑞利亚
阿拉	阿拉
aramex	aramex
区域内部	区域内部
面积	面积
arena	arena
argo	argo
ariant	ariant
Aris	Aris
armtek	armtek
arquia banca	arquia_banca
arredo	arredo
arsamiea	arsamiea
亚瑟·卡普兰	arthur_kaplan
arzum	arzum
asmi	asmi
阿斯特拉	阿斯特拉
阿斯图里亚诺	阿斯图里亚诺
atac	atac
atelkom	atelkom

名称	ID
撒哈拉地图集	atlas_sahara
atmx	atmx
阿提卡银行	attica_bank
有吸引力的	有吸引力的
au	au
au Naturel	au_naturel
a name de la rose	au_nom_de_la_rose
aurigacrown	aurigacrown
auto kreso	auto_kreso
Autonet&Oil	Autonet&Oil
autocité	autocité
autolux	autolux
autoparki	autoparki
向前	向前
avec。	avec。
狂热的酒店	avid_hotel
杜鹃花	杜鹃花
açai beat	acaí_beat
açai no ponto	acaí_no_ponto
b&binn	b&binn

名称	ID
b-fit	b-fit
bySI	bySI
bayGap	bayGap
backWerk	backWerk
balladins	balladins
banak importa	banak_importa
香蕉叶	香蕉叶
BancaPulia	BancaPulia
banca caripe	banca_caripe
banco cencosud	banco_cencosud
bancoctt	bancoctt
银行家。	银行家。
bankoa CRÉDIT AGRICOLE	Bankoa_Crédit_Agricole
大西洋银行	大西洋宴会
bansefi	bansefi
barway	barway
battalzade	battalzade
BauMax	BauMax
baydoner	baydoner
bayn	bayn

名称	ID
bb 的咖啡厅	bb's_café
bbg 购物中心	bbg_shopping_mall
bcu	bcu
bd's 蒙古烧烤	bd's_mongolian_Gril
床的	床的
beefchemghiang	beefchemghiang
begendik	begendik
bem star	bem_estar
berca	berca
伯纳多	伯纳多
bert's	bert's
绿柱石。	绿柱石。
beso de coco	beso_de_coco
beti	beti
betty's	betty's
beypilic	beypilic
bfs Foods	bfs_Foods
bft	bft
bh	bh
bi1	bi1

名称	ID
bi1 超级市场	bi1_supermarché
大红茶	big_big_big_bic_tea
biju	biju
bimba & Lola	bimba_&_lola
宾西纳药房	宾西纳_药房
生物与生物	bio_&_bio
biocoop	biocoop
biougnach	biougnach
Regent 小酒馆	Bistro_Regent
户外露营	露营_户外
black'sm	black'smith
blaqué	blaqué
bloom Senso 听力中心	Bloom_senso_Hearing_Cent
蓝蛙	蓝蛙
bluecorner	bluecorner [ev_charging_station]
bluecorner	bluecorner [女士_服装]
蓝水星	蓝水星
蓝珍珠	蓝珍珠
boali	boali
董事会成员	董事会成员

名称	ID
boesner	boesner
bolero	bolero
bonprix	bonprix
Bonárea	Bonárea
boreli	boreli
bossini	bossini
血压脉冲	bp_pulse
bpost bank	bpost_bank
bpost banque	bpost_banque
休息	休息
brice	brice
brinka	brinka
brissa	brissa
百老汇电影院	百老汇电影院
brodokomerc nova	brodokomerc_nova
布鲁克林	布鲁克林
buddelei	buddelei
廉价运动	预算体育
budmil	budmil
建筑商仓库	建筑商_仓库

名称	ID
bulhosa	bulhosa
burbak	burbak
butcha	butcha
butik	butik
顺其自然	buy_the_way
bebé 9	bebé_9
bellissima	bellissima
c.a.r.o.	c.a.r.o.
CPort CU	cport_cu
cacharel	cacharel
缓存缓存	cache_cache
cafe DROP	cafe_dropt
雀巢咖啡厅	Cafe_nesCa
咖啡协会	咖啡社协会
cafe7gra	cafe7gra
caffe ti-amo	caffe_ti-amo
caffe ben	咖啡_bene
咖啡屋	café_house
Caixa 圣维森特	Caixa_Sant_Vicent
caxa callosa	caixa_callosa

名称	ID
流行的储物柜	caixa_受欢迎
caixacallossa	caixacallossa
caixalmassora	caixalmassora
caixapeter	caixapeter
caixaturís	caixaturís
cajaStur	cajaStur
caja circulo	caja_circulo
cajamar	cajamar
cajaviva	cajaviva
calcantes	calcantes
calçado Guimaraes	Calçado_Guimaráes
坎比奥	坎比奥
骆驼活跃	骆驼_active
西部露营	camp'in_ouest
cantwo	cantwo
caprabo	caprabo
caracol	caracol
carenex	carenex
carifermo	carifermo
caroline biss	caroline_biss

名称	ID
casa viva	casa_viva
cash Crusaders	cash_crusaders
现金快递专卖店	Cash_express_Specialty
castorama	castorama
catch	catch
catimini	catimini
cecabank	cecabank
cedele	cedele
celestino	celestino
大提琴俱乐部	celio_Club
cellC	cellC
CellularPlus	CellularPlus
centea	centea
中心点	中心点
独一无二的中心	centros_unico
cerise	cerise
意大利陶瓷	cerámica_italia
chadotel	chadotel
通道	通道
长沙农村商业银行	长沙农村商业银行

名称	ID
char.gy	char.gy
Charmaine	Charmaine
chausport	chausport
Chauss 博览会	chauss_expo
chawan	chawan
chemmanur 国际	国际 chemmanur_
cherry 手机	cherry_mobile
chi	chi
池尚	池尚
chicco	chicco
鸡共和国	鸡肉共和国
chictea	chictea
放松一下	放松一下
冷藏箱	冷藏箱
中国电影	China_film
中国炒锅	china_wok
chunfunhow	chunfunhow
电影最大值	电影最大值
电影星球	电影星球
辛特拉	辛特拉

名称	ID
citi handlowy	citi_handlowy
城市德尔海兹	City_delhaize
城市超级	城市超级
维也纳城市自行车	Citybike_Wien
城市生活	城市生活
citymax	citymax
克莱尔的	克莱尔的
克拉雷尔	克拉雷尔
clas ohlson	clas_ohlson
class'croute	class'croute
克劳迪奥快车	克劳迪奥快车
clayeux	clayeux
干净的公园	干净的公园
clickviaja.com	clickviaja.com
clim+	clim+
厘米	厘米
cm 化妆品市场	cm_化妆品市场
camana promet	cmana_promet
cmq farmacias	cmq_farmacias
co.op mart	co.op_mart

名称	ID
coccinelle 快车	coccinelle_express
codentry	codentry
Coeur de Fleurs	Coeur_de_Fleurs
今天的咖啡	今日咖啡
咖啡天堂	咖啡天堂
咖啡史密斯	咖啡史密斯
cojean	cojean
colorii	colorii
colruyt	colruyt
逗号	逗号
连接	连接
基本消费	基本消费
续。	续。
contém1g	contém1g
cook seker	cook_seker
coop	coop
coop 额外版	Coop_Extra
coop bau+Hobby	coop_bau+Hobbau
coop city	coop_city
coop pronto	coop_pronto

名称	ID
coop 餐厅	coop_restauran
coop 活力	coop_vitality
cop.copine	cop.copine
coquet	coquet
coradrive	coradrive
日冕	日冕
cort&cose	cort&cose
cosechas	cosechas
我开的课程	Courses_u_Drive
疯狂的摩卡	Crazy_mocha
创造	创造
creps	creps
cristobalcolon	cristobalcolon
crockid	crockid
crostini	crostini
ctt correios	ctt_correios
ctt 快递	ctt_expresso
ctu 拉面	ctu_ramen
cuf	cuf
custo 巴塞罗那	定制_巴塞罗那

名称	ID
cykelringen	cykelringen
cyta	cyta
czerwona torebka	czerwona_torebka
côté halles	côte_halles
d'zzit	d'zzit
da	da
daffer papiernictvo	daffer_papiernictvo
dal.komm COFFEE	dal.komm_Coffee
稻香村	稻香村
达沃尔	达沃尔
dbam o zdrowie	dbam_o_zdrowie
dd 的折扣	dd's_折扣
事实上	事实上
de'Excelso	de'_excelso
decodom	decodom
decolores	decolores
壳牌熟食店	deli_by_shell
deli2go	deli2go
三角洲劳埃德	delta_lloyd
demorhome	demorhome

名称	ID
denn's Biomarkt	denn's_Biomarkt
牙科由	dental_by
牙齿美学	牙齿美学
牙科医生	牙科医生
dfcu	dfcu
dfjb	dfjb
dfs	dfs
dico	dico
柴油加油站	柴油加油站
din sko	din_sko
恐龙	恐龙
磁盘	磁盘
diska	diska
diskonttank	diskonttank
沙发	沙发
dm	dm
docomo	docomo
dodohome 停车	dodoHome_停车
董来顺	董来顺
唐娜克莱尔	唐娜克莱尔

名称	ID
donpiso	donpiso
dookki	dookki
穿得到	dress_to
droper	droper
datc	datc
dukan	dukan
dulces ilusiones	dulces_ilusiones
dulcé café	dulce_café
dum barev	dum_barev
黄昏	黄昏
diy per di	dí_per_di
dukkan 汉堡	dukkan_burger
e HOMINNS	e_hominns
我们去西班牙	e_g_sain
e-shop.gr	e-shop.gr
eplus	eplus
e5 模式	e5 模式
eCo CU	eco_cu
eCar	eCar
eCard	eCard

名称	ID
eCosway	eCosway
eHi 汽车服务	ehi_car_service
eMila	eMila
esmokingWorld.co	esmokingWorld.co
额外门店	额外门店
eZone	eZone
每杯一杯	each_a_cup
地球音乐与生态学	地球_音乐_&_生态学
容易的	容易的
easyApothek	easyApothek
简易酒店	简易酒店
easylife 厨房	easylife_kit
吃寿司	Eat_sushi
Econui	Econui
edc by esprit	edc_by_esprit
自我	自我
埃菲尔光学	eiffel_optic
埃克洛酒店	EKLO_Hotels
超级治安员	el_vigilante_Super
el palacio de la PantalEta	el_palacio_de_la_ PantalEta

名称	ID
电子城市	电子城
艾丽卡	艾丽卡
元素新鲜	element_fresh
elipso	elipso
elisa	elisa
精英风暴	elite_strom
elkedjan	elkedjan
埃尔森	埃尔森
埃尔伍德人员配备	elwood_人员配备
emart	emart
每天都很聪明	emart_every
embar41 cadero	embar41_cadero
empark	empark
empik	empik
empik cafe	empik_cafe
en5 市场	en5_market
ena	ena
encuentro	encuentro
ender	ender
enel-med	enel-med

名称	ID
能源	能源
能源市场	能源市场
engbers	engbers
eni	eni
eni 咖啡厅	eni_café
eni 商店	eni_shop
enkoil	enkoil
enkoil 市场	enkoil_Market
招揽市场	招揽市场
ensitel	ensitel
企业	企业
epk	epk
ept	ept
era-postovni sporitelna	era-postovni_sporitelna
erafone	erafone
esia	esia
诚品书店	eslite_bookstore
space SFR	space_sfr
espaço casa	espaço_casa
espressolab	espressolab

名称	ID
essanelle	essanelle
esse	esse
essec	essec
estafeta	estafeta
住宿	住宿
pete	pete
etem	etem
欧元 SKLEP	Euro_sklep
欧元中心	欧元中心
euroil	euroil
euskaltel	euskaltel
确切！	确切！
专家	专家
专家 Elektro	专家_Elektro
特快租车公司	快车租赁_汽车_代理
快递市场	快递市场
额外区域	额外区域
极度亲密关系	extreme_intimo
eyeQ	eyeQ
眼镜	眼镜

名称	ID
眼睛 + 更多	眼睛_+_更多
f.y.e。	f.y.e。
faik sönmez	faik_sönmez
家族	家族
熟悉	熟悉
家庭市场	家庭市场
familotel	familotel
famrice	famrice
famsa	famsa
法玛西亚	法玛西亚
圣拉斐尔药店	圣拉斐尔药店
圣玛丽亚药房	farmacia_santamarí
farmacias FarmaCon	Farmacias_Farmacon
farmacias yza	farmacias_yza
farmasanitas	farmasanitas
farmashop	farmashop
农场食品	农场食品
fasa	fasa
时尚世界	时尚世界
fashion.chiu-hiang	fashion.chiu-hiang

名称	ID
时尚快车	时尚快车
快车道	快车道
favehotel	favehotel
菲尔曼	菲尔曼
菲奥伦齐	菲奥伦齐
鱼面	鱼面
fitametrica	fitametrica
健身俱乐部概念	健身俱乐部概念
fix'n'go	fix'n'go
火烈鸟	火烈鸟
闪光灯	闪光灯
flexi	flexi
flocafé	flocafé
flormar	flormar
花卉露营	flower_campings
flunch	flunch
fnac	fnac
foreva	foreva
fornet	fornet
论坛现在是	forum_saat

名称	ID
论坛体育	forum_sport
法国 PARE-BRISE	France_pare-Brise
弗朗切斯科·马可尼	francesco_marconi
免费唱片店	免费唱片商店
清新又简单	fresh_&_easy
新鲜的选择	新鲜的选择
新的	新的
新点	新点
friseur coco	friseur_coco
frutiquello Sorvetes	frutiquello_sorvetes
fulli	fulli
futuraplus	futuraplus
future bank	future_bank
fötex	fötex
g.a.s.	g.a.s.
gamayoyo	gamayoyo
游戏百货商店	游戏百货商店
Gamestation	Gamestation
汽油蓝色牛仔裤	汽油_蓝色_牛仔裤
gaziantepli tatlibak	gaziantepli_tatlibak

名称	ID
gdz tea	gdz_tea
gef	gef
getin bank	getin_bank
giap	giap
ginatricot	ginatricot
吉诺·罗西	gino_rossi
glinmermer	glinmermer
全球网	全球网
顺其自然	go_natural
天哪 apotheek	goed_apotheek
黄金积分	golden_point
gomart	gomart
gomex	gomex
gostinos	gostinos
美食汉堡厨房	美食汉堡厨房
免费	免费
绿色更好	green_is_better
绿色公园	绿色公园
绿点	绿点
groszek	groszek

名称	ID
gundogdu	gundogdu
h3	h3
haan	haan
栖息地	栖息地
Habizone	Habizone
hagebau	hagebau
俳句	俳句
洗发剂	洗发剂
hakmar express	hakmar_express
halfords 汽车中心	halfords_autocentre
han's deli	han's_deli
方便	方便
快乐	快乐
happylemon	happylemon
hatemoglu	hatemoglu
hebe	hebe
你好	你好
你好世界	你好世界
hhgregg	hhgregg
河马	河马

名称	ID
希萨尔	希萨尔
lsushi	lsushi
业余工艺	业余工艺
home&you	home&you
homelink	homelink
horno sanbrandan	horno_sanbrandan
热豆腐	hot_dou_fu
hot fish	hot_fish
HotelF1	HotelF1
house	house
咖啡屋	House_offees
火公殿	火公殿
超大限度 XXX	hyper_maxima_xxx
超级城市	超级城市
i-cup	i-cup
i-拉面	i-拉面
i-migoo	i-migoo
i-Mobile	i-Mobile
iBeat	iBeat
iCoffee	iCoffee

名称	ID
我没DENTi错	我没DENTi错
iecon	iecon
爱康国宾医疗中心	爱康_国宾_医疗中心
iLeap	iLeap
iMoneyBank	iMoneyBank
Ni's	Ni's
inoVine	inoVine
iPlace	iPlace
iQ 信用合作社	iq_credit_Uni
iRoo	iRoo
iRent	iRent
iRI	iRI
iShop Mixup	iShop_mixup
iSold It	isold_it
iSpot	iSpot
我工作室	我工作室
iTowns	iTowns
iTalk 蜂窝网络	italk_蜂窝网络
iberCaja	iberCaja
紫莓	紫莓

名称	ID
朱鹭	朱鹭
宜必思经济酒店	宜必思预算酒店
宜必思厨房	宜必思厨房
宜必思风格	ibis_styles
ibq	ibq
ibyte	ibyte
Iceason	Iceason
if	if
怡亨	怡亨
iittala	iittala
继胜	继胜
ikia	ikia
ikingza	ikingza
意大利面市场	il_mercato_di_pasta
伊拉	伊拉
ilunion	ilunion
imo	imo
imo 洗车	imo_car_wash
imza	imza
Influx	Influx

名称	ID
inblu	inblu
incon-okná	incon-okná
indi & cold	indi_&_cold
研究所	研究所
乐器馆	乐器馆
冬季洞穴	Inter_Caves
冬季赛车	inter_cars
设计间	设计间
interio	interio
Internacionale	Internacionale
intimissimi	intimissimi
intu	intu
反转	反转
inwi	inwi
issta	issta
它是	它是
这是时尚	这是 _fashion
这是时尚地铁	这是_fashion_metro
伊萨拉	伊萨拉
itso tea	itso_tea

名称	ID
ixina	ixina
izzibuil	izzibuil
izzue	izzue
jabba	jabba
jacadi	jacadi
jamin	jamin
简·诺曼	jane_noman
jbc	jbc
jellyinn	jellyinn
jem & fix	jem_&_fix
jenjudan	jenjudan
jetline e.com	jetline_e.com
江门融和农村商业银行	江门_荣和_农村通信_银行
江苏洪泽农村商业银行	江苏_洪泽_农村_商业银行
江苏丽水农村商业银行	江苏_丽水_农村通信_银行
江苏遂宁农村 CommBank	江苏_遂宁_农村_ CommBank
江苏宜兴农村商业银行	江苏_宜兴_农村通信_银行
江苏仪征农村商业银行	江苏_仪征_农村通信_银行
江阴农村商业银行	江阴农村商业银行
吉吉镇	吉吉镇

名称	ID
约翰迪尔	john_deere
joueclub	joueclub
喜悦的	喜悦的
juguettos	juguettos
多汁的露西	juicy_lucy
联合汽车玻璃	junited_AutoGlas
justrite	justrite
kahveci HACIBABA	kahveci_HaciBaba
kaissa	kaissa
kak raz	kak_raz
karinca 设计	karinca_design
karlovacka banka	karlovacka_banka
karmen Hali	karmen_hali
kasik-la	kasik-la
喀拉拉邦银行	喀拉拉邦银行
kgcshop	kgcshop
孩子的微笑	孩子的微笑
kika	kika
kikki.k	kikki.k
kimlik	kimlik

名称	ID
景福珠宝	金福珠宝
kinko's	kinko's
Kino HELIOS	kino_Helios
kircicegi	kircicegi
小厨房	小厨房
kokoriko	kokoriko
komsu firin	komsu_firin
koçtas	koçtas
ktc	ktc
kukartka	kukartka
kutxabank	kutxabank
kvik	kvik
kw 葡萄牙	kw_葡萄牙
l'appart 健身	l'appart_Fitnes
la Cai	la_Caisa
la ballena azul	la_ballena_azul
la despensa	la_despensa
la gaviotta	la_gaviotta
la rebaja plus	la_rebaja_plus
laboo	laboo

名称	ID
ladenetz	ladenetz
淑女与绅士之城	Lady_&_gentleman_City
女士移动	女士_移动
陆顶	陆顶
老凤香	老凤香
老天木	老天木
laranjinha	laranjinha
le DRIVE Intermarché	le_drive_intermarché
le Garage	_Garage
le vélo	_velo
左撇子	左撇子
传说	传说
lenskart.com	lenskart.com
生物商行	les_comptoirs_de_la_bio
les enphants	les_enphants
les mille feuilles de Liberte	les_mille_feuilles_de_liberte
lesninaxXXL	lesninaxXXL
leto	leto
莲香楼	莲香楼
生活 8	life_8

名称	ID
生活:)	生活:)
lijiabababy	lijiabababy
莉莉	莉莉
柠檬酒	柠檬酒
Livera	Livera
丽香	丽香
ll-gdb 早午餐	ll-gdb_Brunch
lohongka	lohongka
萝拉雷伊	Lola_rey
伦敦时报	伦敦时报
louwailou	louwailou
lu 火锅	lu_hot_pot
露西运动服	露西_运动服
露露超级市场	lulu_hypermart
lululemon athletica	lululemon_athletica
lumimart	lumimart
午餐花园	午餐_花园
luxus	luxus
四处闲逛	lyn_round
我的电话	m_tel

名称	ID
mBank	mBank
MD-酒店	MD-酒店
mac_drugstore	mac_drugstore
大师	大师
桃花心木	桃花心木
maje	maje
makro	makro
妈妈和爸爸	mamas_&_papas
mammamia	mammamia
mamz'elle	mamz'elle
Maniet	Maniet
manoukian	manoukian
manso	manso
maras	maras
marché franprix	marche_franprix
marimekko	marimekko
市场	市场
玛莎·梅	martha_may
丸井（新泻县）	marui_（新泻县）
吉祥物	吉祥物

名称	ID
mashreq	mashreq
masymas	masymas
马塔斯	马塔斯
materna	materna
莫里斯	莫里斯
maxitoys	maxitoys
市长	市长
mcel	mcel
肉蝇	肉蝇
meble GAWIN	meble_gawin
媒体专家	媒体专家
mediaprint-kapa	mediaprint-kapa
药品	药品
medstop apotek	medstop_apotek
melissa	melissa
回忆录	回忆录
回忆录市场	回忆录市场
孟子元	孟子元
merca mas	merca_mas
mercatto	mercatto

名称	ID
meritum 银行	Meritum_Bank
merkal calzados	merkal_calzados
mi-vami	mi-vami
mibanco	mibanco
mibanco-agente	mibanco-agente
micasa	micasa
migrolino	migrolino
米克尔	米克尔
米拉	米拉
mimo 的	mimo 的
minelli	minelli
迷你 Big C	mini_big_C
迷你 MAXI	mini_maxi
mini raxevsky	mini_raxevsky
minicuotas RIBEIRO	miniCuotas_ribeiro
minisom	minisom
minit	minit
minoshe	minoshe
mio	mio
mipec	mipec

名称	ID
混合电子设备	混音电子
混合电子学	混合电子学
mixirica	mixirica
mo	mo
mobil 88	mobil_88
mobilcom-debitel	mobilcom-debitel
mobilezone	mobilezone
moj DUCAN	MOJ_Ducan
mola	mola
莫拉市场	mola_mark
momo shop	momo_shop
莫娜	莫娜
monnari	monnari
meuble 先生	monsieur_meuble
妈妈护理	妈妈护理
动机	动机
motonet	motonet
电影明星	电影明星
moya	moya
moyo	moyo

名称	ID
bean 先生	mr_bean
bun 先生	mr_bun
mtc	mtc
mudo.conce	mudo.conce
多红色	多红色
多红色 Agente	多红色 Agente
multicoisas	multicoisas
Multifacil-Agente	Multifacil-Agente
多光学	多光学
musgo	musgo
muwubbq	muwubbq
我的新闻	我的新闻
mycar	mycar
家具品牌	mobel_marki
mömax	mömax
nH 精选酒店	NH_精选酒店
naartjie	naartjie
nahkauf	nahkauf
裸体果汁吧	naked_juice_bar
namco	namco

名称	ID
给它起名字	name_it
荷兰国家队	nationale_nederlanden
naturéo	naturéo
nc+	nc+
NeoBank	NeoBank
neoness	neoness
巢穴银行	Nest_Bank
nestro	nestro
netonnet	netonnet
新平衡	新余额
新闻快报	新闻快报
新风格	新风格
next	next
nextel	nextel
漂亮的九	nice_nine
好东西 Paloma S.	nice_Things_Paloma_s。
nick scali	nick_scali
nico 和	nico_and
nino 的	nino 的
ninso	ninso

名称	ID
no+vello	no+vello
nomnom	nomnom
nooi	nooi
norli	norli
诺索洛意大利	nosolo_Italia
nostrum	nostrum
注意！	注意！
NowTV	NowTV
nucha	nucha
num num	num_num
nur pastaneleri	nur_pastaneleri
纽约儿童俱乐部	ny_kids_club
NYDC	NYDC
o2 早午餐	o2_brunch
库存过剩	库存过剩
奇怪的莫莉	odd_molly
odeabank	odeabank
odido	odido
oechsle	oechsle
办公仓库	办公仓库

名称	ID
ofix	ofix
okna MACEK	Okna_macek
oleofat	oleofat
奥尔森	奥尔森
omey	omey
一个宿	one_juku
一个人的露台	一个人的露台
oodji	oodji
oohaney's	oohaney's
ooredoo	ooredoo
opema	opema
打开体育	公开体育
擎天柱	擎天柱
橙色	橙色
猎户座	猎户座
OULA	OULA
oysho	oysho
panignacio	panignacio
parkoil	parkoil
意大利面咖啡	pasta_caffé

名称	ID
pastafiore	pastafiore
pauw	pauw
pc.linic	pc.linic
pdsa	pdsa
pekarna KABAT	pekarna_kabat
鹈鹕	鹈鹕
pellicano	pellicano
完美 10	perfect_10
peros	peros
petit madoca	petit_madoca
花瓣	花瓣
petrolina	petrolina
petrotürk	petrotürk
Petroturk-Market	Petroturk-Market
pia Sapido	Pia_sapido
野餐	野餐
皮埃尔·卡丹	pierre_cardin
pimkie	pimkie
pinchos	pinchos
pingo doce	pingo_doce

名称	ID
pingo doce & GO！	pingo_doce_&_go！
粉色女人	pink_woman
投手的	投手的
pitta pappou	pitta_pappou
披萨披萨	披萨_披萨
番茄披萨	披萨_番茄
plaisio	plaisio
platanitos	platanitos
plazavea	plazavea
pluricosmetica	pluricosmetica
plusfrésc	plusfrésc
pmg	pmg
pomp van het volk	pomp_van_het_volk
ponto	ponto
pop！酒店	pop！ _酒店
波塔	波塔
陶器谷仓的孩子	pottery_barn_kids
ppc	ppc
pramen.cz	pramen.cz
实践	实践

名称	ID
漂亮的	pretty_one
公主餐	Princess_Meals
校长海莉	校长_海莉
prio	prio
prochnik	prochnik
促销	促销
prono	prono
原型	原型
邻近的	邻近的
代理 DELHAIZE	代理_delhaize
psb MROWKA	psb_mrowka
pt	，pt”
puljanka	puljanka
铂尔曼	铂尔曼
脉冲卫星	脉冲卫星
Pupa	Pupa
quem disse	berenice？
quironsalud	quironsalud
rabalder	rabalder
拉斐尔酒店	rafael_hoteles

名称	ID
rapid'flore	rapid'_flore
rbt	rbt
re: bar	re: bar
real	real
redbox	redbox
改革斯塔克·马丁	reformStark_Martin
reifen.com	reifen.com
relexa	relexa
撤销	撤销
里亚德银行	riyad_bank
rochebobois	rochebobois
rompn'roll	rompn'roll
布兰奇街	rue_blanche
rue21	rue21
瑞丽 100 Fitness	ruili_100_Fitness
ruralcaja	ruralcaja
saadeddin 糕点	saadeddin_pastry
saat&saat	saat&saat
萨巴	萨巴
sacha	sacha

名称	ID
saemes	saemes
sames EV	Saemes_ev
野生动物园	野生动物园
salmoiraghi & vigano	salmoiraghi_&_vigano
旧金山咖啡公司	旧金山咖啡公司
桑德罗	桑德罗
佐野	佐野
sanoil	sanoil
sanoil 市场	三原油市场
santafé SUPERMERCADOS	Santafé_Supermercados
sanwa	sanwa
sapatália	sapatália
莎拉·帕西尼	sarah_pacini
sasol delist！	sasol_delight！
节省食物	save_on_foods
sb mart	sb_mart
schuh	schuh
scoozi	scoozi
范围	范围
时光雕刻	在时间内雕刻

名称	ID
海鲜餐厅 YU GONG YU PO	Seafood_restaurant_yu_po
selu	selu
senQ	senQ
senao	senao
sevil	sevil
尚影影院	尚影影院
shasa	shasa
床单街	sheet_street
旅程中的 shi	Shi_by_Journe
shoeby 时尚	shoeby_fashion
shop'n go	shop'n_go
si si	si_si
siba	siba
sidis	sidis
sigma	sigma
简单字符	simple_caractere
simpo	simpo
singulato	singulato
skopunkten	skopunkten
慢的	慢的

名称	ID
小世界	小世界
聪明	聪明
智能合资企业	Smart_JV
智能学习	智能学习
smarteyes	smarteyes
微笑市场	微笑市场
笑起来	向上微笑
smooy	smooy
smp	smp
smyk	smyk
snogurt	snogurt
sodimac	sodimac
sofrapa	sofrapa
土壤市场	土壤市场
索内拉	索内拉
sorli discau	sorli_discau
规格保护器	spec_savers
spendimeglio	spendimeglio
香料烈酒	香料精灵
spizzico	spizzico

名称	ID
运动 1	sport_1
运动的	运动的
运动戒指	运动戒指
体育场景	体育场景
square	square
st1	st1
stara	stara
明星食品	明星食品
starmaxx	starmaxx
斯泰尔曼	斯泰尔曼
steinecke	steinecke
一块一块石头	stone_by_stone
停下咖啡馆	停下来咖啡厅
street	street
工作室 88	studio_88
sublime tentação	sublime_tentação
周日旅馆	周日旅馆
阳光新鲜	阳光新鲜
日落烧烤	sunset_grill
sunward 渔业餐厅	Sunward_fishery_Restauran

名称	ID
supeco	supeco
超级 10 美发公司	超级_10_haircompany
supermec	supermec
surtidora de aves	surtidora_de_aves
susanagateira	susanagateira
寿司屋	sushi_house
sushian	sushian
sushico	sushico
swatch	swatch
SweetFrog 优质冷冻酸奶	SweetFrog_Premium_frozen_酸奶
来自天堂的糖果	来自天堂的甜点
瑞士人	瑞士人
瑞士魅力酒店	swiss_charme_hotels
是的	是的
t&T	t&T
禁忌	禁忌
tailor&co	tailor&co
橘子咖啡	tangerina_caffe
罐装果汁	tank_juice
tanyutou	tanyutou

名称	ID
tashas	tashas
tatlici tombak	tatlici_tombak
tavvuk	tavvuk
team	team
泰迪专卖店	tedi_Speciality_store
telcel	telcel
tele.ring 商店	Tele.ring_Shop
Telenor banka	telenor_banka
远程披萨	远程披萨
十到十	ten_to_ten
网球	网球
TerstAl	TerstAl
tercas	tercas
计数器	计数器
Focus FCU	the_focus_FCU
食品制造商	_foodmaker
大堂咖啡厅	The_Lobby_Café
那个 niu	the_niu
thiriet	thiriet
tigre	tigre

名称	ID
tiimari	tiimari
tk3c	tk3c
tkmart	tkmart
tnconnect	tnConnect_c
太糟糕了	to_daylicious
汤姆·法尔	tom_farr
自上而下	自上而下
topitop	topitop
上衣！ 在 SPAR	上衣！ _at_spar
交通	交通
trgocentar	trgocentar
trgonom	trgonom
trigema	trigema
trinkgut	trinkgut
true	true
真正的咖啡	真咖啡
truenorth	truenorth
turbo limac	turbo_limac
涡轮增压运动	涡轮增压运动
真的	真的

名称	ID
tw 电信	tw_teleco
二十家商店	Twenty_Shop
twhct	twhct
twist & tango	twist_&_tango
twój MARKET	twój_Market
u 巡回演出	u_tour
uBreakiFix	uBreakiFix
ubank	ubank
ubitricity	ubitricity
uLanka	uLanka
unico	unico
联合市场	unide_market
不合时宜的	不合时宜的
unikulma	unikulma
由 PETCO 释放	Unleashed_by_petco
uno-e	uno-e
住宅区	住宅区
usbbank	usbbank
v desyatochku	v_desyatochku
vabbi	vabbi

名称	ID
vacca matta	vacca_matta
vagues océanes	vagues_oceanes
valero forn tradicional	valero_forn_tradicional
valvi	valvi
van Haren	van_Haren
范达伦	van_dalen
vega	vega
veigas	veigas
velcom	velcom
velib'	velib'
veritas	veritas
Viajes Insular	Viajes_Insular
vibo viajes	vibo_viajes
生活和咖啡	vida_e_caffé
vip 消费电子商店	VIP_消费电子产品商店
viriyahcare	viriyahcare
维生素	维生素
viva 家居专卖店	Viva_home_Specialty_store
vivacamp	vivacamp
沃达丰	沃达丰

名称	ID
你的市场	votre_marché
vulkan	vulkan
velo'v	velo'v
velobleu	velobleu
vélomagg'	vélomagg'
vérif'autos	vérif'autos
w biegu cafe	w_biegu_cafe
w52	w52
waffle'ci akin	waffle'ci_akin
wagamama	wagamama
华莱士	华莱士
瓦利斯	瓦利斯
wanko	wanko
欢迎门店	欢迎店铺
weldom	weldom
west49	west49
葡萄酒馆	葡萄酒馆
防风运动服	防风运动服
wojas	wojas
女人的秘密	女人的秘密

名称	ID
wuyutai	wuyutai
哈哈哈哈哈哈哈哈	xiabu_siabu
陈晓畅	陈晓畅
谢老松	谢老松
y1s	y1s
yankys	yankys
yatsan	yatsan
yd。	yd。
你好 KARCHER	yello_Karcher
黄色！ 村庄	黄色！ _村庄
英健跆拳道	英健_跆拳道
yoggi	yoggi
yogoberry	yogoberry
yoigo	yoigo
yotsuba	yotsuba
年轻区	年轻人专区
你的酒店	你的酒店
z.i.p	z.i.p
zdrowie	zdrowie
Zen place 普拉提	zen_place_pilates

名称	ID
张怡源	张怡源
ziaja	ziaja
zolla	zolla
zolotoy	zolotoy
zpizz	zpizz
哈哈哈哈	À_LA_BONNE_HEURE
阿巴科	阿巴科
Agua Doce	Agua_Doce
Agua de Cheiro	Agua_de_Cheiro
风水区	风水区域
AHLÉNS	AHLÉNS
奥兰银行	奥兰银行
Opent Bakeri	Opent_Bakeri
CHIG KÖFTECI BILAL USTA	CHIG_KÖFTECI_BILAL_USTA
CITIR 华夫饼	CITIR_WAFFLE
Chaglar Pen	Chaglar_Pen
Chanakkale Seramik	Chanakkale_Seramik
Cekoil	Cekoil
捷克石油市场	Cekoil_Market
奇蒂尔·乌斯塔	Chitir_Usta

名称	ID
Eggo	Eggo
ECOMARCHÉ	ECOMARCHÉ
艾兰	艾兰
Elephant Bleu	Elephant_Bleu
冰岛银行	冰岛银行
Islandspostur	Islandspostur
BOB	BOB
Otica Lider	Otica_Lider
Oticas DINIZ	Oticas_diniz
Oticas do Povo	Oticas_do_POVO
öoB	öoB
Osterreichische Verkehrskreditbank	österreichische_Verkehrskreditbank
Estgöta Enskilda Bank	östgöta_Enskilda_Bank
厄兹迪莱克	厄兹迪莱克
Ødegaard Dame	Ødegaard_Dame
Økonomisko	Økonomisko
Ønskebörn	Ønskebörn
东基德斯克银行	Østjydsk_Bank
尤尼科酒店	Unico_Hotels
eggo	eggo

名称	ID
epi 服务	epi_service
埃拉姆	埃拉姆
exito vecino	exito_vecino
oticas carol	oticas_carol
أسواق المزرعة	أسواق_المزرعة
أموحمزة	أموحمزة
إف جي فور لندن	____
إنوسيون ووردواي	إنوسيون_ووردواي
???????	??????????
\”	\”
الثوب السعودي	_
\\\\	\\\\
??????????????	??????????????????????
\\\\	\\\\
بدون إسم	_
بروكلي بيتزا وباستا	_
بن داود	_
بن هادي لتأجير السيارات	_
بنز كوكيز	_
بيتزاراتي	بيتزاراتي

名称	ID
زناجيني‌ب	زناجيني‌ب
ته وكات	_
ني‌صلا ةرد	_?
هي‌فاك روتكد	_
هي‌فاك زود	_
ياو اامور	_
نادير	نادير
سواه كي‌تس	_
انيمس	انيمس
جنك امراوش	_
وتيرامراوش	وتيرامراوش
ةياش	ةياش
تارايسلا ريجأتل ني‌سح ةكرش	__
كبوش	كبوش
ءايضلا نرف	_?
ياره ان ياراك	_____
نوتناك	نوتناك
سي‌ربسكا مرك	_
تكرام ليوناام	تكرام_ليوناام
موت موت معطم	__

名称	ID
يركيب شنم	_
هيفاك تلكوش نتلوم	_
تالواقمالوا ةسدنهلل ياديوية	تالواقمالوا_ةسدنهلل_ياديوية
ةوهقال موي	_
ā行	ā行
ā	ā
Ai	Ai
Ai亭	ā亭
！	！
优惠券	优惠券
いなげや	いなげや
うまいものいっぱい　いろはにほへと	うまいものいっぱい　いろはにほへと
おおぎやラーメン	おおぎやラーメン
おぼんｄｅごはん	おぼんｄｅごはん
お仏壇のはせがわ	お仏壇のはせがわ
寿司司	寿司司
かつさと	かつさと
かつや	かつや
道	道
屋	屋

名称	ID
かまどや	かまどや
からやま	からやま
天才	天才
寿司司	寿司司
がんこ	がんこ
。	。
行	行
行	行
“州”	州
くいもの屋わん	くいもの屋わん
forteafortaFortafor	forteafortaFortafor
店	店
寿司	寿司
くるまやラーメン	くるまやラーメン
ごはんどき	ごはんどき
屋	屋
道场	道场
局	局
さと	さと
さぼてんデリカ	さぼてんデリカ

名称	ID
しゃぶしゃぶどん亭	しゃぶしゃぶどん亭
温野菜	温野菜
菜	菜
しゃぶ葉	しゃぶ葉
しーじゃっく	しーじゃっく
本店	本店
すしざんまい	すしざんまい
三崎丸	三崎丸
三崎港	三崎港
子丸	子丸
太金发女郎	太金发女郎
屋	屋
すみれ	すみれ
屋	屋
せんたくウサギ	せんたくウサギ
局	局
そごう	そごう
坊	坊
たいらや	たいらや
友梨	友梨

名称	ID
自拍車	自拍車
ちゃんぽん亭	ちゃんぽん亭
ちょいす	ちょいす
ちよだ鮨	ちよだ鮨
ちりめん亭	ちりめん亭
亭	亭
八	八
つるやゴルフ	つるやゴルフ
てこや	てこや
てんがら	てんがら
てんや	てんや
亭	亭
とりせん	とりせん
とんでん	とんでん
太	太
どらっぐぱぱす	どらっぐぱぱす
どるふぃん	どるふぃん
どんどん亭	どんどん亭
dood	dood
花局	花局

名称	ID
村	村
兵	兵
にぎり長次郎	にぎり長次郎
舞	舞
はなまるうどん	はなまるうどん
寿司	寿司
屋顶 221	屋顶 221
场	场
太金发女郎	太金发女郎
ひごペット	ひごペット
ひらせいホームセンター	ひらせいホームセンター
びっくりドンキー	びっくりドンキー
ふらんす亭	ふらんす亭
ふわふわ	ふわふわ
ほっかほっか亭	ほっかほっか亭
ほっともっと	ほっともっと
FoodHEDOLDAR	FoodHEDOLDAR
まいばすけっと	まいばすけっと
まねきねこ	まねきねこ
まるまつ	まるまつ

名称	ID
信行	信行
优惠券	优惠券
行	行
行	行
行	行
bikinistike Scheadolemy	bikinistike Scheadolemy
むさしの森珈琲	むさしの森珈琲
行	行
屋	屋
やきとりセンター	やきとりセンター
大吉	大吉
店	店
やまや	やまや
やよい軒	やよい軒
行	行
ゆず庵	ゆず庵
太女郎了	太女郎了
ゆめタウン	ゆめタウン
ゆめマート	ゆめマート
よどやドラッグ	よどやドラッグ

名称	ID
ららぽーと	ららぽーと
りそなグループ　ビューアルッテＡＴＭ	りそなグループ　ビューアルッテＡＴＭ
行	行
FoodHodeldar toshh	FoodHodeldar toshh
优惠券	优惠券
局	局
局	局
アイン	アイン
アオキーズピザ	アオキーズピザ
アカカベ	アカカベ
アカチャンホンポ	アカチャンホンポ
アクトス	アクトス
アコレ	アコレ
アストロプロダクツ	アストロプロダクツ
アスビー	アスビー
アスビーファム	アスビーファム
アットパーク	アットパーク
アップガレージ	アップガレージ
アップガレージ　ライダース	アップガレージ　ライダース
アップル	アップル

名称	ID
アップル車検	アップル車検
学会	学会
アピタ	アピタ
アプレ　レ　クール	アプレ　レ　クール
アベイル	アベイル
アミカ	アミカ
アミーゴ	アミーゴ
屋	屋
アヤハディオ	アヤハディオ
アルカドラッグ	アルカドラッグ
アルペン	アルペン
アンデルセン	アンデルセン
アーサー	アーサー
イエローグローブ	イエローグローブ
イエローハット	イエローハット
イオン	イオン
イオンスタイル	イオンスタイル
イオンタウン	イオンタウン
イオンバイク	イオンバイク
イオンペット	イオンペット

名称	ID
行	行
イズミヤ	イズミヤ
イタリアントマト	イタリアントマト
イタリアントマトカフェジュニア	イタリアントマトカフェジュニア
イデーショップ	イデーショップ
イング	イング
イーネット	イーネット
ウェルパーク	ウェルパーク
ウエスト	ウエスト
ウエストパーキング	ウエストパーキング
ウエルシア	ウエルシア
ウォンツ	ウォンツ
ウジエスーパー	ウジエスーパー
エイブルパーキング	エイブルパーキング
エクセルシオールカフェ	エクセルシオールカフェ
エコス	エコス
エスピーシー	エスピーシー
エディオン	エディオン
エフワン	エフワン
エブリィビッグデー	エブリィビッグデー

名称	ID
エブリイ	エブリイ
エルセーヌ	エルセーヌ
エースコンタクト	エースコンタクト
オオゼキ	オオゼキ
オリジンデリカ	オリジンデリカ
当	当
オリックスレンタカー	オリックスレンタカー
オリンピック	オリンピック
オンデーズ	オンデーズ
オークワ	オークワ
オーケー	オーケー
オートアールズ	オートアールズ
オートウェイ	オートウェイ
オートウェーブ	オートウェーブ
オートバックス	オートバックス
カインズ	カインズ
カクヤス	カクヤス
カスミ	カスミ
カフェファディ	カフェファディ
・・	・・

名称	ID
カプリチョーザ	カプリチョーザ
カメラのキタムラ	カメラのキタムラ
人物	人物
カラオケバンバン	カラオケバンバン
歌屋	歌屋
カラオケ館	カラオケ館
CLUUB DAAM	CLUUB DAAM
カリモク家具ショールーム	カリモク家具ショールーム
カルディ	カルディ
カルド	カルド
大将軍将	大将軍将
カワサキ	カワサキ
品	品
部	部
カースタレンタカー	カースタレンタカー
カーセブン	カーセブン
カーチス	カーチス
王国	王国
北海道	北海道
ガスト	ガスト

名称	ID
ガッツレンタカー	ガッツレンタカー
ガリバー	ガリバー
キクチメガネ	キクチメガネ
石油	石油
キッチンオリジン	キッチンオリジン
キデイランド	キデイランド
キムラヤ	キムラヤ
キャンドゥ	キャンドゥ
キョーエイ	キョーエイ
堂	堂
キングファミリー	キングファミリー
クイーンズ伊勢丹	クイーンズ伊勢丹
局	局
クスリのアオキ	クスリのアオキ
クスリのコダマ	クスリのコダマ
クスリのサンロード	クスリのサンロード
岩崎崎	岩崎崎
·	·
局	局
クリエイトＳＤ	クリエイトＳＤ

名称	ID
クリナップショールーム	クリナップショールーム
グッデイ	グッデイ
グラッチェガーデンズ	グラッチェガーデンズ
グランドスラム	グランドスラム
グリーンボックス	グリーンボックス
グルメシティ	グルメシティ
ケンズガレージ	ケンズガレージ
ケーズデンキ	ケーズデンキ
ケーユー	ケーユー
------	------
ゲオ	ゲオ
ゲンキー	ゲンキー
コイデカメラ	コイデカメラ
コインパーク	コインパーク
コインランドリー・サンキュー	コインランドリー・サンキュー
コクミン	コクミン
局	局
コジマ	コジマ
石油	石油
局	局

名称	ID
コナミスポーツクラブ	コナミスポーツクラブ
コノミヤ	コノミヤ
コパン	コパン
コミックバスター	コミックバスター
コムパーク	コムパーク
店	店
局	局
コメリ	コメリ
兵	兵
コモディイイダ	コモディイイダ
部	部
コロラド	コロラド
・	・
コーナン	コーナン
コーナンホームストック	コーナンホームストック
PPR	PPR
コープ	コープ
ゴダイドラッグ	ゴダイドラッグ
ゴルフパートナー	ゴルフパートナー
・	・

名称	ID
ゴルフ 5	ゴルフ 5
ゴーゴーカレー	ゴーゴーカレー
ゴールフリー	ゴールフリー
サイクルスポット	サイクルスポット
サイクルベースあさひ	サイクルベースあさひ
サカゼン	サカゼン
サガミ	サガミ
サツドラ	サツドラ
サニー	サニー
サボイ	サボイ
サミット	サミット
サンエー	サンエー
サンキ	サンキ
サンキュードラッグ	サンキュードラッグ
サンキューマート	サンキューマート
局	局
サンクゼール	サンクゼール
サンジェルマン	サンジェルマン
サンデー	サンデー
サンドラッグ	サンドラッグ

名称	ID
サンマルクカフェ	サンマルクカフェ
サンリブ	サンリブ
ザめしや	ザめしや
ザグザグ	ザグザグ
.	.
.	.
シカゴピザ	シカゴピザ
シメノドラッグ	シメノドラッグ
シャディ	シャディ
シャディサラダ館	シャディサラダ館
シャトレーゼ	シャトレーゼ
シャンブル	シャンブル
シューズ愛ランド	シューズ愛ランド
.	.
ショウワパーク	ショウワパーク
ショップイン	ショップイン
シースリー	シースリー
ジェイエステ	ジェイエステ
ジェーソン	ジェーソン
ジェームス	ジェームス

名称	ID
ジップドラッグ	ジップドラッグ
ジャパン	ジャパン
ジャパンファーマシー	ジャパンファーマシー
ジャンボエンチョー	ジャンボエンチョー
ジャンボカラオケ	ジャンボカラオケ
堂堂店	堂堂店
ジュンテンドー	ジュンテンドー
ジョイス	ジョイス
ジョイフィット	ジョイフィット
ジョイフル	ジョイフル
本田	本田
ジョナサン	ジョナサン
ジョ二ーのからあげ	ジョ二ーのからあげ
ジョリーパスタ	ジョリーパスタ
ジョーカー	ジョーカー
ジーンズメイト	ジーンズメイト
スカイレンタカー	スカイレンタカー
局	局
ニーー	ニーー
スクールＩＥ	スクールＩＥ

名称	ID
スシロー	スシロー
スズラン	スズラン
スタジオアリス	スタジオアリス
スタジオマリオ	スタジオマリオ
・	・
？？？？？？？？？？？？	？？？？？？？？？？？？
ステーキのどん	ステーキのどん
ステーキガスト	ステーキガスト
スパーク	スパーク
ニニ四	ニニ四
スポーツデポ	スポーツデポ
スマイルサンタ	スマイルサンタ
スリムビューティハウス	スリムビューティハウス
行情	行情
スーパーアークス	スーパーアークス
スーパーオートバックス	スーパーオートバックス
スーパースポーツゼビオ	スーパースポーツゼビオ
----------	----------
----------	----------
スーパータイヨー	スーパータイヨー

名称	ID
スーパードラッグひまわり	スーパードラッグひまわり
スーパーバリュー	スーパーバリュー
スーパー・キッド	スーパー・キッド
ーーー玉出来	ーーー玉出来
スープストック東京	スープストック東京
セイコーマート	セイコーマート
セイムス	セイムス
セオサイクル	セオサイクル
局	局
セキスイハイム	セキスイハイム
セキチュー	セキチュー
行情	行情
セリア	セリア
セルモ	セルモ
セントラルスポーツ	セントラルスポーツ
タイム	タイム
タイムズカー	タイムズカー
タイムズカーシェア	タイムズカーシェア
タイムズパーキング	タイムズパーキング
タイヤガーデン	タイヤガーデン

名称	ID
タイヤセレクト	タイヤセレクト
タイヤランド	タイヤランド
タイヤ館	タイヤ館
タイヨー	タイヨー
タウンプラザかねひで	タウンプラザかねひで
タカぎ	タカぎ
タマパーク	タマパーク
タワーレコード	タワーレコード
ダイエー	ダイエー
ダイコクドラッグ	ダイコクドラッグ
ダイシン	ダイシン
ダイユーエイト	ダイユーエイト
ダイレックス	ダイレックス
ダイワハウス	ダイワハウス
ダスキンヘルスレント	ダスキンヘルスレント
ダスキンレントオール	ダスキンレントオール
ダックス	ダックス
局	局
チャレンジ・グループ	チャレンジ・グループ
チヨダ　はきごこち	チヨダ　はきごこち

名称	ID
ティップネス	ティップネス
テリオスパーキング	テリオスパーキング
テンポス	テンポス
・	・
ディオ	ディオ
ディスカウントストア　ルミエール	ディスカウントストア　ルミエール
ディスカウントドラッグコスモス	ディスカウントドラッグコスモス
ディッパーダン	ディッパーダン
ディバロ	ディバロ
デイリーヤマザキ	デイリーヤマザキ
デコホーム	デコホーム
王国	王国
デポ	デポ
トキハ	トキハ
行情	行情
トヨタホーム	トヨタホーム
共通店	共通店
トライアル	トライアル
トライプラス	トライプラス
トラストパーク	トラストパーク

名称	ID
トレジャーファクトリー	トレジャーファクトリー
トーホー	トーホー
トーマス	トーマス
ドクターアイズ	ドクターアイズ
ドトールコーヒーショップ	ドトールコーヒーショップ
ドムドムハンバーガー	ドムドムハンバーガー
ドラッグイレブン	ドラッグイレブン
ドラッグスギ	ドラッグスギ
ドラッグスギヤマ	ドラッグスギヤマ
ドラッグストアスマイル	ドラッグストアスマイル
ドラッグストアモリ	ドラッグストアモリ
ドラッグトップス	ドラッグトップス
ドラッグヤマザワ	ドラッグヤマザワ
・	・
ナビパーク	ナビパーク
biLikinize GEMASKEDY	biLikinize GEMASKEDY
ナフコツーワンスタイル	ナフコツーワンスタイル
（	（
ナポリの窯	ナポリの窯
ニコニコレンタカー	ニコニコレンタカー

名称	ID
ニシムタ	ニシムタ
ニッポンレンタカー	ニッポンレンタカー
ニュー・クイック	ニュー・クイック
ネクステージ	ネクステージ
ノジマ	ノジマ
ハックドラッグ	ハックドラッグ
ハッピードラッグ	ハッピードラッグ
ハニーズ	ハニーズ
ハリカ	ハリカ
ハンプティーダンプティー	ハンプティーダンプティー
ハートアップ	ハートアップ
バロー	バロー
バースデイ	バースデイ
バーミヤン	バーミヤン
パシオス	パシオス
パリミキ	パリミキ
パルコ	パルコ
パンタジー	パンタジー
パーキングナビ	パーキングナビ
三六五	三六五

名称	ID
パークアベニュー	パークアベニュー
パークイット	パークイット
パークジャパン	パークジャパン
パークネット	パークネット
ヒマラヤ	ヒマラヤ
ヒマラヤスポーツ＆ゴルフ	ヒマラヤスポーツ＆ゴルフ
ヒーローズ	ヒーローズ
ビジョンメガネ	ビジョンメガネ
ビックカメラ	ビックカメラ
ビッグウッド	ビッグウッド
ビッグエコー	ビッグエコー
ビッグモーター	ビッグモーター
ビバホーム	ビバホーム
ビブレ	ビブレ
ピアゴ	ピアゴ
ピザポケット	ピザポケット
・	・
ピザーラ	ピザーラ
ファインパーキング	ファインパーキング
哎呀！！	哎呀！！

名称	ID
ファーストキッチン	ファーストキッチン
局	局
局	局
フォーエル	フォーエル
フジカラーパレットプラザ	フジカラーパレットプラザ
フライングタイガー	フライングタイガー
フレスコ	フレスコ
フレスポ	フレスポ
フレッシュネスバーガー	フレッシュネスバーガー
フレッシュフーズナフコ	フレッシュフーズナフコ
フロプレステージュ	フロプレステージュ
フードマーケットマム	フードマーケットマム
ブックオフ	ブックオフ
ブリーズ	ブリーズ
ブルガリ	ブルガリ
ブロンコビリー	ブロンコビリー
プリンス	プリンス
プリンセス	プリンセス
プレミアム・アウトレット	プレミアム・アウトレット
ヘーベルハウス	ヘーベルハウス

名称	ID
ベイシア	ベイシア
器	器
ベストＯＮＥ	ベストＯＮＥ
ベローチェ	ベローチェ
ペガサス	ペガサス
ペットの専門店コジマ	ペットの専門店コジマ
ペットプラス	ペットプラス
ホクレン	ホクレン
ホビーオフ	ホビーオフ
ホリデイスポーツクラブ	ホリデイスポーツクラブ
ホリデー車検	ホリデー車検
ホリーズカフェ	ホリーズカフェ
ホームセンターバロー	ホームセンターバロー
ホームセンタームサシ	ホームセンタームサシ
ホームワイド	ホームワイド
ボーテ	ボーテ
ポムの樹	ポムの樹
ポンパドウル	ポンパドウル
FoodHEDAR	FoodHEDAR
マックスバリュエクスプレス	マックスバリュエクスプレス

名称	ID
マックハウス	マックハウス
マツモトキヨシ	マツモトキヨシ
マツヤデンキ	マツヤデンキ
ママイクコ	ママイクコ
マミーマート	マミーマート
マルエツ	マルエツ
マルエドラッグ	マルエドラッグ
マルキョウ	マルキョウ
マルショク	マルショク
マルナカ	マルナカ
マルヤ	マルヤ
マンマチャオ	マンマチャオ
ミキハウス	ミキハウス
ミサワホーム	ミサワホーム
ミスタータイヤマン	ミスタータイヤマン
ミニコープ	ミニコープ
ミュゼプラチナム	ミュゼプラチナム
ミライザカ	ミライザカ
ムラサキスポーツ	ムラサキスポーツ
メガ	メガ

名称	ID
メガネのタナカ	メガネのタナカ
メガネのヨネザワ	メガネのヨネザワ
三城	三城
メガネの愛眼	メガネの愛眼
メガネストアー	メガネストアー
メガネスーパー	メガネスーパー
本	本
赤札堂	赤札堂
二十二ー	二十二ー
モンタボー	モンタボー
ヤオコー	ヤオコー
ヤックスドラッグ	ヤックスドラッグ
ヤナセ	ヤナセ
ヤマザキYショップ	ヤマザキYショップ
ヤマザワ	ヤマザワ
ヤマダ電機	ヤマダ電機
ヤマナカ	ヤマナカ
·	·
ユアー· パーキング	ユアー· パーキング
ユーコープ	ユーコープ

名称	ID
ユーホー	ユーホー
ユーポス	ユーポス
ヨシヅヤ	ヨシヅヤ
ヨドバシカメラ	ヨドバシカメラ
ヨークベニマル	ヨークベニマル
ヨークマート	ヨークマート
ラ　パウザ	ラ　パウザ
ライトオン	ライトオン
ライフォート	ライフォート
ラウンドワン	ラウンドワン
ラオックス	ラオックス
ラッキーパーキング	ラッキーパーキング
ラビット	ラビット
・	・
リカーマウンテン	リカーマウンテン
リコ	リコ
リサイクルマート	リサイクルマート
リトルマーメイド	リトルマーメイド
リパーク	リパーク
リファレンス	リファレンス

名称	ID
リベラーラ	リベラーラ
リンガーハット	リンガーハット
リーガルシューズ	リーガルシューズ
ルネサンス	ルネサンス
ルミネ	ルミネ
庄屋	庄屋
レッドバロン	レッドバロン
局	局
レモン	レモン
ロイヤル	ロイヤル
ロイヤルハット	ロイヤルハット
ロイヤルホスト	ロイヤルホスト
ロヂャース	ロヂャース
ロッキー	ロッキー
ロピア	ロピア
ロフト	ロフト
1000	1000
ローソン・スリーエフ	ローソン・スリーエフ
----行	----行
ロータスクラブ	ロータスクラブ

名称	ID
ワイズマート	ワイズマート
ワッツ	ワッツ
ワンカルビ	ワンカルビ
ワンラブ	ワンラブ
ワークマン	ワークマン
ヴィクトリア	ヴィクトリア
ヴィクトリアステーション	ヴィクトリアステーション
ヴィレッジヴァンガード	ヴィレッジヴァンガード
・・	・・
一刻魁大堂	一刻魁大堂
一口茶屋	一口茶屋
一百百店	一百百店
七十七行	七十七行
七肉安安	七肉安安
万代	万代
三井	三井
三井住友信行	三井住友信行
三井住友行	三井住友行
三井住友行行行情	三井住友行行行情
三代目	三代目

名称	ID
三洋堂店	三洋堂店
三菱兑换券	三菱兑换券
三越	三越
在店里	在店里
上州屋	上州屋
不二家	不二家
中三	中三
中京行	中京行
中国行	中国行
中央	中央
中学院	中学院
串かつでんがな	串かつでんがな
田中	田中
家物语	家物语
久	久
井	井
善	善
百货店	百货店
yupen	yuguan
丸	丸

名称	ID
久兵屋	久兵屋
九州屋	九州屋
二木头	二木头
五味八珍	五味八珍
井筒屋	井筒屋
kingsordAR	kingsordAR
京	京
京王	京王
京王百货店	京王百货店
京也行	京也行
京	京
京都信贷金库	京都信贷金库
京都行	京都行
京阪百货店	京阪百货店
仙台行	仙台行
我行	我行
丸	丸
但马行行	但马行行
住友林业	住友林业
佐佐共行	佐佐共行

名称	ID
佐佐行	佐佐行
店铺风格	店铺风格
biLigindique	biLigindique
biLigindique	biLigindique
biLiquide	biLiquide
bijoindique sechiondie sechiondaicat	bijoindique sechiondie sechiondaicat
biLigniationa	biLigniationa
bibeastroom	bibeastroom
太塾	太塾
偶	偶
长屋	长屋
元祖寿司司	元祖寿司司
全日美食	全日美食
八	八
八十二行	八十二行
公交车场公交车	公交车场公交车
共享 “未来”	共享 “未来”
李久	李久
kurku 生活馆	kurku 生活馆
ga屋	ga屋

名称	ID
北味行地酒北海道	北味行地酒北海道
北九州行	北九州行
北前高田屋	北前高田屋
北國行	北國行
北日本行	北日本行
北洋行	北洋行
北海道金库	北海道金库
北海道行行	北海道行行
北都行	北都行
Northernalno	Northernalno
北行	北行
十八和行	十八和行
十六行	十六行
千禧之宴	千禧之宴
千行	千行
千eye行情	千eye行情
南日本人行	南日本人行
南都行	南都行
博多体育场	博多体育场
bodoGing	bodoGing

名称	ID
syry Yuan	syry Yuan
古本市体育场	古本市体育场
名字代箱根	名字代箱根
名字屋顶行	名字屋顶行
名学馆	名学馆
名字百货店	名字百货店
和光	和光
和真	和真
四国金库	四国金库
四国行	四国行
四谷学院	四谷学院
回过寿司丸子	回过寿司丸子
回过寿命	回过寿命
土土地	土土地
城南	城南
埼玉行	埼玉行
夢庵	夢庵
大	大
大丸	大丸
大信局	大信局

名称	ID
大光行	大光行
大分行	大分行
大和	大和
大和优惠券	大和优惠券
大共立行	大共立行
大东行	大东行
大大	大大
大局	大局
大起水回过身寿司	大起水回过身寿司
大阪屋顶	大阪屋顶
大阪肉·	大阪肉·
大屋（大屋）	大屋（大屋）
天下一品	天下一品
天屋	天屋
天屋	天屋
天羅羅	天羅羅
SCHEARTROOSCAII	SCHEARTROOSCAII
学会	学会
崎太行了	崎太行了
崎崎行	崎崎行

名称	ID
店	店
富士	富士
富士	富士
富山第一行	富山第一行
富山行	富山行
家	家
寶龍	寶龍
寿命	寿命
小寿命	小寿命
小川	小川
小田急百货店	小田急百货店
simaltoror	simaltoror
居家屋酒田农场	居家屋酒田农场
山内农场	山内农场
山口行	山口行
山形屋	山形屋
山形行	山形行
山新	山新
山梨中央	山梨中央
山田	山田

名称	ID
山合同行	山合同行
岛	岛
岩井兑换券	岩井兑换券
岩手行	岩手行
岩田屋	岩田屋
根行	根行
川	川
市、学院、市、校	市、学院、市、校
常行	常行
平和堂	平和堂
平寿司	平寿司
幸运园	幸运园
行	行
庄	庄
建	建
堂	堂
得得	得得
大正行	大正行
快活了 CLUUB	快活了 CLUUB
情感	情感

名称	ID
爱媛行	爱媛行
爱知道行	爱知道行
成城石井	成城石井
文教堂	文教堂
新星堂	新星堂
新生行	新生行
日本亭	日本亭
日本便	日本便
日本行	日本行
日	日
日产（他是）	日产（他是）
日能调研	日能调研
日	日
日高屋	日高屋
早田田	早田田
明光校	明光校
明屋店	明屋店
明林堂店	明林堂店
星乃夜店	星乃夜店
時遊館	時遊館

名称	ID
衣料	衣料
未来屋顶店	未来屋顶店
本文	本文
林堂	林堂
村来了	村来了
来了亭	来了亭
東京行	東京行
東京必指教学院	東京必指教学院
東京流通	東京流通
東光	東光
東北金库	東北金库
東北行	東北行
东和行	东和行
東急便	東急便
東急便	東急便
東急百货店	東急百货店
東日本行	東日本行
東武	東武
东武百货店	东武百货店
DEGONYANCARMA	DEGONYANCARMA

名称	ID
東海東京兑换券	東海東京兑换券
東進ハイスクール	東進ハイスクール
東京星星校	東京星星校
degbong	degbong
屋	屋
松屋	松屋
松屋（松屋）	松屋（松屋）
松	松
栃木行	栃木行
fotiL'boild'bibly	fotiL'boild'bibly
光之光	光之光
梅的花	梅的花
我的花物店	我的花物店
森宿	森宿
屋顶店	屋顶店
業務スーパー	業務スーパー
市	市
横向幸运信心结合	横向幸运信心结合
横行	横行
武田	武田

名称	ID
武野行	武野行
日日BildSchool 5-daaSe	日日BildSchool 5-daaSe
Epetle taDainsquen	Epetle taDainsquen
海邦行	海邦行
金库	金库
行	行
河合	河合
洋服的青山	洋服的青山
太女郎了	太女郎了
淡路信誉金库	淡路信誉金库
清水行	清水行
上局	上局
滋补行情	滋补行情
肉肉	肉肉
肉肉	肉肉
肉肉	肉肉
肉肉	肉肉
熊本行	熊本行
熟成肉肉	熟成肉肉
家	家

名称	ID
玉屋	玉屋
屋顶	屋顶
“球” 行	“球” 行
LifeCair	LifeCair
田商店	田商店
田本	田本
白十字	白十字
白木屋	白木屋
百五行	百五行
百十四行	百十四行
dometli	dometli
眼睛市立体育场	眼睛市立体育场
水土产	水土产
神奈川行	神奈川行
神元	神元
神田一番寿司	神田一番寿司
福井行	福井行
福屋	福屋
福中中央行	福中中央行
Forg	Forg

名称	ID
Forg	Forg
福屋田百货店	福屋田百货店
福邦行	福邦行
优秀的英格兰学校	优秀的英格兰学校
秋吉	秋吉
秋田行	秋田行
水	水
笑笑	笑笑
第一步	第一步
第四个北越行	第四个北越行
波行	波行
邦行	邦行
地	地
行	行
老虎子房	老虎子房
wishseak'sheak's	wishseak'sheak's
士服	士服
美路	美路
群马行	群马行
人寿司居酒屋台台	人寿司居酒屋台台

名称	ID
肉体	肉体
肉汁子	肉汁子
肥行	肥行
Heai	Heai
自在空中间	自在空中间
花开学会	花开学会
啊家	啊家
英国 HUB	英国 HUB
英才学堂	英才学堂
English	English
necting	necting
屋顶与兵卫	屋顶与兵卫
薬ヒグチ	薬ヒグチ
薬マツモトキヨシ	薬マツモトキヨシ
局长	局长
王堂	王堂
屋	屋
西京行	西京行
西友记	西友记
西日本行情	西日本行情

名称	ID
西松屋	西松屋
西武	西武
西武百货店	西武百货店
局长	局长
和行	和行
資さんうどん	資さんうどん
赤	赤
feetli行	feetli行
boos	boos
coos'spedtaFrond	coos'spedtaFrond
近百货店	近百货店
学会	学会
道堀店	道堀店
酒	酒
重点 Gui	重点 Gui
野村兑换券	野村兑换券
金光商品	金光商品
金研究所	金研究所
銀のさら	銀のさら
座位————————————	座位————————————

名称	ID
座	座
鎌倉パスタ	鎌倉パスタ
长崎行	长崎行
长野行	长野行
校	校
西行	西行
西	西
阪急百货店	阪急百货店
阪神百货店	阪神百货店
阿波行	阿波行
青山	青山
青森行	青森行
静默中央	静默中央
静静金库	静静金库
静止行	静止行
靴チヨダ	靴チヨダ
下屋	下屋
校	校
月	月
来坊	来坊

名称	ID
Fooductay AI	Fooductay AI
食品	食品
子'王将军	子'王将军
老乃伊	老乃伊
香川行	香川行
馬	馬
maromacoCos	maromacoCos
駅レンタカー	駅レンタカー
河屋	河屋
高知行	高知行
魁力屋	魁力屋
魚べい	魚べい
平民	平民
取行	取行
族	族
屋	屋
鹿行	鹿行
生堂局	生堂局

要应用业务链筛选器，请使用将 APIs `businessChain`参数设置为所需链标识符的 Place。有关实施细节，请参阅 Amazon Location Service API 文档。

本地化和国际化

本地化 (L10n) 和国际化 (i18n) 是调整软件、内容或应用程序以适应不同语言、地区和视图的关键过程。以下是特定因素背景下的概述：

过滤器类型	地理编码	反向地理编码	自动完成	获取位置	搜索文本	在附近搜寻	建议
语言	支持	是	是	不适用	支持	是	是
政治观点	支持	是	是	不适用	支持	是	是
时区	支持	是	否	不适用	支持	是	是
音素	否	否	否	不适用	支持	是	是

语言

该功能允许您从 BCP47符合标准的代码中选择首选的响应语言。它根据名称变体检测查询语言，并使用首选语言处理不匹配的标记和模棱两可的情况。如果没有请求语言，Places API 会以官方国家/地区语言提供搜索结果，但它会优先考虑不同区域语言的区域。

作为一种备用策略，如果某些地址元素在请求的语言中不可用，Places 会以默认语言 APIs 返回地址。

政治观点

地点 APIs 允许您调整结果以反映当地的政治敏感性或地方政府的指导方针。默认情况下，Places APIs 会以国际视角看待有争议的政治边界。

- AR：阿根廷对南巴塔哥尼亚冰原和火地岛的看法，包括福克兰群岛、南格鲁吉亚岛和南桑威奇群岛。
- EG：埃及对比尔·塔维尔的看法。
- IN：印度对吉尔吉特-巴尔蒂斯坦的看法。
- KE：肯尼亚对伊莱米三角的看法。
- MA：摩洛哥对西撒哈拉的看法。
- RU：俄罗斯对克里米亚的看法。
- SD：苏丹对哈莱布三角的看法。

- RS：塞尔维亚对科索沃、武科瓦尔和萨伦格勒群岛的看法。
- SR：苏里南对 Courantyne Headwaters 和 Lawa Headwaters 的看法。
- SY：叙利亚对戈兰高地的看法。
- TR：土耳其对塞浦路斯和北塞浦路斯的看法。
- TZ：坦桑尼亚对马拉维湖的看法。
- UY：乌拉圭对 Rincon de Artigas 的看法。
- VN：越南对西沙群岛和南沙群岛的看法。

时区

该功能提供了其他时区信息。这包括时区名称和 UTC 偏移量。

音素

该功能提供了有关如何发音地址或地点的各个组成部分的其他音素信息。

联系方式和开放时间

Amazon Location Service 提供有关各个兴趣点的联系方式和营业时间的详细信息（POIs），使应用程序能够提供有关业务运营的全面信息。本节介绍联系方式和营业时间信息，以及如何有效地检索和解释这些详细信息。

Amazon Location Service 可以提供有关兴趣点 (POIs) 的联系方式和营业时间的结构化数据，使应用程序能够显示准确的信息。这些数据对于实时可用性或业务运营会影响客户体验的应用程序非常有价值。

过滤器类型	地理编码	反向地理编码	自动完成	获取位置	搜索文本	在附近搜寻	建议
联系人	否	否	否	是	是	是	否
开放时间	否	否	否	是	是	是	否

联系人详情

地点的联系方式包括结构化信息，例如电话号码、电子邮件地址和网站。这些详细信息适用于在原地数据中列出的企业。

电话号码

企业的主要联系电话。此字段可能包含国际拨号代码，以确保无论来电者身处何处，都能准确连接。

电子邮件地址

用于查询的主要联系人电子邮件。它可能适用于在公开列表中提供电子邮件地址的企业。

Website

指向官方网站的链接，为用户提供更多信息或访问在线预订或支持等服务。

开放时间

营业时间表示某个地点的正常营业时间，可让用户深入了解空房情况。这些信息对于用户需要知道企业何时开业或关闭的应用程序至关重要。

正常工作时间

标准每周开放时间，通常按每日范围提供（例如，周一至周五，上午 9 点至下午 5 点）。这些指示了通常的运行时间表。

特殊时段

节假日或特殊活动的特殊时段，可优先于正常工作时间。例如，可以指示假日时间或休息时间，以告知用户日程安排的临时变化。

立即打开

以当地时间为基准的营业地点当前是否开放的指示器。这些信息对于寻找在查询时处于营业状态的企业的用户很有帮助。

有关更多详情，请参阅 Amazon Location Service API 文档，探索检索联系方式和景点开放时间的选项。

其他功能

本节概述了该地点支持的其他功能 APIs。这些功能通过包含所需的详细信息（例如联系信息、可访问性和音量）来提供其他位置信息。您需要通过添加请求负载`AdditionalFeatures: [ <Value1>, <Value 2>, ..]`来请求其他功能。

> **Note**
>
> 查看the section called “地点定价”有关费用的更多信息。

有效值	地理编码	反向地理编码	自动完成	获取位置	搜索文本	在附近搜寻	建议
核心实例	默认	默认	是	默认	默认	默认	是
时区	支持	是	否	是	是	是	是
音素	否	否	否	是	是	是	是
联系人	否	否	否	是	是	是	否
开放时间	否	否	否	是	是	是	否
访问点	否	是	否	是	是	是	是
辅助地址	支持	是	否	否	否	否	否
十字路口	支持	是	否	否	否	否	否

要使用这些附加功能，请在地点 API 请求中设置`additionalFeatures`参数。有关详细信息，请参阅 Amazon Locat ion Service API 文档。

- 有关时区和音素的更多信息，请参阅。the section called “本地化和国际化”
- 有关联系方式和营业时间的更多信息，请参阅the section called “联系方式和开放时间”。

定义

时区

显示给定地点的时区。

音素

提供地名的语音表示，有助于发音。

联系人

提供联系方式，例如地点的电话号码和电子邮件地址。

开放时间

根据当地时间显示营业地点当前是否处于营业状态的指标。这些信息对于寻找在查询时处于营业状态的企业的用户很有帮助。

访问点

包括接入点信息，例如入口

辅助地址

包括主地址下的所有辅助地址。

十字路口

包括附近的十字路口。

IntendedUse

> **Note**
>
> 如果存储了结果，则将按更高的存储定价等级对它们进行计费。使用请求参数`IntendedUse`指定结果是供一次性使用还是用于存储。the section called “地点定价”要了解与存储结果相关的成本，请参阅。

当您调用地点 API 时，请根据结果的预期用途`Storage`，`IntendedUse`通过将值设置为`SingleUse`或来指定。如果要存储结果（即使是出于缓存目的），则必须选择存储选项，而不是一次性使用选项。

过滤器类型	地理编码	反向地理编码	自动完成	获取位置	搜索文本	在附近搜寻	建议
SingleUse	支持	是	是	是	是	是	是
存储	支持	是	否	是	是	是	否

地方 APIs

地点使应用程序能够搜索、查找和检索有关兴趣点、地址和特定位置的详细信息。这些功能通过提供上下文和改善搜索功能中的用户体验来增强基于位置的服务。

- 地理编码：将地址或地名转换为地理坐标（经度、纬度），支持需要 address-to-location转换才能进行制图和空间分析的应用程序。有关更多信息，请参阅 the section called “反向地理编码”。
- 反向地理编码：将地理坐标转换为最近的地址或地名，为位置提供背景信息。有关更多信息，请参阅the section called “反向地理编码”。
- 自动完成：为用户输入的文本建议潜在的补全内容，从而提高搜索输入的效率。有关更多信息，请参阅the section called “自动完成”。
- GetPlace：检索有关指定地点的详细信息，包括地址、联系方式和营业时间等属性。有关更多信息，请参阅the section called “GetPlace”。
- SearchNearby：在给定地理点的指定半径范围内查找适合 “我附近” 搜索的地方。有关更多信息，请参阅the section called “在附近搜寻”。
- SearchText：允许基于关键字或短语的文本搜索地点或兴趣点，非常适合按名称或描述查找地点。有关更多信息，请参阅the section called “搜索文本”。
- 建议：在用户键入时提供搜索词建议，增强搜索相关性和用户体验。有关更多信息，请参阅the section called “建议”。

下表列出了一些最好通过 Places 解决的业务用例 APIs。

地点用例

以下部分介绍了一些最好通过 Places 解决的业务用例 APIs。

业务需求	有用的 API	示例
在应用程序中添加邻里商业信息 还会返回业务信息，例如类别和食物类型。	`SearchNearBy`	

业务需求	有用的 API	示例
邻近搜索连锁企业 还会返回业务信息，例如类别、食物类型和接入点。	搜索文字建议	
按电话名搜索 还会返回业务信息，例如类别、食物类型和接入点。	搜索文字建议	
按地点名称、POI、类别搜索 还会返回业务信息，例如类别、食物类型和接入点。	搜索文字建议	the section called “使用名称搜索地点、POI 或企业”
将建议作为用户密钥查询进行预测 还会返回业务信息，例如类别、食物类型和接入点。	建议	the section called “根据输入预测建议”
自动完成或建议在结账页面自动填充地址 对地址进行标准化和标准化。	自动完成，建议	the section called “如何填写地址”
将特定地址转换为经度和纬度坐标 对地址进行标准化和标准化。	地理编码	
将经度和纬度坐标转换为相应的地址 对地址进行标准化和标准化。	反向地理编码	the section called “对某个位置进行反向地理编码”
获取城市的时区 支持 UTC 偏移量和时区名称。	地理编码	

业务需求	有用的 API	示例
获取经度和纬度的时区 支持 UTC 偏移量和时区名称。	反向地理编码	
按地点 ID 获取地点 返回业务信息，例如类别、联系人、营业时间和接入点。	获取位置	the section called “获取 a 的结果 PlaceId”
获取景点的姓名、联系方式和营业时间 返回业务信息，例如类别、食物类型和接入点。	搜索文本、在附近搜索、建议	
为地名提供地点类型 支持预过滤。	地理编码自动完成	
为 latitude/longitude 坐标提供地点类型 支持预过滤。	反向地理编码	
为地址或 POI 提供 POI 类别（例如医院、商店、博物馆等 500 个） 支持预过滤。	搜索文本、在附近搜索、建议	
为完成或预测添加提前键入搜索行为 支持经济实惠、仅限标签和地址组件。	自动完成建议	the section called “根据输入预测建议”

业务需求	有用的 API	示例
在地图上可视化地点搜索和/或地理编码结果 所有 APIs 返回地理坐标，但自动完成除外。	`GetTile`以及`GetStyleDescriptor` 带有地点 API 的渲染引擎 (MapLibre)	
增强、清理、标准化和标准化您的地址数据库 支持地址标签、组件、时区等。	地理编码，反向地理编码	

地理编码

> **Note**
>
> 存储的结果会产生更高的存储价格。使用`IntendedUse`参数表示一次性使用或存储。the section called “地点定价”要了解存储结果的成本影响，请参阅。

地理编码可将文本地址或地名转换为地理坐标，以及详细的地址组成部分和其他信息。此 API 支持灵活查询，包括自由格式文本和包含街道名称、邮政编码和地区的结构化查询。可选功能包括时区数据和政治观点调整。

使用案例

- 向客户地址添加补充数据：通过包括邮政编码、坐标以及利用永久存储来支持明智的业务和营销决策来改善地址记录。
- 地址数据标准化：在数据管道或批处理中应用地理编码来标准化地址数据，利用永久存储进行持续参考。
- 确定时区：确定城市或地址的时区，以便为旅行、日程安排和开具发票等应用程序提供准确的时间戳。

理解请求

该请求接受可选参数AdditionalFeatures，例如BiasPositionFilter、和，以优化搜索结果。其他选项（例如LanguageMaxResultsPoliticalView、和）可进一步自定义响应。必需的参数是Query，它可以作为自由格式文本提供，也可以以结构化形式提供。QueryComponents有关更多信息，请参阅地理编码 API 参考。

该请求包含以下关键参数：

身份验证

对于身份验证，如果正在使用其他方法，则该Key参数是可选的。

- Key: 身份验证的可选参数。

有关更多信息，请参阅 身份验证。

查询和偏差

用于查询结果和对结果进行地理偏差的参数。

- QueryText：用于搜索位置的自由格式文本查询。
- QueryComponents：用于精确搜索的地址号码、国家、地区或邮政编码等结构化组件。
- BiasPosition：搜索结果偏向的地理位置。

有关更多信息，请参阅 the section called “查询和偏差”。

精炼结果

应用过滤器来优化结果。

- Filter：筛选以包括特定的国家或地点类型。有关更多信息，请参阅 the section called “过滤”。

国际化和本地化

为本地化结果指定语言并运用政治观点。

- Language: 结果所用语言。
- PoliticalView: 采用反映领土主张的政治观点。

有关更多信息，请参阅 the section called “本地化和国际化”。

其他功能

索取其他信息，例如时区详情。

- AdditionalFeatures：可以选择请求其他详细信息，例如时区。有关更多信息，请参阅 the section called “其他功能”。

了解回应

响应中提供的ResultItems，其中包含详细的位置数据Address，例如PlaceIdPosition、、和其他相关属性。还可能包括其他功能，例如每个查询组件的TimeZone信息或匹配分数。每个都ResultItem表示基于指定请求参数的匹配位置或地理编码结果。有关更多详细信息，请参阅地理编码 API 参考。

响应包括以下关键参数：

地址和相关详情

有关返回位置的详细信息，包括地址组成部分。

- Address: 完整地址，包括国家、地区、邮政编码和街道详情。
- PostalCodeDetails：与邮政编码相关的其他信息。
- StreetComponents: 街道详情，包括基本名称和类型。

地点类型和类别

有关返回的地点类型和类别的信息。

- Categories：描述地点的类别列表，例如餐厅、学校。
- PlaceType：指定地点的类型，例如城市、地址或区域。

结果分析

分数表示结果与输入查询的匹配程度。

- MatchScores: 匹配精度的分数。

其他详细信息

根据需要包括与位置相关的额外信息。

- AccessPoints：代表接入点的地理坐标。
- TimeZone：该地点的时区信息。

反向地理编码

> **Note**
>
> 如果存储了结果，则将按更高的存储定价等级进行计费。使用IntendedUse参数指定结果是供一次性使用还是用于存储。the section called “地点定价”有关与存储结果相关的成本影响，请参阅。

反向地理编码将地理坐标转换为人类可读的地址或地点。它提供详细的地址组成部分、地点类型、类别和街道信息，并提供基于地点类型的筛选选项以优化结果。API 还可以包括其他功能，例如时区信息和政治观点调整。

使用案例

- 向客户位置数据添加补充数据：通过包括邮政编码、坐标以及利用持久存储来支持明智的业务和营销决策，改善位置数据记录。
- 位置数据标准化：在数据管道或批处理中应用地理编码来标准化位置数据，利用永久存储进行持续参考。
- 确定时区：确定城市或地址的时区，以便为旅行、日程安排和开具发票等应用程序提供准确的时间戳。

理解请求

反向地理编码 API 请求接受必需参数和可选参数的组合以自定义结果。必需的参数QueryPosition（经度和纬度）指定要进行反向地理编码的坐标。可选参数包括AdditionalFeatures用于获取额外数据、Filter优化结果和限制MaxResults返回结果数量的参数。其他选项Language，例如PoliticalView、和，IntendedUse允许进一步自定义。有关更多详细信息，请参阅反向地理编码 API 参考。

该请求接受以下关键参数：

身份验证

如果使用其他身份验证方法，则该Key参数是可选的。

- Key: 身份验证的可选参数。

有关更多信息，请参阅 身份验证。

查询

用于定义地理位置和搜索半径的参数。

- `QueryPosition`：指定反向地理编码请求的经度和纬度。
- `QueryRadius`：定义围绕坐标的搜索半径。

有关更多信息，请参阅 the section called “查询和偏差”。

完善结果

应用筛选条件将搜索结果限制为特定的国家/地区或地点类型。

- `Filter`：根据国家或地点类型筛选结果。

有关更多详细信息，请参阅 the section called “过滤”。

国际化和本地化

自定义语言和将政治观点应用于结果的选项。

- `Language`：指定结果的语言。
- `PoliticalView`: 采用反映领土主张的政治观点。

有关更多信息，请参阅 the section called “本地化和国际化”。

其他功能

请求额外数据，例如时区信息。

- `AdditionalFeatures`：可以选择请求其他数据，例如时区详细信息。

有关更多信息，请参阅 the section called “其他功能”。

了解回应

来自反向地理编码 API 的响应提供了有关指定坐标处位置的详细信息。该`ResultItems`数组包括地点对象列表，每个对象都包含地址详细信息、类别和地理位置。诸如`TimeZone`、`FoodTypes`、和之类的其他数据`PostalCodeDetails`提供了进一步的背景信息。响应还包括用于映射的边界框信息以及与查询位置的距离。有关更多详细信息，请参阅反向地理编码 API 参考。

响应包含以下关键数据：

地址和相关详情

返回地点的详细地址信息。

- Address: 完整的地址信息，包括国家、地区、邮政编码和街道详情。
- PostalCodeDetails：与邮政编码相关的其他详细信息，例如分类和邮政当局。
- StreetComponents：有关街道的其他详细信息，包括基本名称和类型。

地点类型和类别

描述返回的地点的类型和类别。

- Categories：描述地点的类别，例如餐厅或学校。
- PlaceType：指定地点的类型，例如城市、地址或地区。

结果分析

有关每个结果与输入查询的匹配程度的信息。

- MatchScores：表示每个匹配项与输入查询的准确性的分数。

其他详细信息

提供与位置相关的额外数据。

- TimeZone：该地点的时区信息。
- AccessPoints：表示位置入口点的地理坐标。

自动完成

> **Note**
>
> 默认情况下，“自动完成” 仅返回ID和Title字段，这提供了一种经济实惠的选项。通过将设置为additionalFeatures，可以请求其他地址组成部分和亮点Core。the section called “地点定价”有关与存储结果相关的费用详情，请参阅。请注意，自动完成功能不会返回地理编码，而且此功能目前在日本不可用。

Autocomplete API 会根据部分输入在用户键入时完成潜在的地点和地址。它通过在几次击键后使用有效的地址信息完成查询，从而提高了地址输入的效率和准确性。此外，自动完成功能支持基于地理位置、国家/地区或特定地点类型的结果筛选，并且可以使用语言和政治观点等可选参数进行定制。

使用案例

- 增强结账体验：当客户在网站或应用程序上输入位置时，提供实时地址填写功能。确保配送或取件地点与已知地址相匹配，以简化地址验证、减少错误并营造无缝的结账体验。

- Su@@ pport customer services：为客户服务（例如联络中心或紧急服务）提供实时地址建议，简化定位准确地址的流程，并通过缩短获取正确地址信息所需的时间来提高用户满意度。

理解请求

自动完成请求使用可选参数来优化搜索结果。如需了解更多详情，请参阅自动补全 API 参考。

该请求接受以下关键参数：

身份验证

如果使用其他身份验证方法，则该Key参数是可选的。

- Key: 身份验证的可选参数。

有关更多信息，请参阅 身份验证。

查询

定义结果建议的部分文本查询和地理偏差。

- QueryText：用于查找位置结果的自由格式文本查询，此参数是必需的。
- BiasPosition：建议最接近指定经度和纬度坐标的结果，此参数是可选的。

有关更多信息，请参阅 the section called “查询和偏差”。

优化结果

应用筛选条件将搜索结果限制为特定国家/地区或地点类型。

- Filter：允许根据指定的国家或地点类型进行筛选。

有关更多详细信息，请参阅 the section called “过滤”

了解回应

Autocomplete API 返回的结果为ResultItems，每个结果都表示基于请求参数的匹配位置。如需了解更多详情，请参阅自动补全 API 参考。

响应包含以下关键数据：

地址和相关详情

包含每个位置的详细地址组成部分。

- `Address`：提供完整的地址组件，包括街道、邮政编码和国家。

结果分析

为与输入查询相关的每个结果提供分析数据。

- `Distance`：从指定偏置位置到结果的距离。
- `Highlights`：为清楚起见，在结果中突出显示查询的各个部分。

GetPlace

> **Note**
>
> 如果存储了结果，则将按更高的存储定价等级进行计费。使用`IntendedUse`参数指定结果是供一次性使用还是用于存储。the section called “地点定价”有关与存储结果相关的费用详情，请参阅。

GetPlace 使用地点的唯一`PlaceId`信息检索有关该地点的详细信息。该请求支持其他功能的可选参数，例如时区数据或音素以及语言本地化。API 会返回地址、坐标、业务详情、联系方式和时区信息等信息。

使用案例

- 刷新存储的数据：定期刷新存储的数据，以确保您的应用程序提供准确、最新的信息。使用 GetPlace API 更新有关企业和场所的详细信息，确保用户获得最新信息。例如，刷新当地餐馆或商店的营业时间、联系方式和用户评论，以增强用户信任和应用程序功能。
- 根据@@ 预先输入建议检索地点详情：将 GetPlace API 与自动完成功能或建议功能集成，从而改善用户体验。当用户键入地址、公司名称或兴趣点时，建议会提供相关地点，并在做出选择后 GetPlace 可以访问更详细的信息。

理解请求

GetPlace API 请求需要特定的参数，例如`PlaceId`，并且支持用于其他功能或语言本地化的可选参数。这些参数有助于根据语言、预期用途和地理环境定制结果。有关更多详细信息，请参阅 GetPlace API 参考。

该请求接受以下关键参数：

身份验证

如果使用其他身份验证方法，则该Key参数是可选的。

- Key: 用于授权的 API 密钥。

有关更多信息，请参阅 身份验证。

查询

定义用于检索地点信息的唯一标识符。

- PlaceId：地点的唯一标识符。（必填）

有关更多信息，请参阅 the section called “查询和偏差”。

国际化和本地化

自定义语言和将政治观点应用于结果的选项。

- Language：指定结果的语言。
- PoliticalView: 采用反映领土主张的政治观点。

有关更多信息，请参阅 the section called “本地化和国际化”。

其他功能

请求其他数据，例如时区详细信息。

- AdditionalFeatures：可以选择请求其他信息，例如时区或电话详情。

有关更多信息，请参阅 the section called “其他功能”。

了解回应

来自 GetPlace API 的响应包括地址、坐标和业务详细信息等基本数据，以及基于请求参数的其他数据。还可以提供诸如访问限制、时区和发音的音素详细信息之类的信息。有关更多详细信息，请参阅 GetPlace API 参考。

响应包含以下关键数据：

地址和相关详情

提供该地点的完整地址详情。

- Address: 完整的地址信息，包括国家、地区和邮政编码。

- PostalCodeDetails：有关邮政编码和当局的更多信息。

地点类型和类别

描述地点的类型和类别。

- BusinessChains: 有关任何相关业务链的信息。
- Categories：该地点所属的类别，例如食品或零售。
- PlaceType: 地点的类型，例如兴趣点或地点。

其他细节

提供请求中指定的有关该地点的其他数据。

- Contacts: 联系方式，包括电话号码、电子邮件和网站。
- OpeningHours: 营业时间或访问时间。
- AccessPoints：接入点的地理坐标（经度和纬度）。
- Phonemes: 各种地址组成部分的发音。
- TimeZone：时区信息，包括偏移量。

在附近搜寻

> **Note**
>
> 如果存储了结果，则将按更高的存储定价等级进行计费。使用IntendedUse参数指定结果是供一次性使用还是用于存储。the section called “地点定价”有关与存储结果相关的成本影响，请参阅。

Search Neargy API 允许开发者从中心坐标查询指定半径内的兴趣点。API 通过可选筛选条件提供地点结果，包括类别、业务链、食物类型等。回复详情包括地名、地址、电话、类别、食物类型、联系方式和营业时间。其他功能，例如音素、时区等，可根据请求的参数提供。

使用案例

- 房地产应用：为潜在的购房者或租房者提供房产附近的便利设施和服务清单，例如学校、医院和购物中心。
- 本地业务发现：使用户能够在距离当前位置的特定半径范围内找到企业、餐馆或商店。例如，送餐应用程序可以根据实时位置显示附近的餐馆。

- 旅行和旅游应用程序：帮助旅行者探索其所在地或目的地附近的景点、地标、酒店和服务。旅行应用程序可能会推荐附近的景点，例如博物馆、公园和历史遗迹。
- 紧急服务：在紧急情况下帮助用户找到最近的医院、警察局或药房，这对于专注于健康、安全或灾难响应的应用程序特别有用。
- 运输和物流：帮助用户找到最近的加油站、电动汽车充电点或公共交通站点。叫车服务可以显示附近的可用司机。

理解请求

Search Nearch Nearby API 请求包括必需的参数（例如`QueryPosition`筛选条件），以优化搜索。其他选项，例如`Language`和`PoliticalView`，可根据特定需求自定义结果。如需了解更多信息，请参阅 “附近搜索 API 参考”。

该请求包含以下关键参数：

身份验证

如果使用其他身份验证方法，则该`Key`参数是可选的。

- `Key`: 用于授权的 API 密钥。

有关更多信息，请参阅 身份验证。

查询

定义查询的位置和搜索半径。

- `QueryPosition`：指定经度和纬度坐标。
- `QueryRadius`：定义围绕指定坐标的搜索半径。

有关更多信息，请参阅 the section called “查询和偏差”。

精炼结果

筛选结果以缩小搜索条件范围。

- `Filter`：应用类别、业务链或食物类型等筛选条件。

有关更多详细信息，请参阅 the section called “过滤”

国际化和本地化

自定义语言和将政治观点应用于结果的选项。

- `Language`：指定结果的语言。
- `PoliticalView`: 采用反映领土主张的政治观点。

有关更多信息，请参阅 the section called “本地化和国际化”。

其他功能

请求其他数据，例如时区信息。

- `AdditionalFeatures`：可以选择请求其他详细信息，例如时区信息。

有关更多信息，请参阅 the section called “其他功能”。

了解回应

Search Nearby API 的响应提供了与查询相匹配的详细结果，包括位置、地址详情、业务链、联系人、音素、时区和营业时间。如需了解更多详情，请参阅 “附近搜索 API 参考”。

响应包含以下关键数据：

地址和相关详情

返回地点的完整地址信息。

- `Address`：包括完整的地址详情，例如乡村和街道。

地点类型和类别

描述返回的地点的类型和类别。

- `Categories`：描述地点的类别，例如餐厅或学校。
- `FoodTypes`: 餐厅或餐馆提供的食物种类。
- `BusinessChains`：表示任何关联的业务链。

其他细节

根据要求提供与该地点相关的其他信息。

- `Contacts`: 列出电子邮件、电话号码和网站。
- `OpeningHours`：指定地点的营业时间。
- `Phonemes`：提供地址组成部分的语音表示。
- `TimeZone`：包括该地点的时区信息。

搜索文本

> **Note**
>
> 如果存储了结果，则将按更高的存储定价等级进行计费。使用`IntendedUse`参数指定结果是供一次性使用还是用于存储。the section called “地点定价”有关与存储结果相关的成本影响，请参阅。

Search Text 允许您使用单个自由格式的文本输入来查询位置数据，返回带有可选过滤器（如国家、地理圈或边界框）的地点结果。该 API 还支持按从建议 API `QueryId` 返回的内容进行搜索。回复包括地名、地址、电话号码、类别、食物类型、联系方式和营业时间等详细信息。根据指定的参数，也可以使用音素、时区等可选功能。

使用案例

- 通过文字进行邻里搜索：帮助买家找到最近的相关地点。客户可以找到最近的地址、企业或景点，例如博物馆、商店或公园。
- 业务发现：允许用户通过输入 “意大利餐厅” 或 “咖啡店” 等关键字来查找特定的企业或地点类型。此功能对于送餐或餐厅推荐应用程序特别有用。
- 在旅行应用程序中搜索地点：使用户能够使用自由格式的文本（例如 “埃菲尔铁塔附近的酒店”）搜索地点、地标或企业，从而快速收到相关建议。
- 教育机构搜索：允许用户搜索学校、学院或培训中心（例如，“小学” 或 “大学”），这是家长或学生的有用工具。

理解请求

搜索文本 API 请求允许使用各种参数来优化搜索。可选参数启用其他功能、位置偏差、筛选标准、语言首选项和结果限制。如需了解更多详情，请参阅搜索文本 API 参考。

该请求包含以下关键参数：

身份验证

如果使用其他身份验证方法，则该`Key`参数是可选的。

- `Key`: 用于授权的 API 密钥。

有关更多信息，请参阅 身份验证。

查询

用于指定文本搜索和位置偏差的参数。

- `QueryText`：用于搜索地点的自由格式文本。
- `QueryId`：允许完成来自建议 API 的建议查询。
- `BiasPosition`：优先考虑接近特定经度和纬度的结果。

有关更多信息，请参阅 the section called “查询和偏差”。

精炼结果

筛选结果以缩小搜索条件范围。

- `Filter`：允许按边界框或圆形区域进行筛选，以限制搜索结果。

有关更多详细信息，请参阅 the section called “过滤”。

国际化和本地化

自定义语言和将政治观点应用于结果的选项。

- `Language`：指定结果的语言。
- `PoliticalView`: 采用反映领土主张的政治观点。

有关更多信息，请参阅 the section called “本地化和国际化”。

其他功能

请求额外数据，例如时区信息。

- `AdditionalFeatures`：可以选择请求其他数据，例如时区详细信息。

有关更多信息，请参阅 the section called “其他功能”。

了解回应

Search Text API 的响应包括与查询相匹配的详细结果，包括位置、地址详情、业务链、联系人、音素、时区和营业时间等信息。如需了解更多详情，请参阅搜索文本 API 参考。

响应包含以下关键数据：

地址和相关详情

返回地点的完整地址信息。

- `Address`：包括国家、街道和其他地址详情。

地点类型和类别

描述地点的类型和类别。

- `Categories`：描述地点的类别列表，例如餐厅或学校。
- `PlaceType`：指定地点的类型，例如城市、地址或地区。
- `BusinessChains`：表示任何关联的业务链。

其他细节

根据要求提供与该地点相关的其他数据。

- `Contacts`: 提供电子邮件、电话号码和网站。
- `OpeningHours`: 该地点的营业时间。
- `AccessPoints`: 与该地点相关的地理坐标。
- `Phonemes`: 地址组成部分的语音表示。
- `TimeZone`：时区信息，包括偏移量。

建议

> **Note**
>
> 默认情况下，Suggest API 仅返回`ID`和`Title`字段，这提供了一种经济实惠的选项。通过将设置为`additionalFeatures`，可以请求其他地址组成部分和亮点`Core`。the section called “地点定价”有关与存储结果相关的成本影响，请参阅。

Suggest 根据用户输入或上下文（例如相关地点、兴趣点、查询词或搜索类别）提供预测或推荐。它可以帮助用户根据不完整或拼写错误的输入来查找地点、兴趣点或识别后续查询。API 会返回可能的匹配项或细化项列表，该列表可用于制定更准确的查询。用户可以选择适当的建议进行进一步搜索。该 API 支持按位置和其他属性筛选结果，并提供音素和时区等其他功能。该响应包括精确的查询术语和详细的地点信息。

使用案例

- 叫车服务：提供实时建议，以快速填写接送地点的地址，确保准确性和更快的乘车预订。
- 旅行和导航服务：提供位置或地标的实时预测，即使是拼写错误或部分输入的术语，例如 “埃菲尔铁塔” 的 “埃菲尔铁塔”。该API完善了建议，以提供附近的相关景点，从而帮助用户准确地定位地点。

- 餐厅搜索帮助：预测用户对餐厅的兴趣并推荐附近的餐饮选择，从而增强搜索体验。

理解请求

Suggest API 请求使用参数根据用户输入生成建议。可选参数允许使用位置偏差和筛选条件来优化搜索结果。如需了解更多详情，请参阅建议 API 参考。

该请求包含以下关键参数：

身份验证

如果使用其他身份验证方法，则该Key参数是可选的。

- Key: 用于授权的 API 密钥。

有关更多信息，请参阅 身份验证。

查询

定义自由文本搜索和位置偏差。

- QueryText：用于生成建议的自由格式文本。（必填）
- BiasPosition：优先考虑靠近特定经度和纬度的建议。

有关更多信息，请参阅 the section called “查询和偏差”。

完善结果

筛选结果以缩小搜索条件范围。

- Filter：允许按边界框或圆形区域进行筛选，以限制搜索结果。

有关更多详细信息，请参阅 the section called “过滤”

国际化和本地化

自定义语言和将政治观点应用于结果的选项。

- Language：指定结果的语言。
- PoliticalView: 采用反映领土主张的政治观点。

有关更多信息，请参阅 the section called “本地化和国际化”。

其他功能

请求额外数据，例如时区信息。

- AdditionalFeatures：可以选择请求其他详细信息，例如时区或语音数据。

有关更多信息，请参阅 the section called “其他功能”。

限制结果

设置对返回的结果或查询优化数量的限制。

- MaxQueryRefinements：限制返回的查询优化术语的数量。
- MaxResults：限制返回的建议数量。

了解回应

响应会根据输入查询提供建议的地址或地点，包括位置、地址详情、业务链、联系人、音素、时区和营业时间等属性。API 会返回ResultItems，表示完成输入查询时可能的匹配项。有两种类型的结果，按SuggestResultItemType以下方式标识：类型为Query建议后续类别的结果或链式查询，通过将传递给 SearchText API，可以使用它们QueryID来获取指定类别的重点结果。类型的结果Place提供最终结果，其中包含地址和有关该地点的其他信息。如需了解更多详情，请参阅建议 API 的 API 参考。

响应包含以下关键数据：

结果分析

提供有关完善输入查询的信息。

- QueryRefinements：可用于完善搜索查询的术语。
- Highlights：突出显示地址或标题中与查询相匹配的部分。

地点类型和类别

描述地点的类型和类别。

- Categories：描述地点的类别，例如餐厅或学校。
- PlaceType：指定地点的类型，例如城市、地址或地区。
- BusinessChains：表示任何关联的业务链。

其他详细信息

请求中指定的有关该地点的其他信息。

- Contacts: 提供电子邮件、电话号码和网站。
- OpeningHours: 该地点的营业时间。

- `AccessPoints`: 与该地点相关的地理坐标。
- `Phonemes`：地址组成部分的语音表示。
- `TimeZone`：时区信息，包括偏移量。

操作方法

本节提供 step-by-step有关如何使用地点执行特定任务的说明 APIs。这些指南旨在帮助您快速有效地实现关键功能。每个主题都侧重于一个不同的用例，确保您具备将 Places API 功能集成到应用程序中所需的实践知识。

示例

- 如何使用地理编码
- 如何使用 ReverseGeocode
- 如何使用自动完成功能
- 如何使用 Get Place
- 如何使用 SearchNearby
- 如何使用 SearchText
- 如何使用 “建议”

如何使用地理编码

本节包含有关如何使用地理编码 APIs的各种操作指南和示例。

示例

- 如何对行政区和邮政区域进行地理编码
- 如何对地址进行地理编码
- 如何对兴趣点进行地理编码
- 如何使用邮政编码对地址号码进行地理编码
- 如何使用地理空间上下文进行地理编码
- 如何使用过滤器进行地理编码
- 如何根据城市的时区进行地理编码
- 如何使用特定语言进行地理编码

- 如何获取辅助地址
- 如何获得十字路口

如何对行政区和邮政区域进行地理编码

Geocode API 允许您使用查询文本输入对地理区域进行地理编码，例如国家、地区（州或省）或城市的名称。API 响应包括位置详细信息，例如地理坐标、地图可视化的边界框以及表示结果与查询相关性的匹配分数。

潜在用例

- 获取行政区域的坐标：在其他地方使用坐标作为偏差位置或中心 APIs。
- 在@@ 地图上可视化信息：地理编码坐标可用于在地图上直观地显示数据。

示例

对国家进行地理编码

Sample request

```
{
  "QueryText": "Canada"
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId":"<Redacted>",
            "PlaceType": "Country",
            "Title": "Canada",
            "Address": {
                "Label": "Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                }
            },
```

```
            "Position": [-75.69122, 45.42177],
            "MapView": [-141.00271, 41.67659, -52.61901, 83.11062],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Address": { "Country": 1 }
                }
            }
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
  "QueryText": "Canada"
}'
```

AWS CLI

```
aws geo-places geocode --key ${YourAPIKey} --query-text "Canada"
```

对区域进行地理编码

Sample request

```
{
  "QueryText": "BC"
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "Region",
            "Title": "BC, Canada",
```

```
            "Address": {
                "Label": "BC, Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                }
            },
            "Position": [-123.36445, 48.42854],
            "MapView": [-139.04941, 48.22478, -114.05201, 60.00043],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Address": { "Region": 1 }
                }
            }
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
  "QueryText": "BC"
}'
```

AWS CLI

```
aws geo-places geocode --key ${YourAPIKey} --query-text "BC"
```

对城市进行地理编码

Sample request

```
{
```

```
  "QueryText": "Vancouver"
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "Locality",
            "Title": "Vancouver, BC, Canada",
            "Address": {
                "Label": "Vancouver, BC, Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "PostalCode": "V5Y"
            },
            "Position": [-123.11336, 49.26038],
            "MapView": [-123.26754, 49.19891, -123.02301, 49.33557],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Address": { "Locality": 1 }
                }
            }
        }
    ]
}
```

cURL

```
curl --request POST \
```

```
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
  "QueryText": "Vancouver"
}'
```

AWS CLI

```
aws geo-places geocode --key ${YourAPIKey} --query-text "Vancouver"
```

对邮政编码进行地理编码

您可以对邮政编码进行地理编码。IncludePlaceTypes与一起使用["PostalCode"]可获得更精确的结果。

Sample request

```
{
  "QueryText": "800006",
  "Filter": { "IncludePlaceTypes": ["PostalCode"] }
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PostalCodeArea",
            "Title": "800006, Patna, Bihar, India",
            "Address": {
                "Label": "800006, Patna, Bihar, India",
                "Country": {
                    "Code2": "IN",
                    "Code3": "IND",
                    "Name": "India"
                },
                "Region": {
                    "Code": "BR",
                    "Name": "Bihar"
                },
```

```
                "SubRegion": { "Name": "Patna" },
                "Locality": "Patna",
                "PostalCode": "800006"
            },
            "Position": [85.18048, 25.61532],
            "MapView": [85.16599, 25.60054, 85.19103, 25.6221],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Address": { "PostalCode": 1 }
                }
            }
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
  "QueryText": "800006",
  "Filter": { "IncludePlaceTypes": ["PostalCode"] }
}'
```

AWS CLI

```
aws geo-places geocode --key ${YourAPIKey} --query-text "800006" --filter
 '{"IncludePlaceTypes": ["PostalCode"]}'
```

开发者小贴士

使用IncludeCountries和之类的过滤器IncludePlaceTypes可获得更有针对性的结果。例如，为了确保来自美国温哥华的结果，请设置"IncludeCountries": ["USA"]。有关更多详细信息，请参阅 。

```
{
  "QueryText": "Vancouver",
  "Filter": { "IncludeCountries": ["USA"] }
}
```

如何对地址进行地理编码

Geocode API 允许您对特定的点地址、插值地址或街道进行地理编码。API 响应包含位置信息，包括地理坐标和匹配分数，我用这些信息表示结果与查询对齐的准确程度。

潜在用例

- 干净的地址数据库：通过识别和更正地址记录中的错误来提高数据质量。
- 标准化和标准化地址：确保各数据集的地址格式保持一致，以提高数据互操作性。
- 使用其他信息丰富地址：将地理坐标和其他相关详细信息添加到地址记录中，以支持基于位置的分析和见解。

示例

使用查询文本

Sample request

```
{
  "QueryText": "510 W Georgia St, Vancouver, BC"
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointAddress",
            "Title": "510 W Georgia St, Vancouver, BC V6B 0M3, Canada",
            "Address": {
                "Label": "510 W Georgia St, Vancouver, BC V6B 0M3, Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
```

```
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "District": "Downtown Vancouver",
                "PostalCode": "V6B 0M3",
                "Street": "W Georgia St",
                "StreetComponents": [
                    {
                        "BaseName": "Georgia",
                        "Type": "St",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "W",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "510"
            },
            "Position": [
                -123.11694,
                49.28126
            ],
            "MapView": [
                -123.11832,
                49.28036,
                -123.11556,
                49.28216
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -123.11656,
                        49.28151
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Address": {
                        "Region": 1,
                        "Locality": 1,
                        "Intersection": [
```

```
                            1
                        ],
                        "AddressNumber": 1
                    }
                }
            }
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{"QueryText": "510 W Georgia St, Vancouver, BC"}'
```

AWS CLI

```
aws geo-places geocode --key ${YourAPIKey} --query-text "510 W Georgia St,
 Vancouver, BC"
```

使用查询组件

Sample request

```
{
  "QueryComponents": {
    "AddressNumber": "510",
    "Locality": "Vancouver",
    "Region": "BC",
    "Country": "Canada",
    "Street": "Georgia"
  }
}
```

Sample response

```
{
    "ResultItems": [
        {
```

```
            "PlaceId": "<Redacted>",
            "PlaceType": "PointAddress",
            "Title": "510 W Georgia St, Vancouver, BC V6B 0M3, Canada",
            "Address": {
                "Label": "510 W Georgia St, Vancouver, BC V6B 0M3, Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "District": "Downtown Vancouver",
                "PostalCode": "V6B 0M3",
                "Street": "W Georgia St",
                "StreetComponents": [
                    {
                        "BaseName": "Georgia",
                        "Type": "St",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "W",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "510"
            },
            "Position": [
                -123.11694,
                49.28126
            ],
            "MapView": [
                -123.11832,
                49.28036,
                -123.11556,
                49.28216
            ],
            "AccessPoints": [
```

```
                {
                    "Position": [
                        -123.11656,
                        49.28151
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 0.99,
                "Components": {
                    "Address": {
                        "Country": 1,
                        "Region": 1,
                        "Locality": 1,
                        "Intersection": [
                            0.78
                        ],
                        "AddressNumber": 1
                    }
                }
            }
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "InterpolatedAddress",
            "Title": "510 E Georgia St, Vancouver, BC V6A 1Z9, Canada",
            "Address": {
                "Label": "510 E Georgia St, Vancouver, BC V6A 1Z9, Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "District": "Strathcona",
                "PostalCode": "V6A 1Z9",
                "Street": "E Georgia St",
```

```
                "StreetComponents": [
                    {
                        "BaseName": "Georgia",
                        "Type": "St",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "E",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "510"
            },
            "Position": [
                -123.0932,
                49.27829
            ],
            "MapView": [
                -123.09458,
                49.27739,
                -123.09182,
                49.27919
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -123.0932,
                        49.27842
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 0.99,
                "Components": {
                    "Address": {
                        "Country": 1,
                        "Region": 1,
                        "Locality": 1,
                        "Intersection": [
                            0.78
                        ],
                        "AddressNumber": 1
                    }
                }
            }
```

```
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{"QueryComponents": {"AddressNumber": "510", "Locality": "Vancouver",
 "Region": "BC", "Country": "Canada", "Street": "Georgia"}}'
```

AWS CLI

```
./aws geo-places geocode --key ${YourAPIKey} --query-components '{
"AddressNumber" : "510",
"Locality": "vancouver",
"Region": "BC",
"Country": "Canada",
"Street": "Georgia"}'
```

使用混合查询

Sample request

```
{
  "QueryText": "W. 6th St",
  "QueryComponents": {
    "AddressNumber": "415",
    "Locality": "Vancouver"
  }
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointAddress",
            "Title": "415 W 6th St, Vancouver, WA 98660-3375, United States",
```

```
            "Address": {
                "Label": "415 W 6th St, Vancouver, WA 98660-3375, United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "Clark"
                },
                "Locality": "Vancouver",
                "District": "Esther Short",
                "PostalCode": "98660-3375",
                "Street": "W 6th St",
                "StreetComponents": [
                    {
                        "BaseName": "6th",
                        "Type": "St",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "W",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "415"
            },
            "Position": [
                -122.67543,
                45.62527
            ],
            "MapView": [
                -122.67672,
                45.62437,
                -122.67414,
                45.62617
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -122.67543,
```

```
                        45.62506
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Address": {
                        "Locality": 1,
                        "Intersection": [
                            1
                        ],
                        "AddressNumber": 1
                    }
                }
            }
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{"QueryText": "W. 6th St", "QueryComponents": {"AddressNumber": "415",
 "Locality": "Vancouver"}}'
```

AWS CLI

```
./aws geo-places geocode -key ${YourAPIKey} --query-text "W. 6th St" \
--query-components '{"AddressNumber" : "415", "Locality": "Vancouver"}'
```

开发者小贴士

使用IncludeCountries和之类的过滤器IncludePlaceTypes来获得准确的结果。例如，如果您需要来自美国的温哥华，"IncludeCountries": ["USA"]请申请优先考虑美国的结果。要了解更多信息，请参阅如何使用过滤器进行地理编码。

```
{
  "QueryText": "Vancouver",
```

```
  "Filter": {
    "IncludeCountries": ["USA"],
    "IncludePlaceTypes": ["City"]
  }
}
```

如何对兴趣点进行地理编码

您可以使用地理编码 API 对已知地点（兴趣点）的坐标进行地理编码，例如营业地址、地标或旅游景点。API 响应包含位置信息，包括地理坐标和匹配分数，这些信息表明结果与查询的匹配程度。

潜在用例

- 提供与兴趣点相对应的地理坐标可增强用户体验
- 提供与兴趣点相对应的地理坐标可以优化物流

示例

对企业进行地理编码

Sample request

```
{
    "QueryText": "Starbucks , Vancouver"
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "STARBUCKS",
            "Address": {
                "Label": "STARBUCKS, 1099 Robson St, Vancouver, BC V6E 1A9, Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
```

```
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "District": "West End",
                "PostalCode": "V6E 1A9",
                "Street": "Robson St",
                "StreetComponents": [
                    {
                        "BaseName": "Robson",
                        "Type": "St",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "1099"
            },
            "Position": [
                -123.12454,
                49.28462
            ],
            "Categories": [
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -123.12463,
                        49.28447
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
```

```
                "Title": 1,
                "Address": {
                    "Locality": 1
                }
            }
        }
    },
    {
        "PlaceId": "<Redacted>",
        "PlaceType": "PointOfInterest",
        "Title": "STARBUCKS",
        "Address": {
            "Label": "STARBUCKS, 980 Seymour St, Vancouver, BC V6B 1B5, Canada",
            "Country": {
                "Code2": "CA",
                "Code3": "CAN",
                "Name": "Canada"
            },
            "Region": {
                "Code": "BC",
                "Name": "British Columbia"
            },
            "SubRegion": {
                "Name": "Metro Vancouver"
            },
            "Locality": "Vancouver",
            "District": "Downtown Vancouver",
            "PostalCode": "V6B 1B5",
            "Street": "Seymour St",
            "StreetComponents": [
                {
                    "BaseName": "Seymour",
                    "Type": "St",
                    "TypePlacement": "AfterBaseName",
                    "TypeSeparator": " ",
                    "Language": "en"
                }
            ],
            "AddressNumber": "980"
        },
        "Position": [
            -123.12142,
            49.2786
        ],
```

```
            "Categories": [
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -123.12185,
                        49.27873
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Title": 1,
                    "Address": {
                        "Locality": 1
                    }
                }
            }
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "STARBUCKS",
            "Address": {
                "Label": "STARBUCKS, 918 SE 164th Ave, Vancouver, WA 98683-9603,
United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "Clark"
```

```
                },
                "Locality": "Vancouver",
                "District": "Cascade Highlands",
                "PostalCode": "98683-9603",
                "Street": "SE 164th Ave",
                "StreetComponents": [
                    {
                        "BaseName": "164th",
                        "Type": "Ave",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "SE",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "918"
            },
            "Position": [
                -122.50495,
                45.61409
            ],
            "Categories": [
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -122.50493,
                        45.61401
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Title": 1,
                    "Address": {
                        "Locality": 1
                    }
```

```
                }
            }
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "STARBUCKS",
            "Address": {
                "Label": "STARBUCKS, 304 W 8th St, Vancouver, WA 98660-3112, United
States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "Clark"
                },
                "Locality": "Vancouver",
                "District": "Esther Short",
                "PostalCode": "98660-3112",
                "Street": "W 8th St",
                "StreetComponents": [
                    {
                        "BaseName": "8th",
                        "Type": "St",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "W",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "304"
            },
            "Position": [
                -122.67418,
                45.6274
            ],
            "Categories": [
                {
```

```
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -122.67419,
                        45.62725
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Title": 1,
                    "Address": {
                        "Locality": 1
                    }
                }
            }
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "STARBUCKS",
            "Address": {
                "Label": "STARBUCKS, 3707 Main St, Vancouver, WA 98663-2227, United
States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "Clark"
                },
                "Locality": "Vancouver",
```

```
                "District": "Shumway",
                "PostalCode": "98663-2227",
                "Street": "Main St",
                "StreetComponents": [
                    {
                        "BaseName": "Main",
                        "Type": "St",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "3707"
            },
            "Position": [
                -122.667,
                45.64849
            ],
            "Categories": [
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -122.66705,
                        45.64849
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Title": 1,
                    "Address": {
                        "Locality": 1
                    }
                }
            }
        },
```

```
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "STARBUCKS",
            "Address": {
                "Label": "STARBUCKS, 2811 E Fourth Plain Blvd, Vancouver, WA 98661,
United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "Clark"
                },
                "Locality": "Vancouver",
                "District": "Rose Village",
                "PostalCode": "98661",
                "Street": "E Fourth Plain Blvd",
                "StreetComponents": [
                    {
                        "BaseName": "Fourth Plain",
                        "Type": "Blvd",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "E",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "2811"
            },
            "Position": [
                -122.64149,
                45.63818
            ],
            "Categories": [
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
```

```
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -122.64185,
                        45.63828
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Title": 1,
                    "Address": {
                        "Locality": 1
                    }
                }
            }
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "STARBUCKS",
            "Address": {
                "Label": "STARBUCKS, 2500 Columbia House Blvd, Vancouver, WA
98661-7758, United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "Clark"
                },
                "Locality": "Vancouver",
                "District": "Hudson Bay",
                "PostalCode": "98661-7758",
                "Street": "Columbia House Blvd",
```

```
                "StreetComponents": [
                    {
                        "BaseName": "Columbia House",
                        "Type": "Blvd",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "2500"
            },
            "Position": [
                -122.64584,
                45.62036
            ],
            "Categories": [
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -122.64584,
                        45.61986
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Title": 1,
                    "Address": {
                        "Locality": 1
                    }
                }
            }
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
```

```
            "Title": "STARBUCKS",
            "Address": {
                "Label": "STARBUCKS, 8801 NE Hazel Dell Ave, Vancouver, WA
98665-8145, United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "Clark"
                },
                "Locality": "Vancouver",
                "District": "Northeast Hazel Dell-Starcrest",
                "PostalCode": "98665-8145",
                "Street": "NE Hazel Dell Ave",
                "StreetComponents": [
                    {
                        "BaseName": "Hazel Dell",
                        "Type": "Ave",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "NE",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "8801"
            },
            "Position": [
                -122.66834,
                45.68623
            ],
            "Categories": [
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
                }
            ],
```

```
            "AccessPoints": [
                {
                    "Position": [
                        -122.66834,
                        45.68641
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Title": 1,
                    "Address": {
                        "Locality": 1
                    }
                }
            }
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "STARBUCKS",
            "Address": {
                "Label": "STARBUCKS, 7720 NE Highway 99, Vancouver, WA 98665, United
States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "Clark"
                },
                "Locality": "Vancouver",
                "District": "Northeast Hazel Dell",
                "PostalCode": "98665",
                "Street": "NE Highway 99",
                "StreetComponents": [
                    {
                        "BaseName": "Highway 99",
```

```
                        "Prefix": "NE",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "7720"
            },
            "Position": [
                -122.66264,
                45.67787
            ],
            "Categories": [
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -122.66293,
                        45.67793
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Title": 1,
                    "Address": {
                        "Locality": 1
                    }
                }
            }
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "STARBUCKS",
            "Address": {
                "Label": "STARBUCKS, 408 NE 81st St, Vancouver, WA 98665-8111,
United States",
                "Country": {
```

```
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "Clark"
                },
                "Locality": "Vancouver",
                "District": "Northeast Hazel Dell-Starcrest",
                "PostalCode": "98665-8111",
                "Street": "NE 81st St",
                "StreetComponents": [
                    {
                        "BaseName": "81st",
                        "Type": "St",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "NE",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "408"
            },
            "Position": [
                -122.66824,
                45.68161
            ],
            "Categories": [
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -122.66824,
                        45.68138
```

```
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Title": 1,
                    "Address": {
                        "Locality": 1
                    }
                }
            }
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "STARBUCKS",
            "Address": {
                "Label": "STARBUCKS, 6701 E Mill Plain Blvd, Vancouver, WA
98661-7459, United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "Clark"
                },
                "Locality": "Vancouver",
                "District": "Vancouver Heights",
                "PostalCode": "98661-7459",
                "Street": "E Mill Plain Blvd",
                "StreetComponents": [
                    {
                        "BaseName": "Mill Plain",
                        "Type": "Blvd",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "E",
                        "Language": "en"
```

```
                    }
                ],
                "AddressNumber": "6701"
            },
            "Position": [
                -122.6031,
                45.62623
            ],
            "Categories": [
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -122.6031,
                        45.62635
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Title": 1,
                    "Address": {
                        "Locality": 1
                    }
                }
            }
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "STARBUCKS",
            "Address": {
                "Label": "STARBUCKS, 8700 NE Vancouver Mall Dr, Vancouver, WA
98662-6416, United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
```

```
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "Clark"
                },
                "Locality": "Vancouver",
                "PostalCode": "98662-6416",
                "Street": "NE Vancouver Mall Dr",
                "StreetComponents": [
                    {
                        "BaseName": "Vancouver Mall",
                        "Type": "Dr",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "NE",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "8700"
            },
            "Position": [
                -122.58543,
                45.65786
            ],
            "Categories": [
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -122.58512,
                        45.65739
                    ]
                }
            ],
```

```
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Title": 1,
                    "Address": {
                        "Locality": 1
                    }
                }
            }
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "STARBUCKS",
            "Address": {
                "Label": "STARBUCKS, 8302 E Mill Plain Blvd, Vancouver, WA
98664-2007, United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "Clark"
                },
                "Locality": "Vancouver",
                "District": "North Garrison Heights",
                "PostalCode": "98664-2007",
                "Street": "E Mill Plain Blvd",
                "StreetComponents": [
                    {
                        "BaseName": "Mill Plain",
                        "Type": "Blvd",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "E",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "8302"
```

```
            },
            "Position": [
                -122.58719,
                45.62393
            ],
            "Categories": [
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -122.58712,
                        45.62393
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Title": 1,
                    "Address": {
                        "Locality": 1
                    }
                }
            }
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "STARBUCKS",
            "Address": {
                "Label": "STARBUCKS, 7809 NE Vancouver Plaza Dr, Vancouver, WA
98662-6624, United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
```

```
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "Clark"
                },
                "Locality": "Vancouver",
                "District": "Ogden",
                "PostalCode": "98662-6624",
                "Street": "NE Vancouver Plaza Dr",
                "StreetComponents": [
                    {
                        "BaseName": "Vancouver Plaza",
                        "Type": "Dr",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "NE",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "7809"
            },
            "Position": [
                -122.5919,
                45.65126
            ],
            "Categories": [
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -122.59217,
                        45.65112
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
```

```
                "Components": {
                    "Title": 1,
                    "Address": {
                        "Locality": 1
                    }
                }
            }
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "STARBUCKS",
            "Address": {
                "Label": "STARBUCKS, 2615 NE 112th Ave, Vancouver, WA 98684-4283,
United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "Clark"
                },
                "Locality": "Vancouver",
                "District": "Landover-Sharmel",
                "PostalCode": "98684-4283",
                "Street": "NE 112th Ave",
                "StreetComponents": [
                    {
                        "BaseName": "112th",
                        "Type": "Ave",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "NE",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "2615"
            },
            "Position": [
```

```
                -122.55675,
                45.64148
            ],
            "Categories": [
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -122.55675,
                        45.64157
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Title": 1,
                    "Address": {
                        "Locality": 1
                    }
                }
            }
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "STARBUCKS",
            "Address": {
                "Label": "STARBUCKS, 13719 SE Mill Plain Blvd, Vancouver, WA
98684-6945, United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
```

```
                },
                "SubRegion": {
                    "Name": "Clark"
                },
                "Locality": "Vancouver",
                "District": "Mountain View",
                "PostalCode": "98684-6945",
                "Street": "SE Mill Plain Blvd",
                "StreetComponents": [
                    {
                        "BaseName": "Mill Plain",
                        "Type": "Blvd",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "SE",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "13719"
            },
            "Position": [
                -122.53093,
                45.61691
            ],
            "Categories": [
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -122.53093,
                        45.61714
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Title": 1,
```

```
                    "Address": {
                        "Locality": 1
                    }
                }
            }
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "STARBUCKS",
            "Address": {
                "Label": "STARBUCKS, 6711 NE 63rd St, Vancouver, WA 98661-1997,
United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "Clark"
                },
                "Locality": "Vancouver",
                "District": "Walnut Grove",
                "PostalCode": "98661-1997",
                "Street": "NE 63rd St",
                "StreetComponents": [
                    {
                        "BaseName": "63rd",
                        "Type": "St",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "NE",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "6711"
            },
            "Position": [
                -122.60365,
                45.66671
```

```
            ],
            "Categories": [
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -122.60365,
                        45.66679
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Title": 1,
                    "Address": {
                        "Locality": 1
                    }
                }
            }
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "STARBUCKS",
            "Address": {
                "Label": "STARBUCKS, 8720 NE Centerpointe Dr, Vancouver, WA
98665-1153, United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
```

```
                    "Name": "Clark"
                },
                "Locality": "Vancouver",
                "District": "Sunnyside-Walnut Grove",
                "PostalCode": "98665-1153",
                "Street": "NE Centerpointe Dr",
                "StreetComponents": [
                    {
                        "BaseName": "Centerpointe",
                        "Type": "Dr",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "NE",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "8720"
            },
            "Position": [
                -122.6019,
                45.68537
            ],
            "Categories": [
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -122.60189,
                        45.6852
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Title": 1,
                    "Address": {
                        "Locality": 1
```

```
                    }
                }
            }
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "STARBUCKS",
            "Address": {
                "Label": "STARBUCKS, 7411 NE 117th Ave, Vancouver, WA 98662-4705,
United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "Clark"
                },
                "Locality": "Vancouver",
                "District": "Orchards Area",
                "PostalCode": "98662-4705",
                "Street": "NE 117th Ave",
                "StreetComponents": [
                    {
                        "BaseName": "117th",
                        "Type": "Ave",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "NE",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "7411"
            },
            "Position": [
                -122.55103,
                45.67666
            ],
            "Categories": [
```

```
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -122.55103,
                        45.67684
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Title": 1,
                    "Address": {
                        "Locality": 1
                    }
                }
            }
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "STARBUCKS",
            "Address": {
                "Label": "STARBUCKS, 11211 NE Fourth Plain Blvd, Vancouver, WA
98662-5770, United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "Clark"
                },
```

```
                "Locality": "Vancouver",
                "District": "Orchards Area",
                "PostalCode": "98662-5770",
                "Street": "NE Fourth Plain Blvd",
                "StreetComponents": [
                    {
                        "BaseName": "Fourth Plain",
                        "Type": "Blvd",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "NE",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "11211"
            },
            "Position": [
                -122.5575,
                45.66574
            ],
            "Categories": [
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -122.55751,
                        45.66587
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Title": 1,
                    "Address": {
                        "Locality": 1
                    }
                }
```

```
            }
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryText": "Starbucks , Vancouver"
}'
```

AWS CLI

```
aws geo-places geocode --key ${YourKey} --query-text "Starbucks , Vancouver"
```

对旅游景点进行地理编码

Sample request

```
{
    "QueryText": "Eiffel Tower, Paris"
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "The Eiffel Tower Romance",
            "Address": {
                "Label": "The Eiffel Tower Romance, Rue de Rivoli, 75004 Paris,
 France",
                "Country": {
                    "Code2": "FR",
                    "Code3": "FRA",
                    "Name": "France"
```

```
                },
                "Region": {
                    "Code": "IDF",
                    "Name": "Île-de-France"
                },
                "SubRegion": {
                    "Name": "Paris"
                },
                "Locality": "Paris",
                "District": "4e Arrondissement",
                "PostalCode": "75004",
                "Street": "Rue de Rivoli",
                "StreetComponents": [
                    {
                        "BaseName": "Rivoli",
                        "Type": "Rue de",
                        "TypePlacement": "BeforeBaseName",
                        "TypeSeparator": " ",
                        "Language": "fr"
                    }
                ]
            },
            "Position": [
                2.35284,
                48.85702
            ],
            "Categories": [
                {
                    "Id": "theatre,_music_and_culture",
                    "Name": "Theatre, Music and Culture",
                    "LocalizedName": "Théâtre, musique et culture",
                    "Primary": true
                },
                {
                    "Id": "historical_monument",
                    "Name": "Historical Monument",
                    "LocalizedName": "Monument historique",
                    "Primary": false
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        2.3529,
```

```
                        48.85714
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Title": 1,
                    "Address": {
                        "Locality": 1
                    }
                }
            }
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Eiffel Tower",
            "Address": {
                "Label": "Eiffel Tower, 2025 S Collegiate Dr, Paris, TX 75460-7853,
United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "TX",
                    "Name": "Texas"
                },
                "SubRegion": {
                    "Name": "Lamar"
                },
                "Locality": "Paris",
                "PostalCode": "75460-7853",
                "Street": "S Collegiate Dr",
                "StreetComponents": [
                    {
                        "BaseName": "Collegiate",
                        "Type": "Dr",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "S",
                        "Language": "en"
```

```
                }
            ],
            "AddressNumber": "2025"
        },
        "Position": [
            -95.52403,
            33.64055
        ],
        "Categories": [
            {
                "Id": "historical_monument",
                "Name": "Historical Monument",
                "LocalizedName": "Historical Monument",
                "Primary": true
            },
            {
                "Id": "landmark-attraction",
                "Name": "Landmark-Attraction",
                "LocalizedName": "Landmark-Attraction",
                "Primary": false
            }
        ],
        "AccessPoints": [
            {
                "Position": [
                    -95.52411,
                    33.6405
                ]
            }
        ],
        "MatchScores": {
            "Overall": 1,
            "Components": {
                "Title": 1,
                "Address": {
                    "Locality": 1
                }
            }
        }
    },
    {
        "PlaceId": "<Redacted>",
        "PlaceType": "PointOfInterest",
        "Title": "Eiffel Tower",
```

```
            "Address": {
                "Label": "Eiffel Tower, Northland Dr, Paris, MI 49338, United
States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "MI",
                    "Name": "Michigan"
                },
                "SubRegion": {
                    "Name": "Mecosta"
                },
                "Locality": "Paris",
                "PostalCode": "49338",
                "Street": "Northland Dr",
                "StreetComponents": [
                    {
                        "BaseName": "Northland",
                        "Type": "Dr",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ]
            },
            "Position": [
                -85.50268,
                43.78907
            ],
            "Categories": [
                {
                    "Id": "outdoor-recreation",
                    "Name": "Outdoor-Recreation",
                    "LocalizedName": "Outdoor/Recreation",
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -85.50282,
```

```
                        43.78907
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Title": 1,
                    "Address": {
                        "Locality": 1
                    }
                }
            }
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryText": "Eiffel Tower, Paris"
    }'
```

AWS CLI

```
aws geo-places geocode --key ${YourKey} --query-text "Eiffel Tower, Paris"
```

开发者小贴士

- 使用完整的地址或查询组件以获得更好的结果。要了解更多信息，请参阅如何对地址进行地理编码。
- 使用IncludeCountries和之类的过滤器IncludePlaceTypes来获得准确的结果。例如，如果您需要来自美国的温哥华，"IncludeCountries": ["USA"]请申请优先考虑美国的结果。要了解更多信息，请参阅如何使用过滤器进行地理编码。

```
{
  "QueryText": "Vancouver",
  "Filter": {
    "IncludeCountries": ["USA"],
```

```
    "IncludePlaceTypes": ["City"]
  }
}
```

如何使用邮政编码对地址号码进行地理编码

在邮政编码非常具体（仅链接同一条街上的几个地址）的国家，只能使用门牌号和邮政编码来找到地址。这种做法在运输和包裹配送物流中很常见。加拿大、英国、荷兰、美国 (ZIP+4)、以色列、爱尔兰和新加坡支持此功能。在爱尔兰和新加坡，邮政编码都提供精确的位置信息，直至具体的门牌号码。

示例

使用邮政编码对地址进行地理编码

Sample request

使用查询组件

```
{
  "QueryComponents": {
    "AddressNumber": "1368",
    "PostalCode": "V5N 1T2"
  }
}
```

使用自由文本

```
{
  "QueryText": "1368, V5N1T2"
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointAddress",
            "Title": "1368 E 8th Ave, Vancouver, BC V5N 1T2, Canada",
            "Address": {
                "Label": "1368 E 8th Ave, Vancouver, BC V5N 1T2, Canada",
                "Country": {
```

```
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "District": "Grandview-Woodland",
                "PostalCode": "V5N 1T2",
                "Street": "E 8th Ave",
                "StreetComponents": [
                    {
                        "BaseName": "8th",
                        "Type": "Ave",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "E",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "1368"
            },
            "Position": [
                -123.07612,
                49.26306
            ],
            "MapView": [
                -123.0775,
                49.26216,
                -123.07474,
                49.26396
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -123.07611,
                        49.26333
                    ]
                }
```

```
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Address": {
                        "PostalCode": 1,
                        "AddressNumber": 1
                    }
                }
            }
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
  "QueryComponents": {
    "AddressNumber": "1368",
    "PostalCode": "V5N 1T2"
  }
}'
```

AWS CLI

```
aws geo-places geocode --key ${YourKey} --query-components '{"AddressNumber" :
 "1368", "PostalCode": "V5N 1T2"}'
```

使用新加坡和爱尔兰的邮政编码对地址进行地理编码

在爱尔兰，Eircode系统为每个家庭和企业分配一个唯一的代码，而在新加坡，邮政编码同样具体。在这两个国家，仅凭邮政编码就能识别出确切的地址：不需要街道名称、城市或门牌号。

- 新加坡：6 位数的邮政编码
- 爱尔兰：7 个字符的 Eircode

Sample request

使用查询组件

```
{
  "QueryComponents": {
    "PostalCode": "D02 X285"
  }
}
```

使用自由文本

```
{
  "QueryText": "D02 X285"
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointAddress",
            "Title": "29-31 Adelaide Road, Dublin, County Dublin, D02 X285,
 Ireland",
            "Address": {
                "Label": "29-31 Adelaide Road, Dublin, County Dublin, D02 X285,
 Ireland",
                "Country": {
                    "Code2": "IE",
                    "Code3": "IRL",
                    "Name": "Ireland"
                },
                "SubRegion": {
                    "Code": "D",
                    "Name": "County Dublin"
                },
                "Locality": "Dublin",
                "District": "Dublin 2",
                "PostalCode": "D02 X285",
                "Street": "Adelaide Road",
                "StreetComponents": [
                    {
```

```
                        "BaseName": "Adelaide",
                        "Type": "Road",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "29-31"
            },
            "Position": [
                -6.25549,
                53.33207
            ],
            "MapView": [
                -6.257,
                53.33117,
                -6.25398,
                53.33297
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -6.25536,
                        53.33231
                    ]
                }
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Address": {
                        "PostalCode": 1
                    }
                }
            }
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
```

```
  --header 'Content-Type: application/json' \
  --data '{
  "QueryComponents": {
    "PostalCode": "D02 X285"
  }
}'
```

AWS CLI

```
aws geo-places geocode --key ${YourKey} --query-components '{"PostalCode": "V5N
 1T2"}'
```

开发者小贴士

了解有关 ZIP+4（美国）和 Eircode（爱尔兰）的更多信息。此外，还要了解加拿大、英国、荷兰、新加坡和以色列的邮政编码系统。

如何使用地理空间上下文进行地理编码

Geocode API 允许您使用地理空间上下文（例如偏差位置）来获得所需的结果。

> **Note**
>
> “偏置位置” 是指一种根据用户的指定位置或定义的区域对搜索结果进行优先级排序的机制。它本质上是将搜索结果的重点转移到地理上更接近指定偏差点的地点，而不一定排除其他结果。

潜在用途

使用地理空间上下文根据您的业务需求获得正确的结果。

示例

将结果偏置到某个位置

通过偏向某个位置，您可以更改结果的排名。尝试以下带有（然后不带）`BiasPosition`值的示例，然后比较结果。

Sample request

```
{
```

```
    "QueryText": "George Street",
    "BiasPosition": [
                151.2059,
                -33.8691
            ]
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "Street",
            "Title": "George St, Sydney NSW, Australia",
            "Address": {
                "Label": "George St, Sydney NSW, Australia",
                "Country": {
                    "Code2": "AU",
                    "Code3": "AUS",
                    "Name": "Australia"
                },
                "Region": {
                    "Code": "NSW",
                    "Name": "New South Wales"
                },
                "Locality": "Sydney",
                "Street": "George St",
                "StreetComponents": [
                    {
                        "BaseName": "George",
                        "Type": "St",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ]
            },
            "Position": [
                151.20691,
                -33.86974
            ],
            "Distance": 117,
```

```
            "MapView": [
                151.20225,
                -33.88406,
                151.20912,
                -33.85635
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Address": {
                        "Intersection": [
                            1
                        ]
                    }
                }
            }
        },
        ...
        ...
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
  "QueryText": "George Street",
    "BiasPosition": [
                151.2059,
                -33.8691
            ]
}'
```

AWS CLI

```
aws geo-places geocode --key ${YourKey} --query-text "George Street" --bias-position
 151.2059 -33.8691
```

开发者小贴士

对于地址地理编码，请尝试使用完整的地址或带有偏差位置（包括国家和地点）组合的查询组件。要了解更多信息，请参阅[如何对地址进行地理编码]。

如何使用过滤器进行地理编码

Geocode API 允许您使用过滤器来获得所需的结果。

潜在用途

使用筛选条件根据您的业务需求限制结果。

示例

限制一个国家/地区内的搜索结果

为指定一个值`IncludeCountries`以在结果中返回该国家/地区的值。

Sample request

否则`IncludeCountries": ["USA"]`，Geocode API 将返回加拿大不列颠哥伦比亚省温哥华。

```
{
  "QueryText": "Vancouver",
    "Filter" : {
        "IncludeCountries": ["USA"]
    }
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "Locality",
            "Title": "Vancouver, WA, United States",
            "Address": {
                "Label": "Vancouver, WA, United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
```

```
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "Clark"
                },
                "Locality": "Vancouver",
                "PostalCode": "98660"
            },
            "Position": [
                -122.67156,
                45.63248
            ],
            "MapView": [
                -122.77466,
                45.57714,
                -122.46451,
                45.69803
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Address": {
                        "Locality": 1
                    }
                }
            }
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
  "QueryText": "Vancouver",
    "Filter" : {
        "IncludeCountries": ["USA"]
```

```
    }
}'
```

AWS CLI

```
aws geo-places geocode --key ${YourKey} --query-text "Vancouver" --filter
 '{"IncludeCountries": ["USA"]}'
```

使用地点类型筛选器优化结果

为指定一个值IncludePlaceTypes以返回结果中该位置的值。

Sample request

否则IncludePlaceTypes": ["Street"]，Geocode API 将返回格鲁吉亚（一个国家）。您可以通过将CAN和 US A的值**"IncludeCountries"**相加并比较结果来进一步细化结果。

```
{
  "QueryText": "Georgia",
    "Filter" : {
        "IncludePlaceTypes": ["Street"]
    }

}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "Street",
            "Title": "Georgia, Benito Juárez, CDMX, México",
            "Address": {
                "Label": "Georgia, Benito Juárez, CDMX, México",
                "Country": {
                    "Code2": "MX",
                    "Code3": "MEX",
                    "Name": "México"
                },
                "Region": {
```

```
                    "Code": "CDMX",
                    "Name": "Ciudad de México"
                },
                "SubRegion": {
                    "Name": "Ciudad de México"
                },
                "Locality": "Benito Juárez",
                "Street": "Georgia",
                "StreetComponents": [
                    {
                        "BaseName": "Georgia",
                        "Language": "es"
                    }
                ]
            },
            "Position": [
                -99.17754,
                19.38887
            ],
            "MapView": [
                -99.18133,
                19.38755,
                -99.17374,
                19.39016
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Address": {
                        "Intersection": [
                            1
                        ]
                    }
                }
            }
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
```

```
  --header 'Content-Type: application/json' \
  --data '{
  "QueryText": "Georgia",
    "Filter" : {
        "IncludePlaceTypes": ["Street"]
    }

}'
```

AWS CLI

```
aws geo-places geocode --key ${YourKey} --query-text "Georgia" --filter
 '{"IncludePlaceTypes": ["Street"]}'
```

开发者小贴士

对于地址地理编码，请尝试使用完整的地址或带有偏差位置（包括国家和地点）组合的查询组件。要了解更多信息，请参阅如何对地址进行地理编码。

如何根据城市的时区进行地理编码

您可以使用地理编码 API 来提供时区信息，例如 UTC 偏移量和时区名称。

潜在用途

对时区进行地理编码的可能用途：

- 创建世界时钟
- 在不同地区安排会议

示例

示例

根据时区要求在布鲁塞尔进行地理编码。

Sample request

```
{
  "QueryText": "Brussels",
```

```
  "Filter" : {
        "IncludePlaceTypes": ["Locality"]
    },
    "AdditionalFeatures": [
    "TimeZone"
  ]
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "Locality",
            "Title": "Brussel, België",
            "Address": {
                "Label": "Brussel, België",
                "Country": {
                    "Code2": "BE",
                    "Code3": "BEL",
                    "Name": "België"
                },
                "Region": {
                    "Code": "BRU",
                    "Name": "Brussel"
                },
                "SubRegion": {
                    "Name": "Brussel"
                },
                "Locality": "Brussel",
                "PostalCode": "1000"
            },
            "Position": [
                4.35609,
                50.84439
            ],
            "MapView": [
                4.3139,
                50.79628,
                4.43709,
                50.91397
            ],
```

```
            "TimeZone": {
                "Name": "Europe/Brussels",
                "Offset": "+02:00",
                "OffsetSeconds": 7200
            },
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Address": {
                        "Locality": 1
                    }
                }
            }
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
  "QueryText": "Brussels",
  "Filter" : {
        "IncludePlaceTypes": ["Locality"]
    },
    "AdditionalFeatures": [
    "TimeZone"
  ]
}'
```

AWS CLI

```
aws geo-places geocode --key ${YourKey} --query-text "Brussels" \
--filter '{"IncludePlaceTypes": ["Locality"]}' \
--additional-features "TimeZone"
```

如何使用特定语言进行地理编码

此功能允许从 BCP47兼容代码中选择首选响应语言。它根据名称变体检测查询语言，并使用首选语言处理不匹配的标记和模棱两可的情况。如果没有请求的语言，Plac es API 会以官方国家/地区语言提供

搜索结果，但会优先考虑不同区域语言的区域。作为后备策略，如果请求的语言中没有任何地址元素不可用，Plac es 将以默认语言 APIs 返回地址。

潜在用例

一个潜在的用例是本地化查询和/或结果。

示例

以特定语言获取结果

Sample request

```
{
    "QueryText":"Patna, Bihar, Bharat",
    "Language": "HI"
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "Locality",
            "Title": "####, #####, ####",
            "Address": {
                "Label": "####, #####, ####",
                "Country": {
                    "Code2": "IN",
                    "Code3": "IND",
                    "Name": "####"
                },
                "Region": {
                    "Code": "BR",
                    "Name": "#####"
                },
                "SubRegion": {
                    "Name": "####"
                },
                "Locality": "####",
                "PostalCode": "800001"
            },
```

```
            "Position": [
                85.13752,
                25.60134
            ],
            "MapView": [
                85.03222,
                25.55157,
                85.27107,
                25.65917
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Address": {
                        "Country": 1,
                        "Region": 1,
                        "Locality": 1
                    }
                }
            }
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryText":"Patna, Bihar, Bharat",
    "Language": "HI"
}'
```

AWS CLI

```
aws geo-places geocode --key ${YourKey} --query-text "####, #####, ####" --language
 "HI"
```

使用特定语言查询

Sample request

```
{
    "QueryText":"####, #####, ####"
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "Locality",
            "Title": "Patana, Bihar, Bharat",
            "Address": {
                "Label": "Patana, Bihar, Bharat",
                "Country": {
                    "Code2": "IN",
                    "Code3": "IND",
                    "Name": "Bharat"
                },
                "Region": {
                    "Code": "BR",
                    "Name": "Bihar"
                },
                "SubRegion": {
                    "Name": "Patana"
                },
                "Locality": "Patana",
                "PostalCode": "800001"
            },
            "Position": [
                85.13752,
                25.60134
            ],
            "MapView": [
                85.03222,
                25.55157,
                85.27107,
                25.65917
            ],
```

```
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Address": {
                        "Country": 1,
                        "Region": 1,
                        "Locality": 1
                    }
                }
            }
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryText":"####, #####, ####"
}'
```

AWS CLI

```
aws geo-places geocode --key ${YourKey} --query-text "####, #####, ####"
```

使用特定语言进行查询并生成结果

Sample request

```
{
    "QueryText":"####, #####, ####",
    "Language": "HI"
}
```

Sample response

```
{
    "ResultItems": [
        {
```

```
            "PlaceId": "<Redacted>",
            "PlaceType": "Locality",
            "Title": "####, #####, ####",
            "Address": {
                "Label": "####, #####, ####",
                "Country": {
                    "Code2": "IN",
                    "Code3": "IND",
                    "Name": "####"
                },
                "Region": {
                    "Code": "BR",
                    "Name": "#####"
                },
                "SubRegion": {
                    "Name": "####"
                },
                "Locality": "####",
                "PostalCode": "800001"
            },
            "Position": [
                85.13752,
                25.60134
            ],
            "MapView": [
                85.03222,
                25.55157,
                85.27107,
                25.65917
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Address": {
                        "Country": 1,
                        "Region": 1,
                        "Locality": 1
                    }
                }
            }
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryText":"####, #####, ####",
    "Language": "HI"
}'
```

AWS CLI

```
aws geo-places geocode --key ${YourKey} --query-text "####, #####, ####" --language
 "HI"
```

如何获取辅助地址

SecondaryAddresses允许您检索主地址下的所有辅助地址。此外，Geocode还会返回辅助单位（如果其中存在辅助单位）QueryText。

潜在用例

- 填写地址表：选择更准确的辅助地址，其中还包括更准确的位置信息。
- 配送：对于缺乏二级地址覆盖范围的国家/地区，此信息可以包含所有提供的单位信息，从而为配送提供信息。

示例

获取辅助地址

此功能适用于以下国家：澳大利亚、加拿大、新西兰、美国、PRI。

Sample request

```
{
    "QueryText":"910 Beach Avenue, Vancouver",
    "AdditionalFeatures": ["SecondaryAddresses"]
}
```

Sample response


```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointAddress",
            "Title": "910 Beach Ave, Vancouver, BC V6Z 2W7, Canada",
            "Address": {
                "Label": "910 Beach Ave, Vancouver, BC V6Z 2W7, Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "District": "Downtown Vancouver",
                "PostalCode": "V6Z 2W7",
                "Street": "Beach Ave",
                "StreetComponents": [
                    {
                        "BaseName": "Beach",
                        "Type": "Ave",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "910"
            },
            "Position": [
                -123.13325,
                49.27542
            ],
            "MapView": [
                -123.13463,
                49.27452,
```

```
                -123.13187,
                49.27632
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Address": {
                        "Locality": 1,
                        "Intersection": [
                            1
                        ],
                        "AddressNumber": 1
                    }
                }
            },
            "ParsedQuery": {
                "Address": {
                    "Locality": [
                        {
                            "StartIndex": 18,
                            "EndIndex": 27,
                            "Value": "Vancouver",
                            "QueryComponent": "Query"
                        }
                    ],
                    "Street": [
                        {
                            "StartIndex": 4,
                            "EndIndex": 16,
                            "Value": "Beach Avenue",
                            "QueryComponent": "Query"
                        }
                    ],
                    "AddressNumber": [
                        {
                            "StartIndex": 0,
                            "EndIndex": 3,
                            "Value": "910",
                            "QueryComponent": "Query"
                        }
                    ]
                }
            },
            "SecondaryAddresses": [
```

```
                {
                    "PlaceId": "<Redacted>",
                    "PlaceType": "SecondaryAddress",
                    "Title": "101-910 Beach Ave, Vancouver, BC V6Z 2W7, Canada",
                    "Address": {
                        "Label": "101-910 Beach Ave, Vancouver, BC V6Z 2W7, Canada",
                        "SecondaryAddressComponents": [
                            {
                                "Number": "101"
                            }
                        ]
                    },
                    "Position": [
                        -123.1334,
                        49.27532
                    ]
                },
                {
                    "PlaceId": "<Redacted>",
                    "PlaceType": "SecondaryAddress",
                    "Title": "102-910 Beach Ave, Vancouver, BC V6Z 2W7, Canada",
                    "Address": {
                        "Label": "102-910 Beach Ave, Vancouver, BC V6Z 2W7, Canada",
                        "SecondaryAddressComponents": [
                            {
                                "Number": "102"
                            }
                        ]
                    },
                    "Position": [
                        -123.1334,
                        49.27532
                    ]
                },
                ...
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
```

```
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryText":"910 Beach Avenue, Vancouver",
    "AdditionalFeatures": ["SecondaryAddresses"]
}'
```

AWS CLI

```
aws geo-places geocode --key ${YourKey} --query-text "910 Beach Avenue, Vancouver"
 --additional-features "SecondaryAddresses"
```

获取地址中提供的辅助单元

此功能适用于以下国家：澳大利亚、澳大利亚、巴西、加拿大、西班牙、法国、英国、印度尼西亚、印度、新西兰、土耳其、TUR、TWN、美国。

Sample request

```
{
  "QueryText": "101-910 Beach Avenue, Vancouver"
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "SecondaryAddress",
            "Title": "101-910 Beach Ave, Vancouver, BC V6Z 2W7, Canada",
            "Address": {
                "Label": "101-910 Beach Ave, Vancouver, BC V6Z 2W7, Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
```

```
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "District": "Downtown Vancouver",
                "PostalCode": "V6Z 2W7",
                "Street": "Beach Ave",
                "StreetComponents": [
                    {
                        "BaseName": "Beach",
                        "Type": "Ave",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "910",
                "SecondaryAddressComponents": [
                    {
                        "Number": "101"
                    }
                ]
            },
            "Position": [
                -123.1334,
                49.27532
            ],
            "MapView": [
                -123.13478,
                49.27442,
                -123.13202,
                49.27622
            ],
            "MatchScores": {
                "Overall": 1,
                "Components": {
                    "Address": {
                        "Locality": 1,
                        "Intersection": [
                            1
                        ],
                        "AddressNumber": 1,
                        "SecondaryAddressComponents": [
                            {
```

```
                            "Number": 1
                        }
                    ]
                }
            }
        },
        "ParsedQuery": {
            "Address": {
                "Locality": [
                    {
                        "StartIndex": 22,
                        "EndIndex": 31,
                        "Value": "Vancouver",
                        "QueryComponent": "Query"
                    }
                ],
                "Street": [
                    {
                        "StartIndex": 8,
                        "EndIndex": 20,
                        "Value": "Beach Avenue",
                        "QueryComponent": "Query"
                    }
                ],
                "AddressNumber": [
                    {
                        "StartIndex": 4,
                        "EndIndex": 7,
                        "Value": "910",
                        "QueryComponent": "Query"
                    }
                ],
                "SecondaryAddressComponents": [
                    {
                        "StartIndex": 0,
                        "EndIndex": 3,
                        "Value": "101",
                        "Number": "101",
                        "Designator": "unknown"
                    }
                ]
            }
        },
        "MainAddress": {
```

```
                "PlaceId": "<Redacted>",
                "PlaceType": "PointAddress",
                "Title": "910 Beach Ave, Vancouver, BC V6Z 2W7, Canada",
                "Address": {
                    "Label": "910 Beach Ave, Vancouver, BC V6Z 2W7, Canada"
                },
                "Position": [
                    -123.13325,
                    49.27542
                ]
            }
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryText":"101-910 Beach Avenue, Vancouver"
}'
```

AWS CLI

```
aws geo-places geocode --key ${YourKey} --query-text "101-910 Beach Avenue,
 Vancouver"
```

如何获得十字路口

Intersections允许您检索附近的所有十字路口。这可能会被紧急服务部门和送货快递员使用。

潜在用例

- 应急响应：应急响应车辆通常需要识别附近的十字路口，以便在响应呼叫时获得最佳定位。这使他们能够保持清晰的访问路线和更快的响应时间，同时确保多种方法的可见性。
- 送货快递员：送货快递员可以利用十字路口数据来寻找更高效的停车位，尤其是在停车位可能有限或受限的密集城市地区 door-to-door。

示例

获取辅助地址

Sample request

```
{
    "QueryText":"910 Beach Avenue, Vancouver",
    "AdditionalFeatures": ["Intersections"]
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointAddress",
            "Title": "910 Beach Ave, Vancouver, BC V6Z 2W7, Canada",
            "Address": {
                "Label": "910 Beach Ave, Vancouver, BC V6Z 2W7, Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "District": "Downtown Vancouver",
                "PostalCode": "V6Z 2W7",
                "Street": "Beach Ave",
                "StreetComponents": [
                    {
                        "BaseName": "Beach",
                        "Type": "Ave",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
```

```
                    "Language": "en"
                }
            ],
            "AddressNumber": "910"
        },
        "Position": [
            -123.13325,
            49.27542
        ],
        "MapView": [
            -123.13463,
            49.27452,
            -123.13187,
            49.27632
        ],
        "MatchScores": {
            "Overall": 1,
            "Components": {
                "Address": {
                    "Locality": 1,
                    "Intersection": [
                        1
                    ],
                    "AddressNumber": 1
                }
            }
        },
        "ParsedQuery": {
            "Address": {
                "Locality": [
                    {
                        "StartIndex": 18,
                        "EndIndex": 27,
                        "Value": "Vancouver",
                        "QueryComponent": "Query"
                    }
                ],
                "Street": [
                    {
                        "StartIndex": 4,
                        "EndIndex": 16,
                        "Value": "Beach Avenue",
                        "QueryComponent": "Query"
                    }
```

```
                    ],
                    "AddressNumber": [
                        {
                            "StartIndex": 0,
                            "EndIndex": 3,
                            "Value": "910",
                            "QueryComponent": "Query"
                        }
                    ]
                }
            },
            "Intersections": [
                {
                    "PlaceId": "<Redacted>",
                    "Title": "Beach Ave & Hornby St, Vancouver, BC V6Z, Canada",
                    "Address": {
                        "Label": "Beach Ave & Hornby St, Vancouver, BC V6Z, Canada",
                        "PostalCode": "V6Z",
                        "Intersection": [
                            "Beach Ave",
                            "Hornby St"
                        ]
                    },
                    "Position": [
                        -123.1328,
                        49.27536
                    ]
                },
                {
                    "PlaceId": "<Redacted>",
                    "Title": "Beach Ave & Burrard St, Vancouver, BC, Canada",
                    "Address": {
                        "Label": "Beach Ave & Burrard St, Vancouver, BC, Canada",
                        "Intersection": [
                            "Beach Ave",
                            "Burrard St"
                        ]
                    },
                    "Position": [
                        -123.13377,
                        49.27599
                    ]
                }
            ]
```

```
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/geocode?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryText":"910 Beach Avenue, Vancouver",
    "AdditionalFeatures": ["Intersections"]
}'
```

AWS CLI

```
aws geo-places geocode --key ${YourKey} --query-text "910 Beach Avenue, Vancouver"
 --additional-features "Intersections"
```

如何使用 ReverseGeocode

本节包含各种操作指南和使用示例 ReverseGeocode APIs。

主题

- 如何对某个位置进行反向地理编码
- 如何反向地理编码以获得正确的结果
- 如何使用特定语言进行反向地理编码
- 如何对城市的时区进行反向地理编码
- 如何从政治角度进行反向地理编码
- 如何获得十字路口

如何对某个位置进行反向地理编码

反向地理编码 API 允许您根据位置查询将地理编码转换为地理区域。API 响应包含地点详情，提供与特定坐标关联的位置的相关信息。

潜在用例

- 存储地点信息：将地点详细信息添加到包含地理坐标的数据存储中。
- 地图可视化：使用地点信息在地图上显示数据。
- 用户位置检测：根据用户设备位置识别用户的位置。

对位置进行反向地理编码

Sample request

```
{
  "QueryPosition": [
    -123.11694,
    49.28126
  ]
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointAddress",
            "Title": "510 W Georgia St, Vancouver, BC V6B 0M3, Canada",
            "Address": {
                "Label": "510 W Georgia St, Vancouver, BC V6B 0M3, Canada",
                "Country": { "Code2": "CA", "Code3": "CAN", "Name": "Canada" },
                "Region": { "Code": "BC", "Name": "British Columbia" },
                "SubRegion": { "Name": "Metro Vancouver" },
                "Locality": "Vancouver",
                "District": "Downtown Vancouver",
                "PostalCode": "V6B 0M3",
                "Street": "W Georgia St",
                "StreetComponents": [
                    { "BaseName": "Georgia", "Type": "St", "TypePlacement":
 "AfterBaseName", "TypeSeparator": " ", "Prefix": "W", "Language": "en" }
                ],
                "AddressNumber": "510"
            },
            "Position": [-123.11694, 49.28126],
            "Distance": 0,
```

```
            "MapView": [-123.11813, 49.27786, -123.11076, 49.28246],
            "AccessPoints": [
                { "Position": [-123.11656, 49.28151] }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/reverse-geocode?
key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
        "QueryPosition": [
            -123.11694,
            49.28126
        ]
}'
```

AWS CLI

```
aws geo-places reverse-geocode --key ${YourKey} --query-position
 "-123.11694,49.28126"
```

对具有多个结果的位置进行反向地理编码

Sample request

```
{
  "QueryPosition": [
    -123.11694,
    49.28126
  ],
  "MaxResults": "3"
}
```

Sample response

```
{
```

```
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointAddress",
            "Title": "510 W Georgia St, Vancouver, BC V6B 0M3, Canada",
            "Address": { /* Address details */ },
            "Position": [-123.11694, 49.28126],
            "Distance": 0,
            "MapView": [/* Map view details */],
            "AccessPoints": [/* Access point details */]
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "ChargePoint",
            "Address": { /* Address details */ },
            "Position": [-123.11663, 49.28116],
            "Distance": 25,
            "Categories": [/* Category details */],
            "AccessPoints": [/* Access point details */]
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Zipcar",
            "Address": { /* Address details */ },
            "Position": [-123.11715, 49.28094],
            "Distance": 29,
            "Categories": [/* Category details */],
            "AccessPoints": [/* Access point details */]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/reverse-geocode?
key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
        "QueryPosition": [
            -123.11694,
```

```
            49.28126
        ],
        "MaxResults": "3"
}'
```

AWS CLI

```
aws geo-places reverse-geocode --key ${YourKey} --query-position
 "-123.11694,49.28126" --max-results "3"
```

开发者小贴士

要获得目标结果，请在过滤器IncludePlaceTypes中使用。

```
{
  "QueryPosition": [
    -123.11694,
    49.28126
  ],
  "Filter": { "IncludePlaceTypes": ["PointAddress"] }
}
```

如何反向地理编码以获得正确的结果

本指南提供了优化反向地理编码结果的方法，可确保返回的数据与特定的业务需求紧密一致。使用过滤器，用户可以缩小结果范围，以便更精确地匹配地址点、街道或地点等类型。

潜在用例

- 根据@@ 特定需求限制搜索结果：根据业务需求，使用筛选器仅检索最相关的信息，例如确切地址或更广泛的地理位置数据。

示例

筛选点地址

通过筛选PointAddress，您可以检索特定的街道地址，从而提高位置的准确性。

Sample Request

```
{
```

```
    "QueryPosition": [
        -97.721, 30.404
    ],
    "Filter": {
        "IncludePlaceTypes": [
            "PointAddress"
        ]
    }
}
```

Sample Response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointAddress",
            "Title": "11721 Domain Blvd, Austin, TX 78758-0051, United States",
            "Address": {
                "Label": "11721 Domain Blvd, Austin, TX 78758-0051, United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "TX",
                    "Name": "Texas"
                },
                "SubRegion": {
                    "Name": "Travis"
                },
                "Locality": "Austin",
                "District": "North Burnet",
                "PostalCode": "78758-0051",
                "Street": "Domain Blvd",
                "StreetComponents": [
                    {
                        "BaseName": "Domain",
                        "Type": "Blvd",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
```

```
                    }
                ],
                "AddressNumber": "11721"
            },
            "Position": [
                -97.72087,
                30.404
            ],
            "Distance": 5,
            "MapView": [
                -97.72219,
                30.40273,
                -97.72057,
                30.40649
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/reverse-geocode?
key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
  "{
    "QueryPosition": [
        -97.721,  30.404
    ],
    "Filter": {
        "IncludePlaceTypes": [
            "PointAddress"
        ]
    }'
```

AWS CLI

```
aws geo-places reverse-geocode --key ${YourKey} --query-position -97.721 30.404 --
filter '{"IncludePlaceTypes": ["PointAddress"]}'
```

筛选街道

通过筛选`Street`，API 会返回没有特定地址号的街道级数据。

Sample Request

```
{
    "QueryPosition": [
        -97.721, 30.404
    ],
    "Filter": {
        "IncludePlaceTypes": [
            "Street"
        ]
    }
}
```

Sample Response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "Street",
            "Title": "Domain Blvd, Austin, TX 78758, United States",
            "Address": {
                "Label": "Domain Blvd, Austin, TX 78758, United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "TX",
                    "Name": "Texas"
                },
                "SubRegion": {
                    "Name": "Travis"
                },
                "Locality": "Austin",
                "District": "North Burnet",
                "PostalCode": "78758",
                "Street": "Domain Blvd",
```

```
                "StreetComponents": [
                    {
                        "BaseName": "Domain",
                        "Type": "Blvd",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ]
            },
            "Position": [
                -97.72103,
                30.40399
            ],
            "Distance": 3,
            "MapView": [
                -97.72219,
                30.40273,
                -97.72057,
                30.40649
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/reverse-geocode?
key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
  "{
    "QueryPosition": [
        -97.721,  30.404
    ],
    "Filter": {
        "IncludePlaceTypes": [
            "Street"
        ]
    }'
```

AWS CLI

```
aws geo-places reverse-geocode --key ${YourKey} --query-position -97.721 30.404 --
filter '{"IncludePlaceTypes": ["Street"]}'
```

筛选地点

通过筛选Locality，您可以检索更广泛的位置数据，包括城市名称。

Sample Request

```
{
    "QueryPosition": [
        -97.721, 30.404
    ],
    "Filter": {
        "IncludePlaceTypes": [
            "Locality"
        ]
    }
}
```

Sample Response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "Locality",
            "Title": "Austin, TX, United States",
            "Address": {
                "Label": "Austin, TX, United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "TX",
                    "Name": "Texas"
                },
                "SubRegion": {
```

```
                    "Name": "Travis"
                },
                "Locality": "Austin",
                "PostalCode": "78701"
            },
            "Position": [
                -97.74299,
                30.26759
            ],
            "Distance": 0,
            "MapView": [
                -98.06484,
                30.06592,
                -97.55914,
                30.51965
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/reverse-geocode?
key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
  "{
    "QueryPosition": [
        -97.721,  30.404
    ],
    "Filter": {
        "IncludePlaceTypes": [
            "Locality"
        ]
    }'
```

AWS CLI

```
aws geo-places reverse-geocode --key ${YourKey} --query-position -97.721 30.404 --
filter '{"IncludePlaceTypes": ["Locality"]}'
```

如何使用特定语言进行反向地理编码

该功能允许从 BCP47兼容代码中选择首选响应语言。它根据名称变体检测查询语言，并使用首选语言处理不匹配的标记和模棱两可的情况。如果没有请求语言，Pl aces API 会以官方国家/地区语言提供搜索结果，但会优先考虑不同区域语言的区域。作为后备策略，如果请求的语言中没有任何地址元素不可用，Plac es 将以默认语言 APIs 返回地址。

潜在用例

一个潜在的用例是本地化查询和/或结果。

示例

以特定语言获取结果

Sample request

```
{
    "QueryPosition": [
                139.69172,
                35.6895
            ],
    "Language": "JA"
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointAddress",
            "Title": "2-8-1 Nishishinjuku, Shinjuku-ku, Tokyo 160-0023, Japan",
            "Address": {
                "Label": "2-8-1 Nishishinjuku, Shinjuku-ku, Tokyo 160-0023, Japan",
                "Country": {
                    "Code2": "JP",
                    "Code3": "JPN",
                    "Name": "Japan"
                },
                "Region": {
                    "Name": "Tokyo"
                },
```

```
                "SubRegion": {},
                "Locality": "Shinjuku-ku",
                "SubDistrict": "Nishishinjuku",
                "PostalCode": "160-0023",
                "Block": "2 Chome",
                "SubBlock": "8",
                "AddressNumber": "1"
            },
            "Position": [
                139.69171,
                35.68949
            ],
            "Distance": 1,
            "MapView": [
                139.69071,
                35.68861,
                139.69251,
                35.69048
            ],
            "AccessPoints": [
                {
                    "Position": [
                        139.69206,
                        35.68954
                    ]
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/reverse-geocode?
key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryPosition": [
                139.69172,
                35.6895
            ],
    "Language": "JA"
```

```
}'
```

AWS CLI

```
aws geo-places reverse-geocode --key ${YourKey} --query-position 139.69172 35.6895
 --language "JA"
```

如何对城市的时区进行反向地理编码

您可以使用反向地理编码 API 来请求时区信息，例如 UTC 偏移量和时区名称

潜在用途

对时区进行地理编码的可能用途：

- 创建世界时钟
- 在不同地区安排会议

示例

示例

根据时区请求对东京的地点进行反向地理编码。

Sample request

```
{
    "QueryPosition": [
        139.69172,
        35.6895
    ],
    "AdditionalFeatures": [
        "TimeZone"
    ]
}
```

Sample response

```
{
    "ResultItems": [
        {
```

```
            "PlaceId": "<Redacted>",
            "PlaceType": "PointAddress",
            "Title": "#160-0023 #########2##8-1",
            "Address": {
                "Label": "#160-0023 #########2##8-1",
                "Country": {
                    "Code2": "JP",
                    "Code3": "JPN",
                    "Name": "##"
                },
                "Region": {
                    "Name": "###"
                },
                "SubRegion": {},
                "Locality": "###",
                "SubDistrict": "###",
                "PostalCode": "160-0023",
                "Block": "2##",
                "SubBlock": "8",
                "AddressNumber": "1"
            },
            "Position": [
                139.69171,
                35.68949
            ],
            "Distance": 1,
            "MapView": [
                139.69071,
                35.68861,
                139.69251,
                35.69048
            ],
            "AccessPoints": [
                {
                    "Position": [
                        139.69206,
                        35.68954
                    ]
                }
            ],
            "TimeZone": {
                "Name": "Asia/Tokyo",
                "Offset": "+09:00",
                "OffsetSeconds": 32400
```

```
            }
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/reverse-geocode?
key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryPosition": [
                139.69172,
                35.6895
            ],
    "AdditionalFeatures": [
                "TimeZone"
              ]
}'
```

AWS CLI

```
aws geo-places reverse-geocode --key ${YourKey} --query-position 139.69172 35.6895
 --additional-features "TimeZone"
```

如何从政治角度进行反向地理编码

Amazon Location Service 允许您指定政治观点，以确保您的申请符合当地法规。在对争议地区的位置进行地理编码时，此功能非常有用，因为这些地区的地名或边界可能因政治观点而异。

潜在用例

遵守地方政策：确保地名和边界符合所选政治观点的法律要求。

示例

对有争议的位置进行反向地理编码

Sample Request

```
{
```

```
    "QueryPosition": [
        33.95876,
        45.46824
    ],
    "PoliticalView": "RUS"
}
```

Sample Response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "Locality",
            "Title": "Первомайский район, Южный федеральный округ, Россия",
            "Address": {
                "Label": "Первомайский район, Южный федеральный округ, Россия",
                "Country": {
                    "Code2": "RU",
                    "Code3": "RUS",
                    "Name": "Россия"
                },
                "Region": {
                    "Name": "Южный федеральный округ"
                },
                "SubRegion": {
                    "Name": "Республика Крым"
                },
                "Locality": "Первомайский район"
            },
            "Position": [
                33.85939,
                45.7142
            ],
            "Distance": 0,
            "MapView": [
                33.52692,
                45.34303,
                34.12277,
                45.80953
            ],
            "PoliticalView": "RUS"
        }
```

```
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/reverse-geocode?
key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryPosition": [
        33.95876,
        45.46824
    ],
    "PoliticalView": "RUS"
}'
```

AWS CLI

```
aws geo-places reverse-geocode --key ${YourKey} --query-position 33.95876 45.46824
 --political-view "RUS"
```

如何获得十字路口

Geocode API 可以将附近的交叉路口检索到指定位置。

潜在用例

检索附近的所有十字路口。紧急服务部门和送货快递员可以使用。应急车辆在接听电话时通常需要识别附近的十字路口以获得最佳定位。这使他们能够保持清晰的访问路线和更快的响应时间，同时确保多种方法的可见性。同样，快递员可以利用十字路口数据来寻找更高效的停车位，尤其是在 door-to-door停车位可能有限或受限的密集城市地区。

获取附近的十字路口

Sample request

```
{
   "QueryPosition": [-123.11694, 49.28126],
   "AdditionalFeatures": ["Intersections"]
}
```

Sample response


```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>"
            "PlaceType": "PointAddress",
            "Title": "510 W Georgia St, Vancouver, BC V6B 0M3, Canada",
            "Address": {
                "Label": "510 W Georgia St, Vancouver, BC V6B 0M3, Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "District": "Downtown Vancouver",
                "PostalCode": "V6B 0M3",
                "Street": "W Georgia St",
                "StreetComponents": [
                    {
                        "BaseName": "Georgia",
                        "Type": "St",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "W",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "510"
            },
            "Position": [
                -123.11694,
                49.28126
            ],
            "Distance": 0,
            "MapView": [
```

```
                -123.11813,
                49.27786,
                -123.11076,
                49.28246
            ],
            "Intersections": [
                {
                    "PlaceId": "<Redacted>",
                    "Title": "W Georgia St & Richards St, Vancouver, BC V6B,
Canada",
                    "Address": {
                        "Label": "W Georgia St & Richards St, Vancouver, BC V6B,
Canada",
                        "PostalCode": "V6B",
                        "Intersection": [
                            "W Georgia St",
                            "Richards St"
                        ]
                    },
                    "Position": [
                        -123.11614,
                        49.28124
                    ],
                    "Distance": 58
                },
                {
                    "PlaceId": "<Redacted>",
                    "Title": "W Georgia St & Seymour St, Vancouver, BC V6B, Canada",
                    "Address": {
                        "Label": "W Georgia St & Seymour St, Vancouver, BC V6B,
Canada",
                        "PostalCode": "V6B",
                        "Intersection": [
                            "W Georgia St",
                            "W Georgia St",
                            "Seymour St",
                            "Seymour St"
                        ]
                    },
                    "Position": [
                        -123.11712,
                        49.28186
                    ],
                    "Distance": 68
```

```
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
   --url 'https://places.geo.eu-central-1.amazonaws.com/v2/reverse-geocode?
key=Your_Key' \
   --header 'Content-Type: application/json' \
   --data '{
      "QueryPosition": [
         -123.11694, 49.28126
      ],
      "AdditionalFeatures": ["Intersections"]
}
```

AWS CLI

```
aws geo-places reverse-geocode --key ${YourKey} --query-position -123.11694,
 49.28126 --additional-features "Intersections"
```

如何使用自动完成功能

本主题演示如何使用自动补全 API 根据用户输入有效地检索地址建议。它涵盖了两个关键应用程序：基本地址自动补全（在用户键入时建议地址）和经过筛选的自动补全，允许用户按区域或特定位置类型等标准缩小结果范围。这些示例为将自动完成功能集成到应用程序中以增强搜索相关性和用户体验提供了实用指导。

主题

- 如何填写地址
- 如何使用过滤器完成地址

如何填写地址

Autocomplete API 使您能够完成部分输入的地址，为最终用户提供标准化输入，并提高数据输入的效率。

潜在用例

- 在结账时填写地址：当客户输入结账表单时，便于准确、快速地输入地址。
- 填写特定国家/地区的地址：出于合规性或与用户所在位置的相关性考虑，将建议限制在特定国家/地区。

示例

用最少的数据完成地址

在此示例中，为了保持响应的简洁性和成本效益，返回的数据最少。以后可以使用 with GetPlace api 检索完整的地点详情。 PlaceId

Sample request

```
{
  "QueryText": "100 McCullum Rd"
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointAddress",
            "Title": "United Kingdom, E3 5JB, London, 100 McCullum Road"
        }
    ],
    "QueryRefinements": []
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/autocomplete?key=Your_Key'
 \
  --header 'Content-Type: application/json' \
  --data '{
  "QueryText": "100 McCullum Rd"
}'
```

AWS CLI

```
aws geo-places autocomplete --key ${YourKey} --query-text "100 McCullum Rd"
```

包含地址组成部分的完整地址

此示例返回多个地址建议，允许用户选择标准化地址格式来准确填充表单字段。

Sample request

```
{
  "QueryText": "100 McCullum Rd",
  "AdditionalFeatures": [
    "Core"
  ]
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointAddress",
            "Title": "United Kingdom, E3 5JB, London, 100 McCullum Road",
            "Address": {
                "Label": "100 McCullum Road, London, E3 5JB, United Kingdom",
                "Country": {
                    "Code2": "GB",
                    "Code3": "GBR",
                    "Name": "United Kingdom"
                },
                "Region": {
                    "Name": "England"
                },
                "SubRegion": {
                    "Code": "LDN",
                    "Name": "London"
                },
                "Locality": "London",
                "District": "Bow",
                "PostalCode": "E3 5JB",
```

```
                "Street": "McCullum Road",
                "StreetComponents": [
                    {
                        "BaseName": "McCullum",
                        "Type": "Road",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "100"
            },
            "Language": "en",
            "Highlights": {
                "Title": [
                    {
                        "StartIndex": 32,
                        "EndIndex": 35,
                        "Value": "100"
                    },
                    {
                        "StartIndex": 36,
                        "EndIndex": 49,
                        "Value": "McCullum Road"
                    }
                ],
                "Address": {
                    "Label": [
                        {
                            "StartIndex": 0,
                            "EndIndex": 3,
                            "Value": "100"
                        },
                        {
                            "StartIndex": 4,
                            "EndIndex": 17,
                            "Value": "McCullum Road"
                        }
                    ],
                    "Street": [
                        {
                            "StartIndex": 0,
                            "EndIndex": 13,
                            "Value": "McCullum Road"
```

```
                        }
                    ],
                    "AddressNumber": [
                        {
                            "StartIndex": 0,
                            "EndIndex": 3,
                            "Value": "100"
                        }
                    ]
                }
            }
        },
        ...
    ],
    "QueryRefinements": []
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/autocomplete?key=Your_Key'
 \
  --header 'Content-Type: application/json' \
  --data '{
  "QueryText": "100 McCullum Rd",
  "AdditionalFeatures": [
    "Core"
  ]
}'
```

AWS CLI

```
aws geo-places autocomplete --key ${YourKey} --query-text "100 McCullum Rd" --
additional-features "Core"
```

如何使用过滤器完成地址

Autocomplete API 使您能够完成部分输入的地址，从而便于最终用户快速、标准化地输入地址。通过使用`AdditionalFeatures`参数，您可以自定义响应中提供的信息以满足特定要求。

潜在用例

- 最大限度地减少数据以提高成本效益：在预计会有后续查询时，仅请求必要的地址组件，从而减少响应规模和数据传输成本。
- 包括必要的详细信息以供直接使用：检索全面的地址信息，无需进行其他查询。

示例

带国家/地区筛选器的完整地址

此示例应用国家/地区筛选器来优化结果，从而使用户输入的地址的标准化格式能够准确填充表单字段。

Sample request

```
{
  "QueryText": "100 McCullum Rd",
  "Filter": {
    "IncludeCountries": [
      "GBR"
    ]
  }
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointAddress",
            "Title": "United Kingdom, E3 5JB, London, 100 McCullum Road",
            "Address": {
                "Label": "100 McCullum Road, London, E3 5JB, United Kingdom",
                "Country": {
                    "Code2": "GB",
                    "Code3": "GBR",
                    "Name": "United Kingdom"
                },
                "Region": {
                    "Name": "England"
                },
```

```
                "SubRegion": {
                    "Code": "LDN",
                    "Name": "London"
                },
                "Locality": "London",
                "District": "Bow",
                "PostalCode": "E3 5JB",
                "Street": "McCullum Road",
                "StreetComponents": [
                    {
                        "BaseName": "McCullum",
                        "Type": "Road",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "100"
            },
            "Language": "en",
            "Highlights": {
                "Title": [
                    {
                        "StartIndex": 32,
                        "EndIndex": 35,
                        "Value": "100"
                    },
                    {
                        "StartIndex": 36,
                        "EndIndex": 49,
                        "Value": "McCullum Road"
                    }
                ],
                "Address": {
                    "Label": [
                        {
                            "StartIndex": 0,
                            "EndIndex": 3,
                            "Value": "100"
                        },
                        {
                            "StartIndex": 4,
                            "EndIndex": 17,
                            "Value": "McCullum Road"
```

```
                        }
                    ],
                    "Street": [
                        {
                            "StartIndex": 0,
                            "EndIndex": 13,
                            "Value": "McCullum Road"
                        }
                    ],
                    "AddressNumber": [
                        {
                            "StartIndex": 0,
                            "EndIndex": 3,
                            "Value": "100"
                        }
                    ]
                }
            }
        }
    ],
    "QueryRefinements": []
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/autocomplete?key=Your_Key'
 \
  --header 'Content-Type: application/json' \
  --data '{
  "QueryText": "100 McCullum Rd",
  "Filter": {
    "IncludeCountries": [
      "GBR"
    ]
  }
}'
```

AWS CLI

```
aws geo-places autocomplete --key ${YourKey} --query-text "100 McCullum Rd" \
--filter '{"IncludeCountries": ["GBR"]}'
```

完整的地址以及其他数据

此示例返回其他数据，允许在不进行后续查询的情况下使用返回的地址详细信息。对于这个用例来说，这Core组附加功能已经足够了。

Sample request

```
{
  "QueryText": "100 McCullum Rd",
  "Filter": {
    "IncludeCountries": [
      "GBR"
    ]
  },
  "AdditionalFeatures": [
    "Core"
  ]
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointAddress",
            "Title": "United Kingdom, E3 5JB, London, 100 McCullum Road",
            "Address": {
                "Label": "100 McCullum Road, London, E3 5JB, United Kingdom",
                "Country": {
                    "Code2": "GB",
                    "Code3": "GBR",
                    "Name": "United Kingdom"
                },
                "Region": {
                    "Name": "England"
                },
                "SubRegion": {
                    "Code": "LDN",
                    "Name": "London"
                },
                "Locality": "London",
                "District": "Bow",
```

```
                "PostalCode": "E3 5JB",
                "Street": "McCullum Road",
                "StreetComponents": [
                    {
                        "BaseName": "McCullum",
                        "Type": "Road",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "100"
            },
            "Language": "en",
            "Highlights": {
                "Title": [
                    {
                        "StartIndex": 32,
                        "EndIndex": 35,
                        "Value": "100"
                    },
                    {
                        "StartIndex": 36,
                        "EndIndex": 49,
                        "Value": "McCullum Road"
                    }
                ],
                "Address": {
                    "Label": [
                        {
                            "StartIndex": 0,
                            "EndIndex": 3,
                            "Value": "100"
                        },
                        {
                            "StartIndex": 4,
                            "EndIndex": 17,
                            "Value": "McCullum Road"
                        }
                    ],
                    "Street": [
                        {
                            "StartIndex": 0,
                            "EndIndex": 13,
```

```
                            "Value": "McCullum Road"
                        }
                    ],
                    "AddressNumber": [
                        {
                            "StartIndex": 0,
                            "EndIndex": 3,
                            "Value": "100"
                        }
                    ]
                }
            }
        }
    ],
    "QueryRefinements": []
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/autocomplete?key=Your_Key'
 \
  --header 'Content-Type: application/json' \
  --data '{
  "QueryText": "100 McCullum Rd",
  "Filter": {
    "IncludeCountries": [
      "GBR"
    ]
  },
  "AdditionalFeatures": [
    "Core"
  ]
}'
```

AWS CLI

```
aws geo-places autocomplete --key ${YourKey} --query-text "100 McCullum Rd" \
--additional-features "Core" \
--filter '{"IncludeCountries": ["GBR"]}'
```

如何使用 Get Place

本节包含有关如何使用 Get Place 的各种操作指南和示例 APIs。

示例

- 如何获得结果 PlaceId
- 如何获取联系人 PlaceId
- 如何获取 a 的时区 PlaceId
- 如何获取特定语言 PlaceId 的详细信息
- 如何获取地点 ID 的辅助地址

如何获得结果 PlaceId

GetPlace API 使用其唯一性检索有关地点的详细信息 PlaceId，从而允许应用程序访问有关指定位置的全面详细信息。

潜在用例

- 检索商店位置： PlaceId 用于获取有关各种商店位置的详细信息，以改善用户体验和有效的资源管理。
- 访问相关地点详情：获取使用先前查询 PlaceIds 返回的其他地点识别的地点的详细信息。

示例

用一个来找个地方 PlaceId

Sample request

```
https://places.geo.eu-central-1.amazonaws.com/v2/place/
AQAAAFUAcrFHu947JATTY9gIGcfNlNVzD3UftkkI9ayJjtquaC7IquYz-_FFnJnzJSQ7JePd-
sY0MSpA64V0w4aXLc-
lB2fZLJKk6uoAMSgtwvwxzg1fvPxFM9zXsx77EaLXarl7F4gSPTyQ6fiEnj0b0ipOXpnOoIsP?
key=Your_Key
```

Sample response

```
{
    "PlaceId": "<Redacted>",
    "PlaceType": "PointOfInterest",
```

```
    "Title": "London Eye",
    "Address": {
        "Label": "London Eye, County Hall, Westminster Bridge Road, London, SE1 7,
United Kingdom",
        "Country": {
            "Code2": "GB",
            "Code3": "GBR",
            "Name": "United Kingdom"
        },
        "Region": {
            "Name": "England"
        },
        "SubRegion": {
            "Code": "LDN",
            "Name": "London"
        },
        "Locality": "London",
        "District": "Waterloo",
        "PostalCode": "SE1 7",
        "Street": "County Hall, Westminster Bridge Road",
        "StreetComponents": [
            {
                "BaseName": "County Hall, Westminster Bridge Road",
                "Language": "en-GB"
            }
        ]
    },
    "Position": [
        -0.11953,
        51.50336
    ],
    "Categories": [
        {
            "Id": "tourist_attraction",
            "Name": "Tourist Attraction",
            "LocalizedName": "Tourist Attraction",
            "Primary": true
        },
        {
            "Id": "landmark-attraction",
            "Name": "Landmark-Attraction",
            "LocalizedName": "Landmark-Attraction",
            "Primary": false
        },
```

```
        {
            "Id": "residential_area-building",
            "Name": "Residential Area-Building",
            "LocalizedName": "Residential Area/Building",
            "Primary": false
        }
    ]
}
```

cURL

```
curl --request GET \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/place/
AQAAAFUAcrFHu947JATTY9gIGcfNlNVzD3UftkkI9ayJjtquaC7IquYz-_FFnJnzJSQ7JePd-
sY0MSpA64V0w4aXLc-
lB2fZLJKk6uoAMSgtwvwxzg1fvPxFM9zXsx77EaLXarl7F4gSPTyQ6fiEnj0b0ipOXpnOoIsP?
key=Your_Key&language=en'
```

AWS CLI

```
export PLACEID=AQAAAFUAcrFHu947JATTY9gIGcfNlNVzD3UftkkI9ayJjtquaC7IquYz-
_FFnJnzJSQ7JePd-sY0MSpA64V0w4aXLc-
lB2fZLJKk6uoAMSgtwvwxzg1fvPxFM9zXsx77EaLXarl7F4gSPTyQ6fiEnj0b0ipOXpnOoIs
aws geo-places get-place --key ${YourKey} --place-id ${PLACEID}
```

如何获取联系人 PlaceId

该 GetPlace API允许检索与特定联系人相关的联系信息 PlaceId，为您提供电话号码和网站等详细信息 URLs。

潜在用例

- 获取其他已保存的联系信息 PlaceId：通过访问联系人数据来增强存储的地点详细信息。
- 通过@@ 自动完成功能检索地址的联系方式：使用自动补全 API PlaceId 中的来获取特定的联系人详细信息。

示例

获取联系人以获取 PlaceId

Sample request

```
https://places.geo.eu-central-1.amazonaws.com/v2/place/
AQAAAFUAcrFHu947JATTY9gIGcfNlNVzD3UftkkI9ayJjtquaC7IquYz-_FFnJnzJSQ7JePd-
sY0MSpA64V0w4aXLc-
lB2fZLJKk6uoAMSgtwvwxzg1fvPxFM9zXsx77EaLXarl7F4gSPTyQ6fiEnj0b0ipOXpnOoIsP?
additional-features=Contact&key=Your_Key
```

Sample response

```
{
    "PlaceId": "AQAAAFUAcrFHu947JATTY9gIGcfNlNVzD3UftkkI9ayJjtquaC7IquYz-
_FFnJnzJSQ7JePd-sY0MSpA64V0w4aXLc-
lB2fZLJKk6uoAMSgtwvwxzg1fvPxFM9zXsx77EaLXarl7F4gSPTyQ6fiEnj0b0ipOXpnOoIsP",
    "PlaceType": "PointOfInterest",
    "Title": "London Eye",
    "Address": {
        "Label": "London Eye, County Hall, Westminster Bridge Road, London, SE1 7,
 United Kingdom",
        "Country": {
            "Code2": "GB",
            "Code3": "GBR",
            "Name": "United Kingdom"
        },
        "Region": {
            "Name": "England"
        },
        "SubRegion": {
            "Code": "LDN",
            "Name": "London"
        },
        "Locality": "London",
        "District": "Waterloo",
        "PostalCode": "SE1 7",
        "Street": "County Hall, Westminster Bridge Road",
        "StreetComponents": [
            {
                "BaseName": "County Hall, Westminster Bridge Road",
                "Language": "en-GB"
```

```
            }
        ]
    },
    "Position": [
        -0.11953,
        51.50336
    ],
    "Categories": [
        {
            "Id": "tourist_attraction",
            "Name": "Tourist Attraction",
            "LocalizedName": "Tourist Attraction",
            "Primary": true
        },
        {
            "Id": "landmark-attraction",
            "Name": "Landmark-Attraction",
            "LocalizedName": "Landmark-Attraction",
            "Primary": false
        },
        {
            "Id": "residential_area-building",
            "Name": "Residential Area-Building",
            "LocalizedName": "Residential Area/Building",
            "Primary": false
        }
    ],
    "Contacts": {
        "Phones": [
            {
                "Value": "+44870500600"
            },
            {
                "Value": "+448709908883"
            }
        ],
        "Websites": [
            {
                "Value": "http://www.londoneye.com"
            }
        ]
    }
}
```

cURL

```
curl --request GET \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/place/
AQAAAFUAcrFHu947JATTY9gIGcfNlNVzD3UftkkI9ayJjtquaC7IquYz-_FFnJnzJSQ7JePd-
sY0MSpA64V0w4aXLc-
lB2fZLJKk6uoAMSgtwvwxzg1fvPxFM9zXsx77EaLXarl7F4gSPTyQ6fiEnj0b0ipOXpnOoIsP?
key=Your_Key&additional-features=Contact&language=en'
```

AWS CLI

```
export PLACEID=AQAAAFUAcrFHu947JATTY9gIGcfNlNVzD3UftkkI9ayJjtquaC7IquYz-
_FFnJnzJSQ7JePd-sY0MSpA64V0w4aXLc-
lB2fZLJKk6uoAMSgtwvwxzg1fvPxFM9zXsx77EaLXarl7F4gSPTyQ6fiEnj0b0ipOXpnOoIsP
aws geo-places get-place --key ${YourKey} --place-id ${PLACEID} --additional-
features "Contact" --language "en"
```

如何获取 a 的时区 PlaceId

该 GetPlace API 允许检索与特定联系人相关的联系信息 PlaceId，为您提供电话号码、时区和网站等详细信息 URLs。

潜在用例

- 获取保存的更多时区信息 PlaceId：使用商店创建世界时钟 PlaceId。
- 从 “自动完成” 中检索地址的时区详细信息:跟踪夏令时. PlaceId

示例

获取 a 的时区 PlaceId

Sample request

```
https://places.geo.eu-central-1.amazonaws.com/v2/place/
AQAAAFUAcrFHu947JATTY9gIGcfNlNVzD3UftkkI9ayJjtquaC7IquYz-_FFnJnzJSQ7JePd-
sY0MSpA64V0w4aXLc-
lB2fZLJKk6uoAMSgtwvwxzg1fvPxFM9zXsx77EaLXarl7F4gSPTyQ6fiEnj0b0ipOXpnOoIsP?
additional-features=TimeZone&key=Your_Key
```

Sample response


```
{
    "PlaceId": "<Redacted>",
    "PlaceType": "PointOfInterest",
    "Title": "London Eye",
    "Address": {
        "Label": "London Eye, County Hall, Westminster Bridge Road, London, SE1 7,
 United Kingdom",
        "Country": {
            "Code2": "GB",
            "Code3": "GBR",
            "Name": "United Kingdom"
        },
        "Region": {
            "Name": "England"
        },
        "SubRegion": {
            "Code": "LDN",
            "Name": "London"
        },
        "Locality": "London",
        "District": "Waterloo",
        "PostalCode": "SE1 7",
        "Street": "County Hall, Westminster Bridge Road",
        "StreetComponents": [
            {
                "BaseName": "County Hall, Westminster Bridge Road",
                "Language": "en-GB"
            }
        ]
    },
    "Position": [
        -0.11953,
        51.50336
    ],
    "Categories": [
        {
            "Id": "tourist_attraction",
            "Name": "Tourist Attraction",
            "LocalizedName": "Tourist Attraction",
            "Primary": true
        },
        {
```

```
            "Id": "landmark-attraction",
            "Name": "Landmark-Attraction",
            "LocalizedName": "Landmark-Attraction",
            "Primary": false
        },
        {
            "Id": "residential_area-building",
            "Name": "Residential Area-Building",
            "LocalizedName": "Residential Area/Building",
            "Primary": false
        }
    ],
    "TimeZone": {
        "Name": "Europe/London",
        "Offset": "+01:00",
        "OffsetSeconds": 3600
    }
}
```

cURL

```
curl --request GET \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/place/
AQAAAFUAcrFHu947JATTY9gIGcfNlNVzD3UftkkI9ayJjtquaC7IquYz-_FFnJnzJSQ7JePd-
sY0MSpA64V0w4aXLc-
lB2fZLJKk6uoAMSgtwvwxzg1fvPxFM9zXsx77EaLXarl7F4gSPTyQ6fiEnj0b0ipOXpnOoIsP?
key=Your_Key&=TimeZone'
```

AWS CLI

```
export PLACEID=AQAAAFUAcrFHu947JATTY9gIGcfNlNVzD3UftkkI9ayJjtquaC7IquYz-
_FFnJnzJSQ7JePd-sY0MSpA64V0w4aXLc-
lB2fZLJKk6uoAMSgtwvwxzg1fvPxFM9zXsx77EaLXarl7F4gSPTyQ6fiEnj0b0ipOXpnOoIsP

aws geo-places get-place --key ${YourKey} \
--place-id ${PLACEID} \
--additional-features "TimeZone"
```

如何获取特定语言 PlaceId 的详细信息

该功能允许您从 BCP47符合标准的代码中选择首选的响应语言。它根据名称变体检测查询语言，并使用首选语言处理不匹配的标记和模棱两可的情况。如果未注明所请求的语言，Places API 会以该国家/

地区使用的任何语言提供搜索结果，但它会优先考虑区域语言不同的区域。作为后备方案，当某些地址元素在请求的语言中不可用时，Places 会以默认语言 APIs 返回地址。

潜在用例

- 添加 PlaceId 详情：以指定语言为存储的地点 ID 添加更多详情。
- 从 “自动完成” 中检索地址的 PlaceId 详细信息：以特定语言获取存储 PlaceId地址的更多详细信息。

示例

获取特定语言 PlaceId 的 a 的详细信息

Sample request

```
https://places.geo.eu-central-1.amazonaws.com/v2/place/
AQAAAFUAY0R_4qfQ9LZ0j6lpOggbNLAQ31TRf-
sESER_bKKjCar9FF6A3UA0HrYWa4yfeUN5V0qkk6NmdrI3y7fB7PZ4vfuo-Z8Wd-
u-01an4KNvWaqfYmEh14s22yCV9Nb1yMXl4-HTfpX5D-jWQT14FEIBqoiuKwLq?
language=EN&key=Your_Key
```

Sample response

```
{
    "PlaceId": "<Redacted>",
    "PlaceType": "PointOfInterest",
    "Title": "Parking Area",
    "Address": {
        "Label": "Parking Area, Al Nahda Sharjah, United Arab Emirates",
        "Country": {
            "Code2": "AE",
            "Code3": "ARE",
            "Name": "United Arab Emirates"
        },
        "SubRegion": {
            "Name": "Sharjah"
        },
        "Locality": "Sharjah",
        "District": "Al Nahda"
    },
    "Position": [
        55.3733,
        25.30388
```

```
    ],
    "Categories": [
        {
            "Id": "parking_lot",
            "Name": "Parking Lot",
            "LocalizedName": "Parking Lot",
            "Primary": true
        }
    ],
    "Contacts": {
        "Phones": [
            {
                "Value": "+971507766189"
            }
        ]
    },
    "AccessPoints": [
        {
            "Position": [
                55.37348,
                25.30398
            ]
        }
    ]
}
```

cURL

```
curl --request GET \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/place/
AQAAAFUAY0R_4qfQ9LZ0j6lpOggbNLAQ31TRf-
sESER_bKKjCar9FF6A3UA0HrYWa4yfeUN5V0qkk6NmdrI3y7fB7PZ4vfuo-Z8Wd-
u-01an4KNvWaqfYmEh14s22yCV9Nb1yMXl4-HTfpX5D-jWQT14FEIBqoiuKwLq?
language=EN&key=Your_Key`
```

AWS CLI

```
export PLACEID=AQAAAFUAcrFHu947JATTY9gIGcfNlNVzD3UftkkI9ayJjtquaC7IquYz-
_FFnJnzJSQ7JePd-sY0MSpA64V0w4aXLc-
lB2fZLJKk6uoAMSgtwvwxzg1fvPxFM9zXsx77EaLXarl7F4gSPTyQ6fiEnj0b0ipOXpnOoIsP

aws geo-places get-place --key ${YourKey} \
--place-id ${PLACEID} \
```

```
--language "en"
```

如何获取地点 ID 的辅助地址

SecondaryAddressesAPI 命令允许您检索主地址下的所有辅助地址。Geocode 还会返回QueryText参数中存在的辅助单位（如果有）。

潜在用例

检索主地址下的所有辅助地址。这可用于地址表填写用例，以选择更准确的辅助地址，其中还包括更准确的位置信息。

获取地点 ID 的辅助地址

Sample request

```
https://places.geo.eu-central-1.amazonaws.com/v2/place/
AQAAAGAAY8Dw_tn6pyN_R5E9_mfwAP5h9Rd5E_xY2CyyRHpX2wolGeK8O3IY13k5xeHmF4To445fCLg0x3GPnTZnCuUW
xf6jNC_3CUSabHVKy-Or51xHTd3KNi6idTkrfeQOjZOkXZ5TaoXsg0GHGMWqs-sZhmr8npQc?additional-
features=SecondaryAddresses&key=Your_Key
```

Sample response

```
{
    "PlaceId": "<Redacted>",
    "PlaceType": "PointAddress",
    "Title": "910 Beach Ave, Vancouver, BC V6Z 2W7, Canada",
    "Address": {
        "Label": "910 Beach Ave, Vancouver, BC V6Z 2W7, Canada",
        "Country": {
            "Code2": "CA",
            "Code3": "CAN",
            "Name": "Canada"
        },
        "Region": {
            "Code": "BC",
            "Name": "British Columbia"
        },
        "SubRegion": {
            "Name": "Metro Vancouver"
        },
        "Locality": "Vancouver",
```

```
        "District": "Downtown Vancouver",
        "PostalCode": "V6Z 2W7",
        "Street": "Beach Ave",
        "StreetComponents": [
            {
                "BaseName": "Beach",
                "Type": "Ave",
                "TypePlacement": "AfterBaseName",
                "TypeSeparator": " ",
                "Language": "en"
            }
        ],
        "AddressNumber": "910"
    },
    "Position": [
        -123.13325,
        49.27542
    ],
    "MapView": [
        -123.13377,
        49.2741,
        -123.13087,
        49.27599
    ],
    "SecondaryAddresses": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "SecondaryAddress",
            "Title": "101-910 Beach Ave, Vancouver, BC V6Z 2W7, Canada",
            "Address": {
                "Label": "101-910 Beach Ave, Vancouver, BC V6Z 2W7, Canada",
                "SecondaryAddressComponents": [
                    {
                        "Number": "101"
                    }
                ]
            },
            "Position": [
                -123.1334,
                49.27532
            ]
        },
        {
            "PlaceId": "<Redacted>",
```

```
            "PlaceType": "SecondaryAddress",
            "Title": "102-910 Beach Ave, Vancouver, BC V6Z 2W7, Canada",
            "Address": {
                "Label": "102-910 Beach Ave, Vancouver, BC V6Z 2W7, Canada",
                "SecondaryAddressComponents": [
                    {
                        "Number": "102"
                    }
                ]
            },
            "Position": [
                -123.1334,
                49.27532
            ]
        },
        ...
    ]
}
```

cURL

```
curl --request GET \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/place/
AQAAAGAAY8Dw_tn6pyN_R5E9_mfwAP5h9Rd5E_xY2CyyRHpX2wolGeK8O3IY13k5xeHmF4To445fCLg0x3GPnTZnCuUW
xf6jNC_3CUSabHVKy-Or51xHTd3KNi6idTkrfeQOjZOkXZ5TaoXsg0GHGMWqs-sZhmr8npQc?
key=Your_Key&additional-features=SecondaryAddresses'
```

AWS CLI

```
export
 PLACEID=AQAAAGAAY8Dw_tn6pyN_R5E9_mfwAP5h9Rd5E_xY2CyyRHpX2wolGeK8O3IY13k5xeHmF4To445fCLg0x3G
xf6jNC_3CUSabHVKy-Or51xHTd3KNi6idTkrfeQOjZOkXZ5TaoXsg0GHGMWqs-sZhmr8npQc

aws geo-places get-place --key ${YourKey} \
--place-id ${PLACEID} \
--additional-features "SecondaryAddresses"
```

如何使用 SearchNearby

本节包含各种操作指南和使用示例 SearchNearby。

示例

- 如何从某个位置搜索附近
- 如何根据类别搜索附近的地点
- 如何根据食物类型搜索附近的地方
- 如何根据业务链在附近进行搜索
- 如何在某个国家/地区附近搜寻
- 如何使用地理空间上下文在附近进行搜索

如何从某个位置搜索附近

借助 SearchNearby API，无需输入任何特定文本，即可查询附近的所有地点和兴趣点 (POI)。用户可以使用此 API 探索社区 POIs、发现社区等。它需要一个`QueryPosition`，它可以表示设备的位置、基于 IP 的位置或地图视口中心。或者，用户可以指定城市或地点，根据该位置的地理坐标对结果进行偏差。

潜在用例

- 探索附近 POIs：查看当前位置附近的所有景点。
- 探索附近的地方：查看给定位置附近的所有位置或地点。

示例

从某个位置搜索附近

在本示例中，搜索是从迪拜的一个位置进行的，其纬度为25.26951，经度为55.30884。

Sample request

```
{
    "QueryPosition": [
        55.30884,
        25.26951
    ]
}
```

Sample response

```
{
    "ResultItems": [
        {
```

```
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "###",
            "Address": {
                "Label": "####### ######## ,### #### ,##### ## ###### #### ,###
####### ",
                "Country": {
                    "Code2": "AE",
                    "Code3": "ARE",
                    "Name": "####### ####### ########"
                },
                "SubRegion": {
                    "Name": "###"
                },
                "Locality": "###",
                "District": "####",
                "Street": "##### ## ###### ####",
                "StreetComponents": [
                    {
                        "BaseName": "##### ## ######",
                        "Type": "####",
                        "TypePlacement": "BeforeBaseName",
                        "TypeSeparator": " ",
                        "Language": "ar"
                    }
                ]
            },
            "Position": [
                55.30884,
                25.26951
            ],
            "Distance": 0,
            "Categories": [
                {
                    "Id": "department_store",
                    "Name": "Department Store",
                    "LocalizedName": "##### ###",
                    "Primary": true
                }
            ],
            "BusinessChains": [
                {
                    "Name": "HEMA",
                    "Id": "HEMA"
```

```
                }
            ]
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "Locality",
            "Title": "####### ####### ######## ,###",
            "Address": {
                "Label": "####### ####### ######## ,###",
                "Country": {
                    "Code2": "AE",
                    "Code3": "ARE",
                    "Name": "####### ####### ########"
                },
                "SubRegion": {
                    "Name": "###"
                },
                "Locality": "###"
            },
            "Position": [
                55.30884,
                25.26951
            ],
            "Distance": 0,
            "MapView": [
                54.64906,
                24.62308,
                55.7371,
                25.36995
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/search-nearby?
key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryPosition": [
        55.30884,
```

```
        25.26951
    ],
    "MaxResults": 2
}'
```

AWS CLI

```
aws geo-places search-nearby --key ${YourKey} \
--query-position 55.30884 25.26951 \
--max-results 2
```

如何根据类别搜索附近的地点

该 SearchNearby API 允许在包含或排除指定类别的情况下查询兴趣点 (POI)。这可以帮助用户探索社区、探索当地 POIs景点等。API 需要一个`QueryPosition`，它可以基于设备的位置、IP 位置或地图视口的中心。或者，用户可以指定城市或地点，应用程序将根据该位置的坐标对结果进行偏差。

要了解有关支持的类别的更多信息，请参阅the section called “类别过滤器”。

潜在用例

- 探索当地设施：查找附近可用的设施。
- 探索旅游景点：确定城市内的旅游景点。
- 计划城市旅行：组织所选城市 POIs 内不同城市的旅行。

示例

包括类别

Sample request

```
{
    "QueryPosition": [
        4.35609,
        50.84439
    ],
    "Filter": {
        "IncludeCategories": ["airport"]
    }
}
```

Sample response


```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Brussels Airport",
            "Address": {
                "Label": "Brussels Airport, A201, 1930 Zaventem, België",
                "Country": {
                    "Code2": "BE",
                    "Code3": "BEL",
                    "Name": "België"
                },
                "Region": {
                    "Code": "VLG",
                    "Name": "Vlaanderen"
                },
                "SubRegion": {
                    "Name": "Vlaams Brabant"
                },
                "Locality": "Zaventem",
                "PostalCode": "1930",
                "Street": "A201",
                "StreetComponents": [
                    {
                        "BaseName": "A201",
                        "Language": "nl"
                    }
                ]
            },
            "Position": [
                4.47767,
                50.89452
            ],
            "Distance": 10191,
            "Categories": [
                {
                    "Id": "airport",
                    "Name": "Airport",
                    "LocalizedName": "Luchthaven",
                    "Primary": true
                }
```

```
            ]
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Internationale Luchthaven Antwerpen",
            "Address": {
                "Label": "Internationale Luchthaven Antwerpen, Luchthavenlei 1, 2100
Antwerpen, België",
                "Country": {
                    "Code2": "BE",
                    "Code3": "BEL",
                    "Name": "België"
                },
                "Region": {
                    "Code": "VLG",
                    "Name": "Vlaanderen"
                },
                "SubRegion": {
                    "Name": "Antwerpen"
                },
                "Locality": "Antwerpen",
                "District": "Deurne",
                "PostalCode": "2100",
                "Street": "Luchthavenlei",
                "StreetComponents": [
                    {
                        "BaseName": "Luchthaven",
                        "Type": "lei",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": "",
                        "Language": "nl"
                    }
                ],
                "AddressNumber": "1"
            },
            "Position": [
                4.45083,
                51.18867
            ],
            "Distance": 38852,
            "Categories": [
                {
                    "Id": "airport",
```

```
                    "Name": "Airport",
                    "LocalizedName": "Luchthaven",
                    "Primary": true
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/search-nearby?
key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryPosition": [
        4.35609,
        50.84439
    ],
    "Filter": {
        "IncludeCategories": ["airport"]
    },
    "MaxResults": 2
}
```

AWS CLI

```
aws geo-places search-nearby --key ${YourKey} \
--query-position 4.35609 50.84439 \
--filter '{"IncludeCategories": ["airport"]}' \
--max-results 2
```

排除类别

Sample request

```
{
    "QueryPosition": [
        4.35609,
        50.84439
    ],
```

```
    "Filter": {
        "ExcludeCategories": ["airport"]
    }
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Socialbrands Module 2",
            "Address": {
                "Label": "Socialbrands Module 2, Albertinaplein, 1000 Brussel,
 België",
                "Country": {
                    "Code2": "BE",
                    "Code3": "BEL",
                    "Name": "België"
                },
                "Region": {
                    "Code": "BRU",
                    "Name": "Brussel"
                },
                "SubRegion": {
                    "Name": "Brussel"
                },
                "Locality": "Brussel",
                "District": "Koningswijk",
                "PostalCode": "1000",
                "Street": "Albertinaplein",
                "StreetComponents": [
                    {
                        "BaseName": "Albertina",
                        "Type": "plein",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": "",
                        "Language": "nl"
                    }
                ]
            },
            "Position": [
```

```
                4.35609,
                50.84439
            ],
            "Distance": 0,
            "Categories": [
                {
                    "Id": "commercial_services",
                    "Name": "Commercial Services",
                    "LocalizedName": "Commerciële diensten",
                    "Primary": true
                }
            ],
            "Contacts": {
                "Websites": [
                    {
                        "Value": "https://oneread.net"
                    }
                ]
            },
            "AccessPoints": [
                {
                    "Position": [
                        4.35609,
                        50.84439
                    ]
                }
            ]
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Barman Privé",
            "Address": {
                "Label": "Barman Privé, Albertinaplein, 1000 Brussel, België",
                "Country": {
                    "Code2": "BE",
                    "Code3": "BEL",
                    "Name": "België"
                },
                "Region": {
                    "Code": "BRU",
                    "Name": "Brussel"
                },
                "SubRegion": {
```

```
                    "Name": "Brussel"
                },
                "Locality": "Brussel",
                "District": "Koningswijk",
                "PostalCode": "1000",
                "Street": "Albertinaplein",
                "StreetComponents": [
                    {
                        "BaseName": "Albertina",
                        "Type": "plein",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": "",
                        "Language": "nl"
                    }
                ]
            },
            "Position": [
                4.35609,
                50.84439
            ],
            "Distance": 0,
            "Categories": [
                {
                    "Id": "catering_and_other_food_services",
                    "Name": "Catering and Other Food Services",
                    "LocalizedName": "Catering- en horecadiensten",
                    "Primary": true
                }
            ],
            "Contacts": {
                "Phones": [
                    {
                        "Value": "+32476891634"
                    }
                ]
            },
            "OpeningHours": [
                {
                    "Display": [
                        "ma-zo: 00:00 - 24:00"
                    ],
                    "OpenNow": true,
                    "Components": [
                        {
```

```
                            "OpenTime": "T000000",
                            "OpenDuration": "PT24H00M",
                            "Recurrence": "FREQ:DAILY;BYDAY:MO,TU,WE,TH,FR,SA,SU"
                        }
                    ]
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        4.35609,
                        50.84439
                    ]
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/search-nearby?
key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryPosition": [
        4.35609,
        50.84439
    ],
    "Filter": {
        "ExcludeCategories": ["airport"]
    }
}
```

AWS CLI

```
aws geo-places search-nearby --key ${YourKey} \
--query-position 4.35609 50.84439 \
--filter '{"ExcludeCategories": ["airport"]}' \
--max-results 2
```

如何根据食物类型搜索附近的地方

通过 SearchNearby API，您可以查询附近供应特定类型食物的餐厅。您也可以从结果中排除食物类型。

您可以使用 SearchNearby API 让最终用户探索社区并发现名胜古迹。API 要求 QueryPosition，可以是设备位置、IP 位置或地图的视口中心。另一种方法是让最终用户提供城市名称或地名，应用程序可以根据地理坐标对结果进行偏差。

有关食物类型的更多信息，请参阅食物类型过滤器。

潜在用例

- 探索附近可用的食物种类。
- 不包括供应您过敏食物类型的餐厅。

示例

按食物类型搜索附近的地方

以下示例演示如何根据食物类型搜索附近的地点。

Sample request

此请求搜索在指定坐标附近供应中餐的餐厅。

```
{
    "QueryPosition": [
        12.49563,
        41.90325
    ],
    "Filter": {
        "IncludeFoodTypes": ["chinese"]
    }
}
```

Sample response

```
{
    "ResultItems": [
        {
```

```
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Bufala e Pachino",
            "Address": {
                "Label": "Bufala e Pachino, Via Firenze, 53, 00184 Roma RM, Italia",
                "Country": {
                    "Code2": "IT",
                    "Code3": "ITA",
                    "Name": "Italia"
                }
            },
            "Position": [
                12.49409,
                41.90237
            ],
            "Distance": 161,
            "Categories": [
                {
                    "Id": "restaurant",
                    "Name": "Restaurant",
                    "LocalizedName": "Ristorante",
                    "Primary": true
                }
            ],
            "FoodTypes": [
                {
                    "LocalizedName": "Pizza",
                    "Id": "pizza",
                    "Primary": true
                },
                {
                    "LocalizedName": "Cinese",
                    "Id": "chinese",
                    "Primary": false
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
```

```
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/search-nearby?
key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryPosition": [12.49563, 41.90325],
    "Filter": {
        "IncludeFoodTypes": ["chinese"]
    }
}'
```

AWS CLI

```
aws geo-places search-nearby --key ${YourKey} \
--query-position 12.49563 41.90325 \
--filter '{"IncludeFoodTypes": ["chinese"]}'
```

开发者小贴士

- 您可以使用ExludeFoodTypes将某些业务链排除在结果之外。
- 您可以排除或包含多种食物类型。

如何根据业务链在附近进行搜索

SearchNearby API 允许您查询附近的业务链。您可以在搜索中包含或排除特定的业务链。此功能允许最终用户探索社区、发现兴趣点等。

要使用 SearchNearby API，您需要提供一个 QueryPosition，它可以是：

- 设备位置
- 基于知识产权的职位
- 地图的视口中心

或者，用户可以提供城市名称或地点，应用程序可以根据该位置的地理坐标对结果进行偏差。

有关支持的业务链的更多信息，请参阅商业链过滤器。

潜在用例

- 探索附近的企业

- 通过寻找附近的 B2B 客户来扩展您的业务

示例

按连锁店搜索附近的地点

以下示例演示如何搜索附近的星巴克门店。

Sample request

此请求搜索指定坐标附近的星巴克地点。

```
{
    "QueryPosition": [
        12.49563,
        41.90325
    ],
    "Filter" : {
        "IncludeBusinessChains": ["Starbucks"]
    }
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Starbucks",
            "Address": {
                "Label": "Starbucks, Via Giovanni Giolitti, 2, 00185 Roma RM,
 Italia",
                "Country": {
                    "Code2": "IT",
                    "Code3": "ITA",
                    "Name": "Italia"
                },
                "Region": {
                    "Name": "Lazio"
                },
                "SubRegion": {
                    "Code": "RM",
```

```
                    "Name": "Roma"
                },
                "Locality": "Roma",
                "District": "Esquilino",
                 "PostalCode": "00185",
                "Street": "Via Giovanni Giolitti",
                "StreetComponents": [
                    {
                        "BaseName": "Giovanni Giolitti",
                        "Type": "Via",
                        "TypePlacement": "BeforeBaseName",
                        "TypeSeparator": " ",
                        "Language": "it"
                    }
                ],
                "AddressNumber": "2"
            },
            "Position": [
                12.50102,
                41.90093
            ],
            "Distance": 515,
            "Categories": [
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Bar",
                    "Primary": true
                }
            ],
            "BusinessChains": [
                {
                    "Name": "Starbucks",
                    "Id": "Starbucks"
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
```

```
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/search-nearby?
key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryPosition": [
                12.49563,
                41.90325
        ],
        "Filter" : {
        "IncludeBusinessChains": ["Starbucks"]
    }
}'
```

AWS CLI

```
aws geo-places search-nearby --key ${YourKey} \
--query-position 12.49563 41.90325 \
--filter '{"IncludeBusinessChains": ["Starbucks"]}'
```

开发者小贴士

- 您可以使用ExcludeBusinessChains将某些业务链排除在结果之外。
- 您可以排除或包括多个业务链。

如何在某个国家/地区附近搜寻

通过 SearchNearby API，您可以搜索特定国家/地区内附近的地点。要使用 API，您需要提供一个 QueryPosition，它可以是：

- 设备位置
- 基于知识产权的职位
- 地图的视口中心

或者，用户可以提供城市名称或地点，应用程序可以根据该位置的地理坐标对结果进行偏差。

潜在用例

- 探索一个国家内的企业
- 查找一个国家内附近的 B2B 客户

- 探索一个国家的旅游胜地

示例

在加拿大搜索高尔夫球场

以下示例演示如何搜索加拿大境内靠近指定坐标的高尔夫球场。

Sample request

此请求搜索加拿大境内指定地点附近的高尔夫球场。

```
{
    "QueryPosition": [
            -122.741803,
            49.002478
    ],
    "Filter" : {
        "IncludeCategories": ["golf_course"],
        "IncludeCountries": ["CAN"]
    }
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Peace Portal Golf Course",
            "Address": {
                "Label": "Peace Portal Golf Course, 16900 * Ave, Surrey, BC V*Z *P*,
 Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
```

```
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Surrey",
                "District": "South Surrey",
                "PostalCode": "V*Z *P*",
                "Street": "* Ave",
                "StreetComponents": [
                    {
                        "BaseName": "4",
                        "Type": "Ave",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "16900"
            },
            "Position": [
                -122.75086,
                49.00921
            ],
            "Distance": 998,
            "Categories": [
                {
                    "Id": "golf_course",
                    "Name": "Golf Course",
                    "LocalizedName": "Golf Course",
                    "Primary": true
                }
            ],
            "Contacts": {
                "Phones": [
                    {
                        "Value": "+16045384818"                    }
                ],
                "Websites": [
                    {
                        "Value": "http://www.peaceportalgolf.com"
                    }
                ]
            },
            "AccessPoints": [
                {
```

```
                    "Position": [
                        -122.75087,
                        49.00935
                    ]
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/search-nearby?
key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryPosition": [
         -122.741803,
         49.002478
        ],
        "Filter" : {
        "IncludeCategories": ["golf_course"],
              "IncludeCountries": ["CAN"]
    }

}'
```

AWS CLI

```
aws geo-places search-nearby --key ${YourKey} \
--query-position -122.741803 49.002478 \
--filter '{"IncludeCategories": ["golf_course"], "IncludeCountries": ["CAN"]}'
```

如何使用地理空间上下文在附近进行搜索

该 SearchNearby API 使您能够在附近进行探索，同时限制在地理空间上下文（例如圆形或边界框）内的结果。

SearchNearby API 允许您使用圆圈和边界框等地理空间上下文筛选结果。通过定义这些地理边界，您可以将搜索结果限制在特定区域。

潜在用例

- 将搜索结果限制在特定的地理区域
- 在自定义边界内搜索
- 将结果集中在目标社区或地区

示例

使用查询半径进行搜索

以下示例演示如何搜索指定坐标半径 1000 米内的学校。

Sample request

此请求搜索半径 1000 米（1 千米）范围内的学校。

```
{
    "QueryPosition": [
                -122.741803,
         49.002478
        ],
        "QueryRadius":1000,
        "Filter" : {
            "IncludeCategories": ["school"]
    }

}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Grace Lutheran Church",
            "Address": {
                "Label": "Grace Lutheran Church, 702 G St, Blaine, WA 98230-5125,
 United States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
```

```
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "Whatcom"
                },
                "Locality": "Blaine",
                "PostalCode": "98230-5125",
                "Street": "G St",
                "StreetComponents": [
                    {
                        "BaseName": "G",
                        "Type": "St",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "702"
            },
            "Position": [
                -122.74157,
                48.99533
            ],
            "Distance": 795,
            "Categories": [
                {
                    "Id": "church",
                    "Name": "Church",
                    "LocalizedName": "Church",
                    "Primary": true
                },
                {
                    "Id": "kindergarten_and_childcare",
                    "Name": "Kindergarten and Childcare",
                    "LocalizedName": "Kindergarten & Childcare",
                    "Primary": false
                },
                {
                    "Id": "school",
                    "Name": "School",
```

```
                    "LocalizedName": "School",
                    "Primary": false
                }
            ]
        }
        ...
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/searearch-nearby?
key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryPosition": [
         -122.741803,
         49.002478
    ],
    "QueryRadius": 1000,
    "Filter": {
        "IncludeCategories": ["school"]
    }
}'
```

AWS CLI

```
aws geo-places search-nearby --key ${YourKey} \
--query-position -122.741803 49.002478 \
--query-radius 1000 \
--filter '{"IncludeCategories": ["school"]}'
```

开发者小贴士

- 或者，您可以使用边界框过滤器。

如何使用 SearchText

本节包含各种操作指南和使用示例 SearchText APIs。

示例

- 如何使用名称搜索地点、POI 或企业
- 如何使用联系信息搜索地点
- 如何使用类别名称或食物类型进行搜索
- 如何搜索地址
- 如何在某个国家内搜索地点
- 如何搜索联系方式和营业时间
- 如何使用查询 ID 搜索地点
- 如何使用特定语言进行搜索

如何使用名称搜索地点、POI 或企业

该 SearchText API 允许用户使用自由文本输入按名称搜索地点、POI 或企业。可以通过设置偏差位置来细化结果，偏差位置可能基于设备位置、IP 位置或地图视口中心。或者，用户可以提供特定的城市或地点，结果会根据所提供的地理坐标进行偏差。

潜在用例

- 按名称查找地点：根据地名检索位置，例如 “Gas Town”。
- 查找兴趣点 (POIs)：按名称搜索名胜古迹，例如 “斯坦利公园”。
- 按企业名称搜索：按名称查找企业，例如 “星巴克”。

示例

按地名搜索

Sample request

```
{
    "QueryText": "Gas Town",
    "BiasPosition": [
        -123.11336,
        49.26038
    ]
}
```

Sample response


```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "District",
            "Title": "Gastown, Vancouver, BC, Canada",
            "Address": {
                "Label": "Gastown, Vancouver, BC, Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "District": "Gastown",
                "PostalCode": "V6B"
            },
            "Position": [
                -123.10647,
                49.28363
            ],
            "Distance": 2633,
            "MapView": [
                -123.11193,
                49.28141,
                -123.10217,
                49.28648
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/search-text?key=Your_Key'
 \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryText": "Gas Town",
    "BiasPosition": [
        -123.11336,
        49.26038
    ]
}'
```

AWS CLI

```
aws geo-places search-text --key ${YourKey} --query-text "Gas Town" --bias-position
 -123.11336 49.26038
```

按 POI 名称搜索

Sample request

```
{
    "QueryText": "Stanley Park",
    "BiasPosition": [
        -123.11336,
        49.26038
    ]
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Stanley Park",
            "Address": {
                "Label": "Stanley Park, Stanley Park Dr, Vancouver, BC V6G, Canada",
```

```
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "District": "Stanley Park",
                "PostalCode": "V6G",
                "Street": "Stanley Park Dr",
                "StreetComponents": [
                    {
                        "BaseName": "Stanley Park",
                        "Type": "Dr",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ]
            },
            "Position": [
                -123.13593,
                49.29716
            ],
            "Distance": 4405,
            "Categories": [
                {
                    "Id": "park-recreation_area",
                    "Name": "Park-Recreation Area",
                    "LocalizedName": "Park-Recreation Area",
                    "Primary": true
                },
                {
                    "Id": "tourist_attraction",
                    "Name": "Tourist Attraction",
                    "LocalizedName": "Tourist Attraction",
                    "Primary": false
                }
```

```
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/search-text?key=Your_Key'
 \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryText": "Stanley Park",
    "BiasPosition": [
        -123.11336,
        49.26038
    ]
}'
```

AWS CLI

```
aws geo-places search-text --key ${YourKey} --query-text "Stanley Park" --bias-
position -123.11336 49.26038
```

按企业名称搜索

Sample request

```
{
    "QueryText": "Amazon YVR11",
    "BiasPosition": [
        -123.11336,
        49.26038
    ]
}
```

Sample response

```
{
    "ResultItems": [
        {
```

```
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Amazon YVR11",
            "Address": {
                "Label": "Amazon YVR11, 510 W Georgia St, Vancouver, BC V6B 0M3,
Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "District": "Downtown Vancouver",
                "PostalCode": "V6B 0M3",
                "Street": "W Georgia St",
                "StreetComponents": [
                    {
                        "BaseName": "Georgia",
                        "Type": "St",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "W",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "510"
            },
            "Position": [
                -123.11694,
                49.28126
            ],
            "Distance": 2336,
            "Categories": [
                {
                    "Id": "business_facility",
                    "Name": "Business Facility",
                    "LocalizedName": "Business Facility",
```

```
                    "Primary": true
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -123.11656,
                        49.28151
                    ]
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/search-text?key=Your_Key'
 \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryText": "Amazon YVR11",
    "BiasPosition": [
        -123.11336,
        49.26038
    ]
}'
```

AWS CLI

```
aws geo-places search-text --key ${YourKey} --query-text "Amazon YVR11" --bias-
position -123.11336 49.26038
```

如何使用联系信息搜索地点

该 SearchText API 允许用户使用电话号码搜索地点，支持国际和本地格式。用户可以通过根据设备位置、IP 地址或地图视口中心设置位置，或者通过指定城市或地点来根据地理坐标来优化结果，从而对结果进行偏差。

潜在用例

- 使用电话号码查找地点：使用联系电话找到 POI 的地址。

示例

使用电话号码搜索

在此示例中，使用来自不列颠哥伦比亚省温哥华偏见位置的电话号码 “+1 778-655-9554” 搜索温哥华水族馆。

Sample request

```
{
    "QueryText": "+1 778-655-9554",
    "BiasPosition": [
        -123.11336,
        49.26038
    ]
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Vancouver Aquarium",
            "Address": {
                "Label": "Vancouver Aquarium, 834 Avison Way, Vancouver, BC V6G,
 Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
```

```
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "District": "Stanley Park",
                "PostalCode": "V6G",
                "Street": "Avison Way",
                "StreetComponents": [
                    {
                        "BaseName": "Avison",
                        "Type": "Way",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "834"
            },
            "Position": [
                -123.13049,
                49.30013
            ],
            "Distance": 4591,
            "Categories": [
                {
                    "Id": "aquarium",
                    "Name": "Aquarium",
                    "LocalizedName": "Aquarium",
                    "Primary": true
                },
                {
                    "Id": "tourist_attraction",
                    "Name": "Tourist Attraction",
                    "LocalizedName": "Tourist Attraction",
                    "Primary": false
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
```

```
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/search-text?key=Your_Key'
 \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryText": "+1 778-655-9554",
    "BiasPosition": [
        -123.11336,
        49.26038
    ]
}'
```

AWS CLI

```
aws geo-places search-text --key ${YourKey} --query-text "+1 778-655-9554" --bias-
position -123.11336 49.26038
```

如何使用类别名称或食物类型进行搜索

SearchTextAPI 允许您按类别名称进行搜索，例如餐厅、学校等。您也可以按食物类型进行搜索。

使用 SearchText API 的一种方法是让最终用户按名称（类别和食物类型）和应用程序集的偏见位置进行搜索。这些偏置位置可以是设备位置、IP 位置或地图的视口中心。此外，最终用户可以提供城市名称或地点，应用程序可以根据地理坐标对结果进行偏差。

有关更多信息，请参阅类别过滤器和食物类型过滤器。

潜在用例

- 作为探索社区的一部分，搜索特定的 POI 或地点。
- 搜索特定的旅游景点以进行旅行计划。
- 搜索供应特定食物的餐厅。

示例

在类别上搜索

Sample request

```
{
    "QueryText": "Nursing Home",
```

```
    "BiasPosition": [
                -123.11336,
                49.26038
            ],
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Diamond Geriatrics Inc",
            "Address": {
                "Label": "Diamond Geriatrics Inc, 288 W 8th Ave, Vancouver, BC V5Y
 1N5, Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "District": "Mt Pleasant",
                "PostalCode": "V5Y 1N5",
                "Street": "W 8th Ave",
                "StreetComponents": [
                    {
                        "BaseName": "8th",
                        "Type": "Ave",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "W",
                        "Language": "en"
                    }
                ],
```

```
                "AddressNumber": "288"
            },
            "Position": [
                -123.11061,
                49.2636
            ],
            "Distance": 410,
            "Categories": [
                {
                    "Id": "nursing_home",
                    "Name": "Nursing Home",
                    "LocalizedName": "Nursing Home",
                    "Primary": true
                },
                {
                    "Id": "social_service",
                    "Name": "Social Service",
                    "LocalizedName": "Social Services",
                    "Primary": false
                }
            ]
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Para Med Home Health Care",
            "Address": {
                "Label": "Para Med Home Health Care, 601 W Broadway, Vancouver, BC
V5Z 4C2, Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "District": "Fairview",
                "PostalCode": "V5Z 4C2",
```

```
                "Street": "W Broadway",
                "StreetComponents": [
                    {
                        "BaseName": "Broadway",
                        "Prefix": "W",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "601"
            },
            "Position": [
                -123.11772,
                49.26343
            ],
            "Distance": 464,
            "Categories": [
                {
                    "Id": "nursing_home",
                    "Name": "Nursing Home",
                    "LocalizedName": "Nursing Home",
                    "Primary": true
                },
                {
                    "Id": "hospital_or_health_care_facility",
                    "Name": "Hospital or Health Care Facility",
                    "LocalizedName": "Hospital or Health Care Facility",
                    "Primary": false
                }
            ]
        }
        ...
        ...
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/search-text?key=Your_Key'
 \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryText": "Nursing Home",
```

```
    "BiasPosition": [
                -123.11336,
                49.26038
            ]
}'
```

AWS CLI

```
aws geo-places search-text --key ${YourKey} --query-text "Nursing Home" --bias-
position -123.11336 49.26038
```

按食物类型搜索

按食物类型指定食物类型QueryText以选择餐厅。通过按食物类型进行筛选，可以按食物类型对餐厅进行索引、查询和显示。

Sample request

```
{
    "QueryText": "Sushi",
    "BiasPosition": [
                -123.11336,
                49.26038
            ]
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Sushi Tomy",
            "Address": {
                "Label": "Sushi Tomy, 555 W 12th Ave, Vancouver, BC V5Z 3X7,
 Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
```

```
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "District": "Fairview",
                "PostalCode": "V5Z 3X7",
                "Street": "W 12th Ave",
                "StreetComponents": [
                    {
                        "BaseName": "12th",
                        "Type": "Ave",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "W",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "555"
            },
            "Position": [
                -123.11629,
                49.26086
            ],
            "Distance": 219,
            "Categories": [
                {
                    "Id": "deli",
                    "Name": "Deli",
                    "LocalizedName": "Deli",
                    "Primary": true
                },
                {
                    "Id": "restaurant",
                    "Name": "Restaurant",
                    "LocalizedName": "Restaurant",
                    "Primary": false
                },
                {
                    "Id": "casual_dining",
                    "Name": "Casual Dining",
```

```
                    "LocalizedName": "Casual Dining",
                    "Primary": false
                }
            ],
            "FoodTypes": [
                {
                    "LocalizedName": "Japanese - Sushi",
                    "Id": "japanese-sushi",
                    "Primary": true
                },
                {
                    "LocalizedName": "Canadian",
                    "Id": "canadian",
                    "Primary": false
                },
                {
                    "LocalizedName": "Japanese",
                    "Id": "japanese",
                    "Primary": false
                }
            ]
        },
        {
            "PlaceId": "<Redacted>"
            "PlaceType": "PointOfInterest",
            "Title": "Shiro Japanese Restaurant",
            "Address": {
                "Label": "Shiro Japanese Restaurant, 3096 Cambie St, Vancouver, BC
V5Z 2V9, Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "District": "Mt Pleasant",
                "PostalCode": "V5Z 2V9",
```

```
                "Street": "Cambie St",
                "StreetComponents": [
                    {
                        "BaseName": "Cambie",
                        "Type": "St",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "3096"
            },
            "Position": [
                -123.11461,
                49.25797
            ],
            "Distance": 283,
            "Categories": [
                {
                    "Id": "casual_dining",
                    "Name": "Casual Dining",
                    "LocalizedName": "Casual Dining",
                    "Primary": true
                },
                {
                    "Id": "restaurant",
                    "Name": "Restaurant",
                    "LocalizedName": "Restaurant",
                    "Primary": false
                }
            ],
            "FoodTypes": [
                {
                    "LocalizedName": "Japanese - Sushi",
                    "Id": "japanese-sushi",
                    "Primary": true
                },
                {
                    "LocalizedName": "Asian",
                    "Id": "asian",
                    "Primary": false
                },
                {
                    "LocalizedName": "Japanese",
```

```
                    "Id": "japanese",
                    "Primary": false
                }
            ]
        },
        ...
        ...
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/search-text?key=Your_Key'
 \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryText": "Sushi",
    "BiasPosition": [
                -123.11336,
                49.26038
            ]
}'
```

AWS CLI

```
aws geo-places search-text --key ${YourKey} --query-text "Sushi" --bias-position
 -123.11336 49.26038
```

如何搜索地址

SearchTextAPI 允许您搜索地址。

使用 SearchText API 的一种方法是让最终用户搜索地址和应用程序集的偏差位置。这些偏置位置可以是设备位置、IP 位置或地图的视口中心。此外，最终用户可以提供城市名称或地点，应用程序可以根据地理坐标对结果进行偏差。

潜在用例

- 根据偏见位置搜索地址。
- 筛选结果以包含或删除特定地址。

示例

搜索地址

Sample request

```
{
    "QueryText":"1368 E 8 Ave Vancouver",
        "BiasPosition":[
                -123.18544,
                49.24643
        ],
     "MaxResults":1
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointAddress",
            "Title": "1368 E 8th Ave, Vancouver, BC V5N 1T2, Canada",
            "Address": {
                "Label": "1368 E 8th Ave, Vancouver, BC V5N 1T2, Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "District": "Grandview-Woodland",
                "PostalCode": "V5N 1T2",
                "Street": "E 8th Ave",
                "StreetComponents": [
                    {
```

```
                        "BaseName": "8th",
                        "Type": "Ave",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Prefix": "E",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "1368"
            },
            "Position": [
                -123.07612,
                49.26306
            ],
            "Distance": 8147,
            "MapView": [
                -123.07752,
                49.26299,
                -123.05661,
                49.26334
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/search-text?key=Your_Key'
 \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryText": "1368 E 8 Ave Vancouver",
    "BiasPosition": [
                -123.18544,
                49.24643
            ],
    "MaxResults": 1
}'
```

AWS CLI

```
aws geo-places search-text --key ${YourKey} --query-text "1368 E 8 Ave Vancouver" --
bias-position -123.18544 49.24643 --max-results 1
```

如何在某个国家内搜索地点

通过 SearchText API，您可以将搜索结果限制在一个或多个国家/地区内。此功能对于在边境地区或彼此相邻的小国进行搜索很有用。

使用 SearchText API 的一种方法是让最终用户进行自由文本搜索并确定应用程序集的偏见位置。这些偏置位置可以是设备位置、IP 位置或地图的视口中心。此外，最终用户可以提供城市名称或地点，应用程序可以根据地理坐标对结果进行偏差。

潜在用例

- 将搜索结果限制在一个国家范围内，以搜索国际边境为邻近的地区。
- 将结果筛选到多个国家。

示例

在一个国家内搜索

在此示例中，偏差位置位于美国和加拿大边境附近。如果不指定IncludeCountries": ["CAN"]，API 将返回来自加拿大和美国的结果。

Sample request

```
{
    "QueryText": "Starbucks",
    "BiasPosition": [
                -122.741803,
                 49.002478
            ]
}
```

Sample response

```
{
    "ResultItems": [
```

```
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "STARBUCKS",
            "Address": {
                "Label": "STARBUCKS, 16010 24 Ave, Surrey, BC V3Z 0R5, Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Surrey",
                "District": "South Surrey",
                "PostalCode": "V3Z 0R5",
                "Street": "24 Ave",
                "StreetComponents": [
                    {
                        "BaseName": "24",
                        "Type": "Ave",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "16010"
            },
            "Position": [
                -122.77867,
                49.04482
            ],
            "Distance": 5422,
            "Categories": [
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
```

```
            }
        ]
    },
    {
        "PlaceId": "<Redacted>",
        "PlaceType": "PointOfInterest",
        "Title": "STARBUCKS",
        "Address": {
            "Label": "STARBUCKS, 1730 152 St, Surrey, BC V4A 4N4, Canada",
            "Country": {
                "Code2": "CA",
                "Code3": "CAN",
                "Name": "Canada"
            },
            "Region": {
                "Code": "BC",
                "Name": "British Columbia"
            },
            "SubRegion": {
                "Name": "Metro Vancouver"
            },
            "Locality": "Surrey",
            "District": "South Surrey",
            "PostalCode": "V4A 4N4",
            "Street": "152 St",
            "StreetComponents": [
                {
                    "BaseName": "152",
                    "Type": "St",
                    "TypePlacement": "AfterBaseName",
                    "TypeSeparator": " ",
                    "Language": "en"
                }
            ],
            "AddressNumber": "1730"
        },
        "Position": [
            -122.80096,
            49.03341
        ],
        "Distance": 5517,
        "Categories": [
            {
                "Id": "coffee_shop",
```

```
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": true
                }
            ],
            "BusinessChains": [
                {
                    "Name": "Starbucks",
                    "Id": "Starbucks"
                }
            ]
        },
        ...
        ...
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/search-text?key=Your_Key'
 \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryText": "Starbucks",
    "BiasPosition": [
                -122.741803,
                49.002478
            ]
}'
```

AWS CLI

```
aws geo-places search-text --key ${YourKey} --query-text "Starbucks" --bias-position
 -122.741803 49.002478
```

如何搜索联系方式和营业时间

通过 SearchText API，您可以搜索 POI（兴趣点）的联系人和营业时间。

使用该 SearchText API的一种方法是通过搜索公司名称和偏见位置来获取营业地点（电话号码、网站、电子邮件）和营业时间的联系信息。这些偏置位置可以是设备位置、IP 位置、地图的视口中心。另一种方法是让最终用户提供城市名称或地名，应用程序可以根据地理坐标对结果进行偏差。

有关更多信息，请参阅 联系方式和开放时间。

潜在用例

您可以使用 SearchText API 筛选联系人和营业时间。

示例

按类别名称搜索

Sample request

```
{
    "QueryText": "Nursing Home",
    "BiasPosition": [
                -123.11336,
                49.26038
            ],
        "AdditionalFeatures":["Contact"]

}
```

Sample response

```
{
    "PricingBucket": "Advanced",
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Para Med Home Health Care",
            "Address": {
                "Label": "Para Med Home Health Care, 601 W Broadway, Vancouver, BC
 V5Z 4C2, Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
```

```
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "District": "Fairview",
                "PostalCode": "V5Z 4C2",
                "Street": "W Broadway",
                "StreetComponents": [
                    {
                        "BaseName": "Broadway",
                        "Prefix": "W",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "601"
            },
            "Position": [
                -123.11771,
                49.26341
            ],
            "Distance": 462,
            "Categories": [
                {
                    "Id": "nursing_home",
                    "Name": "Nursing Home",
                    "LocalizedName": "Nursing Home",
                    "Primary": true
                },
                {
                    "Id": "hospital_or_health_care_facility",
                    "Name": "Hospital or Health Care Facility",
                    "LocalizedName": "Hospital or Health Care Facility",
                    "Primary": false
                }
            ],
            "Contacts": {
                "Phones": [
                    {
                        "Value": "+16045634663"
                    },
                    {
```

```
                        "Value": "+16047311594",
                        "Categories": [
                            {
                                "Id": "hospital_or_health_care_facility",
                                "Name": "Hospital or Health Care Facility"
                            }
                        ]
                    }
                ],
                "Websites": [
                    {
                        "Value": "http://www.rightathomecanada.com/vancouver",
                        "Categories": [
                            {
                                "Id": "nursing_home",
                                "Name": "Nursing Home"
                            }
                        ]
                    },
                    {
                        "Value": "https://www.facebook.com/507123162974954",
                        "Categories": [
                            {
                                "Id": "nursing_home",
                                "Name": "Nursing Home"
                            }
                        ]
                    }
                ]
            }
        },
         {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Yaletown House",
            "Address": {
                "Label": "Yaletown House, 1099 Cambie St, Vancouver, BC V6B 5A8,
Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
```

```
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
                    "Name": "Metro Vancouver"
                },
                "Locality": "Vancouver",
                "District": "Yaletown",
                "PostalCode": "V6B 5A8",
                "Street": "Cambie St",
                "StreetComponents": [
                    {
                        "BaseName": "Cambie",
                        "Type": "St",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "1099"
            },
            "Position": [
                -123.11949,
                49.27533
            ],
            "Distance": 1721,
            "Categories": [
                {
                    "Id": "nursing_home",
                    "Name": "Nursing Home",
                    "LocalizedName": "Nursing Home",
                    "Primary": true
                },
                {
                    "Id": "restaurant",
                    "Name": "Restaurant",
                    "LocalizedName": "Restaurant",
                    "Primary": false
                },
                {
                    "Id": "clothing_and_accessories",
                    "Name": "Clothing and Accessories",
                    "LocalizedName": "Clothing & Accessories",
                    "Primary": false
```

```
                },
                {
                    "Id": "market",
                    "Name": "Market",
                    "LocalizedName": "Market",
                    "Primary": false
                },
                {
                    "Id": "jeweler",
                    "Name": "Jeweler",
                    "LocalizedName": "Jeweler",
                    "Primary": false
                },
                {
                    "Id": "advertising-marketing,_pr_and_market_research",
                    "Name": "Advertising-Marketing, PR and Market Research",
                    "LocalizedName": "Advertising/Marketing, PR & Market Research",
                    "Primary": false
                },
                {
                    "Id": "management_and_consulting_services",
                    "Name": "Management and Consulting Services",
                    "LocalizedName": "Management and Consulting Services",
                    "Primary": false
                },
                {
                    "Id": "organizations_and_societies",
                    "Name": "Organizations and Societies",
                    "LocalizedName": "Organizations and Societies",
                    "Primary": false
                },
                {
                    "Id": "finance_and_insurance",
                    "Name": "Finance and Insurance",
                    "LocalizedName": "Finance and Insurance",
                    "Primary": false
                },
                {
                    "Id": "consumer_services",
                    "Name": "Consumer Services",
                    "LocalizedName": "Consumer Services",
                    "Primary": false
                },
                {
```

```
                    "Id": "attorney",
                    "Name": "Attorney",
                    "LocalizedName": "Attorney",
                    "Primary": false
                },
                {
                    "Id": "social_service",
                    "Name": "Social Service",
                    "LocalizedName": "Social Services",
                    "Primary": false
                }
            ],
            "Contacts": {
                "Phones": [
                    {
                        "Value": "+16045584546",
                        "Categories": [
                            {
                                "Id": "management_and_consulting_services",
                                "Name": "Management and Consulting Services"
                            },
                            {
                                "Id": "consumer_services",
                                "Name": "Consumer Services"
                            }
                        ]
                    },
                    {
                        "Value": "+16046464065",
                        "Categories": [
                            {
                                "Id": "finance_and_insurance",
                                "Name": "Finance and Insurance"
                            }
                        ]
                    },
                    {
                        "Value": "+16046699800",
                        "Categories": [
                            {
                                "Id": "clothing_and_accessories",
                                "Name": "Clothing and Accessories"
                            },
                            {
```

```
                                "Id": "jeweler",
                                "Name": "Jeweler"
                            }
                        ]
                    },
                    {
                        "Value": "+16046838109",
                        "Categories": [
                            {
                                "Id": "attorney",
                                "Name": "Attorney"
                            }
                        ]
                    },
                    {
                        "Value": "+16046841838",
                        "Categories": [
                            {
                                "Id": "advertising-
marketing,_pr_and_market_research",
                                "Name": "Advertising-Marketing, PR and Market
 Research"
                            },
                            {
                                "Id": "management_and_consulting_services",
                                "Name": "Management and Consulting Services"
                            }
                        ]
                    },
                    {
                        "Value": "+16046848898",
                        "Categories": [
                            {
                                "Id": "attorney",
                                "Name": "Attorney"
                            }
                        ]
                    },
                    {
                        "Value": "+16046890022"
                    },
                    {
                        "Value": "+16046999120",
                        "Categories": [
```

```
                        {
                            "Id": "nursing_home",
                            "Name": "Nursing Home"
                        }
                    ]
                },
                {
                    "Value": "+16046999121",
                    "Categories": [
                        {
                            "Id": "nursing_home",
                            "Name": "Nursing Home"
                        }
                    ]
                },
                {
                    "Value": "+16048016446",
                    "Categories": [
                        {
                            "Id": "management_and_consulting_services",
                            "Name": "Management and Consulting Services"
                        }
                    ]
                }
            ],
            "Faxes": [
                {
                    "Value": "+16046464068",
                    "Categories": [
                        {
                            "Id": "finance_and_insurance",
                            "Name": "Finance and Insurance"
                        }
                    ]
                },
                {
                    "Value": "+16046627954",
                    "Categories": [
                        {
                            "Id": "organizations_and_societies",
                            "Name": "Organizations and Societies"
                        }
                    ]
                },
```

```
                    {
                        "Value": "+16046841801",
                        "Categories": [
                            {
                                "Id": "advertising-
marketing,_pr_and_market_research",
                                "Name": "Advertising-Marketing, PR and Market
 Research"
                            }
                        ]
                    },
                    {
                        "Value": "+16046848608",
                        "Categories": [
                            {
                                "Id": "attorney",
                                "Name": "Attorney"
                            }
                        ]
                    }
                ],
                "Websites": [
                    {
                        "Value": "http://www.yaletown.org",
                        "Categories": [
                            {
                                "Id": "organizations_and_societies",
                                "Name": "Organizations and Societies"
                            },
                            {
                                "Id": "social_service",
                                "Name": "Social Service"
                            },
                            {
                                "Id": "nursing_home",
                                "Name": "Nursing Home"
                            }
                        ]
                    },
                    {
                        "Value": "http://www.yaletownhq.com",
                        "Categories": [
                            {
```

```
                                "Id": "advertising-
marketing,_pr_and_market_research",
                                "Name": "Advertising-Marketing, PR and Market
 Research"
                            }
                        ]
                    },
                    {
                        "Value": "http://www.yaletownlaw.ca",
                        "Categories": [
                            {
                                "Id": "attorney",
                                "Name": "Attorney"
                            }
                        ]
                    },
                    {
                        "Value": "https://www.facebook.com/1509112252725564",
                        "Categories": [
                            {
                                "Id": "nursing_home",
                                "Name": "Nursing Home"
                            }
                        ]
                    }
                ],
                "Emails": [
                    {
                        "Value": "nina@yaletownhq.com",
                        "Categories": [
                            {
                                "Id": "advertising-
marketing,_pr_and_market_research",
                                "Name": "Advertising-Marketing, PR and Market
 Research"
                            }
                        ]
                    }
                ]
            },
            "OpeningHours": [
                {
                    "Display": [
                        "Mon-Fri: 08:30 - 16:00"
```

```
                    ],
                    "OpenNow": true,
                    "Components": [
                        {
                            "OpenTime": "T083000",
                            "OpenDuration": "PT07H30M",
                            "Recurrence": "FREQ:DAILY;BYDAY:MO,TU,WE,TH,FR"
                        }
                    ],
                    "Categories": [
                        {
                            "Id": "organizations_and_societies",
                            "Name": "Organizations and Societies"
                        }
                    ]
                },
                {
                    "Display": [
                        "Mon-Fri: 09:00 - 17:00"
                    ],
                    "OpenNow": true,
                    "Components": [
                        {
                            "OpenTime": "T090000",
                            "OpenDuration": "PT08H00M",
                            "Recurrence": "FREQ:DAILY;BYDAY:MO,TU,WE,TH,FR"
                        }
                    ],
                    "Categories": [
                        {
                            "Id": "attorney",
                            "Name": "Attorney"
                        }
                    ]
                },
                {
                    "Display": [
                        "Tue: 10:00 - 16:30",
                        "Wed-Sat: 09:30 - 16:30"
                    ],
                    "OpenNow": true,
                    "Components": [
                        {
                            "OpenTime": "T100000",
```

```
                            "OpenDuration": "PT06H30M",
                            "Recurrence": "FREQ:DAILY;BYDAY:TU"
                        },
                        {
                            "OpenTime": "T093000",
                            "OpenDuration": "PT07H00M",
                            "Recurrence": "FREQ:DAILY;BYDAY:WE,TH,FR,SA"
                        }
                    ],
                    "Categories": [
                        {
                            "Id": "clothing_and_accessories",
                            "Name": "Clothing and Accessories"
                        },
                        {
                            "Id": "jeweler",
                            "Name": "Jeweler"
                        }
                    ]
                }
            ]
        },
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/search-text?key=Your_Key'
 \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryText": "Nursing Home",
    "BiasPosition": [
                -123.11336,
                49.26038
       ],
    "AdditionalFeatures":["Contact"]
}'
```

AWS CLI

```
aws geo-places search-text --key ${YourKey} --query-text "Nursing Home" \
```

```
--bias-position -123.11336 49.26038 \
--additional-features "Contact"
```

如何使用查询 ID 搜索地点

通过 SearchText API，您可以获取返回的查询结果的详细信息建议。

潜在用例

获取 Suggestion API 返回的查询 ID 的搜索结果。

示例

按查询 ID 搜索

Sample request

```
{
    "QueryId": "<Redacted>"
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Restore",
            "Address": {
                "Label": "Restore, 400 Urban Plz, Kirkland, WA 98033, United
 States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "King"
```

```
                },
                "Locality": "Kirkland",
                "District": "Moss Bay",
                "PostalCode": "98033",
                "Street": "Urban Plz",
                "StreetComponents": [
                    {
                        "BaseName": "Urban",
                        "Type": "Plz",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "400"
            },
            "Position": [
                -122.19958,
                47.67822
            ],
            "Distance": 190670,
            "Categories": [
                {
                    "Id": "wellness_center_and_services",
                    "Name": "Wellness Center and Services",
                    "LocalizedName": "Wellness Center & Services",
                    "Primary": true
                }
            ],
            "BusinessChains": [
                {
                    "Name": "Restore",
                    "Id": "Restore"
                }
            ],
            "Contacts": {
                "Phones": [
                    {
                        "Value": "+14258352271"
                    },
                    {
                        "Value": "+14254205116"
                    }
                ],
```

```
                "Websites": [
                    {
                        "Value": "https://restore.com"
                    },
                    {
                        "Value": "https://restore.com/locations"
                    },
                    {
                        "Value": "https://www.facebook.com/rhwkirkland"
                    },
                    {
                        "Value": "https://www.restore.com/locations/wa-kirkland-
wa001?
utm_source=extnet&utm_medium=Yext&y_source=1_MzE2Mjk5NTgtNTY2LWxvY2F0aW9uLndlYnNpdGU
%3D"
                    },
                    {
                        "Value": "https://www.restore.com/sitemap.xml"
                    }
                ],
                "Emails": [
                    {
                        "Value": "kirkland@restore.com"
                    }
                ]
            },
            "OpeningHours": [
                {
                    "Display": [
                        "Mon-Thu: 10:00 - 19:00",
                        "Fri: 10:00 - 17:00",
                        "Sat: 09:00 - 17:00",
                        "Sun: 10:00 - 16:00"
                    ],
                    "OpenNow": false,
                    "Components": [
                        {
                            "OpenTime": "T100000",
                            "OpenDuration": "PT09H00M",
                            "Recurrence": "FREQ:DAILY;BYDAY:MO,TU,WE,TH"
                        },
                        {
                            "OpenTime": "T100000",
                            "OpenDuration": "PT07H00M",
```

```
                            "Recurrence": "FREQ:DAILY;BYDAY:FR"
                        },
                        {
                            "OpenTime": "T090000",
                            "OpenDuration": "PT08H00M",
                            "Recurrence": "FREQ:DAILY;BYDAY:SA"
                        },
                        {
                            "OpenTime": "T100000",
                            "OpenDuration": "PT06H00M",
                            "Recurrence": "FREQ:DAILY;BYDAY:SU"
                        }
                    ]
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -122.19944,
                        47.67801
                    ]
                }
            ]
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Restore",
            "Address": {
                "Label": "Restore, 1520 Highlands Dr NE, Issaquah, WA 98029, United
States",
                "Country": {
                    "Code2": "US",
                    "Code3": "USA",
                    "Name": "United States"
                },
                "Region": {
                    "Code": "WA",
                    "Name": "Washington"
                },
                "SubRegion": {
                    "Name": "King"
                },
                "Locality": "Issaquah",
```

```
                "District": "Issaquah Highlands",
                "PostalCode": "98029",
                "Street": "Highlands Dr NE",
                "StreetComponents": [
                    {
                        "BaseName": "Highlands",
                        "Type": "Dr",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Suffix": "NE",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "1520"
            },
            "Position": [
                -122.01868,
                47.54262
            ],
            "Distance": 209665,
            "Categories": [
                {
                    "Id": "wellness_center_and_services",
                    "Name": "Wellness Center and Services",
                    "LocalizedName": "Wellness Center & Services",
                    "Primary": true
                }
            ],
            "BusinessChains": [
                {
                    "Name": "Restore",
                    "Id": "Restore"
                }
            ],
            "Contacts": {
                "Phones": [
                    {
                        "Value": "+14256518137"
                    }
                ],
                "Websites": [
                    {
                        "Value": "https://restore.com",
                        "Categories": [
```

```
                            {
                                "Id": "wellness_center_and_services",
                                "Name": "Wellness Center and Services"
                            }
                        ]
                    },
                    {
                        "Value": "https://www.facebook.com/rhwissaquah",
                        "Categories": [
                            {
                                "Id": "wellness_center_and_services",
                                "Name": "Wellness Center and Services"
                            }
                        ]
                    },
                    {
                        "Value": "https://www.restore.com/locations",
                        "Categories": [
                            {
                                "Id": "wellness_center_and_services",
                                "Name": "Wellness Center and Services"
                            }
                        ]
                    },
                    {
                        "Value": "https://www.restore.com/locations/wa-issaquah-
wa002?
utm_source=extnet&utm_medium=Yext&y_source=1_MzE2Mjk5NTktNTY2LWxvY2F0aW9uLndlYnNpdGU
%3D",
                        "Categories": [
                            {
                                "Id": "wellness_center_and_services",
                                "Name": "Wellness Center and Services"
                            }
                        ]
                    },
                    {
                        "Value": "https://www.restore.com/sitemap.xml",
                        "Categories": [
                            {
                                "Id": "wellness_center_and_services",
                                "Name": "Wellness Center and Services"
                            }
                        ]
```

```
                    }
                ],
                "Emails": [
                    {
                        "Value": "issaquah@restore.com",
                        "Categories": [
                            {
                                "Id": "wellness_center_and_services",
                                "Name": "Wellness Center and Services"
                            }
                        ]
                    }
                ]
            },
            "OpeningHours": [
                {
                    "Display": [
                        "Mon-Thu: 10:00 - 18:00",
                        "Fri: 10:00 - 17:00",
                        "Sat: 09:00 - 17:00"
                    ],
                    "OpenNow": false,
                    "Components": [
                        {
                            "OpenTime": "T100000",
                            "OpenDuration": "PT08H00M",
                            "Recurrence": "FREQ:DAILY;BYDAY:MO,TU,WE,TH"
                        },
                        {
                            "OpenTime": "T100000",
                            "OpenDuration": "PT07H00M",
                            "Recurrence": "FREQ:DAILY;BYDAY:FR"
                        },
                        {
                            "OpenTime": "T090000",
                            "OpenDuration": "PT08H00M",
                            "Recurrence": "FREQ:DAILY;BYDAY:SA"
                        }
                    ],
                    "Categories": [
                        {
                            "Id": "wellness_center_and_services",
                            "Name": "Wellness Center and Services"
                        }
```

```
                    ]
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -122.01894,
                        47.54262
                    ]
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/search-text?key=Your_Key'
 \
  --header 'Content-Type: application/json' \
  --data '{
    "QueryId": "<Redacted>"
}'
```

AWS CLI

```
aws geo-places search-text --key ${YourKey} --query-id "<Redacted>"
```

如何使用特定语言进行搜索

此功能允许从 BCP47兼容代码中选择首选响应语言。它根据名称变体检测查询语言，并使用首选语言处理不匹配的标记和模棱两可的情况。如果没有请求的语言，Plac es API 会以官方国家/地区语言提供搜索结果，但会优先考虑不同区域语言的区域。作为后备策略，如果请求的语言中没有任何地址元素不可用，Plac es 将以默认语言 APIs 返回地址。

潜在用例

一个潜在的用例是本地化查询和/或结果。

示例

以特定语言获取结果

如果请求中没有该"Language": "EN"行，结果将以希伯来语返回。

Sample request

```
{
    "QueryText": "Vegan",
        "BiasPosition":[
                34.78953,
                32.08556
        ],
    "Language": "EN"
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Rainbow",
            "Address": {
                "Label": "Rainbow, Ibn Gabirol 88, 6404612 Tel Aviv-Yafo, Israel",
                "Country": {
                    "Code2": "IL",
                    "Code3": "ISR",
                    "Name": "Israel"
                },
                "Region": {
                    "Code": "TA",
                    "Name": "TA"
                },
                "SubRegion": {
                    "Name": "Tel Aviv"
                },
                "Locality": "Tel Aviv-Yafo",
                "District": "The New North-Southern Area",
                "PostalCode": "6404612",
                "Street": "Ibn Gabirol",
```

```
                "StreetComponents": [
                    {
                        "BaseName": "Ibn Gabirol",
                        "Language": "he-Latn"
                    }
                ],
                "AddressNumber": "88"
            },
            "Position": [
                34.78149,
                32.08252
            ],
            "Distance": 829,
            "Categories": [
                {
                    "Id": "restaurant",
                    "Name": "Restaurant",
                    "LocalizedName": "Restaurant",
                    "Primary": true
                },
                {
                    "Id": "fast_food",
                    "Name": "Fast Food",
                    "LocalizedName": "Fast Food",
                    "Primary": false
                }
            ],
            "FoodTypes": [
                {
                    "LocalizedName": "Vegan",
                    "Id": "vegan",
                    "Primary": true
                },
                {
                    "LocalizedName": "Fast Food",
                    "Id": "fast_food",
                    "Primary": false
                },
                {
                    "LocalizedName": "Burgers",
                    "Id": "burgers",
                    "Primary": false
                },
                {
```

```
                    "LocalizedName": "Jewish/Kosher",
                    "Id": "jewish/kosher",
                    "Primary": false
                }
            ]
        },
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Alegria",
            "Address": {
                "Label": "Alegria, Ibn Gabirol 165, 6203305 Tel Aviv-Yafo, Israel",
                "Country": {
                    "Code2": "IL",
                    "Code3": "ISR",
                    "Name": "Israel"
                },
                "Region": {
                    "Code": "TA",
                    "Name": "TA"
                },
                "SubRegion": {
                    "Name": "Tel Aviv"
                },
                "Locality": "Tel Aviv-Yafo",
                "District": "The Old North-Northern Area",
                "PostalCode": "6203305",
                "Street": "Ibn Gabirol",
                "StreetComponents": [
                    {
                        "BaseName": "Ibn Gabirol",
                        "Language": "he-Latn"
                    }
                ],
                "AddressNumber": "165"
            },
            "Position": [
                34.78295,
                32.0923
            ],
            "Distance": 973,
            "Categories": [
                {
                    "Id": "restaurant",
```

```
                    "Name": "Restaurant",
                    "LocalizedName": "Restaurant",
                    "Primary": true
                },
                {
                    "Id": "coffee_shop",
                    "Name": "Coffee Shop",
                    "LocalizedName": "Coffee Shop",
                    "Primary": false
                },
                {
                    "Id": "specialty_food_store",
                    "Name": "Specialty Food Store",
                    "LocalizedName": "Specialty Food Store",
                    "Primary": false
                },
                {
                    "Id": "bakery_and_baked_goods_store",
                    "Name": "Bakery and Baked Goods Store",
                    "LocalizedName": "Bakery & Baked Goods Store",
                    "Primary": false
                }
            ],
            "FoodTypes": [
                {
                    "LocalizedName": "Vegan",
                    "Id": "vegan",
                    "Primary": true
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/search-text?key=Your_Key'
 \
  --header 'Content-Type: application/json' \
  --data '
        {
            "QueryText": "Vegan",
```

```
                "BiasPosition":[
                        34.78953,
                        32.08556
                ],
            "Language": "EN"
        }'
```

AWS CLI

```
aws geo-places search-text --key ${YourKey} --query-text "Vegan" \
--bias-position 34.78953 32.08556 \
--language "EN"
```

如何使用 “建议”

本节包含各种使用方法指南和示例，说明如何将 Suggest API 与响应数据配合使用。

主题

- 如何根据输入预测建议
- 如何获取部分键入或拼写错误的查询的结果
- 使用 Suggest 选择路线，并能够突出显示匹配的查询词
- 使用 Suggest 并能够筛选某个地区或国家
- 如何帮助用户消除相似结果之间的歧义

如何根据输入预测建议

Suggest API 使应用程序能够完成用户对地点或结果类别的查询。这些建议可以直接使用，也可以在 SearchText API 中进一步完善，以检索更多详细信息和结果。

潜在用例

- 餐厅或公园查询：查找特定位置附近的餐厅、公园或其他地方。

示例

查找某个位置附近的餐厅

Sample request

```
{
  "QueryText": "restaura",
  "BiasPosition": [-124.81035775620968, 49.234628873773985],
  "AdditionalFeatures": ["Core"]
}
```

Sample response

```
{
    "ResultItems": [
        {
            "Title": "Restaurant",
            "SuggestResultItemType": "Query",
            "Query": {
                "QueryId": "AQAAAHgAw1x7...",
                "QueryType": "Category"
            },
            "Highlights": {
                "Title": [
                    {
                        "StartIndex": 0,
                        "EndIndex": 8,
                        "Value": "Restaura"
                    }
                ]
            }
        }
    ],
    "QueryRefinements": []
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/suggest?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
```

```
  "QueryText": "restaura",
  "BiasPosition": [-124.81035775620968, 49.234628873773985],
  "AdditionalFeatures": ["Core"]
}'
```

AWS CLI

```
aws geo-places suggest --key ${YourKey} \
--query-text "restaura" \
--bias-position -124.81035775620968 49.234628873773985 \
--additional-features "Core"
```

在某个位置附近查找链条的位置

Sample request

```
{
  "QueryText": "Domin",
  "BiasPosition": [-124.81035775620968, 49.234628873773985],
  "AdditionalFeatures": ["Core"]
}
```

Sample response

```
{
    "ResultItems": [
        {
            "Title": "Domino's",
            "SuggestResultItemType": "Query",
            "Query": {
                "QueryId": "AQAAAHUA4UNp...",
                "QueryType": "BusinessChain"
            },
            "Highlights": {
                "Title": [
                    {
                        "StartIndex": 0,
                        "EndIndex": 5,
                        "Value": "Domin"
                    }
                ]
            }
```

```
        }
    ],
    "QueryRefinements": []
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/suggest?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
  "QueryText": "Domin",
  "BiasPosition": [-124.81035775620968, 49.234628873773985],
  "AdditionalFeatures": ["Core"]
}'
```

AWS CLI

```
aws geo-places suggest --key ${YourKey} \
--query-text "Domin" \
--bias-position -124.81035775620968 49.234628873773985 \
--additional-features "Core"
```

使用完成查询 QueryId

Sample request (SearchText)

```
{
  "QueryId": "<QueryId from Suggest API>",
  "AdditionalFeatures": ["Core"]
}
```

Sample response

```
{
    "ResultItems": [
        {
            "PlaceId": "<Redacted>",
            "PlaceType": "PointOfInterest",
            "Title": "Domino's",
            "Address": {
```

```
                "Label": "Domino's, 233 Shelly Rd, Parksville, BC V9P, Canada",
                "Country": {
                    "Code2": "CA",
                    "Code3": "CAN",
                    "Name": "Canada"
                },
                "Region": {
                    "Code": "BC",
                    "Name": "British Columbia"
                },
                "SubRegion": {
                    "Name": "Nanaimo"
                },
                "Locality": "Parksville",
                "PostalCode": "V9P",
                "Street": "Shelly Rd",
                "StreetComponents": [
                    {
                        "BaseName": "Shelly",
                        "Type": "Rd",
                        "TypePlacement": "AfterBaseName",
                        "TypeSeparator": " ",
                        "Language": "en"
                    }
                ],
                "AddressNumber": "233"
            },
            "Position": [
                -124.29334,
                49.31669
            ],
            "Distance": 38602,
            "Categories": [
                {
                    "Id": "fast_food",
                    "Name": "Fast Food",
                    "LocalizedName": "Fast Food",
                    "Primary": true
                },
                {
                    "Id": "take_out_and_delivery_only",
                    "Name": "Take Out and Delivery Only",
                    "LocalizedName": "Take Out & Delivery Only",
                    "Primary": false
```

```
                }
            ],
            "FoodTypes": [
                {
                    "LocalizedName": "Pizza",
                    "Id": "pizza",
                    "Primary": true
                }
            ],
            "BusinessChains": [
                {
                    "Name": "Domino's",
                    "Id": "Domino's"
                }
            ],
            "Contacts": {
                "Phones": [
                    {
                        "Value": "+12502489296"
                    }
                ],
                "Websites": [
                    {
                        "Value": "http://pizza.dominos.ca/Parksville-British-
Columbia-10039"
                    }
                ]
            },
            "OpeningHours": [
                {
                    "Display": [
                        "Mon-Fri: 11:00 - 24:00",
                        "Sat, Sun: 00:00 - 01:00, 11:00 - 24:00"
                    ],
                    "OpenNow": true,
                    "Components": [
                        {
                            "OpenTime": "T110000",
                            "OpenDuration": "PT13H00M",
                            "Recurrence": "FREQ:DAILY;BYDAY:MO,TU,WE,TH,SU"
                        },
                        {
                            "OpenTime": "T110000",
                            "OpenDuration": "PT14H00M",
```

```
                        "Recurrence": "FREQ:DAILY;BYDAY:FR,SA"
                    }
                ]
            }
        ],
        "AccessPoints": [
            {
                "Position": [
                    -124.29319,
                    49.31669
                ]
            }
        ]
    },
    {
        "PlaceId": "<Redacted>",
        "PlaceType": "PointOfInterest",
        "Title": "Domino's",
        "Address": {
            "Label": "Domino's, 215 Port Augusta St, Comox, BC V9M, Canada",
            "Country": {
                "Code2": "CA",
                "Code3": "CAN",
                "Name": "Canada"
            },
            "Region": {
                "Code": "BC",
                "Name": "British Columbia"
            },
            "SubRegion": {
                "Name": "Comox Valley"
            },
            "Locality": "Comox",
            "PostalCode": "V9M",
            "Street": "Port Augusta St",
            "StreetComponents": [
                {
                    "BaseName": "Port Augusta",
                    "Type": "St",
                    "TypePlacement": "AfterBaseName",
                    "TypeSeparator": " ",
                    "Language": "en"
                }
            ],
```

```
                "AddressNumber": "215"
            },
            "Position": [
                -124.92492,
                49.67378
            ],
            "Distance": 49528,
            "Categories": [
                {
                    "Id": "fast_food",
                    "Name": "Fast Food",
                    "LocalizedName": "Fast Food",
                    "Primary": true
                },
                {
                    "Id": "take_out_and_delivery_only",
                    "Name": "Take Out and Delivery Only",
                    "LocalizedName": "Take Out & Delivery Only",
                    "Primary": false
                }
            ],
            "FoodTypes": [
                {
                    "LocalizedName": "Pizza",
                    "Id": "pizza",
                    "Primary": true
                }
            ],
            "BusinessChains": [
                {
                    "Name": "Domino's",
                    "Id": "Domino's"
                }
            ],
            "Contacts": {
                "Phones": [
                    {
                        "Value": "+17784310222"
                    }
                ],
                "Websites": [
                    {
                        "Value": "http://pizza.dominos.ca/comox-british-
columbia-39035"
```

```
                    }
                ]
            },
            "OpeningHours": [
                {
                    "Display": [
                        "Mon-Thu: 10:30 - 23:00",
                        "Fri: 10:30 - 24:00",
                        "Sat: 00:00 - 01:00, 10:30 - 24:00",
                        "Sun: 00:00 - 01:00, 10:30 - 23:00"
                    ],
                    "OpenNow": true,
                    "Components": [
                        {
                            "OpenTime": "T103000",
                            "OpenDuration": "PT12H30M",
                            "Recurrence": "FREQ:DAILY;BYDAY:MO,TU,WE,TH,SU"
                        },
                        {
                            "OpenTime": "T103000",
                            "OpenDuration": "PT14H30M",
                            "Recurrence": "FREQ:DAILY;BYDAY:FR,SA"
                        }
                    ]
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -124.92518,
                        49.67387
                    ]
                }
            ]
        },
        ...
    ],
    "QueryRefinements": []
}
```

cURL

```
curl --request POST \
```

```
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/search-text?key=Your_Key'
 \
  --header 'Content-Type: application/json' \
  --data '{
  "QueryId": "<QueryId from Suggest API>",
  "AdditionalFeatures": ["Core"]
}'
```

AWS CLI

```
aws geo-places suggest --key ${YourKey} \
--query-id "<QueryId from Suggest API>" \
--additional-features "Core"
```

开发者小贴士

使用诸如`Filter.IncludeCountries`或之类`Filter.BoundingBox`的过滤器`BiasPosition`。这些过滤器适用于使用进行的后续查询`QueryId`。

```
{
  "QueryText": "Domin",
  "BiasPosition": [-124.81035775620968, 49.234628873773985],
  "Filter": {
    "IncludeCountries": ["CAN"]
  }
}
```

如何获取部分键入或拼写错误的查询的结果

Suggest API 使应用程序能够完成用户对地点或结果类别的查询。这些结果按最可能的匹配项和不太可能的匹配项进行排序，允许 API 解析不完整或拼写错误的单词。

潜在用例

- 完成部分键入的兴趣点查询：根据部分键入或不正确的条目提供建议，从而为用户提供帮助。

示例

查找拼写错误的兴趣点

Sample request

```
{
  "QueryText": "Effel tow",
  "Filter": {
    "Circle": {
      "Radius": 10000,
      "Center": [2.3431932014695382, 48.858844492141145]
    },
    "IncludeCountries": ["FRA"]
  },
  "AdditionalFeatures": ["Core"]
}
```

Sample response

```
{
    "ResultItems": [
        {
            "Title": "Tour Eiffel (Eiffel Tower)",
            "SuggestResultItemType": "Place",
            "Place": {
                "PlaceId": "<Redacted>",
                "PlaceType": "PointOfInterest",
                "Address": {
                    "Label": "Tour Eiffel, 5 Avenue Anatole France, 75007 Paris,
 France",
                    "Country": {"Code2": "FR", "Code3": "FRA", "Name": "France"},
                    "Region": {"Code": "IDF", "Name": "Île-de-France"},
                    "SubRegion": {"Name": "Paris"},
                    "Locality": "Paris",
                    "District": "7e Arrondissement",
                    "PostalCode": "75007",
                    "Street": "Avenue Anatole France",
                    "StreetComponents": [{"BaseName": "Anatole France", "Type":
 "Avenue", "Language": "fr"}],
                    "AddressNumber": "5"
                },
                "Position": [2.2945, 48.85824],
```

```
                "Distance": 3563,
                "Categories": [
                    {"Id": "historical_monument", "Name": "Historical Monument",
 "LocalizedName": "Monument historique", "Primary": true},
                    {"Id": "landmark-attraction", "Name": "Landmark-Attraction",
 "LocalizedName": "Lieu d'intérêt/Attraction", "Primary": false},
                    {"Id": "tourist_attraction", "Name": "Tourist Attraction",
 "LocalizedName": "Attraction touristique", "Primary": false},
                    {"Id": "sports_complex-stadium", "Name": "Sports Complex-
Stadium", "LocalizedName": "Stade ou complexe sportif", "Primary": false}
                ]
            },
            "Highlights": {"Title": [{"StartIndex": 13, "EndIndex": 23, "Value":
 "Eiffel Tow"}]}
        }
    ],
    "QueryRefinements": []
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/suggest?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
  "QueryText": "Effel tow",
  "Filter": {
    "Circle": {"Radius": 10000, "Center": [2.3431932014695382, 48.858844492141145]},
    "IncludeCountries": ["FRA"]
  },
  "AdditionalFeatures": ["Core"]
}'
```

AWS CLI

```
aws geo-places suggest --key ${YourKey} \
--query-text "Effel tow" \
--filter '{"Circle": {"Radius": 1000, "Center": [2.3431932014695382,
 48.858844492141145]}, "IncludeCountries": ["FRA"]}' \
--additional-features "Core"
```

开发者小贴士

使用诸如Filter.IncludeCountries或之类Filter.BoundingBox的过滤器BiasPosition。这些过滤器可以帮助缩小可能的结果范围并提高准确性。

```
{
  "QueryText": "Effel tow",
  "BiasPosition": [2.2982750966095398, 48.856078089325294],
  "Filter": {"IncludeCountries": ["FRA"]}
}
```

使用 Suggest 选择路线，并能够突出显示匹配的查询词

借助 Suggest API，当用户键入地点或类别的搜索查询时，您可以获得实时的自动完成建议。结果按相关性排序，最有可能的匹配项排在最前面。API 响应包括突出显示每条建议与用户查询的匹配程度的信息。

潜在用例

使用突出显示功能向用户显示每个建议的哪些部分与他们的输入相匹配，从而更轻松地快速选择所需的结果。

示例

向用户显示结果列表时，响应中的 “亮点” 字段可用于帮助用户识别输入与结果的匹配情况。在下面的示例中，用户查询 “Effel tow”。这与标题 “埃菲尔铁塔（埃菲尔铁塔）” 的结果相符。使用 Highlights 结果字段，可以设置结果样式以显示输入查询与输出匹配的位置，从而向用户显示文本为 “Tour Eiffel（Eiffel Tow er）”。

使用建议 API 突出显示查询结果

此示例演示如何使用 Suggest API 来查找拼写错误的兴趣点。

Sample request

```
{
  "QueryText": "Effel tow",
  "Filter": {
    "Circle": {
      "Radius": 10000,
      "Center": [
        2.3431932014695382, 48.858844492141145
```

```
      ]
    },
    "IncludeCountries": [
      "FRA"
    ]
  },
  "AdditionalFeatures": [
    "Core"
  ]
}
```

Sample response

```
{
    "ResultItems": [
        {
            "Title": "Tour Eiffel (Eiffel Tower)",
            "SuggestResultItemType": "Place",
            "Place": {
                "PlaceId": "<Redacted>",
                "PlaceType": "PointOfInterest",
                "Address": {
                    "Label": "Tour Eiffel, * Avenue Anatole France, ***** Paris,
 France",
                    "Country": {
                        "Code2": "FR",
                        "Code3": "FRA",
                        "Name": "France"
                    },
                    "Region": {
                        "Code": "IDF",
                        "Name": "Île-de-France"
                    },
                    "SubRegion": {
                        "Name": "Paris"
                    },
                    "Locality": "Paris",
                    "District": "*e Arrondissement",
                    "PostalCode": "*****",
                    "Street": "Avenue Anatole France",
                    "StreetComponents": [
                        {
                            "BaseName": "Anatole France",
```

```
                        "Type": "Avenue",
                        "TypePlacement": "BeforeBaseName",
                        "TypeSeparator": " ",
                        "Language": "fr"
                    }
                ],
                "AddressNumber": "5"
            },
            "Position": [
                2.2945,
                48.85824
            ],
            "Distance": 3563,
            "Categories": [
                {
                    "Id": "historical_monument",
                    "Name": "Historical Monument",
                    "LocalizedName": "Monument historique",
                    "Primary": true
                },
                {
                    "Id": "landmark-attraction",
                    "Name": "Landmark-Attraction",
                    "LocalizedName": "Lieu d'intérêt/Attraction",
                    "Primary": false
                },
                {
                    "Id": "tourist_attraction",
                    "Name": "Tourist Attraction",
                    "LocalizedName": "Attraction touristique",
                    "Primary": false
                },
                {
                    "Id": "sports_complex-stadium",
                    "Name": "Sports Complex-Stadium",
                    "LocalizedName": "Stade ou complexe sportif",
                    "Primary": false
                }
            ]
        },
        "Highlights": {
            "Title": [
                {
                    "StartIndex": 13,
```

```
                        "EndIndex": 23,
                        "Value": "Eiffel Tow"
                    }
                ],
                "Address": {}
            }
        }
    ],
    "QueryRefinements": []
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/suggest?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
  "QueryText": "Effel tow",
  "Filter": {
    "Circle": {
      "Radius": 10000,
      "Center": [
        2.3431932014695382, 48.858844492141145
      ]
    },
    "IncludeCountries": [
      "FRA"
    ]
  },
  "AdditionalFeatures": [
    "Core"
  ]
}'
```

AWS CLI

```
aws geo-places suggest --key ${YourKey} \
--query-text "Effel tow" \
--filter '{"Circle": {"Radius": 10000, "Center": [2.3431932014695382,
 48.858844492141145]}, "IncludeCountries": ["FRA"]}' \
--additional-features "Core"
```

开发者小贴士

使用Title响应字段显示搜索结果，为用户提供简洁、可识别的条目。对于可能看起来相似的结果，请使用该Place.Address.Label字段显示其他地址详细信息，以帮助用户区分它们。有关更多信息，请参阅 如何帮助用户消除相似结果之间的歧义。

使用 Suggest 并能够筛选某个地区或国家

使用 Suggest API 可以完成对地点或结果类别的查询。结果按最有可能和不太可能的匹配项进行排序。此外，您还可以进一步筛选特定地区或国家的结果。

潜在用例

- 筛选结果：执行搜索时，您可以按国家/地区进行筛选，也可以仅按业务所在地区进行筛选。

示例

添加按国家/地区筛选条件可确保您只收到最相关的结果。使用过滤器。 IncludeCountries 为此。

允许用户按国家/地区筛选

Sample request

```
{
  "QueryText": "hote",
  "BiasPosition": [
    2.2982750966095398, 48.856078089325294
  ],
  "Filter": {
    "IncludeCountries": [
      "FRA"
    ]
  },
  "AdditionalFeatures": [
    "Core"
  ]
}
```

Sample response

```
{
    "ResultItems": [
        {
```

```
            "Title": "Hôtel de Matignon (Hôtel Matignon)",
            "SuggestResultItemType": "Place",
            "Place": {
                "PlaceId": "Redacted",
                "PlaceType": "PointOfInterest",
                "Address": {
                    "Label": "Hôtel de Matignon, 57 Rue de Varenne, 75007 Paris,
France",
                    "Country": {
                        "Code2": "FR",
                        "Code3": "FRA",
                        "Name": "France"
                    },
                    "Region": {
                        "Code": "IDF",
                        "Name": "Île-de-France"
                    },
                    "SubRegion": {
                        "Name": "Paris"
                    },
                    "Locality": "Paris",
                    "District": "7e Arrondissement",
                    "PostalCode": "75007",
                    "Street": "Rue de Varenne",
                    "StreetComponents": [
                        {
                            "BaseName": "Varenne",
                            "Type": "Rue de",
                            "TypePlacement": "BeforeBaseName",
                            "TypeSeparator": " ",
                            "Language": "fr"
                        }
                    ],
                    "AddressNumber": "57"
                },
                "Position": [
                    2.32072,
                    48.85471
                ],
                "Distance": 1650,
                "Categories": [
                    {
                        "Id": "tourist_attraction",
                        "Name": "Tourist Attraction",
```

```
                    "LocalizedName": "Attraction touristique",
                    "Primary": true
                },
                {
                    "Id": "landmark-attraction",
                    "Name": "Landmark-Attraction",
                    "LocalizedName": "Lieu d'intérêt/Attraction",
                    "Primary": false
                },
                {
                    "Id": "historical_monument",
                    "Name": "Historical Monument",
                    "LocalizedName": "Monument historique",
                    "Primary": false
                },
                {
                    "Id": "residential_area-building",
                    "Name": "Residential Area-Building",
                    "LocalizedName": "Bâtiment résidentiel",
                    "Primary": false
                }
            ]
        },
        "Highlights": {
            "Title": [
                {
                    "StartIndex": 0,
                    "EndIndex": 4,
                    "Value": "Hôt"
                },
                {
                    "StartIndex": 19,
                    "EndIndex": 23,
                    "Value": "(Hô"
                }
            ],
            "Address": {
                "Label": [
                    {
                        "StartIndex": 0,
                        "EndIndex": 4,
                        "Value": "Hôt"
                    }
                ]
```

```
                }
            }
        },
        ...
        {
            "Title": "Hotel",
            "SuggestResultItemType": "Query",
            "Query": {
                "QueryId": "Redacted",
                "QueryType": "Category"
            },
            "Highlights": {
                "Title": [
                    {
                        "StartIndex": 0,
                        "EndIndex": 4,
                        "Value": "Hote"
                    }
                ]
            }
        },
        ...
    ],
    "QueryRefinements": []
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/suggest?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
  "QueryText": "hote",
  "BiasPosition": [
    2.2982750966095398, 48.856078089325294
  ],
  "Filter": {
    "IncludeCountries": [
      "FRA"
    ]
  },
  "AdditionalFeatures": [
    "Core"
```

```
  ]
}'
```

AWS CLI

```
aws geo-places suggest --key ${YourKey} \
--query-text "hote" \
--bias-position 2.2982750966095398 48.856078089325294 \
--filter '{"IncludeCountries": ["FRA"]}' \
--additional-features "Core"
```

为搜索添加 filter-by-region选项会将结果限制为仅显示符合所提供区域的搜索结果。这用于将搜索范围限制在与您的业务相关的区域。以下示例显示了对波士顿地区附近的万豪酒店的拼写错误搜索，其中包含一些酒店搜索结果和后续查询。

允许用户按地区筛选

Sample request

```
{
  "QueryText": "Mariot",
  "Filter": {
    "IncludeCountries": [
      "USA"
    ],
    "BoundingBox": [
      -71.15693983012913,42.261623506672635,
      -70.97249727163558,42.37584075627763
    ]
  },
  "AdditionalFeatures": [
    "Core"
  ]
}
```

Sample response

```
{
    "ResultItems": [
        {
            "Title": "Boston Marriott Copley Place",
```

```
            "SuggestResultItemType": "Place",
            "Place": {
                "PlaceId": "Redacted",
                "PlaceType": "PointOfInterest",
                "Address": {
                    "Label": "Boston Marriott Copley Place, 110 Huntington Ave,
Boston, MA 02116, United States",
                    "Country": {
                        "Code2": "US",
                        "Code3": "USA",
                        "Name": "United States"
                    },
                    "Region": {
                        "Code": "MA",
                        "Name": "Massachusetts"
                    },
                    "SubRegion": {
                        "Name": "Suffolk"
                    },
                    "Locality": "Boston",
                    "District": "Back Bay",
                    "PostalCode": "02116",
                    "Street": "Huntington Ave",
                    "StreetComponents": [
                        {
                            "BaseName": "Huntington",
                            "Type": "Ave",
                            "TypePlacement": "AfterBaseName",
                            "TypeSeparator": " ",
                            "Language": "en"
                        }
                    ],
                    "AddressNumber": "110"
                },
                "Position": [
                    -71.07769,
                    42.34726
                ],
                "Distance": 3347,
                "Categories": [
                    {
                        "Id": "hotel",
                        "Name": "Hotel",
                        "LocalizedName": "Hotel",
```

```
                    "Primary": true
                }
            ],
            "BusinessChains": [
                {
                    "Name": "Marriott",
                    "Id": "Marriott"
                }
            ],
            "AccessPoints": [
                {
                    "Position": [
                        -71.07922,
                        42.34665
                    ]
                }
            ]
        },
        "Highlights": {
            "Title": [
                {
                    "StartIndex": 7,
                    "EndIndex": 14,
                    "Value": "Marriot"
                }
            ],
            "Address": {
                "Label": [
                    {
                        "StartIndex": 7,
                        "EndIndex": 14,
                        "Value": "Marriot"
                    }
                ]
            }
        }
    },
    ...
    {
        "Title": "Marriott",
        "SuggestResultItemType": "Query",
        "Query": {
            "QueryId": "Redacted",
            "QueryType": "BusinessChain"
```

```
            },
            "Highlights": {
                "Title": [
                    {
                        "StartIndex": 0,
                        "EndIndex": 8,
                        "Value": "Marriott"
                    }
                ]
            }
        },
        ...
    ],
    "QueryRefinements": []
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/suggest?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
  "QueryText": "Mariot",
  "Filter": {
    "IncludeCountries": [
      "USA"
    ],
    "BoundingBox": [
      -71.15693983012913,42.261623506672635,
      -70.97249727163558,42.37584075627763
    ]
  },
  "AdditionalFeatures": [
    "Core"
  ]
}'
```

AWS CLI

```
aws geo-places suggest --key ${YourKey} \
--query-text "hote" \
--filter '{"IncludeCountries": ["FRA"], "BoundingBox":
 [ -71.15693983012913,42.261623506672635, -70.97249727163558,42.37584075627763]}' \
```

```
--additional-features "Core"
```

开发者小贴士

使用Title响应字段显示搜索结果，为用户提供简洁、可识别的条目。对于可能看起来相似的结果，请使用该Place.Address.Label字段显示其他地址详细信息，以帮助用户区分它们。有关更多信息，请参阅 如何帮助用户消除相似结果之间的歧义。

如何帮助用户消除相似结果之间的歧义

Suggest API 允许您创建对地点或结果类别的查询。对结果进行排序，以确定可能性更高和不太可能的匹配结果。为每个结果提供了信息，可帮助您快速识别正确的匹配项，尤其是标题和标签字段。

潜在用例

- 显示结果选项：您可以选择最适合您需求的选项。

示例

通常，“标题” 字段提供的信息足以让你了解结果何时达到预期效果。对于标题未提供足够信息的情况，“标签” 字段用于为结果添加地址信息上下文。“标签” 字段可以作为第二行信息提供，当为你呈现结果时，正文中不会强调该信息。在下面的示例中，提供了两个相似的结果，酒店的地址可以帮助您选择更适合您需求的结果。此外，还提供了位置，因此可以在地图上渲染结果，作为在结果之间进行选择的另一种方式。

使用建议 API 消除结果的歧义

Sample request

```
{
  "QueryText": "Marriott",
  "Filter": {
    "IncludeCountries": [
      "USA"
    ],
    "BoundingBox": [
      -71.15693983012913,42.261623506672635,
      -70.97249727163558,42.37584075627763
    ]
  },
```

```
  "AdditionalFeatures": [
    "Core"
  ]
}
```

Sample response

```
{
    "ResultItems": [
        {
            "Title": "Marriott-Boston Cambridge",
            "SuggestResultItemType": "Place",
            "Place": {
                "PlaceId": "Redacted",
                "PlaceType": "PointOfInterest",
                "Address": {
                    "Label": "Marriott-Boston Cambridge, 50 Broadway, Cambridge, MA
 02138-4137, United States",
                    "Country": {
                        "Code2": "US",
                        "Code3": "USA",
                        "Name": "United States"
                    },
                    "Region": {
                        "Code": "MA",
                        "Name": "Massachusetts"
                    },
                    "SubRegion": {
                        "Name": "Middlesex"
                    },
                    "Locality": "Cambridge",
                    "District": "MIT",
                    "PostalCode": "02138-4137",
                    "Street": "Broadway",
                    "StreetComponents": [
                        {
                            "BaseName": "Broadway",
                            "Language": "en"
                        }
                    ],
                    "AddressNumber": "50"
                },
                "Position": [
```

```
                    -71.0858,
                    42.36294
                ],
                "Distance": 5212,
            ...
        },
        {
            "Title": "Marriott-Boston Copley Place",
            "SuggestResultItemType": "Place",
            "Place": {
                "PlaceId": "Redacted",
                "PlaceType": "PointOfInterest",
                "Address": {
                    "Label": "Marriott-Boston Copley Place, 110 Huntington Ave,
Boston, MA 02116-5706, United States",
                    "Country": {
                        "Code2": "US",
                        "Code3": "USA",
                        "Name": "United States"
                    },
                    "Region": {
                        "Code": "MA",
                        "Name": "Massachusetts"
                    },
                    "SubRegion": {
                        "Name": "Suffolk"
                    },
                    "Locality": "Boston",
                    "District": "Back Bay",
                    "PostalCode": "02116-5706",
                    "Street": "Huntington Ave",
                    "StreetComponents": [
                        {
                            "BaseName": "Huntington",
                            "Type": "Ave",
                            "TypePlacement": "AfterBaseName",
                            "TypeSeparator": " ",
                            "Language": "en"
                        }
                    ],
                    "AddressNumber": "110"
                },
                "Position": [
                    -71.0792,
```

```
                    42.34701
                ],
                "Distance": 3362,
            ...
        },
        ...
    ],
    "QueryRefinements": []
}
```

cURL

```
curl --request POST \
  --url 'https://places.geo.eu-central-1.amazonaws.com/v2/suggest?key=Your_Key' \
  --header 'Content-Type: application/json' \
  --data '{
  "QueryText": "Marriott",
  "Filter": {
    "IncludeCountries": [
      "USA"
    ],
    "BoundingBox": [
      -71.15693983012913,42.261623506672635,
      -70.97249727163558,42.37584075627763
    ]
  },
  "AdditionalFeatures": [
    "Core"
  ]
}'
```

AWS CLI

```
aws geo-places suggest --key ${YourKey} \
--query-text "Marriott" \
--filter '{"IncludeCountries": ["USA"], "BoundingBox":
 [ -71.15693983012913,42.261623506672635, -70.97249727163558,42.37584075627763]}' \
--additional-features "Core"
```

管理成本和使用量

在您继续探索 Amazon Location Service 时，重要的是要了解如何管理服务容量和成本，确保遵守使用限制，并通过配额和 API 优化获得最佳结果。通过应用性能和准确性方面的最佳实践，您可以定制应用程序，以高效地处理与地点相关的查询并最大限度地提高 API 请求。

主题

- 最佳实践
- 地点定价
- 放置配额和使用量

最佳实践

本节概述了优化应用程序中 API 交互的性能和准确性的最佳实践。通过实施诸如去抖动和浏览器缓存之类的技术，并利用过滤器和地理环境，您可以改善用户体验并确保您的应用程序提供最相关的结果。

预先打字实现

在 Web 开发中处理事件或 API 调用时，管理性能对于提供流畅的用户体验至关重要。可以帮助实现这一目标的两种常用技术是：

去抖动：去抖限制了函数执行的频率，这对于窗口大小调整或用户输入等高频事件特别有用。这样可以确保只有在一定延迟后才调用函数，从而减少不必要的处理并提高性能。

浏览器缓存：通过在本地缓存最近的搜索查询和结果，浏览器缓存有助于避免对相同数据进行冗余的 API 调用。这通过在可用时提供缓存的数据来最大限度地减少网络流量并加快应用程序的运行速度。

这两种技术协同工作，可以提高处理用户交互和 API 请求的性能和效率。

获得正确的结果

使用地理环境，例如偏置位置或诸如圆圈和边界框之类的过滤器，可以通过关注邻近度并将输出限制在相关位置来增强结果。此外，诸如国家、地点类型、类别、连锁店和食物类型之类的过滤器可以通过包含或排除特定条件来帮助你进一步完善搜索。

地点定价

地点的价格基 APIs 于 API 请求的数量。每个 API 请求的单价取决于您在 API 请求中请求的响应字段以及结果的预期用途。有关 API 和定价桶的定价信息，请参阅 Amazon Location Service 定价页面。

Place 有四个定价区 APIs：标签、核心、高级和存储。

标签

标签定价区提供了一种经济实惠的选项，仅用于获取地址文本和 placeID。当你调用 “自动补全” 和 “建议 APIs ” 时，无论是否存在，API 都会返回 placeID（可以在 GetPlace 请求检索其他信息时使用）PlaceType、“标题” 和 “地址标签” 字段 APIs QueryRefinements QueryType，以及 QueryId “建议 API” 字段。`additionalFeature = []`在这种情况下，将按标签价格向您收费。无法永久存储此定价桶的结果。有关长期用例，请参阅存储定价桶。

核心实例

核心定价区支持 Place 最常见的用例 APIs。当您拨打以下电话时，仍将按核心价格收费：

- `SearchText`、、`Geocode`、`ReverseGeocode`、`SearchNearby`、带有 `additionalFeature = []` 或不存在的 `GetPlace` API
- `Autocomplete`并`Suggest` APIs 用 `additionalFeature = Core`
- `Geocode`用 `additionalFeature = SecondaryAddresses` 或 `Intersections`
- `ReverseGeocode`用 `additionalFeature = Intersections`
- `GetPlace`用 `additionalFeature = SecondaryAddresses`

API 会返回完整的地址组成部分、类别和其他地点详情（如果适用）。有关响应字段的完整列表，请参阅 API 参考。此定价存储段的结果无法永久存储，有关长期用例，请参阅the section called “已存储”存储桶。

高级

高级定价区提供了其他位置或 points-of-interest详细信息，例如营业时间、联系信息和接入点。当您调用 SearchText、、 SearchNearby GetPlace、Geocode 和 Suggest API 并在`additionalFeature`请求字段中包含以下值之一：联系人、Access 或 Phonemes 时，API 会返回您所请求的其他信息的相应值（例如、`opening_hours`和、`access_restriction`和`contact_details`、`access_points`、`Phonemes`或）。 ReverseGeocode TimeZone `Timezone`在这种情况下，将按高级价格向您收费。结果可以缓存，但不能存储以供该定价区长期使用。无法永久存储此定价桶的结果。有关长期用例，请参阅存储定价桶。

已存储

您可以无限期存储 Places 结果以用于长期用例，例如重复使用结果以减少按需 API 调用或用于分析目的。为此，请在您的 API 请求`intendedUse = Stored`中进行设置。在这种情况下，将按存储价格向您收费。Stored 定价区支持上面列出的所有功能，因此，单次 Places API 调用向您收取的最高价格以存储价格为上限。

放置配额和使用量

Amazon Location Service 对 API 使用量设定配额，以管理服务容量并防止过度使用。这些配额可以通过 AWS 服务配额控制台或联系支持人员进行调整。本节介绍地点的服务配额 APIs 和 API 使用限制，包括如何申请提高和其他相关信息。

服务配额

Amazon Location Service 为设置默认配额 APIs 以帮助管理服务容量，可以在 AWS 服务配额管理控制台中查看这些配额。您可以通过自助控制台（链接如下表所示）申请增加 Amazon Location Service 配额，每个 API 的配额最多为默认限制的 2 倍。如果配额限制超过默认限制的 2 倍，请在服务配额控制台中申请，它将提交支持请求。或者，您可以联系您的高级支持团队。增加配额的请求不会直接收费，但是根据消耗的额外资源，更高的使用级别可能会导致服务成本增加。有关更多详细信息，请参阅使用 Service Quotas 管理配额。

服务配额限制

API 名称	默认	可调节的最大限制	超过可调整的最大限制
地理编码	100	200	在服务配额控制台上申请或联系支持团队
ReverseGeocode	100	200	在服务配额控制台上申请或联系支持团队
自动完成	100	200	在服务配额控制台上申请或联系支持团队
GetPlace	100	200	在服务配额控制台上申请或联系支持团队

API 名称	默认	可调节的最大限制	超过可调整的最大限制
SearchText	100	200	在服务配额控制台上申请或联系支持团队
SearchNearby	100	200	在服务配额控制台上申请或联系支持团队
建议	100	200	在服务配额控制台上申请或联系支持团队

其他使用限制

功能可用性

自动完成功能在日本不可用。

API 使用限制

API 名称	限制	值
地理编码	压缩后的响应有效载荷大小	6 MB
ReverseGeocode	压缩后的响应有效载荷大小	6 MB
自动完成	压缩后的响应有效载荷大小	6 MB
GetPlace	压缩后的响应有效载荷大小	6 MB
SearchText	压缩后的响应有效载荷大小	6 MB
SearchNearby	压缩后的响应有效载荷大小	6 MB
建议	压缩后的响应有效载荷大小	6 MB

后续步骤

有关更多信息，请参阅：

- 归因：在使用 Amazon Location Service 数据时，请确保正确的数据归因。
- SLA：查看 Amazon Location Service 服务等级协议，了解可用性保证。
- 服务条款：熟悉有关使用 Amazon Location Service 的法律条款。

Amazon Location Service 路线



借助 Amazon Location Service Routes，您可以计算多个起点和终点之间的行驶时间和距离，可视化与道路对齐的车辆 GPS 轨迹，并更好地了解您的可用区域。这有助于降低运营成本并改善客户体验。

特征

规划路线路径

考虑距离、时间和路况等各种因素，计算两个或多个地点之间的路线。您还可以查看同一组地点的备选路线。

路线优化

根据时间或距离优化路线，选择最快或最短的路线。您还可以对航点进行排序，以针对旅行推销员问题进行优化。

路径分析

分析行驶时间、距离或停靠点数量等绩效指标，以确保路线符合您的预期目标。

服务区域

根据距离或时间限制，定义可以从特定位置提供服务的地理区域。

通行费

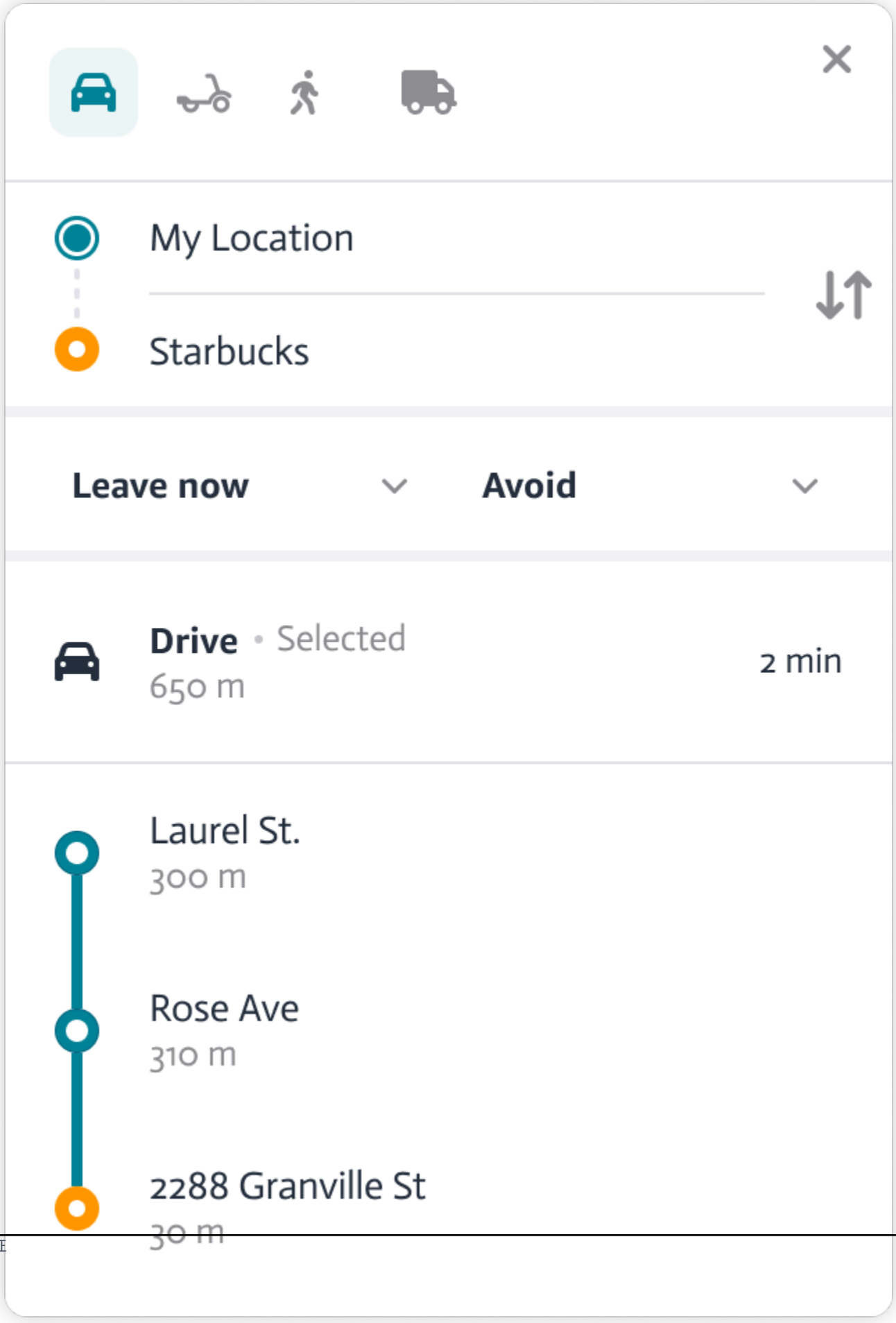
计算与您路线上的收费基础设施相关的成本。

避免

避开高速公路、隧道、渡轮和收费公路，限制路径计算。

速度限制

查找每段路线的限速，确保驾驶员遵守当地法规。


My Location
Starbucks
Leave now
Avoid
Drive • Selected
650 m
2 min
Laurel St.
300 m
Rose Ave
310 m
2288 Granville St
30 m

使用案例

提供高效的路线和 turn-by-turn方向

规划从任何起始位置（地址、POI 或 GPS 坐标）出发的路线，并考虑车辆尺寸和限制，计算通往多个目的地的最佳路径。

有关更多信息，请参阅 the section called “计算路线”。

优化配送路线

使用航点排序来解决旅行推销员问题并计算通往多个目的地的最佳路线，从而最大限度地减少时间和距离。

有关更多信息，请参阅 the section called “优化航点”。

确保从最近的仓库发货

使用矩阵路由和航点排序来确定从仓库到每位客户的最快路线，从而最大限度地提高效率并最大限度地降低成本。

有关更多信息，请参阅the section called “计算路线”和the section called “计算路径矩阵”。

改善出租车和拼车调度

使用 Matrix Routing 来识别最近的可用车辆，并根据实时交通情况计算最佳路线。这样可以确保最近的车辆高效地到达客户手中，从而提高客户满意度和业务生产力。增加行程，节省燃油成本，提供卓越的服务体验。

有关更多信息，请参阅the section called “计算路线”、the section called “计算路径矩阵”和the section called “计算等值线”。

查找服务区

根据时间或距离限制，使用 Isoline 路径来确定企业的地理覆盖范围。这使企业能够识别潜在客户并制定派遣计划。家庭医疗保健提供者可以确保他们有足够的工作人员和资源在15分钟内为所有患者提供服务。Isoline 有助于优化服务区域、确保及时交付并找到新设施。

有关更多信息，请参阅 the section called “计算等值线”。

将 GPS 轨迹捕捉到道路

将 GPS 轨迹与道路对齐，实现车辆行驶的可视化，确保路线合规性和监管合规性。车队经理可以查看车辆是否停留在计划路线上，并识别出偏差。验证司机是否遵守了指导方针，发现效率低下，

并确保遵守法规。纠正 GPS 的不准确性和差异，并呈现真实的车辆活动视图。围绕路线优化、驾驶员行为和车队管理做出更好的决策。

有关更多信息，请参阅 the section called “Snap to R”。

APIs

下表概述了 APIs 用于路线规划和基于位置的数据处理的关键 Amazon Location Service。每个 API 都提供独特的功能，例如计算路线、优化航点以及将 GPS 轨迹捕捉到道路上以实现精确跟踪。

APIs

API 名称	描述	阅读更多
计算路线	考虑车辆限制和实时交通情况，计算起点和多个目的地之间的行驶距离、行驶时间和turn-by-turn路线。	the section called “计算路线”
计算路径矩阵	将实时交通考虑在内，计算一组出发点和一组目的地之间的路线距离和时间。	the section called “计算路径矩阵”
计算等值线	根据您的出行模式，确定在指定时间或距离内可以到达的地理区域。	the section called “计算等值线”
优化航点	找到前往多个目的地的有效顺序，减少旅行时间和距离，同时考虑交通和车辆限制等因素。	the section called “优化路线的航点”
Snap to R	将 GPS 轨迹与最近的路段进行匹配，以提高车辆跟踪和路线可视化的准确性。	the section called “Snap to R”

路线概念

Amazon Location Service 中的路线概念为规划和优化旅程提供了一个强大的框架，无论是简单导航还是复杂的物流。路线包括各种组成部分，例如航点、航程、步数和跨度，每个组成部分都有助于提高路线计算的精度和灵活性。利用 Routes API，您可以计算出行距离、估算出行时间并优化多站旅程。此外，路径几何、交通感知、限速、通行费和灵活的折线编码等功能增强了可视化、分析和运营效率的能力。本节涵盖术语、最佳实践和详细的使用模式，指导您为导航、交付、现场服务等实施有效的路线解决方案。

主题

- 路线术语
- 地点（起点、目的地、航点和轨迹）
- 时间（出发和到达）
- 如何（出行模式、避让和排除）
- 交通意识
- 优化路线和航点顺序
- 司机日程安排和通知

路线术语

路线

路线提供了从出发位置通过航点位置前往目的地的详细信息。它包括行驶距离、行驶时间、路径几何形状、速度限制、跨度和其他属性。

路线矩阵

一个矩阵，表示从一组起点到一组目的地的行驶距离和行驶时间。它可用作路线规划或优化应用程序的输入。

路径点

航点是从出发点到目的地的路线上的中间停靠点。该路线遵循请求中指定的中途停留顺序。

分支

行程表示路径上两个连续位置之间的行程。如果位置是越野的，则会将其移至最近的道路。没有航点的路线由一条路段组成。具有一个或多个航点的路线有多个航段，每个航段代表从一个航点到下一个航点的行程。某些航段，例如涉及渡轮的航段，包含与该航段类型相关的特定信息。

步骤

步骤是指行程中的一个路段，提供旅程中该部分的摘要详细信息。步骤类型包括：

- 默认步骤-带有人类可读说明的基本步骤，适用于显示路线概述的 Web 应用程序。
- Turn-by-Turn 步骤 — 适用于 turn-by-turn应用程序的详细步骤，提供细致的说明。
- 行程前步骤 — 开始旅程之前要完成的步骤。示例：登上渡轮。
- 行程结束@@ 后 — 到达旅程终点后要完成的步骤。示例：登上渡轮。

跨度

跨度表示一条具有一致属性集的连续路段。每当请求的属性之一发生变化时，就会沿着路径创建新的跨度。

客户细分

路段是道路中可通航的部分，通常表示为线性路段。

路径几何

路径几何描绘了路径的路径，用于可视化、分析或其他用途。每个路段的几何形状可以表示为经过压缩的编码折线或简单的线串。

柔性多段线

一种用于表示几何图形的紧凑编码折线格式。建议用于限制响应大小，并针对客户端解码进行了优化。

简单行串

一种表示几何图形的 GeoJSON LineString 格式。这种格式会产生更大的响应有效载荷，并且是一个有序的坐标数组，可用于在地图上绘制路线。

地点（起点、目的地、航点和轨迹）

指定路径计算的位置，包括路径的起点、终点和中间停靠点（或要经过的位置）。

指定路径计算的位置

参数	描述	路线	路线矩阵	Isoline	优化航点	快到公路
起源	路线的起始位置。	是的，有选项	是的，有选项	是的，有选项	是的，有选项	否

参数	描述	路线	路线矩阵	Isoline	优化航点	快到公路
路径点	沿途将包括中间位置。	是的，有选项	是的，有选项	是的，有选项	是的，有选项	否
目的地	路径的终点位置。	是的，有选项	是的，有选项	是的，有选项	是的，有选项	否
追踪点	GPS 追踪，包括出行模式发出的历史位置信息。这些位置包括典型的 GPS 不准确性和设备无法发射或记录此信息时的间隙。	否	否	否	否	是的，有选项

可配置选项

提供自定义航点、起点和目的地路线行为的选项。

航点选项

参数	描述	路线	路线矩阵	Isoline	优化航点	快到路
AvoidActionsForDistance	避免在提供的距离内采取行动。通常用于确保驾驶员有足够的时间在航点附近做出决定。	是	否	否	否	否

参数	描述	路线	路线矩阵	Isoline	优化航点	快到路
避免 UTurns	指定航点是否允许 U 形转弯。	是	否	否	否	否
Heading	GPS 在航点位置行驶。	是	否	否	是	否
匹配	用于配置将提供的位置与道路网络相匹配的选项。	是的，有选项	否	否	否	否
SideOfStreet	指定与航点位置相匹配的街道侧面。	是的，有选项	否	否	是的，有选项	否
StopDuration	在航点位置停留的时间。	是	否	否	否	否
位置	航点的经度和纬度。	是	否	否	是	否
PassThrough	确定应将航点视为停靠点还是通过。	是	否	否	否	否
Id	航点的标识符字符串。	否	否	否	是	否
AccessHours	指定访问目的地的访问时间。	否	否	否	是	否

参数	描述	路线	路线矩阵	Isoline	优化航点	快到路
AppointmentTime	航点的预约时间。	否	否	否	是	否
ServiceDuration	航点的服务时间，例如预约时间。	否	否	否	是	否
早于	定义在此航点之后必须访问哪些航点。	否	否	否	是	否

起源选项

参数	描述	路线	路线矩阵	Isoline	优化航点	快到路
AvoidActionsForDistance	避免在提供的距离内采取行动。	支持	是	是	否	否
避免 UTurns	指定是否允许在原点处进行 U 形转弯。	是	否	否	否	否
Heading	GPS 在起点位置行驶。	支持	是	是	否	否
匹配	用于将起点位置与道路网络进行匹配的选项。	是的，有选项	是的，有选项	是的，有选项	否	否

参数	描述	路线	路线矩阵	Isoline	优化航点	快到路
SideOfStreet	指定街道的侧面以匹配原点位置。	是的，有选项	是的，有选项	是的，有选项	否	否
Id	原点的标识符字符串。	否	否	是	否	否

目的地选项

参数	描述	路线	路线矩阵	Isoline	优化航点	快到路
AvoidActionsForDistance	避免在目的地对提供的距离进行操作。	支持	是	是	否	否
避免 UTurns	指定目的地是否允许 U 形转弯。	是	否	否	否	否
Heading	GPS 驶向目的地位置。	支持	是	是	是	否
匹配	将目的地位置与道路网络相匹配的选项。	是的，有选项	是的，有选项	是的，有选项	否	否
SideOfStreet	指定与目的地位置相匹配的街道侧面。	是的，有选项	是的，有选项	是的，有选项	是的，有选项	否

参数	描述	路线	路线矩阵	Isoline	优化航点	快到路
StopDuration	在目的地停留的时间。	是	否	否	否	否
Id	目标的标识符字符串。	否	否	是	否	否
AccessHours	指定访问目的地的访问时间。	否	否	是	否	否
AppointmentTime	目的地的预约时间。	否	否	是	否	否
ServiceDuration	目的地的服务时间，例如预约时间。	否	否	是	否	否

时间（出发和到达）

指定路径计算的时间。时间不仅决定出发和到达的时间戳，还会影响要使用的交通数据。

参数	描述	路线	路线矩阵	Isoline	优化航点	快到公路
出发时间	离开起点的时间。如果既没有提供到达时间也没有提供离开时间，则不使用动态交通信息，只会应用基于历史交通	支持	是	是	是	否

参数	描述	路线	路线矩阵	Isoline	优化航点	快到公路
	量的自由流动速度。					
立即出发	使用当前时间作为离开 Origin 的时间。	支持	是	是	否	否
到达时间	到达目的地的时间。如果既没有提供到达时间也没有提供离开时间，则不使用动态交通信息，只会应用基于历史交通量的自由流动速度。	是	否	是	否	否

如何（出行模式、避让和排除）

使用以下选项指定用于路径计算的出行模式和相关要素。

出行模式选项

在计算路径时指定交通方式。此设置会影响估计的行驶速度、道路兼容性以及在需要时使用渡轮的可能性。

参数	描述	路线	路线矩阵	Isoline	优化航点	快到公路
Car	汽车出行模式。	是的，有选项	是的，有选项	是的，有选项	支持	是

参数	描述	路线	路线矩阵	Isoline	优化航点	快到公路
摩托车	滑板车出行模式。	是的，有选项	是	是的，有选项	支持	是
行人	步行出行模式。	是的，有选项	支持	是	是的，有选项	是
卡车	卡车出行模式。	是的，有选项	是的，有选项	是的，有选项	是的，有选项	是的，有选项

避免、排除和允许选项

确定在路径计算过程中是应包括、排除还是避免一组特定的要素。

参数	描述	路线	路线矩阵	Isoline	优化航点	快到公路
避免	在最佳情况下尽量避免使用的功能。如果路由器找不到有效的路由，则会忽略该避开路由，并会显示一条通知，表明无法实现避免。	是的，有选项	是的，有选项	是的，有选项	是的，有选项	否
排除	严格排除的功能。如果路由器找不到带有排除选项的有效路由，则不	是的，有选项	是的，有选项	否	否	否

参数	描述	路线	路线矩阵	Isoline	优化航点	快到公路
	会返回任何路由。					
允许	需要明确启用的功能。	是的，有选项	是的，有选项	是的，有选项	是的，有选项	否

回避清单

列出在最佳情况下应避免使用的功能。如果路由器找不到有效的路由，则会忽略该避开路由，并会显示一条通知，表明无法实现避免。

参数	描述	路线	路线矩阵	Isoline	优化航点	快到公路
领域	除例外情况外，应避开的区域。	是的，有选项	是的，有选项	是的，有选项	是的，有选项	否
受控通道高速公路	高速公路，入口点有限。	支持	是	是	是	否
汽车穿梭列车	通过隧道运送车辆的火车。	支持	是	是	否	否
土路	未铺设的道路，表面自然。	支持	是	是	是	否
渡轮	载有车辆穿越水体的船只	支持	是	是	否	否
船渡轮	在轨道上运送车辆的渡轮。	否	否	是	是	否

参数	描述	路线	路线矩阵	Isoline	优化航点	快到公路
铁路渡轮	在轨道上运送车辆的渡轮。	否	否	是	是	否
季节性关闭	道路在某些季节封闭。	是	否	是	否	否
隧道	用于车辆交通的地下通道。	支持	是	是	是	否
收费公路	需要付费才能进入的道路。	支持	是	是	是	否
收费应答器	避开收费应答器是唯一付款方式的道路。	是	否	是	否	否
U 形转弯	允许车辆反向转弯的积分。	支持	是	是	是	否
区域类别	应避免使用的区域类别。	支持	是	是	否	否

排除清单

列出严格排除的功能。如果路由器找不到带有排除选项的有效路由，则不会返回任何路由。

参数	描述	路线	路线矩阵	Isoline	优化航点	快到公路
国家/地区	国家代码 2 或国家/地	支持	是	否	是	否

参数	描述	路线	路线矩阵	Isoline	优化航点	快到公路
	区代码 3 适用于应严格排除在路线计算之外的国家。					

允许清单

列出需要明确启用路径计算的功能。

参数	描述	路线	路线矩阵	Isoline	优化航点	快到公路
HOV	允许使用高占用率车道进行路径计算。	支持	是	是	否	否
热	允许使用占用率高的收费通道进行路径计算。	支持	是	是	否	否

交通意识

确定在计算路径时使用的交通相关信息的类型。流量代表拥堵，不包括与事故相关的长期拥堵。流量数据的准确性会随着时间的推移而降低，这使得历史流量数据对于过去的事件更加可靠。

参数	描述	路线	路线矩阵	Isoline	优化航点	快到公路
使用量	在计算路径时启用或禁用流量数据。启用后，	支持	是	是	是	否

参数	描述	路线	路线矩阵	Isoline	优化航点	快到公路
	如果未提供DepartureTime ArrivalTime 、或，DepartNow 则仅考虑长期关闭。否则，如果提供了时间，则会考虑所有流量数据。					
FlowEventThresholdOverride	流量流量事件被视为有效的持续时间（以秒为单位）。虽然有效，但流量数据将用于历史流量数据。	支持	是	是	否	否

优化路线和航点顺序

优化路由

计算路径时的优化标准。这可以是按时间测量的最快路径，也可以是按距离测量的最短路径。

选项	描述	测量值
最快路线	计算最快的路线，重点是最大限度地减少行驶时间。这考虑了交通状况、道路速度限制和其他因素。	Time

选项	描述	测量值
最短路线	计算最短路线，最小化行驶距离。当距离是关键因素时，例如降低燃料成本或排放，通常使用这种方法。	Distance

优化航点

对路径中航点进行排序的优化标准。

司机日程安排和通知

司机日程安排

用于定义工作和休息时间表的驱动程序设置。许多司法管辖区都必须这样做。

参数	描述	路线	路线矩阵	Isoline	优化航点	快速上路
自定义休息周期	休息周期由行驶时长定义，然后是休息时间。可以提供任意数量的此类循环。	是的，有选项	否	否	否	否
休息周期长，休息周期短	休息周期由短周期和长周期定义。短暂的驾驶时间之后是较短的休息驾驶时间。可以重复短距离行驶，直到达到较	否	否	否	是的，有选项	否

参数	描述	路线	路线矩阵	Isoline	优化航点	快速上路
	长的驾驶持续时间，此时将强制执行较长的休息时间。					

通知、警告和限制

补充信息，可深入了解路线计算期间做出的决策。

参数	描述	路线	路线矩阵	Isoline	优化航点	快速上路
版权声明	有关路线计算的注意事项。此外，还可能包括对通知的影响，表明路径计算的结果是否可以按原样使用，或者在使用前需要手动检查。	是的，有细节	否	否	否	是
FailedConstraints	请求中提供的约束条件无法满足，从而导致优化问题失败。	否	否	否	是的，有细节	否

路线 APIs

路径提供了计算位置之间优化路径的功能。这些功能支持需要物流规划、距离计算和路线优化的应用程序。用户还可以将位置点与道路对齐，以提高准确性。有关更多信息，请参阅路线。

- CalculateIsolines：根据行驶时间或距离生成等值线，可用于定义服务区或可达区域。有关更多信息，请参阅the section called “计算等值线”。
- CalculateRouteMatrix：提供多个出发地和目的地之间的距离和行程时间的矩阵，为后勤和行程计划提供支持。有关更多信息，请参阅the section called “计算路径矩阵”。
- CalculateRoutes：计算多站导航的 point-to-point优化路线，包括可自定义的路线偏好。有关更多信息，请参阅the section called “计算路线”。
- OptimizeWaypoints：优化航点顺序以获得最高效的行驶路线，从而最大限度地减少距离或时间。有关更多信息，请参阅the section called “优化航点”。
- SnapToRoads：将坐标与最近的道路对齐，通过将点捕捉到已知道路来提高 GPS 精度。有关更多信息，请参阅the section called “Snap to R”。

下表列出了一些最好使用 Routes 解决的业务用例 APIs。

业务需求	有用的 API	示例
计算起点和目的地的行驶距离和时间 支持以下各项： • 出行模式，例如卡车、行人、汽车和滑板车 • 最多 25 个航点（中途停留），包括起点和目的地 • 驾驶休息时间	`CalculateRoutes`	
查找起点和目的地的路径折线 支持以下各项： • 出行模式，例如卡车、行人、汽车和滑板车	`CalculateRoutes`	

业务需求	有用的 API	示例
• 最多 25 个航点（中途停留），包括起点和目的地 • 驾驶休息时间		
按转弯方向查找起点和目的地的路线 支持以下各项： • 出行模式，例如卡车、行人、汽车和滑板车 • 最多 25 个航点（中途停留），包括起点和目的地 • 驾驶休息时间	`CalculateRoutes`	
立即出发查找路线 支持以下各项： • 出行模式，例如卡车、行人、汽车和滑板车 • 最多 25 个航点（中途停留），包括起点和目的地 • 驾驶休息时间	`CalculateRoutes`	
如果您在特定时间出发，请查找路线 支持以下各项： • 出行模式，例如卡车、行人、汽车和滑板车 • 最多 25 个航点（中途停留），包括起点和目的地 • 驾驶休息时间	`CalculateRoutes`	

业务需求	有用的 API	示例
如果您需要在特定时间到达，请查找路线 支持以下各项： • 出行模式，例如卡车、行人、汽车和滑板车 • 最多 25 个航点（中途停留），包括起点和目的地 • 驾驶休息时间	`CalculateRoutes`	
查找最短路线 支持以下各项： • 出行模式，例如卡车、行人、汽车和滑板车 • 最多 25 个航点（中途停留），包括起点和目的地 • 驾驶休息时间	`CalculateRoutes`	
寻找最快的路线 支持以下各项： • 出行模式，例如卡车、行人、汽车和滑板车 • 最多 25 个航点（中途停留），包括起点和目的地 • 驾驶休息时间	`CalculateRoutes`	

业务需求	有用的 API	示例
寻找替代路线 支持以下各项： • 出行模式，例如卡车、行人、汽车和滑板车 • 最多 25 个航点（中途停留），包括起点和目的地 • 驾驶休息时间	`CalculateRoutes`	
查找交通感知路线 支持以下各项： • 出行模式，例如卡车、行人、汽车和滑板车 • 最多 25 个航点（中途停留），包括起点和目的地 • 驾驶休息时间	`CalculateRoutes`	
查找避让路线，例如收费站、掉头、渡轮、高速公路、隧道等 支持以下各项： • 出行模式，例如卡车、行人、汽车和滑板车 • 最多 25 个航点（中途停留），包括起点和目的地 • 驾驶休息时间	`CalculateRoutes`	

业务需求	有用的 API	示例
通过传递多段线或多边形来查找具有自定义避让功能的路线 支持以下各项： • 出行模式，例如卡车、行人、汽车和滑板车 • 最多 25 个航点（中途停留），包括起点和目的地 • 驾驶休息时间	`CalculateRoutes`	
计算通行费 支持以下各项： • 出行模式，例如卡车、行人、汽车和滑板车 • 最多 25 个航点（中途停留），包括起点和目的地 • 驾驶休息时间	`CalculateRoutes`	the section called “计算通行费”
查找路径上某段路段的限速 支持以下各项： • 出行模式，例如卡车、行人、汽车和滑板车 • 最多 25 个航点（中途停留），包括起点和目的地 • 驾驶休息时间	`CalculateRoutes`	
在地图上绘制路线 支持航点标记。	`GetTile`并`GetStyleDescriptor` 使用带有计算路径的渲染引擎 (MapLibre)	

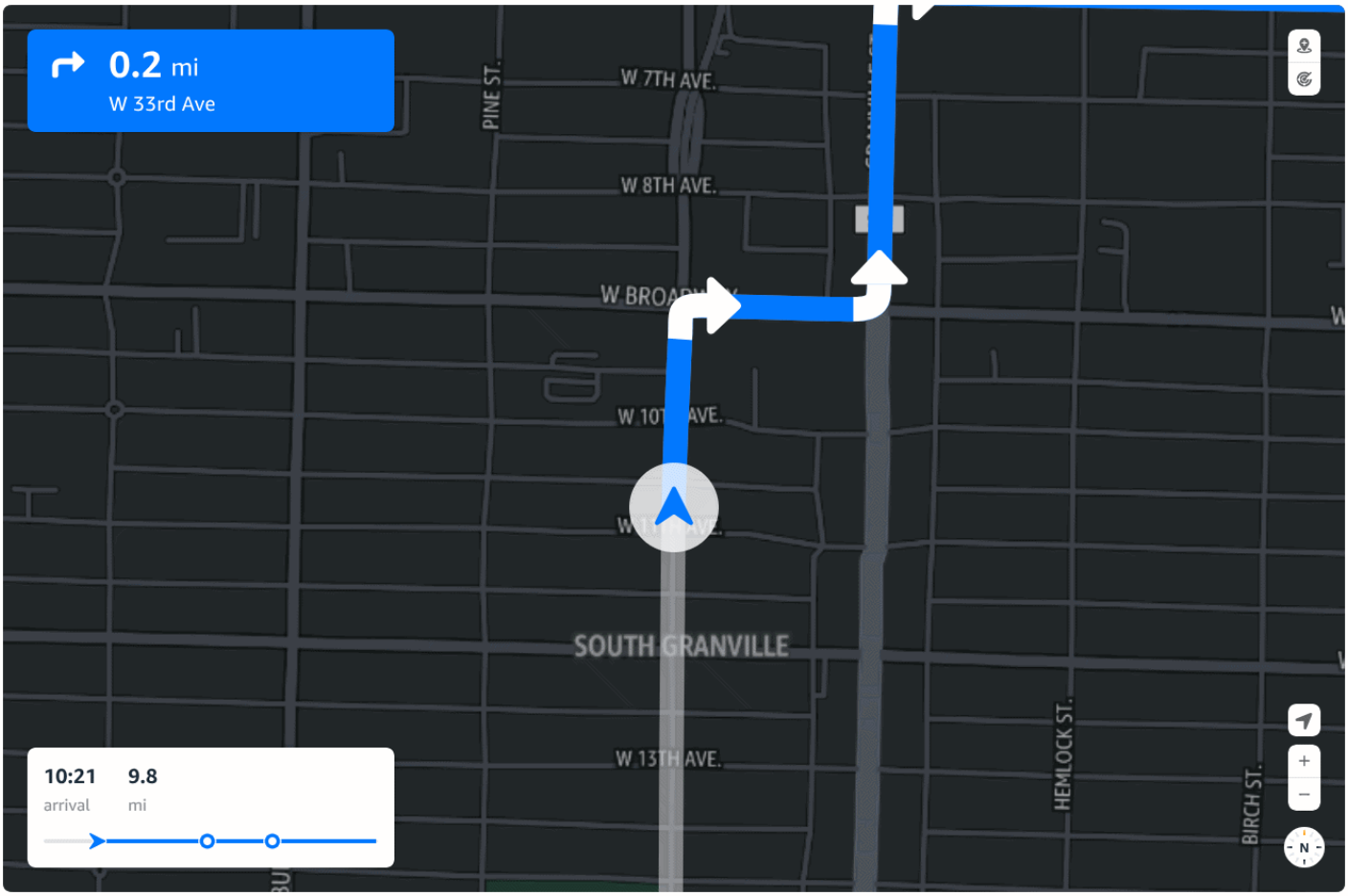
业务需求	有用的 API	示例
计算多个起点和目的地的距离和时间的路线矩阵 支持以下各项： • 超过 10 万个电池 • 出行模式，例如卡车、行人、汽车和滑板车	`CalculateRouteMatrix`	the section called “计算多个起点和目的地的路线矩阵”
使用避让计算路径矩阵 支持以下各项： • 超过 10 万个电池 • 出行模式，例如卡车、行人、汽车和滑板车	`CalculateRouteMatrix`	the section called “使用避让计算路径矩阵”
根据时间（等时线）计算服务区 支持卡车、行人、汽车和踏板车等出行模式。	`CalculateIsoline`	the section called “根据时间范围计算服务区”
根据距离（等距）计算服务区 支持卡车、行人、汽车和踏板车等出行模式。	`CalculateIsoline`	
计算带避让的服务区 支持卡车、行人、汽车和踏板车等出行模式。	`CalculateIsoline`	
从反向计算服务区 支持卡车、行人、汽车和踏板车等出行模式。	`CalculateIsoline`	

业务需求	有用的 API	示例
计算多时间或距离范围内的服务区 支持卡车、行人、汽车和踏板车等出行模式。	`CalculateIsoline`	
优化路线的航点（旅行推销员问题） 支持卡车、行人、汽车和踏板车等出行模式。	`OptimizeWaypoint`	the section called “优化路线的航点”
利用交通感知功能优化路线的航点 支持卡车、行人、汽车和踏板车等出行模式。	`OptimizeWaypoint`	
利用通行时间感知功能优化路线的航点 支持卡车、行人、汽车和踏板车等出行模式。	`OptimizeWaypoint`	
将 GPS 轨迹与道路网络匹配 支持卡车、行人、汽车和踏板车等出行模式。	快到公路	the section called “将 GPS 轨迹与道路网络匹配”
在地图上可视化匹配的 GPS 轨迹 支持卡车、行人、汽车和踏板车等出行模式。	`GetStyleDescriptor` 使用带有 Snap to road 的渲染引擎 (MapLibre)	

计算路线

Routes API 计算两个或多个地点之间的路线，无论是否有避让措施，适用于不同的出行模式，例如汽车、卡车、踏板车和行人。使用此 API，您可以自定义路由选项并请求其他与路线相关的信息以满足特

定需求。此 API 支持 turn-by-turn导航，并通过应用避开收费公路、高速公路或渡轮等参数来自定义路线计算。API 还会返回限速和通行费。



使用案例

- 在@@ 路线图上显示地理细节：使用高级测绘功能，使用丰富的地理信息（包括地标、地形和城市基础设施）可视化详细路线。允许用户查看从起点到目的地的清晰路线，从而增强决策能力。该功能可以支持导航、规划和各种物流场景，并显示汽车、卡车、踏板车和行人等出行模式的路线。通过添加避让或通行费计算等元素来自定义路线。
- 显示 turn-by-turn导航：在 Web 和移动设备上提供无缝导航支持。用户可以访问 turn-by-turn路线，确保高效出行。这两个平台都可以利用导航指令为个人或商务旅行提供路线，包括限速。
- 计算沿线的通行费：将通行费成本计算纳入路线规划，为包括收费公路、桥梁或隧道在内的路线提供准确的定价估算。预先显示通行费，以帮助司机和规划人员做出具有成本效益的决策，并在必要时避免收取通行费。
- 确保遵守限速：整合限速数据，帮助驾驶员保持在法定限制范围内，降低罚款风险，促进更安全、更省油的驾驶。实时监控速度合规性还可以使物流和车队管理受益。

- 协助制定货运和车辆路线解决方案：通过将路线、导航和跟踪功能集成到物流门户网站中，简化货运和车辆路线规划操作。有效规划多次交付的路线，实时跟踪货物，并通过更好的路线管理燃料成本。

理解请求

请求需要`Origin`和`Destination`参数，而可选参数（如`AllowAvoid`、）和`Traffic`自定义路由以满足特定的需求和约束。

Origin

以经度和纬度表示的路径起始位置。

目标位置

路径的终点位置。

航点

在起点和终点位置之间的路径上要包含的中间位置。

OptimizeRoutingFor

路径的优化标准，例如最快或最短。

LegGeometryFormat

为每个路段返回的几何图形的格式。

避免

计算路径时要避开的要素，如果找不到替代路线，则忽略这些要素。

交通

影响路径计算的与交通相关的选项。

通行费

与通行费相关的选项会影响路线计算和通行费。

LegAdditionalFeatures

可以在每段旅程的响应中启用的功能。

SpanAdditionalFeatures

可以在每段旅程的响应中启用的 Span 功能。

了解回应

响应提供路线详细信息，例如行程路段、有关路线计算的通知以及包括距离和持续时间在内的摘要信息。

路线

包含路段和相关属性的路线数组。

版权声明

有关该路线的警告或信息性消息。

LegGeometryFormat

指定路径几何图形的格式。

腿部细节

根据交通方式，每段旅程的类型可以是渡轮、行人或车辆。虽然每条腿都包含与传输模式无关的属性，但具体属性可以在下方找到：

FerryLegDetails

船腿的渡轮特有特性。

VehicleLegDetails

腿部特定于车辆的属性。

PedestrianLegDetails

腿部特定于行人的属性。

步数

旅程的每一段都分为几个步骤，这些步骤描述了部分路径的动作。步骤可以是 “默认”（适用于基本应用程序）TurnByTurn，也可以是适合 turn-by-turn导航。每个步骤都包含与步长类型无关的属性，例如持续时间和距离，以及其他特定属性 ExitStepDetails，例如仅适用于退出步骤。

BeforeTravelSteps

开始旅程之前要执行的步骤。

TravelSteps

旅途中要执行的步骤。

AfterTravelSteps

完成旅程后要执行的步骤。

跨度

旅程的每一段都可以分成多个跨度。跨度是指腿部的一部分，其值与请求的集合的值相同`SpanAdditionalFeatures`。跨度按道路属性划分`SpeedLimit`，例如道路名称或区域。返回的跨度可用于可视化道路属性和出入相关信息。

计算通行费

本主题概述了与计算通行费相关的字段和定义。使用这些字段，您可以指定付款方式、货币和车辆特性等参数，以自定义通行费成本的计算。

字段名称	路线
应答器	是的，有选项
Vignettes	是的，有选项
货币	是的，有选项
EmissionType	是的，有选项
VehicleCategory	是的，有选项

定义

本节简要定义了通行费成本计算中使用的每个字段。

应答器

转发器是一种支付通行费的方式，与其他支付方式相比，其价格可能会有所不同。

Vignettes

小插图是公路定价的一种形式。当用户拥有所需的小插图时，无需支付额外的通行费。

货币

报告通行费用的货币。除本地货币外，还包括转换后的货币，这也会影响响应中通行费摘要中使用的货币。

EmissionType

车辆的排放类型，用于根据车辆排放计算通行费。

VehicleCategory

用于计算通行费成本的车辆子类别。

了解路径步骤

本节定义了完成一段旅程所需采取的各种操作和步骤。路线步骤因出行模式而异，可为概览应用程序和详细 turn-by-turn导航提供指导。

路线步骤概述

以下类型的路径步骤定义了完成路段所需的操作，因出行模式和旅程阶段而异。

Step type (步骤类型)	描述
`Default steps`	提供人类可读指令的基本步骤，通常用于基于 Web 的应用程序中，以概述路线。
`Turn by turn steps`	创建 turn-by-turn导航应用程序的详细步骤，提供更精细的指导。
`Before travel steps`	在开始旅行部分之前需要完成的步骤，例如登上渡轮。
`After travel steps`	旅行部分完成后要执行的步骤，例如下渡轮。

按出行模式划分的步数细分

Section	步骤	旅行前	旅行	旅行后
车辆	到达	否	是	否

Section	步骤	旅行前	旅行	旅行后
车辆	Continue	否	是	否
车辆	ContinueHighway	否	是	否
车辆	出发	否	是	否
车辆	Exit	否	是	否
行人	到达	否	是	否
行人	费用	否	是	否
渡轮	Wait	否	否	是
渡轮	板	是	否	否
渡轮	Deboard	否	否	是

计算路径矩阵

矩阵路径服务计算路径矩阵，提供多个起点和目的地之间的行驶时间或距离。该服务提供灵活的自定义选项，允许您指定出行模式、交通状况和其他路径参数。矩阵计算的大小和形状可能各不相同，支持正方形和非方形矩阵，并适应动态或自由流动的交通数据。

使用案例

- 优化物流和电子商务的配送路线：高效计算多个提货和配送地点之间的旅行时间和距离，以优化路线。物流公司可以使用此功能通过规划跨城市的高效路径来最大限度地降低成本和交货时间。它非常适合为当日或次日服务设置优化的配送窗口，以及规划多站配送路线。
- 在@@ 拼车应用程序中匹配司机和乘客：使用路线计算，通过查找不同地点之间最快的路线，将司机与最近的乘客进行匹配。拼车应用程序可以提供实时的司机到达估算值，确保及时上车和下车，从而增强用户体验。支持各种运输方式，如汽车、自行车和踏板车。
- 为车队管理规划和优化路线：通过优化路线来管理大型车队，从而减少油耗和旅行时间。车队经理可以为多个停靠点的车辆分配最高效的路线，从而提高整体运营效率。用例包括服务车队、运输公司和公用事业，在这些情况下，最佳的路线规划对于实地考察至关重要。

理解请求

该请求包括用于路径计算的起点和目的地，以及用于根据偏好和约束条件定制矩阵的可选参数。有关更多详细信息，请参阅计算路由矩阵 API 的 API 参考。

- `Origins`：以经度和纬度表示的原点坐标列表。
- `Destinations`: 目的地坐标列表。
- `OptimizeRoutingFor`：优化标准，例如 “最快” 或 “最短” 路线。
- `RoutingBoundary`：将计算边界定义为 “无界” 或仅限于特定的几何图形。
- `Avoid`：计算路径时要避免的功能。如果找不到可行的路线，则忽略。
- `Traffic`：影响路径计算的与交通相关的选项。

了解回应

响应包括起点和目的地之间的计算路径矩阵，以及距离和持续时间等详细信息。如果适用，还会提供路线的错误和边界。有关计算路由矩阵 API 的更多详细信息，请参阅 API 参考。

- `RouteMatrix`: 包含起点和目的地之间行驶距离和持续时间的矩阵。
- `ErrorCount`：计算路径时遇到的错误数。
- `RoutingBoundary`：计算矩阵的边界。

计算等值线

计算等值线 API 允许您确定在指定时间或距离限制内可到达的区域。通过考虑道路限制、交通状况和出行模式等因素，它可以生成轮廓无障碍区域的等值线，为城市规划、物流和服务无障碍应用提供支持。此 API 可用于城市规划、房地产分析和无障碍研究，方法是在设定的时间范围或距离限制内对服务范围、交通选择或资源进行可视化。通过在地图上显示这些等值线，用户可以评估特定限制范围内的旅行覆盖范围，从而增强选址、服务覆盖范围和资源分配的决策。

使用案例

- 通过旅行时间等值线评估医疗保健的可及性：生成等值线，根据出行时间评估来自不同社区的医疗机构的访问情况。医疗保健组织可以使用此功能来识别服务不足的地区，并就诊所位置或移动医疗服务做出明智的决定，从而改善社区医疗保健的可及性。

- 使用旅行时间等值线分析零售扩张的市场覆盖范围：创建等值线，根据出行时间来表示客户进入零售地点的渠道。零售企业可以评估新的门店位置并了解客户的人口统计数据，利用这些可视化来战略性地扩大和优化销售潜力。
- 使用等值线优化物流和配送区域：生成等值线以根据时间敏感的物流要求定义交货区域。物流公司可以对配送中心在特定时间范围内到达的区域进行可视化，从而改善路线规划、运营效率和及时交货。
- 使用等@@ 值线规划旅游和娱乐设施：可视化从旅游景点到附近住宿的旅行时间。旅游局可以通过在地图上显示这些等值线来帮助旅行者找到便捷的住宿选择，促进当地企业的发展，并改善旅行体验。
- 通过等值线改善应急响应计划：生成等值线以评估从紧急服务地点到社区各个区域的响应时间。应急管理团队可以在关键响应时间内确定区域，优化资源分配以改善事故期间的响应。
- 使用等值线分析通勤模式以进行劳动力规划：生成等值线以可视化通勤时间并识别出行时间较长的区域。企业可以将这些见解用于远程工作政策或办公室搬迁，从而提高员工的满意度和工作效率。

理解请求

该请求接受诸如`Origin`、和之类的参数`Destination`，`Thresholds`用于定义等值线。可选参数，例如`AllowAvoid`、和`TravelModeOptions`，允许自定义等值线约束。有关更多信息，请参阅。

Origin

以经度和纬度计算等值线的起点。

阈值

用于定义等值线边界的时间或距离限制。

TravelMode

交通方式，例如汽车、行人或卡车。

OptimizeIsolineFor

优化标准，例如 “精确”、“平衡” 或 “快速计算”。

DepartureTime

计算瞬态等值线的出发时间（如果已指定）。

ArrivalTime

计算瞬态等值线的到达时间（如果已指定）。

IsolineGranularity

等值线边界的最大点数和分辨率。

了解回应

响应包括具有以下属性的等值线`IsolineGeometryFormat`，根据请求参数概述可到达区域。

等值线

计算出具有相关属性的等值线，包括几何形状和连接。

几何形状

概述计算出的等值线边界的几何列表。

连接

等值线几何之间的连接，包括每个连接的几何。

优化航点

Optimize Waypoints API 会计算访问沿途多个航点的最有效顺序。此 API 使用高级算法，在考虑交通状况、避让和车辆规格等因素的同时，最大限度地减少了行驶时间和距离。集成 Optimize Waypoints API 可帮助企业简化运营、降低油耗、提高交付效率并提高客户满意度。API 提供优化的路线，可在多站出行场景中实现更好的决策和资源分配。

使用案例

- 提高多站配送效率：有效优化多个配送停靠点的顺序，以减少旅行时间和成本。配送服务可以通过计算最有效的司机路线、最大限度地减少燃油费用和确保及时交付来简化运营，从而提高客户满意度和运营效率。
- 简化现场服务操作：优化一天内对多个工作地点的访问顺序，减少现场服务技术人员的差旅时间。这使公司能够每天完成更多工作，从而提高生产力和服务交付。
- 为旅行社规划高效的旅游路线：优化包含多个景点的行程，以最大限度地提高观光率，同时最大限度地减少旅行时间。旅行社可以使用此功能来制定最佳的导游计划，通过更好地利用可用时间来改善整体游客体验。
- 提高拼车司机的效率：优化多名乘客的接送顺序，减少等待时间并增强乘客体验。拼车服务可以最大限度地提高司机的收入，并通过优化航点确保为乘客提供及时的服务。

- 优化垃圾收集服务路线：规划垃圾收集路线，最大限度地缩短旅行距离和时间，这有助于废物管理公司简化运营并确保及时收集，从而节省成本并减少对环境的影响。
- 协调活动和会议的物流：管理运输物流，将设备和用品运送到多个场地，优化装卸路线。这使活动策划者能够简化交通，减少延误，并确保活动材料及时送达。
- 增强应急响应路线：规划多个紧急情况的最快路线，优化危急情况下的响应时间。紧急服务可以提高响应效率，通过使用经过优化的航点，有可能挽救生命。
- 促进现场代表的销售路线规划：优化销售代表一天内拜访多个客户的路线，最大限度地减少差旅时间并最大限度地提高客户访问量。这可以帮助公司提高生产力，抓住更多的销售机会。

理解请求

该请求需要诸如Origin和之类的参数Waypoints来计算优化的序列。可选参数Avoid，例如Traffic、和，Driver允许进行其他自定义。

航点

要按顺序优化的航点列表。

Origin

优化路径的起始位置。

目标位置

路径的可选终点位置以进行优化。

OptimizeSequencingFor

排序优化的标准，例如最快或最短路径。

交通

影响路径计算的与交通相关的选项。

驱动程序

驾驶员的工作和休息周期，以确保遵守区域驾驶法规。

集群

聚类允许您指定如何对附近的航点进行聚类以改善优化的顺序。

每个航点还可以指定必须满足的约束条件，例如 AppointmentTimeAccessHours，以及像Before其他航点一样对约束进行排序。

了解回应

该响应提供了经过优化的航点顺序的详细信息，包括`OptimizedWaypoints`整个航点序列`Distance`和`Duration`旅程顺序。

OptimizedWaypoints

按优化顺序排列的航点列表。

ImpedingWaypoints

阻止优化顺序的航点，包括未满足的失败约束。

连接

有关航点之间旅行的详细信息，包括距离和持续时间。

TimeBreakdown

路径的总时长`TravelRestService`、、和`Wait`持续时间明细。

ClusterIndex

与航点关联的集群的索引。只有在处理请求时进行了群集时，才会将索引包含在响应中。

Snap to R

Snap to Road API 通过将 GPS 坐标与数字地图上最近的路段对齐，提高了地理定位的准确性。此 API 获取通常从移动设备或车辆收集的原始经度和纬度数据，然后将这些点 “捕捉” 到相应的道路网络，从而纠正由于 GPS 漂移或信号丢失导致的不准确性。通过集成 Snap to Road API，您可以确保您的应用程序提供可靠和准确的数据，从而在各种场景中支持更好的决策和运营效率。

使用案例

- 通过捕捉道路网络来提高导航精度：将 GPS 坐标有效地与最近的路段对齐，以提高导航精度。此功能对于地图和导航服务非常有价值，可为用户提供精确的方向和实时的位置更新，从而增强导航体验。
- 提高车队管理应用程序的数据准确性：通过将车辆的 GPS 坐标捕捉到最近的道路，从而更正报告的车辆位置。在车队管理系统中，此功能可确保精确的车辆跟踪，从而优化物流并提高运营效率。

理解请求

该请求`TracePoints`需要与道路匹配，并使用可选参数，例如`SnappedGeometryFormat`和`SnapRadius`来控制几何格式和捕捉半径。

TracePoints

要捕捉到道路网络的坐标列表。

SnappedGeometryFormat

返回的几何图形的格式，例如 “FlexiblePolyline” 或 “简单”。

SnapRadius

考虑道路捕捉的追踪点周围的半径。

了解回应

响应包含校正后的几何图形和捕捉的追踪点，`SnappedTracePoints`其属性如`SnappedGeometry`和表示精度和捕捉置信度。

SnappedGeometry

已捕捉路径的修正几何图形。

SnappedTracePoints

追踪点的调整坐标与道路对齐。

版权声明

有关捕捉过程的警告或信息性消息。

操作方法

本节提供有关利用路径 APIs 解决路由和导航难题的指南。这些教程涵盖了将路由功能集成到应用程序中的基本任务。每个主题都侧重于特定的用例，使您能够高效地实现根据业务需求量身定制的高级路由功能。

主题

- 学习如何使用 CalculateRoutes
- 学习如何使用 CalculateIsolines
- 学习如何使用 CalculateRouteMatrix
- 学习如何使用 OptimizeWaypoints
- 学习如何使用 SnapToRoads

学习如何使用 CalculateRoutes

本节提供使用 step-by-step说明`CalculateRoutes`。本主题详细介绍了如何查找具有特定配置的路线，例如整合 turn-by-turn导航、设置出行模式和添加航点。

主题

- 如何查找出发地和目的地的路线
- 如何查找带有 turn-by-turn方向的路线
- 如何找到路段的限速
- 如何寻找备选路线
- 如何计算路线的通行费
- 如何创建具有自定义避让功能的路线
- 如何通过自定义避开多个潜在物品来创建路线
- 如何找到最快的路线
- 如何找到最短的路线

如何查找出发地和目的地的路线

CalculateRoutes API 可帮助您找到出发地和目的地之间的最佳路线，以及司机休息的最佳机会。它支持卡车、行人、汽车和踏板车等出行模式。它还支持多达 25 个航点（中途停留），包括起点和目的地，但仅有一些限制。

潜在用例

- 查找 point-to-point路线：根据各种出行模式和其他选项，确定两个地点之间的最佳路线。

示例

使用 Car 计算从起点到目的地的路线 TravelMode

Sample request

```
{
    "Origin": [
        -123.118105,
        49.282423
    ],
    "Destination": [
        -123.020098,
        49.232872
    ]
}
```

Sample response

```
{
    "LegGeometryFormat": "FlexiblePolyline",
    "Notices": [],
    "Routes": [
        {
            "Legs": [
                {
                    "Geometry": {
                        "Polyline": "Redacted"
                    },
                    "TravelMode": "Car",
                    "Type": "Vehicle",
                    "VehicleLegDetails": {
                        "AfterTravelSteps": [],
                        "Arrival": {
                            "Place": {
                                "ChargingStation": false,
                                "OriginalPosition": [
                                    -123.020098,
                                    49.232872
                                ],
                                "Position": [
                                    -123.0203051,
                                    49.2328499
```

```
                ]
            }
        },
        "Departure": {
            "Place": {
                "ChargingStation": false,
                "OriginalPosition": [
                    -123.1181051,
                    49.282423
                ],
                "Position": [
                    -123.1180883,
                    49.2824349
                ]
            }
        },
        "TravelSteps": [
            {
                "Distance": 1288,
                "Duration": 102,
                "Type": "Depart"
            },
            {
                "Distance": 262,
                "Duration": 24,
                "Type": "Ramp"
            },
            {
                "Distance": 1356,
                "Duration": 134,
                "Type": "Turn"
            },
            {
                "Distance": 7092,
                "Duration": 568,
                "Type": "Keep"
            },
            {
                "Distance": 65,
                "Duration": 26,
                "Type": "Turn"
            },
            {
                "Distance": 50,
```

```
                                "Duration": 18,
                                "Type": "Turn"
                            },
                            {
                                "Distance": 0,
                                "Duration": 0,
                                "Type": "Arrive"
                            }
                        ]
                    }
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/routes?key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
  "Origin": [
    -123.118105,
    49.282423
  ],
  "Destination": [
    -123.020098,
    49.232872
  ]
}'
```

AWS CLI

```
aws geo-routes calculate-routes --key ${YourKey} \
--origin -123.118105 49.282423 \
--destination -123.020098 49.232872
```

如何查找带有 turn-by-turn方向的路线

CalculateRoutes API 可帮助您找到出发地和目的地之间的最佳路线，以及司机休息的最佳机会。它支持汽车、卡车、行人和摩托车等出行模式。它还支持多达 25 个航点（中途停留），包括起点和目的地，但仅有一些限制。

潜在用例

- 创建导航移动应用程序：使用 API 获取 turn-by-turn导航说明。
- 在 Web 平台上显示路线：显示 Web 应用程序的详细路线指南，以帮助您进行导航。

示例

使用汽车计算路线 TravelMode

Sample request

```
{
    "Origin": [
        -123.118105,
        49.282423
    ],
    "Destination": [
        -123.020098,
        49.232872
    ],
    "TravelStepType": "TurnByTurn",
    "TravelMode": "Car"
}
```

Sample response

```
{
    "LegGeometryFormat": "FlexiblePolyline",
    "Notices": [],
    "Routes": [
        {
            "Legs": [
                {
                    "Geometry": {
                        "Polyline": "Redacted"
                    },
```

```
                    "TravelMode": "Car",
                    "Type": "Vehicle",
                    "VehicleLegDetails": {
                        "Arrival": {
                            "Place": {
                                "Position": [-123.0203051, 49.2328499]
                            }
                        },
                        "Departure": {
                            "Place": {
                                "Position": [-123.1180883, 49.2824349]
                            }
                        },
                        "TravelSteps": [
                            {
                                "Distance": 1288,
                                "Duration": 102,
                                "Type": "Depart",
                                "NextRoad": {
                                    "RoadName": "W Georgia St",
                                    "RouteNumber": "HWY-1A"
                                }
                            },
                            {
                                "Distance": 262,
                                "Duration": 24,
                                "Type": "Keep",
                                "NextRoad": {
                                    "RoadName": "Main St",
                                    "RouteNumber": "HWY-1A"
                                }
                            },
                            {
                                "Distance": 1356,
                                "Duration": 134,
                                "Type": "Turn",
                                "NextRoad": {
                                    "RoadName": "Main St",
                                    "RouteNumber": "HWY-1A"
                                }
                            },
                            {
                                "Distance": 7092,
                                "Duration": 568,
```

```
                                "Type": "Keep",
                                "NextRoad": {
                                    "RoadName": "Kingsway",
                                    "RouteNumber": "HWY-1A"
                                }
                            },
                            {
                                "Distance": 65,
                                "Duration": 26,
                                "Type": "Turn"
                            },
                            {
                                "Distance": 0,
                                "Duration": 0,
                                "Type": "Arrive"
                            }
                        ]
                    }
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/routes?key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
  "Origin": [
    -123.118105,
    49.282423
  ],
  "Destination": [
    -123.020098,
    49.232872
  ],
  "TravelStepType": "TurnByTurn",
  "TravelMode": "Car"
}'
```

AWS CLI

```
aws geo-routes calculate-routes --key ${YourKey} \
--origin -123.118105 49.282423 \
--destination -123.020098 49.232872 \
--travel-step-type "TurnByTurn" \
--travel-mode "Car"
```

如何找到路段的限速

CalculateRoutes API 可帮助您找到出发地和目的地之间的最佳路线，以及司机休息的最佳机会。它支持汽车、卡车、行人和摩托车等出行模式。它还支持多达 25 个航点（中途停留），包括起点和目的地，但仅有一些限制。

潜在用例

- 审计限速合规性：随时了解限速及其合规性。
- 将限速通知驾驶员：如果可能不清楚限速，则通知驾驶员限速。

示例

在汽车中查找限速 TravelMode

Sample request

```
{
    "Origin": [
        13.055211,
        52.704802
    ],
    "Destination": [
        13.551910,
        52.282705
    ],
    "TravelMode": "Car",
    "SpanAdditionalFeatures": ["SpeedLimit"]
}
```

Sample response


```
{
    "LegGeometryFormat": "FlexiblePolyline",
    "Notices": [],
    "Routes": [
        {
            "Legs": [
                {
                    "Geometry": {
                        "Polyline": "Redacted"
                    },
                    "TravelMode": "Car",
                    "Type": "Vehicle",
                    "VehicleLegDetails": {
                        "AfterTravelSteps": [],
                        "Arrival": {
                            "Place": {
                                "ChargingStation": false,
                                "OriginalPosition": [
                                    13.55191,
                                    52.282705
                                ],
                                "Position": [
                                    13.5507836,
                                    52.2859121
                                ]
                            }
                        },
                        "Departure": {
                            "Place": {
                                "ChargingStation": false,
                                "OriginalPosition": [
                                    13.055211,
                                    52.704802
                                ],
                                "Position": [
                                    13.0555036,
                                    52.7056073
                                ]
                            }
                        },
                        "Incidents": [],
                        "Notices": [],
```

```
                        "PassThroughWaypoints": [],
                        "Spans": [
                            {
                                "GeometryOffset": 0,
                                "SpeedLimit": {
                                    "Unlimited": true
                                }
                            },
                            {
                                "GeometryOffset": 151,
                                "SpeedLimit": {
                                    "MaxSpeed": 120.00000762939453
                                }
                            },
                            {
                                "GeometryOffset": 167,
                                "SpeedLimit": {
                                    "MaxSpeed": 80
                                }
                            },
                            {
                                "GeometryOffset": 195,
                                "SpeedLimit": {
                                    "MaxSpeed": 120.00000762939453
                                }
                            },
                            {
                                "GeometryOffset": 220,
                                "SpeedLimit": {
                                    "Unlimited": true
                                }
                            },
                            {
                                "GeometryOffset": 356,
                                "SpeedLimit": {
                                    "MaxSpeed": 120.00000762939453
                                }
                            },
                            {
                                "GeometryOffset": 358,
                                "SpeedLimit": {
                                    "MaxSpeed": 100
                                }
                            },
```

```
                        {
                            "GeometryOffset": 368,
                            "SpeedLimit": {
                                "MaxSpeed": 80
                            }
                        },
                        {
                            "GeometryOffset": 384,
                            "SpeedLimit": {
                                "MaxSpeed": 60.000003814697266
                            }
                        },
                        {
                            "GeometryOffset": 639,
                            "SpeedLimit": {
                                "MaxSpeed": 80
                            }
                        },
                        {
                            "GeometryOffset": 701,
                            "SpeedLimit": {
                                "MaxSpeed": 60.000003814697266
                            }
                        },
                        {
                            "GeometryOffset": 726,
                            "SpeedLimit": {
                                "MaxSpeed": 80
                            }
                        },
                        {
                            "GeometryOffset": 805,
                            "SpeedLimit": {
                                "MaxSpeed": 60.000003814697266
                            }
                        },
                        {
                            "GeometryOffset": 839,
                            "SpeedLimit": {
                                "MaxSpeed": 80
                            }
                        },
                        {
                            "GeometryOffset": 1384,
```

```
                    "SpeedLimit": {
                        "MaxSpeed": 60.000003814697266
                    }
                },
                {
                    "GeometryOffset": 1393,
                    "SpeedLimit": {
                        "MaxSpeed": 50
                    }
                },
                {
                    "GeometryOffset": 1443,
                    "SpeedLimit": {
                        "MaxSpeed": 30.000001907348633
                    }
                },
                {
                    "GeometryOffset": 1454,
                    "SpeedLimit": {
                        "MaxSpeed": 50
                    }
                },
                {
                    "GeometryOffset": 1504,
                    "SpeedLimit": {
                        "MaxSpeed": 60.000003814697266
                    }
                },
                {
                    "GeometryOffset": 1513,
                    "SpeedLimit": {
                        "MaxSpeed": 80
                    }
                },
                {
                    "GeometryOffset": 1516,
                    "SpeedLimit": {
                        "MaxSpeed": 60.000003814697266
                    }
                },
                {
                    "GeometryOffset": 1555,
                    "SpeedLimit": {
                        "MaxSpeed": 80
```

```
                        }
                    },
                    {
                        "GeometryOffset": 1748,
                        "SpeedLimit": {
                            "MaxSpeed": 120.00000762939453
                        }
                    },
                    {
                        "GeometryOffset": 1904,
                        "SpeedLimit": {
                            "MaxSpeed": 100
                        }
                    },
                    {
                        "GeometryOffset": 1945,
                        "SpeedLimit": {
                            "Unlimited": true
                        }
                    },
                    {
                        "GeometryOffset": 2006,
                        "SpeedLimit": {
                            "MaxSpeed": 70
                        }
                    },
                    {
                        "GeometryOffset": 2017,
                        "SpeedLimit": {
                            "MaxSpeed": 50
                        }
                    },
                    {
                        "GeometryOffset": 2033,
                        "SpeedLimit": {
                            "MaxSpeed": 30.000001907348633
                        }
                    }
                ],
                "Tolls": [],
                "TollSystems": [],
                "TravelSteps": [
                    {
                        "Distance": 9099,
```

```
                "Duration": 262,
                "ExitNumber": [],
                "GeometryOffset": 0,
                "Type": "Depart"
            },
            {
                "Distance": 18849,
                "Duration": 800,
                "ExitNumber": [],
                "GeometryOffset": 162,
                "KeepStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right"
                },
                "Type": "Keep"
            },
            {
                "Distance": 4290,
                "Duration": 212,
                "ExitNumber": [],
                "ExitStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right"
                },
                "GeometryOffset": 701,
                "Type": "Exit"
            },
            {
                "Distance": 14418,
                "Duration": 651,
                "ExitNumber": [],
                "GeometryOffset": 828,
                "Type": "Continue"
            },
            {
                "Distance": 1255,
                "Duration": 62,
                "ExitNumber": [],
                "ExitStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right"
                },
                "GeometryOffset": 1359,
                "Type": "Exit"
```

```
                    },
                    {
                        "Distance": 1607,
                        "Duration": 139,
                        "ExitNumber": [],
                        "GeometryOffset": 1393,
                        "TurnStepDetails": {
                            "Intersection": [],
                            "SteeringDirection": "Left",
                            "TurnIntensity": "Typical"
                        },
                        "Type": "Turn"
                    },
                    {
                        "Distance": 934,
                        "Duration": 92,
                        "ExitNumber": [],
                        "GeometryOffset": 1442,
                        "Type": "Continue"
                    },
                    {
                        "Distance": 320,
                        "Duration": 27,
                        "ExitNumber": [],
                        "GeometryOffset": 1473,
                        "Type": "Continue"
                    },
                    {
                        "Distance": 320,
                        "Duration": 33,
                        "ExitNumber": [],
                        "GeometryOffset": 1491,
                        "Type": "Continue"
                    },
                    {
                        "Distance": 17863,
                        "Duration": 680,
                        "ExitNumber": [],
                        "GeometryOffset": 1504,
                        "RampStepDetails": {
                            "Intersection": [],
                            "SteeringDirection": "Right"
                        },
                        "Type": "Ramp"
```

```
                },
                {
                    "Distance": 2888,
                    "Duration": 95,
                    "ExitNumber": [],
                    "GeometryOffset": 1917,
                    "Type": "Continue"
                },
                {
                    "Distance": 321,
                    "Duration": 28,
                    "ExitNumber": [],
                    "ExitStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Right"
                    },
                    "GeometryOffset": 1977,
                    "Type": "Exit"
                },
                {
                    "Distance": 549,
                    "Duration": 41,
                    "ExitNumber": [],
                    "GeometryOffset": 2006,
                    "TurnStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Right",
                        "TurnIntensity": "Typical"
                    },
                    "Type": "Turn"
                },
                {
                    "Distance": 386,
                    "Duration": 51,
                    "ExitNumber": [],
                    "GeometryOffset": 2023,
                    "TurnStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Left",
                        "TurnIntensity": "Typical"
                    },
                    "Type": "Turn"
                },
                {
```

```
                                "Distance": 98,
                                "Duration": 15,
                                "ExitNumber": [],
                                "GeometryOffset": 2033,
                                "TurnStepDetails": {
                                    "Intersection": [],
                                    "SteeringDirection": "Right",
                                    "TurnIntensity": "Typical"
                                },
                                "Type": "Turn"
                            },
                            {
                                "Distance": 0,
                                "Duration": 0,
                                "ExitNumber": [],
                                "GeometryOffset": 2036,
                                "Type": "Arrive"
                            }
                        ],
                        "TruckRoadTypes": [],
                        "Zones": []
                    }
                }
            ],
            "MajorRoadLabels": [
                {
                    "RouteNumber": {
                        "Language": "de",
                        "Value": "A111"
                    }
                },
                {
                    "RouteNumber": {
                        "Language": "de",
                        "Value": "A113"
                    }
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/routes?key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
  "Origin": [
    13.055211,
    52.704802
  ],
  "Destination": [
    13.551910,
    52.282705
  ],
  "TravelMode": "Car",
  "SpanAdditionalFeatures": ["SpeedLimit"]
}'
```

AWS CLI

```
aws geo-routes calculate-routes --key ${YourKey} \
--origin 13.055211 52.704802 \
--destination 13.551910 52.282705 \
--travel-mode "Car" \
--span-additional-features "SpeedLimit"
```

如何寻找备选路线

CalculateRoutes API 可帮助您找到出发地和目的地之间的最佳路线，以及司机休息的最佳机会。它支持汽车、卡车、行人和摩托车等出行模式。它还支持多达 25 个航点（中途停留），包括起点和目的地，但仅有一些限制。

潜在用例

- 确定备选路线：能够选择适合您业务需求的最佳路线。

示例

在 Car 中查找其他路线 TravelMode

Sample request

```
{
    "Origin": [
        13.055211,
        52.704802
    ],
    "Destination": [
        13.551910,
        52.282705
    ],
    "TravelMode": "Car",
    "MaxAlternatives": 2
}
```

Sample response

```
{
    "LegGeometryFormat": "FlexiblePolyline",
    "Notices": [],
    "Routes": [
        {
            "Legs": [
                {
                    "Geometry": {
                        "Polyline": "Redacted"
                    },
                    "TravelMode": "Car",
                    "Type": "Vehicle",
                    "VehicleLegDetails": {
                        "AfterTravelSteps": [],
                        "Arrival": {
                            "Place": {
                                "ChargingStation": false,
                                "OriginalPosition": [
                                    13.55191,
                                    52.282705
                                ],
                                "Position": [
```

```
                    13.5507836,
                    52.2859121
                ]
            }
        },
        "Departure": {
            "Place": {
                "ChargingStation": false,
                "OriginalPosition": [
                    13.055211,
                    52.704802
                ],
                "Position": [
                    13.0555036,
                    52.7056073
                ]
            }
        },
        "Incidents": [],
        "Notices": [],
        "PassThroughWaypoints": [],
        "Spans": [],
        "Tolls": [],
        "TollSystems": [],
        "TravelSteps": [
            {
                "Distance": 9099,
                "Duration": 262,
                "ExitNumber": [],
                "GeometryOffset": 0,
                "Type": "Depart"
            },
            {
                "Distance": 18849,
                "Duration": 800,
                "ExitNumber": [],
                "GeometryOffset": 162,
                "KeepStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right"
                },
                "Type": "Keep"
            },
            {
```

```
                "Distance": 4290,
                "Duration": 212,
                "ExitNumber": [],
                "ExitStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right"
                },
                "GeometryOffset": 701,
                "Type": "Exit"
            },
            {
                "Distance": 14418,
                "Duration": 651,
                "ExitNumber": [],
                "GeometryOffset": 828,
                "Type": "Continue"
            },
            {
                "Distance": 1255,
                "Duration": 62,
                "ExitNumber": [],
                "ExitStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right"
                },
                "GeometryOffset": 1359,
                "Type": "Exit"
            },
            {
                "Distance": 1607,
                "Duration": 139,
                "ExitNumber": [],
                "GeometryOffset": 1393,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Left",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
            {
                "Distance": 934,
                "Duration": 92,
                "ExitNumber": [],
```

```
                "GeometryOffset": 1442,
                "Type": "Continue"
            },
            {
                "Distance": 320,
                "Duration": 27,
                "ExitNumber": [],
                "GeometryOffset": 1473,
                "Type": "Continue"
            },
            {
                "Distance": 320,
                "Duration": 33,
                "ExitNumber": [],
                "GeometryOffset": 1491,
                "Type": "Continue"
            },
            {
                "Distance": 17863,
                "Duration": 680,
                "ExitNumber": [],
                "GeometryOffset": 1504,
                "RampStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right"
                },
                "Type": "Ramp"
            },
            {
                "Distance": 2888,
                "Duration": 95,
                "ExitNumber": [],
                "GeometryOffset": 1917,
                "Type": "Continue"
            },
            {
                "Distance": 321,
                "Duration": 28,
                "ExitNumber": [],
                "ExitStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right"
                },
                "GeometryOffset": 1977,
```

```
            "Type": "Exit"
        },
        {
            "Distance": 549,
            "Duration": 41,
            "ExitNumber": [],
            "GeometryOffset": 2006,
            "TurnStepDetails": {
                "Intersection": [],
                "SteeringDirection": "Right",
                "TurnIntensity": "Typical"
            },
            "Type": "Turn"
        },
        {
            "Distance": 386,
            "Duration": 51,
            "ExitNumber": [],
            "GeometryOffset": 2023,
            "TurnStepDetails": {
                "Intersection": [],
                "SteeringDirection": "Left",
                "TurnIntensity": "Typical"
            },
            "Type": "Turn"
        },
        {
            "Distance": 98,
            "Duration": 15,
            "ExitNumber": [],
            "GeometryOffset": 2033,
            "TurnStepDetails": {
                "Intersection": [],
                "SteeringDirection": "Right",
                "TurnIntensity": "Typical"
            },
            "Type": "Turn"
        },
        {
            "Distance": 0,
            "Duration": 0,
            "ExitNumber": [],
            "GeometryOffset": 2036,
            "Type": "Arrive"
```

```
                        }
                    ],
                    "TruckRoadTypes": [],
                    "Zones": []
                }
            }
        ],
        "MajorRoadLabels": [
            {
                "RouteNumber": {
                    "Language": "de",
                    "Value": "A111"
                }
            },
            {
                "RouteNumber": {
                    "Language": "de",
                    "Value": "A113"
                }
            }
        ]
    },
    {
        "Legs": [
            {
                "Geometry": {
                    "Polyline": "Redacted"
                },
                "TravelMode": "Car",
                "Type": "Vehicle",
                "VehicleLegDetails": {
                    "AfterTravelSteps": [],
                    "Arrival": {
                        "Place": {
                            "ChargingStation": false,
                            "OriginalPosition": [
                                13.55191,
                                52.282705
                            ],
                            "Position": [
                                13.5507836,
                                52.2859121
                            ]
                        }
```

```
                    },
                    "Departure": {
                        "Place": {
                            "ChargingStation": false,
                            "OriginalPosition": [
                                13.055211,
                                52.704802
                            ],
                            "Position": [
                                13.0555036,
                                52.7056073
                            ]
                        }
                    },
                    "Incidents": [],
                    "Notices": [],
                    "PassThroughWaypoints": [],
                    "Spans": [],
                    "Tolls": [],
                    "TollSystems": [],
                    "TravelSteps": [
                        {
                            "Distance": 91047,
                            "Duration": 2880,
                            "ExitNumber": [],
                            "GeometryOffset": 0,
                            "Type": "Depart"
                        },
                        {
                            "Distance": 3496,
                            "Duration": 119,
                            "ExitNumber": [],
                            "ExitStepDetails": {
                                "Intersection": [],
                                "SteeringDirection": "Right"
                            },
                            "GeometryOffset": 1473,
                            "Type": "Exit"
                        },
                        {
                            "Distance": 321,
                            "Duration": 28,
                            "ExitNumber": [],
                            "ExitStepDetails": {
```

```
                        "Intersection": [],
                        "SteeringDirection": "Right"
                    },
                    "GeometryOffset": 1565,
                    "Type": "Exit"
                },
                {
                    "Distance": 549,
                    "Duration": 41,
                    "ExitNumber": [],
                    "GeometryOffset": 1594,
                    "TurnStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Right",
                        "TurnIntensity": "Typical"
                    },
                    "Type": "Turn"
                },
                {
                    "Distance": 386,
                    "Duration": 51,
                    "ExitNumber": [],
                    "GeometryOffset": 1611,
                    "TurnStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Left",
                        "TurnIntensity": "Typical"
                    },
                    "Type": "Turn"
                },
                {
                    "Distance": 98,
                    "Duration": 15,
                    "ExitNumber": [],
                    "GeometryOffset": 1621,
                    "TurnStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Right",
                        "TurnIntensity": "Typical"
                    },
                    "Type": "Turn"
                },
                {
                    "Distance": 0,
```

```
                            "Duration": 0,
                            "ExitNumber": [],
                            "GeometryOffset": 1624,
                            "Type": "Arrive"
                        }
                    ],
                    "TruckRoadTypes": [],
                    "Zones": []
                }
            }
        ],
        "MajorRoadLabels": [
            {
                "RouteNumber": {
                    "Language": "de",
                    "Value": "A10"
                }
            }
        ]
    },
    {
        "Legs": [
            {
                "Geometry": {
                    "Polyline": "Redacted"
                },
                "TravelMode": "Car",
                "Type": "Vehicle",
                "VehicleLegDetails": {
                    "AfterTravelSteps": [],
                    "Arrival": {
                        "Place": {
                            "ChargingStation": false,
                            "OriginalPosition": [
                                13.55191,
                                52.282705
                            ],
                            "Position": [
                                13.5507836,
                                52.2859121
                            ]
                        }
                    },
                    "Departure": {
```

```
            "Place": {
                "ChargingStation": false,
                "OriginalPosition": [
                    13.055211,
                    52.704802
                ],
                "Position": [
                    13.0555036,
                    52.7056073
                ]
            }
        },
        "Incidents": [],
        "Notices": [],
        "PassThroughWaypoints": [],
        "Spans": [],
        "Tolls": [],
        "TollSystems": [],
        "TravelSteps": [
            {
                "Distance": 9099,
                "Duration": 262,
                "ExitNumber": [],
                "GeometryOffset": 0,
                "Type": "Depart"
            },
            {
                "Distance": 18849,
                "Duration": 800,
                "ExitNumber": [],
                "GeometryOffset": 162,
                "KeepStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right"
                },
                "Type": "Keep"
            },
            {
                "Distance": 4290,
                "Duration": 212,
                "ExitNumber": [],
                "ExitStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right"
```

```
                    },
                    "GeometryOffset": 701,
                    "Type": "Exit"
                },
                {
                    "Distance": 12258,
                    "Duration": 554,
                    "ExitNumber": [],
                    "GeometryOffset": 828,
                    "Type": "Continue"
                },
                {
                    "Distance": 18567,
                    "Duration": 1218,
                    "ExitNumber": [],
                    "ExitStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Right"
                    },
                    "GeometryOffset": 1282,
                    "Type": "Exit"
                },
                {
                    "Distance": 250,
                    "Duration": 28,
                    "ExitNumber": [],
                    "GeometryOffset": 1760,
                    "RampStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Right"
                    },
                    "Type": "Ramp"
                },
                {
                    "Distance": 4856,
                    "Duration": 245,
                    "ExitNumber": [],
                    "GeometryOffset": 1793,
                    "TurnStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Right",
                        "TurnIntensity": "Typical"
                    },
                    "Type": "Turn"
```

```
                },
                {
                    "Distance": 3202,
                    "Duration": 165,
                    "ExitNumber": [],
                    "GeometryOffset": 1970,
                    "RoundaboutExitStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Right"
                    },
                    "Type": "RoundaboutExit"
                },
                {
                    "Distance": 386,
                    "Duration": 51,
                    "ExitNumber": [],
                    "GeometryOffset": 2072,
                    "TurnStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Right",
                        "TurnIntensity": "Typical"
                    },
                    "Type": "Turn"
                },
                {
                    "Distance": 98,
                    "Duration": 15,
                    "ExitNumber": [],
                    "GeometryOffset": 2082,
                    "TurnStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Right",
                        "TurnIntensity": "Typical"
                    },
                    "Type": "Turn"
                },
                {
                    "Distance": 0,
                    "Duration": 0,
                    "ExitNumber": [],
                    "GeometryOffset": 2085,
                    "Type": "Arrive"
                }
            ],
```

```
                        "TruckRoadTypes": [],
                        "Zones": []
                    }
                }
            ],
            "MajorRoadLabels": [
                {
                    "RouteNumber": {
                        "Language": "de",
                        "Value": "A111"
                    }
                },
                {
                    "RouteNumber": {
                        "Language": "de",
                        "Value": "B96"
                    }
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/routes?key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
  "Origin": [
    13.055211,
    52.704802
  ],
  "Destination": [
    13.551910,
    52.282705
  ],
  "TravelMode": "Car",
  "MaxAlternatives": 2
}'
```

AWS CLI

```
aws geo-routes calculate-routes --key ${YourKey} \
--origin 13.055211 52.704802 \
--destination 13.551910 52.282705 \
--travel-mode "Car" \
--max-alternatives 2
```

如何计算路线的通行费

CalculateRoutes API 可帮助您找到出发地和目的地之间的最佳路线，以及司机休息的最佳机会。它支持汽车、卡车、行人和摩托车等出行模式。它还支持多达 25 个航点（中途停留），包括起点和目的地，但仅有一些限制。

潜在用例

- 计算通行费：能够在路线规划期间计算通行费。
- 审计通行费：能够在旅行后审计通行费。

示例

在 Car 中计算路线的通行费 TravelMode

Sample request

```
{
    "Origin": [
        2.234491,
        48.815704
    ],
    "Destination": [
        5.11412,
        47.260723
    ],
    "TravelMode": "Car",
    "Tolls": {
        "AllTransponders": true,
        "AllVignettes": true
    }
}
```

Sample response


```
{
    "LegGeometryFormat": "FlexiblePolyline",
    "Notices": [],
    "Routes": [
        {
            "Legs": [
                {
                    "Geometry": {
                        "Polyline": "Redacted"
                    },
                    "TravelMode": "Car",
                    "Type": "Vehicle",
                    "VehicleLegDetails": {
                        "AfterTravelSteps": [],
                        "Arrival": {
                            "Place": {
                                "ChargingStation": false,
                                "OriginalPosition": [
                                    5.11412,
                                    47.2607229
                                ],
                                "Position": [
                                    5.1128203,
                                    47.2596356
                                ]
                            }
                        },
                        "Departure": {
                            "Place": {
                                "ChargingStation": false,
                                "OriginalPosition": [
                                    2.234491,
                                    48.8157039
                                ],
                                "Position": [
                                    2.2346482,
                                    48.815674
                                ]
                            }
                        },
                        "Incidents": [],
                        "Notices": [],
```

```
                        "PassThroughWaypoints": [],
                        "Spans": [],
                        "Tolls": [
                            {
                                "Country": "FRA",
                                "PaymentSites": [
                                    {
                                        "Name": "FLEURY-EN-BIERE",
                                        "Position": [
                                            2.53988,
                                            48.42578
                                        ]
                                    },
                                    {
                                        "Name": "POUILLY-EN-AUXOIS",
                                        "Position": [
                                            4.56112,
                                            47.25244
                                        ]
                                    }
                                ],
                                "Rates": [
                                    {
                                        "Id": "2d680295-fb0f-45aa-
a8ed-79d9f0f9ff9f",
                                        "LocalPrice": {
                                            "Currency": "EUR",
                                            "Estimate": false,
                                            "Range": false,
                                            "Value": 21.700000762939453
                                        },
                                        "Name": "APRR",
                                        "PaymentMethods": [
                                            "Cash",
                                            "BankCard",
                                            "CreditCard",
                                            "Transponder",
                                            "TravelCard"
                                        ],
                                        "Transponders": [
                                            {
                                                "SystemName": "BipandGo"
                                            },
                                            {
```

```
                                        "SystemName": "BipandGo"
                                    },
                                    {
                                        "SystemName": "BipandGo IDVROOM
carpoorling"
                                    },
                                    {
                                        "SystemName": "Cito30"
                                    },
                                    {
                                        "SystemName": "Easytrip pass"
                                    },
                                    {
                                        "SystemName": "Liane 30"
                                    },
                                    {
                                        "SystemName": "Liber-t"
                                    },
                                    {
                                        "SystemName": "Liber-t mobilitis"
                                    },
                                    {
                                        "SystemName": "Pass Pont-Pont"
                                    },
                                    {
                                        "SystemName": "Progressivi'T
Maurienne"
                                    },
                                    {
                                        "SystemName": "TopEurop"
                                    },
                                    {
                                        "SystemName": "Tunnel Pass+"
                                    },
                                    {
                                        "SystemName": "Ulys"
                                    },
                                    {
                                        "SystemName": "Ulys Europe"
                                    },
                                    {
                                        "SystemName": "Viaduc-t 30"
                                    }
                                ]
```

```
                        }
                    ],
                    "Systems": [
                        0
                    ]
                }
            ],
            "TollSystems": [
                {
                    "Name": "APRR"
                }
            ],
            "TravelSteps": [
                {
                    "Distance": 122,
                    "Duration": 21,
                    "ExitNumber": [],
                    "GeometryOffset": 0,
                    "Type": "Depart"
                },
                {
                    "Distance": 32,
                    "Duration": 7,
                    "ExitNumber": [],
                    "GeometryOffset": 7,
                    "TurnStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Right",
                        "TurnIntensity": "Typical"
                    },
                    "Type": "Turn"
                },
                {
                    "Distance": 74,
                    "Duration": 14,
                    "ExitNumber": [],
                    "GeometryOffset": 8,
                    "TurnStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Right",
                        "TurnIntensity": "Typical"
                    },
                    "Type": "Turn"
                },
```

```
                {
                    "Distance": 199,
                    "Duration": 33,
                    "ExitNumber": [],
                    "GeometryOffset": 9,
                    "TurnStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Right",
                        "TurnIntensity": "Typical"
                    },
                    "Type": "Turn"
                },
                {
                    "Distance": 21,
                    "Duration": 10,
                    "ExitNumber": [],
                    "GeometryOffset": 17,
                    "TurnStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Right",
                        "TurnIntensity": "Typical"
                    },
                    "Type": "Turn"
                },
                {
                    "Distance": 461,
                    "Duration": 63,
                    "ExitNumber": [],
                    "GeometryOffset": 20,
                    "TurnStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Right",
                        "TurnIntensity": "Typical"
                    },
                    "Type": "Turn"
                },
                {
                    "Distance": 1557,
                    "Duration": 145,
                    "ExitNumber": [],
                    "GeometryOffset": 41,
                    "TurnStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Left",
```

```
                "TurnIntensity": "Typical"
            },
            "Type": "Turn"
        },
        {
            "Distance": 3471,
            "Duration": 151,
            "ExitNumber": [],
            "GeometryOffset": 135,
            "RoundaboutExitStepDetails": {
                "Intersection": [],
                "SteeringDirection": "Right"
            },
            "Type": "RoundaboutExit"
        },
        {
            "Distance": 9796,
            "Duration": 430,
            "ExitNumber": [],
            "GeometryOffset": 242,
            "RampStepDetails": {
                "Intersection": [],
                "SteeringDirection": "Right"
            },
            "Type": "Ramp"
        },
        {
            "Distance": 2473,
            "Duration": 112,
            "ExitNumber": [],
            "ExitStepDetails": {
                "Intersection": [],
                "SteeringDirection": "Right"
            },
            "GeometryOffset": 582,
            "Type": "Exit"
        },
        {
            "Distance": 674,
            "Duration": 25,
            "ExitNumber": [],
            "GeometryOffset": 669,
            "Type": "Continue"
        },
```

```
                        {
                            "Distance": 253953,
                            "Duration": 7468,
                            "ExitNumber": [],
                            "GeometryOffset": 679,
                            "Type": "Continue"
                        },
                        {
                            "Distance": 37379,
                            "Duration": 1192,
                            "ExitNumber": [],
                            "ExitStepDetails": {
                                "Intersection": [],
                                "SteeringDirection": "Right"
                            },
                            "GeometryOffset": 5701,
                            "Type": "Exit"
                        },
                        {
                            "Distance": 2021,
                            "Duration": 110,
                            "ExitNumber": [],
                            "GeometryOffset": 6695,
                            "RoundaboutExitStepDetails": {
                                "Intersection": [],
                                "SteeringDirection": "Right"
                            },
                            "Type": "RoundaboutExit"
                        },
                        {
                            "Distance": 7380,
                            "Duration": 316,
                            "ExitNumber": [],
                            "GeometryOffset": 6776,
                            "RoundaboutPassStepDetails": {
                                "Intersection": [],
                                "SteeringDirection": "Right"
                            },
                            "Type": "RoundaboutPass"
                        },
                        {
                            "Distance": 4253,
                            "Duration": 156,
                            "ExitNumber": [],
```

```
                    "GeometryOffset": 6987,
                    "KeepStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Left"
                    },
                    "Type": "Keep"
                },
                {
                    "Distance": 813,
                    "Duration": 54,
                    "ExitNumber": [],
                    "GeometryOffset": 7081,
                    "RampStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Right"
                    },
                    "Type": "Ramp"
                },
                {
                    "Distance": 1909,
                    "Duration": 109,
                    "ExitNumber": [],
                    "GeometryOffset": 7133,
                    "TurnStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Left",
                        "TurnIntensity": "Sharp"
                    },
                    "Type": "Turn"
                },
                {
                    "Distance": 827,
                    "Duration": 47,
                    "ExitNumber": [],
                    "GeometryOffset": 7182,
                    "Type": "Continue"
                },
                {
                    "Distance": 680,
                    "Duration": 41,
                    "ExitNumber": [],
                    "GeometryOffset": 7216,
                    "RoundaboutExitStepDetails": {
                        "Intersection": [],
```

```
                        "SteeringDirection": "Right"
                    },
                    "Type": "RoundaboutExit"
                },
                {
                    "Distance": 224,
                    "Duration": 12,
                    "ExitNumber": [],
                    "GeometryOffset": 7258,
                    "Type": "Continue"
                },
                {
                    "Distance": 25,
                    "Duration": 2,
                    "ExitNumber": [],
                    "GeometryOffset": 7265,
                    "Type": "Continue"
                },
                {
                    "Distance": 748,
                    "Duration": 51,
                    "ExitNumber": [],
                    "GeometryOffset": 7267,
                    "Type": "Continue"
                },
                {
                    "Distance": 656,
                    "Duration": 46,
                    "ExitNumber": [],
                    "GeometryOffset": 7299,
                    "RoundaboutExitStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Right"
                    },
                    "Type": "RoundaboutExit"
                },
                {
                    "Distance": 1119,
                    "Duration": 78,
                    "ExitNumber": [],
                    "GeometryOffset": 7329,
                    "Type": "Continue"
                },
                {
```

```
                "Distance": 432,
                "Duration": 26,
                "ExitNumber": [],
                "GeometryOffset": 7383,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
            {
                "Distance": 1864,
                "Duration": 100,
                "ExitNumber": [],
                "GeometryOffset": 7402,
                "RoundaboutExitStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right"
                },
                "Type": "RoundaboutExit"
            },
            {
                "Distance": 1110,
                "Duration": 93,
                "ExitNumber": [],
                "GeometryOffset": 7470,
                "RoundaboutExitStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right"
                },
                "Type": "RoundaboutExit"
            },
            {
                "Distance": 232,
                "Duration": 10,
                "ExitNumber": [],
                "GeometryOffset": 7563,
                "Type": "Continue"
            },
            {
                "Distance": 642,
                "Duration": 29,
                "ExitNumber": [],
```

```
                "GeometryOffset": 7572,
                "Type": "Continue"
            },
            {
                "Distance": 357,
                "Duration": 25,
                "ExitNumber": [],
                "GeometryOffset": 7580,
                "Type": "Continue"
            },
            {
                "Distance": 63,
                "Duration": 10,
                "ExitNumber": [],
                "GeometryOffset": 7600,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
            {
                "Distance": 2626,
                "Duration": 411,
                "ExitNumber": [],
                "GeometryOffset": 7602,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
            {
                "Distance": 234,
                "Duration": 37,
                "ExitNumber": [],
                "GeometryOffset": 7627,
                "Type": "Continue"
            },
            {
                "Distance": 0,
                "Duration": 0,
```

```
                                "ExitNumber": [],
                                "GeometryOffset": 7632,
                                "Type": "Arrive"
                            }
                        ],
                        "TruckRoadTypes": [],
                        "Zones": []
                    }
                }
            ],
            "MajorRoadLabels": [
                {
                    "RouteNumber": {
                        "Language": "fr",
                        "Value": "A6"
                    }
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/routes?key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
  "Origin": [
    2.234491,
    48.815704
  ],
  "Destination": [
    5.11412,
    47.260723
  ],
  "TravelMode": "Car",
  "Tolls": {
    "AllTransponders": true,
    "AllVignettes": true
  }
}'
```

AWS CLI

```
aws geo-routes calculate-routes --key ${YourKey} \
--origin 13.055211 52.704802 \
--destination 13.551910 52.282705 \
--travel-mode "Car" \
--tolls '{"AllTransponders": true, "AllVignettes": true}'
```

如何创建具有自定义避让功能的路线

CalculateRoutes API 可帮助您找到出发地和目的地之间的最佳路线，以及司机休息的最佳机会。它支持汽车、卡车、行人和摩托车等出行模式。它还支持多达 25 个航点（中途停留），包括起点和目的地，但仅有一些限制。

潜在用例

- 使用自定义避让来计算路线：使用避让功能自定义路线，以实现更好的路线和通勤规划。

示例

CalculateRoutes 与 “避免” 一起使用 BoundingBox 可创建自定义路由。

Sample request

```
{
    "Origin": [
        -123.116655,
        49.281538
    ],
    "Destination": [
        -123.01791785749363,
        49.22782762759908
    ],
    "TravelMode": "Car",
    "Avoid": {
        "Areas": [
            {
                "Geometry": {
                    "BoundingBox": [
                        -123.078693,
                        49.238987,
```

```
                        -123.054638,
                        49.251694
                    ]
                }
            }
        ]
    }
}
```

Sample response

```
{
    "LegGeometryFormat": "FlexiblePolyline",
    "Notices": [],
    "Routes": [
        {
            "Legs": [
                {
                    "Geometry": {
                        "Polyline": "Redacted"
                    },
                    "TravelMode": "Car",
                    "Type": "Vehicle",
                    "VehicleLegDetails": {
                        "AfterTravelSteps": [],
                        "Arrival": {
                            "Place": {
                                "ChargingStation": false,
                                "OriginalPosition": [
                                    -123.0179179,
                                    49.2278276
                                ],
                                "Position": [
                                    -123.0229001,
                                    49.22883
                                ]
                            }
                        },
                        "Departure": {
                            "Place": {
                                "ChargingStation": false,
                                "OriginalPosition": [
                                    -123.116655,
```

```
                        49.281538
                    ],
                    "Position": [
                        -123.1166332,
                        49.2815528
                    ]
                }
            },
            "Incidents": [],
            "Notices": [],
            "PassThroughWaypoints": [],
            "Spans": [],
            "Tolls": [],
            "TollSystems": [],
            "TravelSteps": [
                {
                    "Distance": 1144,
                    "Duration": 85,
                    "ExitNumber": [],
                    "GeometryOffset": 0,
                    "Type": "Depart"
                },
                {
                    "Distance": 262,
                    "Duration": 24,
                    "ExitNumber": [],
                    "GeometryOffset": 35,
                    "RampStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Right"
                    },
                    "Type": "Ramp"
                },
                {
                    "Distance": 433,
                    "Duration": 47,
                    "ExitNumber": [],
                    "GeometryOffset": 40,
                    "TurnStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Right",
                        "TurnIntensity": "Typical"
                    },
                    "Type": "Turn"
```

```
                    },
                    {
                        "Distance": 1131,
                        "Duration": 81,
                        "ExitNumber": [],
                        "GeometryOffset": 54,
                        "TurnStepDetails": {
                            "Intersection": [],
                            "SteeringDirection": "Left",
                            "TurnIntensity": "Typical"
                        },
                        "Type": "Turn"
                    },
                    {
                        "Distance": 54,
                        "Duration": 4,
                        "ExitNumber": [],
                        "GeometryOffset": 79,
                        "Type": "Continue"
                    },
                    {
                        "Distance": 520,
                        "Duration": 39,
                        "ExitNumber": [],
                        "GeometryOffset": 83,
                        "Type": "Continue"
                    },
                    {
                        "Distance": 1123,
                        "Duration": 111,
                        "ExitNumber": [],
                        "GeometryOffset": 101,
                        "TurnStepDetails": {
                            "Intersection": [],
                            "SteeringDirection": "Right",
                            "TurnIntensity": "Typical"
                        },
                        "Type": "Turn"
                    },
                    {
                        "Distance": 1131,
                        "Duration": 81,
                        "ExitNumber": [],
                        "GeometryOffset": 141,
```

```
                        "TurnStepDetails": {
                            "Intersection": [],
                            "SteeringDirection": "Left",
                            "TurnIntensity": "Typical"
                        },
                        "Type": "Turn"
                    },
                    {
                        "Distance": 2843,
                        "Duration": 209,
                        "ExitNumber": [],
                        "GeometryOffset": 167,
                        "Type": "Continue"
                    },
                    {
                        "Distance": 3219,
                        "Duration": 246,
                        "ExitNumber": [],
                        "GeometryOffset": 255,
                        "TurnStepDetails": {
                            "Intersection": [],
                            "SteeringDirection": "Right",
                            "TurnIntensity": "Typical"
                        },
                        "Type": "Turn"
                    },
                    {
                        "Distance": 64,
                        "Duration": 23,
                        "ExitNumber": [],
                        "GeometryOffset": 332,
                        "TurnStepDetails": {
                            "Intersection": [],
                            "SteeringDirection": "Left",
                            "TurnIntensity": "Typical"
                        },
                        "Type": "Turn"
                    },
                    {
                        "Distance": 0,
                        "Duration": 0,
                        "ExitNumber": [],
                        "GeometryOffset": 334,
                        "Type": "Arrive"
```

```
                            }
                        ],
                        "TruckRoadTypes": [],
                        "Zones": []
                    }
                }
            ],
            "MajorRoadLabels": [
                {
                    "RoadName": {
                        "Language": "en",
                        "Value": "Boundary Rd"
                    }
                },
                {
                    "RoadName": {
                        "Language": "en",
                        "Value": "Grandview Hwy"
                    }
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/routes?key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
  "Origin": [
    -123.116655,
    49.281538
  ],
  "Destination": [
    -123.01791785749363,
    49.22782762759908
  ],
  "TravelMode": "Car",
  "Avoid": {
    "Areas": [
      {
```

```
        "Geometry": {
          "BoundingBox": [
            -123.078693,
            49.238987,
            -123.054638,
            49.251694
          ]
        }
      }
    ]
  }
}'
```

AWS CLI

```
aws geo-routes calculate-routes --key ${YourKey} \
--origin -123.13277 49.281538 \
--destination -123.01791785749363 49.22782762759908 \
--travel-mode "Car" \
--avoid '{"Areas":[{"Geometry":{"BoundingBox":
[-123.054638,49.238987,-123.054638,49.251694]}}]}'
```

CalculateRoutes 与避开多边形一起使用可创建自定义路径。

Sample request

```
{
    "Origin": [
        -123.116655,
        49.281538
    ],
    "Destination": [
        -123.01791785749363,
        49.22782762759908
    ],
    "TravelMode": "Car",
    "Avoid": {
        "Areas": [
            {
                "Geometry": {
                    "polygon": [
                        [
```

```
                            [
                                -123.06953,
                                49.256419
                            ],
                            [
                                -123.080486,
                                49.242115
                            ],
                            [
                                -123.058573,
                                49.242115
                            ],
                            [
                                -123.06953,
                                49.256419
                            ]
                        ]
                    ]
                }
            }
        ]
    }
}
```

Sample response

```
{
    "LegGeometryFormat": "FlexiblePolyline",
    "Notices": [],
    "Routes": [
        {
            "Legs": [
                {
                    "Geometry": {
                        "Polyline": "Redacted"
                    },
                    "TravelMode": "Car",
                    "Type": "Vehicle",
                    "VehicleLegDetails": {
                        "AfterTravelSteps": [],
                        "Arrival": {
                            "Place": {
                                "ChargingStation": false,
```

```
                "OriginalPosition": [
                    -123.0179179,
                    49.2278276
                ],
                "Position": [
                    -123.0229001,
                    49.22883
                ]
            }
        },
        "Departure": {
            "Place": {
                "ChargingStation": false,
                "OriginalPosition": [
                    -123.116655,
                    49.281538
                ],
                "Position": [
                    -123.1166332,
                    49.2815528
                ]
            }
        },
        "Incidents": [],
        "Notices": [],
        "PassThroughWaypoints": [],
        "Spans": [],
        "Tolls": [],
        "TollSystems": [],
        "TravelSteps": [
            {
                "Distance": 1144,
                "Duration": 85,
                "ExitNumber": [],
                "GeometryOffset": 0,
                "Type": "Depart"
            },
            {
                "Distance": 262,
                "Duration": 24,
                "ExitNumber": [],
                "GeometryOffset": 35,
                "RampStepDetails": {
                    "Intersection": [],
```

```
                "SteeringDirection": "Right"
            },
            "Type": "Ramp"
        },
        {
            "Distance": 433,
            "Duration": 47,
            "ExitNumber": [],
            "GeometryOffset": 40,
            "TurnStepDetails": {
                "Intersection": [],
                "SteeringDirection": "Right",
                "TurnIntensity": "Typical"
            },
            "Type": "Turn"
        },
        {
            "Distance": 1131,
            "Duration": 81,
            "ExitNumber": [],
            "GeometryOffset": 54,
            "TurnStepDetails": {
                "Intersection": [],
                "SteeringDirection": "Left",
                "TurnIntensity": "Typical"
            },
            "Type": "Turn"
        },
        {
            "Distance": 54,
            "Duration": 4,
            "ExitNumber": [],
            "GeometryOffset": 79,
            "Type": "Continue"
        },
        {
            "Distance": 520,
            "Duration": 39,
            "ExitNumber": [],
            "GeometryOffset": 83,
            "Type": "Continue"
        },
        {
            "Distance": 1123,
```

```
                "Duration": 111,
                "ExitNumber": [],
                "GeometryOffset": 101,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
            {
                "Distance": 1131,
                "Duration": 81,
                "ExitNumber": [],
                "GeometryOffset": 141,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Left",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
            {
                "Distance": 448,
                "Duration": 34,
                "ExitNumber": [],
                "GeometryOffset": 167,
                "Type": "Continue"
            },
            {
                "Distance": 1797,
                "Duration": 151,
                "ExitNumber": [],
                "GeometryOffset": 182,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
            {
                "Distance": 2836,
                "Duration": 224,
```

```
                        "ExitNumber": [],
                        "GeometryOffset": 222,
                        "TurnStepDetails": {
                            "Intersection": [],
                            "SteeringDirection": "Left",
                            "TurnIntensity": "Typical"
                        },
                        "Type": "Turn"
                    },
                    {
                        "Distance": 339,
                        "Duration": 30,
                        "ExitNumber": [],
                        "GeometryOffset": 326,
                        "TurnStepDetails": {
                            "Intersection": [],
                            "SteeringDirection": "Right",
                            "TurnIntensity": "Typical"
                        },
                        "Type": "Turn"
                    },
                    {
                        "Distance": 64,
                        "Duration": 23,
                        "ExitNumber": [],
                        "GeometryOffset": 335,
                        "TurnStepDetails": {
                            "Intersection": [],
                            "SteeringDirection": "Left",
                            "TurnIntensity": "Typical"
                        },
                        "Type": "Turn"
                    },
                    {
                        "Distance": 0,
                        "Duration": 0,
                        "ExitNumber": [],
                        "GeometryOffset": 337,
                        "Type": "Arrive"
                    }
                ],
                "TruckRoadTypes": [],
                "Zones": []
            }
```

```
                }
            ],
            "MajorRoadLabels": [
                {
                    "RouteNumber": {
                        "Language": "en",
                        "Value": "HWY-1A"
                    }
                },
                {
                    "RoadName": {
                        "Language": "en",
                        "Value": "Nanaimo St"
                    }
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/routes?key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
  "Origin": [
    -123.116655,
    49.281538
  ],
  "Destination": [
    -123.01791785749363,
    49.22782762759908
  ],
  "TravelMode": "Car",
  "Avoid": {
    "Areas": [
      {
        "Geometry": {
          "polygon": [
            [
              [
                -123.06953,
```

```
                49.256419
              ],
              [
                -123.080486,
                49.242115
              ],
              [
                -123.058573,
                49.242115
              ],
              [
                -123.06953,
                49.256419
              ]
            ]
          ]
        }
      }
    ]
  }
}'
```

AWS CLI

```
aws geo-routes calculate-routes --key ${YourKey} \
--origin -123.13277 49.281538 \
--destination -123.01791785749363 49.22782762759908 \
--travel-mode "Car" \
--avoid '{"Areas":[{"Geometry":{"Polygon":[[[-123.06953,49.256419],
[-123.05167,49.242115],[-123.02381,49.242115],[-123.06953,49.256419]]]}}]}'
```

CalculateRoutes 与一起使用 PolylinePolygon 可创建自定义路由。

Sample request

```
{
    "Origin": [
        -123.116655,
        49.281538
    ],
    "Destination": [
        -123.01791785749363,
```

```
        49.22782762759908
    ],
    "TravelMode": "Car",
    "Avoid": {
        "Areas": [
            {
                "Geometry": {
                    "PolylinePolygon": [
                        "BF0s0sJxglvXr5CvkCAgpEs5CvkC"
                    ]
                }
            }
        ]
    }
}
```

Sample response

```
{
    "LegGeometryFormat": "FlexiblePolyline",
    "Notices": [],
    "Routes": [
        {
            "Legs": [
                {
                    "Geometry": {
                        "Polyline": "Redacted"
                    },
                    "TravelMode": "Car",
                    "Type": "Vehicle",
                    "VehicleLegDetails": {
                        "AfterTravelSteps": [],
                        "Arrival": {
                            "Place": {
                                "ChargingStation": false,
                                "OriginalPosition": [
                                    -123.0179179,
                                    49.2278276
                                ],
                                "Position": [
                                    -123.0229001,
                                    49.22883
                                ]
```

```
                    }
                },
                "Departure": {
                    "Place": {
                        "ChargingStation": false,
                        "OriginalPosition": [
                            -123.116655,
                            49.281538
                        ],
                        "Position": [
                            -123.1166332,
                            49.2815528
                        ]
                    }
                },
                "Incidents": [],
                "Notices": [],
                "PassThroughWaypoints": [],
                "Spans": [],
                "Tolls": [],
                "TollSystems": [],
                "TravelSteps": [
                    {
                        "Distance": 1144,
                        "Duration": 85,
                        "ExitNumber": [],
                        "GeometryOffset": 0,
                        "Type": "Depart"
                    },
                    {
                        "Distance": 262,
                        "Duration": 24,
                        "ExitNumber": [],
                        "GeometryOffset": 35,
                        "RampStepDetails": {
                            "Intersection": [],
                            "SteeringDirection": "Right"
                        },
                        "Type": "Ramp"
                    },
                    {
                        "Distance": 433,
                        "Duration": 47,
                        "ExitNumber": [],
```

```
                "GeometryOffset": 40,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
            {
                "Distance": 1131,
                "Duration": 81,
                "ExitNumber": [],
                "GeometryOffset": 54,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Left",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
            {
                "Distance": 54,
                "Duration": 4,
                "ExitNumber": [],
                "GeometryOffset": 79,
                "Type": "Continue"
            },
            {
                "Distance": 520,
                "Duration": 39,
                "ExitNumber": [],
                "GeometryOffset": 83,
                "Type": "Continue"
            },
            {
                "Distance": 1123,
                "Duration": 111,
                "ExitNumber": [],
                "GeometryOffset": 101,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right",
                    "TurnIntensity": "Typical"
                },
```

```
                "Type": "Turn"
            },
            {
                "Distance": 1131,
                "Duration": 81,
                "ExitNumber": [],
                "GeometryOffset": 141,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Left",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
            {
                "Distance": 448,
                "Duration": 34,
                "ExitNumber": [],
                "GeometryOffset": 167,
                "Type": "Continue"
            },
            {
                "Distance": 1797,
                "Duration": 151,
                "ExitNumber": [],
                "GeometryOffset": 182,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
            {
                "Distance": 2836,
                "Duration": 224,
                "ExitNumber": [],
                "GeometryOffset": 222,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Left",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
```

```
                        },
                        {
                            "Distance": 339,
                            "Duration": 30,
                            "ExitNumber": [],
                            "GeometryOffset": 326,
                            "TurnStepDetails": {
                                "Intersection": [],
                                "SteeringDirection": "Right",
                                "TurnIntensity": "Typical"
                            },
                            "Type": "Turn"
                        },
                        {
                            "Distance": 64,
                            "Duration": 23,
                            "ExitNumber": [],
                            "GeometryOffset": 335,
                            "TurnStepDetails": {
                                "Intersection": [],
                                "SteeringDirection": "Left",
                                "TurnIntensity": "Typical"
                            },
                            "Type": "Turn"
                        },
                        {
                            "Distance": 0,
                            "Duration": 0,
                            "ExitNumber": [],
                            "GeometryOffset": 337,
                            "Type": "Arrive"
                        }
                    ],
                    "TruckRoadTypes": [],
                    "Zones": []
                }
            }
        ],
        "MajorRoadLabels": [
            {
                "RouteNumber": {
                    "Language": "en",
                    "Value": "HWY-1A"
                }
```

```
                },
                {
                    "RoadName": {
                        "Language": "en",
                        "Value": "Nanaimo St"
                    }
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/routes?key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
  "Origin": [
    -123.116655,
    49.281538
  ],
  "Destination": [
    -123.01791785749363,
    49.22782762759908
  ],
  "TravelMode": "Car",
  "Avoid": {
    "Areas": [
      {
        "Geometry": {
          "PolylinePolygon": [
            "BF0s0sJxglvXr5CvkCAgpEs5CvkC"
          ]
        }
      }
    ]
  }
}'
```

AWS CLI

```
aws geo-routes calculate-routes --key ${YourKey} \
```

```
--origin -123.13277 49.281538 \
--destination -123.01791785749363 49.22782762759908 \
--travel-mode "Car" \
--avoid '{"Areas":[{"Geometry":{"PolylinePolygon":
["BF0s0sJxglvXr5CvkCAgpEs5CvkC"]}}]}'
```

CalculateRoutes 与避开走廊一起使用可创建自定义路线。

Sample request

```
{
    "Origin": [
        -123.116655,
        49.281538
    ],
    "Destination": [
        -123.01791785749363,
        49.22782762759908
    ],
    "TravelMode": "Car",
    "Avoid": {
        "Areas": [
            {
                "Geometry": {
                    "PolylineCorridor": {
                        "Polyline": "Redacted",
                        "Radius": 10
                    }
                }
            }
        ]
    }
}
```

Sample response

```
{
    "LegGeometryFormat": "FlexiblePolyline",
    "Notices": [],
    "Routes": [
        {
            "Legs": [
```

```
                {
                    "Geometry": {
                        "Polyline": "Redacted"
                    },
                    "TravelMode": "Car",
                    "Type": "Vehicle",
                    "VehicleLegDetails": {
                        "AfterTravelSteps": [],
                        "Arrival": {
                            "Place": {
                                "ChargingStation": false,
                                "OriginalPosition": [
                                    -123.0179179,
                                    49.2278276
                                ],
                                "Position": [
                                    -123.0229001,
                                    49.22883
                                ]
                            }
                        },
                        "Departure": {
                            "Place": {
                                "ChargingStation": false,
                                "OriginalPosition": [
                                    -123.116655,
                                    49.281538
                                ],
                                "Position": [
                                    -123.1166332,
                                    49.2815528
                                ]
                            }
                        },
                        "Incidents": [],
                        "Notices": [],
                        "PassThroughWaypoints": [],
                        "Spans": [],
                        "Tolls": [],
                        "TollSystems": [],
                        "TravelSteps": [
                            {
                                "Distance": 1144,
                                "Duration": 85,
```

```
                "ExitNumber": [],
                "GeometryOffset": 0,
                "Type": "Depart"
            },
            {
                "Distance": 262,
                "Duration": 24,
                "ExitNumber": [],
                "GeometryOffset": 35,
                "RampStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right"
                },
                "Type": "Ramp"
            },
            {
                "Distance": 1356,
                "Duration": 134,
                "ExitNumber": [],
                "GeometryOffset": 40,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
            {
                "Distance": 2468,
                "Duration": 204,
                "ExitNumber": [],
                "GeometryOffset": 90,
                "KeepStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Left"
                },
                "Type": "Keep"
            },
            {
                "Distance": 1100,
                "Duration": 88,
                "ExitNumber": [],
                "GeometryOffset": 170,
                "TurnStepDetails": {
```

```
                        "Intersection": [],
                        "SteeringDirection": "Right",
                        "TurnIntensity": "Typical"
                    },
                    "Type": "Turn"
                },
                {
                    "Distance": 1588,
                    "Duration": 133,
                    "ExitNumber": [],
                    "GeometryOffset": 214,
                    "TurnStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Left",
                        "TurnIntensity": "Typical"
                    },
                    "Type": "Turn"
                },
                {
                    "Distance": 47,
                    "Duration": 14,
                    "ExitNumber": [],
                    "GeometryOffset": 260,
                    "Type": "Continue"
                },
                {
                    "Distance": 2478,
                    "Duration": 196,
                    "ExitNumber": [],
                    "GeometryOffset": 265,
                    "TurnStepDetails": {
                        "Intersection": [],
                        "SteeringDirection": "Right",
                        "TurnIntensity": "Typical"
                    },
                    "Type": "Turn"
                },
                {
                    "Distance": 339,
                    "Duration": 30,
                    "ExitNumber": [],
                    "GeometryOffset": 357,
                    "TurnStepDetails": {
                        "Intersection": [],
```

```
                                    "SteeringDirection": "Right",
                                    "TurnIntensity": "Typical"
                                },
                                "Type": "Turn"
                            },
                            {
                                "Distance": 64,
                                "Duration": 23,
                                "ExitNumber": [],
                                "GeometryOffset": 366,
                                "TurnStepDetails": {
                                    "Intersection": [],
                                    "SteeringDirection": "Left",
                                    "TurnIntensity": "Typical"
                                },
                                "Type": "Turn"
                            },
                            {
                                "Distance": 0,
                                "Duration": 0,
                                "ExitNumber": [],
                                "GeometryOffset": 368,
                                "Type": "Arrive"
                            }
                        ],
                        "TruckRoadTypes": [],
                        "Zones": []
                    }
                }
            ],
            "MajorRoadLabels": [
                {
                    "RouteNumber": {
                        "Language": "en",
                        "Value": "HWY-1A"
                    }
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/routes?key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
  "Origin": [
    -123.116655,
    49.281538
  ],
  "Destination": [
    -123.01791785749363,
    49.22782762759908
  ],
  "TravelMode": "Car",
  "Avoid": {
    "Areas": [
      {
        "Geometry": {
          "PolylineCorridor": {
            "Polyline": "Redacted",
            "Radius": 10
          }
        }
      }
    ]
  }
}'
```

AWS CLI

```
aws geo-routes calculate-routes --key ${YourKey} \
--origin -123.116655 49.281538 \
--destination -123.01791785749363 49.22782762759908 \
--travel-mode "Car" \
--avoid '{"Areas": [{"Geometry": {"PolylineCorridor": {"Polyline":
 "BF2mysJnmkvX5ekiC", "Radius": 10}}}]}'
```

CalculateRoutes 与 “避免” 一起使用 PolylineCorridor 可创建自定义路由。

Sample request

```
{
    "Origin": [
        -123.116655,
        49.281538
    ],
    "Destination": [
        -123.01791785749363,
        49.22782762759908
    ],
    "TravelMode": "Car",
    "Avoid": {
        "Areas": [
            {
                "Geometry": {
                    "Corridor": {
                        "LineString": [
                            [
                                -123.06532243038754,
                                49.245226301868776
                            ],
                            [
                                -123.0547357660333,
                                49.24030469850804
                            ]
                        ],
                        "Radius": 10
                    }
                }
            }
        ]
    }
}
```

Sample response

```
{
    "LegGeometryFormat": "FlexiblePolyline",
    "Notices": [],
    "Routes": [
```

```
        {
            "Legs": [
                {
                    "Geometry": {
                        "Polyline": "Redacted"
                    },
                    "TravelMode": "Car",
                    "Type": "Vehicle",
                    "VehicleLegDetails": {
                        "AfterTravelSteps": [],
                        "Arrival": {
                            "Place": {
                                "ChargingStation": false,
                                "OriginalPosition": [
                                    -123.0179179,
                                    49.2278276
                                ],
                                "Position": [
                                    -123.0229001,
                                    49.22883
                                ]
                            }
                        },
                        "Departure": {
                            "Place": {
                                "ChargingStation": false,
                                "OriginalPosition": [
                                    -123.116655,
                                    49.281538
                                ],
                                "Position": [
                                    -123.1166332,
                                    49.2815528
                                ]
                            }
                        },
                        "Incidents": [],
                        "Notices": [],
                        "PassThroughWaypoints": [],
                        "Spans": [],
                        "Tolls": [],
                        "TollSystems": [],
                        "TravelSteps": [
                            {
```

```
                "Distance": 1144,
                "Duration": 85,
                "ExitNumber": [],
                "GeometryOffset": 0,
                "Type": "Depart"
            },
            {
                "Distance": 262,
                "Duration": 24,
                "ExitNumber": [],
                "GeometryOffset": 35,
                "RampStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right"
                },
                "Type": "Ramp"
            },
            {
                "Distance": 1356,
                "Duration": 134,
                "ExitNumber": [],
                "GeometryOffset": 40,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
            {
                "Distance": 2468,
                "Duration": 204,
                "ExitNumber": [],
                "GeometryOffset": 90,
                "KeepStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Left"
                },
                "Type": "Keep"
            },
            {
                "Distance": 1100,
                "Duration": 88,
                "ExitNumber": [],
```

```
                "GeometryOffset": 170,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
            {
                "Distance": 1588,
                "Duration": 133,
                "ExitNumber": [],
                "GeometryOffset": 214,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Left",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
            {
                "Distance": 47,
                "Duration": 14,
                "ExitNumber": [],
                "GeometryOffset": 260,
                "Type": "Continue"
            },
            {
                "Distance": 2478,
                "Duration": 196,
                "ExitNumber": [],
                "GeometryOffset": 265,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
            {
                "Distance": 339,
                "Duration": 30,
                "ExitNumber": [],
                "GeometryOffset": 357,
```

```
                                "TurnStepDetails": {
                                    "Intersection": [],
                                    "SteeringDirection": "Right",
                                    "TurnIntensity": "Typical"
                                },
                                "Type": "Turn"
                            },
                            {
                                "Distance": 64,
                                "Duration": 23,
                                "ExitNumber": [],
                                "GeometryOffset": 366,
                                "TurnStepDetails": {
                                    "Intersection": [],
                                    "SteeringDirection": "Left",
                                    "TurnIntensity": "Typical"
                                },
                                "Type": "Turn"
                            },
                            {
                                "Distance": 0,
                                "Duration": 0,
                                "ExitNumber": [],
                                "GeometryOffset": 368,
                                "Type": "Arrive"
                            }
                        ],
                        "TruckRoadTypes": [],
                        "Zones": []
                    }
                }
            ],
            "MajorRoadLabels": [
                {
                    "RouteNumber": {
                        "Language": "en",
                        "Value": "HWY-1A"
                    }
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/routes?key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
  "Origin": [
    -123.116655,
    49.281538
  ],
  "Destination": [
    -123.01791785749363,
    49.22782762759908
  ],
  "TravelMode": "Car",
  "Avoid": {
    "Areas": [
      {
        "Geometry": {
          "Corridor": {
            "LineString": [
              [
                -123.06532243038754,
                49.245226301868776
              ],
              [
                -123.0547357660333,
                49.24030469850804
              ]
            ],
            "Radius": 10
          }
        }
      }
    ]
  }
}'
```

AWS CLI

```
aws geo-routes calculate-routes --key ${YourKey} \
--origin -123.13277 49.281538 \
--destination -123.01791785749363 49.22782762759908 \
```

```
--travel-mode "Car" \
--avoid '{"Areas":[{"Geometry":{"Corridor":{"LineString":
[[-123.06532243038754,49.245226301868776],
[-123.0547357660333,49.24030469850804]],"Radius":10}}}]}'
```

如何通过自定义避开多个潜在物品来创建路线

CalculateRoutes API 可帮助您找到出发地和目的地之间的最佳路线，以及司机休息的最佳机会。它支持汽车、卡车、行人和摩托车等出行模式。它还支持多达 25 个航点（中途停留），包括起点和目的地，但仅有一些限制。

潜在用例

- 使用自定义避让来计算路线：使用避让功能自定义路线，以实现更好的路线和通勤规划。

示例

使用 CalculateRoutes 时可避开通行费、掉头、渡轮、高速公路、隧道等。

Sample request

```
{
    "Origin": [
        -123.118105,
        49.282423
    ],
    "Destination": [
        -123.020098,
        49.232872
    ],
    "TravelMode": "Car",
    "Avoid": {
        "TollRoads": true,
        "UTurns": true,
        "Ferries": true,
        "ControlledAccessHighways": true,
        "Tunnels": true
    }
}
```

Sample response


```
{
    "LegGeometryFormat": "FlexiblePolyline",
    "Notices": [],
    "Routes": [
        {
            "Legs": [
                {
                    "Geometry": {
                        "Polyline": "Redacted"
                    },
                    "TravelMode": "Car",
                    "Type": "Vehicle",
                    "VehicleLegDetails": {
                        "AfterTravelSteps": [],
                        "Arrival": {
                            "Place": {
                                "ChargingStation": false,
                                "OriginalPosition": [
                                    -123.020098,
                                    49.232872
                                ],
                                "Position": [
                                    -123.0203051,
                                    49.2328499
                                ]
                            }
                        },
                        "Departure": {
                            "Place": {
                                "ChargingStation": false,
                                "OriginalPosition": [
                                    -123.1181051,
                                    49.282423
                                ],
                                "Position": [
                                    -123.1180883,
                                    49.2824349
                                ]
                            }
                        },
                        "Incidents": [],
                        "Notices": [],
```

```
                        "PassThroughWaypoints": [],
                        "Spans": [],
                        "Tolls": [],
                        "TollSystems": [],
                        "TravelSteps": [
                            {
                                "Distance": 1288,
                                "Duration": 102,
                                "ExitNumber": [],
                                "GeometryOffset": 0,
                                "Type": "Depart"
                            },
                            {
                                "Distance": 262,
                                "Duration": 24,
                                "ExitNumber": [],
                                "GeometryOffset": 37,
                                "RampStepDetails": {
                                    "Intersection": [],
                                    "SteeringDirection": "Right"
                                },
                                "Type": "Ramp"
                            },
                            {
                                "Distance": 1356,
                                "Duration": 134,
                                "ExitNumber": [],
                                "GeometryOffset": 42,
                                "TurnStepDetails": {
                                    "Intersection": [],
                                    "SteeringDirection": "Right",
                                    "TurnIntensity": "Typical"
                                },
                                "Type": "Turn"
                            },
                            {
                                "Distance": 7092,
                                "Duration": 568,
                                "ExitNumber": [],
                                "GeometryOffset": 92,
                                "KeepStepDetails": {
                                    "Intersection": [],
                                    "SteeringDirection": "Left"
                                },
```

```
                            "Type": "Keep"
                        },
                        {
                            "Distance": 65,
                            "Duration": 26,
                            "ExitNumber": [],
                            "GeometryOffset": 337,
                            "TurnStepDetails": {
                                "Intersection": [],
                                "SteeringDirection": "Left",
                                "TurnIntensity": "Typical"
                            },
                            "Type": "Turn"
                        },
                        {
                            "Distance": 50,
                            "Duration": 18,
                            "ExitNumber": [],
                            "GeometryOffset": 339,
                            "TurnStepDetails": {
                                "Intersection": [],
                                "SteeringDirection": "Right",
                                "TurnIntensity": "Typical"
                            },
                            "Type": "Turn"
                        },
                        {
                            "Distance": 0,
                            "Duration": 0,
                            "ExitNumber": [],
                            "GeometryOffset": 343,
                            "Type": "Arrive"
                        }
                    ],
                    "TruckRoadTypes": [],
                    "Zones": []
                }
            }
        ],
        "MajorRoadLabels": [
            {
                "RouteNumber": {
                    "Language": "en",
                    "Value": "HWY-1A"
```

```
                    }
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/routes?key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
  "Origin": [
    -123.118105,
    49.282423
  ],
  "Destination": [
    -123.020098,
    49.232872
  ],
  "TravelMode": "Car",
  "Avoid": {
    "TollRoads": true,
    "UTurns": true,
    "Ferries": true,
    "ControlledAccessHighways": true,
    "Tunnels": true
  }
}'
```

AWS CLI

```
aws geo-routes calculate-routes --key ${YourKey} \
--origin -123.118105 49.282423 \
--destination -123.020098 49.232872 \
--travel-mode "Car" \
--avoid '{"TollRoads": true, "UTurns": true, "Ferries": true,
 "ControlledAccessHighways": true, "Tunnels": true}'
```

如何找到最快的路线

CalculateRoutes API 可帮助您找到起点和目的地之间的最快路线。

潜在用例

- 优化路线，提高时间效率：通过计算最快的路线，改善配送操作。这对于减少物流和配送服务的旅行时间非常有用。

示例

计算最快的路线

Sample request

```
{
    "Origin": [
        -123.118105,
        49.282423
    ],
    "Destination": [
        -123.020098,
        49.232872
    ],
    "TravelMode": "Car",
    "OptimizeRoutingFor": "FastestRoute"
}
```

Sample response

```
{
    "LegGeometryFormat": "FlexiblePolyline",
    "Notices": [],
    "Routes": [
        {
            "Legs": [
                {
                    "Geometry": {
                        "Polyline": "Redacted"
                    },
                    "TravelMode": "Car",
                    "Type": "Vehicle",
```

```
                    "VehicleLegDetails": {
                        "AfterTravelSteps": [],
                        "Arrival": {
                            "Place": {
                                "ChargingStation": false,
                                "OriginalPosition": [
                                    -123.020098,
                                    49.232872
                                ],
                                "Position": [
                                    -123.0203051,
                                    49.2328499
                                ]
                            }
                        },
                        "Departure": {
                            "Place": {
                                "ChargingStation": false,
                                "OriginalPosition": [
                                    -123.1181051,
                                    49.282423
                                ],
                                "Position": [
                                    -123.1180883,
                                    49.2824349
                                ]
                            }
                        },
                        "Incidents": [],
                        "Notices": [],
                        "PassThroughWaypoints": [],
                        "Spans": [],
                        "Tolls": [],
                        "TollSystems": [],
                        "TravelSteps": [
                            {
                                "Distance": 1288,
                                "Duration": 102,
                                "ExitNumber": [],
                                "GeometryOffset": 0,
                                "Type": "Depart"
                            },
                            {
                                "Distance": 262,
```

```
                "Duration": 24,
                "ExitNumber": [],
                "GeometryOffset": 37,
                "RampStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right"
                },
                "Type": "Ramp"
            },
            {
                "Distance": 1356,
                "Duration": 134,
                "ExitNumber": [],
                "GeometryOffset": 42,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
            {
                "Distance": 7092,
                "Duration": 568,
                "ExitNumber": [],
                "GeometryOffset": 92,
                "KeepStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Left"
                },
                "Type": "Keep"
            },
            {
                "Distance": 65,
                "Duration": 26,
                "ExitNumber": [],
                "GeometryOffset": 337,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Left",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
```

```
                            {
                                "Distance": 50,
                                "Duration": 18,
                                "ExitNumber": [],
                                "GeometryOffset": 339,
                                "TurnStepDetails": {
                                    "Intersection": [],
                                    "SteeringDirection": "Right",
                                    "TurnIntensity": "Typical"
                                },
                                "Type": "Turn"
                            },
                            {
                                "Distance": 0,
                                "Duration": 0,
                                "ExitNumber": [],
                                "GeometryOffset": 343,
                                "Type": "Arrive"
                            }
                        ],
                        "TruckRoadTypes": [],
                        "Zones": []
                    }
                }
            ],
            "MajorRoadLabels": [
                {
                    "RouteNumber": {
                        "Language": "en",
                        "Value": "HWY-1A"
                    }
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/routes?key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
```

```
  "Origin": [
    -123.118105,
    49.282423
  ],
  "Destination": [
    -123.020098,
    49.232872
  ],
  "TravelMode": "Car",
  "OptimizeRoutingFor": "FastestRoute"
}'
```

AWS CLI

```
aws geo-routes calculate-routes --key ${YourKey} \
--origin -123.118105 49.282423 \
--destination -123.020098 49.232872 \
--travel-mode "Car" \
--optimize-routing-for "FastestRoute"
```

如何找到最短的路线

CalculateRoutes API 可帮助您找到起点和目的地之间的最短路线。

潜在用例

- 优化路线，提高时间效率：通过计算最短路线，改善配送操作。这对于缩短物流和配送服务中的旅行距离非常有用。

示例

用于 CalculateRoutes 查找最短路径并节省行驶距离。

Sample request

```
{
    "Origin": [
        -123.118105,
        49.282423
    ],
    "Destination": [
```

```
        -123.020098,
        49.232872
    ],
    "TravelMode": "Car",
    "OptimizeRoutingFor": "ShortestRoute"
}
```

Sample response

```
{
    "LegGeometryFormat": "FlexiblePolyline",
    "Notices": [],
    "Routes": [
        {
            "Legs": [
                {
                    "Geometry": {
                        "Polyline": "Redacted"
                    },
                    "TravelMode": "Car",
                    "Type": "Vehicle",
                    "VehicleLegDetails": {
                        "AfterTravelSteps": [],
                        "Arrival": {
                            "Place": {
                                "ChargingStation": false,
                                "OriginalPosition": [
                                    -123.020098,
                                    49.232872
                                ],
                                "Position": [
                                    -123.0203051,
                                    49.2328499
                                ]
                            }
                        },
                        "Departure": {
                            "Place": {
                                "ChargingStation": false,
                                "OriginalPosition": [
                                    -123.1181051,
                                    49.282423
                                ],
```

```
                "Position": [
                    -123.1180883,
                    49.2824349
                ]
            }
        },
        "Incidents": [],
        "Notices": [],
        "PassThroughWaypoints": [],
        "Spans": [],
        "Tolls": [],
        "TollSystems": [],
        "TravelSteps": [
            {
                "Distance": 1288,
                "Duration": 102,
                "ExitNumber": [],
                "GeometryOffset": 0,
                "Type": "Depart"
            },
            {
                "Distance": 262,
                "Duration": 24,
                "ExitNumber": [],
                "GeometryOffset": 37,
                "RampStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right"
                },
                "Type": "Ramp"
            },
            {
                "Distance": 1356,
                "Duration": 134,
                "ExitNumber": [],
                "GeometryOffset": 42,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
            {
```

```
                "Distance": 7092,
                "Duration": 568,
                "ExitNumber": [],
                "GeometryOffset": 92,
                "KeepStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Left"
                },
                "Type": "Keep"
            },
            {
                "Distance": 65,
                "Duration": 26,
                "ExitNumber": [],
                "GeometryOffset": 337,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Left",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
            {
                "Distance": 50,
                "Duration": 18,
                "ExitNumber": [],
                "GeometryOffset": 339,
                "TurnStepDetails": {
                    "Intersection": [],
                    "SteeringDirection": "Right",
                    "TurnIntensity": "Typical"
                },
                "Type": "Turn"
            },
            {
                "Distance": 0,
                "Duration": 0,
                "ExitNumber": [],
                "GeometryOffset": 343,
                "Type": "Arrive"
            }
        ],
        "TruckRoadTypes": [],
        "Zones": []
```

```
                    }
                }
            ],
            "MajorRoadLabels": [
                {
                    "RouteNumber": {
                        "Language": "en",
                        "Value": "HWY-1A"
                    }
                }
            ]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/routes?key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
    "Origin": [
        -123.118105,
        49.282423
    ],
    "Destination": [
        -123.020098,
        49.232872
    ],
    "TravelMode": "Car",
    "OptimizeRoutingFor": "ShortestRoute"
}'
```

AWS CLI

```
aws geo-routes calculate-routes --key ${YourKey} \
--origin -123.118105 49.282423 \
--destination -123.020098 49.232872 \
--travel-mode "Scooter" \
--optimize-routing-for "ShortestRoute"
```

学习如何使用 CalculateIsolines

在本节中，您将学习如何CalculateIsolines使用时间或距离范围，以及如何设置要避开的区域。

主题

- 如何根据时间范围计算服务区
- 如何根据距离范围计算服务区
- 如何根据避让计算服务区

如何根据时间范围计算服务区

CalculateIsolines API 允许您根据道路网络和交通状况，确定指定时间或距离范围内可到达的服务区。此功能支持应用程序为餐厅、杂货店或其他服务提供商定义服务区域，这可以帮助规划燃油效率和定义服务覆盖范围的无障碍区域。

潜在用例

- 规划服务区域：使用此 API 根据旅行时间或距离为餐厅或杂货配送等服务规划无障碍区域。

示例

使用 Car 根据时间范围计算服务区 TravelMode

Sample request

```
{
    "Origin": [
        -123.11679620827039,
        49.28147612192166
    ],
    "DepartureTime": "2024-10-28T21:27:56Z",
    "Thresholds": {
        "Time": [
            500,
            1000,
            1500
        ]
    },
    "TravelMode": "Car"
}
```

Sample response


```
{
    "DepartureTime": "2024-10-28T14:27:56-07:00",
    "IsolineGeometryFormat": "FlexiblePolyline",
    "Isolines": [
        {
            "Connections": [],
            "Geometries": [
                {
                    "PolylinePolygon": [
                        "Redacted"
                    ]
                }
            ],
            "TimeThreshold": 500
        },
        {
            "Connections": [],
            "Geometries": [
                {
                    "PolylinePolygon": [
                        "Redacted"
                    ]
                }
            ],
            "TimeThreshold": 1000
        },
        {
            "Connections": [],
            "Geometries": [
                {
                    "PolylinePolygon": [
                        "Redacted"
                    ]
                }
            ],
            "TimeThreshold": 1500
        }
    ],
    "SnappedOrigin": [
        -123.11687,
        49.2813999
    ]
```

```
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/isolines?key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
    "Origin": [
        -123.11679620827039,
        49.28147612192166
    ],
    "DepartureTime": "2024-10-28T21:27:56Z",
    "Thresholds": {
        "Time": [
            500,
            1000,
            1500
        ]
    },
    "TravelMode": "Car"
}'
```

AWS CLI

```
aws geo-routes calculate-isolines --key ${YourKey} \
--origin -123.11679620827039 49.28147612192166 \
--departure-time "2024-10-28T21:27:56Z" \
--thresholds '{"Time": [500, 1000, 1500]}' \
--travel-mode "Car"
```

如何根据距离范围计算服务区

CalculateIsolines API 允许您根据道路网络和交通状况，确定指定时间或距离范围内可到达的服务区。此功能支持应用程序为餐厅、杂货店或其他服务提供商定义服务区域，这可以帮助规划燃油效率和定义服务覆盖范围的无障碍区域。

潜在用例

- 规划服务区域：使用此 API 根据旅行时间或距离为餐厅或杂货配送等服务规划无障碍区域。

示例

根据汽车的距离范围计算服务区 TravelMode

Sample request

```
{
    "Origin": [
        -123.11679620827039,
        49.28147612192166
    ],
    "DepartureTime": "2024-10-28T21:27:56Z",
    "Thresholds": {
        "Distance": [
            1000,
            2000,
            3000
        ]
    },
    "TravelMode": "Car"
}
```

Sample response

```
{
    "DepartureTime": "2024-10-28T14:27:56-07:00",
    "IsolineGeometryFormat": "FlexiblePolyline",
    "Isolines": [
        {
            "Connections": [],
            "DistanceThreshold": 1000,
            "Geometries": [
                {
                    "PolylinePolygon": [
                        "Redacted"
                    ]
                }
            ]
        },
        {
            "Connections": [],
            "DistanceThreshold": 2000,
            "Geometries": [
```

```
                {
                    "PolylinePolygon": [
                        "Redacted"
                    ]
                }
            ]
        },
        {
            "Connections": [],
            "DistanceThreshold": 3000,
            "Geometries": [
                {
                    "PolylinePolygon": [
                        "Redacted"
                    ]
                }
            ]
        }
    ],
    "SnappedOrigin": [
        -123.11687,
        49.2813999
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/isolines?key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
    "Origin": [
        -123.11679620827039,
        49.28147612192166
    ],
    "DepartureTime": "2024-10-28T21:27:56Z",
    "Thresholds": {
        "Distance": [
            1000,
            2000,
            3000
        ]
    },
```

```
    "TravelMode": "Car"
}'
```

AWS CLI

```
aws geo-routes calculate-isolines --key ${YourKey} \
--origin -123.11679620827039 49.28147612192166 \
--departure-time "2024-10-28T21:27:56Z" \
--thresholds '{"Distance": [1000, 2000, 3000]}' \
--travel-mode "Car"
```

如何根据避让计算服务区

该 CalculateIsolines API 允许您在指定的时间或距离范围内确定可到达的服务区，同时考虑要避开的道路网络和交通状况。此功能支持应用程序为餐厅、杂货店或其他服务提供商定义服务区域，这可以帮助规划燃油效率和定义服务覆盖范围的无障碍区域。

潜在用例

- 规划服务区域：使用此 API 根据避让情况为餐厅或杂货配送等服务规划无障碍区域。

示例

使用 Car 根据避让计算服务区 TravelMode

Sample request

```
{
    "Origin": [
        -123.11679620827039,
        49.28147612192166
    ],
    "DepartureTime": "2024-10-28T21:27:56Z",
    "Thresholds": {
        "Distance": [
            4000
        ]
    },
    "Avoid": {
        "TollRoads": true,
        "UTurns": true,
```

```
        "ControlledAccessHighways": true,
        "Ferries": true,
        "DirtRoads": true,
        "SeasonalClosure": true,
        "CarShuttleTrains": true,
        "TollTransponders": true,
        "ZoneCategories": [
            {
                "Category": "Environmental"
            }
        ]
    },
    "TravelMode": "Car"
}
```

Sample response

```
{
    "DepartureTime": "2024-10-28T14:27:56-07:00",
    "IsolineGeometryFormat": "FlexiblePolyline",
    "Isolines": [
        {
            "Connections": [],
            "DistanceThreshold": 4000,
            "Geometries": [
                {
                    "PolylinePolygon": [
                        "Redacted"
                    ]
                }
            ]
        }
    ],
    "SnappedOrigin": [
        -123.11687,
        49.2813999
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/isolines?key=Your_key' \
```

```
  --header 'Content-Type: application/json' \
  --data '{
    "Origin": [
        -123.11679620827039,
        49.28147612192166
    ],
    "DepartureTime": "2024-10-28T21:27:56Z",
    "Thresholds": {
        "Distance": [
            4000
        ]
    },
    "Avoid": {
        "TollRoads": true,
        "UTurns": true,
        "ControlledAccessHighways": true,
        "Ferries": true,
        "DirtRoads": true,
        "SeasonalClosure": true,
        "CarShuttleTrains": true,
        "TollTransponders": true,
        "ZoneCategories": [
            {
                "Category": "Environmental"
            }
        ]
    },
    "TravelMode": "Car"
}'
```

AWS CLI

```
aws geo-routes calculate-isolines --key ${YourKey} \
--origin -123.11679620827039 49.28147612192166 \
--departure-time "2024-10-28T21:27:56Z" \
--thresholds '{"Distance": [4000]}' \
--avoid '{"TollRoads": true, "UTurns": true, "ControlledAccessHighways": true,
 "Ferries": true, "DirtRoads": true, "SeasonalClosure": true, "CarShuttleTrains":
 true, "TollTransponders": true, "ZoneCategories": [{"Category": "Environmental"}]}'
 \
--travel-mode "Car"
```

使用卡车根据避让量计算服务区域 TravelMode

Sample request

```
{
    "Origin": [
        -123.11679620827039,
        49.28147612192166
    ],
    "DepartureTime": "2024-10-28T21:27:56Z",
    "Thresholds": {
        "Distance": [
            4000
        ]
    },
    "Avoid": {
        "TollRoads": true,
        "UTurns": true,
        "ControlledAccessHighways": true,
        "Ferries": true,
        "DirtRoads": true,
        "SeasonalClosure": true,
        "CarShuttleTrains": true,
        "TollTransponders": true,
        "ZoneCategories": [
            {
                "Category": "Environmental"
            }
        ]
    },
    "TravelMode": "Truck"
}
```

Sample response

```
{
    "DepartureTime": "2024-10-28T14:27:56-07:00",
    "IsolineGeometryFormat": "FlexiblePolyline",
    "Isolines": [
        {
            "Connections": [],
            "DistanceThreshold": 4000,
            "Geometries": [
```

```
                {
                    "PolylinePolygon": [
                        "Redacted"
                    ]
                }
            ]
        }
    ],
    "SnappedOrigin": [
        -123.11687,
        49.2813999
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/isolines?key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
    "Origin": [
        -123.11679620827039,
        49.28147612192166
    ],
    "DepartureTime": "2024-10-28T21:27:56Z",
    "Thresholds": {
        "Distance": [
            4000
        ]
    },
    "Avoid": {
        "TollRoads": true,
        "UTurns": true,
        "ControlledAccessHighways": true,
        "Ferries": true,
        "DirtRoads": true,
        "SeasonalClosure": true,
        "CarShuttleTrains": true,
        "TollTransponders": true,
        "ZoneCategories": [
            {
                "Category": "Environmental"
            }
```

```
        ]
    },
    "TravelMode": "Truck"
}'
```

AWS CLI

```
aws geo-routes calculate-isolines --key ${YourKey} \
--origin -123.11679620827039 49.28147612192166 \
--departure-time "2024-10-28T21:27:56Z" \
--thresholds '{"Distance": [4000]}' \
--avoid '{"TollRoads": true, "UTurns": true, "ControlledAccessHighways": true,
 "Ferries": true, "DirtRoads": true, "SeasonalClosure": true, "CarShuttleTrains":
 true, "TollTransponders": true, "ZoneCategories": [{"Category": "Environmental"}]}'
 \
--travel-mode "Truck"
```

根据行人避让情况计算服务区

Sample request

```
{
    "Origin": [
        -123.11679620827039,
        49.28147612192166
    ],
    "DepartureTime": "2024-10-28T21:27:56Z",
    "Thresholds": {
        "Distance": [
            4000
        ]
    },
    "Avoid": {
        "TollRoads": true,
        "ControlledAccessHighways": true,
        "Ferries": true,
        "DirtRoads": true,
        "SeasonalClosure": true,
        "CarShuttleTrains": true,
        "TollTransponders": true,
        "ZoneCategories": [
            {
```

```
                "Category": "Environmental"
            }
        ]
    },
    "TravelMode": "Pedestrian"
}
```

Sample response

```
{
    "DepartureTime": "2024-10-28T14:27:56-07:00",
    "IsolineGeometryFormat": "FlexiblePolyline",
    "Isolines": [
        {
            "Connections": [],
            "DistanceThreshold": 4000,
            "Geometries": [
                {
                    "PolylinePolygon": [
                        "Redacted"
                    ]
                }
            ]
        }
    ],
    "SnappedOrigin": [
        -123.11687,
        49.2813999
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/isolines?key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
    "Origin": [
        -123.11679620827039,
        49.28147612192166
    ],
    "DepartureTime": "2024-10-28T21:27:56Z",
    "Thresholds": {
```

```
        "Distance": [
            4000
        ]
    },
    "Avoid": {
        "TollRoads": true,
        "ControlledAccessHighways": true,
        "Ferries": true,
        "DirtRoads": true,
        "SeasonalClosure": true,
        "CarShuttleTrains": true,
        "TollTransponders": true,
        "ZoneCategories": [
            {
                "Category": "Environmental"
            }
        ]
    },
    "TravelMode": "Pedestrian"
}'
```

AWS CLI

```
aws geo-routes calculate-isolines --key ${YourKey} \
--origin -123.11679620827039 49.28147612192166 \
--departure-time "2024-10-28T21:27:56Z" \
--thresholds '{"Distance": [4000]}' \
--avoid '{"TollRoads": true, "ControlledAccessHighways": true, "Ferries":
 true, "DirtRoads": true, "SeasonalClosure": true, "CarShuttleTrains": true,
 "TollTransponders": true, "ZoneCategories": [{"Category": "Environmental"}]}' \
--travel-mode "Pedestrian"
```

使用 Scooter 根据避让计算服务区域 TravelMode

Sample request

```
{
    "Origin": [
        -123.11679620827039,
        49.28147612192166
    ],
    "DepartureTime": "2024-10-28T21:27:56Z",
```

```
    "Thresholds": {
        "Distance": [
            4000
        ]
    },
    "Avoid": {
        "TollRoads": true,
        "ControlledAccessHighways": true,
        "DifficultTurns": false,
        "Ferries": true,
        "DirtRoads": true,
        "SeasonalClosure": true,
        "CarShuttleTrains": true,
        "TollTransponders": true,
        "ZoneCategories": [
            {
                "Category": "Environmental"
            }
        ]
    },
    "TravelMode": "Scooter"
}
```

Sample response

```
{
    "DepartureTime": "2024-10-28T14:27:56-07:00",
    "IsolineGeometryFormat": "FlexiblePolyline",
    "Isolines": [
        {
            "Connections": [],
            "DistanceThreshold": 4000,
            "Geometries": [
                {
                    "PolylinePolygon": [
                        "Redacted"
                    ]
                }
            ]
        }
    ],
    "SnappedOrigin": [
        -123.11687,
```

```
        49.2813999
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/isolines?key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
    "Origin": [
        -123.11679620827039,
        49.28147612192166
    ],
    "DepartureTime": "2024-10-28T21:27:56Z",
    "Thresholds": {
        "Distance": [
            4000
        ]
    },
    "Avoid": {
        "TollRoads": true,
        "ControlledAccessHighways": true,
        "Ferries": true,
        "DirtRoads": true,
        "SeasonalClosure": true,
        "CarShuttleTrains": true,
        "TollTransponders": true,
        "ZoneCategories": [
            {
                "Category": "Environmental"
            }
        ]
    },
    "TravelMode": "Scooter"
}'
```

AWS CLI

```
aws geo-routes calculate-isolines --key ${YourKey} \
--origin -123.11679620827039 49.28147612192166 \
--departure-time "2024-10-28T21:27:56Z" \
--thresholds '{"Distance": [4000]}' \
```

```
--avoid '{"TollRoads": true, "ControlledAccessHighways": true, "Ferries":
 true, "DirtRoads": true, "SeasonalClosure": true, "CarShuttleTrains": true,
 "TollTransponders": true, "ZoneCategories": [{"Category": "Environmental"}]}' \
--travel-mode "Scooter"
```

学习如何使用 CalculateRouteMatrix

在本节中，您将学习CalculateRouteMatrix如何使用查找多个起点和目的地的有效路线。

主题

- 如何计算多个起点和目的地的距离和时间的路径矩阵
- 如何使用避让计算路径矩阵

如何计算多个起点和目的地的距离和时间的路径矩阵

CalculateRouteMatrix API 计算路线，并提供起点和目的地的每种组合的行驶时间和行驶距离。此功能对于需要在多个地点进行路线规划和优化的应用程序非常有用。

潜在用例

- 优化路线规划：使用路线矩阵作为路径优化软件的输入，以提高服务效率并缩短行程时间。

示例

CalculateRouteMatrix 具有无限的路由边界

Sample request

```
{
    "Origins": [
        {
            "Position": [-123.11679620827039, 49.28147612192166]
        },
        {
            "Position": [-123.11179620827039, 49.3014761219]
        }
    ],
    "Destinations": [
        {
            "Position": [-123.112317039, 49.28897192166]
```

```
        }
    ],
    "DepartureTime": "2024-05-28T21:27:56Z",
    "RoutingBoundary": {
        "Unbounded": true
    }
}
```

Sample response

```
{
    "ErrorCount": 0,
    "RouteMatrix": [
        [
            {
                "Distance": 1907,
                "Duration": 343
            }
        ],
        [
            {
                "Distance": 5629,
                "Duration": 954
            }
        ]
    ],
    "RoutingBoundary": {
        "Unbounded": true
    }
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/route-matrix?key=Your_key'
 \
  --header 'Content-Type: application/json' \
  --data '{
    "Origins": [
        {
            "Position": [-123.11679620827039, 49.28147612192166]
        },
        {
```

```
            "Position": [-123.11179620827039, 49.3014761219]
        }
    ],
    "Destinations": [
        {
            "Position": [-123.112317039, 49.28897192166]
        }
    ],
    "DepartureTime": "2024-05-28T21:27:56Z",
    "RoutingBoundary": {
        "Unbounded": true
    }
}'
```

AWS CLI

```
aws geo-routes calculate-route-matrix --key ${YourKey} \
--origins '[{"Position": [-123.11679620827039, 49.28147612192166]}, {"Position":
 [-123.11179620827039, 49.3014761219]}]' \
--destinations '[{"Position": [-123.11179620827039, 49.28897192166]}]' \
--departure-time "2024-05-28T21:27:56Z" \
--routing-boundary '{"Unbounded": true}'
```

CalculateRouteMatrix 使用基于几何的布线边界

Sample request

```
{
    "Origins": [
        {
            "Position": [-123.11679620827039, 49.28147612192166]
        },
        {
            "Position": [-123.11179620827039, 49.3014761219]
        }
    ],
    "Destinations": [
        {
            "Position": [-123.112317039, 49.28897192166]
        }
    ],
    "DepartureTime": "2024-05-28T21:27:56Z",
```

```
    "RoutingBoundary": {
        "Geometry": {
            "AutoCircle": {
                "Margin": 10000,
                "MaxRadius": 30000
            }
        }
    }
}
```

Sample response

```
{
    "ErrorCount": 0,
    "RouteMatrix": [
        [
            {
                "Distance": 1907,
                "Duration": 344
            }
        ],
        [
            {
                "Distance": 5629,
                "Duration": 950
            }
        ]
    ],
    "RoutingBoundary": {
        "Geometry": {
            "Circle": {
                "Center": [
                    -123.1142962082704,
                    49.29147612191083
                ],
                "Radius": 11127
            }
        },
        "Unbounded": false
    }
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/route-matrix?key=Your_key'
 \
  --header 'Content-Type: application/json' \
  --data '{
    "Origins": [
        {
            "Position": [-123.11679620827039, 49.28147612192166]
        },
        {
            "Position": [-123.11179620827039, 49.3014761219]
        }
    ],
    "Destinations": [
        {
            "Position": [-123.112317039, 49.28897192166]
        }
    ],
    "DepartureTime": "2024-05-28T21:27:56Z",
    "RoutingBoundary": {
        "Geometry": {
            "AutoCircle": {
                "Margin": 10000,
                "MaxRadius": 30000
            }
        }
    }
}'
```

AWS CLI

```
aws geo-routes calculate-route-matrix --key ${YourKey} \
--origins '[{"Position": [-123.11679620827039, 49.28147612192166]}, {"Position":
 [-123.11179620827039, 49.3014761219]}]' \
--destinations '[{"Position": [-123.11179620827039, 49.28897192166]}]' \
--departure-time "2024-05-28T21:27:56Z" \
--routing-boundary '{"Geometry": {"AutoCircle": {"Margin": 10000, "MaxRadius":
 30000}}}'
```

如何使用避让计算路径矩阵

CalculateRouteMatrix API 计算路线，并返回从每个起点到指定列表中每个目的地的行驶时间和距离。该 API 可用于为特定区域或道路要素设置避让选项，确保路线避开指定的区域或条件。如果替代路线不可行，则可以绕过避让偏好。

潜在用例

- 路线规划和优化：使用路径矩阵作为软件的输入，该软件需要优化出行路线，同时避开某些区域或道路特征。

示例

CalculateRouteMatrix 带有避让区域

Sample request

```
{
    "Origins": [
        {
            "Position": [-123.11679620827039, 49.28147612192166]
        }
    ],
    "Destinations": [
        {
            "Position": [-123.112317039, 49.28897192166]
        }
    ],
    "Avoid": {
        "Areas": [
            {
                "Geometry": {
                    "BoundingBox": [
                         -123.116561,
                         49.281517,
                         -123.110165,
                         49.285689
                    ]
                }
            }
        ]
    },
```

```
    "RoutingBoundary": {
        "Unbounded": true
    }
}
```

Sample response

```
{
    "ErrorCount": 0,
    "RouteMatrix": [
        [
            {
                "Distance": 1855,
                "Duration": 295
            }
        ]
    ],
    "RoutingBoundary": {
        "Unbounded": true
    }
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/route-matrix?key=Your_key'
 \
  --header 'Content-Type: application/json' \
  --data '{
    "Origins": [
        {
            "Position": [-123.11679620827039, 49.28147612192166]
        }
    ],
    "Destinations": [
        {
            "Position": [-123.112317039, 49.28897192166]
        }
    ],
    "Avoid": {
        "Areas": [
            {
                "Geometry": {
```

```
                    "BoundingBox": [
                         -123.116561,
                         49.281517,
                         -123.110165,
                         49.285689
                    ]
                }
            }
        ]
    },
    "RoutingBoundary": {
        "Unbounded": true
    }
}'
```

AWS CLI

```
aws geo-routes calculate-route-matrix --key ${YourKey} \
--origins '[{"Position": [-123.11679620827039, 49.28147612192166]}]' \
--destinations '[{"Position": [-123.112317039, 49.28897192166]}]' \
--avoid '{"Areas": [{"Geometry": {"BoundingBox": [-123.116561, 49.281517,
 -123.110165, 49.285689]}}]}' \
--routing-boundary '{"Unbounded": true}'
```

CalculateRouteMatrix 避开收费公路、高速公路和渡轮

Sample request

```
{
    "Origins": [
        {
            "Position": [-123.11679620827039, 49.28147612192166]
        }
    ],
    "Destinations": [
        {
            "Position": [-123.112317039, 49.28897192166]
        }
    ],
    "Avoid": {
        "TollRoads": true,
        "ControlledAccessHighways": true,
```

```
        "Ferries": true
    },
    "RoutingBoundary": {
        "Unbounded": true
    }
}
```

Sample response

```
{
    "ErrorCount": 0,
    "RouteMatrix": [
        [
            {
                "Distance": 1855,
                "Duration": 295
            }
        ]
    ],
    "RoutingBoundary": {
        "Unbounded": true
    }
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/route-matrix?key=Your_key'
 \
  --header 'Content-Type: application/json' \
  --data '{
    "Origins": [
        {
            "Position": [-123.11679620827039, 49.28147612192166]
        }
    ],
    "Destinations": [
        {
            "Position": [-123.112317039, 49.28897192166]
        }
    ],
    "Avoid": {
        "TollRoads": true,
```

```
        "ControlledAccessHighways": true,
        "Ferries": true
    },
    "RoutingBoundary": {
        "Unbounded": true
    }
}'
```

AWS CLI

```
aws geo-routes calculate-route-matrix --key ${YourKey} \
--origins '[{"Position": [-123.11679620827039, 49.28147612192166]}]' \
--destinations '[{"Position": [-123.112317039, 49.28897192166]}]' \
--avoid '{"TollRoads": true, "ControlledAccessHighways": true, "Ferries": true}' \
--routing-boundary '{"Unbounded": true}'
```

学习如何使用 OptimizeWaypoints

学习如何使用查找最佳路线`OptimizeWaypoints`，从而最大限度地缩短行驶时间或距离。

主题

- 如何优化路径的航点
- 如何优化具有交通感知功能的路线的航点
- 如何优化具有通行时间感知功能的路线的航点

如何优化路径的航点

OptimizeWaypoints API 会计算一系列航点之间最有效的路线，从而最大限度地减少行驶时间或总距离。此 API 通过考虑道路网络和交通状况来确定最佳路径，从而解决了旅行推销员问题。

潜在用例

- 分析服务区模式：使用航点优化来做出有关业务服务区域的明智决策并提高物流效率。

示例

使用 Car 优化航点 TravelMode

Sample Request

```
{
    "Origin": [
        -123.095740,
        49.274426
    ],
    "Waypoints": [
        {
            "Position": [
                -123.115193,
                49.280596
            ]
        },
        {
            "Position": [
                -123.089557,
                49.271774
            ]
        }
    ],
    "DepartureTime": "2024-10-25T18:13:42Z",
    "Destination": [
        -123.095185,
        49.263728
    ],
    "TravelMode": "Car"
}
```

Sample Response

```
{
    "Connections": [
        {
            "Distance": 1989,
            "From": "Origin",
            "RestDuration": 0,
            "To": "Waypoint0",
            "TravelDuration": 258,
```

```
            "WaitDuration": 0
        },
        {
            "Distance": 3010,
            "From": "Waypoint0",
            "RestDuration": 0,
            "To": "Waypoint1",
            "TravelDuration": 298,
            "WaitDuration": 0
        },
        {
            "Distance": 2371,
            "From": "Waypoint1",
            "RestDuration": 0,
            "To": "Destination",
            "TravelDuration": 311,
            "WaitDuration": 0
        }
    ],
    "Distance": 7370,
    "Duration": 867,
    "ImpedingWaypoints": [],
    "OptimizedWaypoints": [
        {
            "DepartureTime": "2024-10-25T18:13:42Z",
            "Id": "Origin",
            "Position": [
                -123.09574,
                49.274426
            ]
        },
        {
            "DepartureTime": "2024-10-25T18:18:00Z",
            "Id": "Waypoint0",
            "Position": [
                -123.115193,
                49.280596
            ]
        },
        {
            "DepartureTime": "2024-10-25T18:22:58Z",
            "Id": "Waypoint1",
            "Position": [
                -123.089557,
```

```
                49.271774
            ]
        },
        {
            "ArrivalTime": "2024-10-25T18:28:09Z",
            "Id": "Destination",
            "Position": [
                -123.095185,
                49.263728
            ]
        }
    ],
    "TimeBreakdown": {
        "RestDuration": 0,
        "ServiceDuration": 0,
        "TravelDuration": 867,
        "WaitDuration": 0
    }
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/optimize-waypoints?
key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
    "Origin": [
        -123.095740,
        49.274426
    ],
    "Waypoints": [
        {
            "Position": [
                -123.115193,
                49.280596
            ]
        },
        {
            "Position": [
                -123.089557,
                49.271774
            ]
```

```
        }
    ],
    "DepartureTime": "2024-10-25T18:13:42Z",
    "Destination": [
        -123.095185,
        49.263728
    ],
    "TravelMode": "Car"
}'
```

AWS CLI

```
aws geo-routes optimize-waypoints --key ${YourKey} \
--origin -123.095740 49.274426 \
--waypoints '[{"Position": [-123.115193 , 49.280596]}, {"Position": [-123.089557 ,
 49.271774]}]' \
--destination -123.095185 49.263728 \
--departure-time "2024-10-25T18:13:42Z" \
--travel-mode "Car"
```

如何优化具有交通感知功能的路线的航点

OptimizeWaypoints API 计算多个航点之间的最佳路线，以最大限度地缩短行驶时间或总距离。它利用先进的算法来解决旅行推销员问题，确定最有效的路径，同时考虑道路网络和实时交通状况等因素。

潜在用例

- 优化多站路线，提高配送效率：通过计算多个停靠点中最短或最快的路线，改善配送操作。这对于降低物流和配送服务的运营成本、油耗和旅行时间非常有用。

示例

使用汽车优化具有交通感知功能的航点 TravelMode

Sample request

```
{
    "Origin": [
        -123.095740,
        49.274426
    ],
```

```
    "Waypoints": [
        {
            "Position": [
                -123.115193,
                49.280596
            ]
        },
        {
            "Position": [
                -123.089557,
                49.271774
            ]
        }
    ],
    "DepartureTime": "2024-10-25T18:13:42Z",
    "Destination": [
        -123.095185,
        49.263728
    ],
    "TravelMode": "Car",
    "Traffic": {
        "Usage": "UseTrafficData"
    }
}
```

Sample response

```
{
    "Connections": [
        {
            "Distance": 1989,
            "From": "Origin",
            "RestDuration": 0,
            "To": "Waypoint0",
            "TravelDuration": 324,
            "WaitDuration": 0
        },
        {
            "Distance": 2692,
            "From": "Waypoint0",
            "RestDuration": 0,
            "To": "Waypoint1",
            "TravelDuration": 338,
```

```
            "WaitDuration": 0
        },
        {
            "Distance": 2371,
            "From": "Waypoint1",
            "RestDuration": 0,
            "To": "Destination",
            "TravelDuration": 395,
            "WaitDuration": 0
        }
    ],
    "Distance": 7052,
    "Duration": 1057,
    "ImpedingWaypoints": [],
    "OptimizedWaypoints": [
        {
            "DepartureTime": "2024-10-25T18:13:42Z",
            "Id": "Origin",
            "Position": [
                -123.09574,
                49.274426
            ]
        },
        {
            "ArrivalTime": "2024-10-25T18:19:06Z",
            "DepartureTime": "2024-10-25T18:19:06Z",
            "Id": "Waypoint0",
            "Position": [
                -123.115193,
                49.280596
            ]
        },
        {
            "ArrivalTime": "2024-10-25T18:24:44Z",
            "DepartureTime": "2024-10-25T18:24:44Z",
            "Id": "Waypoint1",
            "Position": [
                -123.089557,
                49.271774
            ]
        },
        {
            "ArrivalTime": "2024-10-25T18:31:19Z",
            "Id": "Destination",
```

```
            "Position": [
                -123.095185,
                49.263728
            ]
        }
    ],
    "TimeBreakdown": {
        "RestDuration": 0,
        "ServiceDuration": 0,
        "TravelDuration": 1057,
        "WaitDuration": 0
    }
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/optimize-waypoints?
key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
    "Origin": [
        -123.095740,
        49.274426
    ],
    "Waypoints": [
        {
            "Position": [
                -123.115193,
                49.280596
            ]
        },
        {
            "Position": [
                -123.089557,
                49.271774
            ]
        }
    ],
    "DepartureTime": "2024-10-25T18:13:42Z",
    "Destination": [
        -123.095185,
        49.263728
```

```
    ],
    "TravelMode": "Car",
    "Traffic": {
        "Usage": "UseTrafficData"
    }
}'
```

AWS CLI

```
aws geo-routes optimize-waypoints --key ${YourKey} \
--origin -123.095740 49.274426 \
--waypoints '[{"Position": [-123.115193 , 49.280596]}, {"Position": [-123.089557 ,
 49.271774]}]' \
--destination -123.095185 49.263728 \
--departure-time "2024-10-25T18:13:42Z" \
--travel-mode "Car" \
--traffic '{"Usage": "UseTrafficData"}'
```

如何优化具有通行时间感知功能的路线的航点

OptimizeWaypoints API 还会计算一组航点之间的最佳路线，目标是最大限度地减少行驶时间或总行驶距离。它解决了旅行推销员在考虑道路网络和交通状况等因素的情况下确定最有效的路径的问题。

潜在用例

- 分析客户访问时间：围绕客户的访问时间制定效率计划。

示例

使用 Car 通过通行时间感知来优化航点 TravelMode

Sample Request

```
{
    "Origin": [
        -123.095740,
        49.274426
    ],
    "Waypoints": [
        {
            "Position": [
```

```
                -123.115193,
                49.280596
            ],
            "SideOfStreet": {
                "Position": [
                    -123.089557,
                    49.271774
                ],
                "UseWith": "AnyStreet"
            },
            "AccessHours": {
                "From": {
                    "DayOfWeek": "Saturday",
                    "TimeOfDay": "00:02:42Z"
                },
                "To": {
                    "DayOfWeek": "Friday",
                    "TimeOfDay": "1:33:36+02:50"
                }
            },
            "Heading": "250",
            "ServiceDuration": "200"
        },
        {
            "Position": [
                -123.089557,
                49.271774
            ],
            "AccessHours": {
                "From": {
                    "DayOfWeek": "Monday",
                    "TimeOfDay": "00:02:42Z"
                },
                "To": {
                    "DayOfWeek": "Tuesday",
                    "TimeOfDay": "1:33:36+02:50"
                }
            },
            "ServiceDuration": "200"
        }
    ],
    "DepartureTime": "2024-10-25T18:13:42Z",
    "Destination": [
        -123.095185,
```

```
        49.263728
    ],
    "TravelMode": "Car"
}
```

Sample Response

```
{
    "Connections": [
        {
            "Distance": 1989,
            "From": "Origin",
            "RestDuration": 0,
            "To": "Waypoint0",
            "TravelDuration": 258,
            "WaitDuration": 20682
        },
        {
            "Distance": 3360,
            "From": "Waypoint0",
            "RestDuration": 0,
            "To": "Waypoint1",
            "TravelDuration": 378,
            "WaitDuration": 172222
        },
        {
            "Distance": 2371,
            "From": "Waypoint1",
            "RestDuration": 0,
            "To": "Destination",
            "TravelDuration": 311,
            "WaitDuration": 0
        }
    ],
    "Distance": 7720,
    "Duration": 194251,
    "ImpedingWaypoints": [],
    "OptimizedWaypoints": [
        {
            "DepartureTime": "2024-10-25T18:13:42Z",
            "Id": "Origin",
            "Position": [
                -123.09574,
```

```
                49.274426
            ]
        },
        {
            "ArrivalTime": "2024-10-25T18:18:00Z",
            "DepartureTime": "2024-10-26T00:06:02Z",
            "Id": "Waypoint0",
            "Position": [
                -123.115193,
                49.280596
            ]
        },
        {
            "ArrivalTime": "2024-10-26T00:12:20Z",
            "DepartureTime": "2024-10-28T00:06:02Z",
            "Id": "Waypoint1",
            "Position": [
                -123.089557,
                49.271774
            ]
        },
        {
            "ArrivalTime": "2024-10-28T00:11:13Z",
            "Id": "Destination",
            "Position": [
                -123.095185,
                49.263728
            ]
        }
    ],
    "TimeBreakdown": {
        "RestDuration": 0,
        "ServiceDuration": 400,
        "TravelDuration": 947,
        "WaitDuration": 192904
    }
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/optimize-waypoints?
key=Your_key' \
```

```
--header 'Content-Type: application/json' \
--data '{
  "Origin": [
      -123.095740,
      49.274426
  ],
  "Waypoints": [
      {
          "Position": [
              -123.115193,
              49.280596
          ],
          "SideOfStreet": {
              "Position": [
                  -123.089557,
                  49.271774
              ],
              "UseWith": "AnyStreet"
          },
          "AccessHours": {
              "From": {
                  "DayOfWeek": "Saturday",
                  "TimeOfDay": "00:02:42Z"
              },
              "To": {
                  "DayOfWeek": "Friday",
                  "TimeOfDay": "1:33:36+02:50"
              }
          },
          "Heading": "250",
          "ServiceDuration": "200"
      },
      {
          "Position": [
              -123.089557,
              49.271774
          ],
          "AccessHours": {
              "From": {
                  "DayOfWeek": "Monday",
                  "TimeOfDay": "00:02:42Z"
              },
              "To": {
                  "DayOfWeek": "Tuesday",
```

```
                    "TimeOfDay": "1:33:36+02:50"
                }
            },
            "ServiceDuration": "200"
        }
    ],
    "DepartureTime": "2024-10-25T18:13:42Z",
    "Destination": [
        -123.095185,
        49.263728
    ],
    "TravelMode": "Car"
}'
```

AWS CLI

```
aws geo-routes optimize-waypoints --key ${YourKey} \
--origin -123.095740 49.274426 \
--waypoints '[{"Position": [-123.115193 , 49.280596], "SideOfStreet": {"Position":
 [-123.089557, 49.271774], "UseWith": "AnyStreet"}, "AccessHours": {"From":
 {"DayOfWeek": "Saturday", "TimeOfDay": "00:02:42Z"}, "To": {"DayOfWeek":
 "Friday", "TimeOfDay": "1:33:36+02:50"}}, "Heading": 250, "ServiceDuration":
 200}, {"Position": [-123.089557, 49.271774], "AccessHours": {"From": {"DayOfWeek":
 "Monday", "TimeOfDay": "00:02:42Z"}, "To": {"DayOfWeek": "Tuesday", "TimeOfDay":
 "1:33:36+02:50"}}, "ServiceDuration": 200}]' \
--destination -123.095185 49.263728 \
--departure-time "2024-10-25T18:13:42Z" \
--travel-mode "Car"
```

学习如何使用 SnapToRoads

本主题介绍SnapToRoads如何使用将 GPS 轨迹与道路网络对齐，从而提高导航和车队管理应用程序的定位精度。此 API 通过将坐标捕捉到最近的路段来纠正 GPS 漂移和信号丢失，同时还要遵守出行模式限制。示例说明了实际用途，例如叠加 GPS 轨迹、填补数据空白以及降低噪音以实现更清晰的路径可视化。

主题

- 如何将 GPS 轨迹与道路网络进行匹配

如何将 GPS 轨迹与道路网络进行匹配

SnapToRoads API 允许您将 GPS 轨迹与道路网络进行匹配。GPS 追踪包括使用 GPS 设备记录的位置和元数据，例如时间戳、速度和方向。这些轨迹通常具有误差幅度，因此很难直接用于分析和可视化。

SnapToRoads 在匹配轨迹时，会考虑指定出行模式的法律和时间限制。如果追踪线强烈暗示存在违反限制的情况，则保留所走的实际路线。

潜在用例

- 将 GPS 轨迹叠加到最有可能行驶的道路上：此功能有助于将 GPS 数据与道路网络上最准确的路径对齐，从而支持更清晰的数据可视化。
- 插值 GPS 轨迹中的间隙： SnapToRoads 可以通过将坐标捕捉到路段来填补空白，从而为应用程序创建更连续、更有用的数据集。
- 过滤噪点和异常值：通过捕捉到最近的道路，此 API 可以帮助移除异常值并降低 GPS 噪音，从而提高分析的数据可靠性。

示例

使用汽车模式匹配 GPS 轨迹

Sample request

```
{
  "TracePoints": [
    {
      "Position": [8.53404,50.16364],
      "Timestamp": "2024-05-22T18:13:42Z"
    },
    {
      "Position": [8.53379056,50.16352417],
      "Speed": 20,
      "Timestamp": "2024-05-22T18:13:59Z"
    }
  ],
  "TravelMode": "Car"
}
```

Sample response

```
{
    "Notices": [],
    "SnappedGeometry": {
        "Polyline": "Redacted"
    },
    "SnappedGeometryFormat": "FlexiblePolyline",
    "SnappedTracePoints": [
        {
            "Confidence": 1,
            "OriginalPosition": [8.53404, 50.16364],
            "SnappedPosition": [8.53402, 50.16367]
        },
        {
            "Confidence": 0.86,
            "OriginalPosition": [8.53379056, 50.16352417],
            "SnappedPosition": [8.53375, 50.16356]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/snap-to-roads?
key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
  "TracePoints": [
    {
      "Position": [8.53404,50.16364],
      "Timestamp": "2024-05-22T18:13:42Z"
    },
    {
      "Position": [8.53379056,50.16352417],
      "Speed": 20,
      "Timestamp": "2024-05-22T18:13:59Z"
    }
  ],
  "TravelMode": "Car"
}'
```

AWS CLI

```
aws geo-routes snap-to-roads --key ${YourKey} \
--trace-points '[{"Position": [8.53404, 50.16364], "Timestamp":
 "2024-05-22T18:13:42Z"}, {"Position": [8.53379056, 50.16352417], "Speed": 20,
 "Timestamp": "2024-05-22T18:13:59Z"}]' \
--travel-mode "Car"
```

使用卡车模式将 GPS 追踪与选项进行匹配

Sample request

```
{
  "TracePoints": [
    {
      "Position": [8.53404,50.16364],
      "Timestamp": "2024-05-22T18:13:42Z"
    },
    {
      "Position": [8.53379056,50.16352417],
      "Speed": 20,
      "Timestamp": "2024-05-22T18:13:59Z"
    }
  ],
  "TravelMode": "Truck",
  "TravelModeOptions": {
    "Truck": {
      "GrossWeight": 10000
    }
  }
}
```

Sample response

```
{
    "Notices": [],
    "SnappedGeometry": {
        "Polyline": "Redacted"
    },
    "SnappedGeometryFormat": "FlexiblePolyline",
    "SnappedTracePoints": [
        {
```

```
            "Confidence": 1,
            "OriginalPosition": [8.53404, 50.16364],
            "SnappedPosition": [8.53402, 50.16367]
        },
        {
            "Confidence": 0.86,
            "OriginalPosition": [8.53379056, 50.16352417],
            "SnappedPosition": [8.53375, 50.16356]
        }
    ]
}
```

cURL

```
curl --request POST \
  --url 'https://routes.geo.eu-central-1.amazonaws.com/v2/snap-to-roads?
key=Your_key' \
  --header 'Content-Type: application/json' \
  --data '{
  "TracePoints": [
    {
      "Position": [8.53404,50.16364],
      "Timestamp": "2024-05-22T18:13:42Z"
    },
    {
      "Position": [8.53379056,50.16352417],
      "Speed": 20,
      "Timestamp": "2024-05-22T18:13:59Z"
    }
  ],
  "TravelMode": "Truck",
  "TravelModeOptions": {
    "Truck": {
      "GrossWeight": 10000
    }
  }
}'
```

AWS CLI

```
aws geo-routes snap-to-roads --key ${YourKey} \
```

```
--trace-points '[{"Position": [8.53404, 50.16364], "Timestamp":
 "2024-05-22T18:13:42Z"}, {"Position": [8.53379056, 50.16352417], "Speed": 20,
 "Timestamp": "2024-05-22T18:13:59Z"}]' \
--travel-mode "Truck" \
--travel-mode-options '{"Truck": {"GrossWeight": 10000}}'
```

管理成本和使用量

在您继续学习 Amazon Location 路线时，了解如何管理服务容量、确保遵守使用限制以及通过配额和 API 优化获得最佳结果非常重要。通过应用性能和准确性方面的最佳实践，您可以定制应用程序，以高效地处理与地点相关的查询并最大限度地提高 API 请求。

主题

- 最佳实践
- 航线定价
- 路线配额和使用情况

最佳实践

本节介绍在与 API 交互时使用压缩以及在简单 (geoJSON) 和 FlexiblePolyline 格式之间进行选择的最佳实践，并提供了有关优化性能、带宽和数据处理的指导。

压缩

为了提高应用程序在与我们的 API 交互时的性能和效率，建议启用响应压缩，尤其是在处理基于文本的大型负载时。您可以通过在 API 请求中包含标`Accept-Encoding`头来激活压缩，并指定您的首选压缩方法。我们支持`gzip`并`deflate`支持它们的压缩功能，`gzip`通常会提供更好的压缩比。

何时启用压缩

大量回应

为基于文本的大型响应启用压缩，以减少带宽使用量并缩短加载时间。

网络限制

如果您的应用程序在有限的带宽或高延迟的网络上运行，则压缩可以提高数据传输效率。

如何有效使用压缩

设置接受编码标头

`Accept-Encoding: gzip, deflate`在您的 HTTP 请求中加入以告知我们的 API 您支持这些压缩方法。启用和处理压缩的方法因 AWS 开发工具包和编程语言而异。例如，适用于 Java 的 AWS 开发工具包 v1 使用`ClientConfiguration`类中的`withGzip`方法来启用 gzip，而 AWS SDK for Go 则需要添加特定的中间件来进行压缩处理。有关其他 SDKs，请参阅 AWS 开发工具包参考指南以获取详细说明。

正确处理解压问题

确保您的客户端应用程序可以根据我们的 API 返回的`Content-Encoding`标头正确解压缩响应。

测试和监控

定期评估压缩对应用程序性能的影响，平衡减小有效载荷大小的好处和解压缩过程产生的额外的 CPU 开销。

折线

在与我们的 API 交互时，在简单 (geoJSON) 和 FlexiblePolyline 格式之间进行选择的最佳实践，以优化地理空间数据的性能和可用性。

使用简单 (GeoJSON) 格式

可读性和标准化

当您需要一种广泛认可且易于理解的格式时，可以使用，以便于调试并与各种地理空间工具互操作。

精度

当您的应用程序需要高精度坐标时，请选择简单格式，因为 GeoJSON 可以保持完整的十进制精度而不会丢失。

较小的数据集

当处理较小的坐标数据集时，简单格式是理想的选择，因为压缩所带来的减小尺寸的好处微乎其微。

使用 FlexiblePolyline 格式

减少数据大小

FlexiblePolyline 当您需要利用有损压缩技术来最大限度地减少传输的数据量（尤其是对于大型坐标列表而言）时，它是理想的选择。

网址安全

FlexiblePolyline 提供了一个紧凑的、URL 安全的字符串，无需额外编码即可直接在查询参数中使用。

性能优化

FlexiblePolyline 有助于减小有效载荷大小，从而加快数据传输速度和降低带宽使用量，因此对于高性能应用程序或在受限网络上运行的应用程序至关重要。

航线定价

请参阅下文，了解每个 API 的定价范围：

计算路线

此价格基于 API 请求的数量。 `CalculateRoutes`有三个定价区间：核心版、高级版和高级版。

核心实例

此价格区间支持汽车、卡车和行人出行模式，无需计算通行费。

高级

此价格区间支持其他出行模式，例如滑板车，无需计算通行费。

高级

此价格区间支持通行费成本计算。当您请求通过设置请求参数计算通行费时，`LegAdditionalFeatures["Tolls"]`无论出行模式如何`SpanAdditionalFeatures["TollSystems"]`，都将按高级价格收费。

计算路径矩阵

此价格基于计算出的路线数量。每次请求中计算的路径数量等于起点数乘以目的地数量。`Number of Routes = Number of origins x Number of Destinations`例如，当使用大小为 300 个起点乘以 100 个目的地的矩阵时，计算出的路径总数为 30,000（300 x 100 = 30,000）。

> **Note**
>
> 每对起点和目的地对均按路径计算费用。如果您使用大量的出发地和目的地矩阵，您的费用将相应增加。

CalculateRouteMatrix有 2 个定价区：核心和高级。

核心实例

此价格区间支持汽车、卡车和行人出行模式。

高级

此价格区间支持其他出行模式，例如滑板车。

优化航点

此价格基于 API 请求的数量。 OptimizeWaypoint有 2 个定价区间：高级版和高级版。

高级

此定价区在单个请求中支持多达 30 个航点；汽车、卡车和行人的出行模式，输入点的边界框在 200 千米以内，没有可选参数（例如、Avoid、、、、、、ClusteringDriverExclude.CountriesTravelModeOptions.Truck.Ha也没有其他航点或目的地限制，例如AccessHours、、、、、AppointmentTime、Before。Heading ServiceDuration SideOfStreet

> **Note**
>
> 当航点近在咫尺时，可能会发生自动聚类，但它仍被视为高级定价桶。

高级

此定价区在单个请求中最多支持 50 个航点；对出行模式没有限制；500 公里以内的输入点的边界框；并带有可选参数Avoid，例如、、、、、ClusteringDriver、Exclude.Countries。TravelModeOptions.Truck.HazardouTravelModeOptions.Truck.TunnelRestrictionCode此外，此定价区还支持可选的航点和

目的地限制，例
如AccessHours、AppointmentTime、、Before、HeadingServiceDuration、SideOfStreet。

> **Note**
>
> 如果应用了任何可选的航点和目的地限制，则单个请求最多只能支持 20 个航点。

S nap-to-road

此价格基于 API 请求的数量。 SnaptoRoad有 2 个定价区间：高级版和高级版。

高级

此定价区支持汽车、卡车和行人出行模式，TracePoints计数最多为 200，最大飞行距离介于 100 TracePoints 千米之间。

高级

此定价套餐对出行方式没有限制，最高可达 5,000 TracePoints 点积分。

计算等值线

此价格基于响应中计算的等值线数量。 CalculateIsoline有 2 个定价区间：高级版和高级版。

高级

此定价区支持汽车、卡车和行人出行模式，其Thresholds.Time值不超过 60 分钟，Thresholds.Distance值不超过 100 千米。

高级

此定价区对出行模式没有限制，Thresholds.Time值不超过 180 分钟，Thresholds.Distance值不超过 300 公里。

路线配额和使用情况

服务配额

Amazon Location Service APIs 有默认配额。您可以使用服务配额控制台增加配额。如果限制超过默认值的 2 倍，请通过自助服务控制台申请或联系支持人员。

服务配额限制

API 名称	默认	最大可调极限	超过可调节的最大限制
the section called “计算路线”	20	40	在服务配额控制台上申请或联系支持团队
CalculateIsoline	20	40	在服务配额控制台上申请或联系支持团队
the section called “Snap to R”	20	40	在服务配额控制台上申请或联系支持团队
the section called “计算路径矩阵”	5	10	在服务配额控制台上申请或联系支持团队
the section called “优化航点”	5	10	在服务配额控制台上申请或联系支持团队

其他使用限制

除服务配额外，还适用以下 API 使用限制：

其他使用限制

API 名称	限制	值
the section called “Snap to R”	所有距离之间的测地线距离之和 TracePoints	500 千米
the section called “优化航点”	起点、按提供的顺序排列的航点和目的地之间的测地距离之和	100KM
the section called “优化航点”	起点、航点和目的地周围的边界框的周长	500 千米
the section called “计算路径矩阵”	无界路径的起点和目的地之间的最大距离（如果使用	60 千米

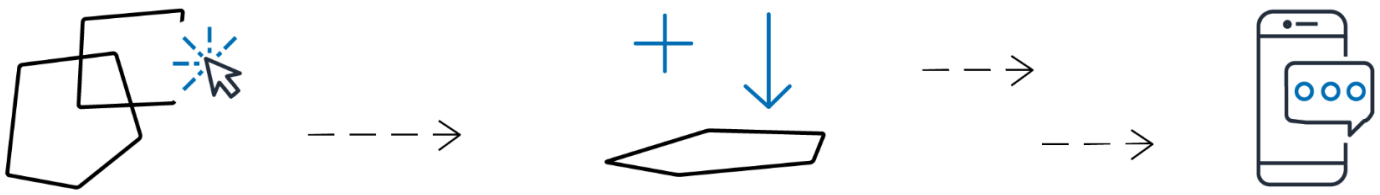
API 名称	限制	值
	A TravelModeOptions void 或 .Truck ）	
the section called “计算路径矩阵”	无界路径的起点和目的地之间的最大距离	10000KM
the section called “计算路线”	压缩后的响应有效载荷大小	6 MB
the section called “计算路径矩阵”	压缩后的响应有效载荷大小	6 MB
the section called “计算等值线”	压缩后的响应有效载荷大小	6 MB
the section called “优化航点”	压缩后的响应有效载荷大小	6 MB
the section called “Snap to R”	压缩后的响应有效载荷大小	6 MB

后续步骤

请查看以下内容了解更多详情：

- 归因：有关亚马逊定位服务（Amazon Location Service）的数据归因要求的信息。
- SLA：Amazon Location Service 的服务等级协议，包括正常运行时间承诺和响应时间。
- 服务条款：管理使用 Amazon Location Service 的条款，包括限制和限制。

Amazon Location Service 地理围栏



Geofence 集合资源允许您存储和管理地理围栏（地图上的虚拟边界）。您可以根据地理围栏集合资源评估位置，并在位置更新越过集合中任何地理围栏的边界时收到通知。

地理围栏和地理围栏集合

地理围栏是一种在地图上定义虚拟边界的圆形、多边形或多面几何。地理围栏集合包含零个或多个地理围栏。它能够通过发射ENTER和EXIT事件进行地理围栏监控，并在需要时根据其地理围栏评估设备的位置。

地理围栏事件

您正在监控的位置由名为 a 的 ID 引用DeviceId。这些位置被称为设备位置。您可以将要评估的设备位置列表直接发送到地理围栏集合资源，也可以使用跟踪器。有关使用跟踪器的更多信息，请参阅跟踪器。

只有当设备进入或退出地理围栏时，您才会收到事件（通过 Amazon EventBridge），而不是每次位置变化都会收到事件。这意味着您通常会收到事件，并且响应事件的频率要比每次设备位置更新的频率低得多。

> **Note**
>
> 对于特定 DeviceID 的首次位置评估，假设该设备以前不在任何地理围栏中。因此，如果在集合中的地理围栏内，则第一次更新将生成一个 ENTER 事件，如果不是，则不会生成任何事件。
>
> 为了计算设备是进入还是退出地理围栏，Amazon Location Service 必须保持该设备的先前位置状态。此位置状态可存储 30 天。在设备未更新 30 天后，新的位置信息更新将被视为首次位置更新。

Amazon Location Service Geofences 的用例

以下是 Amazon Location Service Geofences 的一些常见用法。

改善现场服务运营

通过实时跟踪，随时掌握移动办公人员的脉搏。在客户地点和服务区周围设置地理围栏，以便在员工到达和离开时收到警报。使用位置数据来优化日程安排，派遣最近的可用技术人员并缩短响应时间。让您的现场团队（例如管道或暖通空调维修业务）提高工作效率，同时增强客户体验。

监控和控制关键资产

利用 Amazon Location Service 来跟踪您的贵重设备、库存和其他移动资产的实时位置和状态。设置地理围栏以接收有关未经授权的移动或搬迁的警报，从而增强安全性和合规性。利用这种位置可视性来提高资产利用率，优化维护计划，并确保您的关键资源始终得到考虑。始终精确监控您的重型机械、IT 硬件或零售库存，减少损失，并做出更明智的运营决策。

提高供应链可见性

利用 Amazon Location Service 来跟踪整个供应链中的货件和配送。在配送中心、商店和其他关键设施周围定义地理围栏，以监控库存和资产的流动。使用实时位置数据来改善库存管理、优化物流计划并提供卓越的客户体验。深入 end-to-end了解您的供应链运营，识别瓶颈，并做出以数据为导向的决策，从而提高效率和响应能力。

加强安全保障

地理围栏使您能够在安全区域、禁区和其他关键位置周围设置虚拟边界。当未经授权的人员或资产进入或离开这些预定义的地理围栏时，会立即收到警报。利用这种实时位置监控来增强工作场所安全，阻止非法侵入，并确保合规性。无论您是管理制造工厂、建筑工地还是企业园区，地理围栏都使您能够更严格地控制准入，改善事件响应，并保护您的人员、财产和资产。

基于位置的营销

释放位置数据的力量，增强您的地理营销工作。使用 Amazon Location Service 在竞争对手的地点、活动和高流量区域周围设置虚拟边界。当客户进入这些地理围栏区域时，触发个性化广告、优惠和通知。分析客流量模式以优化广告投放位置，发现新营业地点的主要网站。监控客户在自己的地理围栏空间内的活动，以更深入地了解浏览行为和. path-to-purchase 将实时位置跟踪与精确的地理围栏相结合，提供超有针对性的情境式互动，从而推动现实世界中的销售和忠诚度。

地理围栏概念

本节介绍一些常见的地理围栏概念，包括常用术语以及如何管理地理围栏。

Amazon Location Service 地理围栏术语

地理围栏

地图上地理区域的虚拟边界。

地理围栏集合

一组地理围栏，能够在需要时发射Entry和Exit事件，以评估设备相对于其组件地理围栏的位置。

地理围栏几何

一种几何形状或一组形状，用于定义要包含或排除在地理围栏之外的区域。地理围栏几何形状可以表示为CirclePolygon、或。MultiPolygon

A Circle 是一个围绕它有一段距离的点。如果您想在设备距离某个位置一定距离内时收到通知，请使用圆。

A Polygon 是代表地理围栏形状的线性环列表。此列表必须包括代表地理围栏外围的外环，并且可以选择包括多个表示周长内多边形空间的内环，这些空间不在地理围栏区域之外。

A MultiPolygon 是代表地理围栏形状的列表Polygon元素。MultiPolygon几何结构的Polygon组成部分可以定义被视为同一地理围栏一部分的单独地理区域、被排除在地理围栏之外的具有较小内部空间的较大外部区域的周长，或者将这些用例的某种组合来形成复杂的地理围栏边界。

有关定义地理围栏几何图形的更多信息，包括语法要求、限制和示例，请参阅GeofenceGeometry《Amazon Location S ervice API 参考》。

开始使用 Amazon Location Service Geofences

地理围栏是定义地理边界和根据位置更新触发操作的强大工具。本指南将引导您完成在 Amazon Location 中创建和使用地理围栏收集资源的过程。通过设置地理围栏并根据地理围栏评估位置，您可以监控移动情况并生成自动事件，例如设备进入或离开定义区域时的通知。这些功能非常适合车队跟踪、基于位置的通知等应用程序。

1. 在您的 AWS 账户中创建地理围栏收集资源。

2. 将地理围栏添加到集合中。您可以使用亚马逊位置控制台上的地理围栏上传工具或亚马逊定位地理围栏 API。有关可用选项的更多信息，请参阅身份验证。地理围栏可以定义为`CirclePolygon`、或。`MultiPolygon`使用`Polygon`或`MultiPolygon`来确定设备何时进入特定区域。使用确定设备何时到达距离某一点的特定距离（半径）以内。`Circle`有关更多信息，请参阅 Amazon Location Service 地理围栏术语。
3. 您可以开始根据所有地理围栏评估位置。当位置更新跨越一个或多个地理围栏的边界时，您的地理围栏收集资源会在亚马逊上发出以下地理围栏事件类型之一： EventBridge

   - ENTER – 为每个地理围栏生成一个事件，其中位置更新通过进入来越过边界。
   - EXIT – 每个地理围栏都会生成一个事件，其中位置更新通过离开来越过边界。

   有关更多信息，请参阅 the section called “通过以下方式对事件做出反应 EventBridge”。您还可以使用诸如Amazon CloudWatch 和之类的服务来集成监控 AWS CloudTrail。有关更多信息，请参阅 the section called “使用 Amazon 进行监控 CloudWatch” 和 the section called “使用监控和登录 AWS CloudTrail”。

例如，您正在追踪一支卡车车队，并希望在卡车驶入任何仓库的特定区域时收到通知。为每个仓库周围的区域创建地理围栏。当卡车向您发送更新的位置时，请使用 Amazon Location Service 评估这些位置，并查看卡车是否进入（或退出）了其中一个地理围栏区域。

> **Note**
>
> 按您评估的地理围栏集合数量计费。您的计费不受每个集合中地理围栏数量的影响。由于每个地理围栏集合可能包含多达 50,000 个地理围栏，因此您可能需要尽可能将地理围栏合并成更少的集合，以降低地理围栏评估的成本。生成的事件将包括集合中各个地理围栏的 ID 以及该集合的 ID。

如何使用 Amazon Location Service Geofences

本节为在 Ama step-by-step zon Location 中处理与地理围栏相关的任务提供了指导。了解如何根据地理围栏评估设备位置，如何使用 Amazon 响应地理围栏事件 EventBridge，以及如何有效管理您的地理围栏资源。这些教程旨在帮助您实现轻松跟踪和管理基于位置的事件的关键功能。

主题

- 根据地理围栏评估设备位置

- 使用亚马逊对亚马逊定位服务事件做出反应 EventBridge
- 管理您的地理围栏收集资源

根据地理围栏评估设备位置

有两种方法可以根据地理围栏评估位置以生成地理围栏事件：

- 您可以关联跟踪器和地理围栏集合。想要了解更多信息，请参阅 将跟踪器关联到地理围栏集合 部分。
- 您可以直接向地理围栏集合资源提出请求，以评估一个或多个位置。

如果您还想跟踪设备位置记录或在地图上显示位置，请将跟踪器链接到地理围栏集合。或者，您可能不想评估所有位置更新，或者您不打算将位置数据存储在跟踪器资源中。如果是其中任何一种情况，则可以直接向地理围栏集合发出请求，并根据其地理围栏评估一个或多个设备的位置。

根据地理围栏评估设备位置会生成事件。您可以对这些事件做出反应并将其路由到其他 AWS 服务。有关接收地理围栏事件时可以采取的操作的更多信息，请参阅通过亚马逊对亚马逊定位服务事件做出反应。 EventBridge

Amazon Location 事件包括生成该事件的设备位置更新的属性，包括时间、位置、准确性和键值元数据，以及进入或退出的地理围栏的一些属性。有关地理围栏事件中包含的数据的更多信息，请参阅 the section called “事件示例”。

以下示例使用或 AWS CLI Amazon 地点 APIs。

API

使用 Amazon 位置根据地理围栏的位置评估设备位置 APIs

使用 Amazon 定位地理围 APIs栏中的BatchEvaluateGeofences操作。

以下示例使用 API 请求来评估设备*ExampleDevice*在关联的地理围栏集合中的位置。*ExampleGeofenceCollection*将这些值替换为您自己的地理围栏和设备。 IDs

```
POST /geofencing/v0/collections/ExampleGeofenceCollection/positions HTTP/1.1
Content-type: application/json

{
   "DevicePositionUpdates": [
```

```
      {
         "DeviceId": "ExampleDevice",
         "Position": [-123.123, 47.123],
         "SampleTime": "2021-11-30T21:47:25.149Z",
         "Accuracy": {
            "Horizontal": 10.30
         },
         "PositionProperties": {
            "field1": "value1",
            "field2": "value2"
         }
      }
   ]
}
```

AWS CLI

使用命令根据地理围栏的位置评估设备位置 AWS CLI

使用 batch-evaluate-geofences 命令。

以下示例使用 AWS CLI 来评估*ExampleDevice*相对于关联的地理围栏集合的位置。*ExampleGeofenceCollection*将这些值替换为您自己的地理围栏和设备。 IDs

```
aws location \
    batch-evaluate-geofences \
        --collection-name ExampleGeofenceCollection \
        --device-position-updates '[{"DeviceId":"ExampleDevice","Position":
[-123.123,47.123],"SampleTime":"2021-11-30T21:47:25.149Z","Accuracy":
{"Horizontal":10.30},"PositionProperties":{"field1":"value1","field2":"value2"}}]'
```

使用亚马逊对亚马逊定位服务事件做出反应 EventBridge

Amazon EventBridge 是一种无服务器事件总线，它使用来自诸如 Amazon Location 之类的 AWS 服务的数据有效地将应用程序连接在一起。 EventBridge 从 Amazon Location 接收事件，并将这些数据路由到目标，例如 AWS Lambda。您可以设置路由规则来确定发送数据的目的地，以便构建能够实时响应的应用程序架构。

默认情况下，只有地理围栏EXIT事件（ENTER以及设备进入或离开地理围栏区域时的事件）才会发送到。 EventBridge 您也可以为跟踪器资源启用所有已筛选的位置更新事件。有关更多信息，请参阅 the section called “为跟踪器启用更新事件”。

有关更多信息，请参阅 Amazon EventBridge 用户指南中的事件和事件模式。

主题

- 为跟踪器启用更新事件
- 为 Amazon Location 创建活动规则
- 亚马逊定位服务（Amazon Location Service）EventBridge 的事件示例

为跟踪器启用更新事件

默认情况下，Amazon Location 仅ENTER向发送EXIT地理围栏事件。 EventBridge您可以启用所有筛选过的位置UPDATE事件，以便将跟踪器发送到该跟踪器 EventBridge。您可以在创建或更新跟踪器时执行此操作。

例如，要使用更新现有跟踪器 AWS CLI，您可以使用以下命令（使用您的跟踪器资源名称代替*MyTracker*）。

```
aws location update-tracker --tracker-name MyTracker --event-bridge-enabled
```

要关闭跟踪器的位置事件，您必须使用 API 或 Amazon Location Service 控制台。

为 Amazon Location 创建活动规则

您可以为每个事件总线创建多达 300 条规则， EventBridge 以配置为响应 Amazon Location 事件而采取的操作。

例如，您可以为地理围栏事件创建规则，在该规则中，当在地理围栏边界内检测到电话时，将发送推送通知。

为 Amazon Location 事件创建规则

使用以下值创建基于 Amazon 位置事件的 EventBridge 规则：

- 对于规则类型，选择具有事件模式的规则。
- 在事件模式框中，添加以下模式。

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Geofence Event"]
```

```
}
```

要创建跟踪器位置更新规则，您可以改用以下模式：

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Device Position Event"]
}
```

您可以选择通过添加 `detail` 标签来仅指定 ENTER 或 EXIT 事件（如果您的规则是针对跟踪器位置更新，则只有一个 `EventType`，因此无需对其进行筛选）：

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Geofence Event"],
  "detail": {
    "EventType": ["ENTER"]
  }
}
```

您也可以选择根据位置的属性或地理围栏进行筛选：

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Geofence Event"],
  "detail": {
    "EventType": ["ENTER"],
    "GeofenceProperties": {
      "Type": "LoadingDock"
    },
    "PositionProperties": {
      "VehicleType": "Truck"
    }
  }
}
```

- 对于选择目标，选择目标从 Amazon Location Service 收到事件时要执行的目标操作。

  例如，使用 Amazon Simple Notification Service (SNS) 主题在事件发生时发送电子邮件或短信。您首先需要使用 Amazon SNS 控制台创建 Amazon SNS 主题。想要了解更多信息，请参阅使用 Amazon SNS 发送通知。

> **Warning**
>
> 最佳做法是确认事件规则已成功应用，否则您的自动操作可能无法按预期启动。要验证您的事件规则，请为事件规则启动条件。例如，模拟进入地理围栏区域的设备。

您也可以从 Amazon Location 捕获所有事件，只需排除 `detail-type` 部分即可。例如：

```
{
  "source": [
    "aws.geo"
  ]
}
```

> **Note**
>
> 同一事件可以多次交付。您可以使用事件 ID 对收到的事件进行重复数据删除。

亚马逊定位服务（Amazon Location Service）EventBridge 的事件示例

以下是通过调用 `BatchUpdateDevicePosition` 启动的进入地理围栏的事件示例。

```
{
  "version": "0",
  "id": "aa11aa22-33a-4a4a-aaa5-example",
  "detail-type": "Location Geofence Event",
  "source": "aws.geo",
  "account": "636103698109",
  "time": "2020-11-10T23:43:37Z",
  "region": "eu-west-1",
  "resources": [
    "arn:aws:geo:eu-west-1:0123456789101:geofence-collection/GeofenceEvents-
GeofenceCollection_EXAMPLE",
    "arn:aws:geo:eu-west-1:0123456789101:tracker/Tracker_EXAMPLE"
  ],
  "detail": {
    "EventType": "ENTER",
    "GeofenceId": "polygon_14",
    "DeviceId": "Device1-EXAMPLE",
    "SampleTime": "2020-11-10T23:43:37.531Z",
```

```
    "Position": [
      -123.12390073297821,
      49.23433613216247
    ],
    "Accuracy": {
      "Horizontal": 15.3
    },
    "GeofenceProperties": {
      "ExampleKey1": "ExampleField1",
      "ExampleKey2": "ExampleField2"
    },
    "PositionProperties": {
      "ExampleKey1": "ExampleField1",
      "ExampleKey2": "ExampleField2"
    }
  }
}
```

以下是通过调用 BatchUpdateDevicePosition 启动的退出地理围栏的事件示例。

```
{
  "version": "0",
  "id": "aa11aa22-33a-4a4a-aaa5-example",
  "detail-type": "Location Geofence Event",
  "source": "aws.geo",
  "account": "123456789012",
  "time": "2020-11-10T23:41:44Z",
  "region": "eu-west-1",
  "resources": [
    "arn:aws:geo:eu-west-1:0123456789101:geofence-collection/GeofenceEvents-
GeofenceCollection_EXAMPLE",
    "arn:aws:geo:eu-west-1:0123456789101:tracker/Tracker_EXAMPLE"
  ],
  "detail": {
    "EventType": "EXIT",
    "GeofenceId": "polygon_10",
    "DeviceId": "Device1-EXAMPLE",
    "SampleTime": "2020-11-10T23:41:43.826Z",
    "Position": [
      -123.08569321875426,
      49.23766166742559
    ],
    "Accuracy": {
```

```
      "Horizontal": 15.3
    },
    "GeofenceProperties": {
      "ExampleKey1": "ExampleField1",
      "ExampleKey2": "ExampleField2"
    },
    "PositionProperties": {
      "ExampleKey1": "ExampleField1",
      "ExampleKey2": "ExampleField2"
    }
  }
}
```

以下是通过调用 BatchUpdateDevicePosition 启动的位置更新事件的示例。

```
{
  "version": "0",
  "id": "aa11aa22-33a-4a4a-aaa5-example",
  "detail-type": "Location Device Position Event",
  "source": "aws.geo",
  "account": "123456789012",
  "time": "2020-11-10T23:41:44Z",
  "region": "eu-west-1",
  "resources": [
    "arn:aws:geo:eu-west-1:0123456789101:tracker/Tracker_EXAMPLE"
  ],
  "detail": {
    "EventType": "UPDATE",
    "TrackerName": "tracker_2",
    "DeviceId": "Device1-EXAMPLE",
    "SampleTime": "2020-11-10T23:41:43.826Z",
    "ReceivedTime": "2020-11-10T23:41:39.235Z",
    "Position": [
      -123.08569321875426,
      49.23766166742559
    ],
    "Accuracy": {
      "Horizontal": 15.3
    },
    "PositionProperties": {
      "ExampleKey1": "ExampleField1",
      "ExampleKey2": "ExampleField2"
    }
```

```
  }
}
```

管理您的地理围栏收集资源

使用 Amazon Location 控制台 AWS CLI、或 Amazon 位置管理您的地理围栏收藏。 APIs

列出您的地理围栏集合资源

您可以使用 Amazon Location 控制台、或 Amazon Location 来查看您的地理围栏收藏列表： AWS CLI APIs

Console

使用 Amazon Location 控制台查看地理围栏集合列表

1. 打开 Amazon Location 控制台，网址为https://console.aws.amazon.com/location/。
2. 从左侧导航窗格中选择地理围栏集合。
3. 在我的地理围栏集合下查看您的地理围栏集合列表。

API

使用 Amazon 定位地理围 APIs栏中的ListGeofenceCollections操作。

以下示例是获取账户中地理围栏集合列表的 API 请求。 AWS

```
POST /geofencing/v0/list-collections
```

以下为 ListGeofenceCollections 响应示例：

```
{
    "Entries": [
    {
        "CollectionName": "ExampleCollection",
        "CreateTime": 2020-09-30T22:59:34.142Z,
        "Description": "string",
        "UpdateTime": 2020-09-30T23:59:34.142Z
    },
    "NextToken": "1234-5678-9012"
}
```

CLI

使用 `list-geofence-collections` 命令。

以下示例是 AWS CLI 获取账户中地理围栏集合列表的示例。 AWS

```
aws location list-geofence-collections
```

获取地理围栏集合的详细信息

您可以使用 Amazon Location 控制台、或 Amazon L AWS ocation 获取有关您账户中任何地理围栏收集资源的详细信息： AWS CLI APIs

Console

使用 Amazon Location 控制台查看地理围栏集合的详细信息

1. 打开 Amazon Location 控制台，网址为https://console.aws.amazon.com/location/。
2. 从左侧导航窗格中选择地理围栏集合。
3. 在我的地理围栏集合下，选择目标地理围栏集合的名称链接。

API

使用 Amazon 定位地理围 APIs栏中的DescribeGeofenceCollection操作。

以下示例是获取地理围栏收集详细信息的 API 请求。*ExampleCollection*

```
GET /geofencing/v0/collections/ExampleCollection
```

以下为 DescribeGeofenceCollection 响应示例：

```
{
    "CollectionArn": "arn:aws:geo:us-west-2:123456789012:geofence-collection/
GeofenceCollection",
    "CollectionName": "ExampleCollection",
    "CreateTime": 2020-09-30T22:59:34.142Z,
    "Description": "string",
    "KmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "Tags": {
        "Tag1" : "Value1"
    },
```

```
    "UpdateTime": 2020-09-30T23:59:34.142Z
}
```

CLI

使用 describe-geofence-collection 命令。

以下示例是 AWS CLI 获取地理围栏集合详细信息的示例。*ExampleCollection*

```
aws location describe-geofence-collection \
    --collection-name "ExampleCollection"
```

删除地理围栏集合

您可以使用 Amazon Location 控制台、或 Amazon Location 从您的 AWS 账户中 AWS CLI删除地理围栏集合。 APIs

Console

使用 Amazon Location 控制台删除地理围栏集合

> **Warning**
> 此操作将永久删除资源。

1. 打开 Amazon Location 控制台，网址为https://console.aws.amazon.com/location/。
2. 从左侧导航窗格中选择地理围栏集合。
3. 在我的地理围栏集合下，选择目标地理围栏集合。
4. 选择删除地理围栏集合。

API

使用 Amazon 营业地点的DeleteGeofenceCollection操作 APIs。

以下示例是删除地理围栏集合的 API 请求。*ExampleCollection*

```
DELETE /geofencing/v0/collections/ExampleCollection
```

以下为 DeleteGeofenceCollection 响应示例：

```
HTTP/1.1 200
```

CLI

使用 delete-geofence-collection 命令。

以下示例是删除 geofence 集合的 AWS CLI 命令。*ExampleCollection*

```
aws location delete-geofence-collection \
    --collection-name "ExampleCollection"
```

列出存储的地理围栏

您可以使用亚马逊定位控制台、 AWS CLI或亚马逊位置列出存储在指定地理围栏集合中的地理围栏。APIs

Console

使用 Amazon Location 控制台查看地理围栏列表

1. 打开 Amazon Location 控制台，网址为https://console.aws.amazon.com/location/。
2. 从左侧导航窗格中选择地理围栏集合。
3. 在我的地理围栏集合下，选择目标地理围栏集合的名称链接。
4. 在地理围栏下查看地理围栏集合中的地理围栏。

API

使用 Amazon 定位地理围 APIs栏中的ListGeofences操作。

以下示例是一个 API 请求，用于获取存储在 geofence 集合中的地理围栏列表。*ExampleCollection*

```
POST /geofencing/v0/collections/ExampleCollection/list-geofences
```

以下为 ListGeofences 响应示例：

```
{
```

```
   "Entries": [
      {
         "CreateTime": 2020-09-30T22:59:34.142Z,
         "GeofenceId": "geofence-1",
         "Geometry": {
             "Polygon": [
                 [-5.716667, -15.933333,
                 [-14.416667, -7.933333],
                 [-12.316667, -37.066667],
                 [-5.716667, -15.933333]
             ]
         },
         "Status": "ACTIVE",
         "UpdateTime": 2020-09-30T23:59:34.142Z
      }
   ],
   "NextToken": "1234-5678-9012"
}
```

CLI

使用 `list-geofences` 命令。

以下示例是获取存储在 geofence 集合中的地理围栏列表的示例。 AWS CLI *ExampleCollection*

```
aws location list-geofences \
    --collection-name "ExampleCollection"
```

获取地理围栏详细信息

您可以使用 Amazon Location 控制台 AWS CLI或亚马逊位置从地理围栏集合中获取特定地理围栏的详细信息，例如创建时间、更新时间、几何图形和状态。 APIs

Console

使用 Amazon Location 控制台查看地理围栏的状态

1. 打开 Amazon Location 控制台，网址为https://console.aws.amazon.com/location/。
2. 从左侧导航窗格中选择地理围栏集合。
3. 在我的地理围栏集合下，选择目标地理围栏集合的名称链接。

4. 在地理围栏 ，你可以查看地理围栏的状态。

API

使用 Amazon 定位地理围 APIs栏中的GetGeofence操作。

以下示例是从地理围栏集合中获取地理围栏详细信息的 API 请求。*ExampleCollection*

```
GET /geofencing/v0/collections/ExampleCollection/geofences/ExampleGeofence1
```

以下为 GetGeofence 响应示例：

```
{
   "CreateTime": 2020-09-30T22:59:34.142Z,
   "GeofenceId": "ExampleGeofence1",
   "Geometry": {
      "Polygon": [
          [-1,-1],
          [1,-1],
          [0,1],
          [-1,-1]
      ]
   },
   "Status": "ACTIVE",
   "UpdateTime": 2020-09-30T23:59:34.142Z
}
```

CLI

使用 get-geofence 命令。

以下示例是 AWS CLI 获取地理围栏集合详细信息的示例。*ExampleCollection*

```
aws location get-geofence \
    --collection-name "ExampleCollection" \
    --geofence-id "ExampleGeofence1"
```

删除地理围栏

您可以使用 Amazon Location 控制台、 AWS CLI或 Amazon 位置从地理围栏集合中删除地理围栏。APIs

Console

使用 Amazon Location 控制台删除地理围栏

> **Warning**
> 此操作将永久删除资源。

1. 打开 Amazon Location 控制台，网址为https://console.aws.amazon.com/location/。
2. 从左侧导航窗格中选择地理围栏集合。
3. 在我的地理围栏集合下，选择目标地理围栏集合的名称链接。
4. 在地理围栏下，选择目标地理围栏。
5. 选择删除地理围栏。

API

使用 Amazon 定位地理围 APIs栏中的BatchDeleteGeofence操作。

以下示例是从地理围栏集合中删除地理围栏的 API 请求。*ExampleCollection*

```
POST /geofencing/v0/collections/ExampleCollection/delete-geofences
Content-type: application/json

{
   "GeofenceIds": [ "ExampleGeofence11" ]
}
```

以下是 BatchDeleteGeofence 成功响应的示例。

```
HTTP/1.1 200
```

CLI

使用 batch-delete-geofence 命令。

以下示例是从 geofence 集合中删除 geofence 的 AWS CLI 命令。*ExampleCollection*

```
aws location batch-delete-geofence \
    --collection-name "ExampleCollection" \
    --geofence-ids "ExampleGeofence11"
```

管理成本和使用量

在您继续学习 Amazon Location Geofences 的过程中，重要的是要了解如何管理服务容量，确保遵守使用限制，并通过配额和 API 优化获得最佳结果。通过应用性能和准确性方面的最佳实践，您可以定制应用程序，以高效处理与地点相关的查询并最大限度地提高 API 请求。

主题

- 地理围栏定价
- 地理围栏配额和使用情况

地理围栏定价

有关追踪和地理围栏 APIs的定价信息，请参阅 Amazon Location S ervice 定价页面。

职位评估

您可以使用根据BatchEvaluateGeofences给定地理围栏集合中的地理围栏几何形状来评估设备位置。一个请求将针对单个地理围栏集合中的所有地理围栏评估多达十个设备位置。价格基于您的 API 请求中的设备头寸数量。评估的每台设备位置的单价基于每月的总使用量。有关单价和数量套餐的详细信息，请参阅 Amazon Location Service 定价页面。

您可以通过配置跟踪设备上的设备位置更新频率（也称为 ping 速率），并利用跟踪器上的过滤功能仅评估相关的位置更新，从而优化职位评估成本。

地理围栏管理和存储

您可以使用GetGeofence、PutGeofenceBatchPutGeofence、和BatchDeleteGeofence来管理地理围栏集合中的地理围栏。其价格取决于您 APIs 的 API 请求中的地理围栏数量。

地理围栏的存储空间将按月收费（仅适用于存储时间超过一个月的
地理围栏）。您还可以使用以下方式管理您的 Geofence 收藏
APIs：CreateGeofenceCollection、DeleteGeofenceCollection、DescribeGeofenceCollecti
和。ListGeofences其价格基 APIs 于 API 请求的数量。

Geofence 事件预测

您可以使用`ForecastGeofenceEvents`预测未来的地理围栏事件，如果设备继续以当前速度移动，则这些事件很可能在指定的时间范围内发生。价格基于 API 请求的数量。

地理围栏配额和使用情况

本主题概述了 Amazon Location Service Geofences 的速率限制和配额。

> **Note**
>
> 如果您需要更高的配额，可以使用服务配额控制台请求增加可调配额的配额。申请增加配额时，请选择您要求增加配额的区域，因为大多数配额都是特定于该 AWS 地区的。您最多可以为每个 API 请求两倍的默认限制。
> 对于超过默认限制两倍的请求，您的请求将提交支持请求。您也可以联系您的高级支持团队。
> 增加配额的请求不会直接收费，但是根据消耗的额外资源，更高的使用级别可能会导致服务成本增加。请参阅the section called “管理配额”了解更多信息。

Service Quotas 是每个 AWS 账户和每个 AWS 地区可以拥有的最大资源数量。Amazon Location Service 拒绝超过服务配额的其他请求。

资源

API 名称	默认	最大可调极限
每个账户的收款资源	1500	3000 如果您需要的不止于此，请申请增加配额或联系支持团队。
每个集合的地理围栏	50000	联系支持团队。

CRUD API

> **Note**
>
> 如果您需要更高的限额 APIs，请申请增加配额或联系支持团队。

API 名称	默认	最大可调极限
CreateGeofenceCollection	10	20
DeleteGeofenceCollection	10	20
DescribeGeofenceCollection	10	20
ListGeofenceCollections	10	20
UpdateGeofenceCollection	10	20

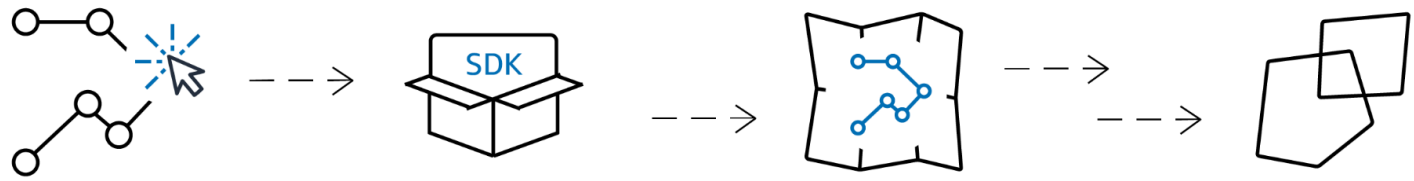
数据 API

> **Note**
>
> 如果您需要更高的限额 APIs，请申请增加配额或联系支持团队。

API 名称	默认	最大可调极限
BatchEvaluateGeofences	50	100
PutGeofence	50	100
BatchPutGeofence	50	100
ListGeofences	50	100
GetGeofence	50	100
BatchDeleteGeofence	50	100

其他使用限制

亚马逊 Location Service 追踪器



> **Note**
>
> Tracker 存储空间会自动使用 AWS 自有密钥进行加密。您可以使用您管理的 KMS 密钥添加另一层加密，以确保只有您才能访问您的数据。有关更多信息，请参阅 the section called “静态数据加密”。

跟踪器存储一组设备的位置更新。跟踪器可用于查询设备的当前位置或位置记录。它存储更新，但通过在存储更新之前筛选位置来减少存储空间和视觉噪音。

存储在您的跟踪器资源中的每个位置更新都可能包括一个位置精度指标，以及最多 3 个与您要存储的位置或设备相关的元数据字段。元数据以键值对的形式存储，可以存储速度、方向、轮胎压力或发动机温度等信息。

跟踪器位置筛选和查询本身很有用，但是与地理围栏配对时，跟踪器尤其有用。您可以将跟踪器链接到您的一个或多个地理围栏集合资源，并根据这些集合中的地理围栏自动评估位置更新。正确使用筛选还可以大大降低地理围栏评估的成本。

1. 首先，在 AWS 账户中创建跟踪器资源。
2. 接下来，决定如何向跟踪器资源发送位置更新。用于AWS SDKs将跟踪功能集成到您的移动应用程序中。或者，您可以按照使用 MQTT 进行跟踪的 step-by-step说明来使用 MQTT。
3. 现在，您可以使用跟踪器资源来记录位置记录并在地图上将其可视化。
4. 您还可以将您的跟踪器资源链接到一个或多个地理围栏集合，这样发送到跟踪器资源的每个位置更新都会根据所有链接的地理围栏集合中的所有地理围栏自动进行评估。您可以在 Amazon Location 控制台的跟踪器资源详情页面上链接资源，也可以使用 Amazon Location 跟踪器 API 来链接资源。
5. 然后，您可以使用诸如Amazon CloudWatch 和之类的服务来集成监控 AWS CloudTrail。有关更多信息，请参阅 the section called “使用 Amazon 进行监控 CloudWatch” 和 the section called “使用监控和登录 AWS CloudTrail”。

特征

- 位置筛选-跟踪器可以自动筛选发送给他们的头寸。您可能出于多个原因想要筛选掉某些设备位置更新。如果您的系统每分钟左右发送一次报告，则可能需要按时间筛选设备，仅每 30 秒存储和评估一次位置。即使您的监控频率更高，您也可能需要过滤位置更新，以清除与 GPS 硬件和位置报告相关的固有噪音。它们的精度并不是 100% 完美，因此即使是静止的设备也看似在轻微移动。在低速下，这种抖动会导致凌乱的视觉效果，如果设备靠近地理围栏的边缘，则可能导致错误的进入和退出事件。

  位置筛选的工作原理是跟踪器收到位置更新，从而减少设备路径中的视觉噪音（抖动），减少错误的地理围栏进入和退出事件的数量，并通过减少存储的位置更新和触发的地理围栏评估的数量来帮助管理成本。

  跟踪器提供三种位置筛选选项，以帮助管理成本并减少位置更新中的抖动。

- 基于精度——与任何提供精度测量的设备一起使用。大多数 GPS 和移动设备都提供此信息。

  每次位置测量的精度受许多环境因素的影响，包括 GPS 卫星接收、景观 WiFi以及蓝牙设备的距离。大多数设备，包括大多数移动设备，都可以提供测量精度的估计值以及测量结果。通过 `AccuracyBased` 筛选，如果设备移动小于测量的精度，Amazon Location 将忽略位置更新。

  例如，如果一个设备两次连续更新的精度范围分别为 5 米和 10 米，则该设备的移动距离小于 15 米时 Amazon Location 忽略第二次更新。Amazon Location 既不会根据地理围栏评估被忽略的更新，也不会存储这些更新。

  如果未提供精度，则将其视为零，并且测量结果被视为完全准确，并且不会对更新进行筛选。

  > **Note**
  >
  > 您可以使用基于精度的筛选来移除所有筛选。如果您选择基于精度的筛选，但将所有精度数据覆盖为零，或者完全省略准确性，那么 Amazon Location 将不会筛选出任何更新。

- 基于距离 — 当您的设备不提供精度测量值，但您仍希望利用过滤功能来减少抖动并管理成本时使用。

  `DistanceBased`过滤会忽略设备移动小于 30 米（98.4 英尺）的位置更新。当您使用 `DistanceBased` 位置筛选时，Amazon Location 既不会根据地理围栏评估这些被忽略的更新，也不会存储更新。

大多数移动设备的精度，包括 iOS 和 Android 设备的平均精度，都在 15 米以内。在大多数应用程序中，`DistanceBased` 筛选可以减少在地图上显示设备轨迹时位置不准确的影响，以及设备靠近地理围栏边界时连续多次进入和退出事件的反弹效果。它还可以减少对照链接的地理围栏进行评估或检索设备位置的调用，从而帮助降低应用程序的成本。

如果你想进行筛选，但你的设备不提供精度测量，或者你想筛选掉比基于精度的更新数量更多的更新，那么基于距离的筛选非常有用。

- 基于时间——（默认）当您的设备非常频繁地发送位置更新（每 30 秒钟超过一次），并且您希望在不存储每个更新的情况下实现近乎实时的地理围栏评估时使用。

  在 `TimeBased` 筛选中，每个位置更新根据链接的地理围栏集合进行评估，但并非每个位置更新都会存储。如果更新频率超过 30 秒，则每 30 秒仅为每个唯一的设备 ID 存储一次更新。

  当您想要存储更少的位置，但希望根据链接的地理围栏集合评估每个位置更新时，基于时间的筛选特别有用。

> **Note**
>
> 在决定筛选方法和位置更新的频率时，请注意追踪应用程序的成本。您需要为每次位置更新付费，并根据每个链接的地理围栏集合评估位置更新一次付费。
> 例如，使用基于时间的筛选时，如果您的跟踪链接到两个地理围栏集合，则每次位置更新都将计为一个位置更新请求和两次地理围栏收集评估。如果您报告设备每 5 秒更新一次位置并使用基于时间的筛选，则每台设备将按每小时 720 次位置更新和 1,440 次地理围栏评估计费。

Amazon Location Service 追踪器的用例

以下是 Amazon Location Service 追踪器的一些常见用途。

使用带有地理围栏的追踪器

与地理围栏配对时，跟踪器可提供更多功能。您可以通过 Amazon Location 控制台或 API 将跟踪器与地理围栏集合进行链接，以自动评估跟踪器位置。每次跟踪器收到更新的位置时，系统都会根据集合中的每个地理围栏评估该位置，并在 Amazon 中生成相应`EXIT`的事件`ENTER`和事件。 EventBridge您还可以对跟踪器应用筛选，根据筛选情况，您可以通过仅评估有意义的位置更新来降低地理围栏评估的成本。

如果您在跟踪器已经收到一些位置更新后将其与地理围栏集合相链接，则链接后的第一个位置更新将被视为地理围栏评估的初始更新。如果它在地理围栏内，您将收到一个 ENTER 事件。如果它不在任何地理围栏内，则无论之前的状态如何，您都不会收到任何 EXIT 事件。

改善现场服务运营

通过实时跟踪，随时掌握移动办公人员的脉搏。在客户地点和服务区周围设置地理围栏，以便在员工到达和离开时收到警报。使用位置数据来优化日程安排，派遣最近的可用技术人员并缩短响应时间。让您的现场团队（例如管道或暖通空调维修业务）提高工作效率，同时增强客户体验。

监控和控制关键资产

利用 Amazon Location Service 来跟踪您的贵重设备、库存和其他移动资产的实时位置和状态。设置地理围栏以接收有关未经授权的移动或搬迁的警报，从而增强安全性和合规性。利用这种位置可视性来提高资产利用率，优化维护计划，并确保您的关键资源始终得到考虑。始终精确监控您的重型机械、IT 硬件或零售库存，减少损失，并做出更明智的运营决策。

提高供应链可见性

利用 Amazon Location Service 来跟踪整个供应链中的货件和配送。在配送中心、商店和其他关键设施周围定义地理围栏，以监控库存和资产的流动。使用实时位置数据来改善库存管理、优化物流计划并提供卓越的客户体验。深入 end-to-end了解您的供应链运营，识别瓶颈，并做出以数据为导向的决策，从而提高效率和响应能力。

基于位置的营销

释放位置数据的力量，增强您的地理营销工作。使用 Amazon Location Service 在竞争对手的地点、活动和高流量区域周围设置虚拟边界。当客户进入这些地理围栏区域时，触发个性化广告、优惠和通知。分析客流量模式以优化广告投放位置，发现新营业地点的主要网站。监控客户在自己的地理围栏空间内的活动，以更深入地了解浏览行为和. path-to-purchase 将实时位置跟踪与精确的地理围栏相结合，提供超有针对性的情境式互动，从而推动现实世界中的销售和忠诚度。

追踪器概念

本节详细介绍了常见的跟踪器概念。

常见的亚马逊 Location Service 追踪器术语

跟踪器资源

一种从设备接收位置更新的 AWS 资源。跟踪器资源为位置查询提供支持，例如当前和历史设备位置。将跟踪器资源链接到地理围栏集合会根据链接的地理围栏集合中的所有地理围栏自动评估位置更新。

跟踪位置数据

跟踪器资源会随着时间的推移存储有关您的设备的信息。这些信息包括一系列位置更新，其中每次更新都包括地点、时间和可选的元数据。元数据可以包括位置的精度，以及最多三个键值对，以帮助您跟踪每个位置的关键信息，例如正在跟踪的车辆的速度、方向、轮胎压力、剩余燃油或发动机温度。跟踪器会将设备位置记录保存 30 天。

位置筛选

在存储更新或根据地理围栏评估更新之前，位置筛选可以筛选掉不提供有价值信息的位置更新，从而帮助您控制成本并提高跟踪应用程序的质量。

您可以选择 `AccuracyBased`、`DistanceBased` 或 `TimeBased` 筛选。默认情况下，位置筛选设置为 `TimeBased`。

您可以在创建或更新跟踪器资源时配置位置筛选。

RFC 3339 时间戳格式

Amazon Location Service 追踪器使用 RFC 3339 格式，该格式遵循国际标准化组织 (ISO) 8601 格式的日期和时间。

格式为 “YYYYY-MM--: mm: ss.sssz+ 00:00”：DDThh

- `YYYY-MM-DD` – 表示日期格式。
- `T` – 表明接下来将是时间值。
- `hh:mm:ss.sss` – 以 24 小时制表示时间。
- `Z` – 表示使用的时区是 UTC，后面可以有与 UTC 时区的偏差。
- `+00:00` – （可选）表示与 UTC 时区的偏差。例如，+ 01:00 表示 UTC + 1 小时。

示例

2020 年 7 月 2 日下午 12:15:20，与 UTC 的时区相比，再调整 1 个小时。

```
2020-07-02T12:15:20.000Z+01:00
```

开始使用 Amazon Location Service 追踪器

本节提供了有关创建和使用带有 Amazon Location 位置的跟踪器的综合指南。跟踪器允许您存储、处理和评估设备位置，同时筛选位置更新以减少噪音和管理成本。借助高级位置筛选选项、对关联地理围栏集合的支持以及与 EventBridge IoT Core 等 AWS 服务的集成，跟踪器可根据您的特定需求提供精确的实时跟踪和地理围栏应用程序。

主题

- 创建跟踪器
- 对您的请求进行身份验证
- 使用设备位置更新跟踪器
- 通过跟踪器获取设备的位置记录
- 列出您的设备位置

创建跟踪器

创建跟踪器资源以存储和处理来自您设备的位置更新。您可以使用 Amazon Location Service 控制台 AWS CLI、或亚马逊位置 APIs。

存储在您的跟踪器资源中的每个位置更新都可能包括一个位置精度指标，以及最多三个与您要存储的位置或设备相关的元数据字段。元数据以键值对的形式存储，可以存储速度、方向、轮胎压力或发动机温度等信息。

跟踪器在收到的位置更新时对其进行过滤。这可以减少设备路径中的视觉噪音（称为抖动），并减少错误的地理围栏进入和退出事件的数量。这还可以减少启动的地理围栏评估的数量，从而帮助管理成本。

跟踪器提供三种位置筛选选项，以帮助管理成本并减少位置更新中的抖动。

- 基于精度——与任何提供精度测量的设备一起使用。大多数移动设备都提供此信息。每次位置测量的准确性受许多环境因素的影响，包括 GPS 卫星接收、场景以及 Wi-Fi 和蓝牙设备的距离。大多数设备，包括大多数移动设备，都可以提供测量精度的估计值以及测量结果。通过 `AccuracyBased` 筛选，如果设备移动小于测量的精度，Amazon Location 将忽略位置更新。例如，如果一个设备两次连续更新的精度范围分别为 5 米和 10 米，则该设备的移动距离小于 15 米时 Amazon Location 忽略第二次更新。Amazon Location 既不会根据地理围栏评估被忽略的更新，也不会存储这些更新。

如果未提供精度，则将其视为零，并且测量结果被视为完全准确。

> **Note**
>
> 您也可以使用基于精度的筛选来移除所有筛选。如果您选择基于精度的筛选，但将所有精度数据覆盖为零，或者完全省略准确性，那么 Amazon Location 将不会筛选出任何更新。

- 基于距离——当您的设备不提供精度测量值，但您仍希望利用筛选功能来减少抖动并管理成本时使用。`DistanceBased` 筛选忽略设备移动小于 30 米（98.4 英尺）的位置更新。当您使用 `DistanceBased` 位置筛选时，Amazon Location 既不会根据地理围栏评估这些被忽略的更新，也不会存储更新。

  大多数移动设备的精度，包括 iOS 和 Android 设备的平均精度，都在 15 米以内。在大多数应用程序中，`DistanceBased` 筛选可以减少在地图上显示设备轨迹时位置不准确的影响，以及设备靠近地理围栏边界时连续多次进入和退出事件的反弹效果。它还可以减少对照链接的地理围栏进行评估或检索设备位置的调用，从而帮助降低应用程序的成本。

- 基于时间——（默认）当您的设备非常频繁地发送位置更新（每 30 秒钟超过一次），并且您希望在不存储每个更新的情况下实现近乎实时的地理围栏评估时使用。在 `TimeBased` 筛选中，每个位置更新根据链接的地理围栏集合进行评估，但并非每个位置更新都会存储。如果更新频率超过 30 秒，则每 30 秒仅为每个唯一的设备 ID 存储一次更新。

> **Note**
>
> 在决定筛选方法和位置更新的频率时，请注意追踪应用程序的成本。您需要为每次位置更新付费，并根据每个链接的地理围栏集合评估位置更新一次付费。例如，使用基于时间的筛选时，如果您的跟踪链接到两个地理围栏集合，则每次位置更新都将计为一个位置更新请求和两次地理围栏收集评估。如果您报告设备每 5 秒更新一次位置并使用基于时间的筛选，则每台设备将按每小时 720 次位置更新和 1,440 次地理围栏评估计费。
>
> 您的计费不受每个集合中地理围栏数量的影响。由于每个地理围栏集合可能包含多达 50,000 个地理围栏，因此您可能需要尽可能将地理围栏合并成更少的集合，以降低地理围栏评估的成本。

默认情况下，每次被跟踪的设备进入或退出关联的地理围栏时，您都会收到 EventBridge 事件。有关更多信息，请参阅 将跟踪器关联到地理围栏集合。

您可以为跟踪器资源的所有筛选位置更新启用事件。有关更多信息，请参阅 the section called “为跟踪器启用更新事件”。

> **Note**
>
> 如果您希望使用自己的 AWS KMS 客户托管密钥加密数据，则默认情况下将禁用 “边界多边形查询” 功能。这是因为使用此边界多边形查询功能，您的设备位置表示不会使用您的 AWS KMS 托管密钥进行加密。但是，设备的确切位置仍使用您的托管密钥进行加密。
> 在创建或更新跟踪器时，您可以通过将 `KmsKeyEnableGeospatialQueries` 参数设置为 true 来选择使用“边界多边形查询”功能。

Console

使用 Amazon Location 控制台创建跟踪器

1. 打开 Amazon Location Service 控制台，网址为https://console.aws.amazon.com/location/。
2. 在左侧导航窗格中，选择跟踪器。
3. 选择创建跟踪器。
4. 填写以下字段：
   - 名称——输入唯一名称。例如 *`ExampleTracker`*。最多 100 个字符。有效条目包括：字母数字字符、连字符、句号和下划线。
   - 描述——输入可选描述。
5. 在位置筛选下，选择最适合您打算如何使用跟踪器资源的选项。如果未设置位置筛选，则默认设置为 `TimeBased`。想要了解更多信息，请参阅本指南 跟踪器 和 Amazon Location Service Trackers API 参考中的 `PositionFiltering`。
6. （可选）在 Tags (标签) 下，输入标签 Key (键) 和 Value (值)。这会为您的新地理围栏集合添加一个标签。有关更多信息，请参阅 the section called “如何使用标签”。
7. （可选）在“客户托管密钥加密”下，您可以选择添加客户托管密钥。这会添加一个对称的客户托管密钥，您可以通过默认 AWS 拥有的加密来创建、拥有和管理该密钥。想要了解更多信息，请参阅加密静态数据。
8. （可选）在下方 KmsKeyEnableGeospatialQueries，您可以选择启用地理空间查询。这允许您使用边界多边形查询功能，同时使用客户 AWS KMS 托管密钥对数据进行加密。

> **Note**
>
> 当您使用边界多边形查询功能时，您的设备位置表示不会使用您的 AWS KMS 托管密钥进行加密。但是，设备的确切位置仍使用您的托管密钥进行加密。

9. （可选）在EventBridge 配置下，您可以选择为筛选的职位更新启用 EventBridge 事件。每当该跟踪器中设备的位置更新符合位置筛选评估时，都会发送一个事件。
10. 选择创建跟踪器。

API

使用 Amazon 位置创建追踪器 APIs

使用 Amazon 位置追踪器 APIs中的`CreateTracker`操作。

以下示例使用 API 请求创建名为的跟踪器*`ExampleTracker`*。跟踪器资源与客户管理的 AWS KMS 密钥相关联，用于加密客户数据，并且不在中启用位置更新 EventBridge。

```
POST /tracking/v0/trackers
Content-type: application/json

{

   "TrackerName": "ExampleTracker",
   "Description": "string",
   "KmsKeyEnableGeospatialQueries": false,
   "EventBridgeEnabled": false,
   "KmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
   "PositionFiltering": "AccuracyBased",
   "Tags": {
      "string" : "string"
   }
}
```

在 **`KmsKeyEnableGeospatialQueries`** 启用状态下创建跟踪器

以下示例将参数 `KmsKeyEnableGeospatialQueries` 设置为 true。这允许您使用边界多边形查询功能，同时使用客户 AWS KMS 管理的密钥对数据进行加密。

有关使用“边界面查询”功能的信息，请参阅 the section called “列出您的设备位置”

> **Note**
>
> 当您使用边界多边形查询功能时，您的设备位置表示不会使用您的 AWS KMS 托管密钥进行加密。但是，设备的确切位置仍使用您的托管密钥进行加密。

```
POST /tracking/v0/trackers
Content-type: application/json

{

   "TrackerName": "ExampleTracker",
   "Description": "string",
   "KmsKeyEnableGeospatialQueries": true,
   "EventBridgeEnabled": false,
   "KmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
   "PositionFiltering": "AccuracyBased",
   "Tags": {
      "string" : "string"
   }
}
```

AWS CLI

使用 AWS CLI 命令创建跟踪器

使用 `create-tracker` 命令。

以下示例使用创建名 AWS CLI 为的跟踪器*ExampleTracker*。跟踪器资源与客户管理的 AWS KMS 密钥相关联，用于加密客户数据，并且不在中启用位置更新 EventBridge。

```
aws location \
  create-tracker \
  --tracker-name "ExampleTracker" \
  --position-filtering "AccuracyBased" \
  --event-bridge-enabled false \
  --kms-key-enable-geospatial-queries false \
  --kms-key-id "1234abcd-12ab-34cd-56ef-1234567890ab"
```

在 **KmsKeyEnableGeospatialQueries** 启用状态下创建跟踪器

以下示例将参数 KmsKeyEnableGeospatialQueries 设置为 true。这允许您使用边界多边形查询功能，同时使用客户 AWS KMS 管理的密钥对数据进行加密。

有关使用“边界面查询”功能的信息，请参阅 the section called “列出您的设备位置”

> **Note**
>
> 当您使用边界多边形查询功能时，您的设备位置表示不会使用您的 AWS KMS 托管密钥进行加密。但是，设备的确切位置仍使用您的托管密钥进行加密。

```
aws location \
  create-tracker \
  --tracker-name "ExampleTracker" \
  --position-filtering "AccuracyBased" \
  --event-bridge-enabled false \
  --kms-key-enable-geospatial-queries true \
  --kms-key-id "1234abcd-12ab-34cd-56ef-1234567890ab"
```

> **Note**
>
> 计费取决于您的使用情况。您可能会因为使用其他 AWS 服务而产生费用。想要了解更多信息，请参阅 Amazon Location Service 定价。

创建跟踪链接后，您可以通过选择编辑跟踪链接来编辑描述、位置筛选和EventBridge 配置。

对您的请求进行身份验证

创建跟踪器资源并准备好开始根据地理围栏评估设备位置后，请选择如何对请求进行身份验证：

- 要探索访问服务的方式，请参阅身份验证。
- 如果您想通过未经身份验证的请求发布设备位置，则可能需要使用 Amazon Cognito。

  示例

  以下示例展示了如何使用 Amazon Cognito 身份池进行授权、使用 AWS JavaScript SDK v3 和亚马逊位置。the section called “Web”

```
import { LocationClient, BatchUpdateDevicePositionCommand } from "@aws-sdk/client-
location";
import { withIdentityPoolId } from "@aws/amazon-location-utilities-auth-helper";

// Unauthenticated identity pool you created
const identityPoolId = "us-east-1:1234abcd-5678-9012-abcd-sample-id";

// Create an authentication helper instance using credentials from Cognito
const authHelper = await withIdentityPoolId(identityPoolId);

const client = new LocationClient({
  region: "us-east-1", // The region containing both the identity pool and tracker
 resource
  ...authHelper.getLocationClientConfig(), // Provides configuration required to make
 requests to Amazon Location
});

const input = {
  TrackerName: "ExampleTracker",
  Updates: [
    {
      DeviceId: "ExampleDevice-1",
      Position: [-123.4567, 45.6789],
      SampleTime: new Date("2020-10-02T19:09:07.327Z"),
    },
    {
      DeviceId: "ExampleDevice-2",
      Position: [-123.123, 45.123],
      SampleTime: new Date("2020-10-02T19:10:32Z"),
    },
  ],
};

const command = new BatchUpdateDevicePositionCommand(input);

// Send device position updates
const response = await client.send(command);
```

使用设备位置更新跟踪器

要追踪您的设备，您可以将设备位置更新发布到跟踪器。您稍后可以从您的跟踪器资源中检索这些设备位置或设备位置历史记录。

每次位置更新都必须包括设备 ID、时间戳和位置。您可以选择添加其他元数据，包括精度和最多 3 个键值对供您自己使用。

如果您的跟踪器关联到一个或多个地理围栏集合，则系统将根据这些地理围栏对更新进行评估（遵循您为跟踪器指定的过滤规则）。如果设备突破了地理围栏区域（从该区域内部移动到外部，反之亦然），您将在其中收到事件。 EventBridge这些 ENTER 或 EXIT 事件包括位置更新详细信息，包括设备 ID、时间戳和任何相关的元数据。

> **Note**
>
> 有关位置筛选的更多信息，请参阅 the section called “创建跟踪器”。
> 有关地理围栏事件的更多信息，请参阅 the section called “通过以下方式对事件做出反应 EventBridge”。

使用以下任一方法发送设备更新：

- 向 Io@@ T Core 资源发送 MQT AWS T 更新并将其链接到您的跟踪器资源。
- 使用亚马逊位置追踪器 API AWS CLI、或亚马逊位置来发送位置 APIs更新。您可以使用 APIs 从 iOS 或 Android 应用程序中调用。AWS SDKs

API

使用 Amazon 位置发送职位更新 APIs

使用 Amazon 位置追踪器 APIs中的`BatchUpdateDevicePosition`操作。

以下示例使用 API 请求向跟踪链接发布设备位置更新*`ExampleTracker`*。*`ExampleDevice`*

```
POST /tracking/v0/trackers/ExampleTracker/positions
Content-type: application/json
{
 "Updates": [
    {
    "DeviceId": "1",
    "Position": [
```

```
-123.12245146162303, 49.27521118043802
 ],
 "SampleTime": "2022-10-24T19:09:07.327Z",
  "PositionProperties": {
         "name" : "device1"
      },
      "Accuracy": {
         "Horizontal": 10
      }
 },

 {
 "DeviceId": "2",
 "Position": [
-123.1230104928471, 49.27752402723152
 ],
 "SampleTime": "2022-10-02T19:09:07.327Z"
 },
 {
 "DeviceId": "3",
 "Position": [
 -123.12325592118916, 49.27340530543111
 ],
 "SampleTime": "2022-10-02T19:09:07.327Z"
 },
 {
 "DeviceId": "4",
 "Position": [
 -123.11958813096311, 49.27774641063121
 ],
 "SampleTime": "2022-10-02T19:09:07.327Z"
 },
 {
 "DeviceId": "5",
 "Position": [
 -123.1277418058896, 49.2765989015285
 ],
 "SampleTime": "2022-10-02T19:09:07.327Z"
 },
 {
 "DeviceId": "6",
 "Position": [
-123.11964267059481, 49.274188155916534
 ],
```

```
    "SampleTime": "2022-10-02T19:09:07.327Z"
    }
    ]
}
```

AWS CLI

使用 AWS CLI 命令发送位置更新

使用 batch-update-device-position 命令。

以下示例使用 AWS CLI 向跟踪器发布*ExampleDevice-1*和*ExampleDevice-2*向跟踪器发布设备位置更新*ExampleTracker*。

```
aws location batch-update-device-position \
--tracker-name ExampleTracker \
--updates '[{"DeviceId":"ExampleDevice-1","Position":
[-123.123,47.123],"SampleTime":"2021-11-30T21:47:25.149Z"},
{"DeviceId":"ExampleDevice-2","Position":
[-123.123,47.123],"SampleTime":"2021-11-30T21:47:25.149Z","Accuracy":
{"Horizontal":10.30},"PositionProperties":{"field1":"value1","field2":"value2"}}]'
```

通过跟踪器获取设备的位置记录

您的 Amazon Location 跟踪器资源会将所有被追踪设备的位置记录保存 30 天。您可以从跟踪器资源中检索设备位置记录，包括所有关联的元数据。以下示例使用或 AWS CLI Amazon 地点 APIs。

API

使用 Amazon 位置从追踪器获取设备位置记录 APIs

使用 Amazon 位置追踪器 APIs中的GetDevicePositionHistory操作。

以下示例使用 API URI 请求*ExampleDevice*从名为的跟踪器获取设备位置历史记录，该跟踪器*ExampleTracker*起始于19:05:07（包含），结尾为19:20:07（独家）2020-10-02。

```
POST /tracking/v0/trackers/ExampleTracker/devices/ExampleDevice/list-positions
Content-type: application/json
{
  "StartTimeInclusive": "2020-10-02T19:05:07.327Z",
  "EndTimeExclusive": "2020-10-02T19:20:07.327Z"
}
```

AWS CLI

使用 AWS CLI 命令从追踪器获取设备位置记录

使用 get-device-position-history 命令。

以下示例使用*ExampleDevice*从名为 “*ExampleTracker*起始于19:05:07（包含）”、“独家” 结束于19:20:07（独家）的跟踪器获取设备位置历史记录2020-10-02。 AWS CLI

```
aws location \
    get-device-position-history \
        --device-id "ExampleDevice" \
        --start-time-inclusive "2020-10-02T19:05:07.327Z" \
        --end-time-exclusive "2020-10-02T19:20:07.327Z" \
        --tracker-name "ExampleTracker"
```

列出您的设备位置

您可以使用 ListDevicePositions API 或 Amazon 位置查看追踪器的设备位置 APIs列表。 AWS CLI当您调用 ListDevicePositions API 时，系统会返回与给定跟踪器关联的所有设备的最新位置列表。默认情况下，此 API 会在给定跟踪器的每页结果中返回 100 个最新的设备位置。要仅返回特定区域内的设备，请使用 FilterGeometry 参数创建“边界多边形查询”。这样，当你调用时 ListDevicePositions，只会返回多边形内的设备。

> **Note**
>
> 如果您希望使用自己的 AWS KMS 客户托管密钥加密数据，则默认情况下将禁用 “边界多边形查询” 功能。这是因为使用此功能，您的设备位置表示不会使用您的 AWS KMS 托管密钥进行加密。但是，设备的确切位置仍使用您的托管密钥进行加密。
> 您可以选择加入“边界多边形查询”功能。这是通过在创建或更新跟踪器时将 KmsKeyEnableGeospatialQueries 参数设置为 true 来完成的。

API

使用 Amazon 位置追踪器 APIs中的ListDevicePositions操作。

以下示例是使用可选参数 FilterGeometry 获取多边形区域中设备位置列表的 API 请求。该示例返回 Polygon 数组定义的区域中存在的 3 个设备位置。

```
POST /tracking/v0/trackers/TrackerName/list-positions HTTP/1.1
Content-type: application/json

{
   "FilterGeometry": {
        "Polygon": [
          [
            [
              -123.12003339442259,
              49.27425121147397
            ],
            [
              -123.1176984148229,
              49.277063620879744
            ],
            [
              -123.12389509145294,
              49.277954183760926
            ],
            [
              -123.12755921328647,
              49.27554025235713
            ],
            [
              -123.12330236586217,
              49.27211836076236
            ],
            [
              -123.12003339442259,
              49.27425121147397
            ]
          ]
        ]
    },
   "MaxResults": 3,
   "NextToken": "1234-5678-9012"
}
```

以下为 ListDevicePositions 响应示例：

```
{
    "Entries": [
        {
```

```
            "DeviceId": "1",
            "SampleTime": "2022-10-24T19:09:07.327Z",
            "Position": [
                -123.12245146162303,
                49.27521118043802
            ],
            "Accuracy": {
                "Horizontal": 10
            },
            "PositionProperties": {
                "name": "device1"
            }
        },
        {
            "DeviceId": "3",
            "SampleTime": "2022-10-02T19:09:07.327Z",
            "Position": [
                -123.12325592118916,
                49.27340530543111
            ]
        },
        {
            "DeviceId": "2",
            "SampleTime": "2022-10-02T19:09:07.327Z",
            "Position": [
                -123.1230104928471,
                49.27752402723152
            ]
        }
    ],
   "NextToken": "1234-5678-9012"
}
```

CLI

使用 list-trackers 命令。

以下示例是 AWS CLI 获取多边形区域中的设备列表的示例。

```
aws location list-device-positions TODO: add arguments add props for filter geo
```

如何使用 Amazon Location Service 追踪器

本节提供有关使用 Amazon 位置追踪器的说明。了解如何使用物联网和 MQT AWS T 验证设备位置、将跟踪器关联到地理围栏集合以及跟踪位置。此外，请学习如何有效管理您的跟踪器，以支持基于位置的应用程序并确保准确、实时的跟踪。

主题

- 验证设备位置
- 将跟踪器关联到地理围栏集合
- 使用 Amazon Location Service AWS IoT 和 MQTT 进行跟踪
- 管理您的亚马逊 Location Service 追踪器

验证设备位置

要检查设备位置的完整性，请使用 VerifyDevicePositionAPI。此 API 通过评估诸如设备的蜂窝信号、Wi-Fi 接入点、IPv4 地址以及是否正在使用代理等属性来返回有关设备位置完整性的信息。

先决条件

在能够使用列 APIs 出的设备验证之前，请确保满足以下先决条件：

- 您已经为要查看的一个或多个设备创建跟踪器。有关更多信息，请参阅 开始使用 Amazon Location Service 追踪器。

以下示例显示了对 Amazon 定位 VerifyDevicePositionAPI 的请求。

API

使用 Amazon 位置验证设备位置 APIs

使用 Amazon 位置追踪中的VerifyDevicePosition操作 APIs。

以下示例显示的是对评估设备位置完整性的 API 请求。用您自己的设备替换这些值 IDs。

```
POST /tracking/v0/trackers/TrackerName/positions/verify HTTP/1.1
Content-type: application/json

{
   "DeviceState": {
```

```
      "Accuracy": {
         "Horizontal": number
      },
      "CellSignals": {
         "LteCellDetails": [
            {
               "CellId": number,
               "LocalId": {
                  "Earfcn": number,
                  "Pci": number
               },
               "Mcc": number,
               "Mnc": number,
               "NetworkMeasurements": [
                  {
                     "CellId": number,
                     "Earfcn": number,
                     "Pci": number,
                     "Rsrp": number,
                     "Rsrq": number
                  }
               ],
               "NrCapable": boolean,
               "Rsrp": number,
               "Rsrq": number,
               "Tac": number,
               "TimingAdvance": number
            }
         ]
      },
      "DeviceId": "ExampleDevice",
      "Ipv4Address": "string",
      "Position": [ number ],
      "SampleTime": "string",
      "WiFiAccessPoints": [
         {
            "MacAddress": "string",
            "Rss": number
         }
      ]
   },
   "DistanceUnit": "string"
}
```

> **Note**
>
> 完整性 SDK 提供了与设备验证相关的增强功能，可通过请求使用该功能。要获取对该 SDK 的访问权限，请联系销售支持。

将跟踪器关联到地理围栏集合

现在你已经有了地理围栏集合和跟踪器，你可以将它们链接在一起，这样就可以根据你的所有地理围栏自动评估位置更新。如果您不想评估所有位置更新，或者如果您不想将某些位置存储在跟踪器资源中，则可以按需根据地理围栏评估设备位置。

根据地理围栏评估设备位置时，会生成事件。您可以为这些事件设置操作。有关您可以为地理围栏事件设置的操作的更多信息，请参阅通过亚马逊对亚马逊定位服务事件做出反应。 EventBridge

Amazon Location 事件包括生成该事件的设备位置更新的属性以及进入或退出的地理围栏的一些属性。有关地理围栏事件中包含的数据的更多信息，请参阅 the section called “事件示例”。

以下示例使用控制台、或 Amazon 位置将跟踪器资源链接到地理围栏集合。 AWS CLI APIs

Console

使用 Amazon Location Service 控制台将跟踪器资源关联到地理围栏集合

1. 打开 Amazon Location Service 控制台，网址为https://console.aws.amazon.com/location/。
2. 在左侧导航窗格中，选择跟踪器。
3. 在设备跟踪器下，选择目标跟踪器的名称链接。
4. 在链接的地理围栏集合下，选择链接地理围栏集合。
5. 在链接的地理围栏集合窗口中，从下拉菜单中选择一个地理围栏集合。
6. 选择链接。

在您关联跟踪器资源后，它将处于活动状态。

API

使用 Amazon 位置将追踪器资源关联到地理围栏集合 APIs

使用 Amazon 位置追踪器 APIs中的`AsssociateTrackerConsumer`操作。

以下示例使用一个 API 请求，该请求使用其亚马逊资源名称 (ARN) *ExampleTracker* 与地理围栏集合相关联。

```
POST /tracking/v0/trackers/ExampleTracker/consumers
Content-type: application/json

{
   "ConsumerArn": "arn:aws:geo:us-west-2:123456789012:geofence-
collection/ExampleGeofenceCollection"
}
```

AWS CLI

使用命令将跟踪器资源链接到地理围栏集合 AWS CLI

使用 `associate-tracker-consumer` 命令。

以下示例使用创建名 AWS CLI 为的地理围栏集合。*ExampleGeofenceCollection*

```
aws location \
    associate-tracker-consumer \
        --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-
collection/ExampleGeofenceCollection" \
        --tracker-name "ExampleTracker"
```

使用 Amazon Location Service AWS IoT 和 MQTT 进行跟踪

MQTT 是一种轻量级且被广泛采用的消息协议，专为受限设备而设计。 AWS IoT Core 支持使用 MQTT 协议和 MQTT over S WebSocket ecure (WSS) 协议的设备连接。

AWS IoT Core 将设备连接到设备 AWS 并使您能够在它们之间发送和接收消息。 AWS IoT Core 规则引擎存储有关设备消息主题的查询，并允许您定义向其他 AWS 服务（例如 Amazon Location Service）发送消息的操作。知道自己位置为坐标的设备可以通过规则引擎将其位置转发到 Amazon Location 。

> **Note**
>
> 设备可能知道自己的位置，例如通过内置 GPS。 AWS IoT 还支持第三方设备位置跟踪。想要了解更多信息，请参阅 AWS IoT Core 开发人员指南中的 AWS IoT Core Device Location。

以下演练介绍了使用 AWS IoT Core 规则进行跟踪。如果您需要在发送到 Amazon Locati AWS Lambda on 之前对其进行处理，也可以将设备信息发送到您自己的部门。有关使用 Lambda 处理您的设备位置的更多详细信息，请参阅 AWS Lambda 与 MQTT 一起使用。

主题

- 先决条件
- 创建 AWS IoT Core 规则
- 在控制台中测试你的 AWS IoT Core 规则
- AWS Lambda 与 MQTT 一起使用

先决条件

在您开始跟踪之前，您必须完成以下前提条件：

- 创建跟踪器资源，将设备位置数据发送到该资源。
- 创建 IAM 角色以授予对跟踪器的 AWS IoT Core 访问权限。

  执行这些步骤时，请使用以下策略授予对您的跟踪器的访问权限：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "WriteDevicePosition",
      "Effect": "Allow",
      "Action": "geo:BatchUpdateDevicePosition",
      "Resource": "arn:aws:geo:*:*:tracker/*"
    }
  ]
}
```

创建 AWS IoT Core 规则

接下来，创建一条 AWS IoT Core 规则，将设备的位置遥测数据转发给 Amazon Location Service。有关创建规则的更多信息，请参阅 AWS IoT Core 开发人员指南中的以下主题：

- 为有关@@ 创建新 AWS IoT 规则的信息创建规则。
- 位置操作，用于特定于创建发布到 Amazon Location 的规则的信息

在控制台中测试你的 AWS IoT Core 规则

如果当前没有设备发布包含位置的遥测数据，则可以使用 AWS IoT Core 控制台测试规则。控制台有一个测试客户端，您可以在其中发布一条示例消息来验证解决方案的结果。

1. 登录 AWS IoT Core 控制台，网址为https://console.aws.amazon.com/iot/。
2. 在左侧导航窗格中，展开测试，然后选择 MQTT 测试客户端。
3. 在“发布到主题”下，将主题名称设置为*iot/topic*（或您在 AWS IoT Core 规则中设置的主题的名称，如果不同），然后为消息负载提供以下内容。将时间戳*1604940328*替换为最近 30 天内的有效时间戳（Amazon Location Service 跟踪器会忽略任何超过 30 天的时间戳）。

```
{
  "payload": {
    "deviceid": "thing123",
    "timestamp": 1604940328,
    "location": { "lat": 49.2819, "long": -123.1187 },
    "accuracy": { "Horizontal": 20.5 },
    "positionProperties": { "field1": "value1", "field2": "value2" }
  }
}
```

4. 选择发布到主题来发送测试消息。
5. 要验证 Amazon Location Service 是否已收到该消息，请使用以下 AWS CLI 命令。如果您在设置过程中对其进行了修改，请将跟踪器名称替换为您使用的名称。

```
aws location batch-get-device-position --tracker-name MyTracker --device-ids
 thing123
```

AWS Lambda 与 MQTT 一起使用

虽然在将设备位置数据发送到 Amazon Location 进行跟踪时不再需要使用 AWS Lambda ，但在某些情况下，您可能仍想使用 Lambda。例如，如果您想在将设备位置数据发送到 Amazon Location 之前自己处理设备位置数据。以下主题介绍如何在消息发送到您的跟踪器之前使用 Lambda 处理消息。有关此模式的更多信息，请参阅参考架构。

主题

- 先决条件
- 创建 Lambda 函数

- 创建 AWS IoT Core 规则
- 在控制台中测试你的 AWS IoT Core 规则

先决条件

在开始追踪之前，必须先创建跟踪器资源。要创建追踪器资源，您可以使用亚马逊定位控制台 AWS CLI、或亚马逊位置 APIs。

以下示例使用 Amazon Location Service 控制台创建跟踪器资源：

1. 打开 Amazon Location Service 控制台，网址为https://console.aws.amazon.com/location/。
2. 在左侧导航窗格中，选择跟踪器。
3. 选择创建跟踪器。
4. 填写以下选框：
   - 名称 – 输入一个最多包含 100 个字符的唯一名称。有效条目包括：字母数字字符、连字符和下划线。例如 *MyTracker*。
   - 描述 – 输入可选描述。例如 *Tracker for storing AWS IoT Core device positions*。
   - 位置筛选 – 选择要用于位置更新的筛选。例如，基于精度的筛选。
5. 选择创建跟踪器。

创建 Lambda 函数

要在 AWS IoT Core 和 Amazon Location Service 之间建立连接，您需要一个 AWS Lambda 函数来处理由转发的消息 AWS IoT Core。此函数将提取所有位置数据，将其格式化为 Amazon Location Service，然后通过 Amazon Location 跟踪器 API 提交。您可以通过 AWS Lambda 控制台创建此函数，也可以使用 AWS Command Line Interface (AWS CLI) 或 AWS Lambda APIs。

要使用控制台创建 Lambda 函数，将位置更新发布到 Amazon Location，请执行以下操作：

1. 打开 AWS Lambda 控制台，网址为https://console.aws.amazon.com/lambda/。
2. 从左侧导航窗格中，选择函数。
3. 选择创建函数，并确保选择从头开始创作。
4. 填写以下选框：

- 函数名称 – 输入您的函数的唯一名称。有效条目包括字母数字字符、连字符和下划线，不带空格。例如 *MyLambda*。
- 运行时间-选择*Python 3.8*。

5. 选择创建函数。
6. 选择 代码 选项卡以打开编辑器。
7. 用以下代码替换 lambda_function.py 中的占位符代码，将分配给 TRACKER_NAME 的值替换为您作为先决条件创建的跟踪器的名称。

```
from datetime import datetime
import json
import os

import boto3

# Update this to match the name of your Tracker resource
TRACKER_NAME = "MyTracker"

"""
This Lambda function receives a payload from AWS IoT Core and publishes device
 updates to
Amazon Location Service via the BatchUpdateDevicePosition API.

Parameter 'event' is the payload delivered from AWS IoT Core.

In this sample, we assume that the payload has a single top-level key 'payload' and
 a nested key
'location' with keys 'lat' and 'long'. We also assume that the name of the device
 is nested in
the payload as 'deviceid'. Finally, the timestamp of the payload is present as
 'timestamp'. For
example:

>>> event
{ 'payload': { 'deviceid': 'thing123', 'timestamp': 1604940328,
  'location': { 'lat': 49.2819, 'long': -123.1187 },
  'accuracy': {'Horizontal': 20.5 },
  'positionProperties': {'field1':'value1','field2':'value2'} }
}
```

```
If your data doesn't match this schema, you can either use the AWS IoT Core rules
 engine to
format the data before delivering it to this Lambda function, or you can modify the
 code below to
match it.
"""
def lambda_handler(event, context):
  update = {
      "DeviceId": event["payload"]["deviceid"],
      "SampleTime": datetime.fromtimestamp(event["payload"]
["timestamp"]).strftime("%Y-%m-%dT%H:%M:%SZ"),
      "Position": [
        event["payload"]["location"]["long"],
        event["payload"]["location"]["lat"]
        ]
    }
  if "accuracy" in event["payload"]:
      update["Accuracy"] = event["payload"]['accuracy']
  if "positionProperties" in event["payload"]:
      update["PositionProperties"] = event["payload"]['positionProperties']

  client = boto3.client("location")
  response = client.batch_update_device_position(TrackerName=TRACKER_NAME,
 Updates=[update])

  return {
    "statusCode": 200,
    "body": json.dumps(response)
  }
```

8. 选择部署以保存您更新的函数。
9. 选择配置选项卡。
10. 在权限部分，选择超链接的角色名称以授予 Amazon Location Service 对您的 Lambda 函数的权限。
11. 在角色的摘要页面中，选择添加权限，然后从下拉列表中选择创建内联策略。
12. 选择 JSON 选项卡，并用以下文档覆盖策略。这允许您的 Lambda 函数更新由所有区域的所有跟踪器资源管理的设备位置。

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Sid": "WriteDevicePosition",
      "Effect": "Allow",
      "Action": "geo:BatchUpdateDevicePosition",
      "Resource": "arn:aws:geo:*:*:tracker/*"
    }
  ]
}
```

13. 选择查看策略。
14. 输入策略名称。例如 *AmazonLocationTrackerWriteOnly*。
15. 选择创建策略。

您可以根据需要修改此函数代码，以适应您自己的设备消息架构。

创建 AWS IoT Core 规则

接下来，创建一条 AWS IoT Core 规则，将设备的位置遥测数据转发给该 AWS Lambda 函数，以便转换并发布到 Amazon Location Service。提供的示例规则假设设备有效负载的任何必要转换均由您的 Lambda 函数处理。您可以通过 AWS IoT Core 控制台、 AWS Command Line Interface (AWS CLI) 或创建此规则 AWS IoT Core APIs。

> **Note**
>
> 虽然 AWS IoT 控制台处理 AWS IoT Core 允许调用您的 Lambda 函数所需的权限，但如果您从 AWS CLI 或 SDK 创建规则，则必须配置策略以授予权限。 AWS IoT

AWS IoT Core 使用控制台创建

1. 登录 AWS IoT Core 控制台，网址为https://console.aws.amazon.com/iot/。
2. 在左侧导航窗格中，展开行动，并选择规则。
3. 选择创建规则以启动新规则向导。
4. 为您的规则输入名称和描述。
5. 对于规则查询语句，请更新 FROM 属性，以引用至少一台设备正在发布包含位置的遥测的主题。如果您正在测试解决方案，则无需进行任何修改。

```
SELECT * FROM 'iot/topic'
```

6. 在设置一个或多个操作下，选择添加操作。
7. 选择向 Lambda 函数发送消息。
8. 选择 Configure action。
9. 从列表中选择您的 Lambda 函数。
10. 选择添加操作。
11. 选择创建规则。

在控制台中测试你的 AWS IoT Core 规则

如果当前没有设备发布包含位置信息的遥测数据，则可以使用 AWS IoT Core 控制台测试您的规则和此解决方案。控制台有一个测试客户端，您可以在其中发布一条示例消息来验证解决方案的结果。

1. 登录 AWS IoT Core 控制台，网址为https://console.aws.amazon.com/iot/。
2. 在左侧导航窗格中，展开测试，然后选择 MQTT 测试客户端。
3. 在 “发布到主题” 下，将主题名称设置为*iot/topic*（或您在 AWS IoT Core 规则中设置的主题的名称，如果不同），然后为消息负载提供以下内容。将时间戳*1604940328*替换为最近 30 天内的有效时间戳（任何超过 30 天的时间戳都将被忽略）。

```
{
  "payload": {
    "deviceid": "thing123",
    "timestamp": 1604940328,
    "location": { "lat": 49.2819, "long": -123.1187 },
    "accuracy": { "Horizontal": 20.5 },
    "positionProperties": { "field1": "value1", "field2": "value2" }
  }
}
```

4. 选择发布到主题来发送测试消息。
5. 要验证 Amazon Location Service 是否已收到该消息，请使用以下 AWS CLI 命令。如果您在设置过程中对其进行了修改，请将跟踪器名称和设备 ID 替换为您使用的跟踪器名称和设备 ID。

```
aws location batch-get-device-position --tracker-name MyTracker --device-ids
 thing123
```

管理您的亚马逊 Location Service 追踪器

您可以使用 Amazon Location 控制台 AWS CLI、或 Amazon 位置来管理您的追踪器。 APIs

列出您的跟踪器

您可以使用 Amazon Location 控制台 AWS CLI、或 Amazon 位置查看您的追踪器列表： APIs

Console

使用 Amazon Location 控制台查看现有跟踪器列表

1. 打开 Amazon Location 控制台，网址为https://console.aws.amazon.com/location/。
2. 从左侧导航窗格中选择跟踪器。
3. 在我的跟踪器下查看您的跟踪器资源列表。

API

使用 Amazon 位置追踪器 APIs中的ListTrackers操作。

以下示例是一个 API 请求，用于获取您 AWS 账户中的跟踪器列表。

```
POST /tracking/v0/list-trackers
```

以下为 ListTrackers 响应示例：

```
{
   "Entries": [
      {
         "CreateTime": 2020-10-02T19:09:07.327Z,
         "Description": "string",
         "TrackerName": "ExampleTracker",
         "UpdateTime": 2020-10-02T19:10:07.327Z
      }
   ],
   "NextToken": "1234-5678-9012"
}
```

CLI

使用 list-trackers 命令。

以下示例是 AWS CLI 获取您 AWS 账户中跟踪器列表的示例。

```
aws location list-trackers
```

断开跟踪器与地理围栏集合的连接

您可以使用 Amazon Location 控制台、或 Amazon Location 将 AWS CLI追踪器与地理围栏集合断开连接： APIs

Console

使用 Amazon Location 控制台取消跟踪器与关联的地理围栏集合的关联

1. 打开 Amazon Location 控制台，网址为https://console.aws.amazon.com/location/。
2. 从左侧导航窗格中选择跟踪器。
3. 在我的跟踪器下，选择目标跟踪器的名称链接。
4. 在链接的地理围栏集合下，选择状态为已链接的地理围栏集合。
5. 选择断开关联。

API

使用 Amazon 位置追踪器 APIs中的DisassociateTrackerConsumer操作。

以下示例是一个用于取消跟踪器与链接的地理围栏集合的关联的 API 请求。

```
DELETE /tracking/v0/trackers/ExampleTracker/consumers/arn:aws:geo:us-
west-2:123456789012:geofence-collection/ExampleCollection
```

以下为 DisassociateTrackerConsumer 响应示例：

```
HTTP/1.1 200
```

CLI

使用 disassociate-tracker-consumer 命令。

以下示例是取消跟踪器与关联的地理围栏集合的关联的 AWS CLI 命令。

```
aws location disassociate-tracker-consumer \
```

```
    --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-collection/
ExampleCollection" \
    --tracker-name "ExampleTracker"
```

获取跟踪器详细信息

您可以使用 Amazon Location 控制台、或 Amazon 位置来获取有关您 AWS 账户中 AWS CLI任何追踪器的详细信息 APIs。

Console

使用 Amazon Location 控制台查看跟踪器详情

1. 打开 Amazon Location 控制台，网址为https://console.aws.amazon.com/location/。
2. 从左侧导航窗格中选择跟踪器。
3. 在我的跟踪器下，选择目标跟踪器的名称链接。
4. 在信息下查看跟踪器详情。

API

使用 Amazon 位置追踪器中的DescribeTracker操作 APIs。

以下示例是获取其跟踪链接详情的 API 请求*ExampleTracker*。

```
GET /tracking/v0/trackers/ExampleTracker
```

以下为 DescribeTracker 响应示例：

```
{
   "CreateTime": 2020-10-02T19:09:07.327Z,
   "Description": "string",
   "EventBridgeEnabled": false,
   "KmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
   "PositionFiltering": "TimeBased",
   "Tags": {
      "Tag1" : "Value1"
   },
   "TrackerArn": "arn:aws:geo:us-west-2:123456789012:tracker/ExampleTracker",
   "TrackerName": "ExampleTracker",
   "UpdateTime": 2020-10-02T19:10:07.327Z
```

```
}
```

CLI

使用 describe-tracker 命令。

以下示例是获取跟踪器详细信息的 AWS CLI 命令*ExampleTracker*。

```
aws location describe-tracker \
    --tracker-name "ExampleTracker"
```

删除跟踪器

您可以使用 Amazon Location 控制台 AWS CLI、或 Amazon 位置从 AWS 账户中删除追踪器 APIs：

Console

使用 Amazon Location 控制台删除现有地图资源

> **Warning**
>
> 此操作将永久删除资源。如果跟踪器资源正在使用中，您可能会遇到错误。确保目标资源不是应用程序的依赖项。

1. 打开 Amazon Location 控制台，网址为https://console.aws.amazon.com/location/。
2. 从左侧导航窗格中选择跟踪器。
3. 在我的跟踪器下，选择目标跟踪器。
4. 选择删除跟踪器。

API

使用 Amazon 位置追踪器中的DeleteTracker操作 APIs。

以下示例是删除跟踪器的 API 请求*ExampleTracker*。

```
DELETE /tracking/v0/trackers/ExampleTracker
```

以下为 DeleteTracker 响应示例：

```
HTTP/1.1 200
```

CLI

使用 delete-tracker 命令。

以下示例是删除跟踪器的 AWS CLI 命令*ExampleTracker*。

```
aws location delete-tracker \
    --tracker-name "ExampleTracker"
```

管理成本和使用量

在您继续学习 Amazon Location 跟踪器时，重要的是要了解如何管理服务容量，确保遵守使用限制，并通过配额和 API 优化获得最佳结果。通过应用性能和准确性方面的最佳实践，您可以定制应用程序，以高效地处理与地点相关的查询并最大限度地提高 API 请求。

主题

- 追踪器定价
- 追踪器配额和使用情况

追踪器定价

有关追踪和地理围栏 APIs的定价信息，请参阅 Amazon Location S ervice 定价页面。

已写职位

您可以使用将一台或多台设备的位置更新数据上传`BatchUpdateDevicePosition`到跟踪器资源（每批最多十台设备）。价格基于您的 API 请求中的设备头寸数量。每台设备头寸更新的单价基于每月的总使用量。有关单价和数量套餐的详细信息，请参阅 Amazon Location Service 定价页面。

您可以通过配置跟踪设备上的设备位置更新频率（也称为 ping 速率）来优化您的持仓成本，也可以选择使用本地筛选器仅将相关的设备位置更新上传到 Amazon Location Service。

位置读取

您可以使用列出`BatchGetDevicePosition`所请求设备的最新设备位置，每个请求最多可列出 100 台设备。您还可以使用`GetDevicePosition`根据采样时间检索设备的最新位置。

价格基于 API 请求的数量。

位置删除

您可以使用BatchDeleteDevicePositionHistory从跟踪器资源中删除一台或多台设备的位置历史记录，每次请求最多可删除 100 台设备。

价格基于您的 API 请求中的设备数量。

职位完整性评估

您可以使用VerifyDevicePosition来验证设备位置的完整性，方法是确定设备位置是否是在代理服务器后面报告的，然后将其与根据设备状态估计的推断位置进行比较。

价格基于 API 请求的数量。

追踪器配额和使用情况

本主题概述了 Amazon Location Service 追踪器的速率限制和配额。

> **Note**
>
> 如果您需要更高的配额，可以使用服务配额控制台请求增加可调配额的配额。申请增加配额时，请选择您要求增加配额的区域，因为大多数配额都是特定于该 AWS 地区的。您最多可以为每个 API 请求两倍的默认限制。
> 对于超过默认限制两倍的请求，您的请求将提交支持请求。您也可以联系您的高级支持团队。增加配额的请求不会直接收费，但是根据消耗的额外资源，更高的使用级别可能会导致服务成本增加。请参阅the section called “管理配额”了解更多信息。

Service Quotas 是每个 AWS 账户和每个 AWS 地区可以拥有的最大资源数量。Amazon Location Service 拒绝超过服务配额的其他请求。

资源

API 名称	默认	最大可调极限
每个账户的跟踪器资源	500	1000 如果您需要的不止于此，请申请增加配额或联系支持团队。

API 名称	默认	最大可调极限
每个跟踪器的跟踪器使用者人数	5	最大可调限制不适用。 请联系支持团队。

CRUD API

> **Note**
>
> 如果您需要为其中任何一项设置更高的上限 APIs，请申请增加配额或联系支持团队。

API 名称	默认	最大可调极限
AssociateTrackerConsumer	10	20
CreateTracker	10	20
DeleteTracker	10	20
DescribeTracker	10	20
DisassociateTrackerConsumer	10	20
ListTrackerConsumers	10	20
ListTrackers	10	20
UpdateTracker	10	20

数据 API

> **Note**
>
> 如果您需要为其中任何一项设置更高的上限 APIs，请申请增加配额或联系支持团队。

API 名称	默认	最大可调极限
BatchGetDevicePosition	50	100
BatchUpdateDevicePosition	50	100
GetDevicePosition	50	100
GetDevicePositionHistory	50	100
BatchDeleteDevicePositionHistory	50	100
ListDevicePositions	50	100

其他使用限制

用于使用 Amazon Location Service 的开发者工具

AWS 提供适用于多种编程语言的软件开发套件 (SDKs)，使您可以轻松地将 Amazon Location Service 集成到您的应用程序中。本页概述了可用的 SDKs软件及其安装过程和代码示例，以帮助您在首选的开发环境中开始使用 Amazon Location Service。

有几种工具可以帮助您使用 Amazon Location Service。

主题

- SDKs 以及 Amazon Location Service 的框架
- Amazon Location Service API 和 CLI
- 示例和学习资源

SDKs 以及 Amazon Location Service 的框架

AWS 提供适用于多种编程语言的软件开发套件 (SDKs)，使您可以轻松地将 Amazon Location Service 集成到您的应用程序中。本页概述了可用的 SDKs软件及其安装过程和代码示例，以帮助您在首选的开发环境中开始使用 Amazon Location Service。

有几种工具可以帮助您使用 Amazon Location Service。

- AWS SDKs— AWS 软件开发套件 (SDKs) 提供多种常用编程语言版本，提供了 API、代码示例和文档，可让您更轻松地使用首选语言构建应用程序。 AWS SDKs 包括 Amazon 的核心位置 APIs 和功能，包括访问地图、地点、路线、地理围栏和追踪器。要详细了解 SDKs 可用于不同应用程序和语言的 Amazon Location Service，请参阅the section called “SDKs 按语言分类”。
- MapLibre— Amazon Location Service 建议使用MapLibre渲染引擎渲染地图。 MapLibre 是用于在 Web 或移动应用程序中显示地图的引擎。 MapLibre 还具有插件模型，并支持某些语言和平台的用户界面进行搜索和路由。要了解有关使用 MapLibre 及其提供的功能的更多信息，请参阅the section called “使用 MapLibre 工具”。
- Amazon Loc ation SDKs — Amazon Loc SDKs ation 是一组开源库，可以更轻松地使用 Amazon Location Service 开发应用程序。这些库提供了支持移动和网络应用程序身份验证、移动应用程序位置跟踪、亚马逊位置数据类型与 GeoJSON 之间的转换以及适用于 S AWS DK v3 的 Amazon Location 客户端托管包的功能。要了解有关 Amazon 营业地点的更多信息 SDKs，请参阅the section called “SDKs 按语言分类”。
- Amazon Location Migration SDK – Amazon Location Migration SDK 提供一个使您能够将现有应用程序从 Google 地图迁移到 Amazon Location 的桥梁。如果 Amazon Location 支持所使用的功能，

迁移软件开发工具包为您使用 Google Maps SDK 构建的应用程序提供了一个使用亚马逊定位服务的选项，无需重写任何应用程序或业务逻辑。 JavaScript 该 Migration SDK 会将所有 API 调用重定向到 Amazon Location，而不是 Google Map。要开始使用，请参阅上的 Amazon 位置迁移软件开发工具包 GitHub。

开发者教程

使用本节学习如何使用亚马逊定位服务 SDK 的各个方面。

主题

- 如何使用身份验证助手
- 使用 Amazon 位置 MapLibre 地理编码器 GL 插件
- 如何使用追踪 SDKs
- 在 Amazon Location 中使用 MapLibre 工具和相关库

如何使用身份验证助手

本节提供有关身份验证助手的其他信息。

Web

当从应用程序调用 Amazon Location Service API 时，亚马逊位置 JavaScript 认证实用程序可帮助进行身份验证。 JavaScript 这些实用程序特别支持使用 API 密钥或 Amazon Cognito 进行身份验证。

安装

- 使用 NPM 安装此库：

```
npm install @aws/amazon-location-utilities-auth-helper
```

- 要直接在浏览器中使用它，请在您的 HTML 文件中包含以下内容：

```
<script src="https://cdn.jsdelivr.net/npm/@aws/amazon-location-utilities-auth-
helper@1"></script>
```

使用量

要使用身份验证助手，请导入库并调用必要的实用函数。该库支持对来自 Amazon Location Service 的请求进行身份验证 SDKs，包括独立的地图 SDKs、地点和路线，以及使用 MapLibre GL JS 渲染地图。

与模块一起使用

此示例演示如何使用独立的 Places SDK 发出使用 API 密钥进行身份验证的请求：

```
npm install @aws-sdk/geo-places-client

import { GeoPlacesClient, GeocodeCommand } from "@aws-sdk/geo-places-client";
import { withAPIKey } from "@aws/amazon-location-utilities-auth-helper";

const authHelper = withAPIKey("<API Key>", "<Region>");
const client = new GeoPlacesClient(authHelper.getClientConfig());

const input = { ... };
const command = new GeocodeCommand(input);
const response = await client.send(command);
```

此示例演示如何使用独立的 Routes SDK 发出使用 API 密钥进行身份验证的请求：

```
npm install @aws-sdk/geo-routes-client

import { GeoRoutesClient, CalculateRoutesCommand } from "@aws-sdk/geo-routes-client";
import { withAPIKey } from "@aws/amazon-location-utilities-auth-helper";

const authHelper = withAPIKey("<API Key>", "<Region>");
const client = new GeoRoutesClient(authHelper.getClientConfig());

const input = { ... };
const command = new CalculateRoutesCommand(input);
const response = await client.send(command);
```

此示例使用带有 API 密钥身份验证的定位 SDK：

```
npm install @aws-sdk/client-location

import { LocationClient, ListGeofencesCommand } from "@aws-sdk/client-location";
import { withAPIKey } from "@aws/amazon-location-utilities-auth-helper";
```

```
const authHelper = withAPIKey("<API Key>", "<Region>");
const client = new LocationClient(authHelper.getClientConfig());

const input = { ... };
const command = new ListGeofencesCommand(input);
const response = await client.send(command);
```

在浏览器中使用

当直接在浏览器环境中使用时，可以在 amazonLocationAuth Helper 全局对象下访问实用函数。

此示例演示了使用 Amazon 定位客户端的请求，该请求使用 API 密钥进行身份验证：

```
<script src="https://cdn.jsdelivr.net/npm/@aws/amazon-location-client@1"></script>

const authHelper = amazonLocationClient.withAPIKey("<API Key>", "<Region>");
const client = new amazonLocationClient.GeoRoutesClient(authHelper.getClientConfig());
const input = { ... };
const command = new amazonLocationClient.routes.CalculateRoutesCommand(input);
const response = await client.send(command);
```

此示例演示了使用 MapLibre GL JS 渲染地图，并使用 API 密钥进行身份验证：

```
<script src="https://cdn.jsdelivr.net/npm/maplibre-gl@4"></script>

const apiKey = "<API Key>";
const region = "<Region>";
const styleName = "Standard";

const map = new maplibregl.Map({
  container: "map",
  center: [-123.115898, 49.295868],
  zoom: 10,
  style: `https://maps.geo.${region}.amazonaws.com/v2/styles/${styleName}/descriptor?
key=${apiKey}`,
});
```

此示例演示如何使用 Amazon Cognito 使用 MapLibre GL JS 渲染地图：

```
<script src="https://cdn.jsdelivr.net/npm/maplibre-gl@4"></script>
<script src="https://cdn.jsdelivr.net/npm/@aws/amazon-location-utilities-auth-
helper@1"></script>
```

```
const identityPoolId = "<Identity Pool ID>";
const authHelper = await amazonLocationAuthHelper.withIdentityPoolId(identityPoolId);

const map = new maplibregl.Map({
  container: "map",
  center: [-123.115898, 49.295868],
  zoom: 10,
  style: `https://maps.geo.${region}.amazonaws.com/v2/styles/${styleName}/descriptor`,
  ...authHelper.getMapAuthenticationOptions(),
});
```

经过身份验证的替代用法

您可以修改 withIdentityPool Id 函数以包含经过身份验证的身份的自定义参数：

```
const userPoolId = "<User Pool ID>";

const authHelper = await amazonLocationAuthHelper.withIdentityPoolId(identityPoolId, {
  logins: {
    [`cognito-idp.${region}.amazonaws.com/${userPoolId}`]: "cognito-id-token"
  }
});
```

iOS

适用于 iOS 的 Amazon Location Service 移动身份验证 SDK 可帮助验证来 APIs 自 iOS 应用程序的亚马逊定位服务请求。它特别支持通过 API 密钥或 Amazon Cognito 进行身份验证。

安装

- 打开 Xcode，然后转到 “文件” > “添加包依赖关系”。
- 在搜索栏中键入软件包网址 (https://github.com/aws-geospatial/amazon-location-mobile-auth-sdk-ios/)，然后按 Enter。
- 选择 “amazon-location-mobile-auth-sdk-ios” 软件包，然后单击 “添加包”。
- 选择 “AmazonLocationiOSAuthSDK” 套餐产品，然后单击 Add Packag e。

使用量

安装库后，使用该`AuthHelper`类为 API 密钥或 Amazon Cognito 配置客户端设置。

API 密钥

以下是使用带有 API 密钥身份验证的独立 Places SDK 的示例：

```
import AmazonLocationiOSAuthSDK
import AWSGeoPlaces

func geoPlacesExample() {
    let apiKey = "<API key>"
    let region = "<Region>"

    let authHelper = try await AuthHelper.withApiKey(apiKey: apiKey, region: region)
    let client: GeoPlacesClient = GeoPlacesClient(config:
 authHelper.getGeoPlacesClientConfig())

    let input = AWSGeoPlaces.SearchTextInput(
        biasPosition: [-97.7457518, 30.268193],
        queryText: "tacos"
    )

    let output = try await client.searchText(input: input)
}
```

以下是使用带有 API 密钥身份验证的独立 Routes SDK 的示例：

```
import AmazonLocationiOSAuthSDK
import AWSGeoRoutes

func geoRoutesExample() {
    let apiKey = "<API key>"
    let region = "<Region>"

    let authHelper = try await AuthHelper.withApiKey(apiKey: apiKey, region: region)
    let client: GeoRoutesClient = GeoRoutesClient(config:
 authHelper.getGeoRoutesClientConfig())

    let input = AWSGeoRoutes.CalculateRoutesInput(
        destination: [-123.1651031, 49.2577281],
        origin: [-97.7457518, 30.268193]
    )

    let output = try await client.calculateRoutes(input: input)
}
```

以下是使用带有 API 密钥身份验证的定位 SDK 的示例：

```
import AmazonLocationiOSAuthSDK
import AWSLocation

func locationExample() {
    let apiKey = "<API key>"
    let region = "<Region>"

    let authHelper = try await AuthHelper.withApiKey(apiKey: apiKey, region: region)
    let client: LocationClient = LocationClient(config:
 authHelper.getLocationClientConfig())

    let input = AWSLocation.ListGeofencesInput(
        collectionName: "<Collection name>"
    )

    let output = try await client.listGeofences(input: input)
}
```

以下是使用带有 Amazon Cognito 的独立地点软件开发工具包的示例：

```
import AmazonLocationiOSAuthSDK
import AWSGeoPlaces

func geoPlacesExample() {
    let identityPoolId = "<Identity Pool ID>"

    let authHelper = try await AuthHelper.withIdentityPoolId(identityPoolId:
 identityPoolId)
    let client: GeoPlacesClient = GeoPlacesClient(config:
 authHelper.getGeoPlacesClientConfig())

    let input = AWSGeoPlaces.SearchTextInput(
        biasPosition: [-97.7457518, 30.268193],
        queryText: "tacos"
    )

    let output = try await client.searchText(input: input)
}
```

以下是使用带有 Amazon Cognito 的独立路由 SDK 的示例：

```
import AmazonLocationiOSAuthSDK
```

```
import AWSGeoRoutes

func geoRoutesExample() {
    let identityPoolId = "<Identity Pool ID>"

    let authHelper = try await AuthHelper.withIdentityPoolId(identityPoolId:
 identityPoolId)
    let client: GeoRoutesClient = GeoRoutesClient(config:
 authHelper.getGeoRoutesClientConfig())

    let input = AWSGeoRoutes.CalculateRoutesInput(
        destination: [-123.1651031, 49.2577281],
        origin: [-97.7457518, 30.268193]
    )

    let output = try await client.calculateRoutes(input: input)
}
```

以下是使用带有 Amazon Cognito 的定位软件开发工具包的示例：

```
import AmazonLocationiOSAuthSDK
import AWSLocation

func locationExample() {
    let identityPoolId = "<Identity Pool ID>"

    let authHelper = try await AuthHelper.withIdentityPoolId(identityPoolId:
 identityPoolId)
    let client: LocationClient = LocationClient(config:
 authHelper.getLocationClientConfig())

    let input = AWSLocation.ListGeofencesInput(
        collectionName: "<Collection name>"
    )

    let output = try await client.listGeofences(input: input)
}
```

Android

适用于 Android 的 Amazon Location Service 移动身份验证 SDK 可帮助您对 APIs 来自安卓应用程序的亚马逊定位服务请求进行身份验证，特别是支持使用 Amazon Cognito 进行身份验证。

安装

- 此身份验证 SDK 可与整个 K AWS otlin SDK 配合使用。两者 SDKs 都发布到 Maven Central。在 Maven Central 上查看最新版本的身份验证 SDK。
- 在 Android Studio 中将以下几行添加到`build.gradle`文件的依赖项部分：

```
implementation("software.amazon.location:auth:1.1.0")
implementation("org.maplibre.gl:android-sdk:11.5.2")
implementation("com.squareup.okhttp3:okhttp:4.12.0")
```

- 对于独立地图、地点和路线 SDKs，请添加以下几行：

```
implementation("aws.sdk.kotlin:geomaps:1.3.65")
implementation("aws.sdk.kotlin:geoplaces:1.3.65")
implementation("aws.sdk.kotlin:georoutes:1.3.65")
```

- 对于包含地理围栏和跟踪功能的合并定位 SDK，请添加以下行：

```
implementation("aws.sdk.kotlin:location:1.3.65")
```

使用量

在您的代码中导入以下类：

```
// For the standalone Maps, Places, and Routes SDKs
import aws.sdk.kotlin.services.geomaps.GeoMapsClient
import aws.sdk.kotlin.services.geoplaces.GeoPlacesClient
import aws.sdk.kotlin.services.georoutes.GeoRoutesClient

// For the consolidated Location SDK
import aws.sdk.kotlin.services.location.LocationClient

import software.amazon.location.auth.AuthHelper
import software.amazon.location.auth.LocationCredentialsProvider
import software.amazon.location.auth.AwsSignerInterceptor
import org.maplibre.android.module.http.HttpRequestUtil
import okhttp3.OkHttpClient
```

你可以创建一个`AuthHelper`并将其与 AWS Kotlin SDK 配合使用：

示例：带有身份池 ID 的凭据提供商

```
private suspend fun exampleCognitoLogin() {
    val authHelper = AuthHelper.withCognitoIdentityPool("MY-COGNITO-IDENTITY-POOL-ID",
 applicationContext)

    var geoMapsClient = GeoMapsClient(authHelper?.getGeoMapsClientConfig())
    var geoPlacesClient = GeoPlacesClient(authHelper?.getGeoPlacesClientConfig())
    var geoRoutesClient = GeoRoutesClient(authHelper?.getGeoRoutesClientConfig())

    var locationClient = LocationClient(authHelper?.getLocationClientConfig())
}
```

示例：带有自定义凭据提供程序的凭据提供商

```
private suspend fun exampleCustomCredentialLogin() {
    var authHelper = AuthHelper.withCredentialsProvider(MY-CUSTOM-CREDENTIAL-PROVIDER,
 "MY-AWS-REGION", applicationContext)

    var geoMapsClient = GeoMapsClient(authHelper?.getGeoMapsClientConfig())
    var geoPlacesClient = GeoPlacesClient(authHelper?.getGeoPlacesClientConfig())
    var geoRoutesClient = GeoRoutesClient(authHelper?.getGeoRoutesClientConfig())

    var locationClient = LocationClient(authHelper?.getLocationClientConfig())
}
```

示例：带有 API 密钥的凭证提供商

```
private suspend fun exampleApiKeyLogin() {
    var authHelper = AuthHelper.withApiKey("MY-API-KEY", "MY-AWS-REGION",
 applicationContext)

    var geoMapsClient = GeoMapsClient(authHelper?.getGeoMapsClientConfig())
    var geoPlacesClient = GeoPlacesClient(authHelper?.getGeoPlacesClientConfig())
    var geoRoutesClient = GeoRoutesClient(authHelper?.getGeoRoutesClientConfig())

    var locationClient = LocationClient(authHelper?.getLocationClientConfig())
}
```

您可以使用`LocationCredentialsProvider`来加载 MapLibre 地图。示例如下：

```
HttpRequestUtil.setOkHttpClient(
    OkHttpClient.Builder()
```

```
        .addInterceptor(
            AwsSignerInterceptor(
                "geo",
                "MY-AWS-REGION",
                locationCredentialsProvider,
                applicationContext
            )
        )
        .build()
)
```

使用创建的客户端拨打 Amazon Location Service。以下是搜索靠近指定纬度和经度的地点的示例：

```
val suggestRequest = SuggestRequest {
       biasPosition = listOf(-97.718833, 30.405423)
       maxResults = MAX_RESULT
       language = "PREFERRED-LANGUAGE"
   }
val nearbyPlaces = geoPlacesClient.suggest(suggestRequest)
```

使用 Amazon 位置 MapLibre 地理编码器 GL 插件

Amazon Location MapLibre 地理编码器插件旨在让您在使用库进行地图渲染和地理编码时更轻松地将 Amazon Location 功能整合到 JavaScript 应用程序中。maplibre-gl-geocoder

安装

安装来自 NPM 的 Amazon Location MapLibre 地理编码器插件，以便与模块一起使用。键入以下命令：

```
npm install @aws/amazon-location-for-maplibre-gl-geocoder
```

您还可以使用脚本导入 HTML 和 CSS 文件，以便直接在浏览器中使用：

```
<script src="https://cdn.jsdelivr.net/npm/@aws/amazon-location-for-maplibre-gl-
geocoder@2"></script>
<link
  href="https://cdn.jsdelivr.net/npm/@aws/amazon-location-for-maplibre-gl-geocoder@2/
dist/amazon-location-for-mlg-styles.css"
  rel="stylesheet"
/>
```

与模块一起使用-独立 GeoPlaces SDK

此示例使用AWS 适用于 JavaScript V3 的 SDK 来获取 GeoPlacesClient 要提供给库的，并AuthHelper对库进行身份验证。 GeoPlacesClient它 APIs 为地理编码器启用了所有功能。

```
// Import MapLibre GL JS
import maplibregl from "maplibre-gl";
// Import from the AWS JavaScript SDK V3
import { GeoPlacesClient } from "@aws-sdk/client-geo-places";
// Import the utility functions
import { withAPIKey } from "@aws/amazon-location-utilities-auth-helper";
// Import the AmazonLocationMaplibreGeocoder
import {
  buildAmazonLocationMaplibreGeocoder,
  AmazonLocationMaplibreGeocoder,
} from "@aws/amazon-location-for-maplibre-gl-geocoder";

const apiKey = "<API Key>";
const mapName = "Standard";
const region = "<Region>"; // region containing Amazon Location API Key

// Create an authentication helper instance using an API key and region
const authHelper = await withAPIKey(apiKey, region);

const client = new GeoPlacesClient(authHelper.getClientConfig());

// Render the map
const map = new maplibregl.Map({
  container: "map",
  center: [-123.115898, 49.295868],
  zoom: 10,
  style: `https://maps.geo.${region}.amazonaws.com/v2/styles/${mapStyle}/descriptor?
key=${apiKey}`,
});

// Gets an instance of the AmazonLocationMaplibreGeocoder Object.
const amazonLocationMaplibreGeocoder = buildAmazonLocationMaplibreGeocoder(client,
 { enableAll: true });

// Now we can add the Geocoder to the map.
map.addControl(amazonLocationMaplibreGeocoder.getPlacesGeocoder());
```

使用浏览器-独立 GeoPlaces SDK

此示例使用 Amazon Location 客户端发出使用 API 密钥进行身份验证的请求。

> **Note**
>
> 其中一些示例使用了 Amazon 地点 GeoPlacesClient。此客户端基于AWS 适用于 JavaScript V3 的软件开发工具包，允许通过 HTML 文件中引用的脚本调用 Amazon Location。

在 HTML 文件中包含以下内容：

```
<!-- Import the Amazon Location For Maplibre Geocoder -->
<script src="https://cdn.jsdelivr.net/npm/@aws/amazon-location-for-maplibre-gl-
geocoder@2"></script>
<link
  href="https://cdn.jsdelivr.net/npm/@aws/amazon-location-for-maplibre-gl-geocoder@2/
dist/amazon-location-for-mlg-styles.css"
  rel="stylesheet"
/>
<!-- Import the Amazon GeoPlacesClient -->
<script src="https://cdn.jsdelivr.net/npm/@aws/amazon-location-client@1"></script>
```

在 JavaScript 文件中包含以下内容：

```
const apiKey = "<API Key>";
const mapStyle = "Standard";
const region = "<Region>"; // region containing Amazon Location API key

// Create an authentication helper instance using an API key and region
const authHelper = await amazonLocationClient.withAPIKey(apiKey, region);

const client = new amazonLocationClient.GeoPlacesClient(authHelper.getClientConfig());

// Render the map
const map = new maplibregl.Map({
  container: "map",
  center: [-123.115898, 49.295868],
  zoom: 10,
  style: `https://maps.geo.${region}.amazonaws.com/v2/styles/${mapStyle}/descriptor?
key=${apiKey}`,
});
```

```
// Initialize the AmazonLocationMaplibreGeocoder object
const amazonLocationMaplibreGeocoderObject =
 amazonLocationMaplibreGeocoder.buildAmazonLocationMaplibreGeocoder(
  client,
  { enableAll: true },
);

// Use the AmazonLocationWithMaplibreGeocoder object to add a geocoder to the map.
map.addControl(amazonLocationMaplibreGeocoderObject.getPlacesGeocoder());
```

函数

下面列出了 Amazon Location MapLibre 地理编码器插件中使用的函数：

- buildAmazonLocationMaplibreGeocoder

  该类创建了一个的实例AmazonLocationMaplibreGeocder，它是其他所有调用的入口点。

  使用独立 GeoPlacesClient API 调用（客户端是的实例GeoPlacesClient）：

  ```
  const amazonLocationMaplibreGeocoder = buildAmazonLocationMaplibreGeocoder(client,
   { enableAll: true });
  ```

  使用整合 LocationClient API 调用（客户端是的实例LocationClient）：

  ```
  const amazonLocationMaplibreGeocoder = buildAmazonLocationMaplibreGeocoder(client, {
    enableAll: true,
    placesIndex: placeIndex,
  });
  ```

- getPlacesGeocoder

  返回可以直接添加到地图的 ready-to-use IControl 对象。

  ```
  const geocoder = getPlacesGeocoder();

  // Initialize map see: <insert link to initializing a map instance here>
  let map = await initializeMap();

  // Add the geocoder to the map.
  map.addControl(geocoder);
  ```

如何使用追踪 SDKs

本主题提供有关如何使用跟踪的信息 SDKs。

iOS

Amazon Location 移动跟踪 SDK 提供的实用程序可帮助您轻松进行身份验证、捕获设备位置以及向 Amazon Location 跟踪器发送位置更新。该 SDK 支持通过可配置的更新间隔对位置更新进行本地筛选。这样可以降低数据成本，并优化 iOS 应用程序的间歇性连接。

iOS 追踪 SDK 可在以下网址获得 GitHub：适用于 iOS 的亚马逊定位移动追踪软件开发工具包。

本节涵盖 Amazon Location 移动跟踪 iOS SDK 的以下主题：

主题

- 安装
- 使用量
- 筛选条件
- iOS 移动 SDK 跟踪函数
- 示例

安装

使用以下步骤安装适用于 iOS 的移动跟踪 SDK：

1. 在您的 Xcode 项目中，转到文件，然后选择添加软件包依赖项。
2. 在搜索栏中键入以下 URL：https://github.com/aws-geospatial/amazon-location-mobile-tracking-sdk-ios/，然后按回车键。
3. 选择 `amazon-location-mobile-tracking-sdk-ios` 软件包并单击添加软件包。
4. 选择 `AmazonLocationiOSTrackingSDK` 软件包产品并单击添加软件包。

使用量

以下过程向您展示了如何使用来自 Amazon Cognito 的凭证创建身份验证助手。

1. 安装库后，需要在 `info.plist` 文件中添加一个或两个描述：

```
Privacy - Location When In Use Usage Description
```

```
Privacy - Location Always and When In Use Usage Description
```

2. 接下来，AuthHelper 在你的课堂中导入：

```
import AmazonLocationiOSAuthSDKimport AmazonLocationiOSTrackingSDK
```

3. 然后，您将使用来自 Amazon Cognito 的凭证创建身份验证助手，从而创建一个AuthHelper对象并将其与 AWS 软件开发工具包一起使用。

```
let authHelper = AuthHelper()
let locationCredentialsProvider =
 authHelper.authenticateWithCognitoUserPool(identityPoolId: "My-Cognito-Identity-
Pool-Id", region: "My-region") //example: us-east-1
let locationTracker = LocationTracker(provider: locationCredentialsProvider,
 trackerName: "My-tracker-name")

// Optionally you can set ClientConfig with your own values in either initialize or
 in a separate function
// let trackerConfig = LocationTrackerConfig(locationFilters:
 [TimeLocationFilter(), DistanceLocationFilter()],

trackingDistanceInterval: 30,
trackingTimeInterval: 30,
logLevel: .debug)

// locationTracker = LocationTracker(provider: credentialsProvider, trackerName:
 "My-tracker-name",config: trackerConfig)
// locationTracker.setConfig(config: trackerConfig)
```

筛选条件

Amazon Location 移动跟踪 iOS SDK 有三个内置的位置筛选器。

- `TimeLocationFilter`：根据定义的时间间隔筛选要上传的当前位置。
- `DistanceLocationFilter`：根据指定的距离阈值筛选位置更新。
- `AccuracyLocationFilter`：通过将自上次更新以来移动的距离与当前位置的精度进行比较来筛选位置更新。

此示例在创建时在 `LocationTracker` 中添加筛选器：

```
val config = LocationTrackerConfig(
    trackerName = "MY-TRACKER-NAME",
    logLevel = TrackingSdkLogLevel.DEBUG,
    accuracy = Priority.PRIORITY_HIGH_ACCURACY,
    latency = 1000,
    frequency = 5000,
    waitForAccurateLocation = false,
    minUpdateIntervalMillis = 5000,
    locationFilters = mutableListOf(TimeLocationFilter(), DistanceLocationFilter(),
 AccuracyLocationFilter())
)

locationTracker = LocationTracker(
    applicationContext,
    locationCredentialsProvider,
    config,
)
```

此示例在运行时使用 LocationTracker 启用和禁用筛选器：

```
// To enable the filter
locationTracker?.enableFilter(TimeLocationFilter())

// To disable the filter
locationTracker?.disableFilter(TimeLocationFilter())
```

iOS 移动 SDK 跟踪函数

适用于 iOS 的 Amazon Location 移动跟踪 SDK 包括以下函数：

- 类：LocationTracker

  init(provider: LocationCredentialsProvider, trackerName: String, config: LocationTrackerConfig? = nil)

  这是用于创建 LocationTracker 对象的初始化程序函数。它需要 LocationCredentialsProvider、trackerName 的实例，以及（可选）LocationTrackingConfig 的实例。如果未提供配置，将使用默认值对其进行初始化。

- 类：LocationTracker

  setTrackerConfig(config: LocationTrackerConfig)

这会将 Tracker 的配置设置为在位置跟踪器初始化后的任何时候生效。

- 类：LocationTracker

  getTrackerConfig()

  它将获取要在您的应用程序中使用或修改的位置跟踪配置。

  返回：LocationTrackerConfig
- 类：LocationTracker

  getDeviceId()

  获取位置跟踪器生成的设备 ID。

  返回：String?
- 类：LocationTracker

  startTracking()

  启动访问用户位置并将其发送到 AWS 跟踪器的过程。
- 类：LocationTracker

  resumeTracking()

  恢复访问用户位置并将其发送到 AWS 跟踪器的过程。
- 类：LocationTracker

  stopTracking()

  停止跟踪用户位置的过程。
- 类：LocationTracker

  startBackgroundTracking(mode: BackgroundTrackingMode)

  启动访问用户位置的过程，并在应用程序处于后台时将其发送到 AWS 跟踪器。
  BackgroundTrackingMode有以下选项：
  - Active: 此选项不会自动暂停位置更新。
  - BatterySaving:此选项会自动暂停位置更新。
  - None:此选项总体上禁用后台位置更新。

- 类：LocationTracker

  resumeBackgroundTracking(mode: BackgroundTrackingMode)

  当应用程序处于后台时，恢复访问用户位置并将其发送到 AWS 跟踪器的过程。

- 类：LocationTracker

  stopBackgroundTracking()

  当应用程序处于后台时，停止访问用户位置并将其发送到 AWS 跟踪器的过程。

- 类：LocationTracker

  getTrackerDeviceLocation(nextToken: String?, startTime: Date? = nil, endTime: Date? = nil, completion: @escaping (Result<GetLocationResponse, Error>)

  在开始和结束日期和时间之间检索为用户设备上传的跟踪位置。

  返回：Void

- 类：LocationTrackerConfig

  init()

  这将 LocationTrackerConfig 使用默认值初始化。

- 类：LocationTrackerConfig

  init(locationFilters: [LocationFilter]? = nil, trackingDistanceInterval: Double? = nil, trackingTimeInterval: Double? = nil, trackingAccuracyLevel: Double? = nil, uploadFrequency: Double? = nil, desiredAccuracy: CLLocationAccuracy? = nil, activityType: CLActivityType? = nil, logLevel: LogLevel? = nil)

  它使用用户定义的参数值初始化 LocationTrackerConfig。如果未提供参数值，会将其设置为默认值。

- 类：LocationFilter

  shouldUpload(currentLocation: LocationEntity, previousLocation: LocationEntity?, trackerConfig: LocationTrackerConfig)

`LocationFilter` 是用户可以为其自定义筛选器实现实施的协议。用户需要实现 `shouldUpload` 函数来比较以前和当前的位置，并返回是否应该上传当前位置。

示例

本节详细介绍了使用适用于 iOS 的 Amazon Location 移动跟踪 SDK 的示例。

> **Note**
>
> 确保在 `info.plist` 文件中设置了必要的权限。这些权限与 the section called “使用量” 部分中列出的权限相同。

以下示例演示了跟踪设备位置和检索跟踪位置的功能：

```
Privacy - Location When In Use Usage Description
Privacy - Location Always and When In Use Usage Description
```

开始跟踪位置：

```
do {
    try locationTracker.startTracking()
    }
catch TrackingLocationError.permissionDenied {
        // Handle permissionDenied by showing the alert message or opening the app
 settings
        }
```

恢复跟踪位置：

```
do {
    try locationTracker.resumeTracking()
    }
catch TrackingLocationError.permissionDenied {
    // Handle permissionDenied by showing the alert message or opening the app settings
    }
```

停止跟踪位置：

```
locationTracker.stopTracking()
```

开始后台跟踪：

```
do {

 locationTracker.startBackgroundTracking(mode: .Active) // .Active, .BatterySaving, .None
    }
catch TrackingLocationError.permissionDenied {
   // Handle permissionDenied by showing the alert message or opening the app settings
   }
```

恢复后台跟踪：

```
do {
    locationTracker.resumeBackgroundTracking(mode: .Active)
    }
catch TrackingLocationError.permissionDenied {
    // Handle permissionDenied by showing the alert message or opening the app settings
    }
```

停止后台跟踪：

```
locationTracker.stopBackgroundTracking()
```

从跟踪器检索设备的跟踪位置：

```
func getTrackingPoints(nextToken: String? = nil) {
let startTime: Date = Date().addingTimeInterval(-86400) // Yesterday's day date and
 time
let endTime: Date = Date()
locationTracker.getTrackerDeviceLocation(nextToken: nextToken, startTime: startTime,
 endTime: endTime, completion: { [weak self] result in
    switch result {
    case .success(let response):

        let positions = response.devicePositions
        // You can draw positions on map or use it further as per your requirement

        // If nextToken is available, recursively call to get more data
        if let nextToken = response.nextToken {
```

```
            self?.getTrackingPoints(nextToken: nextToken)
        }
    case .failure(let error):
        print(error)
    }
})
}
```

Android 移动跟踪 SDK

Amazon Location 移动跟踪 SDK 提供的实用程序可帮助您轻松进行身份验证、捕获设备位置以及向 Amazon Location 跟踪器发送位置更新。该 SDK 支持通过可配置的更新间隔对位置更新进行本地筛选。这样可以降低数据成本，优化 Android 应用程序的间歇性连接。

Android 追踪 SDK 可在以下网址获得 GitHub：适用于安卓的亚马逊定位移动追踪 SDK。此外，M AWS aven 存储库中还提供了移动身份验证 AWS SDK 和 SDK。Android 追踪 SDK 专为与通用 AWS SDK 配合使用而设计。

本节涵盖 Amazon Location 移动跟踪 Android SDK 的以下主题：

主题

- 安装
- 使用量
- 筛选条件
- Android 移动 SDK 跟踪函数
- 示例

安装

要安装该 SDK，请在 Android Studio 中将以下几行添加到 build.gradle 文件的依赖项部分：

```
implementation("software.amazon.location:tracking:0.0.1")
implementation("software.amazon.location:auth:0.0.1")
implementation("com.amazonaws:aws-android-sdk-location:2.72.0")
```

使用量

以下步骤展示了如何使用该 SDK 进行身份验证和创建 `LocationTracker` 对象：

> **Note**
>
> 以下步骤假设已经导入the section called “安装”部分中提到的库。

1. 在您的代码中导入以下类：

```
import software.amazon.location.tracking.LocationTracker
import software.amazon.location.tracking.config.LocationTrackerConfig
import software.amazon.location.tracking.util.TrackingSdkLogLevel
import com.amazonaws.services.geo.AmazonLocationClient
import software.amazon.location.auth.AuthHelper
import software.amazon.location.auth.LocationCredentialsProvider
```

2. 接下来，创建一个 AuthHelper，因为创建 LocationTracker 对象需要 LocationCredentialsProvider 参数：

```
// Create an authentication helper using credentials from Amazon Cognito
val authHelper = AuthHelper(applicationContext)
val locationCredentialsProvider : LocationCredentialsProvider =
 authHelper.authenticateWithCognitoIdentityPool("My-Cognito-Identity-Pool-Id")
```

3. 现在，使用 LocationCredentialsProvider 和 LocationTrackerConfig 创建 LocationTracker 对象：

```
val config = LocationTrackerConfig(
    trackerName = "MY-TRACKER-NAME",
    logLevel = TrackingSdkLogLevel.DEBUG,
    accuracy = Priority.PRIORITY_HIGH_ACCURACY,
    latency = 1000,
    frequency = 5000,
    waitForAccurateLocation = false,
    minUpdateIntervalMillis = 5000,
)
locationTracker = LocationTracker(
    applicationContext,
    locationCredentialsProvider,
    config,
)
```

筛选条件

Amazon Location 移动跟踪 Android SDK 有三个内置的位置筛选器。

- `TimeLocationFilter`：根据定义的时间间隔筛选要上传的当前位置。
- `DistanceLocationFilter`：根据指定的距离阈值筛选位置更新。
- `AccuracyLocationFilter`：通过将自上次更新以来移动的距离与当前位置的精度进行比较来筛选位置更新。

此示例在创建时在 `LocationTracker` 中添加筛选器：

```
val config = LocationTrackerConfig(
    trackerName = "MY-TRACKER-NAME",
    logLevel = TrackingSdkLogLevel.DEBUG,
    accuracy = Priority.PRIORITY_HIGH_ACCURACY,
    latency = 1000,
    frequency = 5000,
    waitForAccurateLocation = false,
    minUpdateIntervalMillis = 5000,
    locationFilters = mutableListOf(TimeLocationFilter(), DistanceLocationFilter(),
 AccuracyLocationFilter())
)
locationTracker = LocationTracker(
    applicationContext,
    locationCredentialsProvider,
    config,
)
```

此示例在运行时使用 `LocationTracker` 启用和禁用筛选器：

```
// To enable the filter
locationTracker?.enableFilter(TimeLocationFilter())

// To disable the filter
locationTracker?.disableFilter(TimeLocationFilter())
```

Android 移动 SDK 跟踪函数

适用于 Android 的 Amazon Location 移动跟踪 SDK 包括以下函数：

- 类：`LocationTracker`

```
constructor(context: Context,locationCredentialsProvider:
LocationCredentialsProvider,trackerName: String) 或
constructor(context: Context,locationCredentialsProvider:
LocationCredentialsProvider,clientConfig: LocationTrackerConfig)
```

这是用于创建 LocationTracker 对象的初始化程序函数。它需要 LocationCredentialsProvider、trackerName 的实例，以及（可选）LocationTrackingConfig 的实例。如果未提供配置，将使用默认值对其进行初始化。

- 类：LocationTracker

  ```
  start(locationTrackingCallback: LocationTrackingCallback)
  ```

  启动访问用户位置并将其发送到 Amazon Location 跟踪器的过程。

- 类：LocationTracker

  ```
  isTrackingInForeground()
  ```

  检查当前是否正在进行位置跟踪。

- 类：LocationTracker

  ```
  stop()
  ```

  停止跟踪用户位置的过程。

- 类：LocationTracker

  ```
  startTracking()
  ```

  启动访问用户位置并将其发送到 AWS 跟踪器的过程。

- 类：LocationTracker

  ```
  startBackground(mode: BackgroundTrackingMode, serviceCallback:
  ServiceCallback)
  ```

  启动访问用户位置的过程，并在应用程序处于后台时将其发送到 AWS 跟踪器。BackgroundTrackingMode有以下选项：

  - ACTIVE_TRACKING：此选项主动跟踪用户的位置更新。
  - BATTERY_SAVER_TRACKING：此选项每 15 分钟跟踪一次用户的位置更新。

- 类：LocationTracker

```
stopBackgroundService()
```

当应用程序处于后台时，停止访问用户位置并将其发送到 AWS 跟踪器的过程。

- 类：`LocationTracker`

```
getTrackerDeviceLocation()
```

从 Amazon Location Service 检索设备位置。

- 类：`LocationTracker`

```
getDeviceLocation(locationTrackingCallback: LocationTrackingCallback?)
```

从融合位置提供程序客户端检索当前的设备位置，并将其上传到 Amazon Location 跟踪器。

- 类：`LocationTracker`

```
uploadLocationUpdates(locationTrackingCallback:
LocationTrackingCallback?)
```

根据配置的位置筛选器进行筛选后，将设备位置上传到 Amazon Location Service。

- 类：`LocationTracker`

```
enableFilter(filter: LocationFilter)
```

启用特定位置筛选器。

- 类：`LocationTracker`

```
checkFilterIsExistsAndUpdateValue(filter: LocationFilter)
```

禁用特定位置筛选器。

- 类：`LocationTrackerConfig`

```
LocationTrackerConfig( // Required var trackerName:
String, // Optional var locationFilters: MutableList =
mutableListOf( TimeLocationFilter(), DistanceLocationFilter(), ), var
logLevel: TrackingSdkLogLevel = TrackingSdkLogLevel.DEBUG, var accuracy:
Int = Priority.PRIORITY_HIGH_ACCURACY, var latency: Long = 1000, var
frequency: Long = 1500, var waitForAccurateLocation: Boolean = false, var
```

```
minUpdateIntervalMillis: Long = 1000, var persistentNotificationConfig:
NotificationConfig = NotificationConfig())
```

它使用用户定义的参数值初始化 `LocationTrackerConfig`。如果未提供参数值，会将其设置为默认值。

- 类：`LocationFilter`

```
shouldUpload(currentLocation: LocationEntry, previousLocation:
LocationEntry?): Boolean
```

`LocationFilter` 是用户可以为其自定义筛选器实现实施的协议。您需要实现 `shouldUpload` 函数来比较以前和当前的位置，并返回是否应该上传当前位置。

示例

以下代码示例显示了移动跟踪 SDK 功能。

此示例使用 `LocationTracker` 在后台开始和停止跟踪：

```
// For starting the location tracking
locationTracker?.startBackground(
BackgroundTrackingMode.ACTIVE_TRACKING,
object : ServiceCallback {
    override fun serviceStopped() {
        if (selectedTrackingMode == BackgroundTrackingMode.ACTIVE_TRACKING) {
            isLocationTrackingBackgroundActive = false
        } else {
            isLocationTrackingBatteryOptimizeActive = false
        }
    }
},
)

// For stopping the location tracking
locationTracker?.stopBackgroundService()
```

在 Amazon Location 中使用 MapLibre 工具和相关库

MapLibre主要是用于在 Web 或移动应用程序中显示地图的渲染引擎。但是，它还包括对插件的支持，并提供用于处理 Amazon Location 其他方面的功能。以下内容根据您想要使用的区域或位置描述了您可以使用的工具。

> **Note**
>
> 要使用 Amazon Location 的任何功能，请为您要使用的语言安装AWS SDK。

- 映射

  要在应用程序中显示地图，您需要一个地图渲染引擎，该引擎将使用 Amazon Location 提供的数据并绘制到屏幕上。地图渲染引擎还提供平移和缩放地图，或者向地图添加标记、图钉和其他注释的功能。

  Amazon Location Service 建议使用MapLibre渲染引擎渲染地图。 MapLibre GL JS 是用于在中显示地图的引擎 JavaScript，而 MapLibre Native 则提供适用于 iOS 或 Android 的地图。

  MapLibre 还具有用于扩展核心功能的插件生态系统。欲了解更多信息，请访问 https://maplibre。org/maplibre-gl-js-docs/plugins/。
- 位置搜索

  为了简化搜索用户界面的创建，您可以使用网页版MapLibre 地理编码器（Android 应用程序可以使用 Android Plac es 插件）。

  使用 Amazon Location for MapLibre geocoder 库可简化在应用程序`amazon-location-for-maplibre-gl-geocoder`中 JavaScript 使用亚马逊位置信息的过程。

  有关更多信息，请参阅 the section called “使用 MapLibre Geocoder GL 插件”。
- 路线
- 地理围栏和跟踪器

  MapLibre 没有任何用于地理围栏和跟踪的特定渲染或工具，但您可以使用渲染功能和插件在地图上显示地理围栏和被跟踪的设备。

  被跟踪的设备可以使用 MQTT 或手动向 Amazon Location Service 发送更新。可以使用 AWS Lambda 对地理围栏事件进行响应。

许多开源库可用于为 Amazon Location Service 提供其他功能，例如提供空间分析功能的 Turf。

许多库都使用开放标准 GeoJSON 格式的数据。Amazon Location Service 提供了一个库，用于将响应转换为 geoJSON，以便在应用程序中使用。 JavaScript 有关更多信息，请参阅 the section called “SDKs 和框架”。

SDKs 按语言分类

> **开发工具包版本**
>
> 我们建议您使用项目中使用的最新版本的 AWS SDK 以及任何其他 SDKs版本，并保持最新版本。 SDKs S AWS DK 为您提供最新的特性和功能以及安全更新。例如，要查找最新版本的 AWS SDK JavaScript，请参阅软件开发AWS 工具包中的浏览器安装主题以获取 JavaScript文档。

下表按应用程序类型（Web、移动应用程序或后端应用程序）提供了有关 AWS SDKs 语言和框架的地图渲染框架版本的信息。

Web frontend

以下版本 AWS SDKs 和地图渲染框架版本可用于 Web 前端应用程序开发。

语言/框架	AWS SDK	地图渲染框架
完全支持		
JavaScript	https://aws.amazon.com/sdk-for-javascript/	https://maplibre。 org/projects/maplibre-gl-js/
ReactJS	https://aws.amazon.com/sdk-for-javascript/	https://github.com/maplibre/maplibre-react-native
TypeScript	https://aws.amazon.com/sdk-for-javascript/	https://maplibre。 org/projects/maplibre-gl-js/
部分支持		
Flutter	https://docs.amplify。 aws/start/q/integration/flutter/ Amplify 尚未完全支持 Flutter AWS，但通过 Amplify 提供的支持有限。	https://github.com/maplibre/flutter-maplibre-gl MapLibre Flutter 库被认为是实验性的。

语言/框架	AWS SDK	地图渲染框架
Node.js	https://aws.amazon.com/sdk-for-javascript/	https://github.com/maplibre/maplibre-native https://www.npmjs.com/package/@maplibre/maplibre-gl-native
PHP	https://aws.amazon.com/sdk-for-php/	不 MapLibre 支持 PHP。

Mobile frontend

以下版本 AWS SDKs 和地图渲染框架版本可用于移动前端应用程序开发。

语言/框架	AWS SDK	地图渲染框架
完全支持		
Java	https://aws.amazon.com/sdk-for-java/	https://maplibre。 org/projects/maplibre-native/
Kotlin	https://aws.amazon.com/sdk-for-kotlin/ 适用于安卓的 Amazon Location Service 移动身份验证 SDK：https://github.com/aws-geospatial/amazon-location-mobile-auth-sdk- android 适用于安卓的 Amazon Location Service 移动追踪 SDK：https://github.com/aws-geospatial/amazon-location-mobile-tracking-sdk- android	https://maplibre。 org/projects/maplibre-native/ 需要自定义绑定，就像基 MapLibre 于 Java 的绑定一样。

语言/框架	AWS SDK	地图渲染框架
ObjectiveC	https://github.com/aws-amplify/aws-sdk-ios	https://maplibre。 org/projects/maplibre-native/
ReactNative	https://aws.amazon.com/sdk-for-javascript/	https://github.com/maplibre/maplibre-react-native
Swift	https://aws.amazon.com/sdk-for-swift/ 适用于 iOS 的 Amazon Location Service 移动身份验证 SDK：https://github.com/aws-geospatial/amazon-location-mobile-auth-sdk- ios 适用于 iOS 的 Amazon Location Service 移动追踪 SDK：https://github.com/aws-geospatial/amazon-location-mobile-tracking-sdk- ios	https://maplibre。 org/projects/maplibre-native/
部分支持		
Flutter	https://docs.amplify。 aws/start/q/integration/flutter/ Amplify 尚未完全支持 Flutter AWS，但通过 Amplify 提供的支持有限。	https://github.com/maplibre/flutter-maplibre-gl MapLibre Flutter 库被认为是实验性的。

Backend application

AWS SDKs 以下内容可用于后端应用程序开发。此处未列出地图渲染框架，因为后端应用程序通常不需要地图渲染。

语言	AWS SDK
.NET	https://aws.amazon.com/sdk-for-net/
C++	https://aws.amazon.com/sdk-for-cpp/
Go	https://aws.amazon.com/sdk-for-go/
Java	https://aws.amazon.com/sdk-for-java/
JavaScript	https://aws.amazon.com/sdk-for-javascript/
Node.js	https://aws.amazon.com/sdk-for-javascript/
TypeScript	https://aws.amazon.com/sdk-for-javascript/
Kotlin	https://aws.amazon.com/sdk-for-kotlin/
PHP	https://aws.amazon.com/sdk-for-php/
Python	https://aws.amazon.com/sdk-for-python/
Ruby	https://aws.amazon.com/sdk-for-ruby/
Rust	https://aws.amazon.com/sdk-for-rust/ 适用于 Rust 的 AWS SDK 处于开发者预览版中。

按语言划分的地图渲染 SDK

我们建议使用渲染引擎MapLibre渲染 Amazon Location Service 地图。

MapLibre 是用于在 Web 或移动应用程序中显示地图的引擎。 MapLibre还具有插件模型，并支持某些语言和平台的用户界面进行搜索和路由。

要了解有关使用 MapLibre 及其提供的功能的更多信息，请参阅the section called “使用 MapLibre 工具”和the section called “动态地图”。

下表按应用程序类型（Web 或移动应用程序）提供了有关语言和框架的地图渲染 SDKs 版本的信息。

Web frontend

以下地图渲染可用 SDKs 于 Web 前端应用程序开发。

语言/框架	地图渲染框架
完全支持	
JavaScript	https://maplibre。 org/projects/maplibre-gl-js/
ReactJS	https://github.com/maplibre/maplibre-react-native
TypeScript	https://maplibre。 org/projects/maplibre-gl-js/
部分支持	
Flutter	https://github.com/maplibre/flutter-maplibre-gl MapLibre Flutter 库被认为是实验性的。
Node.js	不 MapLibre 支持 Node.js。
PHP	不 MapLibre 支持 PHP。

Mobile frontend

以下地图渲染 SDKs 可用于移动前端应用程序开发。

语言/框架	地图渲染框架
完全支持	
Java	https://maplibre。 org/projects/maplibre-native/
Kotlin	https://maplibre。 org/projects/maplibre-native/ 需要自定义绑定，就像基 MapLibre 于 Java 的绑定一样。
ObjectiveC	https://maplibre。 org/projects/maplibre-native/
ReactNative	https://github.com/maplibre/maplibre-react-native

语言/框架	地图渲染框架
Swift	https://maplibre。 org/projects/maplibre-native/
部分支持	
Flutter	https://github.com/maplibre/flutter-maplibre-gl MapLibre Flutter 库被认为是实验性的。

Amazon Location Service API 和 CLI

Amazon Location Service 提供 API 和 CLI 操作对定位功能的访问权限。有关更多信息，请参阅下面的列表。

亚马逊 Location Service API

这包括以下内容 APIs：

- 地方
- 路线
- 身份验证
- 映射
- 地理围栏
- 追踪器
- 标签

Amazon Location Service CLI

这包括以下内容 CLIs：

Amazon Location Service 的 AWS CLI 操作

Amazon Location Service 提供 AWS CLI（命令行界面）操作来访问定位功能，包括以下内容： APIs

位数

- 自动完成

- 地理编码
- get-place
- 反向地理编码
- 在附近搜寻
- 搜索文本
- 了解更多

路线

- 计算等值线
- calculate-route-matrix
- 计算路线
- 优化航点
- snap-to-roads
- 了解更多

身份验证

- 创建密钥
- 删除密钥
- 描述密钥
- 列表键
- 更新密钥
- 了解更多

映射

- get-glyphs
- get-sprites
- get-static-map
- get-style-descriptor
- get-tile
- 了解更多

地理围栏

- batch-delete-geofence

- batch-evaluate-geofences
- batch-put-geofence
- forecast-geofence-events
- create-geofence-collection
- delete-geofence-collection
- describe-geofence-collection
- list-geofence-collections
- update-geofence-collection
- get-geofence
- 列表地理围栏
- put-geofence
- 了解更多

跟踪器

- batch-get-device-position
- batch-update-device-position
- batch-delete-device-position-历史
- get-device-position
- get-device-position-history
- associate-tracker-consumer
- disassociate-tracker-consumer
- 创建追踪器
- 删除追踪器
- 描述追踪器
- 列表追踪器
- 更新追踪器
- list-tracker-consumers
- verify-device-position
- 了解更多

标签

- list-tags-for-resource
- tag-resource
- untag-resource
- 了解更多

示例和学习资源

本主题提供有关我们可用演示和示例项目的链接和详细信息。这些资源旨在帮助您快速了解和实现我们工具的关键功能，以及 APIs. 此外，您还可以找到包含源代码的 GitHub 存储库的链接和示例解决方案。

演示

- Web
  - 访问：https://location.aws.com/demo
  - 浏览源代码：https://github.com/aws-geospatial/amazon-location-features-demo-web
- Android
  - 安装：https://play.google.com/store/应用程序/详情？ id=com.aws.amazonlocation&trk=devguid
  - 浏览源代码：https://github.com/aws-geospatial/amazon-location-features-demo-安卓
- iOS
  - 要使用我们的应用程序，请执行以下操作：
    - 安装 TestFlight
    - 加入 Amazon 定位测试版
  - 浏览源代码：https://github.com/aws-geospatial/amazon-location-features-demo-ios

样本

1. 访问：https://location.aws.com/samples。
2. 浏览源代码：
   - JS: https://github.com/aws-geospatial/amazon-location-samples-js/
   - 反应：https://github.com/aws-geospatial/amazon-location-samples-react
   - 斯威夫特：https://github.com/aws-geospatial/amazon-location-samples-ios

- 安卓：https://github.com/aws-geospatial/amazon-location-samples-android

GitHub

请参阅https://github.com/aws-geospatial/。

AWS 亚马逊 Location Service 集成

Amazon Location Service 与各种 AWS 服务集成，可实现高效的身份验证、监控、管理和开发。

监控

- Amazon CloudWatch — 查看有关服务使用情况和运行状况的指标，包括请求、延迟、故障和日志。有关更多信息，请参阅 the section called “使用 Amazon 进行监控 CloudWatch”。
- AWS CloudTrail— 记录和监控您的 API 调用，包括用户、角色或 AWS 服务采取的操作。有关更多信息，请参阅 the section called “使用监控和登录 AWS CloudTrail”。

管理

- AWS CloudFormation— Amazon Location 与一项服务集成 AWS CloudFormation，可帮助您对 AWS 资源进行建模和设置，从而减少创建和管理资源和基础设施所花费的时间。有关更多信息，请参阅 the section called “使用创建资源 AWS CloudFormation”。
- 服务配额-使用 Service Quota AWS CLI s 控制台请求更改您的可调整配额。有关更多信息，请参阅 the section called “管理配额”。
- 标签 — 使用 Amazon Location 中的资源标签创建标签，按用途、所有者、环境或标准对资源进行分类。标记您的资源可以帮助您管理、识别、组织、搜索和筛选资源。有关更多信息，请参阅 the section called “使用标签管理资源”。

身份验证

- Amazon Cognito — 您可以使用亚马逊 Cognito 身份验证作为直接 AWS Identity and Access Management 使用 (IAM) 处理 SDKs 前端和直接 HTTPS 请求的替代方案。有关更多信息，请参阅 the section called “使用 Amazon Cognito”。
- IAM — AWS Identity and Access Management (IAM) 是一项 AWS 服务，可帮助管理员安全地控制对 AWS 资源的访问。IAM 管理员控制谁可以接受身份验证（登录）和授权（有权限）使用 Amazon Location Service 资源。有关更多信息，请参阅 the section called “使用 IAM”。

增值

- Amazon EventBridge — 启用事件驱动的应用程序架构，以便您可以使用 AWS Lambda 函数激活应用程序和工作流程的其他部分。有关更多信息，请参阅 the section called “通过以下方式对事件做出反应 EventBridge”。
- AWS IoT— AWS IoT Core 规则引擎存储有关设备消息主题的查询，并允许您定义向其他 AWS 服务（例如 Amazon Location Service）发送消息的操作。知道自己位置为坐标的设备可以通过规则引擎将其位置转发到 Amazon Location 。有关更多信息，请参阅 the section called “使用 AWS IoT 和 MQTT 进行跟踪”。

开发人员工具

- SDKs— Amazon Location Service 为开发人员提供了各种工具，用于构建支持定位的应用程序。其中包括标准版 AWS SDKs、移动版和网络版 SDKs。有关更多信息，请参阅 the section called “SDKs 和框架”。
- AWS CLI— AWS Command Line Interface (AWS CLI) 是一个开源工具，可让您使用命令行 shell 中的命令与 AWS 服务进行交互。只需最少的配置。有关更多信息，请参阅AWS Command Line Interface或了解更多信息AWS CLI。
- 示例代码 — 使用 AWS SDKs移动设备和网页 SDKs版的示例代码 MapLibre 来演示如何使用 Amazon Location。有关更多信息，请参阅示例。
- Amazon Location Service 控制台 — 使用 Amazon Location 控制台来了解 APIs、获取资源，并开始使用可视化的交互式学习工具。有关更多信息，请参阅 Amazon Location Service 控制台。

费用和账单

- AWS B@@ illing and Cost Managem ent — 服务可帮助您支付账单和优化成本。Amazon Web Services 会向您的账户收取使用费，从而确保您只需为实际用量付费。有关更多信息，请参阅 the section called “定价模式”或 the section called “管理账单和费用”。

资源管理

资源管理提供了用于管理配额、使用标签组织资源、控制成本和自动创建资源的工具和流程 AWS CloudFormation。这些功能使您能够在 Amazon Location 中高效分配、监控和管理您的资源。

使用这些工具通过设置服务限制、标记资源以实现更好的组织、监控开支以及使用基础设施即代码 CloudFormation 以编程方式创建和管理资源，从而保持运营效率。

主题

- 使用 Service Quotas 管理配额
- 使用标签管理资源
- 使用 Billing and Cost Management 管理 AWS 账单和成本
- 使用创建资源 AWS CloudFormation

使用 Service Quotas 管理配额

> **Note**
>
> 如果您需要更高的配额，可以使用服务配额控制台请求增加可调配额的配额。申请增加配额时，请选择您需要增加配额的区域，因为大多数配额都是特定于该 AWS 地区的。

Service Quotas 控制台允许您请求增加配额或减少可调整配额的配额。Service Quotas 是每个 AWS 账户和每个 AWS 地区可以拥有的最大 API 调用或资源数量。Amazon Location Service 拒绝超过服务配额的其他请求。

速率限制（以...速率开头的配额）是为每个 API 操作定义的每秒最大请求数，每秒钟任意部分内的突发速率为限制的 80%。Amazon Location Service 会限制超过操作速率限制的请求。

管理您的 Amazon Location Service 配额

Amazon Location Service 与 AWS 服务配额集成，该服务使您能够从中央位置查看和管理配额。有关更多信息，请参阅什么是服务配额？ 在 S ervice Quotas 用户指南中。

使用 Service Quotas，可使用轻松查找 Amazon Location Service 配额的值。

AWS Management Console

1. 打开服务限额控制台。
2. 在导航窗格中，选择 AWS 服务。
3. 从 AWS 服务列表中，搜索并选择 Amazon Location。
4. 在服务配额列表中，您可以看到服务配额名称、应用的值（如果可用）、 AWS 默认配额以及配额值是否可调整。
5. 要查看有关服务配额的其他信息（如描述），请选择配额名称。
6. （可选）要请求增加配额，请选择要增加的配额，选择 Request quota increase（请求增加配额），输入或选择所需信息，然后选择 Request（请求）。

要使用控制台更多地处理服务配额，请参阅 S ervice Quotas 用户指南。要请求提高配额，请参阅《服务配额用户指南》中的请求提高配额。

AWS CLI

运行以下命令查看 Amazon Location 原定设置配额。

```
aws service-quotas list-aws-default-service-quotas \
    --query 'Quotas[*].
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
    --service-code geo \
    --output table
```

要使用更多地使用服务配额 AWS CLI，请参阅《AWS CLI 命令参考》中的 “Service Quotas” 部分。要申请增加配额，请参阅《AWS CLI 命令参考》`request-service-quota-increase`中的。

常见问题解答

Amazon Location Service 资源的默认配额是多少？

Amazon Location Service 为设置默认配额 APIs 以帮助管理服务容量，可以在AWS 服务配额管理控制台中查看这些配额。您可以在下面的the section called “监控您的 Amazon 定位服务配额”部分中找到所有这些内容的链接。

如何申请增加 Amazon Location Service 的配额？

您可以通过自助控制台请求增加 Amazon Location Service 配额，每个 API 的配额最多为默认限额的 2 倍。如果配额限制超过默认限制的 2 倍，请在自助服务控制台中申请，自助服务控制台将提交支持请求。或者，您可以联系您的高级支持团队

为亚马逊定位服务（Amazon Location Service）申请更高的配额会产生额外费用吗？

增加配额的请求不会直接收费，但是根据消耗的额外资源，更高的使用级别可能会导致服务成本增加。

监控您的 Amazon 定位服务配额

您可以在 Amazon 上根据配额监控您的使用情况 CloudWatch。有关更多信息，请参阅 the section called “使用 Amazon 进行监控 CloudWatch”。

名称	默认值	可调整	描述
每个账户的 API 密钥资源	每个受支持的区域：500 个	否	每个账户可以拥有的最大 API 密钥资源（活跃或已过期）数量。

名称	默认值	可调整	描述
每个账户的地理围栏集合资源	每个受支持的区域：1,500 个	是	对于每个账户可以创建的地理围栏集合资源的最大数量。
每个地理围栏集合的地理围栏数	每个受支持的区域：5 万个	否	对于每个地理围栏集合可以创建的地理围栏的最大数量。
每个账户的映射资源	每个受支持的区域：40 个	是	对于每个账户可以创建的映射资源的最大数量。
每个账户的位置索引资源	每个受支持的区域：40 个	是	对于每个账户可以创建的位置索引资源的最大数量。
AssociateTrackerConsumer API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 AssociateTrackerConsumer 请求数。额外的请求将被阻止。
BatchDeleteDevicePositionHistory API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 BatchDeleteDevicePositionHistory 请求数。额外的请求将被阻止。
BatchDeleteGeofence API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 BatchDeleteGeofence 请求数。额外的请求将被阻止。
BatchEvaluateGeofences API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 BatchEvaluateGeofences 请求数。额外的请求将被阻止。

名称	默认值	可调整	描述
BatchGetDevicePosition API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 BatchGetDevicePosition 请求数。额外的请求将被阻止。
BatchPutGeofence API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 BatchPutGeofence 请求数。额外的请求将被阻止。
BatchUpdateDevicePosition API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 BatchUpdateDevicePosition 请求数。额外的请求将被阻止。
CalculateRoute API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 CalculateRoute 请求数。额外的请求将被阻止。
CalculateRouteMatrix API 请求的速率	每个受支持的区域：每秒 5 个	是	您每秒可以发出的最大 CalculateRouteMatrix 请求数。额外的请求将被阻止。
CreateGeofenceCollection API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 CreateGeofenceCollection 请求数。额外的请求将被阻止。
CreateKey API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 CreateKey 请求数。额外的请求将被阻止。

名称	默认值	可调整	描述
CreateMap API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 CreateMap 请求数。额外的请求将被阻止。
CreatePlaceIndex API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 CreatePlaceIndex 请求数。额外的请求将被阻止。
CreateRouteCalculator API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 CreateRouteCalculator 请求数。额外的请求将被阻止。
CreateTracker API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 CreateTracker 请求数。额外的请求将被阻止。
DeleteGeofenceCollection API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 DeleteGeofenceCollection 请求数。额外的请求将被阻止。
DeleteKey API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 DeleteKey 请求数。额外的请求将被阻止。
DeleteMap API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 DeleteMap 请求数。额外的请求将被阻止。

名称	默认值	可调整	描述
DeletePlaceIndex API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 DeletePlaceIndex 请求数。额外的请求将被阻止。
DeleteRouteCalculator API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 DeleteRouteCalculator 请求数。额外的请求将被阻止。
DeleteTracker API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 DeleteTracker 请求数。额外的请求将被阻止。
DescribeGeofenceCollection API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 DescribeGeofenceCollection 请求数。额外的请求将被阻止。
DescribeKey API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 DescribeKey 请求数。额外的请求将被阻止。
DescribeMap API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 DescribeMap 请求数。额外的请求将被阻止。
DescribePlaceIndex API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 DescribePlaceIndex 请求数。额外的请求将被阻止。

名称	默认值	可调整	描述
DescribeRouteCalculator API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 DescribeRouteCalculator 请求数。额外的请求将被阻止。
DescribeTracker API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 DescribeTracker 请求数。额外的请求将被阻止。
DisassociateTrackerConsumer API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 DisassociateTrackerConsumer 请求数。额外的请求将被阻止。
ForecastGeofenceEvents API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 ForecastGeofenceEvents 请求数。额外的请求将被阻止。
GetDevicePosition API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 GetDevicePosition 请求数。额外的请求将被阻止。
GetDevicePositionHistory API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 GetDevicePositionHistory 请求数。额外的请求将被阻止。
GetGeofence API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 GetGeofence 请求数。额外的请求将被阻止。

名称	默认值	可调整	描述
GetMapGlyphs API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 GetMapGlyphs 请求数。额外的请求将被阻止。
GetMapSprites API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 GetMapSprites 请求数。额外的请求将被阻止。
GetMapStyleDescriptor API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 GetMapStyleDescriptor 请求数。额外的请求将被阻止。
GetMapTile API 请求的速率	每个受支持的区域：每秒 500 个	是	您每秒可以发出的最大 GetMapTile 请求数。额外的请求将被阻止。
GetPlace API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 GetPlace 请求数。额外的请求将被阻止。
ListDevicePositions API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 ListDevicePositions 请求数。额外的请求将被阻止。
ListGeofenceCollections API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 ListGeofenceCollections 请求数。额外的请求将被阻止。
ListGeofences API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 ListGeofences 请求数。额外的请求将被阻止。

名称	默认值	可调整	描述
ListKeys API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 ListKeys 请求数。额外的请求将被阻止。
ListMaps API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 ListMaps 请求数。额外的请求将被阻止。
ListPlaceIndexes API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 ListPlaceIndexes 请求数。额外的请求将被阻止。
ListRouteCalculators API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 ListRouteCalculators 请求数。额外的请求将被阻止。
ListTagsForResource API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 ListTagsForResource 请求数。额外的请求将被阻止。
ListTrackerConsumers API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 ListTrackerConsumers 请求数。额外的请求将被阻止。
ListTrackers API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 ListTrackers 请求数。额外的请求将被阻止。

名称	默认值	可调整	描述
PutGeofence API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 PutGeofence 请求数。额外的请求将被阻止。
SearchPlaceIndexForPosition API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 SearchPlaceIndexForPosition 请求数。额外的请求将被阻止。
SearchPlaceIndexForSuggestions API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 SearchPlaceIndexForSuggestions 请求数。额外的请求将被阻止。
SearchPlaceIndexForText API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 SearchPlaceIndexForText 请求数。额外的请求将被阻止。
TagResource API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 TagResource 请求数。额外的请求将被阻止。
UntagResource API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 UntagResource 请求数。额外的请求将被阻止。
UpdateGeofenceCollection API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 UpdateGeofenceCollection 请求数。额外的请求将被阻止。

名称	默认值	可调整	描述
UpdateKey API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 UpdateKey 请求数。额外的请求将被阻止。
UpdateMap API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 UpdateMap 请求数。额外的请求将被阻止。
UpdatePlaceIndex API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 UpdatePlaceIndex 请求数。额外的请求将被阻止。
UpdateRouteCalculator API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 UpdateRouteCalculator 请求数。额外的请求将被阻止。
UpdateTracker API 请求的速率	每个受支持的区域：每秒 10 个	是	您每秒可以发出的最大 UpdateTracker 请求数。额外的请求将被阻止。
VerifyDevicePosition API 请求的速率	每个受支持的区域：每秒 50 个	是	您每秒可以发出的最大 VerifyDevicePosition 请求数。额外的请求将被阻止。
地理地图的速率：API 请求 GetStaticMap	每个受支持的区域：每秒 50 个	是	地理地图的最大数量：您每秒可以发出的GetStaticMap 请求。额外的请求将被阻止。

名称	默认值	可调整	描述
地理地图的速率：API 请求 GetTile	每个受支持的区域：每秒 2,000 个	是	地理地图的最大数量：您每秒可以发出的GetTile请求。额外的请求将被阻止。
地理位置比率:自动完成 API 请求	每个受支持的区域：每秒 100 个	是	您每秒可以发出的 Geo-Places: AutoComplete 请求的最大数量。额外的请求将被阻止。
地理位置比率:Geocode API 请求率	每个受支持的区域：每秒 100 个	是	您每秒可以发出的 Geo-Places: Geocode 请求的最大数量。额外的请求将被阻止。
地理位置比率：GetPlace API 请求	每个受支持的区域：每秒 100 个	是	地理位置的最大数量：您每秒可以发出的GetPlace请求。额外的请求将被阻止。
地理位置比率：ReverseGeocode API 请求	每个受支持的区域：每秒 100 个	是	地理位置的最大数量：您每秒可以发出的ReverseGeocode 请求。额外的请求将被阻止。
地理位置比率：SearchNearby API 请求	每个受支持的区域：每秒 100 个	是	地理位置的最大数量：您每秒可以发出的SearchNearby 请求。额外的请求将被阻止。
地理位置比率：SearchText API 请求	每个受支持的区域：每秒 100 个	是	地理位置的最大数量：您每秒可以发出的SearchText 请求。额外的请求将被阻止。

名称	默认值	可调整	描述
地理位置比率:建议 API 请求	每个受支持的区域：每秒 100 个	是	您每秒可以发出的最大地理位置：建议请求数。额外的请求将被阻止。
地理路线速率：CalculateIsolines API 请求	每个受支持的区域：每秒 20 个	是	地理路径的最大数量：您每秒可以发出的CalculateIsolines 请求。额外的请求将被阻止。
地理路线速率：CalculateRouteMatrix API 请求	每个受支持的区域：每秒 5 个	是	地理路径的最大数量：您每秒可以发出的CalculateRouteMatrix 请求。额外的请求将被阻止。
地理路线速率：CalculateRoutes API 请求	每个受支持的区域：每秒 20 个	是	地理路径的最大数量：您每秒可以发出的CalculateRoutes 请求。额外的请求将被阻止。
地理路线速率：OptimizeWaypoints API 请求	每个受支持的区域：每秒 5 个	是	地理路径的最大数量：您每秒可以发出的OptimizeWaypoints 请求。额外的请求将被阻止。
地理路线速率：SnapToRoads API 请求	每个受支持的区域：每秒 20 个	是	地理路径的最大数量：您每秒可以发出的SnapToRoads 请求。额外的请求将被阻止。
每个账户的路线计算器资源	每个受支持的区域：40 个	是	对于每个账户可以创建的路线计算器资源的最大数量。

名称	默认值	可调整	描述
每个跟踪器的跟踪器使用者人数	每个受支持的区域：5 个	否	可以与跟踪器资源关联的地理围栏集合的最大数量。
每个账户的跟踪器资源	每个受支持的区域：500 个	是	对于每个账户可以创建的跟踪器资源的最大数量。

了解更多

要了解有关服务配额的更多信息，请参阅 Ser vice Quotas 文档。

使用标签管理资源

使用 Amazon Location 中的资源标签来创建标签，按用途、拥有者、环境或标准对您的资源进行分类。标记您的资源可以帮助您管理、识别、组织、搜索和筛选资源。例如，使用 Res AWS ource Groups，您可以基于一个或多个标签或部分标签创建 AWS 资源组。您也可以根据群组在 AWS CloudFormation 堆栈中的出现情况来创建群组。使用 Resource Groups 和标签编辑器，您可以在一个位置整合和查看由多个服务、资源和区域组成的应用程序的数据。有关常用标记策略的更多信息，请参阅《AWS 一般参考》。

每个标签都是包含您定义的一个键和值的标记：

- 标签密钥 – 对标签值进行分类的通用标签。例如 `CostCenter`。
- 标签值 – 标签密钥类别的可选描述。例如 `MobileAssetTrackingResourcesProd`。

本主题通过查看标签限制来帮助您开始标记。它还向您展示了如何使用 AWS 成本分配报告创建标签和使用标签来跟踪每个有效标签的成本。

有关以下内容的更多信息：

- 标记最佳实践，请参阅《AWS 一般参考》中的为AWS 资源添加标签。
- 使用标签控制对 AWS 资源的访问，请参阅《Identity and Access Managem AWS ent 用户指南》中的使用标签控制对 AWS 资源的访问权限。

限制

> **Note**
>
> 如果您添加的新标签的标签键与现有标签的相同，则新标签将覆盖现有标签。

使用标签，使您能够更有效地组织和管理您的资源。本页概述了在 Amazon Location Service 中管理标签使用的具体规则和限制。通过了解这些标记限制，便可以确保遵守最佳实践，并避免在为基于位置的资源和应用程序实施标记策略时出现潜在问题。

下面是适用于标签的基本限制：

- 每个资源的最大标签数：50
- 对于每个资源，每个标签键都必须是唯一的，每个标签键只能有一个值。
- 最大键长度：128 个 Unicode 字符（采用 UTF-8 格式）
- 最大值长度：256 个 Unicode 字符（采用 UTF-8 格式）
- 允许在不同的服务中使用的字符包括：可以使用 UTF-8 表示的字母、数字和空格以及以下字符：+ - = . _ : / @。
- 标签键和值区分大小写。
- 该`aws:`前缀已保留供 AWS 使用。如果某个标签具有带有此标签键，则您无法编辑该标签的键或值。带有 aws: 前缀的标签不计入每个资源的标签限制。

授予标记资源的权限

您可以使用 IAM policy 来控制对您的 Amazon Location 资源的访问权限，并授予在创建资源时标记资源的权限。除了授予创建资源的权限外，该策略还可以包括允许标记操作的操作权限：

- `geo:TagResource` – 允许用户将一个或多个标签分配给指定的 Amazon Location 资源。
- `geo:UntagResource` – 允许用户从指定的 Amazon Location 资源中删除一个或多个标签。
- `geo:ListTagsForResource` – 允许用户列出分配给 Amazon Location 资源的所有标签。

以下是允许用户创建地理围栏集合并标记资源的策略示例：

```
{
"Version": "2012-10-17",
```

```
    "Statement": [
        {
"Sid": "AllowTaggingForGeofenceCollectionOnCreation",
            "Effect": "Allow",
            "Action": [
                "geo:CreateGeofenceCollection",
                "geo:TagResource"
            ],
            "Resource": "arn:aws:geo:region:accountID:geofence-collection/*"
    ]
}
```

将标签添加到资源中

您可以使用亚马逊位置控制台、 AWS CLI 或亚马逊位置在创建资源时添加标签 APIs：

- the section called “管理 资源”
- the section called “创建跟踪器”

为现有资源添加、编辑或删除标签

1. 打开 Amazon Location 控制台。
2. 在左侧导航窗格中，选择要标记的资源。例如，地图。
3. 从列表中选择一个资源。
4. 选择管理标签以添加、编辑或删除标签。

如何使用标签

您可以使用标签进行成本分配，以详细跟踪您的 AWS 成本。激活成本分配标签后， AWS 使用成本分配标签在成本分配报告中整理资源账单。这可以帮助您对使用成本进行分类和跟踪。

Amazon Location 支持用户定义的-标签。这些是您创建的自定义标签。用户定义的标签使用 `user:` 前缀，例如 `user:CostCenter`。

必须单独激活每种标签类型。激活标签后，您可以启用 AWS Cost Explorer 或查看每月成本分配报告。

激活用户定义的标签：

1. 打开账单和成本管理控制台。

2. 在左侧导航窗格中，选择成本分配标签。
3. 在用户定义的成本分配标签选项卡下，选择要激活的标签密钥。
4. 选择激活。

激活标签后， AWS 会针对您的资源使用情况和成本生成每月成本分配报告。此成本分配报告包括您在每个账单周期的所有 AWS 费用，包括带标签和未标记的资源。有关更多信息，请参阅《AWS 账单用户指南》中的使用 AWS 成本分配标签组织和跟踪成本。

使用标签控制对 资源的访问

AWS Identity and Access Management (IAM) 策略支持基于标签的条件，这使您能够根据特定的标签键和值管理资源的授权。例如，IAM 角色策略可以包含条件来限制对特定环境的访问，如基于标签的开发、测试或生产环境。想要了解更多信息，请参阅基于标签控制资源访问的主题。

使用 Billing and Cost Management 管理 AWS 账单和成本

AWS Billing and Cost Management 是一项网络服务，它提供的功能可帮助您支付账单和优化成本。Amazon Web Services 会向您的账户收取使用费，从而确保您只需为实际用量付费。

如何查看账单和管理成本

1. 在中打开 “账单和成本管理” AWS Management Console。
2. 在 Amazon Web Services, Inc. 中搜索定位服务按服务收费
3. 选择 [+] 定位服务
4. 选择 [+] 地区名称

要了解更多信息，请参阅中的账单和成本管理 AWS Management Console。

使用创建资源 AWS CloudFormation

Amazon Location Service 与 AWS CloudFormation一项服务集成，可帮助您对 AWS 资源进行建模和设置，从而减少创建和管理资源和基础设施所花费的时间。您可以创建一个描述所需的全部 AWS 资源的模板（例如 Amazon Location 资源），将为您 AWS CloudFormation 预置和配置这些资源。

使用时 AWS CloudFormation，您可以重复使用模板来一致且重复地设置您的亚马逊位置资源。描述一次您的资源，然后在多个 AWS 账户和区域中一遍又一遍地配置相同的资源。

相关 AWS CloudFormation 模板

要为 Amazon Location 和相关服务设置和配置资源，您必须了解 AWS CloudFormation 模板。模板是 JSON 或 YAML 格式的文本文件。这些模板可描述您要在 AWS CloudFormation 堆栈中预置的资源。如果你不熟悉 JSON 或 YAML，可以使用 D AWS CloudFormation esigner 来帮助你开始使用 AWS CloudFormation 模板。有关更多信息，请参阅什么是 AWS CloudFormation 设计器？ 在《AWS CloudFormation 用户指南》中。

Amazon Location 支持在 AWS CloudFormation中创建以下资源类型：

- AWS:: 位置:: 追踪器
- AWS:: 地点:: TrackerConsumer
- AWS:: 地点:: GeofenceCollection

有关更多信息（包括 Amazon Location 资源的 JSON 和 YAML 模板示例），请参阅 AWS CloudFormation 用户指南中的 Amazon Location Service 资源类型参考。

了解更多关于 AWS CloudFormation

要了解更多信息 AWS CloudFormation，请参阅以下资源：

- AWS CloudFormation
- AWS CloudFormation 用户指南
- AWS CloudFormation API 引用
- AWS CloudFormation 命令行界面用户指南

监控和审计

监控和审计提供了在您的亚马逊定位服务环境中跟踪、监控和记录活动的功能。借助 Amazon CloudWatch 和 AWS CloudTrail，您可以确保应用程序的可靠性、安全性和合规性。

这些工具可帮助您观察资源性能指标、检测异常并记录 API 活动以进行审计和故障排除。使用它们来增强运营洞察力、诊断问题并确保遵守合规标准。

主题

- 使用 Amazon 进行监控 CloudWatch
- 使用监控和登录 AWS CloudTrail

使用 Amazon 进行监控 CloudWatch

Amazon 近乎实时地 CloudWatch 监控您的 AWS 资源和您运行 AWS 的应用程序。您可以使用监控 Amazon Location 资源 CloudWatch，该资源可以近乎实时地收集原始数据并将指标处理成有意义的统计数据。您可以查看长达 15 个月的历史信息，也可以搜索指标以在 Amazon CloudWatch 控制台中查看，以更深入地了解您的应用程序或服务的性能。您还可以通过定义阈值来设置警报，并在达到相应阈值时发送通知或执行操作。

有关更多信息，请参阅 Amazon CloudWatch 用户指南

主题

- Amazon Location Service 的指标和维度
- 查看 Amazon Location Service
- 为 Amazon Location Service 指标创建 CloudWatch 警报
- 用于 CloudWatch 根据配额监控使用情况
- CloudWatch Amazon Location Service 的指标示例

Amazon Location Service 的指标和维度

指标是导出到 CloudWatch的时间顺序数据点。维度是用于标识指标的名称/值对。有关更多信息，请参阅 Amazon CloudWatch 用户指南中的使用 CloudWatch 指标和CloudWatch 维度。

> **Note**
>
> 由于 Amazon Location Service 采用分布式架构，因此结果是近似值。在大多数情况下，该计数应接近发送的 API 操作的实际数量。

Amazon Location Service 指标

以下是 Amazon Location Service 在`AWS/Location`命名空间 CloudWatch 中导出的指标。

指标	描述	Dimensions
`CallCount`	对给定 API 端点的调用次数。 有效统计数据：Sum	OperationName OperationName, ResourceName

指标	描述	Dimensions
	单位：计数	ApiKeyName, OperationName ApiKeyName, OperationName, ResourceName OperationName, OperationVersion OperationName, OperationVersion, ResourceName ApiKeyName, OperationName, OperationVersion ApiKeyName, OperationName, OperationVersion, ResourceName
`ErrorCount`	对给定 API 端点的调用所产生的错误响应数量。 有效统计数据：Sum 单位：计数	OperationName OperationName, ResourceName ApiKeyName, OperationName ApiKeyName, OperationName, ResourceName
`SuccessCount`	成功调用给定 API 端点的次数。 有效统计数据：Sum 单位：计数	OperationName OperationName, ResourceName ApiKeyName, OperationName ApiKeyName, OperationName, ResourceName

指标	描述	Dimensions
CallLatency	向给定 API 端点发出调用时，该操作处理和返回响应所花费的时间。 有效统计数据：平均值 单位：毫秒	OperationName OperationName, ResourceName ApiKeyName, OperationName ApiKeyName, OperationName, ResourceName

Amazon Location Service 指标维度

您可以使用下表中的维度来筛选 Amazon Location Service 指标。

维度	描述
OperationName	使用指定的操作名称筛选 API 操作的 Amazon 位置指标。
OperationName, ResourceName	使用指定的操作名称和资源名称筛选 API 操作的 Amazon 位置指标。
ApiKeyName, OperationName	使用指定的操作名称并使用给定的 API 密钥名称筛选 API 操作的 Amazon 位置指标。
ApiKeyName, OperationName, ResourceName	使用指定的操作名称、资源名称并使用给定的 API 密钥名称筛选 API 操作的 Amazon 位置指标。
OperationName, OperationVersion	使用指定的操作名称筛选 API 操作的 Amazon 位置指标。 Amazon Location Service 的独立地图、地点和路线将成为该维度的导出指标。
OperationName, OperationVersion, ResourceName	使用指定的操作名称、版本和亚马逊位置资源名称筛选 API 操作的亚马逊位置指标。 Amazon Location 独立地图、地点和路线将成为该维度的导出指标。

维度	描述
ApiKeyName, OperationName, OperationVersion	使用指定的操作名称、版本和给定的 API 密钥名称筛选 API 操作的 Amazon 位置指标。 Amazon Location 独立地图、地点和路线将成为该维度的导出指标。
ApiKeyName, OperationName, OperationVersion, ResourceName	使用指定的操作名称、版本、资源名称并使用给定的 API 密钥名称筛选 API 操作的 Amazon 位置指标。 Amazon Location 独立地图、地点和路线将成为该维度的导出指标。

查看 Amazon Location Service

您可以在 Amazon CloudWatch 控制台上或使用 Amazon CloudWatch API 查看 Amazon Location Service 的指标。

使用 CloudWatch 控制台查看指标

Example

1. 打开 CloudWatch 控制台，网址为https://console.aws.amazon.com/cloudwatch/。
2. 在导航窗格中，选择指标。
3. 在所有指标选项卡上，选择位置命名空间。
4. 选择要查看的指标类型。
5. 选择指标，然后添加到图表。

有关更多信息，请参阅 Amazon CloudWatch 用户指南中的查看可用指标。

为 Amazon Location Service 指标创建 CloudWatch 警报

您可以使用 CloudWatch 对您的 Amazon Location Service 指标设置警报。例如，您可以在中创建警报， CloudWatch 以便在错误计数出现峰值时发送电子邮件。

以下主题从较高层面上概括介绍了如何使用 CloudWatch 设置警报。有关详细说明，请参阅 Amazon CloudWatch 用户指南中的使用警报。

使用 CloudWatch 控制台设置警报

Example

1. 打开 CloudWatch 控制台，网址为https://console.aws.amazon.com/cloudwatch/。
2. 在导航窗格中，选择警报。
3. 选择创建警报。
4. 选择选择指标。
5. 在所有指标选项卡上，选择位置命名空间。
6. 选择指标类别。
7. 找到您要为其创建警报的指标所在行，然后选中该行旁边的复选框。
8. 选择选择指标。
9. 在 “指标” 下，填写值。
10指定警报条件。
11选择下一步。
12如果您想在满足警报条件时发送通知，请执行以下操作：
   - 在警报状态触发器下，选择警报状态以提示发送通知。
   - 在选择 SNS 主题下，选择创建新主题以创建新的 Amazon Simple Notification Service (Amazon SNS) 主题。输入主题名称和要向其发送通知的电子邮件。
   - 在发送通知下，输入要向其发送通知的其他电子邮件地址。
   - 选择 Add notification（添加通知）。此列表将保存下来并会在将来的警报字段中显示出来。
13完成后选择下一步。
14输入警报的名称和描述，然后选择下一步。
15确认警报详情，然后选择下一步。

> **Note**
>
> 创建新的 Amazon SNS 主题时，必须先验证电子邮件地址，然后才能发送通知。如果电子邮件未通过验证，则当警报因状态变化而启动时，将不会收到通知。

有关如何使用 CloudWatch 控制台设置警报的更多信息，请参阅 Amazon CloudWatch 用户指南中的创建发送电子邮件的警报。

用于 CloudWatch 根据配额监控使用情况

您可以创建 Amazon CloudWatch 警报，以便在给定配额的使用率超过可配置的阈值时通知您。这使您能够识别何时接近配额限制，并调整利用率以避免成本超支，或者在需要时请求增加配额。

有关 CloudWatch 如何使用监控配额的信息，请参阅 Amazon CloudWatch 用户指南中的可视化服务配额和设置警报。

CloudWatch Amazon Location Service 的指标示例

您可以使用 GetMetricDataAPI 来检索 Amazon 位置的指标。

- 例如，您可以在数字下降时监控 `CallCount` 并设置警报。

  监控 `SendDeviceLocation` 的 `CallCount` 指标可以帮助您深入了解被跟踪的资产。如果 `CallCount` 下降，则意味着被追踪的资产（例如卡车车队）已停止发送其当前位置。为此设置警报可以帮助通知您发生了问题。
- 再举一个例子，您可以监控 `ErrorCount` 并设置警报，以了解何时出现数量激增。

  跟踪器必须与地理围栏集合相链接，才能根据地理围栏评估设备位置。如果您的设备群需要持续更新位置，看到 `BatchEvaluateGeofence` 或 `BatchPutDevicePosition` 的 `CallCount` 为零表示更新不再流动。

以下是创建地图资源的示例输出 GetMetricData，其中包含`ErrorCount`用于创建地图资源的指标。`CallCount`

```
{
"StartTime": 1518867432,
  "EndTime": 1518868032,
  "MetricDataQueries": [
    {
"Id": "m1",
      "MetricStat": {
"Metric": {
"Namespace": "AWS/Location",
          "MetricName": "CallCount",
          "Dimensions": [
            {
"Name": "SendDeviceLocation",
              "Value": "100"
            }
```

```
          ]
        },
        "Period": 300,
        "Stat": "SampleCount",
        "Unit": "Count"
      }
    },
    {
"Id": "m2",
      "MetricStat": {
"Metric": {
"Namespace": "AWS/Location",
          "MetricName": "ErrorCount",
          "Dimensions": [
            {
"Name": "AssociateTrackerConsumer",
              "Value": "0"
            }
          ]
        },
        "Period": 1,
        "Stat": "SampleCount",
        "Unit": "Count"
      }
    }
  ]
}
```

使用监控和登录 AWS CloudTrail

AWS CloudTrail 是一项提供用户、角色或服务所执行操作记录的 AWS 服务。 CloudTrail 将所有 API 调用记录为事件。您可以将 Amazon Location Service 与配合使用 CloudTrail 来监控您的 API 调用，其中包括来自亚马逊定位服务控制台的调用和对亚马逊定位服务 API 操作的 AWS SDK 调用。

CloudTrail 在您创建 AWS 账户时会自动启用。当 Amazon Location Service 中发生活动时，该活动会与其他 AWS 服务 CloudTrail 事件一起记录在事件历史记录中。您可以查看、搜索和下载每个 AWS 地区过去 90 天的事件历史记录。

有关的更多信息 CloudTrail，请参阅《AWS CloudTrail 用户指南》。查看活动历史记录不 CloudTrail 收取任何费用。

对于您的 AWS 账户中过去 90 天的持续事件记录，包括来自 Amazon Location Service 的事件，请创建跟踪或 CloudTrail Lake 数据存储。

CloudTrail 步道

跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。当您在中创建跟踪时 AWS Management Console，该跟踪将应用于所有 AWS 区域。跟踪记录来自 AWS 分区中所有区域的事件，并将日志文件传送到您指定的 S3 存储桶。此外，您可以配置其他 AWS 服务，以进一步分析和处理 CloudTrail 日志中收集的事件数据。

有关如何创建跟踪的更多信息，请参阅创建跟踪概述。

有关 CloudTrail 支持的服务和集成的列表，请参阅CloudTrail 支持的服务和集成。

通过创建跟踪，您可以免费将正在进行的管理事件的一份副本传送到 Amazon S3 存储桶。 CloudTrail 但是，会收取 Amazon S3 存储费用。

有关 CloudTrail 定价的更多信息，请参阅AWS CloudTrail 定价。

有关亚马逊 S3 定价的信息，请参阅亚马逊 S3 定价。

CloudTrail 湖泊事件数据存储

CloudTrail Lake 允许你对自己的事件运行基于 SQL 的查询。事件将被聚合到事件数据存储中，它是基于您通过应用高级事件选择器选择的条件的不可变的事件集合。应用于事件数据存储的选择器用于控制哪些事件持续存在并可供您查询。

有关 CloudTrail Lake 的更多信息，请参阅使用 AWS CloudTrail Lake。

CloudTrail 湖泊事件数据存储和查询会产生费用。创建事件数据存储时，您可以选择要用于事件数据存储的定价选项。定价选项决定了摄取和存储事件的成本，以及事件数据存储的默认和最长保留期。

有关 CloudTrail 定价的更多信息，请参阅AWS CloudTrail 定价。

主题

- 中的 Amazon 位置管理事件 CloudTrail
- 中的亚马逊位置数据事件 CloudTrail
- 了解 Amazon Location Service 日志文件条目
- 示例：Amazon 位置管理事件的 CloudTrail 日志文件条目
- 示例：Amazon 位置数据事件的 CloudTrail 日志文件条目

中的 Amazon 位置管理事件 CloudTrail

您可以在活动历史记录中查看 Amazon 位置管理 CloudTrail 事件。这些事件包括管理 Amazon Location 资源和配置的所有 API 调用。有关支持的操作的完整列表，请参阅 Amazon Location Service API 参考资料。

中的亚马逊位置数据事件 CloudTrail

数据事件提供有关直接对资源执行的操作的信息。这些事件（也称为数据平面操作）的容量可能很大。默认情况下，CloudTrail不记录数据事件，CloudTrail 事件历史记录也不记录这些事件。启用数据事件时会产生额外费用。有关 CloudTrail 定价的更多信息，请参阅 AWS CloudTrail 定价。

您可以使用 CloudTrail控制台、AWS CLI 或 CloudTrail API 操作来选择记录数据事件的 Amazon Location 资源类型。有关如何启用和管理数据事件的说明，请参阅使用 AWS 管理控制台记录数据事件和使用 A WS 命令行界面记录数据事件。

下表列出了您可以记录数据事件的 Amazon Location 资源类型：

支持的亚马逊位置数据事件

数据事件类型（控制台）	resources.type 值	数据 APIs 已记录到 CloudTrail
地理地图	AWS::GeoMaps::Provider	参见亚马逊 GeoMaps API 参考
地理位置	AWS::GeoPlaces::Provider	参见亚马逊 GeoPlaces API 参考
地理路线	AWS::GeoRoutes::Provider	参见亚马逊 GeoRoutes API 参考

> **Note**
>
> Amazon Location 不会发布以下 CloudTrail 事件 GeoMaps APIs：`GetStyleDescriptor`、`GetGlyphs`和`GetSprites`。这些 APIs 都是免费的，不需要身份验证。

您可以将高级事件选择器配置为按`eventName`、`readOnly`和`resources.ARN`筛选事件。这可以帮助你只记录那些对你来说重要的事件。有关更多信息，请参阅 AdvancedFieldSelector 。

了解 Amazon Location Service 日志文件条目

配置跟踪时，会将事件作为日志文件 CloudTrail 传输到您指定的 S3 存储桶或 Amazon CloudWatch Logs。有关更多信息，请参阅《 AWS CloudTrail 用户指南》中的使用 CloudTrail 日志文件。

CloudTrail 日志文件可以包含一个或多个日志条目。每个事件条目代表来自任何来源的单个请求，包括请求的操作、操作的日期和时间、请求参数等详细信息。

> **Note**
>
> CloudTrail 日志文件不是 API 调用的有序堆栈跟踪。它们不按时间顺序出现。要确定操作顺序，请使用 `eventTime`。

每个事件或日志条目都包含有关谁发出请求的信息。此身份信息可帮助您确定：

- 请求是使用根凭证还是用户凭证发出的。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其他 AWS 服务发出。

示例：Amazon 位置管理事件的 CloudTrail 日志文件条目

以下示例显示了该`CreateTracker`操作的 CloudTrail 日志条目，该日志条目创建了跟踪器资源。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "111122223333",
        "arn": "arn:aws:geo:us-east-1:111122223333:tracker/ExampleTracker",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "111122223333",
                "arn": "arn:aws:geo:us-east-1:111122223333:tracker/ExampleTracker",
                "accountId": "111122223333",
                "userName": "exampleUser"
            },
            "webIdFederationData": {},
```

```
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-10-22T16:36:07Z"
            }
        }
    },
    "eventTime": "2020-10-22T17:43:30Z",
    "eventSource": "geo.amazonaws.com",
    "eventName": "CreateTracker",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "SAMPLE_IP_ADDRESS",
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.864
 Linux/4.14.193-110.317.amzn2.x86_64 OpenJDK_64-Bit_Server_VM/11.0.8+10-LTS java/11.0.8
 kotlin/1.3.72 vendor/Amazon.com_Inc. exec-env/AWS_Lambda_java11",
    "requestParameters": {
        "TrackerName": "ExampleTracker",
        "Description": "Resource description"
    },
    "responseElements": {
        "TrackerName": "ExampleTracker",
        "Description": "Resource description",
        "TrackerArn": "arn:partition:service:region:account-id:resource-id",
        "CreateTime": "2020-10-22T17:43:30.521Z"
    },
    "requestID": "557ec619-0674-429d-8e2c-eba0d3f34413",
    "eventID": "3192bc9c-3d3d-4976-bbef-ac590fa34f2c",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

示例：Amazon 位置数据事件的 CloudTrail 日志文件条目

以下示例显示了该Geocode操作的 CloudTrail 日志条目，该条目用于检索有关地点的坐标、地址和其他详细信息。

```
{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROA6ODU7M35SFGUCGXHMSAMPLE",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/vingu-Isengard",
    "accountId": "111122223333",
```

```
    "accessKeyId": "ASIA6ODU7M352GLR5CFMSAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROA6ODU7M35SFGUCGXHMSAMPLE",
        "arn": "arn:aws:iam::111122223333:role/Admin",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "attributes": {
        "creationDate": "2024-09-16T14:41:33Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2024-09-16T14:42:16Z",
  "eventSource": "geo-places.amazonaws.com",
  "eventName": "Geocode",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "52.94.133.129",
  "userAgent": "Amazon CloudFront",
  "requestParameters": {
    "Query": "***",
    "Filter": {
      "IncludeCountries": [
        "USA"
      ]
    }
  },
  "responseElements": null,
  "requestID": "1ef7e0b8-c9fc-4a20-80c3-b5340d634c4e",
  "eventID": "913d256c-3a9d-40d0-9bdf-705f12c7659f",
  "readOnly": true,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::GeoPlaces::Provider",
      "ARN": "arn:aws:geoplaces:us-west-2:111122223333:provider"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": false,
  "recipientAccountId": "111122223333",
  "eventCategory": "Data"
```

```
}
```

最佳实践

以下是与 Amazon Location Service 集成的一些最佳实践。

资源管理

为了帮助有效管理您在 Amazon Location Service 中的位置资源，请考虑以下最佳实践：

- 使用对您的预期用户群至关重要的区域终端节点来改善他们的体验。有关区域端点的更多信息，请参阅 the section called “支持的区域”。
- 对于使用数据提供程序的资源，例如地图资源和位置索引资源，请务必遵守特定数据提供程序的使用条款协议。有关更多信息，请参阅 the section called “使用条款和数据属性”。
- 地图、位置索引或路线的每种配置都有一个资源，从而最大限度地减少资源的创建。在一个区域内，每个数据提供程序或地图样式通常只需要一个资源。大多数应用程序使用现有资源，并且不会在运行时创建资源。
- 在单个应用程序中使用不同的资源（例如地图资源和路线计算器）时，请在每个资源中使用相同的数据提供程序来确保数据匹配。例如，使用路线计算器创建的路线几何与使用地图资源绘制的地图上的街道对齐。

账单和成本管理

为了帮助您管理费用和账单，请考虑以下最佳实践：

- 使用诸如 Amazon CloudWatch 之类的监控工具来跟踪您的资源使用情况。您可以设置警报，在使用量即将超过您的指定限制时通知您。有关更多信息，请参阅 Amazon CloudWatch 用户指南中的创建账单警报以监控您的预估 AWS 费用。

配额和使用量

您 AWS 账户 包含的配额会为您的使用量设置默认限制。您可以设置警报，以便在使用量接近上限时提醒您，也可以在需要时请求提高配额。有关如何处理配额的信息，请参阅以下主题。

- the section called “管理配额”
- the section called “创建 CloudWatch 警报”

- 在 Amazon CloudWatch 用户指南中@@ 可视化您的服务配额并设置警报。

您可以创建警报，以便在接近超出限制时提前向您发出警告。我们建议您在使用 Amazon 地点的每个配额中为每个 AWS 区域 配额设置警报。例如，您可以监控 `SearchPlaceIndexForText` 操作的使用情况，并在超过当前配额的 80% 时创建警报。

当你收到有关配额的警报警告时，你必须决定该怎么做。您可能正在使用其他资源，因为您的客户群已经扩大。在这种情况下，您可能需要申请增加配额，例如将该区域的 API 地理围栏配额增加 50%。或者，您的服务中可能存在错误，导致您对 Amazon Location 进行其他不必要的调用。在这种情况下，你需要解决服务中的问题。

Amazon Location Service 中的安全

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方共同承担 AWS 的责任。责任共担模式将其描述为云的安全性和云中的安全性：

- 云安全 — AWS 负责保护在云中运行 AWS 服务的基础架构 AWS Cloud。 AWS 还为您提供可以安全使用的服务。作为AWS 合规计划合规计划合规计划合的一部分，第三方审计师定期测试和验证我们安全的有效性。要了解适用于 Amazon Location Service 的合规计划，请参阅按合规计划AWS 提供的范围内的AWS 服务服务。
- 云端安全-您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

该文档帮助您了解如何在使用 Amazon Location 时应用责任共担模式。以下主题说明如何配置 Amazon Location 以实现您的安全性和合规性目标。您还将学习如何使用其他 AWS 服务来帮助您监控和保护您的 Amazon Location 资源。

主题

- Amazon Location Service 中的数据保护
- Amazon Location Service 中的事件响应
- Amazon Location Service 的合规性验证
- Amazon Location Service 中的韧性
- Amazon Location Service 中的基础设施安全性
- AWS PrivateLink 适用于 Amazon 地点
- Amazon Location 中的配置和漏洞分析
- 防止跨服务混淆代理
- Amazon Location Service 的最佳实践

Amazon Location Service 中的数据保护

分担责任模式 AWS 分担责任模型适用于 Amazon Location Service 中的数据保护。如本模型所述 AWS ，负责保护运行所有内容的全球基础架构 AWS Cloud。您负责维护对托管在此基础结构上的内容的控制。您还负责您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请

参阅数据隐私常见问题。有关欧洲数据保护的信息，请参阅 AWS Security Blog 上的 AWS Shared Responsibility Model and GDPR 博客文章。

出于数据保护目的，我们建议您保护 AWS 账户 凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证（MFA）。
- 使用 SSL/TLS 与资源通信。 AWS 我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用设置 API 和用户活动日志 AWS CloudTrail。有关使用 CloudTrail 跟踪捕获 AWS 活动的信息，请参阅《AWS CloudTrail 用户指南》中的使用跟 CloudTrail 踪。
- 使用 AWS 加密解决方案以及其中的所有默认安全控件 AWS 服务。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在 AWS 通过命令行界面或 API 进行访问时需要经过 FIPS 140-3 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准（FIPS）第 140-3 版》。

强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括您使用控制台、API 或 AWS 服务 使用 Amazon Location 或其他网站时 AWS SDKs。 AWS CLI在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供网址，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

数据隐私

使用 Amazon Location Service，您可以保留对组织数据的控制权。Amazon Location 会删除客户元数据和账户信息，从而匿名化发送给数据提供程序的所有查询。

Amazon Location 不使用数据提供程序进行跟踪和地理围栏。这意味着您的敏感数据仍保留在您的 AWS 帐户中。这有助于保护敏感的位置信息（例如设施、资产和人员位置）免受第三方的侵害，保护用户隐私，并降低应用程序的安全风险。

有关更多信息，请参阅 AWS 数据隐私常见问题解答。

Amazon Location 的数据保留

以下特征与 Amazon Location 如何收集和存储服务数据有关：

- Amazon Location Servic e 追 APIs 踪器 — 当您使用跟踪器跟踪实体的位置时，可以存储其坐标。设备位置会存储 30 天，然后才会被服务删除。
- Amazon Loc ation Service Geofences — 当您使用地理围栏 APIs 定义感兴趣区域时，该服务会存储您提供的几何图形。必须明确删除它们。

> **Note**
>
> 删除您的 AWS 账户会删除其中的所有资源。有关更多信息，请参阅 AWS 数据隐私常见问题解答。

Amazon Location Service 中的静态数据加密

Amazon Location Service 默认提供加密，以使用 AWS 自有的加密密钥保护敏感的静态客户数据。

- AWS 自有密钥 — Amazon Location 默认使用这些密钥来自动加密个人身份数据。您无法查看、管理或使用 AWS 自有密钥，也无法审核其使用情况。但是，无需采取任何措施或更改任何计划即可保护用于加密数据的密钥。有关更多信息，请参阅 AWS Key Management Service 开发人员指南中的 AWS 自有密钥。

默认情况下，静态数据加密有助于减少保护敏感数据时涉及的操作开销和复杂性。同时，它还支持构建符合严格加密合规性和监管要求的安全应用程序。

虽然您无法禁用此加密层或选择其他加密类型，但您可以在创建跟踪器和地理围栏收集资源时选择客户管理的密钥，从而在现有的 AWS 加密密钥上添加第二层加密：

- 客户托管密钥 — Amazon Location 支持使用由您创建、拥有和管理的对称客户托管密钥，在现有 AWS 自有加密的基础上添加第二层加密。由于您可以完全控制这层加密，因此可以执行以下任务：
  - 制定和维护关键策略
  - 建立和维护 IAM 策略和授权
  - 启用和禁用密钥策略
  - 轮换加密材料
  - 添加标签
  - 创建密钥别名
  - 计划删除密钥

有关更多信息，请参阅 AWS Key Management Service 开发人员指南中的客户托管密钥。

下表汇总了 Amazon Location 如何加密个人身份数据。

数据类型	AWS 自有密钥加密	客户托管密钥加密（可选）
`Position` 包含设备位置详细信息的造点几何体。	已启用	已启用
`PositionProperties` 一组与位置更新关联的键值对。	已启用	已启用
`GeofenceGeometry` 表示地理围栏区域的多边形地理围栏几何。	已启用	已启用
`DeviceId` 将设备位置更新上传到跟踪器资源时指定的设备标识符。	已启用	不支持
`GeofenceId` 在给定地理围栏集合中存储地理围栏几何或一批地理围栏时指定的标识符。	已启用	不支持

> **Note**
>
> Amazon Location 使用 AWS 自有密钥自动启用静态加密，从而免费保护个人身份数据。但是，使用客户管理的密钥需要 AWS KMS 付费。有关定价的更多信息，请参阅 AWS Key Management Service 定价。

有关的更多信息 AWS KMS，请参阅什么是 AWS Key Management Service？

Amazon Location Service 如何使用补助金 AWS KMS

Amazon Location 需要授权，才能使用客户托管密钥。

当您创建使用客户托管密钥加密的追踪器资源或地理围栏集合时，Amazon Location 会通过向发送CreateGrant请求来代表您创建授权。 AWS KMS中的授权 AWS KMS 用于让 Amazon Location 访问客户账户中的 KMS 密钥。

Amazon Location 需要授权，才能将客户托管的密钥用于以下内部操作：

- 向发送DescribeKey请求， AWS KMS 以验证在创建跟踪器或地理围栏集合时输入的对称客户托管 KMS 密钥 ID 是否有效。
- 向发送GenerateDataKeyWithoutPlaintext请求 AWS KMS 以生成由您的客户托管密钥加密的数据密钥。
- 将 Decrypt 请求发送 AWS KMS 到以解密加密的数据密钥，以便它们可用于加密您的数据。

您可以随时撤销授予访问权限，或删除服务对客户托管密钥的访问权限。如果您这样做，Amazon Location 将无法访问由客户托管的密钥加密的任何数据，这会影响依赖于该数据的操作。例如，如果您尝试从 Amazon Location 无法访问的加密跟踪器中获取设备位置，则该操作将返回 `AccessDeniedException` 错误。

创建客户托管密钥

您可以使用 AWS Management Console、或，创建对称的客户托管密钥。 AWS KMS APIs

创建对称的客户托管密钥

按照《AWS Key Management Service 开发人员指南》中创建对称的客户托管密钥的步骤进行操作。

密钥策略

密钥策略控制对客户自主管理型密钥的访问。每个客户托管式密钥必须只有一个密钥策略，其中包含确定谁可以使用密钥以及如何使用密钥的声明。创建客户托管式密钥时，可以指定密钥策略。有关更多信息，请参阅 AWS Key Management Service 开发人员指南中的管理对客户托管密钥的访问。

要将客户托管密钥与 Amazon Location 资源结合使用，密钥政策中必须允许以下 API 操作：

- kms:CreateGrant – 向客户托管密钥添加授权。授予对指定 KMS 密钥的控制访问权限，从而允许访问授予操作，这些操作是 Amazon Location 所需的。有关使用授权的更多信息，请参阅 AWS Key Management Service 开发人员指南。

  这将允许 Amazon Location 执行以下操作：

  - 调用 GenerateDataKeyWithoutPlainText 生成加密的数据密钥并将其存储，因为数据密钥不会立即用于加密。
  - 调用 Decrypt 使用存储的加密数据密钥访问加密数据。
  - 设置停用委托人，以允许服务 RetireGrant。
- kms:DescribeKey – 提供客户托管式密钥详细信息以允许 Amazon Location 验证密钥。

以下是您可以为 Amazon Location 添加的政策声明示例：

```
 "Statement" : [
   {
     "Sid" : "Allow access to principals authorized to use Amazon Location",
     "Effect" : "Allow",
     "Principal" : {
       "AWS" : "*"
     },
     "Action" : [
       "kms:DescribeKey",
       "kms:CreateGrant"
     ],
     "Resource" : "*",
     "Condition" : {
       "StringEquals" : {
         "kms:ViaService" : "geo.region.amazonaws.com",
         "kms:CallerAccount" : "111122223333"
       }
   },
   {
     "Sid": "Allow access for key administrators",
     "Effect": "Allow",
     "Principal": {
       "AWS": "arn:aws:iam::111122223333:root"
      },
     "Action" : [
       "kms:*"
      ],
```

```
      "Resource": "arn:aws:kms:region:111122223333:key/key_ID"
    },
    {
      "Sid" : "Allow read-only access to key metadata to the account",
      "Effect" : "Allow",
      "Principal" : {
        "AWS" : "arn:aws:iam::111122223333:root"
      },
      "Action" : [
        "kms:Describe*",
        "kms:Get*",
        "kms:List*",
        "kms:RevokeGrant"
      ],
      "Resource" : "*"
    }
  ]
```

有关在策略中指定权限的更多信息，请参阅 AWS Key Management Service 开发人员指南。

有关密钥访问故障排除更多的信息，请参阅 AWS Key Management Service 开发人员指南。

指定 Amazon Location 的客户托管密钥

您可以指定客户托管密钥作为以下资源的第二层加密：

- the section called “创建跟踪器”
- the section called “开始使用”

创建资源时，您可以通过输入 KMS ID 来指定数据密钥，Amazon Location 使用该 ID 来加密资源存储的可识别个人数据。

- KMS ID — AWS KMS 客户托管密钥的密钥标识符。输入密钥 ID、密钥 ARN、别名名称或别名 ARN。

Amazon Location Service 加密上下文

加密上下文是一组可选的键值对，包含有关数据的其他上下文信息。

AWS KMS 使用加密上下文作为其他经过身份验证的数据来支持经过身份验证的加密。当您在加密数据的请求中包含加密上下文时，会将加密上下文 AWS KMS 绑定到加密数据。要解密数据，您需要在请求中包含相同的加密上下文。

Amazon Location Service 加密上下文

Amazon Location 在所有 AWS KMS 加密操作中使用相同的加密上下文，其中密钥为`aws:geo:arn`，值为资源亚马逊资源名称 (ARN)。

Example

```
"encryptionContext": {
    "aws:geo:arn": "arn:aws:geo:us-west-2:111122223333:geofence-collection/SAMPLE-
GeofenceCollection"
}
```

使用加密上下文进行监控

当您使用对称的客户托管密钥来加密您的跟踪器或地理围栏集合时，您还可以使用审计记录和日志中的加密上下文来识别客户托管密钥的使用情况。加密上下文还会显示在AWS CloudTrail 或 Amazon Logs 生成的 CloudWatch 日志中。

使用加密上下文控制对客户托管式密钥的访问

您可以使用密钥策略和 IAM 策略中的加密上下文作为 `conditions` 来控制对您的对称客户托管密钥的访问。您还可以在授权中使用加密上下文限制。

Amazon Location 在授权中使用加密上下文限制，以控制对您账户或区域中的客户管理密钥的访问。授权约束要求授权允许的操作使用指定的加密上下文。

Example

以下是密钥政策声明示例，用于授予对特定加密上下文的客户托管密钥的访问权限。此策略声明中的条件要求授权具有指定加密上下文的加密上下文约束。

```
{
    "Sid": "Enable DescribeKey",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleReadOnlyRole"
     },
     "Action": "kms:DescribeKey",
     "Resource": "*"
```

```
},
{
     "Sid": "Enable CreateGrant",
     "Effect": "Allow",
     "Principal": {
         "AWS": "arn:aws:iam::111122223333:role/ExampleReadOnlyRole"
     },
     "Action": "kms:CreateGrant",
     "Resource": "*",
     "Condition": {
         "StringEquals": {
             "kms:EncryptionContext:aws:geo:arn": "arn:aws:geo:us-
west-2:111122223333:tracker/SAMPLE-Tracker"
          }
     }
}
```

监控您的 Amazon Location Service 的加密密钥

当您将 AWS KMS 客户托管密钥与 Amazon Location Service 资源一起使用时，您可以使用AWS CloudTrail或 Amazon CloudWatch Logs 来跟踪亚马逊位置发送到的请求 AWS KMS。

以下示例是CreateGrant、GenerateDataKeyWithoutPlainTextDecrypt、和DescribeKey监控 Amazon Location 调用的 KMS 操作 AWS CloudTrail 的事件，以访问由您的客户托管密钥加密的数据：

CreateGrant

当您使用 AWS KMS 客户托管密钥加密您的追踪器或地理围栏收集资源时，Amazon Location 会代表您发送访问您账户中的 KMS 密钥的CreateGrant请求。 AWS Amazon Location 创建的授权特定于与 AWS KMS 客户托管密钥关联的资源。此外，当您删除资源时，Amazon Location 会使用 RetireGrant 操作来删除授权。

以下示例事件记录了 CreateGrant 操作：

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
```

```
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
                "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-04-22T17:02:00Z"
            }
        },
        "invokedBy": "geo.amazonaws.com"
    },
    "eventTime": "2021-04-22T17:07:02Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "172.12.34.56",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "retiringPrincipal": "geo.region.amazonaws.com",
        "operations": [
            "GenerateDataKeyWithoutPlaintext",
            "Decrypt",
            "DescribeKey"
        ],
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "granteePrincipal": "geo.region.amazonaws.com"
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE"
    },
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
```

```
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKeyWithoutPlainText

当您为追踪器或地理围栏收集资源启用 AWS KMS 客户托管密钥时，Amazon Location 会创建一个唯一的表密钥。它向发送`GenerateDataKeyWithoutPlainText`请求 AWS KMS ，指定资源的 AWS KMS 客户托管密钥。

以下示例事件记录了 `GenerateDataKeyWithoutPlainText` 操作：

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "geo.amazonaws.com"
    },
    "eventTime": "2021-04-22T17:07:02Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyWithoutPlaintext",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "172.12.34.56",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "encryptionContext": {
            "aws:geo:arn": "arn:aws:geo:us-west-2:111122223333:geofence-collection/
SAMPLE-GeofenceCollection"
        },
        "keySpec": "AES_256",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
```

```
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "sharedEventID": "57f5dbee-16da-413e-979f-2c4c6663475e"
}
```

Decrypt

当您访问加密的跟踪器或地理围栏集合时，Amazon Location 会地理围栏 `Decrypt` 操作以使用存储的加密数据密钥来访问加密数据。

以下示例事件记录了 `Decrypt` 操作：

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "geo.amazonaws.com"
    },
    "eventTime": "2021-04-22T17:10:51Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "172.12.34.56",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "encryptionContext": {
            "aws:geo:arn": "arn:aws:geo:us-west-2:111122223333:geofence-collection/
SAMPLE-GeofenceCollection"
        },
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
    },
```

```
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "sharedEventID": "dc129381-1d94-49bd-b522-f56a3482d088"
}
```

DescribeKey

Amazon Location 使用 DescribeKey 操作来验证账户和地区中是否存在与您的跟踪器或地理围栏集合关联的 AWS KMS 客户托管密钥。

以下示例事件记录了 DescribeKey 操作：

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
                "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
```

```
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-04-22T17:02:00Z"
            }
        },
        "invokedBy": "geo.amazonaws.com"
    },
    "eventTime": "2021-04-22T17:07:02Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "172.12.34.56",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "keyId": "00dd0db0-0000-0000-ac00-b0c000SAMPLE"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
}
```


了解更多

以下资源提供有关静态数据加密的更多信息。

- 有关 AWS Key Management Service 基本概念的更多信息，请参阅 《AWS Key Management Service 开发人员指南》。
- 有关安全最佳实践的更多信息 AWS Key Management Service，请参阅《AWS Key Management Service 开发人员指南》。

Amazon Location Service 的传输中数据加密

Amazon Location 使用传输层安全性协议 (TLS) 1.2 加密协议自动加密所有网络间数据，从而保护往返服务的过程中的传输中数据。发送到 Amazon Location Servic APIs e 的直接 HTTPS 请求使用AWS 签名版本 4 算法进行签名，以建立安全连接。

Amazon Location Service 中的事件响应

AWS非常重视安全性。作为 AWS 云责任共担模式的一部分， AWS 管理满足大多数安全敏感组织要求的数据中心和网络架构。作为 AWS 客户，您有责任维护云端的安全。这意味着你可以从你有权访问的 AWS 工具和功能中控制你选择实施的安全性。

通过建立符合云端运行应用程序目标的安全基准，您可以检测出可以响应的偏差。由于安全事件响应可能是一个复杂的主题，因此我们鼓励您查看以下资源，以便更好地了解事件响应 (IR) 和您的选择对企业目标的影响：AWS 安全事件响应指南、AWS 安全最佳实践白皮书和AWS 云采用框架 (AWS CAF)。

Amazon Location Service 中的日志记录和监控

记录和监控是事件响应的重要组成部分。它允许您建立安全基线来检测可以调查和响应的偏差。通过对 Amazon Location Service 实施日志记录和监控，您可以保持项目和资源的可靠性、可用性、性能。

AWS 提供了多种工具，可以帮助您记录和收集事件响应数据：

AWS CloudTrail

> Amazon Location Service 与 AWS CloudTrail集成，后者是一项记录用户、角色或 AWS 服务所执行操作的服务。这包括来自 Amazon Location Service 控制台的操作以及对 Amazon Location API 操作的编程调用。这些操作记录称为事件。有关更多信息，请参阅使用记录和监控 Amazon Location Service AWS CloudTrail。

Amazon CloudWatch

> 您可以使用亚马逊收集和分析 CloudWatch 与您的亚马逊定位服务账户相关的指标。您可以启用 CloudWatch 警报，以便在指标满足特定条件并达到指定阈值时通知您。当您创建警报时，CloudWatch会向您定义的 Amazon 简单通知服务发送通知。有关更多信息，请参阅 “使用亚马逊监控亚马逊定位服务” CloudWatch。

AWS Health 仪表板

使用 AWS Health 控制面板，您可以验证 Amazon Location Service 服务的状态。您还可以监控和查看有关可能影响 AWS 环境的任何事件或问题的历史数据。有关更多信息，请参阅 AWS Health 用户指南。

Amazon Location Service 的合规性验证

要了解是否属于特定合规计划的范围，请参阅AWS 服务 “按合规计划划分的范围” ”，然后选择您感兴趣的合规计划。 AWS 服务 有关一般信息，请参阅AWS 合规计划AWS。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的 “下载报告” 中的 “ AWS Artifact。

您在使用 AWS 服务 时的合规责任取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。 AWS 提供了以下资源来帮助实现合规性：

- Security Compliance & Governance：这些解决方案实施指南讨论了架构考虑因素，并提供了部署安全性和合规性功能的步骤。
- 符合 HIPAA 要求的服务参考：列出符合 HIPAA 要求的服务。并非所有 AWS 服务 人都符合 HIPAA 资格。
- AWS 合AWS 规资源 — 此工作簿和指南集合可能适用于您的行业和所在地区。
- AWS 客户合规指南 — 从合规角度了解责任共担模式。这些指南总结了保护的最佳实践， AWS 服务并将指南映射到跨多个框架（包括美国国家标准与技术研究院 (NIST)、支付卡行业安全标准委员会 (PCI) 和国际标准化组织 (ISO)）的安全控制。
- 使用AWS Config 开发人员指南中的规则评估资源 — 该 AWS Config 服务评估您的资源配置在多大程度上符合内部实践、行业准则和法规。
- AWS Security Hub— 这 AWS 服务 可以全面了解您的安全状态 AWS。Security Hub 通过安全控制措施评估您的 AWS 资源并检查其是否符合安全行业标准和最佳实践。有关受支持服务及控制措施的列表，请参阅 Security Hub 控制措施参考。
- Amazon GuardDuty — 它通过监控您的 AWS 账户环境中是否存在可疑和恶意活动，来 AWS 服务检测您的工作负载、容器和数据面临的潜在威胁。 GuardDuty 通过满足某些合规性框架规定的入侵检测要求，可以帮助您满足各种合规性要求，例如 PCI DSS。
- AWS Audit Manager— 这 AWS 服务 可以帮助您持续审计 AWS 使用情况，从而简化风险管理以及对法规和行业标准的合规性。

Amazon Location Service 中的韧性

AWS 全球基础设施是围绕 AWS 区域 可用区构建的。 AWS 区域 提供多个物理隔离和隔离的可用区，这些可用区通过低延迟、高吞吐量和高度冗余的网络连接。利用可用区，您可以设计和操作在可用区之间无中断地自动实现失效转移的应用程序和数据库。与传统的单个或多个数据中心基础结构相比，可用区具有更高的可用性、容错性和可扩展性。

有关 AWS 区域 和可用区的更多信息，请参阅AWS 全球基础设施。

除了 AWS 全球基础设施外，Amazon Location 还提供多项功能来帮助支持您的数据弹性和备份需求。

Amazon Location Service 中的基础设施安全性

作为一项托管服务，Amazon Location Service 受 AWS 全球网络安全的保护。有关 AWS 安全服务以及如何 AWS 保护基础设施的信息，请参阅AWS 云安全。要使用基础设施安全的最佳实践来设计您的 AWS 环境，请参阅 S AWS ecurity Pillar Well-Architected Fram ework 中的基础设施保护。

您可以使用 AWS 已发布的 API 调用通过网络访问 Amazon 位置。客户端必须支持以下内容：

- 传输层安全性协议（TLS）。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 具有完全向前保密（PFS）的密码套件，例如 DHE（临时 Diffie-Hellman）或 ECDHE（临时椭圆曲线 Diffie-Hellman）。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

此外，必须使用访问密钥 ID 和与 IAM 主体关联的秘密访问密钥来对请求进行签名。或者，您可以使用 AWS Security Token Service（AWS STS）生成临时安全凭证来对请求进行签名。

AWS PrivateLink 适用于 Amazon 地点

使用 F AWS PrivateLink or Amazon Location，您可以在虚拟私有云（亚马逊 VPC）中配置接口 Amazon VPC 终端节点（接口终端节点）。这些终端节点可以通过 VPN 直接从本地应用程序访问 AWS Direct Connect，也可以通过 Amazon VPC AWS 区域 对等互连进行其他访问。使用 AWS PrivateLink 和接口终端节点，您可以简化从应用程序到 Amazon Location 的私有网络连接。

您的 VPC 中的应用程序不需要公有 IP 地址即可与 Amazon Location 接口 VPC 终端节点进行通信以进行亚马逊定位操作。接口终端节点由一个或多个弹性网络接口 (ENIs) 表示，这些接口从您的 Amazon VPC 中的子网中分配私有 IP 地址。通过接口终端节点向亚马逊位置发出的请求仍保留在亚马逊网络上。您还可以通过 AWS Direct Connect 或 AWS Virtual Private Network (AWS VPN) 从本地应用程序

访问您的 Amazon VPC 中的接口终端节点。有关如何将 Amazon VPC 与本地网络连接的更多信息，请参阅 AWS Direct Connect User Guide 和《AWS Site-to-Site VPN User Guide》。

有关接口端点的一般信息，请参阅《AWS PrivateLink 指南》中的接口 Amazon VPC 端点（AWS PrivateLink）。

主题

- Amazon Location Service 的亚马逊 VPC 终端节点类型
- 使用 Amazon Location AWS PrivateLink Service 时的注意事项
- 为 Amazon Location Service 创建接口终端节点
- 从亚马逊定位接口终端节点访问亚马逊定位 API 操作
- 更新本地 DNS 配置
- 为亚马逊位置创建亚马逊 VPC 终端节点策略

Amazon Location Service 的亚马逊 VPC 终端节点类型

您可以使用一种类型的亚马逊 VPC 终端节点来访问 Amazon Location Service：接口终端节点（使用 AWS PrivateLink）。接口终端节点使用私有 IP 地址， AWS 区域 通过亚马逊 VPC 对等互连将请求从您的亚马逊 VPC 内部、本地或另一个 Amazon VPC 中的亚马逊 VPC 路由到亚马逊位置。有关更多信息，请参阅 What is Amazon VPC peering? 和 Transit Gateway 与 Amazon VPC 对等连接。

接口端点与网关端点兼容。如果您在 Amazon VPC 中有现有网关端点，则可以在同一 Amazon VPC 中使用这两种类型的端点。

Amazon Location 的接口终端节点具有以下属性：

- 您的网络流量仍留在 AWS 网络上
- 使用您的亚马逊 VPC 中的私有 IP 地址访问亚马逊定位服务
- 允许从本地访问
- 允许通过使用 Amazon VPC 对等互连从另一个 AWS 区域 Amazon VPC 终端节点进行访问或 AWS Transit Gateway
- 接口终端节点已计费

使用 Amazon Location AWS PrivateLink Service 时的注意事项

亚马逊 VPC 注意事项 AWS PrivateLink 适用于亚马逊 Location Service。有关更多信息，请参阅《AWS PrivateLink 指南》中的接口端点注意事项和 AWS PrivateLink 限额。此外，以下限制将适用：

AWS PrivateLink 适用于 Amazon Location Service 不支持以下内容：

- 传输层安全性协议（TLS）1.1
- 私有域名系统和混合域名系统（DNS）服务

亚马逊 VPC 终端节点：

- 不支持 Amazon Location Service 地图 API 操作，包括：`GetGlyphs`、`GetSprites`、和 `GetStyleDescriptor`
- 不支持跨区域请求。确保在计划向 Amazon Location Service 发出 API 调用的同一区域创建终端节点。
- 仅支持亚马逊通过亚马逊 Route 53 提供的 DNS。如果您想使用自己的 DNS，请使用有条件的 DNS 转发。有关更多信息，请参阅《Amazon VPC 用户指南》中的 DHCP 选项集。
- 必须允许端口 443 通过连接到 VPC 终端节点的安全组从 VPC 的私有子网传入连接

对于您启用的每个 AWS PrivateLink 终端节点，您每秒最多可以提交 50,000 个请求。

> **Note**
>
> AWS PrivateLink 终端节点的网络连接超时不在 Amazon Location 错误响应的范围内，需要由连接到 AWS PrivateLink 终端节点的应用程序进行适当处理。

为 Amazon Location Service 创建接口终端节点

您可以使用亚马逊 VPC 控制台或 AWS Command Line Interface (AWS CLI) 为 Amazon Location Service 创建接口终端节点。有关更多信息，请参阅《AWS PrivateLink 指南》中的创建接口端点。

有六个不同的 VPC 终端节点，Amazon Location Service 提供的每项功能各一个。

类别	终端节点
映射	com.amazonaws. *region*.geo.maps
位数	com.amazonaws. *region*.geo.places
路线	com.amazonaws. *region*.geo.routes
地理围栏	com.amazonaws. *region*.geo.geofencing
跟踪器	com.amazonaws. *region*.geo.tracking
元数据	com.amazonaws. *region*.geo.metadata

例如：

```
com.amazonaws.us-east-2.geo.maps
```

在创建端点后，您可以选择启用私有 DNS 主机名。要启用，请在创建 VPC 终端节点时在 Amazon VPC 控制台中选择启用私有 DNS 名称。

如果您为接口终端节点启用私有 DNS，则可以使用其默认区域 DNS 名称向 Amazon Location Service 服务发出 API 请求。以下示例显示了默认的区域 DNS 名称格式。

- maps.geo.*region*.amazonaws.com
- places.geo.*region*.amazonaws.com
- routes.geo.*region*.amazonaws.com
- tracking.geo.*region*.amazonaws.com
- geofencing.geo.*region*.amazonaws.com
- metadata.geo.*region*.amazonaws.com

之前的 DNS 名称是针对 IPv4 域名的。以下 IPV6 DNS 名称也可以用于接口终端节点。

- maps.geo.*region*.api.aws
- places.geo.*region*.api.aws
- routes.geo.*region*.api.aws

- tracking.geo.*region*.api.aws
- geofencing.geo.*region*.api.aws
- metadata.geo.*region*.api.aws

从亚马逊定位接口终端节点访问亚马逊定位 API 操作

您可以使用AWS CLI或AWS SDKs通过亚马逊定位接口终端节点访问亚马逊定位 API 操作。

示例：创建 VPC 端点

```
aws ec2 create-vpc-endpoint \
--region us-east-1 \
--service-name location-service-name \
--vpc-id client-vpc-id \
--subnet-ids client-subnet-id \
--vpc-endpoint-type Interface \
--security-group-ids client-sg-id
```

示例：修改 VPC 端点

```
aws ec2 modify-vpc-endpoint \
--region us-east-1 \
--vpc-endpoint-id client-vpc-endpoint-id \
--policy-document policy-document \ #example optional parameter
--add-security-group-ids security-group-ids \ #example optional parameter
# any additional parameters needed, see PrivateLink documentation for more details
```

更新本地 DNS 配置

使用终端节点特定的 DNS 名称访问 Amazon Location 的接口终端节点时，您无需更新本地 DNS 解析器。您可以使用公共 Amazon Location DNS 域中的接口终端节点的私有 IP 地址解析终端节点特定的 DNS 名称。

使用接口终端节点访问亚马逊位置，无需网关终端节点或亚马逊 VPC 中的互联网网关

您的 Amazon VPC 中的接口终端节点可以通过亚马逊网络将亚马逊 VPC 内部应用程序和本地应用程序路由到亚马逊地点。

为亚马逊位置创建亚马逊 VPC 终端节点策略

您可以将终端节点策略附加到控制亚马逊位置访问权限的 Amazon VPC 终端节点。该策略指定以下信息：

- 可以执行操作的 AWS Identity and Access Management (IAM) 委托人
- 可执行的操作
- 可对其执行操作的资源

示例：访问 Amazon Location Service 地点的 VPCe 政策示例 APIs：

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "Allow-access-to-location-service-places-opeartions",
   "Effect": "Allow",
   "Action": [
    "geo-places:*",
    "geo:*"
   ],
   "Resource": [
    "arn:aws:geo-places:us-east-1::provider/default",
    "arn:aws:geo:us-east-1:*:place-index/*"
   ]
  }
 ]
}
```

Amazon Location 中的配置和漏洞分析

配置和 IT 控制由您（我们的客户）共同 AWS 负责。有关更多信息，请参阅责任 AWS 共担模型。

防止跨服务混淆代理

混淆代理问题是一个安全性问题，即不具有某操作执行权限的实体可能会迫使具有更高权限的实体执行该操作。在中 AWS，跨服务模仿可能会导致混乱的副手问题。一个服务（呼叫服务）调用另一项服务（所谓的服务）时，可能会发生跨服务模拟。可以操纵调用服务，使用其权限以在其他情况下该服务不

应有访问权限的方式对另一个客户的资源进行操作。为防止这种情况，AWS 提供可帮助您保护所有服务的数据的工具，而这些服务中的服务主体有权限访问账户中的资源。

Amazon Location Service 不会代表您充当其他 AWS 服务的呼叫服务，因此在这种情况下，您无需添加这些保护措施。要了解更多关于混淆代理问题，请参阅 AWS Identity and Access Management 用户指南中的混淆代理问题。

Amazon Location Service 的最佳实践

本主题提供了可帮助您使用 Amazon Location Service 的最佳实践。虽然这些最佳实践可以帮助您充分利用 Amazon Location Service，但它们并不代表完整的解决方案。您应该只遵循适用于您的环境的建议。

主题

- 安全性

安全性

为了帮助管理甚至避免安全风险，请考虑以下最佳实践：

- 使用联合身份和 IAM 角色来管理、控制或限制对您的Amazon Location资源的访问。有关更多信息，请参阅《IAM 用户指南》中的 IAM 最佳实操。
- 遵循最低权限原则，仅授予对您的 Amazon Location Service 资源所需的最低访问权限。
- 对于网络应用程序中使用的 Amazon Location Service 资源，请使用 `aws:referer` IAM 条件限制访问权限，限制除允许名单中包含的网站以外的网站使用。
- 使用监控和日志工具来跟踪器资源的访问和使用情况。有关更多信息，请参阅 AWS CloudTrail 用户指南中的the section called “日志记录和监控”和记录跟踪的数据事件。
- 使用安全连接，例如以 `https://` 开头的连接，以增加安全性并在服务器与浏览器之间传输数据时保护用户免受攻击。

Amazon Location Service 的检测性安全最佳实践

Amazon Location Service 的以下最佳实践可以帮助检测安全事故：

实施 AWS 监控工具

监控对于事件响应至关重要，可以维护 Amazon Location Service 资源和解决方案的可靠性和安全性。您可以从多种可用工具和服务中实施监控工具 AWS ，以监控您的资源和其他 AWS 服务。

例如，亚马逊 CloudWatch 允许您监控亚马逊定位服务（Amazon Location Service）的指标，并允许您设置警报，以便在指标满足您设定的特定条件并达到您定义的阈值时通知您。创建警报时，您可以设置 CloudWatch 为使用 Amazon 简单通知服务发送提醒通知。有关更多信息，请参阅 the section called “日志记录和监控”。

启用 AWS 日志工具

日志记录记录用户、角色或 AWS 服务在 Amazon Location Service 中采取的操作。您可以实现日志工具，例如 AWS CloudTrail 收集操作数据以检测异常 API 活动。

创建跟踪时，可以配置 CloudTrail 为记录事件。事件是对资源或资源内部执行的资源操作的记录，如向 Amazon Location 发出的请求、发出请求的 IP 地址、请求的发出请求的 IP 地址、时间、时间，以及其他数据。有关更多信息，请参阅 AWS CloudTrail 用户指南中的记录跟踪的数据事件。

Amazon Location Service 的预防性安全最佳实践

Amazon Location Service 的以下最佳实践可以帮助预防安全事故：

使用安全连接

请务必使用加密连接，例如以 `https://` 开头的连接，以确保敏感信息在传输过程中的安全。

实施最低权限访问资源

当您为 Amazon Location 资源创建自定义策略时，只授予执行任务所需的权限。建议最开始只授予最低权限，然后根据需要授予其他权限，则样会更加安全。实施最低权限访问对于减小风险以及可能由错误或恶意攻击造成的影响至关重要。有关更多信息，请参阅 the section called “使用 IAM”。

使用全球唯一的设备作为设备 IDs IDs

对设备使用以下约定 IDs。

- 设备 IDs 必须是唯一的。
- 设备 IDs 不应是秘密的，因为它们可以用作其他系统的外键。
- 设备 IDs 不应包含个人身份信息 (PII)，例如电话设备 IDs 或电子邮件地址。
- 设备 IDs 不应是可预测的。建议使用不透明的标识符 UUIDs ，例如。

不要在设备位置属性中包含 PII

发送设备更新（例如，使用 DevicePositionUpdate）时，请勿在中包含个人身份信息 (PII)，例如电话号码或电子邮件地址。`PositionProperties`

文档历史记录

下表介绍 Amazon Location Service 的文档。如需有关更新的通知，您可以订阅 RSS 源。

变更	说明	日期
Amazon Location Service 发布增强版正式上市	Amazon Location Service 现在提供增强的地点、路线和地图功能，使开发人员能够更轻松地在应用程序中添加高级定位功能。这些改进引入了新的功能和全新的简化开发者体验，以支持医疗保健、运输和物流以及零售等行业的基于位置的用例。如需了解更多信息，请参阅发行版。	2024 年 10 月 31 日

AWS 词汇表

有关最新 AWS 术语，请参阅《AWS 词汇表 参考资料》中的AWS 词汇表。

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。