



Entwicklerhandbuch

Amazon Location Service



Amazon Location Service: Entwicklerhandbuch

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Willkommen	1
Was ist Amazon Location Service?	1
Schlüsselfeatures	2
Zugehörige Services	3
Schnellstart	5
Erstellen einer Webanwendung	5
Erstellen von -Ressourcen	6
Einrichtung der Authentifizierung	7
Erstellen von HTML	9
Hinzufügen der Karte	12
Hinzufügen der Suche	16
Endgültige Anwendung	21
Was ist als Nächstes?	26
Erstellen einer Android-App	26
Erstellen von -Ressourcen	27
Einrichtung der Authentifizierung	29
Erstellen der App	31
Hinzufügen der Karte	31
Hinzufügen einer Suche	39
Endgültige Anwendung	46
Was ist als Nächstes	46
Erstellen einer iOS-App	47
Erstellen von -Ressourcen	47
Einrichtung der Authentifizierung	49
Erstellen der App	52
Hinzufügen der Karte	52
Hinzufügen der Suche	57
Was ist als Nächstes?	46
Konzepte von Amazon Location	65
Übersicht	66
Zuordnungen	67
Kartenstile	68
Politische Ansichten	68
Benutzerspezifische Layers	69

Map-Rendering	69
Zuordnungs-Terminologie	71
Ortssuche	72
Geokodierungskonzepte	74
Suchergebnisse	74
Mehrere Ergebnisse und Relevanz	75
Ergebnisse adressieren	75
Speichern von Geocode-Ergebnissen	77
Ortsterminologie	78
Routen	79
Routenrechner-Ressourcen	79
Berechnen einer Route	80
Planen von Routen	82
Routing-Terminologie	83
Geofences und Tracker	84
Geofences	85
Tracker	87
Geofence-Terminologie	91
Tracker-Terminologie	92
Häufige Anwendungsfälle	94
Anwendungen für Benutzerinteraktion und GeoMarketing	95
Anwendungen zur Bestandsverfolgung	97
Bereitstellungsanwendungen	98
Datenanbieter	100
Abdeckung und Features von Datenanbietern	100
Kartenstile	102
Weitere Details	102
Esri	102
GrabMaps	111
HERE-Technologien	116
Daten öffnen	123
Funktionen nach Datenanbieter	133
Nutzungsbedingungen und Datenzuordnung	138
Regionen und Endpunkte	138
Regionen	139
Endpunkte	140

API-Operation Endpunkte	141
Service Quotas	143
Verwalten Ihrer Amazon Location Service Quotas	160
Entwickeln mit Amazon Location	162
Szenarien und Anwendungsfälle	162
SDKs und Tools	164
SDKs nach Sprache	164
MapLibre	168
Amazon Location SDK	169
Amazon Location APIs	176
Verwenden von Amazon Location mit einem AWS-SDK	177
Aktualisierungen der Fehlermeldung	177
Codebeispiele	209
Amazon-Standort-Demo-Website	211
Tutorial: Schnellstart	211
Tutorial: Datenbankanreicherung	212
Beispiel: Erkunden der App	213
Beispiel: Entwerfen einer Karte	214
Beispiel: Grafikmarkierungen	214
Beispiel: Zeichnen von gruppierten Punkten	215
Beispiel: Zeichnen eines Polygons	215
Beispiel: Ändern der Kartensprache	216
Blog: Geschätzte Zustellungszeit-Benachrichtigungen	217
So verwenden Sie Amazon Location	218
Kontovoraussetzungen	219
So melden Sie sich für ein AWS-Konto an	220
Erstellen eines Administratorbenutzers	220
Zugriff auf Amazon Location Service gewähren	221
Verwenden von Karten	223
Voraussetzungen	224
Anzeigen von Karten	227
Zeichnen auf einer Karte	283
Einstellungserweiterungen für eine Zuordnung	283
Verwalten von Kartenressourcen	285
Ortssuche	288
Voraussetzungen	289

Geokodierung	293
Geokodierung umkehren	301
AutoVervollständigen	305
Verwenden von Orts-IDs	311
Kategorien und Filterung	313
Tutorial: Datenbankanreicherung	318
Verwalten von Ortsindexressourcen	333
Berechnen von Routen	337
Voraussetzungen	338
Berechnen der Route	341
Routenplanung	345
Positionen, die sich nicht auf einer Straßen befinden	352
Zeit für die Warteschlange	354
Reisemodus	355
Verwalten von Routenressourcen	357
Geofencing und Nachverfolgung	361
Schritt 1: Hinzufügen von Geofences	362
Schritt 2: Starten der Nachverfolgung	369
Schritt 3: Verknüpfen eines Trackers mit einer Geofence-Sammlung	384
Schritt 4: Gerätepositionen anhand von Geofences bewerten	386
Reagieren auf Ereignisse mit EventBridge	388
Nachverfolgung mit AWS IoT und MQTT	394
Verwalten von Geofence-Ressourcen	402
Verwalten von Tracker-Ressourcen	410
Markieren Ihrer -Ressourcen	415
Einschränkungen	416
Gewährt die Berechtigung zum Markieren	417
Hinzufügen eines Tags zu einer Ressource	417
Nachverfolgen von Kosten nach Tag	418
Kontrollieren des Zugriffs auf -Ressourcen mithilfe von Tags	419
Weitere Informationen	419
Gewähren des Zugriffs auf Amazon Location	420
Verwenden von API-Schlüsseln	421
Verwenden von Amazon Cognito	428
Überwachen von Amazon Location Service	439
Überwachung mit CloudWatch	439

Verwenden von CloudTrail mit Amazon Location	445
Verwenden von AWS CloudFormation zum Erstellen von Ressourcen	449
Amazon Location und AWS CloudFormation Vorlagen	450
Weitere Informationen zu AWS CloudFormation	450
Sicherheit	451
Datenschutz	452
Datenschutz	453
Datenaufbewahrung	453
Verschlüsselung von Daten im Ruhezustand	453
Datenverschlüsselung während der Übertragung	467
Identitäts- und Zugriffsverwaltung	467
Zielgruppe	468
Authentifizierung mit Identitäten	468
Verwalten des Zugriffs mit Richtlinien	472
Funktionsweise von Amazon Location Service mit IAM	475
Funktionsweise von Amazon Location Service mit nicht authentifizierten Benutzern	484
Beispiele für identitätsbasierte Richtlinien	484
Fehlerbehebung	497
Vorfallreaktion	499
Protokollieren und Überwachen	500
Compliance-Validierung	501
Ausfallsicherheit	502
Sicherheit der Infrastruktur	502
Konfigurations- und Schwachstellenanalyse	503
Confused-Deputy-Prävention	503
Bewährte Methoden für die Gewährleistung der Sicherheit	503
Bewährte Methoden für Detective	504
Bewährte Methoden für Präventive	505
Bewährte Methoden	505
Sicherheit	506
Ressourcenmanagement	507
Fakturierung und Kostenmanagement	507
Kontingente und Nutzung	507
Dokumentverlauf	509
AWS-Glossar	520
.....	dxxi

Willkommen beim Amazon Location Service

Willkommen beim Entwicklerhandbuch für Amazon Location Service .

Die folgenden Themen können Ihnen bei den ersten Schritten in der -Dokumentation helfen, je nachdem, was Sie versuchen zu tun.

Einen Überblick über Amazon Location erhalten

- Erfahren Sie mehr über die [Konzepte in Amazon Location](#) .
- Verschaffen Sie sich einen tieferen Einblick in die Funktionalität im [So verwenden Sie Amazon Location Service](#) Kapitel.
- Siehe Demo-Apps auf der [Demo-Website von Amazon Location](#) .
- Wenn Sie bereits über ein verfügbares AWS-Konto, können Sie die [Amazon Location Service-Konsole](#) verwenden, um die Funktionalität aus erster Hand zu erkunden.

Amazon Location als Entwickler verwenden

- Erstellen Sie Ihre erste App mit der [Schnellstart](#).
- Erfahren Sie im [So verwenden Sie Amazon Location Service](#) Kapitel, wie die verschiedenen Funktionen von Amazon Location Service funktionieren.
- Weitere Informationen finden Sie in den SDKs und Tools, die Ihnen im [Entwickeln mit Amazon Location](#) Kapitel zur Verfügung stehen.
- Sehen Sie sich [Codebeispiele und Tutorials](#) an, die Sie in Ihren eigenen Apps verwenden können. Sie können auch die Seite mit den [Demo-Websitebeispielen](#) von Amazon Location besuchen, um Beispiele zu finden, die nach Feature, Sprache oder Plattform gefiltert werden können.
- Informationen zu Amazon Location APIs finden Sie im [API-Referenzhandbuch für](#) .

Was ist Amazon Location Service?

Mit Amazon Location Service können Sie Standortdaten und Funktionen zu Anwendungen hinzufügen, die Funktionen wie Karten, Interessenpunkte, Geokodierung, Routing, Geofences und Nachverfolgung umfassen. Amazon Location bietet standortbasierte Services (LBS) unter Verwendung hochwertiger Daten von globalen, vertrauenswürdigen Anbietern Esri, Grab und HERE. Mit kostengünstigen Daten, Nachverfolgungs- und Geofencing-Funktionen und integrierten Metriken für die Zustandsüberwachung können Sie anspruchsvolle standortfähige Anwendungen erstellen.

Mit Amazon Location behalten Sie die Kontrolle über die Daten Ihrer Organisation. Amazon Location anonymisiert alle an Datenanbieter gesendeten Abfragen, indem Kundenmetadaten und Kontoinformationen entfernt werden. Darüber hinaus verlassen vertrauliche Standortinformationen wie Anlagen-, Komponenten- und Personalstandorte Ihr AWS Konto überhaupt nicht. Auf diese Weise können Sie vertrauliche Informationen von Drittanbietern schützen, den Datenschutz von Benutzern schützen und die Sicherheitsrisiken Ihrer Anwendung reduzieren. Mit Amazon Location haben Amazon und Dritte keine Rechte, Ihre Daten zu verkaufen oder für Werbung zu verwenden.

Amazon Location ist vollständig in -Services wie AWS CloudTrail, Amazon EventBridge, CloudWatchAmazon und AWS Identity and Access Management (IAM) integriert. Amazon Location vereinfacht Ihren Entwicklungsworkflow mit Datenintegration und verfolgt Apps mit integrierten Überwachungs-, Sicherheits- und Compliance-Funktionen schnell in die Produktion.

Highlights, Produktdetails und Preise finden Sie auf der Serviceseite für [Amazon Location Service](#) .

Wichtige Funktionen in Amazon Location

Amazon Location bietet die folgenden Funktionen:

Zuordnungen

Mit Amazon Location Service Maps können Sie Standortinformationen visualisieren und sind die Grundlage für viele standortbasierte Service-Funktionen. Amazon Location Service bietet Kartenkacheln verschiedener Stile, die von globalen Standortdatenanbietern Esri, Grab und HERE stammen, sowie Open Data Maps.

Orte

Mit Amazon Location Service Places können Sie Suchfunktionen in Ihre Anwendung integrieren, Adressen in geografische Koordinaten mit Breiten- und Längengrad (Geokodierung) konvertieren und eine Koordinate in eine Straßenadresse (Geokodierung umkehren) konvertieren. Amazon Location Service bezieht qualitativ hochwertige Geodaten von Esri, Grab und HERE, um Places-Funktionen zu unterstützen.

Routing

Mit Amazon Location Service Routes können Sie Routen finden und die Reisezeit auf der Grundlage von up-to-date Straßen- und Live-Datenverkehrsinformationen schätzen. Erstellen Sie

Funktionen, mit denen Ihre Anwendung die Reisezeit, Entfernung und Richtungen zwischen zwei beliebigen Standorten anfordern kann. Berechnen Sie die Zeit und Entfernung für eine Matrix von Routen, die bei der Routenplanung verwendet werden sollen.

Geofencing

Mit Amazon Location Service Geofences können Sie Ihrer Anwendung die Möglichkeit geben, zu erkennen und zu handeln, wenn ein Gerät eine definierte geografische Grenze betritt oder verlässt, die als Geofence bezeichnet wird. Senden Sie automatisch ein Ein- oder Ausgangsereignis an Amazon EventBridge, wenn eine Geofence-Verstöße erkannt werden. Auf diese Weise können Sie nachgelagerte Aktionen initiieren, z. B. das Senden einer Benachrichtigung an ein Ziel.

Tracker

Mit Amazon Location Service Trackers können Sie den aktuellen und historischen Standort der Geräte abrufen, auf denen Ihre Anwendung mit aktivierter Nachverfolgung ausgeführt wird. Sie können Tracker auch mit Amazon Location Service Geofences verknüpfen, um Standortaktualisierungen von Ihren Geräten automatisch anhand Ihrer Geofences auszuwerten. Tracker können Ihnen helfen, Kosten zu senken, indem sie Positionsaktualisierungen filtern, die sich vor dem Speichern oder Auswerten anhand von Geofences nicht bewegt haben.

Wenn Sie Tracker verwenden, verlassen vertrauliche Standortinformationen auf Ihren verfolgten Geräten Ihr AWS Konto nicht. Dies trägt dazu bei, vertrauliche Informationen vor Dritten zu schützen, den Datenschutz von Benutzern zu schützen und Sicherheitsrisiken zu reduzieren.

Services, die Sie mit Amazon Location verwenden können

Verwenden Sie die folgenden Services zusammen mit Amazon Location Service.

Integrierte Überwachung und Verwaltung

Amazon Location Service ist für eine EventBridge effiziente Überwachung und Datenverwaltung in Amazon CloudWatchAWS CloudTrail, und integriert:

- Amazon CloudWatch – Zeigen Sie Metriken zur Servicenutzung und zum Servicestatus an, einschließlich Anfragen, Latenz, Fehlern und Protokollen. Weitere Informationen finden Sie unter [the section called “Überwachung mit CloudWatch”](#).

- AWS CloudTrail – Protokollieren und überwachen Sie Ihre API-Aufrufe, die Aktionen eines Benutzers, einer Rolle oder eines AWS-Services umfassen. Weitere Informationen finden Sie unter [the section called “Verwenden von CloudTrail mit Amazon Location”](#).
- Amazon EventBridge – Aktivieren Sie eine ereignisgesteuerte Anwendungsarchitektur, damit Sie -AWS LambdaFunktionen verwenden können, um andere Teile Ihrer Anwendung und Workflows zu aktivieren. Weitere Informationen finden Sie unter [the section called “Reagieren auf Ereignisse mit EventBridge”](#).

Entwickler-Tools

Amazon Location Service bietet eine Vielzahl von Tools, mit denen Entwickler standortfähige Anwendungen erstellen können. Dazu gehören die Standard-AWSSDKs , mobile und Web-SDKs sowie Beispielcode, um sie mit Open-Source-Bibliotheken wie zu kombinieren MapLibre. Verwenden Sie die [Amazon Location Service-Konsole](#), um mehr über -Ressourcen zu erfahren und mit einem visuellen und interaktiven Lerntool zu beginnen.

Schnellstart mit Amazon Location Service

Der effizienteste Weg, um mit Amazon Location Service zu beginnen, ist die Verwendung der [Amazon Location-Konsole](#). Sie können Ihre -Ressourcen erstellen und verwalten und die Amazon Location-Funktionalität auf [der Seite Erkunden](#) ausprobieren.

Note

Um die Amazon Location Service-Konsole zu verwenden oder den Rest dieses Tutorials zu befolgen, erfordert , dass Sie zuerst die abschließen [Voraussetzungen für die Verwendung von Amazon Location Service](#), einschließlich der Erstellung eines AWS Kontos und der Erlaubung des Zugriffs auf Amazon Location.

Verwenden Sie das folgende APIs, um eine einfache Anwendung zu erstellen, die eine interaktive Karte anzeigt und Suchfunktionen verwendet, um mehr über die Amazon Location APIs zu erfahren. Es gibt drei Versionen des Tutorials: Eine zeigt Ihnen, wie Sie eine einfache Webseite mit erstellen JavaScript, die zweite zeigt dasselbe für eine Android-Anwendung mit Kotlin und die dritte zeigt dasselbe für eine iOS-Anwendung mit Swift.

Themen

- [Erstellen einer Webanwendung](#)
- [Erstellen einer Android-App](#)
- [Erstellen einer iOS-App](#)

Erstellen einer Webanwendung

In diesem Abschnitt erstellen Sie eine statische Webseite mit einer Karte und der Möglichkeit, an einem Standort zu suchen. Zunächst erstellen Sie Ihre Amazon Location-Ressourcen und einen API-Schlüssel für Ihre Anwendung.

Themen

- [Erstellen von Amazon Location-Ressourcen für Ihre App](#)
- [Einrichten der Authentifizierung für Ihre Anwendung](#)
- [Erstellen des HTML-Codes für Ihre Anwendung](#)

- [Hinzufügen einer interaktiven Zuordnung zu Ihrer Anwendung](#)
- [Hinzufügen einer Suche zu Ihrer Anwendung](#)
- [Anzeigen der endgültigen Anwendung](#)
- [Was ist als Nächstes?](#)

Erstellen von Amazon Location-Ressourcen für Ihre App

Wenn Sie sie noch nicht haben, müssen Sie die Amazon Location-Ressourcen erstellen, die Ihre Anwendung verwenden wird. Hier erstellen Sie eine Kartenressource, um Karten in Ihrer Anwendung anzuzeigen, und einen Ortsindex, um nach Speicherorten auf der Karte zu suchen.

So fügen Sie Ihrer Anwendung Standortressourcen hinzu

1. Wählen Sie den Kartenstil aus, den Sie verwenden möchten.
 - a. Wählen Sie in der Amazon-Standortkonsole auf der Seite [Karten](#) die Option Karte erstellen aus, um eine Vorschau der Kartenstile anzuzeigen.
 - b. Fügen Sie einen Name nund eine Beschreibung für die neue Kartenressource hinzu. Notieren Sie sich den Namen, den Sie für die Kartenressource verwenden. Sie benötigen sie später im Tutorial, um Ihre Skriptdatei zu erstellen.
 - c. Wählen Sie eine Zuordnung aus.

Note

Wenn Sie einen Kartenstil auswählen, wird auch ausgewählt, welchen Kartendatenanbieter Sie verwenden werden. Wenn Ihre Anwendung Komponenten verfolgt oder weiterleitet, die Sie in Ihrem Unternehmen verwenden, z. B. Liefermodelle oder Mitarbeiter, können Sie HERE nur als Geolokalisierungsanbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS-Servicebedingungen](#).

- d. Stimmen Sie den Allgemeinen Geschäftsbedingungen für Amazon Location zu und wählen Sie dann Karte erstellen aus. Sie können mit der ausgewählten Karte interagieren: Vergrößern, Verkleinern oder Vergrößern in eine beliebige Richtung.
- e. Notieren Sie sich den Amazon-Ressourcennamen (ARN), der für Ihre neue Kartenressource angezeigt wird. Sie verwenden sie später in diesem Tutorial, um die richtige Authentifizierung zu erstellen.

2. Wählen Sie den Ortsindex aus, den Sie verwenden möchten.
 - a. Wählen Sie in der Amazon Location-Konsole auf der Seite [Ortsindizes](#) die Option Ortsindex erstellen aus.
 - b. Fügen Sie einen Name und eine Beschreibung für die neue Ortsindex-Ressource hinzu. Notieren Sie sich den Namen, den Sie für die Ortsindex-Ressource verwenden. Sie benötigen sie später im Tutorial, um Ihre Skriptdatei zu erstellen.
 - c. Wählen Sie einen Datenanbieter aus.

 Note

Wählen Sie in den meisten Fällen den Datenanbieter aus, der dem bereits ausgewählten Kartenanbieter entspricht. Auf diese Weise können Sie sicherstellen, dass die Suchen mit den Karten übereinstimmen.

Wenn Ihre Anwendung Komponenten verfolgt oder weiterleitet, die Sie in Ihrem Unternehmen verwenden, z. B. Liefermodelle oder Mitarbeiter, können Sie HERE nur als Geolokalisierungsanbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS-Servicebedingungen](#).

- d. Wählen Sie die Option Datenspeicher aus. Für dieses Tutorial werden die Ergebnisse nicht gespeichert, sodass Sie Nein wählen können, nur für den einmaligen Gebrauch.
- e. Stimmen Sie den Allgemeinen Geschäftsbedingungen für Amazon Location zu und wählen Sie dann Ortsindex erstellen aus.
- f. Notieren Sie sich den ARN, der für Ihre neue Ortsindexressource angezeigt wird. Sie verwenden sie, um die richtige Authentifizierung im nächsten Abschnitt dieses Tutorials zu erstellen.

Einrichten der Authentifizierung für Ihre Anwendung

Die Anwendung, die Sie in diesem Tutorial erstellen, hat eine anonyme Nutzung, was bedeutet, dass sich Ihre Benutzer nicht bei anmelden müssen, um die Anwendung AWS zu verwenden. Standardmäßig erfordern die Amazon Location Service APIs jedoch eine Authentifizierung, um sie zu verwenden. Sie können entweder Amazon Cognito oder API-Schlüssel verwenden, um Authentifizierung und Autorisierung für anonyme Benutzer bereitzustellen. In diesem Tutorial erstellen Sie API-Schlüssel zur Verwendung in der Beispielanwendung.

 Note

Weitere Informationen zur Verwendung von API-Schlüsseln oder Amazon Cognito mit Amazon Location Service finden Sie unter [Gewähren des Zugriffs auf Amazon Location Service](#).

So richten Sie die Authentifizierung für Ihre Anwendung ein

1. Gehen Sie zur [Amazon Location-Konsole](#) und wählen Sie im linken Menü API-Schlüssel aus.
2. Klicken Sie auf API-Schlüssel erstellen.

 Important

Der API-Schlüssel, den Sie erstellen, muss sich in demselben AWS-Konto und derselben AWS Region befinden wie die Amazon Location Service-Ressourcen, die Sie im vorherigen Abschnitt erstellt haben.

3. Geben Sie auf der Seite API-Schlüssel erstellen die folgenden Informationen ein.
 - Name – Ein Name für Ihren API-Schlüssel, z. B. MyWebAppKey.
 - Ressourcen – Wählen Sie die Amazon Location Map- und Place-Indexressourcen aus, die Sie im vorherigen Abschnitt erstellt haben. Sie können mehr als eine Ressource hinzufügen, indem Sie Ressource hinzufügen auswählen. Auf diese Weise kann der API-Schlüssel mit diesen Ressourcen verwendet werden.
 - Aktionen – Geben Sie die Aktionen an, die Sie mit diesem API-Schlüssel autorisieren möchten. Sie müssen mindestens `geo:GetMap*` und `geo:SearchPlaceIndexForPosition` auswählen, damit das Tutorial wie erwartet funktioniert.
 - Sie können Ihrem API-Schlüssel optional eine Beschreibung, eine Ablaufzeit oder Tags hinzufügen. Sie können auch einen Referer (z. B. `*.example.com`) hinzufügen, um den Schlüssel auf die Verwendung von einer bestimmten Domain zu beschränken. Das bedeutet, dass das Tutorial nur von dieser Domain aus funktioniert.

Note

Es wird empfohlen, Ihre API-Schlüsselnutzung zu schützen, indem Sie entweder eine Ablaufzeit oder einen Referer festlegen, falls nicht beides der Fall ist.

4. Wählen Sie API-Schlüssel erstellen, um den API-Schlüssel zu erstellen.
5. Wählen Sie API-Schlüssel anzeigen und kopieren Sie den Schlüsselwert zur Verwendung später im Tutorial. Es wird das Format `habenv1.public.a1b2c3d4...`

Important

Sie benötigen diesen Schlüssel, wenn Sie den Code für Ihre Anwendung später in diesem Tutorial schreiben.

Erstellen des HTML-Codes für Ihre Anwendung

In diesem Tutorial erstellen Sie eine statische HTML-Seite, die eine Karte einbettet und es dem Benutzer ermöglicht, zu finden, was sich an einem Speicherort auf der Karte befindet. Die App besteht aus drei Dateien: einer HTML-Datei und einer CSS-Datei für die Webseite und einer JavaScript (.js)-Datei für den Code, der die Karte erstellt und auf die Interaktionen und Kartenereignisse des Benutzers reagiert.

Lassen Sie uns zunächst das HTML- und CSS-Framework erstellen, das für die Anwendung verwendet wird. Dies ist eine einfache Seite mit einem `<div>`-Element, das den Kartencontainer enthält, und einem `<pre>`-Element, um die JSON-Antworten auf Ihre Abfragen anzuzeigen.

So erstellen Sie den HTML-Code für Ihre Schnellstartanwendung

1. Erstellen Sie eine neue Datei mit dem Namen `quickstart.html`.
2. Bearbeiten Sie die Datei im Texteditor oder in einer Umgebung Ihrer Wahl. Fügen Sie der Datei den folgenden HTML-Code hinzu.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Quick start tutorial</title>
```

```
<!-- Styles -->
<link href="main.css" rel="stylesheet" />
</head>

<body>
  <header>
    <h1>Quick start tutorial</h1>
  </header>
  <main>
    <div id="map"></div>
    <aside>
      <h2>JSON Response</h2>
      <pre id="response"></pre>
    </aside>
  </main>
  <footer>This is a simple Amazon Location Service app. Pan and zoom. Click to
  see details about entities close to a point.</footer>

</body>
</html>
```

Dieser HTML-Code enthält einen Zeiger auf die CSS-Datei, die Sie im nächsten Schritt erstellen werden, einige Platzhalterelemente für die Anwendung und etwas erklärenden Text.

Es gibt zwei Platzhalterelemente, die Sie später in diesem Tutorial verwenden werden. Die erste ist das `<div id="map">`Element, das die Kartensteuerung enthält. Die zweite ist das `<pre id="response">`Element, das die Ergebnisse der Suche auf der Karte anzeigt.

3. Speichern Sie Ihre Datei.

Fügen Sie nun das CSS für die Webseite hinzu. Dadurch wird der Stil des Textes und der Platzhalterelemente für die Anwendung festgelegt.

So erstellen Sie das CSS für Ihre Schnellstartanwendung

1. Erstellen Sie eine neue Datei mit dem Namen im selben Ordner wie die Datei `quickstart.htmlmain.css`, die im vorherigen Verfahren erstellt wurde.
2. Bearbeiten Sie die Datei in einem beliebigen Editor, den Sie verwenden möchten. Fügen Sie der Datei den folgenden Text hinzu.

```
* {
  box-sizing: border-box;
  font-family: Arial, Helvetica, sans-serif;
}

body {
  margin: 0;
}

header {
  background: #000000;
  padding: 0.5rem;
}

h1 {
  margin: 0;
  text-align: center;
  font-size: 1.5rem;
  color: #ffffff;
}

main {
  display: flex;
  min-height: calc(100vh - 94px);
}

#map {
  flex: 1;
}

aside {
  overflow-y: auto;
  flex: 0 0 30%;
  max-height: calc(100vh - 94px);
  box-shadow: 0 1px 1px 0 #001c244d, 1px 1px 1px 0 #001c2426, -1px 1px 1px 0
#001c2426;
  background: #f9f9f9;
  padding: 1rem;
}

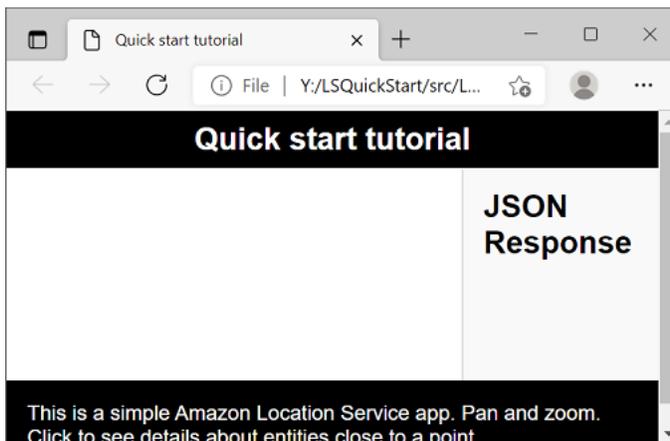
h2 {
  margin: 0;
}
```

```
pre {
  white-space: pre-wrap;
  font-family: monospace;
  color: #16191f;
}

footer {
  background: #000000;
  padding: 1rem;
  color: #ffffff;
}
```

Dadurch wird die Karte so eingestellt, dass sie den Raum ausfüllt, der nicht von anderen verwendet wird, der Bereich für unsere Antworten wird so eingestellt, dass er 30 % der Breite der App in Anspruch nimmt, und Farbe und Stile für den Titel und den erklärenden Text festgelegt.

3. Speichern Sie die Datei.
4. Sie können die `quickstart.html` Datei jetzt in einem Browser anzeigen, um das Layout der Anwendung anzuzeigen.



Als Nächstes fügen Sie das Kartensteuerelement der Anwendung hinzu.

Hinzufügen einer interaktiven Zuordnung zu Ihrer Anwendung

Nachdem Sie nun ein Framework und einen div-Platzhalter haben, können Sie die Kartensteuerung zu Ihrer Anwendung hinzufügen. In diesem Tutorial wird [MapLibre microSD JS](#) als Kartensteuerung verwendet, um Daten von Amazon Location Service abzurufen. Sie verwenden auch die

[Authentifizierungshelfer](#), um das Signieren von Aufrufen an die Amazon Location APIs mit Ihrem API-Schlüssel zu erleichtern.

So fügen Sie Ihrer Anwendung eine interaktive Zuordnung hinzu

1. Öffnen Sie die `quickstart.html` Datei, die Sie im vorherigen Abschnitt erstellt haben.
2. Fügen Sie Verweise auf die benötigten Bibliotheken und die Skriptdatei hinzu, die Sie erstellen werden. Die Änderungen, die Sie vornehmen müssen, werden unter **angezeigtgreen**.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Quick start tutorial</title>

    <!-- Styles -->
    <link href="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.css"
rel="stylesheet" />
    <link href="main.css" rel="stylesheet" />
  </head>

  <body>
    ...
    <footer>This is a simple Amazon Location Service app. Pan and zoom. Click to
    see details about entities close to a point.</footer>

    <!-- JavaScript dependencies -->
    <script src="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.js"></script>
    <script src="https://unpkg.com/@aws/amazon-location-client@1.x/dist/
amazonLocationClient.js"></script>
    <script src="https://unpkg.com/@aws/amazon-location-utilities-auth-helper@1.x/
dist/amazonLocationAuthHelper.js"></script>

    <!-- JavaScript for the app -->
    <script src="main.js"></script>
  </body>
</html>
```

Dadurch werden Ihrer App die folgenden Abhängigkeiten hinzugefügt:

- MapLibre Bol JS . Diese Bibliothek und dieses Stylesheet enthalten ein Kartensteuerelement, das Kartenkacheln anzeigt und Interaktivität wie Pan und Zoom enthält. Die Kontrolle ermöglicht auch Erweiterungen, z. B. das Zeichnen eigener Features auf der Karte.
- Amazon Location-Client . Dies bietet Schnittstellen für die Amazon Location-Funktionalität, die zum Abrufen von Kartendaten und zum Suchen nach Orten auf der Karte erforderlich ist. Der Amazon Location-Client basiert auf dem AWS SDK für JavaScript v3.
- Amazon Location Authentication Helper . Dies bietet hilfreiche Funktionen für die Authentifizierung von Amazon Location Service mit API-Schlüsseln oder Amazon Cognito .

In diesem Schritt wird auch ein Verweis auf hinzugefügt `main.js`, den Sie als Nächstes erstellen.

3. Speichern Sie die `quickstart.html`-Datei.
4. Erstellen Sie eine neue Datei mit dem Namen `main.js` im selben Ordner wie Ihre HTML- und CSS-Dateien und öffnen Sie sie zur Bearbeitung.
5. Fügen Sie der Datei das folgende Skript hinzu. Der Text in *Rot* sollte durch den API-Schlüsselwert, den Kartenressourcennamen und den Ortsressourcennamen ersetzt werden, den Sie zuvor erstellt haben, sowie durch die Regions-ID für Ihre Region (z. B. `us-east-1`).

```
// Amazon Location Service resource names:
const mapName = "explore.map";
const placesName = "explore.place";
const region = "your_region";
const apiKey = "v1.public.a1b2c3d4..."

// Initialize a map
async function initializeMap() {
  const mglMap = new maplibregl.Map({
    container: "map", // HTML element ID of map element
    center: [-77.03674, 38.891602], // Initial map centerpoint
    zoom: 16, // Initial map zoom
    style: 'https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/style-
descriptor?key=${apiKey}', // Defines the appearance of the map and authenticates
using an API key
  });

  // Add navigation control to the top left of the map
  mglMap.addControl(new maplibregl.NavigationControl(), "top-left");
```

```
    return mlg1Map;
  }

  async function main() {
    // Initialize map and Amazon Location SDK client:
    const map = await initializeMap();
  }

  main();
```

Dieser Code richtet Amazon Location-Ressourcen ein, konfiguriert und initialisiert dann eine MapLibre microSD JS-Kartensteuerung und platziert sie in Ihrem Element `<div>` mit der ID `map`.

Die `initializeMap()` Funktion ist wichtig zu verstehen. Es erstellt ein neues MapLibre Kartensteuerelement (mlglMaplokal genannt, aber `map` im Rest des Codes aufgerufen), das zum Rendern der Karte in Ihrer Anwendung verwendet wird.

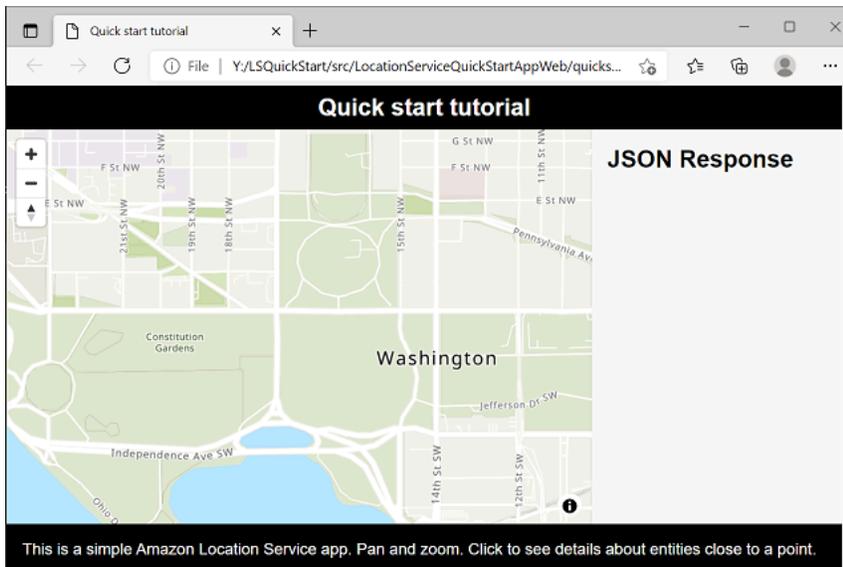
```
// Initialize the map
const mlg1Map = new maplibregl.Map({
  container: "map", // HTML element ID of map element
  center: [-77.03674, 38.891602], // Initial map centerpoint
  zoom: 16, // Initial map zoom
  style: 'https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/style-
descriptor?key=${apiKey}', // Defines the appearance of the map and authenticates
  using an API key
});
```

Wenn Sie ein neues MapLibre Kartensteuerelement erstellen, geben die von Ihnen übergebenen Parameter den Anfangsstatus des Kartensteuerelements an. Hier legen wir die folgenden Parameter fest.

- HTML-Container, der das Map-Div-Element in unserem HTML verwendet.
 - Die anfängliche Mitte der Zuordnung zu einem Punkt in Bol, DC.
 - Die Zoomstufe auf 16 (in eine Nachbarschafts- oder Blockstufe gezoomt).
 - Der für die Karte zu verwendende Stil, der MapLibre eine URL zum Abrufen der Kartenkacheln und andere Informationen zum Rendern der Karte bereitstellt. Beachten Sie, dass diese URL Ihren API-Schlüssel für die Authentifizierung enthält.
6. Speichern Sie Ihre JavaScript Datei und öffnen Sie sie mit einem Browser. Sie haben jetzt eine Karte auf Ihrer Seite, auf der Sie Pan- und Zoom-Aktionen verwenden können.

Note

Sie können diese App verwenden, um zu sehen, wie sich die MapLibre Kartensteuerung verhält. Sie können versuchen, Strg oder Verschiebung zu verwenden, während Sie eine Ziehen-Operation verwenden, um andere Möglichkeiten für die Interaktion mit der Karte anzuzeigen. Alle diese Funktionen sind anpassbar.



Ihre App ist fast vollständig. Im nächsten Abschnitt übernehmen Sie die Auswahl eines Standorts auf der Karte und zeigen die Adresse des ausgewählten Standorts an. Sie zeigen auch den resultierenden JSON-Code auf der Seite an, um die vollständigen Ergebnisse anzuzeigen.

Hinzufügen einer Suche zu Ihrer Anwendung

Der letzte Schritt für Ihre Anwendung besteht darin, die Suche auf der Karte hinzuzufügen. In diesem Fall fügen Sie eine umgekehrte Geocodierungssuche hinzu, bei der Sie die Elemente an einem Speicherort finden.

Note

Amazon Location Service bietet auch die Möglichkeit, nach Namen oder Adresse zu suchen, um die Standorte auf der Karte zu finden.

So fügen Sie Ihrer Anwendung Suchfunktionen hinzu

1. Öffnen Sie die `main.js` Datei, die Sie im vorherigen Abschnitt erstellt haben.
2. Ändern Sie die `main` Funktion wie gezeigt. Die Änderungen, die Sie vornehmen müssen, werden unter **angezeigtgreen**.

```
async function main() {
  // Create an authentication helper instance using an API key
  const authHelper = await amazonLocationAuthHelper.withAPIKey(apiKey);

  // Initialize map and Amazon Location SDK client:
  const map = await initializeMap();

  const client = new amazonLocationClient.LocationClient({
    region,
    ...authHelper.getLocationClientConfig(), // Provides configuration required to
    make requests to Amazon Location
  });

  // On mouse click, display marker and get results:
  map.on("click", async function (e) {
    // Set up parameters for search call
    let params = {
      IndexName: placesName,
      Position: [e.lngLat.lng, e.lngLat.lat],
      Language: "en",
      MaxResults: "5",
    };

    // Set up command to search for results around clicked point
    const searchCommand = new
amazonLocationClient.SearchPlaceIndexForPositionCommand(params);

    try {
      // Make request to search for results around clicked point
      const data = await client.send(searchCommand);

      // Write JSON response data to HTML
      document.querySelector("#response").textContent = JSON.stringify(data,
undefined, 2);
    }
  });
}
```

```
// Display place label in an alert box
alert(data.Results[0].Place.Label);
} catch (error) {
  // Write JSON response error to HTML
  document.querySelector("#response").textContent = JSON.stringify(error,
undefined, 2);

  // Display error in an alert box
  alert("There was an error searching.");
}
});
}
```

Dieser Code beginnt mit der Einrichtung des Amazon Location Authentication Helpers für die Verwendung Ihres API-Schlüssels.

```
const authHelper = await amazonLocationAuthHelper.withAPIKey(apiKey);
```

Dann verwendet es dieses Authentifizierungshelferobjekt und die Region, die Sie zum Erstellen eines neuen Amazon-Location-Clients verwenden.

```
const client = new amazonLocationClient.LocationClient({
  region,
  ...authHelper.getLocationClientConfig(),
});
```

Als Nächstes antwortet der Code dem Benutzer, der einen Spot auf dem Kartensteuerelement auswählt. Dazu wird ein MapLibre bereitgestelltes Ereignis für `abgefangenClick`.

```
map.on("click", async function(e) {
  ...
});
```

Das MapLibre `click` Ereignis stellt Parameter bereit, die den Breiten- und Längengrad enthalten, den der Benutzer ausgewählt hat (`e.lngLat`). Innerhalb des `click` Ereignisses erstellt der Code die `searchPlaceIndexForPositionCommand` um die Entitäten mit dem angegebenen Breiten- und Längengrad zu finden.

```
// Set up parameters for search call
let params = {
```

```
    IndexName: placesName,  
    Position: [e.lngLat.lng, e.lngLat.lat],  
    Language: "en",  
    MaxResults: "5"  
  };  
  
  // Set up command to search for results around clicked point  
  const searchCommand = new  
amazonLocationClient.SearchPlaceIndexForPositionCommand(params);  
  
  try {  
    // Make request to search for results around clicked point  
    const data = await client.send(searchCommand);  
    ...  
  });
```

Hier `IndexName` ist der Name der zuvor erstellten Ortsindex-Ressource, der `Position` Breiten- und Längengrad, nach dem gesucht werden soll, `Language` die bevorzugte Sprache für Ergebnisse und `MaxResults` weist Amazon Location an, maximal fünf Ergebnisse zurückzugeben.

Der verbleibende Code prüft auf einen Fehler und zeigt dann die Ergebnisse der Suche im `<pre>` Element mit dem Namen `response` und zeigt das oberste Ergebnis in einem Warnfeld an.

3. (Optional) Wenn Sie die `quickstart.html` Datei jetzt in einem Browser speichern und öffnen, wird Ihnen bei Auswahl eines Speicherorts auf der Karte der Name oder die Adresse des ausgewählten Speicherorts angezeigt.
4. Der letzte Schritt in der Anwendung besteht darin, die - MapLibre Funktionalität zu verwenden, um eine Markierung an dem Punkt hinzuzufügen, den der Benutzer ausgewählt hat. Ändern Sie die `main` Funktion wie folgt. Die Änderungen, die Sie vornehmen müssen, werden unter **angezeigt green**.

```
async function main() {  
  // Create an authentication helper instance using an API key  
  const authHelper = await amazonLocationAuthHelper.withAPIKey(apiKey);  
  
  // Initialize map and Amazon Location SDK client  
  const map = await initializeMap();  
  const client = new amazonLocationClient.LocationClient({  
    region,
```

```
...authHelper.getLocationClientConfig(), // Provides configuration required to
make requests to Amazon Location
});

// Variable to hold marker that will be rendered on click
let marker;

// On mouse click, display marker and get results:
map.on("click", async function (e) {
  // Remove any existing marker
  if (marker) {
    marker.remove();
  }

  // Render a marker on clicked point
  marker = new maplibregl.Marker().setLngLat([e.lngLat.lng,
e.lngLat.lat]).addTo(map);

  // Set up parameters for search call
  let params = {
    IndexName: placesName,
    Position: [e.lngLat.lng, e.lngLat.lat],
    Language: "en",
    MaxResults: "5",
  };

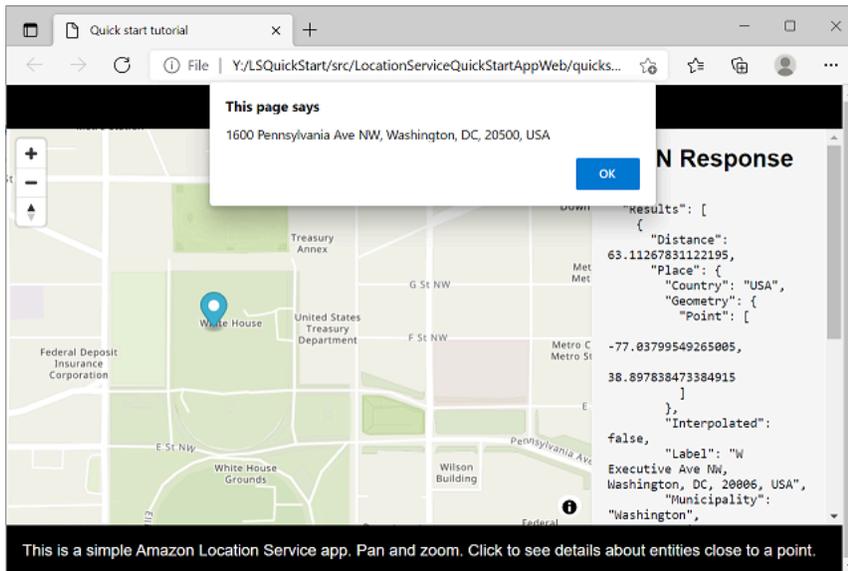
  // Set up command to search for results around clicked point
  const searchCommand = new
amazonLocationClient.SearchPlaceIndexForPositionCommand(params);

...

```

Dieser Code deklariert eine `marker` Variable, die jedes Mal ausgefüllt wird, wenn der Benutzer einen Standort auswählt, und zeigt, wo er ihn ausgewählt hat. Die Markierung wird automatisch vom Kartensteuerelement gerendert, sobald sie der Karte mit hinzugefügt wurde `.addTo(map)`. Der Code sucht auch nach einer vorherigen Markierung und entfernt sie, sodass jeweils nur eine Markierung auf dem Bildschirm angezeigt wird.

5. Speichern Sie die `main.js` Datei und öffnen Sie die `quickstart.html` Datei in einem Browser. Sie können die Karte wie zuvor schwenken und zoomen, aber wenn Sie jetzt einen Standort auswählen, werden Ihnen Details zu dem ausgewählten Standort angezeigt.



Ihre Schnellstartanwendung ist abgeschlossen. In diesem Tutorial erfahren Sie, wie Sie eine statische HTML-Anwendung erstellen, die:

- Erstellt eine Zuordnung, mit der Benutzer interagieren können.
- Verarbeitet ein Karteneignis (`click`).
- Ruft eine Amazon Location Service-API auf, insbesondere zum Durchsuchen der Karte an einem Standort mit `searchPlaceIndexForPosition`.
- Verwendet das MapLibre Kartensteuerelement, um eine Markierung hinzuzufügen.

Anzeigen der endgültigen Anwendung

Der endgültige Quellcode für diese Anwendung ist in diesem Abschnitt enthalten. Sie finden das endgültige Projekt auch [unter GitHub](#).

Sie finden auch eine Version der Anwendung, die Amazon Cognito anstelle von API-Schlüsseln [auf GitHub](#) verwendet.

Overview

Wählen Sie jede Registerkarte aus, um den endgültigen Quellcode der Dateien in diesem Schnellstart-Tutorial anzuzeigen.

Die Dateien sind:

- quickstart.html – das Framework für Ihre Anwendung, einschließlich der HTML-Elementinhaber für die Karte und die Suchergebnisse.
- main.css – das Stylesheet für die Anwendung.
- main.js – das Skript für Ihre Anwendung, das den Benutzer authentifiziert, die Zuordnung erstellt und nach einem click Ereignis sucht.

quickstart.html

Das HTML-Framework für die Schnellstartanwendung.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Quick start tutorial</title>

    <!-- Styles -->
    <link href="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.css"
rel="stylesheet" />
    <link href="main.css" rel="stylesheet" />
  </head>

  <body>
    ...
    <footer>This is a simple Amazon Location Service app. Pan and zoom. Click to see
details about entities close to a point.</footer>

    <!-- JavaScript dependencies -->
    <script src="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.js"></script>
    <script src="https://unpkg.com/@aws/amazon-location-client@1.x/dist/
amazonLocationClient.js"></script>
    <script src="https://unpkg.com/@aws/amazon-location-utilities-auth-helper@1.x/
dist/amazonLocationAuthHelper.js"></script>

    <!-- JavaScript for the app -->
    <script src="main.js"></script>
  </body>
</html>
```

main.css

Das Stylesheet für die Schnellstartanwendung.

```
* {
  box-sizing: border-box;
  font-family: Arial, Helvetica, sans-serif;
}

body {
  margin: 0;
}

header {
  background: #000000;
  padding: 0.5rem;
}

h1 {
  margin: 0;
  text-align: center;
  font-size: 1.5rem;
  color: #ffffff;
}

main {
  display: flex;
  min-height: calc(100vh - 94px);
}

#map {
  flex: 1;
}

aside {
  overflow-y: auto;
  flex: 0 0 30%;
  max-height: calc(100vh - 94px);
  box-shadow: 0 1px 1px 0 #001c244d, 1px 1px 1px 0 #001c2426, -1px 1px 1px 0 #001c2426;
  background: #f9f9f9;
  padding: 1rem;
}

h2 {
  margin: 0;
}
```

```
pre {
  white-space: pre-wrap;
  font-family: monospace;
  color: #16191f;
}

footer {
  background: #000000;
  padding: 1rem;
  color: #ffffff;
}
```

main.js

Der Code für die Schnellstartanwendung. Der Text in *Rot* sollte durch die entsprechenden Amazon Location-Objektnamen ersetzt werden.

```
// Amazon Location Service resource names:
const mapName = "explore.map";
const placesName = "explore.place";
const region = "your_region";
const apiKey = "v1.public.a1b2c3d4...";

// Initialize a map
async function initializeMap() {
  // Initialize the map
  const mlglMap = new maplibregl.Map({
    container: "map", // HTML element ID of map element
    center: [-77.03674, 38.891602], // Initial map centerpoint
    zoom: 16, // Initial map zoom
    style: `https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/style-
descriptor?key=${apiKey}`, // Defines the appearance of the map and authenticates
    using an API key
  });

  // Add navigation control to the top left of the map
  mlglMap.addControl(new maplibregl.NavigationControl(), "top-left");

  return mlglMap;
}

async function main() {
```

```
// Create an authentication helper instance using an API key
const authHelper = await amazonLocationAuthHelper.withAPIKey(apiKey);

// Initialize map and Amazon Location SDK client
const map = await initializeMap();
const client = new amazonLocationClient.LocationClient({
  region,
  ...authHelper.getLocationClientConfig(), // Provides configuration required to
make requests to Amazon Location
});

// Variable to hold marker that will be rendered on click
let marker;

// On mouse click, display marker and get results:
map.on("click", async function (e) {
  // Remove any existing marker
  if (marker) {
    marker.remove();
  }

  // Render a marker on clicked point
  marker = new maplibregl.Marker().setLngLat([e.lngLat.lng,
e.lngLat.lat]).addTo(map);

  // Set up parameters for search call
  let params = {
    IndexName: placesName,
    Position: [e.lngLat.lng, e.lngLat.lat],
    Language: "en",
    MaxResults: "5",
  };

  // Set up command to search for results around clicked point
  const searchCommand = new
amazonLocationClient.SearchPlaceIndexForPositionCommand(params);

  try {
    // Make request to search for results around clicked point
    const data = await client.send(searchCommand);

    // Write JSON response data to HTML
    document.querySelector("#response").textContent = JSON.stringify(data,
undefined, 2);
```

```
// Display place label in an alert box
alert(data.Results[0].Place.Label);
} catch (error) {
    // Write JSON response error to HTML
    document.querySelector("#response").textContent = JSON.stringify(error,
undefined, 2);

    // Display error in an alert box
    alert("There was an error searching.");
}
});
}

main();
```

Was ist als Nächstes?

Sie haben das Schnellstart-Tutorial abgeschlossen und sollten sich ein Bild davon machen, wie Amazon Location Service zum Erstellen von Anwendungen verwendet wird. Um Amazon Location besser zu nutzen, können Sie sich die folgenden Ressourcen ansehen:

- Weitere Informationen zu den [Konzepten von Amazon Location Service](#)
- Weitere Informationen [zur Verwendung von Amazon Location-Funktionen und -Funktionen](#)
- Sehen Sie sich [Codebeispiele mit Amazon Location](#) an, um dieses Beispiel zu erweitern und komplexere Anwendungen zu erstellen

Erstellen einer Android-App

In diesem Abschnitt erstellen Sie eine Android-Anwendung mit einer Karte und der Möglichkeit, an einem Standort zu suchen. Zunächst erstellen Sie Ihre Amazon Location-Ressourcen und eine Amazon Cognito-Identität für Ihre Anwendung.

Note

In diesem Tutorial werden Amplify AndroidStudio und die Kotlin-Sprache verwendet. Sie müssen und AndroidStudio Kotlin bereits mit einer Arbeitsumgebung eingerichtet haben.

Themen

- [Erstellen von Amazon Location-Ressourcen für Ihre App](#)
- [Einrichten der Authentifizierung für Ihre Anwendung](#)
- [Erstellen der Android-Basisanwendung](#)
- [Hinzufügen einer interaktiven Zuordnung zu Ihrer Anwendung](#)
- [Hinzufügen einer Suche zu Ihrer Anwendung](#)
- [Anzeigen der endgültigen Anwendung](#)
- [Was ist als Nächstes](#)

Erstellen von Amazon Location-Ressourcen für Ihre App

Wenn Sie sie noch nicht haben, müssen Sie die Amazon Location-Ressourcen erstellen, die Ihre Anwendung verwenden wird. Hier erstellen Sie eine Kartenressource, um Karten in Ihrer Anwendung anzuzeigen, und einen Ortsindex, um nach Speicherorten auf der Karte zu suchen.

So fügen Sie Ihrer Anwendung Standortressourcen hinzu

1. Wählen Sie den Kartenstil aus, den Sie verwenden möchten.
 - a. Wählen Sie in der Amazon-Standortkonsole auf der Seite [Karten](#) die Option Karte erstellen aus, um eine Vorschau der Kartenstile anzuzeigen.
 - b. Fügen Sie einen Name nund eine Beschreibung für die neue Kartenressource hinzu. Notieren Sie sich den Namen, den Sie für die Kartenressource verwenden. Sie benötigen sie später im Tutorial, um Ihre Skriptdatei zu erstellen.
 - c. Wählen Sie eine Zuordnung aus.

Note

Wenn Sie einen Kartenstil auswählen, wird auch ausgewählt, welchen Kartendatenanbieter Sie verwenden werden. Wenn Ihre Anwendung Komponenten verfolgt oder weiterleitet, die Sie in Ihrem Unternehmen verwenden, z. B. Liefermodelle oder Mitarbeiter, können Sie HERE nur als Geolokalisierungsanbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS - Servicebedingungen](#).

- d. Stimmen Sie den Allgemeinen Geschäftsbedingungen für Amazon Location zu und wählen Sie dann Karte erstellen aus. Sie können mit der ausgewählten Karte interagieren: Vergrößern, Verkleinern oder Vergrößern in eine beliebige Richtung.
 - e. Notieren Sie sich den Amazon-Ressourcennamen (ARN), der für Ihre neue Kartenressource angezeigt wird. Sie verwenden sie später in diesem Tutorial, um die richtige Authentifizierung zu erstellen.
2. Wählen Sie den Ortsindex aus, den Sie verwenden möchten.
- a. Wählen Sie in der Amazon Location-Konsole auf der Seite [Ortsindizes](#) die Option Ortsindex erstellen aus.
 - b. Fügen Sie einen Namen und eine Beschreibung für die neue Ortsindex-Ressource hinzu. Notieren Sie sich den Namen, den Sie für die Ortsindex-Ressource verwenden. Sie benötigen sie später im Tutorial, um Ihre Skriptdatei zu erstellen.
 - c. Wählen Sie einen Datenanbieter aus.

 Note

Wählen Sie in den meisten Fällen den Datenanbieter aus, der dem bereits ausgewählten Kartenanbieter entspricht. Auf diese Weise können Sie sicherstellen, dass die Suchen mit den Karten übereinstimmen.

Wenn Ihre Anwendung Komponenten verfolgt oder weiterleitet, die Sie in Ihrem Unternehmen verwenden, z. B. Liefermodelle oder Mitarbeiter, können Sie HERE nur als Geolokalisierungsanbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS -Servicebedingungen](#).

- d. Wählen Sie die Option Datenspeicher aus. Für dieses Tutorial werden die Ergebnisse nicht gespeichert, sodass Sie Nein wählen können, nur für den einmaligen Gebrauch.
- e. Stimmen Sie den Allgemeinen Geschäftsbedingungen für Amazon Location zu und wählen Sie dann Ortsindex erstellen aus.
- f. Notieren Sie sich den ARN, der für Ihre neue Ortsindexressource angezeigt wird. Sie verwenden sie, um die richtige Authentifizierung im nächsten Abschnitt dieses Tutorials zu erstellen.

Einrichten der Authentifizierung für Ihre Anwendung

Die Anwendung, die Sie in diesem Tutorial erstellen, hat eine anonyme Nutzung, was bedeutet, dass sich Ihre Benutzer nicht bei anmelden müssen AWS , um die Anwendung zu verwenden. Die Amazon Location Service APIs erfordern jedoch eine Authentifizierung, um zu verwenden. Sie können entweder API-Schlüssel oder Amazon Cognito verwenden, um Authentifizierung und Autorisierung für anonyme Benutzer bereitzustellen. In diesem Tutorial wird Amazon Cognito verwendet. Bevor Sie Amazon Cognito in Ihrer Anwendung verwenden können, müssen Sie einen Amazon Cognito-Identitätspool erstellen.

Note

Weitere Informationen zur Verwendung von Amazon Cognito oder API-Schlüsseln mit Amazon Location Service finden Sie unter [Gewähren des Zugriffs auf Amazon Location Service](#).

So richten Sie die Authentifizierung für Ihre Anwendung ein

1. Melden Sie sich bei der [Amazon-Cognito-Konsole](#) an.
2. Klicken Sie auf Manage Identity Pools (Identitäten-Pools verwalten).
3. Wählen Sie Neuen Identitäten-Pool erstellen und geben Sie dann einen Namen für Ihren Identitäten-Pool ein.

Important

Der Pool, den Sie erstellen, muss sich in demselben AWS-Konto und AWS derselben Region befinden wie die Amazon Location Service-Ressourcen, die Sie im vorherigen Abschnitt erstellt haben.

4. Wählen Sie im Abschnitt Einklappbare nicht authentifizierte Identitäten die Option Zugriff auf nicht authentifizierte Identitäten aktivieren aus.
5. Wählen Sie Pool erstellen.
6. Wählen Sie aus, welche IAM-Rollen Sie mit Ihrem Identitäten-Pool verwenden möchten.
7. Erweitern Sie Details anzeigen.
8. Geben Sie unter Nicht authentifizierte Identitäten einen Rollennamen ein.

9. Erweitern Sie den Abschnitt Richtliniendokument anzeigen und wählen Sie dann Bearbeiten, um Ihre Richtlinie hinzuzufügen.
10. Fügen Sie Ihre Richtlinie hinzu, um Zugriff auf Ihre -Ressourcen zu gewähren.

Im Folgenden finden Sie Richtlinienbeispiele für Karten, Orte, Tracker und Routen. Um die Beispiele für Ihre eigene Richtlinie zu verwenden, ersetzen Sie die Platzhalter *region* und *accountID*:

Die folgende Richtlinie gewährt schreibgeschützten Zugriff auf Amazon Location Service-Ressourcen. Geben Sie die richtigen ARNs für die Ressourcen an, die Sie im vorherigen Abschnitt erstellt haben. Wenn Sie sich nicht an die ARNs erinnern, finden Sie sie, indem Sie die entsprechende Ressource auf der Seite [Karten](#) der Amazon-Standortkonsole oder auf der Seite [Indizes platzieren](#) auswählen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "LocationReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:GetMapStyleDescriptor",
        "geo:GetMapGlyphs",
        "geo:GetMapSprites",
        "geo:GetMapTile",
        "geo:SearchPlaceIndex*"
      ],
      "Resource": [
        "arn:aws:geo:<region>:<accountID>:map/<ExampleMap>",
        "arn:aws:geo:<region>:<accountID>:place-index/<ExamplePlaceIndex>"
      ]
    }
  ]
}
```

Note

Während nicht authentifizierte Identitätspools für die Offenlegung auf ungesicherten Websites vorgesehen sind, beachten Sie, dass sie gegen standardmäßige, zeitlich begrenzte AWS Anmeldeinformationen ausgetauscht werden.

Es ist wichtig, die IAM-Rollen, die nicht authentifizierten Identitätspools zugeordnet sind, entsprechend einzugrenzen. Weitere Informationen zur Verwendung von und geeigneten Umfangsrichtlinien in Amazon Cognito mit Amazon Location Service finden Sie unter [Gewähren des Zugriffs auf Amazon Location Service](#).

11. Wählen Sie Zulassen, um Ihre Identitätspools zu erstellen.
12. Wählen Sie auf der Seite Beispielcode die Plattform von ausJavaScript. Kopieren oder notieren Sie sich dann die `IdentityPoolId`, die Sie später in Ihrem Browserskript verwenden werden.

Erstellen der Android-Basisanwendung

In diesem Tutorial erstellen Sie eine Android-Anwendung, die eine Karte einbettet und es dem Benutzer ermöglicht, herauszufinden, was sich an einem Speicherort auf der Karte befindet.

Lassen Sie uns zunächst eine leere Kotlin-Anwendung mit AndroidStudio dem neuen Projektassistenten von erstellen.

So erstellen Sie eine leere Anwendung (AndroidStudio)

1. Starten Sie AndroidStudio. Wählen Sie im Menü Datei , Neu , Neues Projekt aus.
2. Wählen Sie auf der Registerkarte Telefon und Tablet die Option Leere Aktivität und dann Weiter aus.
3. Wählen Sie einen Namen , Paketnamen und Speicherort für Ihre Anwendung aus.
4. Wählen Sie in der Dropdown-Liste für Sprache die Option Kotlin aus.
5. Wählen Sie Fertig stellen, um Ihre leere Anwendung zu erstellen.
6. (Optional). Standardmäßig enthält das für eine leere Anwendung verwendete Design eine Textleiste oben in der App mit dem Namen der Anwendung. Sie können die Textleiste entfernen, indem Sie die `app/manifests/AndroidManifest.xml` Datei in Ihrem Projekt bearbeiten. Ändern Sie die Zeile `android:theme="@style/Theme.<packagename>"` in `android:theme="@style/Theme.AppCompat.NoActionBar"`.

Als Nächstes fügen Sie das Kartensteuerelement der Anwendung hinzu.

Hinzufügen einer interaktiven Zuordnung zu Ihrer Anwendung

Nachdem Sie nun ein Framework haben, können Sie die Kartensteuerung zu Ihrer Anwendung hinzufügen. In diesem Tutorial werden Amplify-Bibliotheken für die Verwaltung der Kartenansicht in

der Anwendung verwendet. Die Kartensteuerung selbst ist Teil der [MapLibre microSD JS](#)-Bibliothek, Amplify und MapLibre ruft die Kartendaten von Amazon Location Service ab. Die [Amplify](#)-Bibliotheken erleichtern das Signieren von Aufrufen an die Amazon Location APIs mit Amazon Cognito .

Um die Zuordnung hinzuzufügen, führen Sie die folgenden Schritte aus:

1. Fügen Sie Ihrem Projekt die Standortabhängigkeiten hinzu.
2. Hinzufügen von Amazon Location Authenticaton SDK Android zu Ihrem Projekt
3. Richten Sie die Kartenansicht in Ihrem Layout ein.
4. Schreiben Sie den Code für die Anzeige der Karte.
5. Erstellen Sie eine Konfigurationsdatei für Ihre -Ressourcen.

Die folgenden Verfahren führen Sie durch jeden dieser Schritte.

1 – Hinzufügen von Standortabhängigkeiten zu Ihrem Projekt

1. Wählen AndroidStudioSie in das Menü Anzeigen und dann Tool Windows , Projekt aus. Dadurch wird das Projektfenster geöffnet, in dem Sie auf alle Dateien in Ihrem Projekt zugreifen können.
2. Öffnen Sie im Fenster Projekt Gradle Scripts in der Strukturansicht und wählen Sie die `build.gradle` Datei für Ihr Anwendungsmodul aus. Dadurch wird die `build.gradle` Datei zur Bearbeitung geöffnet.
3. Fügen Sie unten in der Datei im Abschnitt Abhängigkeiten die folgenden Abhängigkeiten hinzu. Die Änderungen, die Sie vornehmen müssen, werden unter angezeigt **green**. Die vorhandenen Abhängigkeiten sehen je nach Ihrer Version von AndroidStudio und den für Ihre Projekte ausgewählten Optionen möglicherweise nicht wie folgt aus.

```
dependencies {  
  
    implementation 'androidx.core:core-ktx:1.7.0'  
    implementation 'androidx.appcompat:appcompat:1.3.0'  
    implementation 'com.google.android.material:material:1.4.0'  
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'  
    testImplementation 'junit:junit:4.13.2'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'  
  
    // added for location services:  
    implementation 'org.maplibre.gl:android-sdk:9.5.2'  
}
```

```
implementation 'com.amplifyframework:aws-auth-cognito:1.31.3'  
implementation 'com.amplifyframework:aws-geo-location:0.3.1'  
implementation 'com.amplifyframework:maplibre-adapter:0.3.1'  
}
```

4. Nachdem Sie die Gradle-Abhängigkeiten bearbeitet haben, AndroidStudio müssen Sie das Projekt erneut synchronisieren. Oben im `build.gradle` Bearbeitungsfenster werden Sie AndroidStudio von aufgefordert, zu synchronisieren. Wählen Sie aus `Sync Now`, ob Sie Ihr Projekt synchronisieren möchten, bevor Sie fortfahren.
5. Öffnen Sie im Fenster Projekt App , Java , <Ihr Paketname> in der Strukturansicht und wählen Sie die `MainActivity.kt` Datei aus.
6. Erweitern Sie in der `MainActivity.kt`-Datei den Abschnitt Imports, falls noch nicht geschehen. Fügen Sie die **green** Importe aus der folgenden Liste hinzu. Auch hier stimmen Ihre vorhandenen Importe je nach Ihren AndroidStudio Optionen möglicherweise nicht überein.

```
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.util.Log  
import android.view.animation.OvershootInterpolator  
import android.widget.ImageView  
import android.widget.TextView  
import com.amplifyframework.AmplifyException  
import com.amplifyframework.auth.cognito.AWSCognitoAuthPlugin  
import com.amplifyframework.core.Amplify  
import com.amplifyframework.geo.location.AWSLocationGeoPlugin  
import com.amplifyframework.geo.location.models.AmazonLocationPlace  
import com.amplifyframework.geo.maplibre.view.MapLibreView  
import com.amplifyframework.geo.maplibre.view.support.fadeIn  
import com.amplifyframework.geo.maplibre.view.support.fadeOut  
import com.amplifyframework.geo.models.Coordinates  
import com.amplifyframework.geo.options.GeoSearchByCoordinatesOptions  
import com.mapbox.mapboxsdk.camera.CameraPosition  
import com.mapbox.mapboxsdk.geometry.LatLng  
import com.mapbox.mapboxsdk.maps.MapboxMap  
import kotlin.math.abs
```

Note

Wenn Ihre Importe nicht `android.util.log`, `android.widget.ImageView` oder `android.widget.TextView` enthalten, müssen Sie diese Importe ebenfalls hinzufügen.

7. Speichern Sie die von Ihnen bearbeiteten Dateien, `build.gradle` und `MainActivity.kt`.

Dadurch werden Ihrer App die folgenden Abhängigkeiten hinzugefügt:

- AWS Amplify. Amplify bietet viele hilfreiche Funktionen für die Arbeit mit AWS, einschließlich Amazon Location.
- MapLibre Kotlin JS. Diese Bibliothek enthält ein Kartensteuerelement, das Kartenkacheln anzeigt und Interaktivität wie Pan und Zoom enthält. Die Kontrolle ermöglicht auch Erweiterungen, z. B. das Zeichnen eigener Features auf der Karte.

2 – Hinzufügen von Amazon Location Authentication SDK Android zu Ihrem Projekt

1. Geben Sie im Fenster Pakete die Android-URL des Amazon Location Authentication SDK ein: <https://github.com/aws-geospatial/amazon-location-mobile-auth-sdk-android/> und drücken Sie die Eingabetaste.
2. Wählen Sie die aus `amazon-location-mobile-auth-sdk-android` und drücken Sie die Schaltfläche Paket hinzufügen.
3. Wählen Sie die aus `AWSAuthSDK` und drücken Sie die Schaltfläche Paket hinzufügen.

3 – Fügen Sie Ihrem Projekt die Kartenansicht hinzu

1. Öffnen Sie im Fenster AndroidStudio Projekt die App, `res`, `Layout` in der Strukturansicht und wählen Sie die `activity_main.xml` Datei aus. Dadurch wird die `build.gradle` Datei zur Bearbeitung geöffnet. Wenn es noch nicht ausgewählt ist, wählen Sie oben rechts im `activity_main.xml` Bearbeitungsfenster Code aus, um die XML für Ihr Layout direkt anzuzeigen.
2. Löschen Sie den von TextView AndroidStudioerstellten Standard und fügen Sie stattdessen eine Amplify-Map-Ansicht hinzu. Die von Ihnen hinzugefügte XML wird in **green** wie folgt angezeigt.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://
schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```
tools:context=".MainActivity">

<com.amplifyframework.geo.maplibre.view.MapLibreView
    android:id="@+id/mapView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:mapbox_foregroundLoadColor="@color/white"
    app:mapbox_renderTextureMode="true"
    app:mapbox_renderTextureTranslucentSurface="true" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

3. Speichern Sie die bearbeitete Datei, `activity_main.xml`.

Jetzt können Sie den Code schreiben, um die Kartenansicht zu erstellen.

3 – Fügen Sie den Code für Ihre interaktive Karte hinzu

1. Öffnen Sie im Fenster Projekt App , Java , <Ihr Paketname> in der Strukturansicht und wählen Sie die `MainActivity.kt` Datei aus.
2. Fügen Sie nun folgenden Code in die Datei `MainActivity.kt` ein: Der neue Code befindet sich in **green**.

```
class MainActivity : AppCompatActivity() {
    private val mapView by lazy {
        findViewById<MapLibreView>(R.id.mapView)
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        initAmplify()
        setContentView(R.layout.activity_main)

        mapView.getMapAsync { map ->
            val initialPosition = LatLng(47.6160281982247, -122.32642111977668)
            map.cameraPosition = CameraPosition.Builder()
                .target(initialPosition)
                .zoom(13.0)
                .build()
        }
    }
}
```

```
private fun initAmplify() {
    try {
        Amplify.addPlugin(AWSCognitoAuthPlugin())
        Amplify.addPlugin(AWSLocationGeoPlugin())
        Amplify.configure(applicationContext)
        Log.i("AndroidQuickStart", "Initialized Amplify")
    } catch (error: AmplifyException) {
        Log.e("AndroidQuickStart", "Could not initialize Amplify", error)
    }
}

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

override fun onResume() {
    super.onResume()
    mapView?.onResume()
}

override fun onPause() {
    super.onPause()
    mapView?.onPause()
}

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

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

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

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

```
}  
}
```

3. Speichern Sie die MainActivity.kt-Datei.

Sehen wir uns an, was der neue Code macht.

- `mapView` -Funktion – Dies ist eine Hilfsfunktion, um die `mapView` zu finden und darauf zu verweisen, die Sie im Layout erstellt haben.
- `initAmplify` – Diese Funktion initialisiert die Amplify-Bibliothek.
- `mapView.getMapAsync` – Dieser Code namens `in onCreate` richtet die anfängliche Position und die Zoomstufe der Kartenansicht ein.
- Überschriebene Ereignisse (`onStart` usw.) – Diese sollen sicherstellen, dass die `mapView` die Ereignisse abrufen, die sie für den ordnungsgemäßen Betrieb innerhalb der Anwendung benötigt.

Jetzt benötigt die App eine geringe Konfiguration und kann ausgeführt werden, wobei eine interaktive Karte angezeigt wird.

4 – Erstellen einer Konfiguration für Amplify

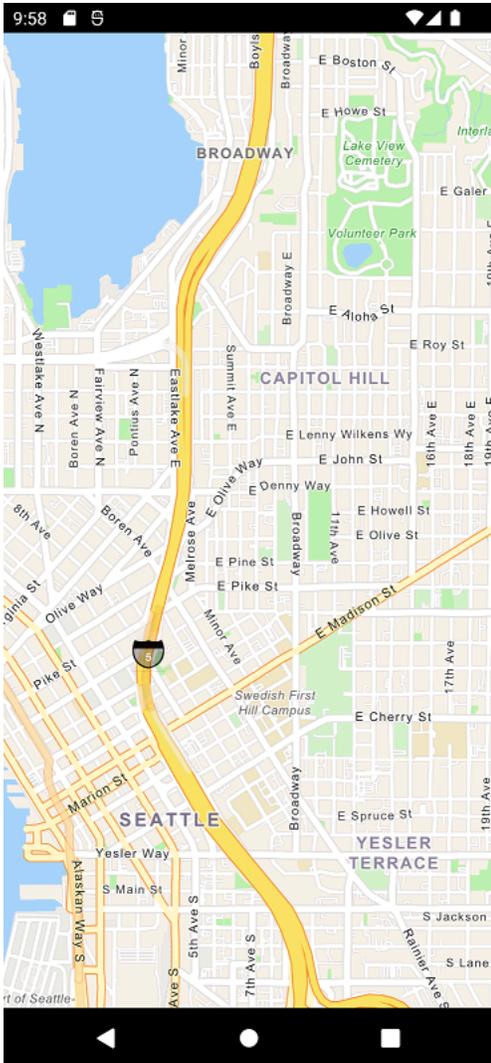
1. Wählen AndroidStudio Sie in das Menü Datei und dann Neu, Android-Ressourcenverzeichnis aus. Wählen Sie für den Ressourcentyp die Option Raw aus der Dropdown-Liste aus. Dadurch erhält das Verzeichnis auch `raw` automatisch den Namen `.` Wählen Sie OK, um das Verzeichnis zu erstellen.
2. Öffnen Sie im Fenster Projekt die App `, res ,raw` in der Baumansicht. Wählen Sie in der Strukturansicht Raw aus oder klicken Sie mit der rechten Maustaste darauf und wählen Sie New `, File` . Wählen Sie für den neuen Dateinamen aus `amplifyconfiguration.json` und drücken Sie dann die Eingabetaste.
3. Fügen Sie in der neuen Datei den folgenden Code hinzu. Ersetzen Sie den *roten Text* durch Werte aus Ihrem eigenen Amazon Location und Amazon Cognito-Ressourcen. Sie haben diese Ressourcen zu Beginn des Kapitels Schnellstart erstellt.

```
{  
  "UserAgent": "aws-amplify-cli/2.0",  
  "Version": "1.0",  
  "geo": {  
    "plugins": {
```

```
"awsLocationGeoPlugin": {
  "region": "<Amazon Location region>",
  "maps": {
    "items": {
      "<Amazon Location map resource name>": {
        "style": "<A name you choose for the style of the resource>"
      }
    },
    "default": "<the map resource name again>"
  },
  "searchIndices": {
    "items": [
      "<Amazon Location place resource name>"
    ],
    "default": "<the place resource name again>"
  }
}
},
"auth": {
  "plugins": {
    "awsCognitoAuthPlugin": {
      "UserAgent": "aws-amplify-cli/0.1.0",
      "Version": "0.1.0",
      "IdentityManager": {
        "Default": {}
      },
      "CredentialsProvider": {
        "CognitoIdentity": {
          "Default": {
            "PoolId": "<your Amazon Cognito identity pool ID>",
            "Region": "<the region of your Amazon Cognito identity pool>"
          }
        }
      }
    }
  }
}
}
```

4. Speichern Sie die `amplifyconfiguration.json`-Datei.

Sie können jetzt die Anwendung erstellen. Um es auszuführen, müssen Sie möglicherweise ein Gerät einrichten, um in zu emulieren AndroidStudio, oder die App auf Ihrem eigenen Gerät verwenden. Verwenden Sie diese App, um zu sehen, wie sich die Amplify-Kartensteuerung verhält. Sie können schwenken, indem Sie auf die Karte ziehen und zum Zoomen verkleinern. Sie können selbst ändern, wie die Kartensteuerung funktioniert, um sie an die Anforderungen Ihrer Anwendung anzupassen.



Im nächsten Abschnitt fügen Sie eine Markierung auf der Karte hinzu und zeigen die Adresse des Standorts an, an dem sich die Markierung befindet, während Sie die Karte verschieben.

Hinzufügen einer Suche zu Ihrer Anwendung

Der letzte Schritt für Ihre Anwendung besteht darin, die Suche auf der Karte hinzuzufügen. In diesem Fall fügen Sie eine umgekehrte Geocodierungssuche hinzu, bei der Sie die Elemente an einem Speicherort finden. Um die Verwendung für eine Android-App zu vereinfachen, durchsuchen wir die Mitte des Bildschirms. Um einen neuen Speicherort zu finden, verschieben Sie die Zuordnung an

die Stelle, an der Sie suchen möchten. Wir platzieren eine Markierung in der Mitte der Karte, um anzuzeigen, wo wir suchen.

Note

Amazon Location Service bietet auch die Möglichkeit, nach Namen oder Adresse zu suchen, um die Standorte auf der Karte zu finden.

Das Hinzufügen der Suche besteht aus zwei Teilen.

1. Fügen Sie eine Markierung in der Mitte des Bildschirms hinzu, um dem Benutzer anzuzeigen, wo wir suchen.
2. Fügen Sie ein Textfeld für Ergebnisse hinzu, durchsuchen Sie dann, was sich an der Position der Markierung befindet, und zeigen Sie im Textfeld an.

So fügen Sie Ihrer Anwendung eine Markierung hinzu

1. Speichern Sie dieses Image in Ihrem Projekt, in der -App, in einem zeichnungsfähigen Ordner als `red_marker.png` (Sie können auch [von GitHub](#) aus auf das Image zugreifen). Alternativ können Sie Ihr eigenes Image erstellen. Sie können auch eine PNG-Datei mit Transparenz für die Teile verwenden, die nicht angezeigt werden sollen.



2. Erweitern Sie AndroidStudio in App, res, Layout in der Strukturansicht und öffnen Sie dann die `activity_main.xml` Datei.
3. Fügen Sie ein `ImageView` für die Markierung hinzu. Die Änderungen, die Sie vornehmen müssen, werden unter **angezeigtgreen**.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://
schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
```

```

    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/map_view_marker"
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:elevation="2dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/red_marker" />

    <com.amplifyframework.geo.maplibre.view.MapLibreView
        android:id="@+id/mapView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:mapbox_foregroundLoadColor="@color/white"
        app:mapbox_renderTextureMode="true"
        app:mapbox_renderTextureTranslucentSurface="true" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

4. Speichern Sie Ihre Dateien und erstellen und führen Sie Ihre App (optional) aus, um eine Vorschau der Funktionalität anzuzeigen.

Ihre App hat jetzt eine Markierung auf dem Bildschirm. In diesem Fall handelt es sich um ein statisches Image, das sich nicht bewegt. Es wird verwendet, um die Mitte der Kartenansicht anzuzeigen, in der wir suchen werden. Im nächsten Verfahren fügen wir die Suche an dieser Stelle hinzu.

So fügen Sie Ihrer App eine Suche an einem Speicherort hinzu

1. Wenn es noch nicht geöffnet ist, öffnen Sie die `activity_main.xml` Datei, wie im vorherigen Verfahren. Fügen Sie den folgenden Code hinzu, wobei sich die Änderungen, die Sie vornehmen müssen, in **grün** befinden. Dadurch wird eine erstellte `TextView`, in der wir die Ergebnisse unserer Suche am Standort sehen können.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://
schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/map_view_marker"
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:elevation="2dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/red_marker" />

    <TextView
        android:id="@+id/description_text_view"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="70dp"
        android:background="#E6FFFFFF"
        android:elevation="4dp"
        android:padding="14dp"
        android:textAlignment="center"
        android:textColor="@android:color/black"
        app:layout_constraintBottom_toBottomOf="parent" />

    <com.amplifyframework.geo.maplibre.view.MapLibreView
        android:id="@+id/mapView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:mapbox_foregroundLoadColor="@color/white"
        app:mapbox_renderTextureMode="true"
        app:mapbox_renderTextureTranslucentSurface="true" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

2. Öffnen Sie in der App den Ordner , java , <package name> in AndroidStudio die MainActivity.kt Datei . Ändern Sie den Code wie gezeigt, wobei die Änderungen grün angezeigt werden (in diesem Fall wird nur ein Teil der Datei angezeigt).

```
...
class MainActivity : AppCompatActivity() {
    private val mapView by lazy {
        findViewById<MapLibreView>(R.id.mapView)
    }

    private val descriptionView by lazy {
        findViewById<TextView>(R.id.description_text_view)
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        initAmplify()
        setContentView(R.layout.activity_main)

        mapView.getMapAsync { map ->
            val initialPosition = LatLng(47.6160281982247, -122.32642111977668)
            map.cameraPosition = CameraPosition.Builder()
                .target(initialPosition)
                .zoom(13.0)
                .build()

            map.addOnCameraMoveStartedListener { toggleDescriptionText() }
            map.addOnCameraIdleListener { reverseGeocode(map) }
        }
    }

    private fun reverseGeocode(map: MapboxMap) {
        val options = GeoSearchByCoordinatesOptions.builder()
            .maxResults(1)
            .build()

        val centerCoordinates = Coordinates().apply {
            longitude = map.cameraPosition.target.longitude
            latitude = map.cameraPosition.target.latitude
        }
        Amplify.Geo.searchByCoordinates(centerCoordinates, options,
            { result ->
                result.places.firstOrNull()?.let { place ->
```

```
        val amazonPlace = (place as AmazonLocationPlace)
        runOnUiThread { toggleDescriptionText(amazonPlace.label) }
    }
},
{ exp ->
    Log.e("AndroidQuickStart", "Failed to reverse geocode : $exp")
}
)
}

private fun toggleDescriptionText(label: String? = "") {
    if (label.isNullOrBlank()) {
        descriptionView.fadeOut()
    } else {
        descriptionView.text = label
        descriptionView.fadeIn()
    }
}

private fun initAmplify() {
    try {
        Amplify.addPlugin(AWSCognitoAuthPlugin())
        Amplify.addPlugin(AWSLocationGeoPlugin())
        Amplify.configure(applicationContext)
        Log.i("AndroidQuickStart", "Initialized Amplify")
    } catch (error: AmplifyException) {
        Log.e("AndroidQuickStart", "Could not initialize Amplify", error)
    }
}

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

...

```

Dieser Code funktioniert mit der Kartenansicht. Die Kartenansicht in MapLibre wird durch eine virtuelle Kameraposition definiert. Das Verschieben der Karte kann sich als Verschieben dieser virtuellen Kamera vorstellen.

- `descriptionView` -Funktion – Dies ist eine Hilfsfunktion, um die zu finden und auf die zu verweisen `TextView`, die Sie im Layout erstellt haben.

- `onCreate` – Diese Funktion wird aktualisiert, um zwei neue Ereignisse zu registrieren.

Das `onCameraMove` Ereignis tritt auf, wenn der Benutzer die Karte verschiebt. Im Allgemeinen möchten wir beim Verschieben der Karte die Suche ausblenden, bis der Benutzer mit dem Verschieben der Karte fertig ist.

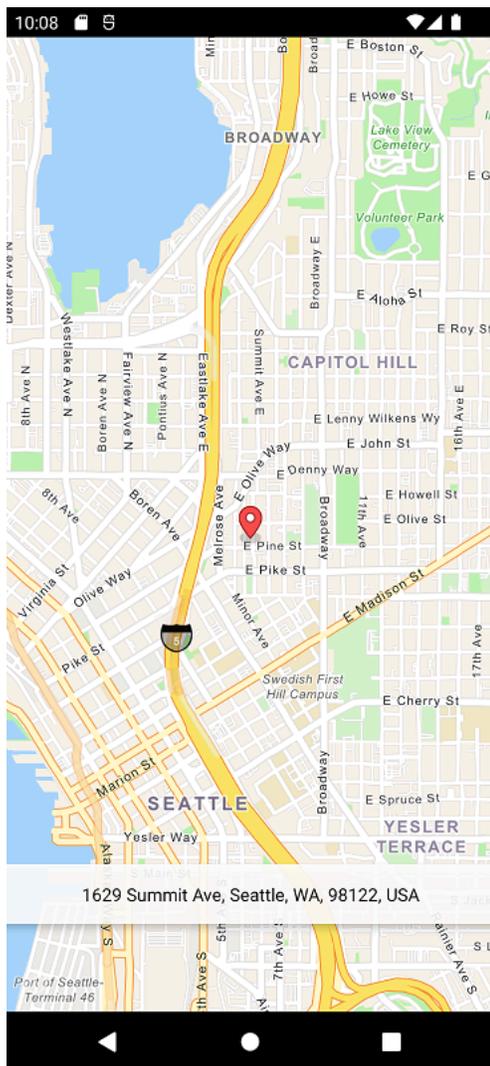
Das `-onCameraIdle` Ereignis tritt auf, wenn der Benutzer das Verschieben der Karte anhält. Dieses Ereignis ruft unsere `reverseGeocode` Funktion auf, um in der Mitte der Karte zu suchen.

- `reverseGeocode` – Diese Funktion, die im Ereignis aufgerufen wird `onCameraIdle`, sucht in der Mitte der Karte nach einem Standort und aktualisiert die `TextView` um die Ergebnisse anzuzeigen. Es verwendet das Kameraziel, das die Mitte der Karte definiert (wohin die Kamera schaut).
- `toggleDescriptionText` – Diese Funktion aktualisiert den Ergebnistext oder blendet aus, `TextView` wenn nichts zu sehen ist.

3. Speichern Sie Ihre Dateien und erstellen und führen Sie Ihre App aus, damit sie funktioniert.

Ihre Schnellstartanwendung ist abgeschlossen. In diesem Tutorial erfahren Sie, wie Sie eine Android-Anwendung erstellen, die:

- Erstellt eine Zuordnung, mit der Benutzer interagieren können.
- Verarbeitet mehrere Kartenereignisse, die dem Benutzer zugeordnet sind, die die Kartenansicht ändert (`cameraMoveStarted` und `CameraIdle`).
- Ruft eine Amazon Location Service-API auf, insbesondere um die Karte mithilfe der `-searchByCoordinatesOperation` von Amplify an einem Standort zu durchsuchen.



Anzeigen der endgültigen Anwendung

Das endgültige Projekt und der Quellcode für diese Anwendung sind [auf GitHub](#) verfügbar.

Was ist als Nächstes

Sie haben das Schnellstart-Tutorial abgeschlossen und sollten sich ein Bild davon machen, wie Amazon Location Service zum Erstellen von Anwendungen verwendet wird. Um Amazon Location besser zu nutzen, können Sie sich die folgenden Ressourcen ansehen:

- Weitere Informationen zu den [Konzepten von Amazon Location Service](#)
- Weitere Informationen [zur Verwendung von Amazon Location-Funktionen und -Funktionen](#)
- Erfahren Sie, wie Sie dieses Beispiel erweitern und komplexere Anwendungen erstellen können, indem Sie sich [Codebeispiele mit Amazon Location](#) ansehen

Erstellen einer iOS-App

In diesem Abschnitt erstellen Sie eine iOS-Anwendung mit einer Karte und der Möglichkeit, an einem Standort zu suchen. Zunächst erstellen Sie Ihre Amazon Location-Ressourcen und eine Amazon Cognito-Identität für Ihre Anwendung.

Note

In diesem Tutorial werden Amplify, Xcode und die Swift-Sprache verwendet. Sie müssen Xcode bereits mit einer Arbeitsumgebung eingerichtet haben.

Themen

- [Erstellen von Amazon Location-Ressourcen für Ihre App](#)
- [Einrichten der Authentifizierung für Ihre Anwendung](#)
- [Erstellen der iOS-Basisanwendung](#)
- [Hinzufügen einer interaktiven Zuordnung zu Ihrer Anwendung](#)
- [Hinzufügen einer Suche zu Ihrer Anwendung](#)
- [Was ist als Nächstes?](#)

Erstellen von Amazon Location-Ressourcen für Ihre App

Wenn Sie sie noch nicht haben, müssen Sie die Amazon Location-Ressourcen erstellen, die Ihre Anwendung verwenden wird. Hier erstellen Sie eine Kartenressource, um Karten in Ihrer Anwendung anzuzeigen, und einen Ortsindex, um nach Speicherorten auf der Karte zu suchen.

So fügen Sie Ihrer Anwendung Standortressourcen hinzu

1. Wählen Sie den Kartenstil aus, den Sie verwenden möchten.
 - a. Wählen Sie in der Amazon-Standortkonsole auf der Seite [Karten](#) die Option Karte erstellen aus, um eine Vorschau der Kartenstile anzuzeigen.
 - b. Fügen Sie einen Namen und eine Beschreibung für die neue Kartenressource hinzu. Notieren Sie sich den Namen, den Sie für die Kartenressource verwenden. Sie benötigen sie später im Tutorial, um Ihre Skriptdatei zu erstellen.
 - c. Wählen Sie eine Zuordnung aus.

 Note

Wenn Sie einen Kartenstil auswählen, wird auch ausgewählt, welchen Kartendatenanbieter Sie verwenden werden. Wenn Ihre Anwendung Komponenten verfolgt oder weiterleitet, die Sie in Ihrem Unternehmen verwenden, z. B. Liefermodelle oder Mitarbeiter, können Sie HERE nur als Geolokalisierungsanbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS - Servicebedingungen](#).

- d. Stimmen Sie den Allgemeinen Geschäftsbedingungen für Amazon Location zu und wählen Sie dann Karte erstellen aus. Sie können mit der ausgewählten Karte interagieren: Vergrößern, Verkleinern oder Vergrößern in eine beliebige Richtung.
 - e. Notieren Sie sich den Amazon-Ressourcennamen (ARN), der für Ihre neue Kartenressource angezeigt wird. Sie verwenden sie später in diesem Tutorial, um die richtige Authentifizierung zu erstellen.
2. Wählen Sie den Ortsindex aus, den Sie verwenden möchten.
- a. Wählen Sie in der Amazon Location-Konsole auf der Seite [Ortsindizes](#) die Option Ortsindex erstellen aus.
 - b. Fügen Sie einen Namen und eine Beschreibung für die neue Ortsindex-Ressource hinzu. Notieren Sie sich den Namen, den Sie für die Ortsindex-Ressource verwenden. Sie benötigen sie später im Tutorial, um Ihre Skriptdatei zu erstellen.
 - c. Wählen Sie einen Datenanbieter aus.

 Note

Wählen Sie in den meisten Fällen den Datenanbieter aus, der dem bereits ausgewählten Kartenanbieter entspricht. Auf diese Weise können Sie sicherstellen, dass die Suchen mit den Karten übereinstimmen.

Wenn Ihre Anwendung Komponenten verfolgt oder weiterleitet, die Sie in Ihrem Unternehmen verwenden, z. B. Liefermodelle oder Mitarbeiter, können Sie HERE nur als Geolokalisierungsanbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS -Servicebedingungen](#).

- d. Wählen Sie die Option Datenspeicher aus. Für dieses Tutorial werden die Ergebnisse nicht gespeichert, sodass Sie Nein wählen können, nur für den einmaligen Gebrauch.

- e. Stimmen Sie den Allgemeinen Geschäftsbedingungen für Amazon Location zu und wählen Sie dann Ortsindex erstellen aus.
- f. Notieren Sie sich den ARN, der für Ihre neue Ortsindexressource angezeigt wird. Sie verwenden sie, um die richtige Authentifizierung im nächsten Abschnitt dieses Tutorials zu erstellen.

Einrichten der Authentifizierung für Ihre Anwendung

Die Anwendung, die Sie in diesem Tutorial erstellen, hat eine anonyme Nutzung, was bedeutet, dass sich Ihre Benutzer nicht bei anmelden müssen AWS , um die Anwendung zu verwenden. Die Amazon Location Service APIs erfordern jedoch eine Authentifizierung, um zu verwenden. Sie können entweder API-Schlüssel oder Amazon Cognito verwenden, um Authentifizierung und Autorisierung für anonyme Benutzer bereitzustellen. In diesem Tutorial wird Amazon Cognito verwendet. Bevor Sie Amazon Cognito in Ihrer Anwendung verwenden können, müssen Sie einen Amazon Cognito-Identitätspool erstellen.

Note

Weitere Informationen zur Verwendung von Amazon Cognito oder API-Schlüsseln mit Amazon Location Service finden Sie unter [Gewähren des Zugriffs auf Amazon Location Service](#).

Erstellen einer IAM-Richtlinie für Ihre Kartenanwendung

1. Melden Sie sich bei der IAM-Konsole unter <https://console.aws.amazon.com/iam/> mit Ihrem Benutzer an, der über Administratorrechte verfügt.
2. Wählen Sie im Navigationsbereich Policies.
3. Wählen Sie im Inhaltsbereich die Option Create policy (Richtlinie erstellen).
4. Wählen Sie die Option JSON aus und kopieren Sie den Text aus dem folgenden JSON-Richtliniendokument. Fügen Sie den folgenden Text in das JSON-Eingabefeld ein.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "LocationReadOnly",
```

```

    "Effect": "Allow",
    "Action": [
      "geo:GetMapStyleDescriptor",
      "geo:GetMapGlyphs",
      "geo:GetMapSprites",
      "geo:GetMapTile",
      "geo:SearchPlaceIndex*"
    ],
    "Resource": [
      "arn:aws:geo:<region>:<accountID>:map/<ExampleMap>",
      "arn:aws:geo:<region>:<accountID>:place-index/<ExamplePlaceIndex>"
    ]
  }
]
}

```

Dies ist ein Richtlinienbeispiel für Maps. Um das Beispiel für Ihre eigene Richtlinie zu verwenden, ersetzen Sie die Platzhalter *region* und *accountID*:

Die folgende Richtlinie gewährt schreibgeschützten Zugriff auf Amazon Location Service-Ressourcen. Geben Sie die richtigen ARNs für die Ressourcen an, die Sie im vorherigen Abschnitt erstellt haben. Wenn Sie sich nicht an die ARNs erinnern, finden Sie sie, indem Sie die entsprechende Ressource auf der Seite [Karten](#) der Amazon-Standortkonsole oder auf der Seite [Indizes platzieren](#) auswählen.

Note

Während nicht authentifizierte Identitätspools für die Offenlegung auf ungesicherten Websites vorgesehen sind, beachten Sie, dass sie gegen standardmäßige, zeitlich begrenzte AWS Anmeldeinformationen ausgetauscht werden.

Es ist wichtig, die IAM-Rollen, die nicht authentifizierte Identitätspools zugeordnet sind, entsprechend einzugrenzen. Weitere Informationen zur Verwendung von und geeigneten Umfangsrichtlinien in Amazon Cognito mit Amazon Location Service finden Sie unter [Gewähren des Zugriffs auf Amazon Location Service](#).

5.

Note

Sie können jederzeit zwischen den Editoroptionen Visual und JSON wechseln. Wenn Sie jedoch Änderungen vornehmen oder auf der Registerkarte Visual editor die Option

Review policy auswählen, strukturiert IAM möglicherweise Ihre Richtlinie neu, um sie für den visuellen Editor zu optimieren.

6. Geben Sie auf der Seite Überprüfen und erstellen einen Namen für das Feld Richtliniename ein. Überprüfen Sie die von ihrer Richtlinie erteilten Berechtigungen und wählen Sie dann zum Speichern Ihrer Arbeit Create policy (Richtlinie erstellen) aus.

Die neue Richtlinie wird in der Liste der verwalteten Richtlinien angezeigt und ist bereit.

Als Nächstes fügen Sie die Richtlinie, die Sie gerade erstellt haben, Ihrem IAM-Benutzer an.

So fügen Sie die Richtlinie an Ihren IAM-Benutzer an

1. Wählen Sie im Navigationsbereich der IAM-Konsole die Option Policies.
2. Beginnen Sie oben in der Richtlinienliste im Suchfeld mit der Eingabe des Namens Ihrer Richtlinie, bis Sie Ihre Richtlinie sehen. Wählen Sie dann das Optionsfeld neben Ihrer Richtlinie in der Liste aus.
3. Wählen Sie die Schaltfläche Actions (Aktionen) und dann Attach (Anfügen).
4. Wählen Sie bei den IAM-Entitäten die Option zum Filtern nach Users (Benutzer).
5. Beginnen Sie im Suchfeld mit der Eingabe von **PolicyUser**, bis dieser Benutzer in der Liste angezeigt wird. Aktivieren Sie dann in der Liste das Kontrollkästchen neben diesem Benutzer.
6. Wählen Sie Richtlinie anfügen aus.

Einrichten der Authentifizierung für Ihre Kartenanwendung

1. Melden Sie sich bei der [Amazon-Cognito-Konsole](#) an.

 **Important**

Der Pool, den Sie erstellen, muss sich in demselben AWS-Konto und AWS derselben Region befinden wie die Amazon Location Service-Ressourcen, die Sie im vorherigen Abschnitt erstellt haben.

2. Öffnen Sie die Seite Wichtigkeitspools.
3. Wählen Sie Identitäten-Pool erstellen

4. Beginnend mit dem Schritt Identitätspool-Vertrauen konfigurieren. Wählen Sie für Authentocation des Benutzerzugriffs Gastzugriff aus und drücken Sie Weiter.
5. Wählen Sie auf der Seite Berechtigungen konfigurieren die Option Vorhandene IAM-Rolle verwenden aus und geben Sie den Namen der IAM-Rolle ein, die Sie im vorherigen Schritt erstellt haben. Wenn Sie bereit sind, drücken Sie neben , um mit dem nächsten Schritt fortzufahren.
6. Geben Sie auf der Seite Eigenschaften konfigurieren einen Namen für Ihren Einrückungspool ein. Drücken Sie dann Weiter .
7. Überprüfen Sie auf der Seite Überprüfen und erstellen alle vorhandenen Informationen und drücken Sie dann Identitätspool erstellen.
8. Öffnen Sie die Seite Identitäten-Pools und wählen Sie den soeben erstellten Untrennbarkeitspool aus. Kopieren oder notieren Sie sich dann die `IdentityPoolId`, die Sie später in Ihrem Browserskript verwenden werden.

Erstellen der iOS-Basisanwendung

In diesem Tutorial erstellen Sie eine iOS-Anwendung, die eine Karte einbettet und es dem Benutzer ermöglicht, zu finden, was sich an einem Speicherort auf der Karte befindet.

Lassen Sie uns zunächst eine Swift-Anwendung mit dem Projektassistenten von Xcode erstellen.

So erstellen Sie eine leere Anwendung (Xcode)

1. Starten Sie Xcode. Wählen Sie im Menü Datei , Neu , Neues Projekt aus.
2. Wählen Sie auf der Registerkarte iOS die Option App und dann Weiter aus.
3. Wählen Sie einen Produktnamen und eine Organisations-ID und dann Weiter aus.
4. Wählen Sie den Speicherort aus, an dem Sie Ihr Projekt speichern möchten, und klicken Sie auf Erstellen.

Als Nächstes fügen Sie das Kartensteuerelement der Anwendung hinzu.

Hinzufügen einer interaktiven Zuordnung zu Ihrer Anwendung

In diesem Abschnitt fügen Sie die Kartensteuerung zu Ihrer Anwendung hinzu. In diesem Tutorial werden MapLibre und die AWS -API für die Verwaltung der Kartenansicht in der Anwendung verwendet. Die Kartensteuerung selbst ist Teil der [MapLibre microSD Native iOS](#)-Bibliothek.

Um die Zuordnung hinzuzufügen, führen Sie die folgenden Schritte aus:

1. Fügen Sie die MapLibre zu Ihrem Projekt hinzu.
2. Fügen Sie Ihrem Projekt das Amazon Location Authentication SDK iOS hinzu.
3. Fügen Sie Ihrem Projekt AWS SDK for iOS hinzu.
4. Fügen Sie AWS Ressourcenwerte hinzu.
5. Einrichtungscode

Die folgenden Verfahren führen Sie durch jeden dieser Schritte.

1 – Fügen Sie die MapLibre zu Ihrem Projekt hinzu

1. Öffnen Sie in Xcode das Dateimenü und wählen Sie die Option Add Packages Dependencies aus. Dadurch wird die Paketliste geöffnet, mit der Sie auf die Swift-Pakete zugreifen können.
2. Geben Sie im Fenster Pakete die URL des Maplibre Native-Pakets ein: <https://github.com/maplibre/maplibre-gl-native-distribution> und drücken Sie die Eingabetaste
3. Wählen Sie das `maplibre-gl-native-distribution` Paket aus und drücken Sie die Schaltfläche Paket hinzufügen.
4. Wählen Sie das Mapbox Produkt aus und drücken Sie die Schaltfläche Paket hinzufügen.

2 – Amazon Location Authentication SDK iOS zu Ihrem Projekt hinzufügen

1. Geben Sie im Fenster Pakete die iOS-URL des Amazon Location Authentication SDK ein: <https://github.com/aws-amplify/aws-sdk-ios-spm> und drücken Sie die Eingabetaste.
2. Wählen Sie das `aws-sdk-ios-spm` Paket aus und drücken Sie die Schaltfläche Paket hinzufügen.
3. Wählen Sie das `AWSMobileClientXCF` Produkt aus und drücken Sie die Schaltfläche Paket hinzufügen.

3 – Hinzufügen von AWS SDK für iOS zu Ihrem Projekt

1. Geben Sie im Fenster Pakete die iOS-URL des Amazon Location Authentication SDK ein: <https://github.com/aws-amplify/aws-sdk-ios-spm> und drücken Sie die Eingabetaste.
2. Wählen Sie die `amazon-location-mobile-auth-sdk-ios` und drücken Sie die Schaltfläche Paket hinzufügen.

3. Wählen Sie die aus AWSAuthSDKiOS und drücken Sie die Schaltfläche Paket hinzufügen.

4 – AWS Ressourcenwerte hinzufügen

1. Geben Sie im Fenster Pakete > die Paket-URL SDK für iOS ein AWS : <https://github.com/aws-amplify/aws-sdk-ios-spm> und drücken Sie die Eingabetaste.
2. Wählen Sie das aws-sdk-ios-spm Paket aus und klicken Sie auf die Schaltfläche Paket hinzufügen.
3. Wählen Sie das AWSMobileClientXCFProdukt aus und klicken Sie auf die Schaltfläche Paket hinzufügen.

5 – Einrichtungscode

1. Fügen Sie die -AWSSignatureV4Delegate.swiftDatei hinzu, die die folgende Klassendefinition enthält, um AWS Anfragen abzufangen und sie mit Signature Version 4 zu signieren. Eine Instance dieser Klasse wird als Offline-Speicherdelegierter zugewiesen, der auch für das Umschreiben von URLs in der Kartenansicht verantwortlich ist.

```
import AWSCore
import Mapbox

class AWSSignatureV4Delegate : NSObject, MGLOfflineStorageDelegate {
    private let region: AWSRegionType
    private let identityPoolId: String
    private let credentialsProvider: AWSCredentialsProvider

    init(region: AWSRegionType, identityPoolId: String) {
        self.region = region
        self.identityPoolId = identityPoolId
        self.credentialsProvider = AWSCognitoCredentialsProvider(regionType:
region, identityPoolId: identityPoolId)
        super.init()
    }

    class func doubleEncode(path: String) -> String? {
        return path.addingPercentEncoding(withAllowedCharacters: .urlPathAllowed)?
            .addingPercentEncoding(withAllowedCharacters: .urlPathAllowed)
    }
}
```

```

func offlineStorage(_ storage: MGLOfflineStorage, urlForResourceOf kind:
MGLResourceKind, with url: URL) -> URL {
    if url.host?.contains("amazonaws.com") != true {
        // not an AWS URL
        return url
    }

    // URL-encode spaces, etc.
    let keyPath = String(url.path.dropFirst())
    guard let percentEncodedKeyPath =
keyPath.addingPercentEncoding(withAllowedCharacters: .urlPathAllowed) else {
        print("Invalid characters in path '\(keyPath)'; unsafe to sign")
        return url
    }

    let endpoint = AWSEndpoint(region: region, serviceName: "geo", url: url)
    let requestHeaders: [String: String] = ["host": endpoint!.hostName]

    // sign the URL
    let task = AWSSignatureV4Signer
        .generateQueryStringForSignatureV4(
            withCredentialProvider: credentialsProvider,
            httpMethod: .GET,
            expireDuration: 60,
            endpoint: endpoint!,
            // workaround for https://github.com/aws-amplify/aws-sdk-ios/
issues/3215
            keyPath: AWSSignatureV4Delegate.doubleEncode(path:
percentEncodedKeyPath),
            requestHeaders: requestHeaders,
            requestParameters: .none,
            signBody: true)
    task.waitUntilFinished()

    if let error = task.error as NSError? {
        print("Error occurred: \(error)")
    }

    if let result = task.result {
        var urlComponents = URLComponents(url: (result as URL),
resolvingAgainstBaseURL: false)!
        // re-use the original path; workaround for https://github.com/aws-
amplify/aws-sdk-ios/issues/3215
        urlComponents.path = url.path
    }
}

```

```
        // have Mapbox GL fetch the signed URL
        return (urlComponents.url)!
    }

    // fall back to an unsigned URL
    return url
}
}
```

2. Öffnen Sie die `ViewController.swift` Datei und fügen Sie den folgenden Code hinzu:

```
import UIKit
import Mapbox

class ViewController: UIViewController, MGLMapViewDelegate {

    var identityPoolId = Bundle.main.infoDictionary?["identityPoolId"] as! String
    var regionName = Bundle.main.infoDictionary?["regionName"] as! String
    var mapName = Bundle.main.infoDictionary?["mapName"] as! String

    var authHelper: AuthHelper!
    var locationCredentialsProvider: LocationCredentialsProvider!
    var signingDelegate: MGLOfflineStorageDelegate!
    var mapView: MGLMapView!

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

    func loadMap() {
        let region = (regionName as NSString).aws_regionTypeValue()
        // MGLOfflineStorage doesn't take ownership, so this needs to be a member
        here
        signingDelegate = AWSSignatureV4Delegate(region: region, identityPoolId:
        identityPoolId)

        // register a delegate that will handle SigV4 signing
        MGLOfflineStorage.shared.delegate = signingDelegate

        let styleURL = URL(string: "https://maps.geo.\(regionName).amazonaws.com/
        maps/v0/maps/\(mapName)/style-descriptor")
    }
}
```

```
        mapView = MGLMapView(frame: view.bounds, styleURL: styleURL)
        mapView.autoresizingMask = [.flexibleWidth, .flexibleHeight]
        mapView.setCenter(CLLocationCoordinate2D(latitude: 49.246559, longitude:
-123.063554), zoomLevel: 10, animated: false)
        view.addSubview(mapView!)
        mapView.delegate = self
    }
}
```

3. Speichern Sie die `ViewController.swift`-Datei.

Sehen wir uns die Aktivitäten des neuen Codes an.

- `viewDidLoad`: Wird aufgerufen, nachdem die Ansicht geladen wurde.
- `loadmap`: Diese Funktion erstellt und lädt das MGL MapView. Das `signingDelegate` ist eine Instance von `AWSSignatureV4Delegate`, einer Klasse, die dem `MGLOfflineStorageDelegate` Protokoll entspricht. Es ist dafür verantwortlich, die Anforderungen an - AWS Services mit Signature Version 4 (SigV4) zu signieren. Der Delegierte wird mit der entsprechenden AWS Region und einer Identitätspool-ID initialisiert, die verwendet werden, um die erforderlichen Anmeldeinformationen für die Signatur zu erhalten. Dieser Delegierte wird dann der Delegierungseigenschaft von `MGL OfflineStorage.shared` zugewiesen, die die Download-Anforderungen für Kartenkacheln verarbeitet. Die `styleURL` soll den spezifischen Kartenstil wie ESRI oder HERE basierend auf dem Kartennamen laden, der in der Stil-URL angegeben ist. Die `mapView` ist auf die angegebenen Koordinaten zentriert

Sie können jetzt die Anwendung erstellen. Um es auszuführen, müssen Sie möglicherweise ein Gerät einrichten, um in Xcode zu emulieren, oder die App auf Ihrem eigenen Gerät verwenden. Verwenden Sie diese App, um zu sehen, wie sich die Kartensteuerung verhält. Sie können schwenken, indem Sie auf die Karte ziehen, und die Verdichtung verkleinern, um zu zoomen. Sie können selbst ändern, wie die Kartensteuerung funktioniert, um sie an die Anforderungen Ihrer Anwendung anzupassen.

Im nächsten Abschnitt fügen Sie eine Markierung auf der Karte hinzu und zeigen die Adresse des Standorts an, an dem sich die Markierung befindet, während Sie die Karte verschieben.

Hinzufügen einer Suche zu Ihrer Anwendung

Der letzte Schritt für Ihre Anwendung besteht darin, eine Suche auf der Karte hinzuzufügen. In diesem Fall fügen Sie eine umgekehrte Geocodierungssuche hinzu, bei der Sie die Elemente an einem Speicherort finden. Um die Verwendung einer iOS-App zu vereinfachen, durchsuchen wir

die Mitte des Bildschirms. Um einen neuen Speicherort zu finden, verschieben Sie die Zuordnung an die Stelle, an der Sie suchen möchten. Wir platzieren eine Markierung in der Mitte der Karte, um anzuzeigen, wo wir suchen.

Note

Amazon Location Service bietet auch die Möglichkeit, nach Namen oder Adresse zu suchen, um die Standorte auf der Karte zu finden.

So fügen Sie Ihrer App eine Suche an einem Speicherort hinzu

1. Wenn es noch nicht geöffnet ist, öffnen Sie die `ViewController.swift` Datei, wie im vorherigen Verfahren. Fügen Sie den folgenden Code hinzu, wobei die Änderungen, die Sie vornehmen müssen, grün dargestellt sind. Dadurch wird ein Label erstellt, auf dem wir die Ergebnisse unserer Suche am Standort sehen können.

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

    setupAddressLabel()
}

func setupAddressLabel() {
    addressLabel = UILabel(frame: CGRect(x: 0, y: view.bounds.height - 150, width:
view.bounds.width, height: 50))
    addressLabel.backgroundColor = UIColor.white.withAlphaComponent(0.9)
    addressLabel.textColor = UIColor.black
    addressLabel.textAlignment = .center
    view.addSubview(addressLabel)
}
```

2. Jetzt zeigen wir die Markierung in der Mitte der Karte mit dem folgenden Code:

```
import AWSAuthSDKiOS
import AWSLocationXCF

...
override func viewDidLoad() {
```

```
super.viewDidLoad()
loadMap()
setupAddressLabel()

authHelper = AuthHelper()
locationCredentialsProvider =
authHelper.authenticateWithCognitoUserPool(identityPoolId: identityPoolId)
addCenterMarker()
}

func addCenterMarker() {
    let centerCoordinate = mapView.centerCoordinate
    let marker = MGLPointAnnotation()
    marker.coordinate = centerCoordinate
    mapView.addAnnotation(marker)
    centerMarker = marker

    // Reverse geocode the center coordinate
    reverseGeocodeCenter()
}

func reverseGeocodeCenter() {
    let centerCoordinate = mapView.centerCoordinate
    let position = [NSNumber(value: centerCoordinate.longitude), NSNumber(value:
centerCoordinate.latitude)]
    searchPositionCognito(position: position)
}

func searchPositionCognito(position: [NSNumber]) {
    let request = AWSLocationSearchPlaceIndexForPositionRequest()!
    request.language = "en"
    request.position = position
    request.indexName = indexName

    let result = AWSLocation.default().searchPlaceIndex(forPosition: request)

    result.continueWith { response in
        if let results = response.result?.results {
            let address = results.first?.place?.label
            DispatchQueue.main.async {
                self.addressLabel.text = address
            }
        } else {
            let defaultError = NSError(domain: "Location", code: -1)

```

```

        let error = response.error ?? defaultError
        print("error \(error)")
    }
    return nil
}

func mapView(_ mapView: MGLMapView, regionDidChangeAnimated animated: Bool) {
    if let marker = centerMarker {
        marker.coordinate = mapView.centerCoordinate
    }
    reverseGeocodeCenter()
}

```

Dieser Code funktioniert mit der Kartenansicht. Die Kartenansicht in MapLibre wird durch eine virtuelle Kameraposition definiert. Das Verschieben der Karte kann sich als Verschieben dieser virtuellen Kamera vorstellen.

- `addCenterMarker()`: Dies ist eine Funktion zum Hinzufügen von Markierungen an den mittleren Koordinaten der Karte.
- `reverseGeocode`: Diese Funktion, die in der Event `mapView (_ mapView : MGL MapView, regionDidChangeAnimated: Bool)` aufgerufen wird, ruft das Zentrum der Karte für eine Position ab und ruft auf `searchPositionCognito`. Es verwendet das Kameraziel, das die Mitte der Karte definiert (wo die Kamera aussieht).
- `searchPositionCognito`: Diese Funktion sucht nach einem Speicherort und aktualisiert das Label, um die Ergebnisse anzuzeigen. Die Suchfunktion wird über das Amplify iOS SDK (`erreichtAWSLocationXCF`).

3. AThe letzte ViewController.swift-Datei sollte wie folgt aussehen:

```

import UIKit
import Mapbox
import AWSAuthSDKiOS
import AWSLocationXCF

class ViewController: UIViewController, MGLMapViewDelegate {

    var identityPoolId = Bundle.main.infoDictionary?["identityPoolId"] as! String
    var regionName = Bundle.main.infoDictionary?["regionName"] as! String
    var mapName = Bundle.main.infoDictionary?["mapName"] as! String
    var indexName = Bundle.main.infoDictionary?["searchPlaceIndex"] as! String

```

```
var authHelper: AuthHelper!
var locationCredentialsProvider: LocationCredentialsProvider!
var signingDelegate: MGLOfflineStorageDelegate!
var mapView: MGLMapView!
var addressLabel: UILabel!
var centerMarker: MGLPointAnnotation?

override func viewDidLoad() {
    super.viewDidLoad()
    loadMap()
    setupAddressLabel()
    authHelper = AuthHelper()
    locationCredentialsProvider =
authHelper.authenticateWithCognitoUserPool(identityPoolId: identityPoolId)
    addCenterMarker()
}

func loadMap() {
    let region = (regionName as NSString).aws_regionTypeValue()
    // MGLOfflineStorage doesn't take ownership, so this needs to be a member
here
    signingDelegate = AWSSignatureV4Delegate(region: region, identityPoolId:
identityPoolId)

    // register a delegate that will handle SigV4 signing
    MGLOfflineStorage.shared.delegate = signingDelegate

    let styleURL = URL(string: "https://maps.geo.\(regionName).amazonaws.com/
maps/v0/maps/\(mapName)/style-descriptor")

    mapView = MGLMapView(frame: view.bounds, styleURL: styleURL)
    mapView.autoresizingMask = [.flexibleWidth, .flexibleHeight]
    mapView.setCenter(CLLocationCoordinate2D(latitude: 49.246559, longitude:
-123.063554), zoomLevel: 10, animated: false)
    view.addSubview(mapView!)
    mapView.delegate = self
}

func setupAddressLabel() {
    addressLabel = UILabel(frame: CGRect(x: 0, y: view.bounds.height - 150,
width: view.bounds.width, height: 50))
    addressLabel.backgroundColor = UIColor.white.withAlphaComponent(0.9)
    addressLabel.textColor = UIColor.black
}
```

```
        addressLabel.textAlignment = .center
        view.addSubview(addressLabel)
    }

    func addCenterMarker() {
        let centerCoordinate = mapView.centerCoordinate
        let marker = MGLPointAnnotation()
        marker.coordinate = centerCoordinate
        mapView.addAnnotation(marker)
        centerMarker = marker

        // Reverse geocode the center coordinate
        reverseGeocodeCenter()
    }

    func reverseGeocodeCenter() {
        let centerCoordinate = mapView.centerCoordinate
        let position = [NSNumber(value: centerCoordinate.longitude),
NSNumber(value: centerCoordinate.latitude)]
        searchPositionCognito(position: position)
    }

    func searchPositionCognito(position: [NSNumber]) {
        let request = AWSLocationSearchPlaceIndexForPositionRequest()!
        request.language = "en"
        request.position = position
        request.indexName = indexName

        let result = AWSLocation.default().searchPlaceIndex(forPosition: request)

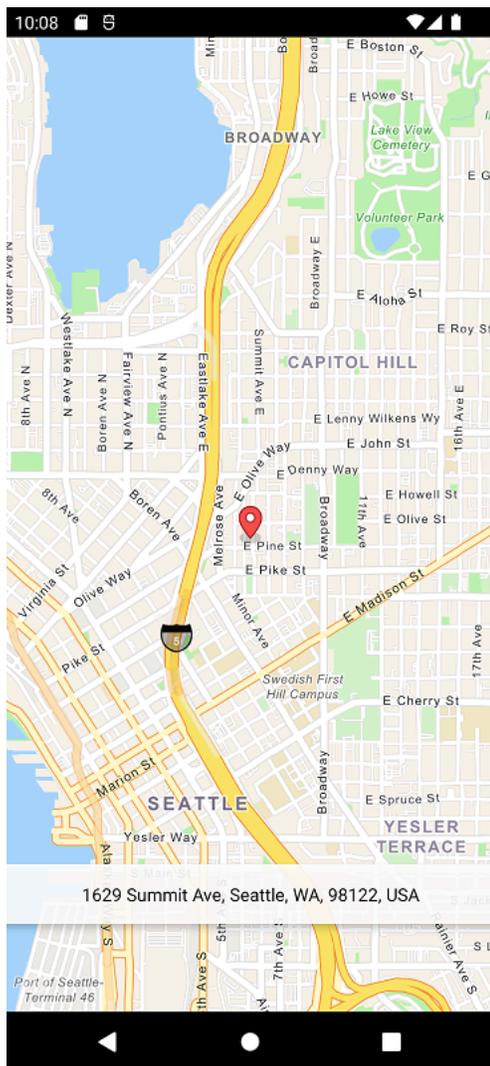
        result.continueWith { response in
            if let results = response.result?.results {
                let address = results.first?.place?.label
                DispatchQueue.main.async {
                    self.addressLabel.text = address
                }
            } else {
                let defaultError = NSError(domain: "Location", code: -1)
                let error = response.error ?? defaultError
                print("error \(error)")
            }
            return nil
        }
    }
}
```

```
func mapView(_ mapView: MGLMapView, regionDidChangeAnimated animated: Bool) {
    if let marker = centerMarker {
        marker.coordinate = mapView.centerCoordinate
    }
    reverseGeocodeCenter()
}
}
```

4. Speichern Sie Ihre Dateien und erstellen und führen Sie Ihre App (optional) aus, um eine Vorschau der Funktionalität anzuzeigen.

Ihre Schnellstartanwendung ist abgeschlossen. In diesem Tutorial erfahren Sie, wie Sie eine iOS-Anwendung erstellen, die:

- Erstellt eine Zuordnung, mit der Benutzer interagieren können.
- Verarbeitet mehrere Kartenereignisse, die dem Benutzer zugeordnet sind, der die Kartenansicht ändert (`cameraMoveStarted` und `CameraIdle`).
- Ruft eine Amazon Location Service-API auf, insbesondere um die Karte mithilfe der `-searchByCoordinatesOperation` von Amplify an einem Standort zu durchsuchen.



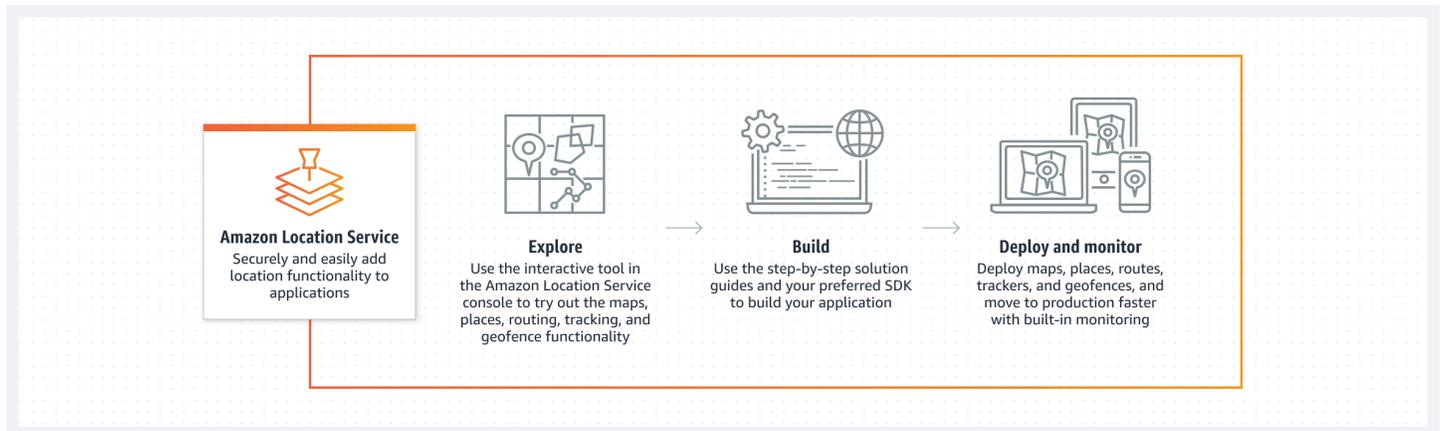
Was ist als Nächstes?

Sie haben das Schnellstart-Tutorial abgeschlossen und sollten sich ein Bild davon machen, wie Amazon Location Service zum Erstellen von iOS-Anwendungen verwendet wird. Um Amazon Location besser zu nutzen, können Sie sich die folgenden Ressourcen ansehen:

- Weitere Informationen zu den [Konzepten von Amazon Location Service](#)
- Weitere Informationen [zur Verwendung von Amazon Location-Funktionen und -Funktionen](#)
- Erfahren Sie, wie Sie dieses Beispiel erweitern und komplexere Anwendungen erstellen können, indem Sie sich [Codebeispiele mit Amazon Location](#) ansehen

Konzepte von Amazon Location Service

Mit Amazon Location Service können Sie Ihrer Anwendung sicher Standortdaten hinzufügen. Erkunden Sie einige der Funktionen mithilfe des [visuellen und interaktiven Tools](#), das in der Amazon Location-Konsole verfügbar ist. Mit dem Explorer-Tool können Sie eine Standardkarte manipulieren, nach Interessenpunkten suchen, Geofences nach Interessenbereichen ziehen und Sendegerätestandorte an einen Tracker simulieren.



Wenn Sie zum Erstellen bereit sind, erstellen Sie Ihre Ressourcen und wählen Sie aus einer Vielzahl von Kartenstilen und Datenanbietern. Anschließend können Sie das SDK installieren, das Ihrer Entwicklungsumgebung entspricht, und die Amazon Location APIs mithilfe der Anweisungen in diesem Handbuch verwenden. Darüber hinaus können Sie die Überwachung mithilfe von Amazon CloudWatch und integrieren AWS CloudTrail.

Die Themen in diesem Abschnitt bieten Ihnen einen Überblick über die Kernkonzepte von Amazon Location und bereiten Sie sich auf die Arbeit mit dem Standort in Ihren eigenen Anwendungen vor.

Themen

- [Übersicht über Amazon Location](#)
- [Zuordnungen](#)
- [Ortssuche](#)
- [Routen](#)
- [Geofences und Tracker](#)
- [Häufige Anwendungsfälle für die Verwendung von Amazon Location Service](#)
- [Was ist ein Datenanbieter?](#)

- [Regionen und Endpunkte von Amazon Location](#)
- [Amazon Location Service-Kontingente](#)

Übersicht über Amazon Location

Amazon Location Service bietet über AWS -Ressourcen Zugriff auf standortbasierte Funktionen und Datenanbieter. Amazon Location bietet fünf Arten von AWS Ressourcen, je nachdem, welche Art von Funktionalität Sie benötigen. Verwenden Sie die verschiedenen Ressourcen zusammen, um eine vollständige standortbasierte Anwendung zu erstellen. Sie können eine oder mehrere dieser Ressourcen mithilfe der Amazon Location-Konsole, der Amazon Location APIs oder der SDKs erstellen.

Jede Ressource definiert den zu verwendenden zugrunde liegenden [Datenanbieter](#) (falls zutreffend) und bietet Zugriff auf Funktionen, die sich auf seinen Typ beziehen.

Beispielsweise:

- Mit [Amazon Location Service Maps](#) können Sie eine Karte von einem Kartenanbieter auswählen, die Sie in Ihrer mobilen oder Webanwendung verwenden möchten.
- Mit [Amazon Location Service Places](#) können Sie einen Datenanbieter für die Suche nach Interessenpunkten, die Vervollständigung von Teilttext, Geokodierung und Reverse-Geokodierung auswählen.
- Mit [Amazon Location Service Routes](#) können Sie einen Datenanbieter auswählen und Routen finden und die Reisezeit auf der Grundlage von up-to-date Straßen- und Live-Datenverkehrsinformationen schätzen.
- Mit [Amazon Location Service Geofences](#) können Sie Interessenbereiche als virtuelle Grenze definieren. Anschließend können Sie Standorte anhand dieser bewerten und Benachrichtigungen über Ein- und Ausgangsereignisse erhalten.
- [Amazon Location Service Trackers](#) erhalten Standortaktualisierungen von Ihren Geräten. Sie können Tracker mit Geofence-Sammlungen verknüpfen, sodass alle Positionsaktualisierungen automatisch anhand Ihrer Geofences ausgewertet werden.

Sie können IAM-Richtlinien verwenden, um den Zugriff auf Ihre Amazon Location-Ressourcen zu verwalten und zu autorisieren. Sie können Ihre Ressourcen auch in Ressourcengruppen organisieren, um Aufgaben zu verwalten und zu automatisieren, wenn Ihre Ressourcennummern

zunehmen. Weitere Informationen zum Verwalten von AWS Ressourcen finden Sie unter [Was sind AWS-Ressourcengruppen?](#) Im AWS Resource Groups-Benutzerhandbuch.

Der Standort wird mithilfe von Breiten- und Längengradkoordinaten definiert, die dem [World Geodetic System \(WGS 84\)](#) folgen, das häufig als Standard-Koordinatenreferenzsystem für Global Positioning System (GPS)-Services verwendet wird.

In den folgenden Abschnitten wird beschrieben, wie die Komponenten von Amazon Location funktionieren.

Zuordnungen

Die Amazon Location Service Map-Ressource gibt Ihnen Zugriff auf die zugrunde liegenden Basiskartendaten für eine Zuordnung. Sie verwenden die Map-Ressource mit einer Map-Rendering-Bibliothek, um Ihrer Anwendung eine interaktive Map hinzuzufügen. Sie können Ihrer Karte je nach Bedarf weitere Funktionen hinzufügen, z. B. Markierungen (oder Pins), Routen und Polygonbereiche.

Note

Informationen zur Verwendung von Kartenressourcen in der Praxis finden Sie unter [Verwenden von Amazon Location Maps in Ihrer Anwendung](#).

Im Folgenden finden Sie eine Übersicht über das Erstellen und Verwenden von Kartenressourcen:

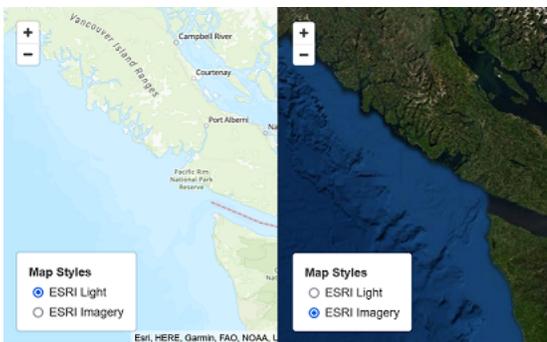


1. Sie erstellen eine Kartenressource in Ihrem AWS Konto, indem Sie einen Kartenstil von einem Datenanbieter auswählen.
2. Anschließend können Sie das SDK auswählen und installieren, das Ihrer Entwicklungsumgebung und Ihren Anwendungen entspricht. Weitere Informationen zu verfügbaren Optionen finden Sie im Thema Zugriff [auf Amazon Location](#).
3. Um eine Zuordnung in Ihrer Anwendung anzuzeigen, kombinieren Sie eine Zuordnungsressource mit einer Rendering-Bibliothek, z. B. Amplify MapLibre oder Tangram. Weitere Informationen finden Sie unter [Verwenden von Karten](#) in diesem Handbuch.

- Anschließend können Sie die Überwachung mithilfe von -Services wie Amazon CloudWatch und AWS CloudTrail mit Amazon Location integrieren. Weitere Informationen finden Sie unter [Überwachen von Amazon Location Service mit Amazon CloudWatch](#) und [Protokollierung und Überwachung mit AWS CloudTrail](#).

Kartenstile

Wenn Sie eine Kartenressource erstellen, müssen Sie einen Kartenstil für diese Ressource auswählen. Kartenstile definieren das Erscheinungsbild der gerenderten Karte. Die folgende Abbildung zeigt beispielsweise denselben Datenanbieter mit zwei verschiedenen Stilen aus verschiedenen Kartenressourcen in Amazon Location. Ein Stil ist ein typischer Straßenstil, der auf den Vektordaten in der Karte basiert. Die andere enthält Rasterdaten, die Satellitenbilder zeigen. Der Stil kann sich ändern, wenn Sie die Karte vergrößern oder verkleinern, aber in der Regel haben Stile ein konsistentes Design. Es ist möglich, Teile oder alle Stilinformationen zu überschreiben, bevor sie an die Map-Rendering-Bibliothek übergeben werden.



Politische Ansichten

Bestimmte Kartenstile in Amazon Location Service unterstützen zusätzliche politische Ansichten.

Note

Die Politische Ansicht muss in Übereinstimmung mit den geltenden Gesetzen verwendet werden, einschließlich der Gesetze zum Mapping des Landes oder der Region, in dem/der die Karten, Bilder und andere Daten sowie Inhalte von Drittanbietern, auf die Sie über Amazon Location Service zugreifen, verfügbar gemacht werden.

Die folgenden Kartenstile unterstützen eine IND-Politikansicht (India).

- [Esri-Kartenstile:](#)
 - Esri-Navigation
 - Esri Light
 - Esri Straßenkarte
 - Esri Dark Gray Canvas
 - Esri Light Gray Canvas
- [Open Data Map-Stile:](#)
 - Open Data Standard Light
 - Open-Data-Standard-Dunkel
 - Open Data Visualization Light
 - Open-Data-Visualisierung im Hintergrund

In der Amazon Location Service-Konsole können Sie die angezeigten Stile so filtern, dass nur die Stile angezeigt werden, die die indische Polnische Ansicht unterstützen.

Benutzerspezifische Layers

Eine benutzerdefinierte Ebene ist eine zusätzliche Ebene, die Sie für einen Kartenstil aktivieren können. Derzeit unterstützt nur der VectorEsriNavigation Kartenstil die POI benutzerdefinierte Ebene.

Wenn Sie die POI benutzerdefinierte Ebene aktivieren, fügt sie Ihrer Karte eine umfassendere Reihe von Orten hinzu, z. B. Arzt, Services, Trichter, Trichter und andere wichtige Punkte. Standardmäßig ist die benutzerdefinierte Ebene unset. Weitere Informationen finden Sie unter [MapConfiguration](#) in der Location API-Referenz.

Map-Rendering

Um eine Karte in Ihrer Anwendung zu rendern, verwenden Sie in der Regel eine Karten-Rendering-Bibliothek. Es gibt mehrere allgemeine Optionen für Bibliotheken:

- MapLibre – MapLibre ist eine Open-Source-Bibliothek, die speziell für das Rendern interaktiver Karten entwickelt wurde und die bevorzugte Methode zum Rendern von Karten aus Amazon Location Service ist. MapLibre umfasst die Möglichkeit, Raster- und Vektordaten aus einer Datenquelle (z. B. einer Amazon-Location-Kartenressource) zu rendern. Sie können erweitern MapLibre , um Ihre eigenen Daten auf der Karte zu zeichnen.

- **Amplify** – Amplify ist ein Open-Source-Framework zum Erstellen von Anwendungen für Web, iOS, Android und mehr. Wenn Ihre Anwendung Amplify verwendet, können Sie sie erweitern, um die Amazon-Location-Funktionalität einzubeziehen. Amplify enthält Bibliotheken, die speziell für die Erstellung von Amazon Location-basierten Anwendungen entwickelt wurden, einschließlich Rendering-Karten. Amplify verwendet , MapLibre um die Karte zu rendern, bietet jedoch zusätzliche Funktionen, die für Amazon Location Service spezifisch sind, um die Verwendung effizienter zu gestalten und Such- und andere Funktionen hinzuzufügen.
- **Tangram** – Tangram ist eine alternative Open-Source-Bibliothek, die interaktive Karten rendert, ähnlich wie MapLibre.

Die Karten-Rendering-Bibliothek ruft zur Laufzeit Daten aus Amazon Location Service ab und rendert die Kartendaten basierend auf der von Ihnen ausgewählten Kartenressource. Die Kartenressource definiert den Datenanbieter und den Kartenstil, der verwendet werden soll.

Die folgende Abbildung zeigt, wie die Kartenressource in Amazon Location Service zusammen mit einer Karten-Rendering-Bibliothek verwendet wird, um die endgültige Karte zu erstellen.



1. Sie erstellen eine Kartenressource in Amazon Location Service mithilfe der AWS Management Console oder AWS CLI. Dies definiert den Datenanbieter und den Kartenstil, den Sie verwenden möchten.
2. Ihre Anwendung enthält eine Map-Rendering-Bibliothek. Sie geben der Map-Rendering-Bibliothek den Namen der zu verwendenden Map-Ressource. Die Karten-Rendering-Bibliothek ruft Daten und Stilinformationen für diese Kartenressource von Amazon Location ab und rendert die Karte auf dem Bildschirm.

Zuordnungs-Terminologie

Zuordnungsressource

Ermöglicht den Zugriff auf Kartendaten von einem ausgewählten Anbieter. Verwenden Sie die Kartenressource, um Kartenkacheln abzurufen, die Kartendaten enthalten, und einen Stildeskriptor, um anzugeben, wie Features auf einer Karte gerendert werden.

Basiskarte

Bietet geografischen Kontext für Ihre Karte, die als Vektorkachelschichten gespeichert wird. Zu den Kachelebenen gehören geografischer Kontext wie Straßennamen, Gebiete und Landnutzung zur visuellen Referenz.

Vektor

Vektordaten sind Formdaten, die aus Punkten, Linien und Polygonen bestehen. Es wird häufig verwendet, um Straßen, Standorte und Gebiete auf einer Karte zu speichern und anzuzeigen. Eine Vektorform kann auch als Symbole für Markierungen auf einer Karte verwendet werden.

Raster

Rasterdaten sind Bilddaten, die aus einem Raster bestehen, normalerweise aus Farben. Es wird häufig verwendet, um eine Darstellung kontinuierlicher Daten auf Karten zu speichern und anzuzeigen, z. B. Trichter, Satellitenbilder oder Heatmaps. Rasterbilder können auch als Bilder oder Symbole verwendet werden.

Kartenstil

Vektordaten enthalten von Natur aus keine Informationen darüber, wie die Datenebenen zur Erstellung der endgültigen Karte zeichnen werden. Ein Kartenstil definiert Farb- und andere Stilinformationen für die Daten, um zu definieren, wie sie bei der Rendering aussehen werden. Kartenressourcen enthalten Stilinformationen für die Karte.

Amazon Location Service bietet Stile, die der [Mapbox-Spezifikation im GL-Stil](#) entsprechen.

Vektorkachel

Ein Kachelformat, das Kartendaten mithilfe von Vektorformen speichert. Diese Daten führen zu einer Zuordnung, die sich an die Anzeigeauflösung anpassen und Features selektiv auf verschiedene Arten rendern kann, während eine kleine Dateigröße beibehalten wird, um eine optimale Leistung zu erzielen.

Unterstütztes Vektordateiformat: Mapbox Vector Tiles (MVT).

Glyph-Datei

Eine Binärdatei mit codierten Unicode-Zeichen. Wird von einem Karten-Renderer verwendet, um Labels anzuzeigen.

Sprite-Datei

Eine PNG-Bilddatei (Portable Network Graphic), die kleine Rasterbilder mit Standortbeschreibungen in einer JSON-Datei enthält. Wird von einem Karten-Renderer verwendet, um Symbole oder Streuungen auf einer Karte zu rendern.

Ortssuche

Eine wichtige Funktion von Amazon Location Service ist die Möglichkeit, die Geolocation-Informationen zu durchsuchen. Amazon Location stellt diese Funktionalität über die Ortsindex-Ressource bereit.

Note

Informationen zur Verwendung von Ortsindexressourcen für die Suche in der Praxis finden Sie unter [Suchen von Orts- und Geolocation-Daten mit Amazon Location](#).

Sie können die Ortsindex-APIs verwenden, um nach Folgendem zu suchen:

- Interessenpunkte, z. B. Telefonie und Kennzeichen. Suchen Sie nach Namen und optionalem Speicherort für die Suche und erhalten Sie eine Liste der Optionen, die nach Relevanz sortiert sind.
- Eine Straßenadresse, die einen Breiten- und Längengrad für diese Adresse erhält. Dies wird als Geokodierung bezeichnet.

- Eine Breiten- und Längengradposition, wobei die zugehörige Straßenadresse oder andere Informationen über den Standort empfangen werden. Dies wird als Reverse Geocoding bezeichnet.
- Eine teilweise oder falsch geschriebene Freiform-Textabfrage, in der Regel als Benutzertypen. Dies wird als automatische Vervollständigung, automatisches Zufalls- oder Fuzzy-Abgleich bezeichnet.

Der Ortsindex enthält den Datenanbieter, der für die Suche verwendet werden soll.

Note

Kartendaten und andere Geolokalisierungsinformationen, einschließlich exakter Standorte, können je nach Datenanbieter variieren. Verwenden Sie als bewährte Methode denselben Datenanbieter für Ihren Ortsindex, Ihre Karte und andere Amazon Location-Ressourcen. Wenn beispielsweise die von Ihrem Ortsindex zurückgegebenen Stellen nicht mit der Position derselben von Ihrer Kartenressource bereitgestellten Stellen übereinstimmen, können Sie eine Markierung an dem Ort platzieren, an dem es sich anscheinend um den falschen Ort auf der Karte zu handeln scheint.

Im Folgenden wird gezeigt, wie Sie Ortsindexressourcen erstellen und verwenden:



1. Zunächst erstellen Sie eine Ortsindexressource in Ihrem AWS Konto, indem Sie einen Datenanbieter auswählen.
2. Anschließend können Sie das SDK auswählen und installieren, das Ihrer Entwicklungsumgebung und Ihren Anwendungen entspricht. Weitere Informationen zu verfügbaren Optionen finden Sie im Thema Zugriff [auf Amazon Location](#).
3. Beginnen Sie mit der Verwendung der Amazon Location Places-APIs. Weitere Informationen finden Sie im Thema zur Verwendung der [Ortssuche](#).
4. Anschließend können Sie die Überwachung mithilfe von -Services wie Amazon CloudWatch und integrieren AWS CloudTrail. Weitere Informationen finden Sie unter [the section called "Überwachung mit CloudWatch"](#) und [the section called "Verwenden von CloudTrail mit Amazon Location"](#).

Geokodierungskonzepte

Ein Amazon Location Place Index bietet eine Aktion namens [SearchPlaceIndexForText](#), mit der Sie Text für die Suche angeben können. Sie können beispielsweise nach Folgendem suchen:

- Orte – eine Suche nach **Paris** könnte den Standort der Stadt in Frankreich zurückgeben.
- Unternehmen – Eine Suche nach **coffee shop** könnte eine Liste von Fußbällen zurückgeben, einschließlich ihrer Namen und Standorte. Sie können auch einen Speicherort angeben, an dem gesucht werden soll, oder einen Begrenzungsrahmen, nach dem gesucht werden soll, um die Ergebnisse relevanter zu gestalten. In diesem Fall würde die Angabe eines Standorts in microSD Phoenix, Bol, in diesem Bereich Kern zurückgeben.
- Adressen – eine Suche nach **1600 Pennsylvania Ave, Washington D.C.** könnte den Standort des Whitehouse in den USA (der sich an dieser Adresse befindet) zurückgeben.

Die Suche nach Text auf diese Weise wird im Allgemeinen als Geokodierung bezeichnet, bei der ein geografischer Standort für die Adresse oder den Ort gefunden wird.

Amazon Location Service bietet auch eine umgekehrte Geocodierungsaktion namens [SearchPlaceIndexForPosition](#). Dies nimmt einen geografischen Standort und gibt die Adresse, das Unternehmen oder andere Informationen darüber zurück, was sich an diesem Standort befindet.

Suchergebnisse

Wenn Sie eine erfolgreiche Suchanfrage in Amazon Location Service stellen, werden ein oder mehrere Ergebnisse zurückgegeben. Jedes Ergebnis enthält eine Bezeichnung, bei der es sich um den Namen oder die Beschreibung des Ergebnisses handelt. Eine Suche nach **coffee shop** könnte beispielsweise ein Ergebnis mit der Bezeichnung zurückgeben, das Ihnen mitteilt **Home town Cafe**, dass ein Lebensmittelgeschäft namens „Home Bol“ gefunden wurde. Das Suchergebnis enthält in der Regel auch eine strukturierte Adresse (mit Eigenschaften wie Adressnummer, Einheit, Straße und Postleitzahl). Je nach Datenanbieter werden weitere Metadaten enthalten, z. B. das Land und die Zeitzone.

Bei einer Suche nach einem Geschäftsnamen oder einer Kategorie (z. B. **coffee shop**) möchten Sie möglicherweise alle zurückgegebenen Ergebnisse auf einer Karte anzeigen. Für eine Adresssuche möchten Sie möglicherweise nur das erste Ergebnis automatisch verwenden. Weitere Informationen zur Relevanz finden Sie im nächsten Thema.

Mehrere Ergebnisse und Relevanz

Bei der Textsuche findet Amazon Location Service oft mehr als ein einziges Ergebnis. Eine Suche nach **Paris** kann beispielsweise die Stadt in Frankreich, aber auch die Stadt in Phoenix zurückgeben. Die Ergebnisse werden nach Relevanz sortiert, wie vom Datenanbieter bestimmt.

Note

Die Ergebnisse werden in relevanter Reihenfolge von allen Anbietern zurückgegeben. Wenn Sie Esri oder Grab als Datenanbieter wählen, enthalten die Ergebnisse einen Relevanzwert, anhand dessen Sie die relative Relevanz zwischen den Ergebnissen einer einzelnen Anforderung verstehen können.

Wenn Sie zusätzliche Informationen angeben, z. B. einen Ländernamen oder einen Ort, an dem gesucht werden soll, kann sich die Reihenfolge der Ergebnisse ändern, die Anzahl der Ergebnisse reduzieren oder sogar die Menge der zurückgegebenen Ergebnisse ändern. Beispielsweise gibt eine Suche nach **Paris** mit einem Ort in Phoenix, an dem gesucht werden soll, eher **Paris, Texas** als erstes Ergebnis zurückParis, France.

In einer interaktiven Anwendung können Sie anhand der Relevanz entscheiden, ob das oberste Ergebnis akzeptiert werden soll, oder ob Sie einen Benutzer bitten möchten, zwischen mehreren zurückgegebenen Ergebnissen zu disambiguieren. Wenn das erste Ergebnis eine hohe Relevanz hat, akzeptieren Sie es möglicherweise einfach als richtige Antwort. Wenn es mehrere Ergebnisse mit hoher Relevanz oder keine Ergebnisse mit hoher Relevanz gibt, sollten Sie die Ergebnisse auflisten und den Benutzer das beste Ergebnis auswählen lassen.

Ergebnisse adressieren

Sie können mit Amazon Location Service mit derselben [SearchPlaceIndexForText](#) Aktion nach Adressen suchen. Je mehr Informationen Sie angeben, desto wahrscheinlicher ist es, dass die zurückgegebene Adresse mit der angegebenen übereinstimmt. Beispielsweise **123 Main St** ist es weniger wahrscheinlich, dass ein korrektes Ergebnis findet als **123 Main St, Anytown, California, 90210**.

Adressen haben mehrere Attribute, z. B. Straßenummer, Straßen, Stadt, Region und Postleitzahl usw. Diese Attribute werden verwendet, um eine Adresse im Ortsindex zu finden, die so vielen Aspekten wie möglich entspricht. Je mehr Attribute gefunden werden, desto relevanter wird die Übereinstimmung berücksichtigt und desto wahrscheinlicher wird sie zurückgegeben.

Note

Die Relevanz der Adressergebnisse hängt davon ab, wie genau das Ergebnis mit der Eingabe übereinstimmt. Dies könnte die Anzahl der Attribute sein, die übereinstimmen, aber auch, wie genau die Ergebnisse mit der Eingabe übereinstimmen. Beispielsweise **123 Main St** hätte eine Eingabe von einer höheren Relevanz, wenn in den Daten gefunden **Main St** wird, als wenn das einzige Ergebnis **Maine St** ist. **Maine St** wird weiterhin zurückgegeben, aber wahrscheinlich mit einem niedrigeren Relevanzwert.

Die Suchergebnisse enthalten eine Bezeichnung für die vollständige Adresse (123 Main St, Anytown, California, 90210), aber auch die einzelnen strukturierten Attribute der zurückgegebenen Adresse. Dies ist hilfreich, da Sie dies beispielsweise verwenden können, um Adressfelder in einer Datenbank auszufüllen oder die Ergebnisse zu überprüfen und die Stadt, Region oder Postleitzahl des gefundenen Standorts zu finden.

Interpolation

Adressen in den Ortsindexdaten enthalten exakte Adressübereinstimmungen. Angenommen, es gibt eine Straße **9th street**- und ein Block hat 2 Häuser, 220 und 240, wie in der folgenden Abbildung dargestellt.



Der Datenanbieter erstellt die Geolocation-Daten mit diesen beiden bekannten Adressen. Sie können nach diesen beiden Adressen suchen und sie werden gefunden. Nachdem der Datenanbieter die Kartendaten erstellt hat, nehmen wir an, dass zwischen den ersten beiden Adressen ein neues Haus hinzugefügt wird. Dieses neue Haus erhält die Adresse 230. Wenn Sie nach **230 S 9th St** suchen, findet der Datenanbieter immer noch ein Ergebnis. Anstatt eine bekannte Adresse zu verwenden, interpoliert sie zwischen den bereits bekannten Adressen und schätzt die Position der neuen Adresse anhand dieser. In diesem Fall könnte es davon ausgehen, dass 230 zwischen 220 und 240 (und auf derselben Seite der Straßen) liegt und basierend darauf einen ungefähren Standort zurückgibt.

Note

Datenanbieter aktualisieren ihre Geolocation-Daten regelmäßig mit neuen Adressen. In diesem Fall **230 S 9th St** würde zu den Datenanbieterdaten hinzugefügt, aber es gibt in der Regel einen Zeitraum, in dem eine neue Adresse erstellt, aber noch nicht zu den Daten hinzugefügt wurde.

In diesem Fall kann der Datenanbieter nicht sagen, ob die neue Adresse in der Welt existiert, da sie sich noch nicht in den Daten befindet, sondern die beste Antwort anhand der enthaltenen Informationen liefert. Dieses Ergebnis wird als interpoliert bezeichnet und kann vom Datenanbieter in den Ergebnissen zurückgegeben werden. Wenn `interpolated` zurückgibt `false`, ist es eine bekannte Adresse. Wenn es zurückgibt `true`, handelt es sich um eine ungefähre Adresse. Wenn es nicht zurückgegeben wird, hat der Datenanbieter die Informationen darüber, ob das Ergebnis aus der Interpolation stammt, nicht bereitgestellt.

Important

Der Datenanbieter kann auch interpolierte Ergebnisse für Adressen zurückgeben, die überhaupt nicht vorhanden sind. Wenn Sie beispielsweise eingegeben haben **232 S 9th St**, findet der Anbieter diese nicht vorhandene Adresse und gibt einen Speicherort nahe 230 zurück, aber auf der 240-Seite. Interpolierte Adressen sind nützlich, um Sie an den richtigen Ort zu bringen. Beachten Sie jedoch, dass es sich nicht um bekannte Adressen handelt.

Speichern von Geocode-Ergebnissen

Wenn Sie eine Ortsindexressource erstellen, müssen Sie eine Datenspeicheroption angeben (`IntendedUse` in der API als bezeichnet). Dies kann entweder auf `Single Use` oder auf `gespeicherte`

Ergebnisse eingestellt werden. Dies fragt nach Ihrer beabsichtigten Verwendung der Ergebnisse. Wenn Sie die Ergebnisse speichern möchten (auch für Caching-Zwecke), müssen Sie die Speicheroption auswählen, nicht die Option für die einmalige Verwendung.

Note

Wenn Sie die gespeicherte Option ausgewählt haben (als Ja gekennzeichnet, Ergebnisse werden in der Konsole gespeichert oder `storage` in der `CreatePlaceIndex` API ausgewählt), speichert Amazon Location Service die Ergebnisse nicht für Sie. Dies ist ein Hinweis darauf, dass Sie die Ergebnisse speichern möchten.

Wenn Sie sich ansehen, wie Sie die Ergebnisse Ihrer Abfragen an Amazon Location Service verwenden werden, sollten Sie immer die geltenden [AWS Servicebedingungen](#) beachten.

Ortsterminologie

Ortsindex-Ressource

Ermöglicht Ihnen die Auswahl einer Datenquelle zur Unterstützung von Suchabfragen. Sie können beispielsweise nach Interessenpunkten, Adressen oder Koordinaten suchen. Wenn eine Suchabfrage an eine Ortsindexressource gesendet wird, wird sie mit der konfigurierten Datenquelle der Ressource erfüllt.

Geokodierung

Geokodierung ist der Prozess, bei dem eine Texteingabe entgegengenommen, im Ortsindex nach ihr gesucht und Ergebnisse mit Position zurückgegeben werden.

Geokodierung umkehren

Reverse Geocoding ist der Prozess, bei dem eine Position angenommen und Informationen über diese Position aus dem Ortsindex zurückgegeben werden, z. B. die Adresse, die Stadt oder das Unternehmen an diesem Standort.

Relevanz

Relevanz gibt an, wie genau ein Ergebnis mit der Eingabe übereinstimmt. Es ist kein Maß für die Richtigkeit.

Interpolation

Interpolation ist der Prozess, bei dem unbekannte Adressen anhand bekannter Adresstandorte als Leitpunkte gefunden werden.

ISO 3166-Ländercodes

Amazon Location Service Places verwendet [die Ländercodes der International Organization for Standardization \(ISO\) 3166](#), um auf Länder oder Regionen zu verweisen.

Um den Code für ein bestimmtes Land oder eine bestimmte Region zu finden, verwenden Sie die [ISO Online Browsing Platform](#).

Routen

Dieser Abschnitt bietet einen Überblick über die Konzepte zum Routing mit Amazon Location Service.

Note

Informationen zur Verwendung von Routenressourcen in der Praxis finden Sie unter [Berechnen von Routen mit Amazon Location Service](#).

Routenrechner-Ressourcen

Mit Routenrechnerressourcen können Sie Routen finden und die Reisezeit basierend auf den Informationen zum up-to-date Straßennetz und Live-Verkehr von Ihrem ausgewählten Datenanbieter schätzen.

Sie können die Routes-APIs verwenden, um Funktionen zu erstellen, mit denen Ihre Anwendung die Zeit, Entfernung und Geometrie der Route zwischen zwei beliebigen Standorten anfordern kann. Sie können die Routes-API auch verwenden, um die Reisezeit und Entfernung zwischen einer Reihe von Abgängen und Zielen in einer einzigen Anforderung anzufordern, um eine Matrix zu berechnen.

Im Folgenden erfahren Sie, wie Sie eine Routenrechner-Ressource erstellen und verwenden:



1. Zunächst erstellen Sie eine Routenrechner-Ressource in Ihrem AWS Konto, indem Sie einen Datenanbieter auswählen.
2. Anschließend können Sie das SDK auswählen und installieren, das Ihrer Entwicklungsumgebung und Ihren Anwendungen entspricht.
3. Beginnen Sie mit der Verwendung der Amazon Location Routes APIs . Weitere Informationen zur Verwendung der Routing-APIs finden Sie im Thema auf [Berechnen von Routen mit Amazon Location Service](#).
4. Anschließend können Sie die Überwachung mithilfe von -Services wie Amazon CloudWatch und integrieren AWS CloudTrail. Weitere Informationen finden Sie unter [Überwachen von Amazon Location Service mit Amazon CloudWatch](#) und [Protokollierung und Überwachung mit AWS CloudTrail](#).

Berechnen einer Route

Eine Routenrechner-Ressource von Amazon Location bietet eine Aktion namens `CalculateRoute`, mit der Sie eine Route zwischen zwei geografischen Standorten (der Ankunft und dem Ziel-) erstellen können. Die berechnete Route enthält die Geometrie zum Zeichnen der Route auf einer Karte sowie die Gesamtzeit und Entfernung der Route.

Verwenden von Wegpunkten

Wenn Sie Ihre Routenanforderung erstellen, können Sie der Route zusätzliche Wegpunkte hinzufügen. Dies sind Punkte zwischen dem Verlassen und dem Ziel, die als Stopps auf der Route fungieren. Die Route wird über jeden der angegebenen Wegpunkte berechnet. Die Route von einem Punkt in der Anforderung zum nächsten wird als `bezeichnetLeg` bezeichnet. Jedes Ende enthält eine Entfernung, eine Zeit und die Geometrie für diesen Teil der Route.

Note

Die Wegpunkte werden in der Reihenfolge weitergeleitet, die in der Anforderung angegeben ist. Sie werden nicht für den kürzesten Pfad neu angeordnet. Informationen zum Auffinden des kürzesten Pfads finden Sie im [Planen von Routen](#) Abschnitt .

Sie können bis zu 25 Wegpunkte in eine einzelne Anforderung aufnehmen, um eine Route zu berechnen.

Datenverkehr und Abgangszeit

Der Amazon Location Service berücksichtigt den Datenverkehr bei der Berechnung einer Route. Der Datenverkehr, den es berücksichtigt, basiert auf der von Ihnen angegebenen Zeit. Sie können angeben, dass Sie jetzt abfahren möchten, oder Sie können einen bestimmten Zeitpunkt angeben, den Sie verlassen möchten, was sich auf das Routenergebnis auswirkt, indem Sie den Datenverkehr zum angegebenen Zeitpunkt anpassen.

Note

Sie können die Ankunftszeit anhand der Ankunftszeit und der Antwortzeit der Route berechnen, um beispielsweise die Ankunft eines Treibers abzuschätzen.

Wenn Sie möchten, dass Amazon Location keinen Datenverkehr berücksichtigt, geben Sie keine Abgangszeit an und geben Sie jetzt keinen Abgang an. Dadurch wird eine Route berechnet, die die besten Datenverkehrsbedingungen für die Route annimmt.

Optionen für den Reisemodus

Sie können den Reisemodus festlegen, wenn Sie eine Route mit Amazon Location Service berechnen. Der Standardreisemodus ist Auto , aber Sie können abwechselnd entweder Lkw oder Walking auswählen.

Wenn Sie entweder den Auto- oder den Lkw-Modus angeben, können Sie auch zusätzliche Optionen angeben.

Für den Automodus können Sie angeben, dass Sie gebührenpflichtige Straßen oder Fahrten vermeiden möchten. Dadurch wird versucht, Spuren und gebührenfreie Straßen zu vermeiden, aber

sie werden immer noch entlang weitergeleitet, wenn dies die einzige Möglichkeit ist, zum Ziel zu gelangen.

Im Lkw-Modus können Sie auch Spuren und Mautwege vermeiden, aber zusätzlich können Sie die Größe und das Gewicht des Lkw angeben, um Routen zu vermeiden, die den Lkw nicht aufnehmen.

Planen von Routen

Sie können Amazon Location Service verwenden, um Eingaben für Ihre Routenplanungs- und Optimierungssoftware zu erstellen. Sie können Routenergebnisse erstellen, einschließlich Reisezeit und Entfernung, für Routen zwischen einer Reihe von Abstiegspositionen und einer Reihe von Zielpositionen. Dies wird als Erstellen einer Routenmatrix bezeichnet.

Note

Es gibt viele unterschiedliche Szenarien, die Routenplanungs- und Optimierungssoftware lösen kann. Die Planungssoftware kann beispielsweise die Häufigkeiten und Entfernungen zwischen Punkten verwenden, um den kürzesten Pfad zu berechnen, der an jedem Punkt stoppt, und bietet eine effiziente Route für einen einzelnen Treiber. Alternativ kann die Planungssoftware verwendet werden, um Stopps auf mehrere Lkw aufzuteilen, die Effizienz einer Flotte zu gewährleisten oder sicherzustellen, dass jeder Kunde innerhalb des von ihm benötigten Zeitrahmens besucht wird. Amazon Location stellt die Routing-Funktionen auf effiziente Weise bereit, damit die Planungssoftware ihre Aufgabe erledigen kann.

Angenommen, bei den Ausgangspositionen A und B sowie den Zielpositionen X und Y gibt Amazon Location Service die Zeit und Entfernung für Routen von A zu X, A zu Y, B zu X und B zu Y zurück.

Wie bei der Berechnung einer einzelnen Route können Sie die Routen mit unterschiedlichen Transportarten, Abwägungen und Verkehrsbedingungen berechnen. Sie können beispielsweise angeben, dass das Fahrzeug ein Lkw mit einer Länge von 35 Fuß ist, und die berechnete Route verwendet diese Einschränkungen, um die Fahrtzeit und die Entfernung zu bestimmen. Sie können keine Wegpunkte in eine Routenmatrixberechnung aufnehmen.

Die Anzahl der zurückgegebenen Ergebnisse (und berechneten Routen) ist die Anzahl der Abstiegspositionen multipliziert mit der Anzahl der Zielpositionen. Ihnen wird jede berechnete Route in Rechnung gestellt, nicht jede Anforderung an den Service. Daher wird eine Routenmatrix mit 10 Abgängen und 10 Zielen als 100 Routen abgerechnet.

Routing-Terminologie

Routenrechner-Ressource

Eine - AWS Ressource, mit der Sie die Reisezeit, Entfernung und die Plot-Routen auf einer Karte mit Verkehrs- und Straßennetzdaten schätzen können, die von Ihrem ausgewählten Datenanbieter stammen.

Mithilfe von Routenrechnerressourcen berechnen Sie Routen für verschiedene Transportarten, Entwürfen und Verkehrsbedingungen.

Route

Eine Route enthält Details, die verwendet werden, wenn ein Pfad von der Ausgangsposition, den Wegpunktpositionen und der Zielposition aus entlanggefahren wird.

Beispiele für Details in einer Route sind:

- Der Abstand von einer Position zu einer anderen Position.
- Die Zeit, die benötigt wird, um von einer Position zur nächsten zu gelangen.
- Die LineString Geometrie, die den Pfad der Route darstellt.

Weitere Informationen zu Routen finden Sie in der [Antwortsyntax für die Operation CalculateRoute](#) in der API-Referenz zu Amazon Location Service Routes .

Routenmatrix

Eine Liste von Routen, von einer Reihe von Verlassenspositionen bis hin zu einer Reihe von Zielpositionen. Nützlich als Eingaben für Routenplanungs- oder Optimierungssoftware.

Weitere Informationen zum Berechnen einer Routenmatrix finden Sie in der [Syntax für die Operation CalculateRouteMatrix](#) in der Amazon Location Service Routes API-Referenz .

LineString Geometrie

Eine Amazon-Standort-Route besteht aus einem oder mehreren Abschnitten (einer Route von einem Wegpunkt zu einem anderen innerhalb der gesamten Route). Die Geometrie jedes Blatts ist eine Polylinie, die als dargestellt wird LineString. Ein LineString ist ein geordnetes Array von Positionen, mit denen eine Route auf einer Karte dargestellt werden kann.

Im Folgenden finden Sie ein Beispiel für eine LineString mit drei Punkten:

```
[  
  [-122.7565, 49.0021],  
  [-122.3394, 47.6159],  
  [-122.1082, 45.8371]  
]
```

Zwischenposition

Wegpunkte sind Zwischenpositionen, die als Stopps entlang einer Route zwischen der Ausgangsposition und der Zielposition fungieren. Die Stopover-Reihenfolge auf der Route folgt der Reihenfolge, in der Sie die Wegpunktpositionen in der Anforderung angeben.

Zweig

Ein einzelner Fuß ist der Weg von einer Position zu einer anderen. Wenn sich die Positionen nicht auf einer Straßen befinden, werden sie auf die nächste Straßen verschoben. Die Anzahl der Abschnitte in einer Route ist um eins geringer als die Gesamtzahl der Positionen.

Eine Route ohne Wegpunkte besteht aus einem einzelnen Abschnitt, von der Ausgangsposition bis zum Ziel. Eine Route mit 1 Wegpunkt besteht aus 2 Zielen, von der Ausgangsposition bis zum Wegpunkt und dann vom Wegpunkt bis zum Ziel.

Schritt

Ein Schritt ist ein Unterabschnitt eines Fußes. Jeder Schritt enthält zusammenfassende Informationen für diesen Schritt im Schritt.

Geofences und Tracker

Dieser Abschnitt bietet einen Überblick über die Konzepte der Arbeit mit Amazon Location Service Geofences und Trackern. Geofences sind Polygongrenzen, mit denen Sie benachrichtigt werden können, wenn sich Geräte oder Positionen in die Bereiche und aus ihnen bewegen. Tracker-Ressourcen werden verwendet, um Positionen für Geräte zu speichern und zu aktualisieren, während sie sich bewegen.

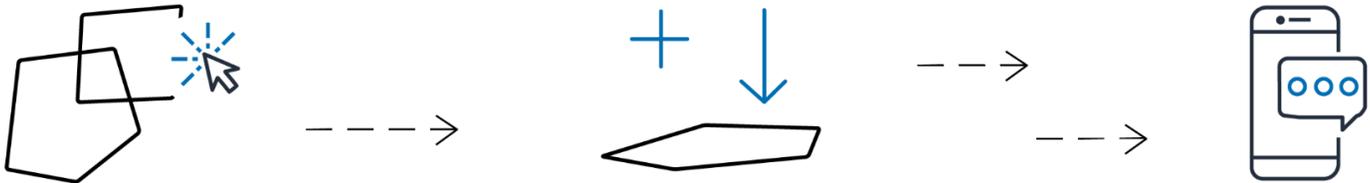
Note

Informationen zur Verwendung von Geofences und Trackern in der Praxis finden Sie unter [Geofencing eines Interessengebiets mit Amazon Location](#).

Geofences

Mit Geofence-Sammlungsressourcen können Sie Geofences – virtuelle Grenzen auf einer Karte – speichern und verwalten. Sie können Standorte anhand einer Geofence-Sammlungsressource auswerten und Benachrichtigungen erhalten, wenn die Standortaktualisierung die Grenze eines der Geofences in der Geofence-Sammlung überschreitet.

Im Folgenden erfahren Sie, wie Sie Geofence-Sammlungsressourcen erstellen und verwenden:



1. Erstellen Sie eine Geofence-Sammlungsressource in Ihrem AWS Konto.
2. Fügen Sie dieser Sammlung Geofences hinzu. Sie können dies entweder über das Geofence-Upload-Tool in der Amazon-Location-Konsole oder über die Amazon-Location-Geofences-API tun. Weitere Informationen zu verfügbaren Optionen finden Sie unter [Zugriff auf Amazon Location](#).

Geofences können entweder durch ein Polygon oder durch einen Kreis definiert werden. Verwenden Sie ein Polygon, um festzustellen, wann ein Gerät in einen bestimmten Bereich gelangt. Verwenden Sie einen Kreis, um festzustellen, wann ein Gerät innerhalb eines bestimmten Abstands (radius) eines Punktes liegt.

3. Sie können mit der Bewertung von Standorten anhand all Ihrer Geofences beginnen. Wenn eine Standortaktualisierung die Grenzen eines oder mehrerer Geofences überschreitet, gibt Ihre Geofence-Erfassungsressource einen der folgenden Geofence-Ereignistypen auf Amazon aus EventBridge:
 - EINGABE – Für jeden Geofence, in dem die Standortaktualisierung ihre Grenze überschreitet, wird ein Ereignis generiert, indem sie eingegeben wird.
 - EXIT – Für jeden Geofence, bei dem die Standortaktualisierung ihre Grenze durch Beenden überschreitet, wird ein Ereignis generiert.

Weitere Informationen finden Sie unter [the section called “Reagieren auf Ereignisse mit EventBridge”](#). Sie können die Überwachung auch mithilfe von -Services wie Amazon CloudWatch und integrieren AWS CloudTrail. Weitere Informationen finden Sie unter [the section called “Überwachung mit CloudWatch”](#) und [the section called “Verwenden von CloudTrail mit Amazon Location”](#).

Wenn Sie beispielsweise eine Flotte von Lkws verfolgen und benachrichtigt werden möchten, wenn sich ein Lkw in einem bestimmten Gebiet eines Ihrer Lager befindet. Sie können ein Geofence für den Bereich um jedes Lager erstellen. Wenn Ihnen die Lkw dann aktualisierte Standorte senden, können Sie Amazon Location Service verwenden, um diese Positionen zu bewerten und festzustellen, ob ein Lkw in einen der Geofence-Bereiche eingetreten (oder verlassen) ist.

Note

Ihre werden anhand der Anzahl der Geofence-Sammlungen abgerechnet, anhand der Sie bewerten. Ihre Rechnung wird nicht von der Anzahl der Geofences in jeder Sammlung beeinflusst. Da jede Geofence-Sammlung bis zu 50.000 Geofences enthalten kann, sollten Sie Ihre Geofences nach Möglichkeit zu weniger Sammlungen kombinieren, um Ihre Kosten für Geofence-Bewertungen zu senken. Die generierten Ereignisse enthalten die ID des einzelnen Geofence in der Sammlung sowie die ID der Sammlung.

Geofence-Ereignisse

Auf Standorte für Positionen, die Sie überwachen, wird durch eine ID namens verwiesen `DeviceId` (und die Positionen werden als Gerätepositionen bezeichnet). Sie können eine Liste von Gerätepositionen senden, um sie direkt an die Geofence-Erfassungsressource auszuwerten, oder Sie können einen Tracker verwenden. Weitere Informationen zu Trackern finden Sie im nächsten Abschnitt.

Sie erhalten Ereignisse (über Amazon EventBridge) nur, wenn ein Gerät einen Geofence betritt oder verlässt, nicht für jede Positionsänderung. Das bedeutet, dass Sie in der Regel Ereignisse empfangen und viel seltener darauf reagieren müssen als bei jeder Aktualisierung der Geräteposition.

Note

Bei der ersten Standortbewertung für eine bestimmte wird davon ausgegangen `DeviceID`, dass sich das Gerät zuvor nicht in Geofences befand. Das erste Update generiert also ein `ENTER` Ereignis, wenn es sich innerhalb eines Geofence in der Sammlung befindet, und kein Ereignis, wenn nicht.

Um zu berechnen, ob ein Gerät einen Geofence erreicht oder verlassen hat, muss Amazon Location Service den vorherigen Positionsstatus für das Gerät beibehalten. Dieser Positionsstatus

wird 30 Tage lang gespeichert. Nach 30 Tagen ohne Aktualisierung für ein Gerät wird eine neue Standortaktualisierung als erste Positionsaktualisierung behandelt.

Tracker

Ein Tracker speichert Positionsaktualisierungen für eine Sammlung von Geräten. Der Tracker kann verwendet werden, um den aktuellen Standort oder den Standortverlauf der Geräte abzufragen. Es speichert die Updates, reduziert jedoch den Speicherplatz und das visuelle Rauschen, indem die Standorte vor dem Speichern gefiltert werden.

Jede in Ihren Tracker-Ressourcen gespeicherte Positionsaktualisierung kann ein Maß für die Positionsgenauigkeit und bis zu 3 Metadatenfelder über die Position oder das Gerät enthalten, die Sie speichern möchten. Die Metadaten werden als Schlüssel-Wert-Paare gespeichert und können Informationen wie Geschwindigkeit, Richtung, Temperaturdruck oder Motortemperatur speichern.

Note

Tracker-Speicher wird automatisch mit AWS -eigenen Schlüsseln verschlüsselt. Sie können mithilfe von KMS-Schlüsseln, die Sie verwalten, eine weitere Verschlüsselungsebene hinzufügen, um sicherzustellen, dass nur Sie auf Ihre Daten zugreifen können. Weitere Informationen finden Sie unter [Datenverschlüsselung im Ruhezustand für Amazon Location Service](#).

Tracker-Positionsfilterung und -speicherung sind allein nützlich, aber Tracker sind besonders nützlich, wenn sie mit Geofences kombiniert werden. Sie können Tracker mit einer oder mehreren Ihrer Geofence-Sammlungsressourcen verknüpfen, und Positionsaktualisierungen werden automatisch anhand der Geofences in diesen Sammlungen ausgewertet. Die ordnungsgemäße Verwendung von Filterung kann auch die Kosten Ihrer Geofence-Bewertungen erheblich senken.

Das folgende Diagramm zeigt Ihnen, wie Sie Tracker-Ressourcen erstellen und verwenden:



1. Zunächst erstellen Sie eine Tracker-Ressource in Ihrem AWS Konto.

2. Legen Sie als Nächstes fest, wie Sie Standortaktualisierungen an Ihre Tracker-Ressourcen senden. Verwenden Sie [-AWS SDKs](#), um Nachverfolgungsfunktionen in Ihre mobilen Anwendungen zu integrieren. Alternativ können Sie MQTT verwenden, indem Sie den step-by-step Anweisungen bei der [Nachverfolgung mit MQTT](#) folgen.
3. Sie können jetzt Ihre Tracker-Ressource verwenden, um den Standortverlauf aufzuzeichnen und auf einer Karte zu visualisieren.
4. Sie können Ihre Tracker-Ressource auch mit einer oder mehreren Geofence-Sammlungen verknüpfen, sodass jedes an Ihre Tracker-Ressource gesendete Positions-Update automatisch anhand des gesamten Geofence in allen verknüpften Geofence-Sammlungen ausgewertet wird. Sie können eine Ressource auf der Detailseite der Tracker-Ressource der Amazon Location-Konsole oder mithilfe der Amazon Location Trackers-API verknüpfen.
5. Anschließend können Sie die Überwachung mithilfe von -Services wie Amazon CloudWatch und integrieren AWS CloudTrail. Weitere Informationen finden Sie unter [the section called "Überwachung mit CloudWatch"](#) und [the section called "Verwenden von CloudTrail mit Amazon Location"](#).

Verwenden von Trackern mit Geofences

Tracker bieten zusätzliche Funktionen in Kombination mit Geofences. Sie verknüpfen einen Tracker mit einer Geofence-Sammlung, entweder über die Amazon Location-Konsole oder die API, um die Tracker-Standorte automatisch auszuwerten. Jedes Mal, wenn der Tracker einen aktualisierten Standort erhält, wird dieser Standort anhand jedes Geofence in der Sammlung ausgewertet und die entsprechenden ENTER - und -EXITEreignisse werden in Amazon generiert EventBridge. Sie können auch eine Filterung auf den Tracker anwenden, und je nach Filterung können Sie die Kosten für Geofence-Bewertungen senken, indem Sie nur sinnvolle Standortaktualisierungen auswerten.

Wenn Sie den Tracker einer Geofence-Sammlung zuordnen, nachdem er bereits einige Positionsaktualisierungen erhalten hat, wird die erste Positionsaktualisierung nach der Zuordnung als erste Aktualisierung für die Geofence-Bewertungen behandelt. Wenn es sich innerhalb eines Geofence befindet, erhalten Sie ein -ENTEREreignis. Wenn es sich nicht innerhalb von Geofences befindet, erhalten Sie kein EXIT Ereignis, unabhängig vom vorherigen Status.

Positionsfilerung

Tracker können die Positionen, die an sie gesendet werden, automatisch filtern. Es gibt mehrere Gründe, warum Sie einige Ihrer Gerätestandortaktualisierungen herausfiltern möchten. Wenn Sie ein System haben, das nur Berichte etwa jede Minute sendet, sollten Sie Geräte nach Zeit

filtern und Positionen nur alle 30 Sekunden speichern und auswerten. Selbst wenn Sie häufiger überwachen, sollten Sie Positionsaktualisierungen filtern, um die Störungen der GPS-Hardware zu bereinigen. GPS-Positionen sind von Natur aus verrauscht. Ihre Genauigkeit ist nicht zu 100 % perfekt, so dass sich selbst ein Gerät, das Telefonie ist, leicht zu bewegen scheint. Bei niedrigen Geschwindigkeiten verursacht dieser Jitter visuelles Überladen und kann zu falschen Ein- und Ausgangsereignissen führen, wenn sich das Gerät nahe am Rand eines Geofence befindet.

Die Positionsfilterung funktioniert, wenn Positionsaktualisierungen von einem Tracker empfangen werden, wodurch das visuelle Rauschen in Ihren Gerätepfaden (Jitter) reduziert wird, die Anzahl der falschen Geofence-Ein- und -Beendigungsereignisse reduziert wird und die Kosten besser verwaltet werden können, indem die Anzahl der gespeicherten Positionsaktualisierungen und ausgelösten Geofence-Bewertungen reduziert werden.

Tracker bieten drei Optionen zur Positionsfilterung, um die Kosten zu verwalten und Jitter in Ihren Standortaktualisierungen zu reduzieren.

- Genauigkeitsbasiert – wird mit jedem Gerät verwendet, das eine Genauigkeitsmessung bietet. Die meisten GPS- und Mobilgeräte stellen diese Informationen bereit. Die Genauigkeit jeder Positionsmessung wird durch viele Umgebungsfaktoren beeinflusst, darunter GPS-Satellitenempfang, Landschaft und Nähe von WLAN- und Blau/Zahngeräten. Die meisten Geräte, einschließlich der meisten Mobilgeräte, können eine Schätzung der Genauigkeit der Messung zusammen mit der Messung liefern. Beim AccuracyBased Filtern ignoriert Amazon Location Standortaktualisierungen, wenn sich das Gerät weniger als die gemessene Genauigkeit bewegt hat. Wenn beispielsweise zwei aufeinanderfolgende Updates von einem Gerät einen Genauigkeitsbereich von 5 m und 10 m haben, ignoriert Amazon Location die zweite Aktualisierung, wenn sich das Gerät weniger als 15 m bewegt hat. Amazon Location wertet weder ignorierte Updates anhand von Geofences aus noch speichert es sie.

Wenn keine Genauigkeit angegeben wird, wird sie als Null behandelt, und die Messung wird als völlig genau angesehen, und es wird keine Filterung auf die Updates angewendet.

Note

Sie können die genauigkeitsbasierte Filterung verwenden, um alle Filter zu entfernen. Wenn Sie eine genauigkeitsbasierte Filterung auswählen, aber alle Genauigkeitsdaten auf Null überschreiben oder die Genauigkeit vollständig weglassen, filtert Amazon Location keine Aktualisierungen heraus.

In den meisten Szenarien ist eine genauigkeitsbasierte Filterung eine gute Wahl für die Filterung von Positionsaktualisierungen, die ein ausgewogenes Verhältnis zwischen dem Ort der Nachverfolgung und der Filterung nicht benötigter Aktualisierungen ermöglicht und so die Kosten senkt.

- Entfernungsbasiert – Wird verwendet, wenn Ihre Geräte keine Genauigkeitsmessung bieten, Sie aber dennoch die Vorteile der Filterung nutzen möchten, um Jitter zu reduzieren und Kosten zu verwalten. -DistanceBasedFilterung ignoriert Standortaktualisierungen, bei denen sich Geräte weniger als 30 m (98,4 Fuß) bewegt haben. Wenn Sie die DistanceBased Positionsfilerung verwenden, wertet Amazon Location diese ignorierten Updates weder anhand von Geofences aus noch speichert es die Updates.

Die Genauigkeit der meisten Mobilgeräte, einschließlich der durchschnittlichen Genauigkeit von iOS- und Android-Geräten, liegt innerhalb von 15 m. In den meisten Anwendungen kann die DistanceBased Filterung die Auswirkungen von Standortungenauigkeiten reduzieren, wenn Geräte auf einer Karte angezeigt werden, und die Klonungseffekte mehrerer aufeinanderfolgender Ein- und Ausgangsereignisse, wenn sich Geräte nahe der Grenze eines Geofence befinden. Es kann auch dazu beitragen, die Kosten Ihrer Anwendung zu senken, indem weniger Aufrufe getätigt werden, um anhand verknüpfter Geofences zu bewerten oder Gerätepositionen abzurufen.

Entfernungsbasierte Filterung ist nützlich, wenn Sie filtern möchten, Ihr Gerät jedoch keine Genauigkeitsmessungen liefert oder wenn Sie eine größere Anzahl von Updates herausfiltern möchten als mit genauigkeitsbasierten.

- Zeitbasiert – (Standard) Wird verwendet, wenn Ihre Geräte sehr häufig (mehr als einmal alle 30 Sekunden) Positionsaktualisierungen senden und Sie Geofence-Bewertungen nahezu in Echtzeit vornehmen möchten, ohne jede Aktualisierung zu speichern. Beim TimeBased Filtern wird jede Standortaktualisierung anhand verknüpfter Geofence-Sammlungen ausgewertet, aber nicht jede Standortaktualisierung wird gespeichert. Wenn Ihre Aktualisierungs-Häufigkeit häufiger als 30 Sekunden beträgt, wird nur ein Update pro 30 Sekunden für jede eindeutige Geräte-ID gespeichert.

Zeitbasierte Filterung ist besonders nützlich, wenn Sie weniger Positionen speichern möchten, aber möchten, dass jede Positionsaktualisierung anhand der zugehörigen Geofence-Sammlungen ausgewertet wird.

Note

Beachten Sie bei der Entscheidung für Ihre Filtermethode die Kosten Ihrer Tracking-Anwendung und die Häufigkeit von Positionsaktualisierungen. Ihnen werden jede Standortaktualisierung und einmal die Bewertung der Positionsaktualisierung anhand jeder verknüpften Geofence-Sammlung in Rechnung gestellt. Wenn Sie beispielsweise eine zeitbasierte Filterung verwenden und Ihr Tracker mit zwei Geofence-Sammlungen verknüpft ist, zählt jede Positionsaktualisierung als eine Standortaktualisierungsanforderung und zwei Geofence-Sammlungsauswertungen. Wenn Sie alle 5 Sekunden Positionsaktualisierungen für Ihre Geräte melden und zeitbasierte Filterung verwenden, werden Ihnen 720 Standortaktualisierungen und 1 440 Geofence-Bewertungen pro Stunde für jedes Gerät in Rechnung gestellt.

Geofence-Terminologie

Geofence-Sammlung

Enthält null oder mehr Geofences. Es ist in der Lage, die Geofence-Überwachung zu überwachen, indem es Ein- und Ausgangsereignisse ausgibt, wenn dies angefordert wird, um eine Geräteposition anhand seiner Geofences zu bewerten.

Geofence

Eine Polygon- oder Kreisgeometrie, die eine virtuelle Grenze auf einer Karte definiert.

Polygongeometrie

Ein Amazon Location Geofence ist eine virtuelle Grenze für ein geografisches Gebiet und wird als Polygongeometrie oder als Kreis dargestellt.

Ein Kreis ist ein Punkt mit einer Entfernung um ihn herum. Verwenden Sie einen Kreis, wenn Sie benachrichtigt werden möchten, wenn sich ein Gerät in einer bestimmten Entfernung von einem Standort befindet.

Ein Polygon ist ein Array, das aus 1 oder mehr linearen Ringe besteht. Verwenden Sie ein Polygon, wenn Sie eine bestimmte Grenze für Gerätebenachrichtigungen definieren möchten. Ein linearer Ring ist ein Array von vier oder mehr Eckpunkten, wobei der erste und der letzte Eckpunkt

identisch sind, um eine geschlossene Grenze zu bilden. Jeder Eckpunkt ist ein zweidimensionaler Punkt der Form *[Längengrad, Breitengrad]*, wobei die Längen- und Breitengradeinheiten Grad sind. Die Scheitelpunkte müssen in entgegengesetzter Reihenfolge um das Polygon herum aufgeführt werden.

 Note

Amazon Location Service unterstützt keine Polygone mit mehr als einem Ring. Dazu gehören Trichter, Trichter oder Multipolygone. Amazon Location unterstützt auch keine Polygone, die im Uhrzeigersinn ausgerichtet sind oder den Antimeridian durchqueren.

Im Folgenden finden Sie ein Beispiel für einen einzelnen linearen externen Ring:

```
[
  [
    [-5.716667, -15.933333],
    [-14.416667, -7.933333],
    [-12.316667, -37.066667],
    [-5.716667, -15.933333]
  ]
]
```

Tracker-Terminologie

Tracker-Ressource

Eine - AWS Ressource, die Standortaktualisierungen von Geräten empfängt. Die Tracker-Ressource bietet Unterstützung für Standortabfragen, z. B. den aktuellen und historischen Gerätestandort. Durch die Verknüpfung einer Tracker-Ressource mit einer Geofence-Sammlung werden Standortaktualisierungen automatisch anhand aller Geofences in der verknüpften Geofence-Sammlung ausgewertet.

Nachverfolgte Positionsdaten

Eine Tracker-Ressource speichert Informationen über Ihre Geräte im Laufe der Zeit. Die Informationen umfassen eine Reihe von Positionsaktualisierungen, wobei jede Aktualisierung Standort, Zeit und optionale Metadaten enthält. Die Metadaten können die Genauigkeit einer Position und bis zu drei Schlüssel-Wert-Paare enthalten, mit denen Sie Schlüsselinformationen zu

jeder Position verfolgen können, z. B. Geschwindigkeit, Richtung, Temperaturdruck, verbleibender Wasserkraft oder Engine-Wasser des von Ihnen verfolgten Fahrzeugs. Tracker behalten den Gerätestandortverlauf 30 Tage lang bei.

Positionsfiltrung

Die Positionsfiltrung kann Ihnen helfen, die Kosten zu kontrollieren und die Qualität Ihrer Tracking-Anwendung zu verbessern, indem Positionsaktualisierungen herausgefiltert werden, die keine wertvolle Informationen liefern, bevor die Updates gespeichert oder anhand von Geofences bewertet werden.

Sie können die TimeBased Filtrung AccuracyBasedDistanceBased, oder auswählen. Standardmäßig ist die Positionsfiltrung auf festgelegtTimeBased.

Sie können die Positionsfiltrung konfigurieren, wenn Sie Tracker-Ressourcen erstellen oder aktualisieren.

RFC 3339-Zeitstempelformat

Amazon Location Service Trackers verwendet das [RFC 3339](#)-Format, das dem ISO-Format ([International Organization for Standardization](#)) 8601 für Datum und Uhrzeit folgt.

Das Format ist „JJJJ-MM-DDThh:mm:ss.sssZ +00:00“:

- YYYY-MM-DD – Stellt das Datumsformat dar.
- T – Zeigt an, dass die Zeitwerte folgen.
- hh:mm:ss.sss – Stellt die Zeit im 24-Stunden-Format dar.
- Z – Zeigt an, dass die verwendete Zeitzone UTC ist, auf die Abweichungen von der UTC-Zeitzone folgen können.
- +00:00 – Optionale Angabe von Abweichungen von der UTC-Zeitzone. Beispielsweise gibt +01:00 UTC + 1 Stunde an.

Beispiel

Für den 2. Juli 2020 um 12:15:20 Uhr nachmittags mit einer Anpassung um eine zusätzliche Stunde an die UTC-Zeitzone.

```
2020-07-02T12:15:20.000Z+01:00
```

Häufige Anwendungsfälle für die Verwendung von Amazon Location Service

Mit Amazon Location Service können Sie eine Reihe von Anwendungen erstellen, von der Asset-Nachverfolgung bis hin zum standortbasierten Marketing. Im Folgenden sind häufige Anwendungsfälle aufgeführt:

Benutzerinteraktion und GeoMarketing

Verwenden Sie Standortdaten, um Lösungen zu entwickeln, die die Interaktion der Benutzer mit dem Marketing für Kunden verbessern. Amazon Location kann beispielsweise ein Ereignis auslösen, das eine Benachrichtigung auslöst, wenn sich ein Kunde, der einen Lebensmittel in seiner mobilen App bestellt hat, in der Nähe befindet. Darüber hinaus können Sie Geotargeting-Features erstellen, damit Telefonie Rabattcodes oder digitale Flyer an Kunden senden können, die sich in der Nähe von Zielspeichern befinden.

Verfolgung von Komponenten

Erstellen Sie Funktionen zur Komponentenverfolgung, um Unternehmen dabei zu unterstützen, die aktuellen und historischen Standorte ihrer Produkte, Mitarbeiter und Infrastruktur zu verstehen. Mit den Features zur Komponentenverfolgung können Sie eine Reihe von Lösungen entwickeln, die das Remote-Personal optimieren, den Versand während der Übertragung sichern und die Dispatch-Effizienz maximieren.

Lieferung

Integrieren Sie Standort-Features in Lieferanwendungen, um den Zielort, die Liefermodelle und ihr Ziel zu speichern, zu verfolgen und zu koordinieren. Beispielsweise verfügt eine Lebensmittelzustellungsanwendung mit integrierten Amazon Location-Funktionen über Standortverfolgungs- und Geofencing-Funktionen, mit denen ein Studios automatisch benachrichtigt werden kann, wenn sich ein Zustelltreiber in der Nähe befindet. Dies reduziert die Wartezeit und trägt dazu bei, die Qualität der gelieferten Lebensmittel aufrechtzuerhalten.

Dieses Thema bietet Ihnen einen Überblick über die Architektur und Schritte für Anwendungen, die Sie mit Amazon Location erstellen können.

Themen

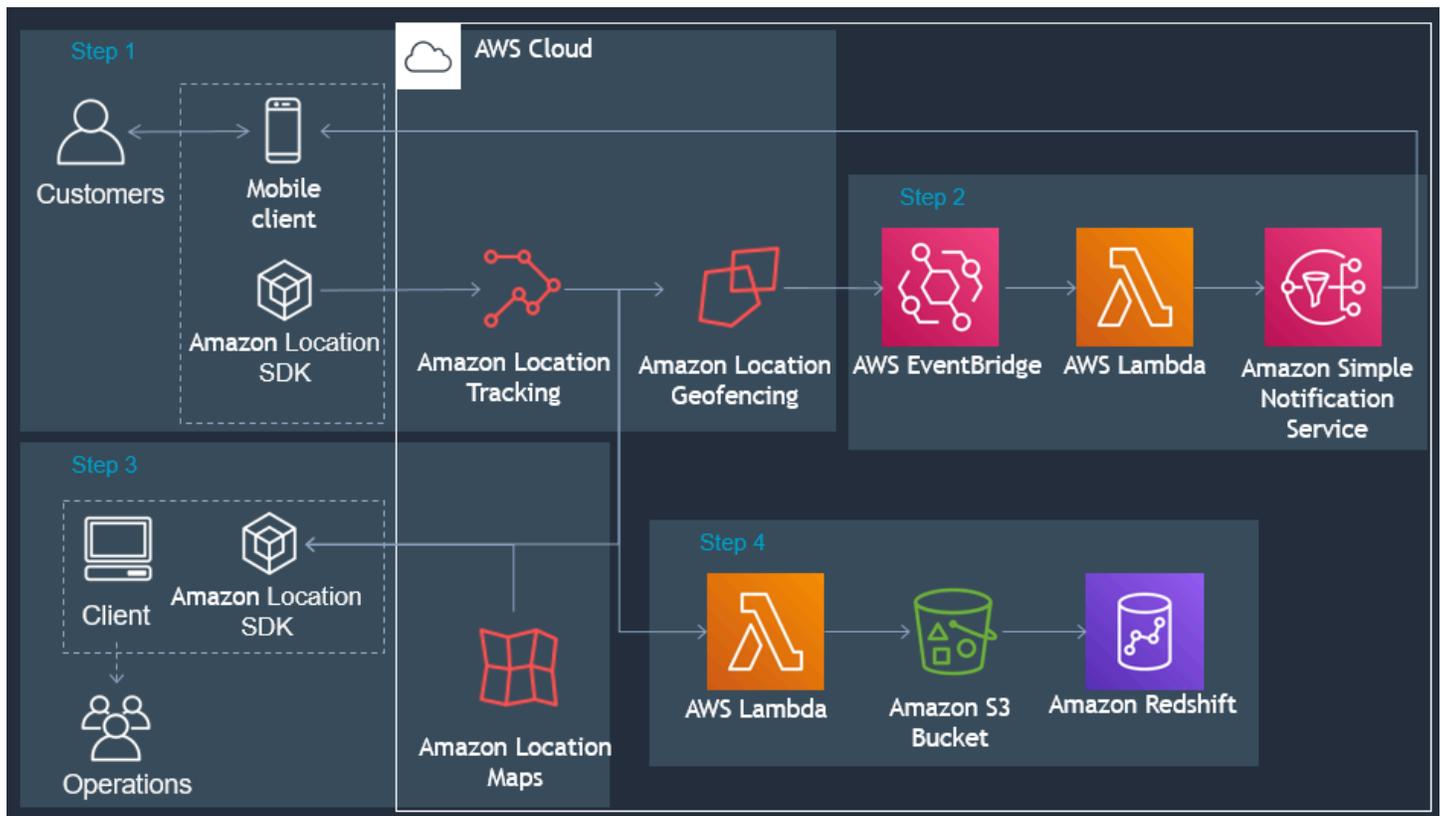
- [Anwendungen für Benutzerinteraktion und GeoMarketing](#)
- [Anwendungen zur Bestandsverfolgung](#)
- [Bereitstellungsanwendungen](#)

Anwendungen für Benutzerinteraktion und GeoMarketing

Im Folgenden finden Sie eine Veranschaulichung einer Benutzerinteraktions- und GeoMarketing-Anwendungsarchitektur mit Amazon Location:

Mit dieser Architektur können Sie:

- Initiieren Sie Ereignisse basierend auf der Nähe eines Ziels, damit Sie Angebote an nahegelegene Kunden senden oder diejenigen ansprechen können, die kürzlich Ihren Betrieb verlassen haben (Geotargeting genannt).
- Visualisieren Sie die Standorte von Kundengeräten auf einer Karte, um Trends im Laufe der Zeit zu überwachen.
- Speichern Sie Kundengerätstandorte, die Sie im Laufe der Zeit analysieren können.
- Analysieren Sie den Standortverlauf, um Trends und Optimierungsmöglichkeiten zu identifizieren.



Im Folgenden finden Sie einen Überblick über die Schritte, die erforderlich sind, um eine Anwendung für Benutzerinteraktionen und GeoMarketing zu erstellen:

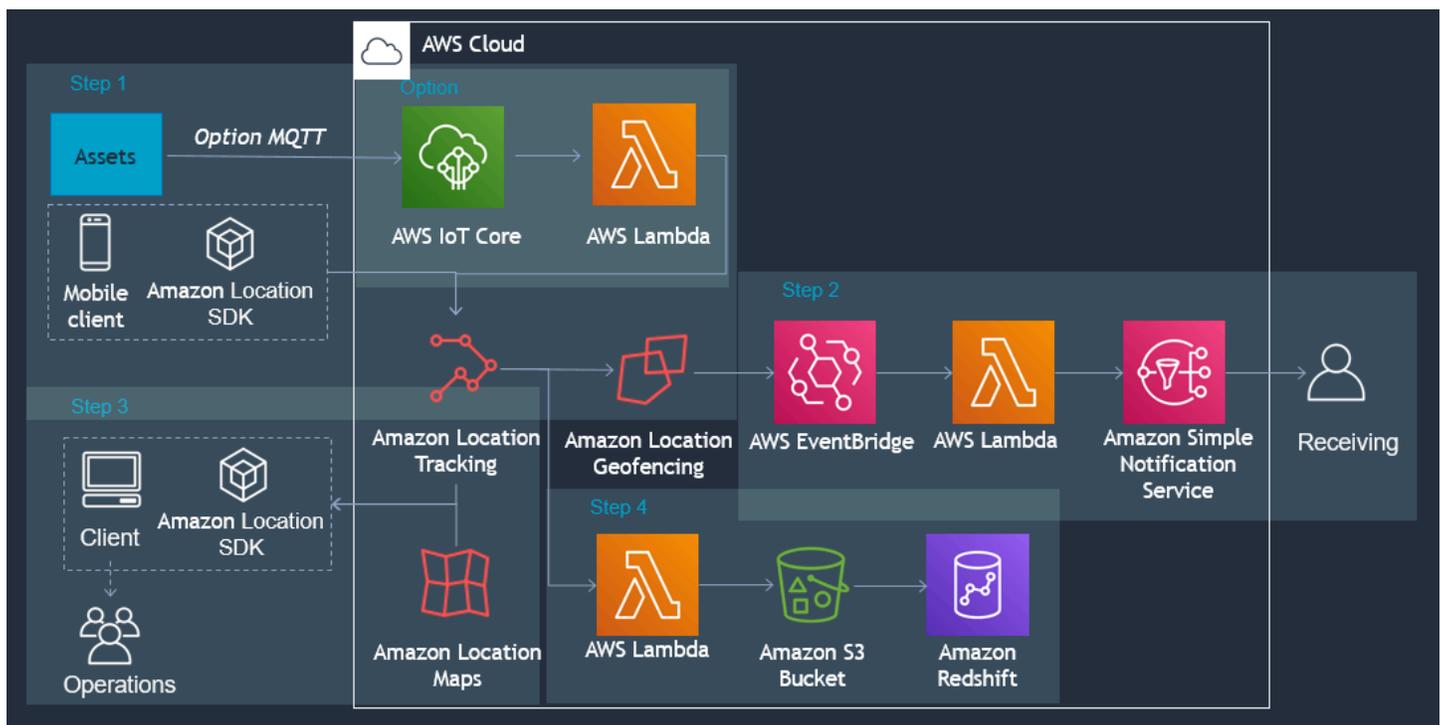
1. Erstellen Sie Ihre Geofences in Geofence Collections und verknüpfen Sie Tracker mit ihnen. Weitere Informationen finden Sie unter [the section called “Geofencing und Nachverfolgung”](#).
2. Konfigurieren Sie Amazon EventBridge so, dass eine Benachrichtigung an Kunden gesendet wird, die einen geografischen Interessenbereich betreten oder verlassen. Weitere Informationen finden Sie unter [the section called “Reagieren auf Ereignisse mit EventBridge”](#).
3. Zeigen Sie Kundenstandorte und Geofences auf einer Karte an. Weitere Informationen finden Sie unter [Verwenden von Karten](#).
4. Speichern Sie Standortdaten zur weiteren Analyse im Langzeitspeicher.
5. Sobald Sie Ihre Anwendung erstellt haben, können Sie Amazon CloudWatch und verwenden, AWS CloudTrail um Ihre Anwendung zu verwalten. Weitere Informationen finden Sie unter [the section called “Überwachung mit CloudWatch”](#) und [the section called “Verwenden von CloudTrail mit Amazon Location”](#).

Anwendungen zur Bestandsverfolgung

Im Folgenden finden Sie eine Veranschaulichung einer Komponentenverfolgungs-Anwendungsarchitektur mit Amazon Location:

Mit dieser Architektur können Sie:

- Zeigen Sie Asset-Standorte auf einer Karte an, um das große Bild zu veranschaulichen. Beispiel: Zeigt eine Heatmap an, die historische Standorte oder Ereignisse verwendet, um einem Betriebs- oder Planungsteam zu helfen.
- Initiieren Sie Ereignisse auf der Grundlage der Nähe der Komponenten, um eine Benachrichtigung an eine empfangende Abteilung zu senden, um sich auf eine Ankunft der Sendung vorzubereiten und die Verarbeitungszeit zu verkürzen.
- Speichern Sie Komponentenstandorte, um Aktionen in Ihren Backend-Anwendungen zu initiieren oder Daten im Laufe der Zeit zu analysieren.
- Analysieren Sie den Standortverlauf, um Trends und Optimierungsmöglichkeiten zu identifizieren.



Im Folgenden finden Sie einen Überblick über die Schritte, die zum Erstellen einer Anwendung zur Komponentenverfolgung erforderlich sind:

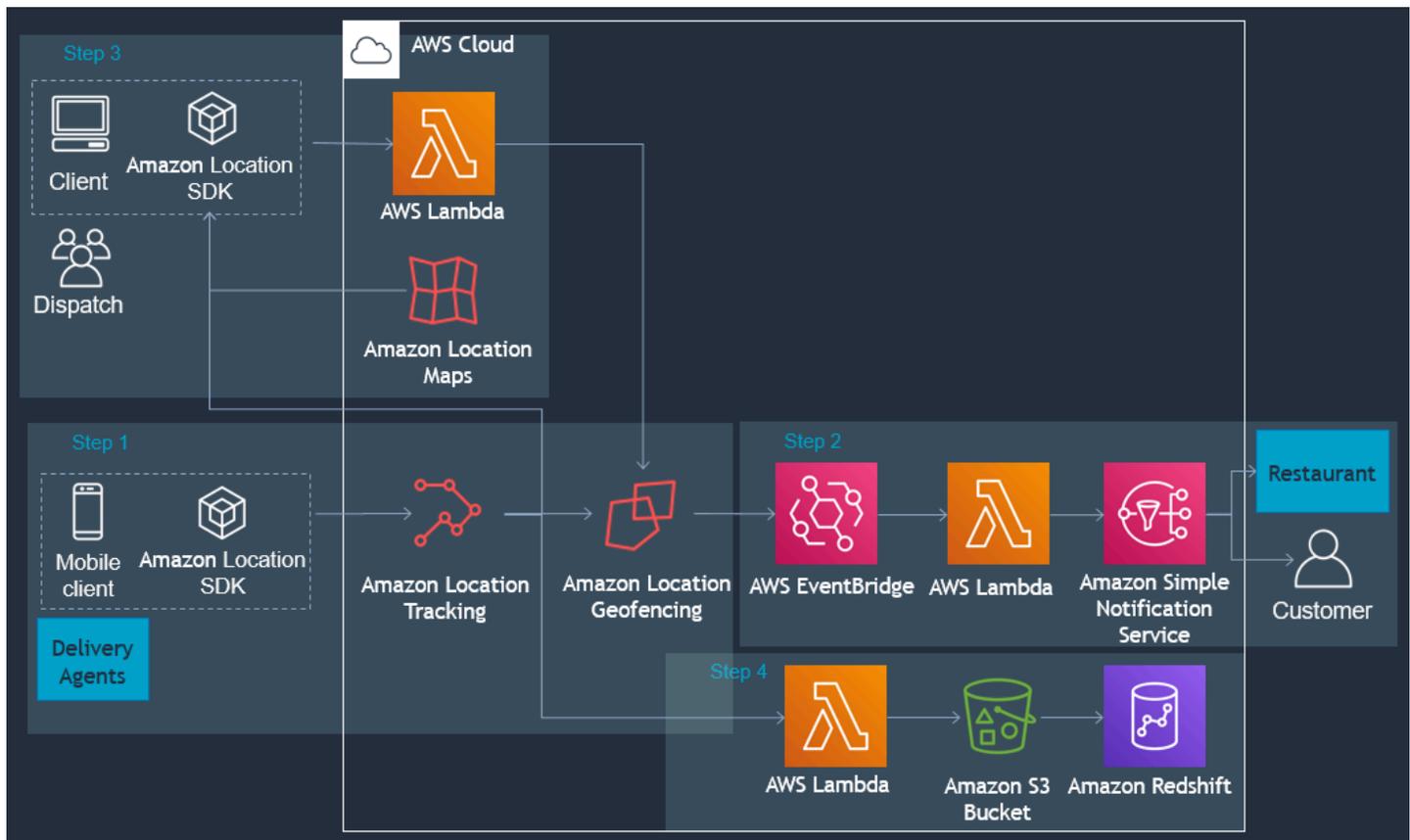
1. Erstellen Sie Ihre Geofences in Geofence Collections und verknüpfen Sie Tracker mit ihnen. Weitere Informationen finden Sie unter [the section called “Geofencing und Nachverfolgung”](#).
2. Konfigurieren Sie Amazon EventBridge so, dass eine Benachrichtigung gesendet oder ein Prozess initiiert wird. Weitere Informationen finden Sie unter [the section called “Reagieren auf Ereignisse mit EventBridge”](#).
3. Zeigen Sie Ihre verfolgten Assets und Ihre aktiven Geofences auf einer Karte an. Weitere Informationen finden Sie unter [Verwenden von Karten](#).
4. Speichern Sie Standortdaten zur weiteren Analyse im Langzeitspeicher.
5. Sobald Sie Ihre Anwendung erstellt haben, können Sie Amazon CloudWatch und verwenden, AWS CloudTrail um Ihre Anwendung zu verwalten. Weitere Informationen finden Sie unter [the section called “Überwachung mit CloudWatch”](#) und [the section called “Verwenden von CloudTrail mit Amazon Location”](#).

Bereitstellungsanwendungen

Im Folgenden finden Sie eine Abbildung einer Bereitstellungsanwendungsarchitektur mit Amazon Location.

Mit dieser Architektur können Sie:

- Initiieren Sie Ereignisse, die auf der Nähe der Zusteller basieren, damit die Annahme rechtzeitig bereit ist und die Kunden benachrichtigt werden können, wenn ihre Lieferung eintrifft.
- Zeigen Sie Treiberstandorte sowie Start- und Drop-off-Standorte nahezu in Echtzeit auf einer Karte an, um Dispatch-Teams das große Bild zu zeigen.
- Speichern Sie die Standorte von Zustellern, damit Sie in Ihrer Backend-Anwendung darauf reagieren oder sie im Laufe der Zeit analysieren können.
- Analysieren Sie den Standortverlauf, um Trends und Optimierungsmöglichkeiten zu identifizieren.



Im Folgenden finden Sie einen Überblick über die Schritte, die zum Erstellen einer Bereitstellungsanwendung erforderlich sind:

1. Erstellen Sie Ihre Geofence-Sammlungen und verknüpfen Sie verfolgte Geräte mit der Sammlung. Weitere Informationen finden Sie unter [the section called “Geofencing und Nachverfolgung”](#).
2. Erstellen Sie eine - AWS Lambda Funktion, um Geofences automatisch hinzuzufügen und zu entfernen, wenn Ihre Bestellungen abgerechnet werden.
3. Konfigurieren Sie Amazon EventBridge so, dass Benachrichtigungen gesendet oder Prozesse initiiert werden. Weitere Informationen finden Sie unter [the section called “Reagieren auf Ereignisse mit EventBridge”](#).
4. Zeigen Sie verfolgte Komponenten und aktive Geofences auf einer Karte an. Weitere Informationen finden Sie unter [Verwenden von Karten](#).
5. Speichern Sie Standortdaten zur weiteren Analyse im Langzeitspeicher.
6. Sobald Sie Ihre Anwendung erstellt haben, können Sie Amazon CloudWatch und verwenden, AWS CloudTrail um Ihre Anwendung zu verwalten. Weitere Informationen finden Sie unter [the section called “Überwachung mit CloudWatch”](#) und [the section called “Verwenden von CloudTrail mit Amazon Location”](#).

Was ist ein Datenanbieter?

Verwenden Sie Amazon Location Service, um über Ihr AWS Konto auf Geolokalisierungsressourcen von mehreren Datenanbietern zuzugreifen, ohne dass Verträge oder Integrationen von Drittanbietern erforderlich sind. Auf diese Weise können Sie sich auf die Erstellung Ihrer Anwendung konzentrieren, ohne Konten, Anmeldeinformationen, Lizenzen und Fakturierung von Drittanbietern verwalten zu müssen.

Die folgenden Amazon-Location-Services verwenden Datenanbieter.

- Karten – Wählen Sie Stile von verschiedenen Kartenanbietern aus, wenn Sie [eine Kartenressource erstellen](#). Sie können Kartenressourcen verwenden, um eine interaktive Karte zur Visualisierung von Daten zu erstellen.
- Ort – Wählen Sie einen Datenanbieter aus, wenn Sie [eine Ortsindexressource erstellen](#), um Abfragen für Geokodierung, Reverse-Geokodierung und Suchen zu unterstützen.
- Routen – Wählen Sie einen Datenanbieter aus, um Abfragen für Routenberechnungen in verschiedenen Regionen und Anwendungen zu unterstützen, wenn Sie [eine Routenrechner-Ressource erstellen](#). Mit dem von Ihnen ausgewählten Datenanbieter können Sie Routen basierend auf up-to-date Straßennetzdaten, Live-Datenverkehrsdaten, geplanten Schließungen und historischen Verkehrsmustern berechnen.

Jeder Anbieter sammelt und kuratiert seine Daten auf unterschiedliche Weise. Sie können auch über unterschiedliche Kenntnisse in verschiedenen Regionen der Welt verfügen. Dieser Abschnitt enthält Details zu unseren Datenanbietern. Sie können je nach Ihren Wünschen einen beliebigen Datenanbieter auswählen.

Stellen Sie sicher, dass Sie die Nutzungsbedingungen für die Verwendung von Amazon Location Service-Datenanbietern lesen. Weitere Informationen finden Sie in den [AWS-Servicebedingungen](#). Weitere Informationen darüber, wie Amazon Location Ihre Privatsphäre schützt, finden Sie auch im [the section called “Datenschutz”](#) Abschnitt .

Abdeckung und Features von Datenanbietern

Die folgende Tabelle zeigt Abdeckung und Features auf hoher Ebene für jeden Datenanbieter.

Datenanbieter	Geografische Abdeckung	Funktionsabdeckung	AWS-Region
Esri	Global	Karten, Orte, Routen	Alle Regionen , in denen Amazon Location verfügbar ist.
Grab	Südostasien	Karten, Orte, Routen	Asien-Pazifik (Singapur), ap-southeast-1 nur .
Hier	Global	Karten, Orte, Routen	Alle Regionen , in denen Amazon Location verfügbar ist.
Daten öffnen	Global	Zuordnungen	Alle Regionen , in denen Amazon Location verfügbar ist.

Weitere Informationen zu den spezifischen Funktionen der einzelnen Datenanbieter finden Sie unter [Funktionen nach Datenanbieter](#).

Jeder Datenanbieter sammelt und erzeugt Daten auf unterschiedliche Weise. Weitere Informationen zu ihren Abdeckungsbereichen finden Sie in den folgenden Themen:

- [Abdeckung: Esri](#)
- [Abdeckung: Grab](#)
- [Abdeckung: HIER](#)
- [Abdeckung: Open Data](#)

Wenn Sie auf ein Problem mit den Daten stoßen und dem Datenanbieter einen Fehler melden möchten, lesen Sie die folgenden Themen:

- [Fehlerberichterstattung an Esri](#)
- [Fehlerberichte für GrabMaps Daten](#)
- [Fehlerberichte an HERE](#)

- [Fehlerberichte und Beitrag zu Open Data](#)

Kartenstile

Jeder Datenanbieter stellt eine Reihe von Kartenstilen bereit, um die von ihm bereitgestellten Kartendaten zu rendern. Ein Stil könnte beispielsweise Satellitenbilder enthalten oder optimiert sein, um die Straßen für die Navigation zu zeigen. Die Liste und Beispiele der Stile für jeden Anbieter finden Sie in den folgenden Themen.

- [Esri-Kartenstile](#)
- [Grab-Kartenstile](#)
- [HERE-Kartenstile](#)
- [Open Data Map-Stile](#)

Weitere Informationen zu den einzelnen Datenanbietern

Die folgenden Links enthalten weitere Informationen zu den einzelnen Datenanbietern.

- [Esri](#)
- [GrabMaps](#)
- [HERE-Technologien](#)
- [Daten öffnen](#)

Esri

Amazon Location Service verwendet die Standortservices von Esri, um AWS Kunden bei der effektiven Verwendung von Karten, Geocode und Berechnung von Routen zu unterstützen. Die Standortservices von Esri werden mit qualitativ hochwertige, autoritativen und ready-to-use Standortdaten erstellt, die von Expertenteams von Kartographen, Geographen und Demographen kuratiert werden.

Weitere Informationen zu Funktionen finden Sie unter [Esri](#) auf Amazon Location Service-Datenanbietern.

Themen

- [Esri-Kartenstile](#)

- [Abdeckung: Esri](#)
- [Nutzungsbedingungen und Datenzuordnung: Esri](#)
- [Fehlerberichterstattung an Esri](#)

Esri-Kartenstile

Amazon Location Service unterstützt die folgenden Esri-Kartenstile beim [Erstellen einer Kartenressource](#).

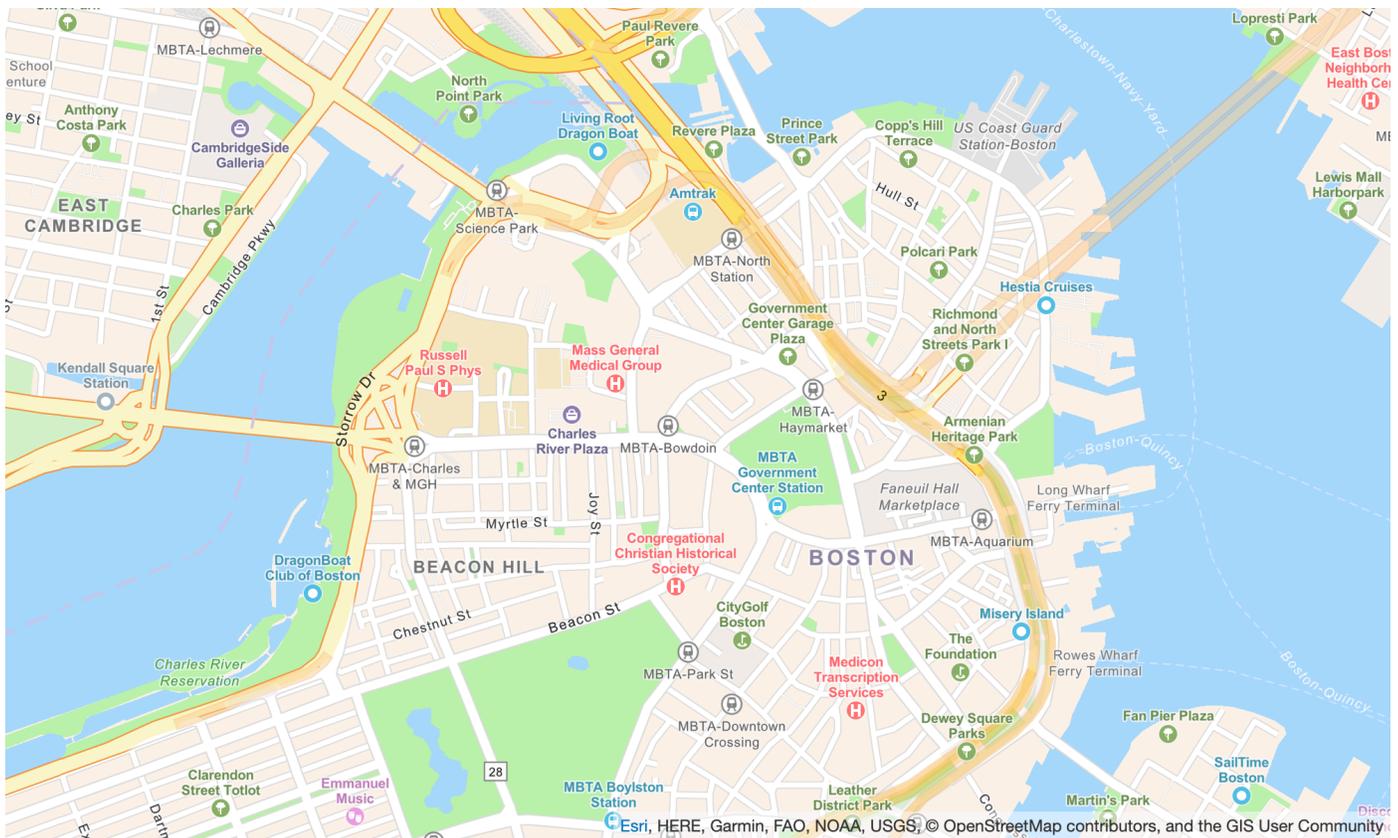
Note

Esri-Kartenstile, die in diesem Abschnitt nicht aufgeführt sind, werden nicht unterstützt.

Die Esri-Vektorstile unterstützen alternative [Politische Ansichten](#).

Esri Navigation

Esri-Navigation



Name des Kartenstils: `VectorEsriNavigation`

Diese Karte bietet eine detaillierte Basiskarte für die Welt, die mit einem benutzerdefinierten Navigationskartenstil symbolisiert ist, der für die Verwendung am Tag auf Mobilgeräten entwickelt wurde.

Diese umfassende Straßenkarte umfasst Hoch-, Haupt-, Haupt-, Neben-, Wasser- und Wassergebiete, Städte, Parks, Kennzeichen, Gebäudegrundrisse und Verwaltungsgrenzen. Die Vektorkachelschicht in dieser Karte wird mit denselben Datenquellen erstellt, die für die World Straßen Map und andere Esri-Basiskarten verwendet werden. Aktivieren Sie die POI Ebene, indem Sie sie in festlegen [CustomLayers](#), um die zusätzlichen Ortsdaten zu nutzen.

Weitere Informationen finden Sie unter [Esri World Navigation](#) auf der Esri-Website.

Note

Für `VectorEsriNavigation` die obige Abbildung ist die POI Ebene aktiviert.

Schriftarten

Amazon Location stellt Schriftarten mithilfe von bereit [GetMapGlyphs](#). Die folgenden Schrift-Stacks sind für diese Zuordnung verfügbar:

- Arialisch kursiv
- Arial Normal
- Arial Bold
- Arial Unicode MS Bold
- Arial Unicode MS Regulär

Esri Imagery

Esri-Imagery

Name des Kartenstils: `RasterEsriImagery`

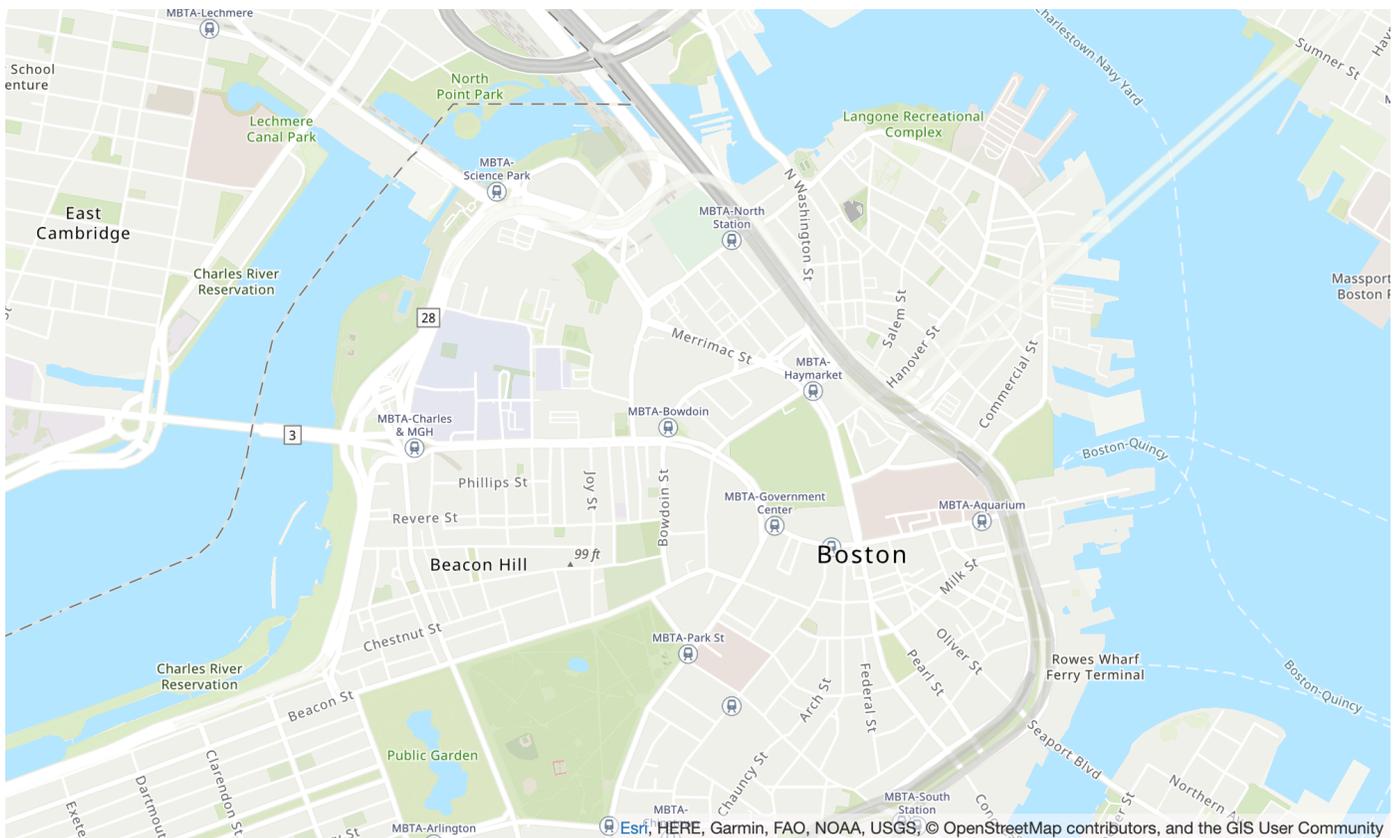
Diese Karte bietet ein Meter oder bessere Satelliten- und aerielle Bilder in vielen Teilen der Welt und Satellitenbilder mit niedrigerer Auflösung weltweit.

Die Karte enthält 15 Millionen Bilder im kleinen und mittleren Maßstab (~1:591M bis ~1:72k) und 2,5 Millionen SPOT Imagery (~1:288k bis ~1:72k) für die Welt. Die Karte enthält Bilder mit einer Auflösung von 0,5 Millionen in den kontinentalen USA und Teilen Westeuropas von Maxar. Diese Karte enthält zusätzliche Maxar-Submeterbilder in vielen Teilen der Welt. In anderen Teilen der Welt hat die GIS User Community Bilder mit unterschiedlichen Auflösungen beigetragen. In ausgewählten Communities sind sehr hochauflösende Bilder (bis zu 0,03 Min.) bis ~1:280 verfügbar.

Weitere Informationen finden Sie unter [Esri World Imagery](#) auf der Esri-Website.

Esri Light

Esri Light



Name des Kartenstils: `VectorEsriTopographic`

Dies bietet eine detaillierte Basiskarte für die Welt, die mit einem klassischen Esri-Kartenstil symbolisiert ist. Dazu gehören Hoch-, Haupt-, Haupt-, Neben-, Land-, Wassergebiete, Städte, Parks, Kennzeichen, Gebäudegrundrisse und Verwaltungsgrenzen.

Diese Basiskarte wird aus einer Vielzahl autoritativer Quellen mehrerer Datenanbieter kompiliert, darunter US Geology microSD (USGS), US Regional Protection Agency (EPA), US National Park

Service (NPS), Bol and Agriculture Organization der Vereinten Nationen (FAO), Department of natural Resources Canada (NRCAN), HERE und Esri. Daten für ausgewählte Bereiche stammen von OpenStreetMap Mitwirkenden. Darüber hinaus werden Daten von der GIS-Community bereitgestellt.

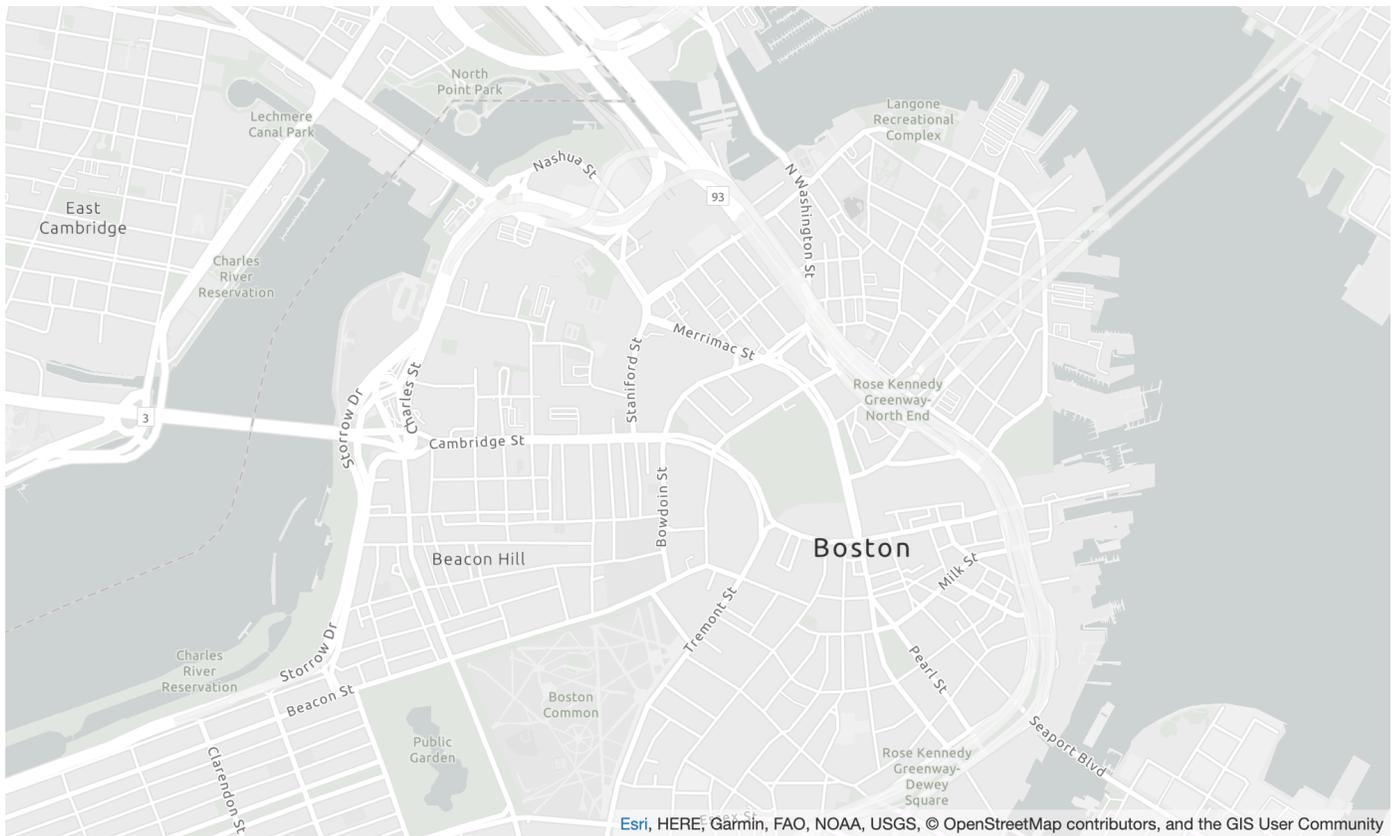
Schriftarten

Amazon Location stellt Schriftarten mithilfe von bereit[GetMapGlyphs](#). Die folgenden Schrift-Stacks sind für diese Zuordnung verfügbar:

- Noto Sans kursiv
- Noto Sans Regulär
- Noto Sans Bold
- Noto Serif Regulär
- Roboto Condensed Light Italic

Esri Light Gray Canvas

Esri Light Gray Canvas



Esri, HERE, Garmin, FAO, NOAA, USGS, © OpenStreetMap contributors, and the GIS User Community

Name des Kartenstils: VectorEsriLightGrayCanvas

Diese Karte bietet eine detaillierte Basiskarte für die Welt, die mit einem hellgrauen, neutralen Hintergrundstil mit minimalen Farben, Labels und Features symbolisiert ist und die Aufmerksamkeit auf Ihre thematischen Inhalte lenken soll.

Diese Vektorkachelschicht wird mit denselben Datenquellen erstellt, die für Light Gray Canvas und andere Esri-Basiskarten verwendet werden. Die Karte umfasst Hoch-, Haupt- und Nebenwege, Baugebiete, Wassergebiete, Städte, Parks, Kennzeichen, Gebäudegrundrisse und Verwaltungsgrenzen.

Weitere Informationen finden Sie unter [Esri Light Gray Canvas](#) auf der Esri-Website.

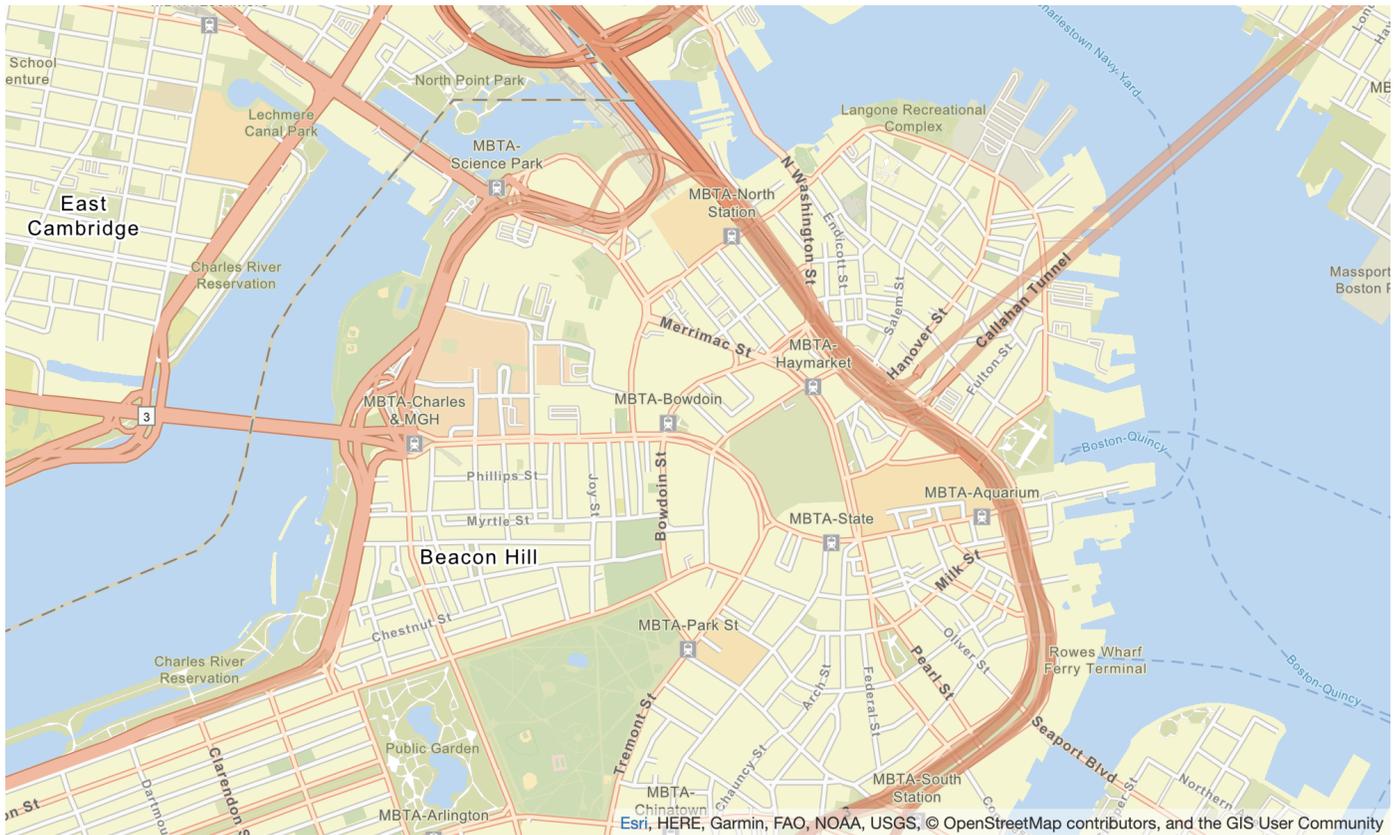
Schriftarten

Amazon Location stellt Schriftarten mithilfe von bereit[GetMapGlyphs](#). Die folgenden Schrift-Stacks sind für diese Zuordnung verfügbar:

- Ubuntu kursiv
- Ubuntu Normal
- Ubuntu Light
- Ubuntu Bold

Esri Street Map

Esri Straßenkarte



Name des Kartenstils: VectorEsriStreets

Diese Karte bietet eine detaillierte Basiskarte für die Welt, die mit einem benutzerdefinierten Navigationskartenstil symbolisiert ist, der für die Verwendung am Tag auf Mobilgeräten entwickelt wurde.

Diese umfassende Straßenkarte umfasst Hoch-, Haupt-, Haupt-, Neben-, Wasser- und Wassergebiete, Städte, Parks, Kennzeichen, Gebäudegrundrisse und Verwaltungsgrenzen. Sie enthält auch eine umfassendere Palette von Orten, z. B. Arzt-, -Services, Jungferne, Gebiete und andere wichtige Punkte. Die Vektorkachelschicht in dieser Karte wird mit denselben Datenquellen erstellt, die für die World Straßen Map und andere Esri-Basiskarten verwendet werden.

Weitere Informationen finden Sie unter [Esri World Straßen](#) auf der Esri-Website.

Schriftarten

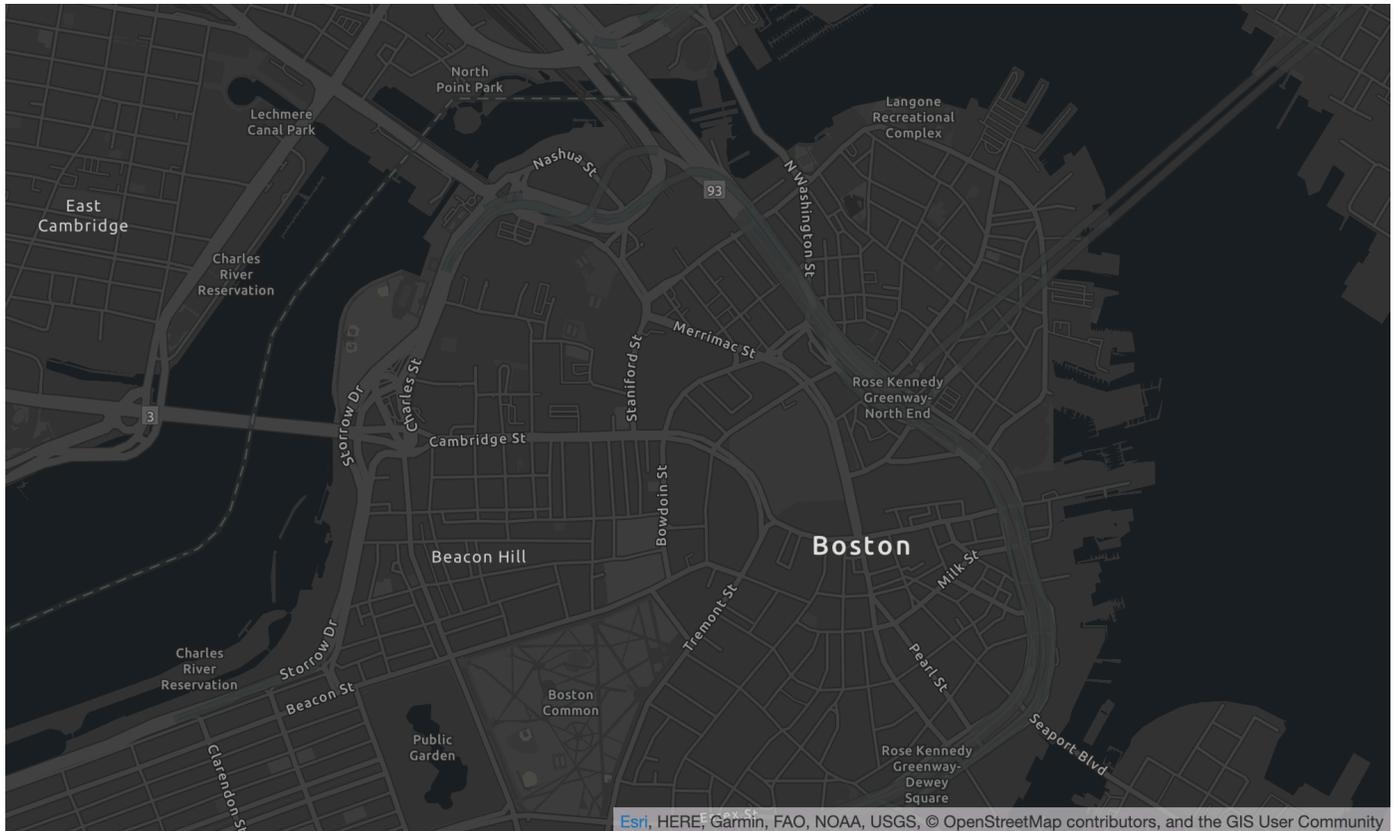
Amazon Location stellt Schriftarten mithilfe von bereit [GetMapGlyphs](#). Die folgenden Schrift-Stacks sind für diese Zuordnung verfügbar:

- Arialisch kursiv

- Arial Normal
- Arial Bold
- Arial Unicode MS Bold
- Arial Unicode MS Regulär

Esri Dark Gray Canvas

Esri Dark Gray Canvas



Name des Kartenstils: `VectorEsriDarkGrayCanvas`

Diese Karte bietet eine detaillierte Vektor-Grundkarte für die Welt, die mit einem hellgrauen, neutralen Hintergrundstil mit minimalen Farben, Labels und Features symbolisiert ist und die Aufmerksamkeit auf Ihre thematischen Inhalte lenken soll.

Diese Karte umfasst Hoch-, Haupt-, Haupt-, Neben-, Land-, Wassergebiete, Städte, Parks, Kennzeichen, Gebäudegrundrisse und Verwaltungsgrenzen. Die Vektorkachelschichten in dieser Karte werden mit denselben Datenquellen erstellt, die für die Canvas-Rasterkarte in Dark Gray und andere Esri-Basiskarten verwendet werden.

Weitere Informationen finden Sie unter [Esri Dark Gray Canvas](#) auf der Esri-Website.

Schriftarten

Amazon Location stellt Schriftarten mithilfe von bereitgestellten [GetMapGlyphs](#). Die folgenden Schrift-Stacks sind für diese Zuordnung verfügbar:

- Ubuntu Medium kursiv
- Ubuntu-Medium
- Ubuntu kursiv
- Ubuntu Normal
- Ubuntu Bold

Abdeckung: Esri

Sie können Esri als Datenanbieter verwenden, um Abfragen für Geokodierung, Reverse-Geokodierung und Suchen zu unterstützen, wenn Sie [eine Ortsindexressource erstellen](#), oder um Abfragen zur Berechnung einer Route zu unterstützen, wenn Sie [eine Routenrechnerressource erstellen](#).

Esri bietet verschiedene Ebenen der Datenqualität in verschiedenen Regionen der Welt. Weitere Informationen zur Abdeckung in Ihrer relevanten Region finden Sie unter:

- [Esri-Details zur Abdeckung der Geokodierung](#)
- [Esri-Details zu Straßennetzen und Verkehrsabdeckung](#)

Nutzungsbedingungen und Datenzuordnung: Esri

Bevor Sie die Daten von Esri verwenden, stellen Sie sicher, dass Sie alle geltenden gesetzlichen Anforderungen erfüllen können, einschließlich der für Esri und geltenden Lizenzbedingungen AWS.

Weitere Informationen zu den AWS Anforderungen finden Sie unter [AWS-Servicebedingungen](#).

Informationen zu den Zuordnungsrichtlinien von Esri finden Sie unter Esri [Data Attributions and Terms of Use](#).

Fehlerberichterstattung an Esri

Wenn Sie auf ein Problem mit den Daten stoßen und Fehler und Unstimmigkeiten an Esri melden möchten, folgen Sie dem Artikel zum technischen Support von Esri unter [How to: Feedback zu Basemaps und Geocodierung](#).

GrabMaps

Grab ist die größte Lieferorganisation in Südostasien mit Millionen von Treiberpartnern und Kunden. Ihre Wahl, [GrabMaps](#), erstellt up-to-date Zuordnungsdaten in diesen Ländern/Regionen für ihren eigenen Gebrauch und andere. Amazon Location Service verwendet die - GrabMapsStandortservices von , um AWS Kunden bei der effektiven Verwendung von Karten, Geocode und Berechnung von Routen zu unterstützen. GrabMapsDie -Standortservices von wurden entwickelt, um qualitativ hochwertige, autoritative und ready-to-use Standortdaten bereitzustellen, insbesondere für Länder im Gebiet der Osten.

Weitere Informationen zu zusätzlichen Funktionen finden Sie unter [GrabMaps](#) auf Amazon Location Service-Datenanbietern.

Important

Grab bietet Karten nur für Nordostasien und ist nur in der Region Asien-Pazifik (Singapur) (ap-southeast-1) verfügbar. Weitere Informationen finden Sie unter [Abgedeckte Länder/Regionen und Gebiete](#).

Themen

- [Grab-Kartenstile](#)
- [Abdeckung: Grab](#)
- [Abgedeckte Länder/Regionen und Gebiete](#)
- [Nutzungsbedingungen und Datenzuordnung: Grab](#)
- [Fehlerberichte für GrabMaps Daten](#)

Grab-Kartenstile

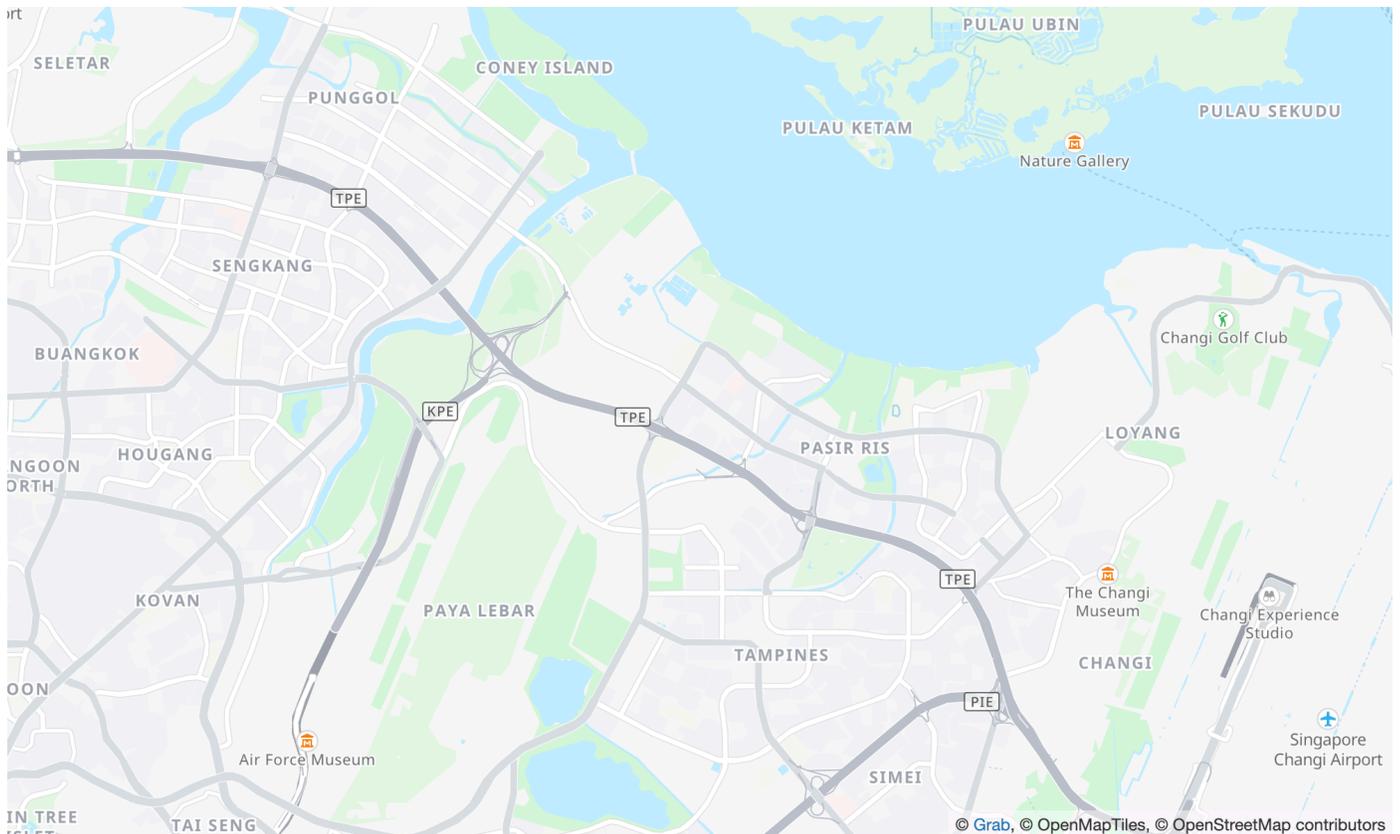
Amazon Location Service unterstützt beim [Erstellen einer Kartenressource](#) die folgenden Grab-Kartenstile:

Note

Grab-Map-Stile, die in diesem Abschnitt nicht aufgeführt sind, werden derzeit nicht unterstützt.

Grab Standard Light Map

Grab Standard Light Map



Name des Kartenstils: `VectorGrabStandardLight`

Die Standard-Grundkarte von Grab mit detaillierter Landnutzungsfarbe, Gebietsnamen, Straßen, Kennzeichen und Präferenzen für Südostasien.

Schriftarten

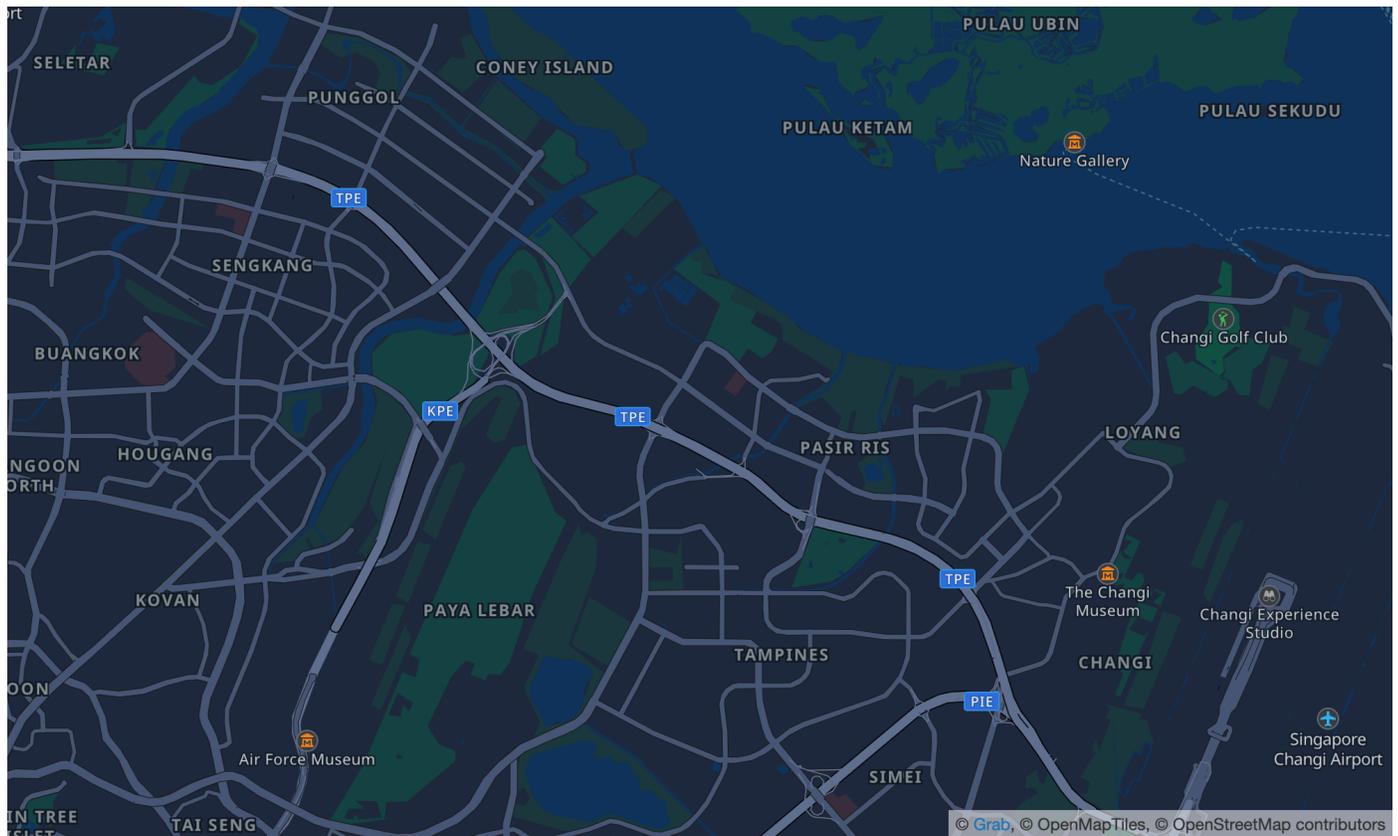
Amazon Location stellt Schriftarten mithilfe von bereit [GetMapGlyphs](#). Die folgenden Schrift-Stacks sind für diese Zuordnung verfügbar:

- Noto Sans Regulär

- Noto Sans Medium
- Noto Sans Bold

Grab Standard Dark Map

Grab Standard-Dunkelkarte



Name des Kartenstils: `VectorGrabStandardDark`

Grabs intensive Variante ihrer Standard-Grundkarte mit detaillierter Grundnutzungsfarbe, Gebietsnamen, Straßen, Merkmalen und Interessenpunkte für Südostasien.

Schriftarten

Amazon Location stellt Schriftarten mithilfe von bereit [GetMapGlyphs](#). Die folgenden Schrift-Stacks sind für diese Zuordnung verfügbar:

- Noto Sans Regulär
- Noto Sans Medium
- Noto Sans Bold

Abdeckung: Grab

Sie können Grab als Datenanbieter verwenden, um Abfragen für Geokodierung, Reverse-Geokodierung und Suchen zu unterstützen, wenn Sie [eine Ortsindexressource erstellen](#), oder um Abfragen zur Berechnung einer Route zu unterstützen, wenn Sie [eine Routenrechnerressource erstellen](#).

Abgedeckte Länder/Regionen und Gebiete

Grab bietet Karten nur für Südostasien und ist nur in der Region Asien-Pazifik (Singapur) verfügbar (ap-southeast-1).

Grab bietet detaillierte Daten für die folgenden Länder/Regionen:

- Malaysia
- Philippinen
- Thailand
- Singapur
- Vietnam
- Indonesien
- Myanmar
- Kambodscha

Note

Außerhalb dieser Bereiche liefern die mit Grab als Datenanbieter erstellten Amazon Location Service-Ressourcen keine Ergebnisse. Dazu gehören Suchergebnisse oder Routen.

Die Karten von Grab liegen innerhalb der folgenden Grenzen:

- Süd – Breitengrad -21,943045533438166
- West – Längengrad 90,0
- Nord – Breitengrad 31,952162238024968
- Ost – Längengrad 146,25

Für die Zoomstufen 1–4 beinhaltet Grab die globale Abdeckung. Für die Zoomstufen 5 und darunter werden Kartenkacheln nur innerhalb dieses Begrenzungsrahmens bereitgestellt.

Note

Außerhalb dieses Begrenzungsrahmens geben die Kartenressourcen des Amazon Location Service, die mit Grab als Datenanbieter erstellt wurden, keine Kartenkacheln zurück. Um 404-Fehler in Ihrer Anwendung zu vermeiden, können Sie die Karte mit einem Begrenzungsrahmen begrenzen, wie unter beschrieben [Festlegen von Erweiterungen für eine Zuordnung mit MapLibre](#).

Grab-Routing-reisemodi

Für das Routing bietet Grab das Auto- und das Netz-Routing für alle zuvor aufgeführten Länder/Regionen.

Grab unterstützt kein Lkw-Routing.

Für Jungfern- und Walking-Routen unterstützt Grab die folgenden Städte:

- Singapur
- Jakarta
- Manila
- Bol-Tab
- Kabel
- Ho Chi Minh Stadt
- Bolo

Nutzungsbedingungen und Datenzuordnung: Grab

Wenn Sie die Daten von Grab verwenden, müssen Sie alle geltenden gesetzlichen Anforderungen erfüllen, einschließlich der für Grab und geltenden Lizenzbedingungen AWS.

Weitere Informationen zu den AWS Anforderungen finden Sie unter [AWS-Servicebedingungen](#).

Informationen zu den Zuordnungsrichtlinien GrabMapsvon finden Sie in Abschnitt 9.23 von Grabs [Datenzuordnungen und Nutzungsbedingungen](#).

Fehlerberichte für GrabMaps Daten

Wenn Sie auf ein Problem mit den Daten von stoßen GrabMaps und Fehler oder Abweichungen melden möchten, [wenden Sie sich an den AWS technischen Support von](#) .

HERE-Technologien

Amazon Location Service verwendet die Standortservices von HERE Technologies, um AWS Kunden bei der effektiven Verwendung von Karten, Geocode und Berechnung von Routen zu unterstützen. Die Standortdaten von HERE bieten eine standortorientierte Plattform, die offen, sicher und privat ist. Durch die Auswahl von HERE-Standortdaten wählen Sie genaue, aktuelle und robuste Daten aus, die nativ in der AWS Cloud bereitgestellt werden.

Weitere Informationen zu Funktionen finden Sie unter [HERE](#) auf Amazon Location Service-Datenanbietern.

Themen

- [HERE-Kartenstile](#)
- [Abdeckung: HIER](#)
- [Nutzungsbedingungen und Datenzuordnung: HERE](#)
- [Fehlerberichte an HERE](#)

HERE-Kartenstile

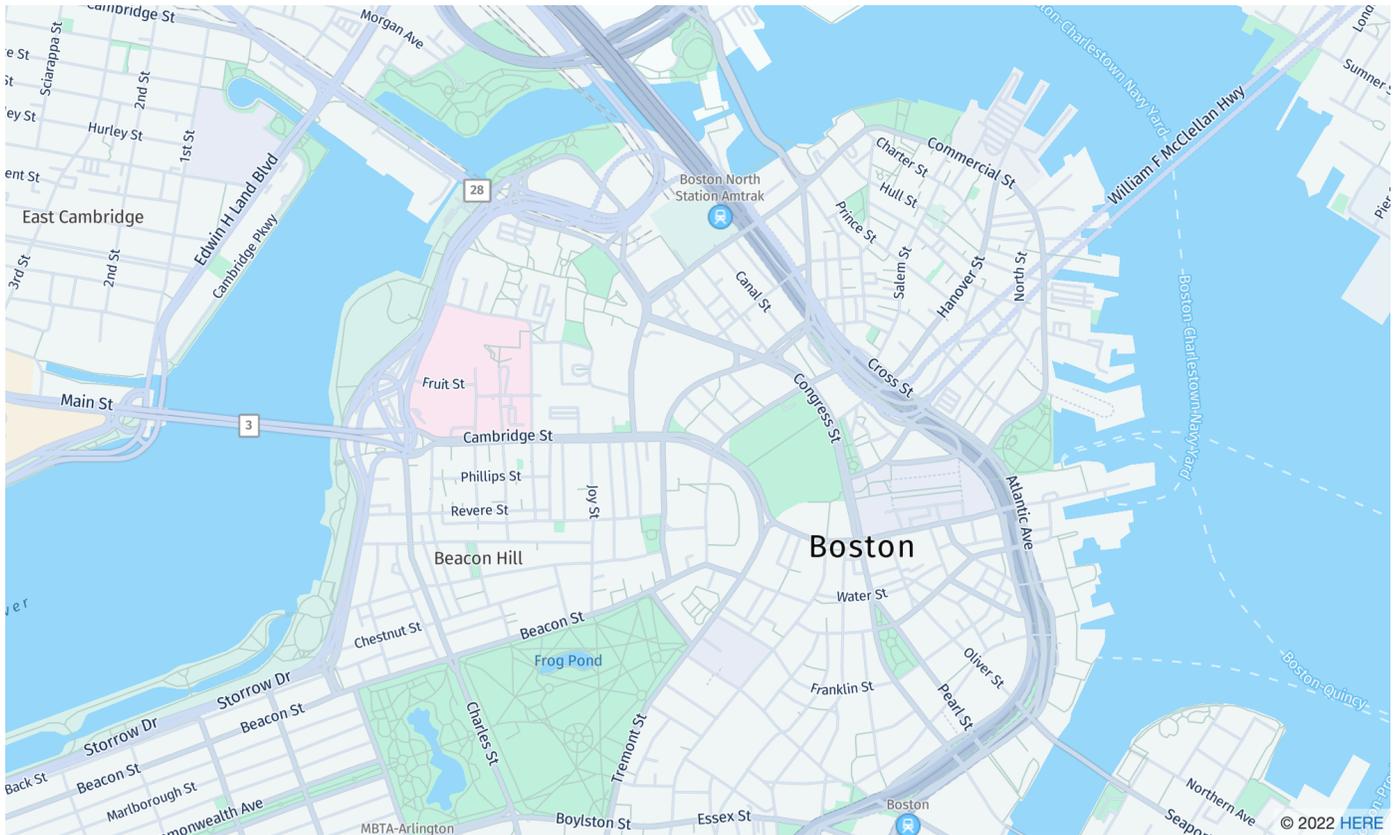
Amazon Location Service unterstützt beim [Erstellen einer Kartenressource](#) die folgenden HERE-Kartenstile:

Note

HERE-Kartenstile, die in diesem Abschnitt nicht aufgeführt sind, werden derzeit nicht unterstützt.

HERE Explore

HERE Erkunden



Name des Kartenstils: VectorHereExplore

HERE Erkunden

Eine detaillierte, neutrale Basiskarte der Welt. Die Straßenkarte umfasst Hoch-, Haupt-, Haupt-, Neben-, Straßen-, Wassergebiete, Städte, Parks, Kennzeichen, Gebäudegrundrisse und Verwaltungsgrenzen. Enthält eine vollständig konzipierte Karte von Japan.

Schriftarten

Amazon Location stellt Schriftarten mithilfe von bereit [GetMapGlyphs](#). Die folgenden Schrift-Stacks sind für diese Zuordnung verfügbar:

- Kursiv für Fura GO
- F Bol GO Regulär
- Fura GO Bold
- Noto Sans CJK JP Light
- Noto Sans CJK JP Regulär
- Noto Sans CJK JP Bold

HERE Imagery

HERE Imagery



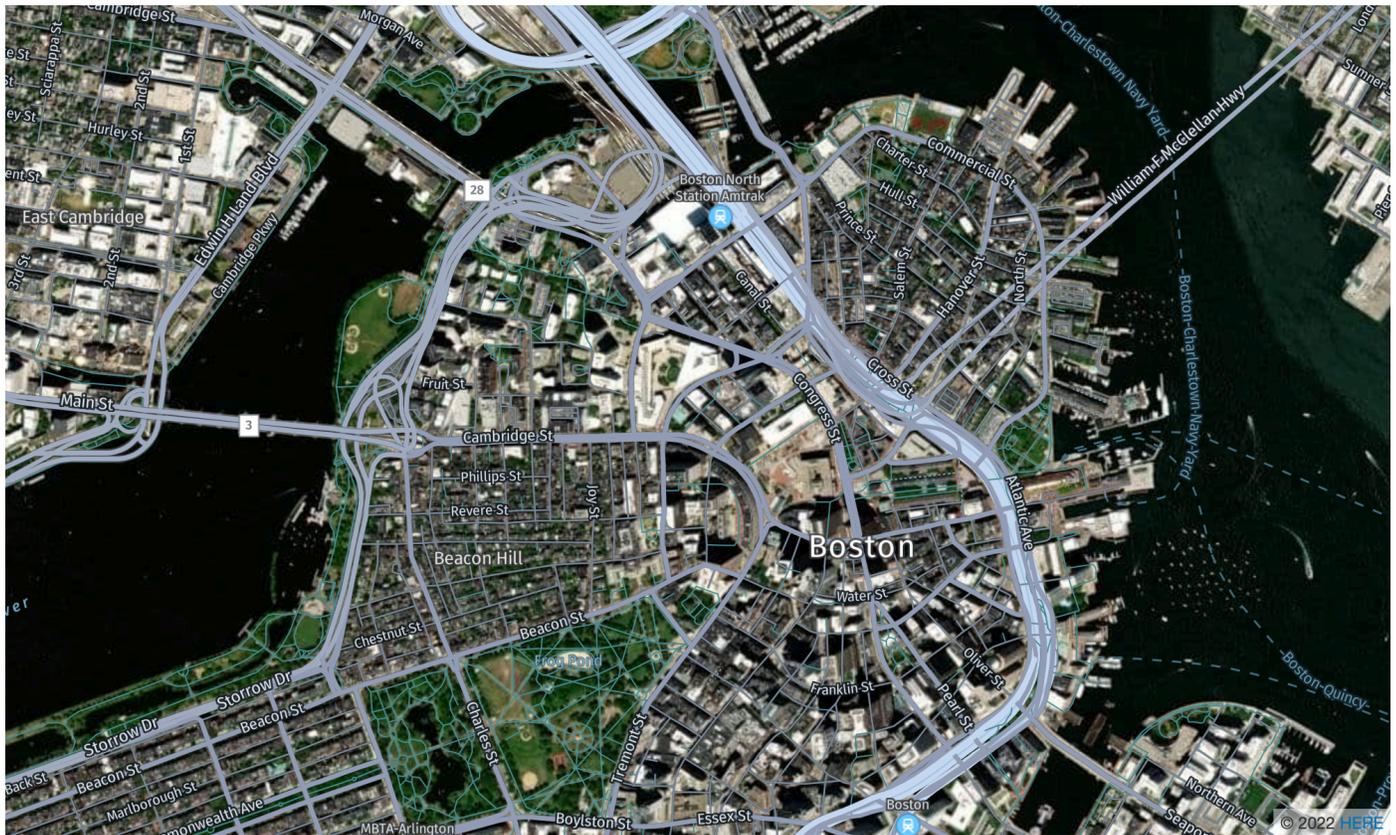
Name des Kartenstils: `RasterHereExploreSatellite`

HERE Imagery

HERE Imagery bietet hochauflösende Satellitenbilder mit globaler Abdeckung.

HERE Hybrid

HERE Hybrid



Name des Kartenstils: HybridHereExploreSatellite

HERE Hybrid

Der Hybridstil HERE zeigt das Straßennetz, Straßennamen und Stadtbezeichnungen über Satellitenbildern an. Dieser Stil überlagert zwei Kartenkacheln: das Satellitenbild (aster-Kachel) im Hintergrund und das Straßennetz und die Labels (Vektorkachel) oben. Dieser Stil ruft automatisch sowohl die Raster- als auch die Vektorkacheln ab, die zum Rendern der Karte erforderlich sind.

Note

Hybridstile verwenden beim Rendern der angezeigten Karte sowohl Vektor- als auch Rasterkacheln. Dies bedeutet, dass mehr Kacheln abgerufen werden als bei alleiniger Verwendung von Vektor- oder Rasterkacheln. Ihre Gebühren beinhalten alle abgerufenen Kacheln.

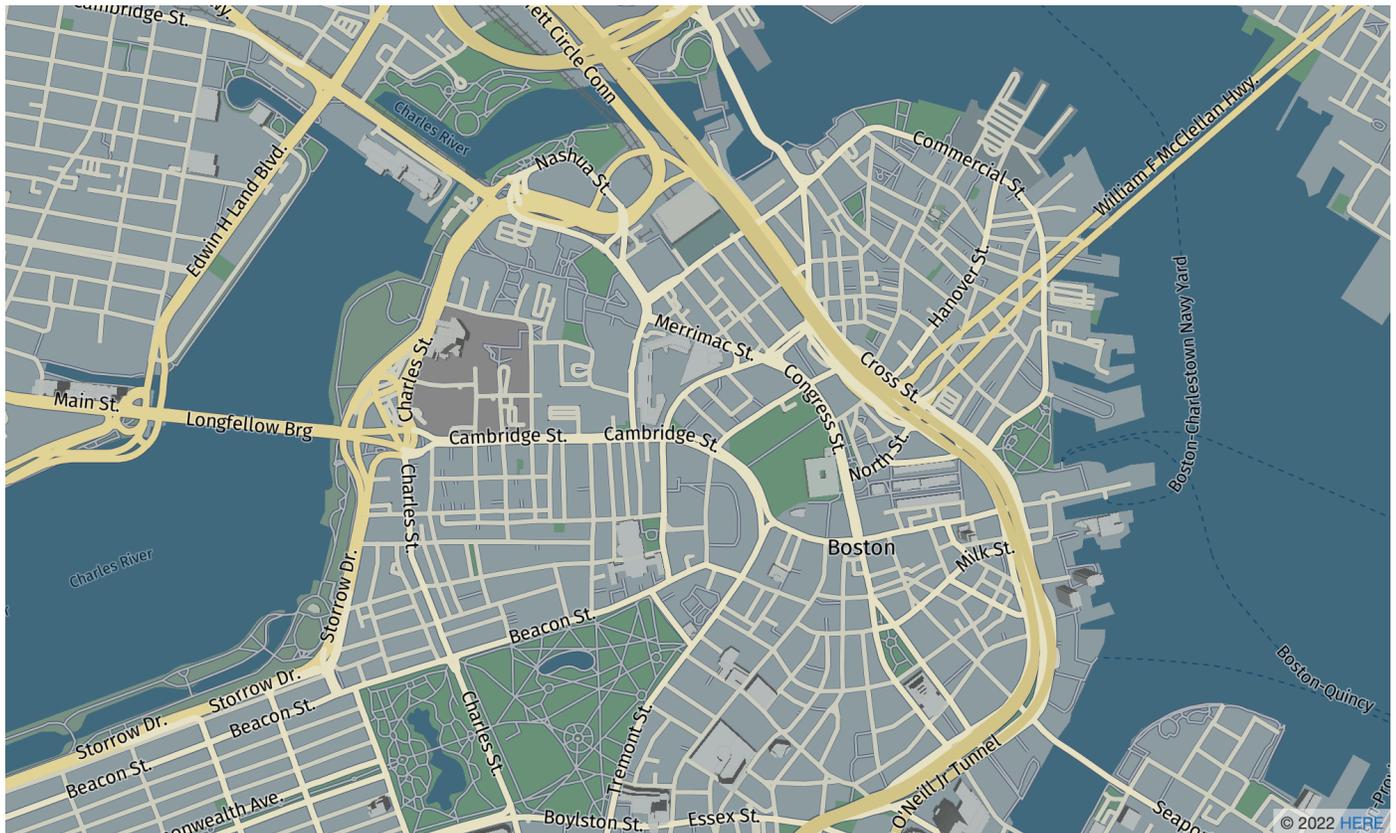
Schriftarten

Amazon Location stellt Schriftarten mithilfe von bereitgestellten [GetMapGlyphs](#). Die folgenden Schrift-Stacks sind für diese Zuordnung verfügbar:

- Kursiv für Fura GO
- F Bol GO Regulär
- Fura GO Bold
- Noto Sans CJK JP Light
- Noto Sans CJK JP Regulär
- Noto Sans CJK JP Bold

HERE Contrast (Berlin)

HERE-Watz (Berlin)



Name des Kartenstils: VectorHereContrast

HERE-Watz (Berlin)

Eine detaillierte Basiskarte der Welt, die 3D- und 2D-Rendering kombiniert. Die Straßenkarte mit hohem Unterschied umfasst Hoch-, Haupt-, Haupt-, Neben-, Straßen-, Baugebiete, Wassergebiete, Städte, Parks, Kennzeichen, Gebäudegrundrisse und Verwaltungsgrenzen.

Schriftarten

Amazon Location stellt Schriftarten mithilfe von bereit[GetMapGlyphs](#). Die folgenden Schrift-Stacks sind für diese Zuordnung verfügbar:

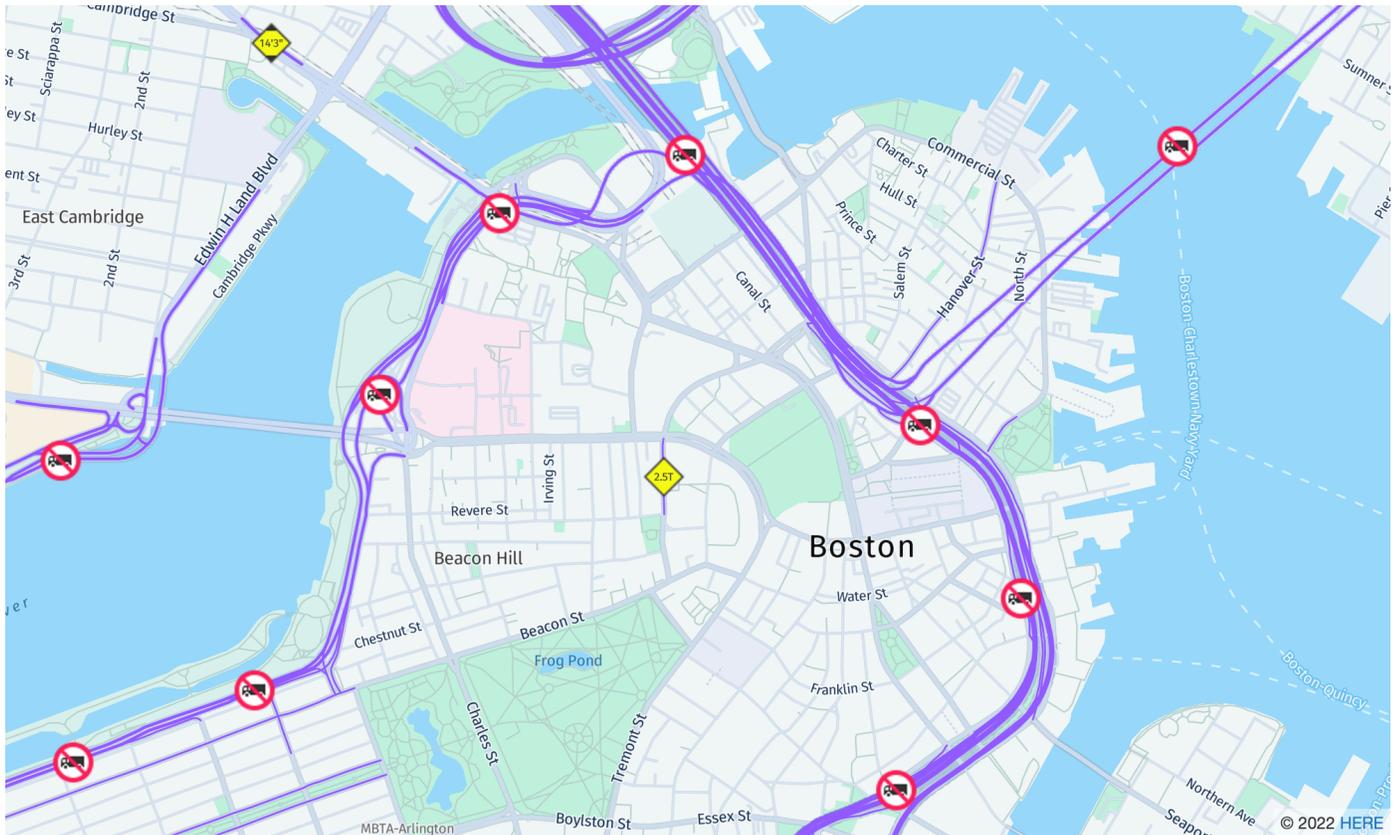
- F Bol GO Regulär
- Fura GO Bold

Note

Dieser Stil wurde von `VectorHereBerlin` (HERE Berlin-Karten) umbenannt. `VectorHereBerlin` ist veraltet, funktioniert aber weiterhin in Anwendungen, die es verwenden.

HERE Explore Truck

Hier erfahren Sie mehr über Truck



Name des Kartenstils: VectorHereExploreTruck

HERE Erkunden von Lkw

Eine detaillierte, neutrale Basiskarte der Welt. Die Straßenkarte baut auf dem HERE-Explore-Stil auf und hebt Einschränkungen und Attribute (einschließlich Breite, Höhe und HAZMAT) mit Symbolen und Symbolen hervor, um Anwendungsfälle im Transport und in der Logistik zu unterstützen.

Schriftarten

Amazon Location stellt Schriftarten mithilfe von bereit [GetMapGlyphs](#). Die folgenden Schrift-Stacks sind für diese Zuordnung verfügbar:

- Kursiv für Fura GO
- F Bol GO Regulär
- Fura GO Bold
- Noto Sans CJK JP Light
- Noto Sans CJK JP Regulär

- Noto Sans CJK JP Bold

Weitere Informationen zur Kartendatenqualität in verschiedenen Regionen der Welt finden Sie unter [HERE-Kartenabdeckung](#).

Abdeckung: HIER

Sie können HERE als Datenanbieter verwenden, um Abfragen für Geokodierung, Reverse-Geokodierung und Suchen zu unterstützen, wenn Sie [eine Ortsindex-Ressource erstellen](#), oder um Abfragen zur Berechnung einer Route zu unterstützen, wenn Sie [eine Routenrechner-Ressource erstellen](#).

HERE bietet verschiedene Ebenen der Datenqualität in verschiedenen Regionen der Welt. Weitere Informationen zur Abdeckung in Ihrer relevanten Region finden Sie hier:

- [HERE-Geokodierungsabdeckung](#)
- [HERE-Abdeckung des Auto-Routings](#)
- [HERE-Abdeckung des Lkw-Routings](#)

Nutzungsbedingungen und Datenzuordnung: HERE

Bevor Sie HERE-Daten verwenden, stellen Sie sicher, dass Sie alle geltenden gesetzlichen Anforderungen erfüllen können, einschließlich der für HERE und geltenden Lizenzbedingungen AWS. Aufgrund von Lizenzbeschränkungen können Sie HERE nicht verwenden, um Geocodierungsergebnisse für Standorte in Japan zu speichern.

Informationen zu den AWS Anforderungen finden Sie unter [AWS-Servicebedingungen](#).

Weitere Informationen zu den Zuordnungsrichtlinien von HERE finden Sie in Abschnitt 2 der [Begriffe HERE Technologies, die für Standort und andere Inhalte gelten](#).

Fehlerberichte an HERE

Um Zuordnungsfehler und Abweichungen zu HERE zu melden, gehen Sie zu <https://www.here.com/contact> und wählen Sie Zuordnungsfehler melden aus.

Daten öffnen

Amazon Location Service bietet über den Open-Data-Anbieter Zugriff auf Open-Source-Map-Daten. Open Data stellt globale Basiskarten bereit, die aus der [Daylight-Map-Verteilung](#) von

[OpenStreetMap \(OSM\)](#) , [natürliche Erde](#) und anderen offenen Datenquellen erstellt wurden. Die bereitgestellten Karten sind so konzipiert, dass sie verschiedene Anwendungen und Anwendungsfälle unterstützen, einschließlich Logistik und Bereitstellung sowie Datenvisualisierung in Web- und mobilen Umgebungen. Mit über einer Million Kartenproduzenten aktualisiert die OSM-Community Hunderttausende von Funktionen pro Tag. Amazon Location Service integriert diese Änderungen regelmäßig.

Weitere Informationen zu Funktionen finden Sie unter [Open Data](#) auf Amazon Location Service-Datenanbietern.

Themen

- [Open Data Map-Stile](#)
- [Abdeckung: Open Data](#)
- [Nutzungsbedingungen und Datenzuordnung: Open Data](#)
- [Fehlerberichte und Beitrag zu Open Data](#)

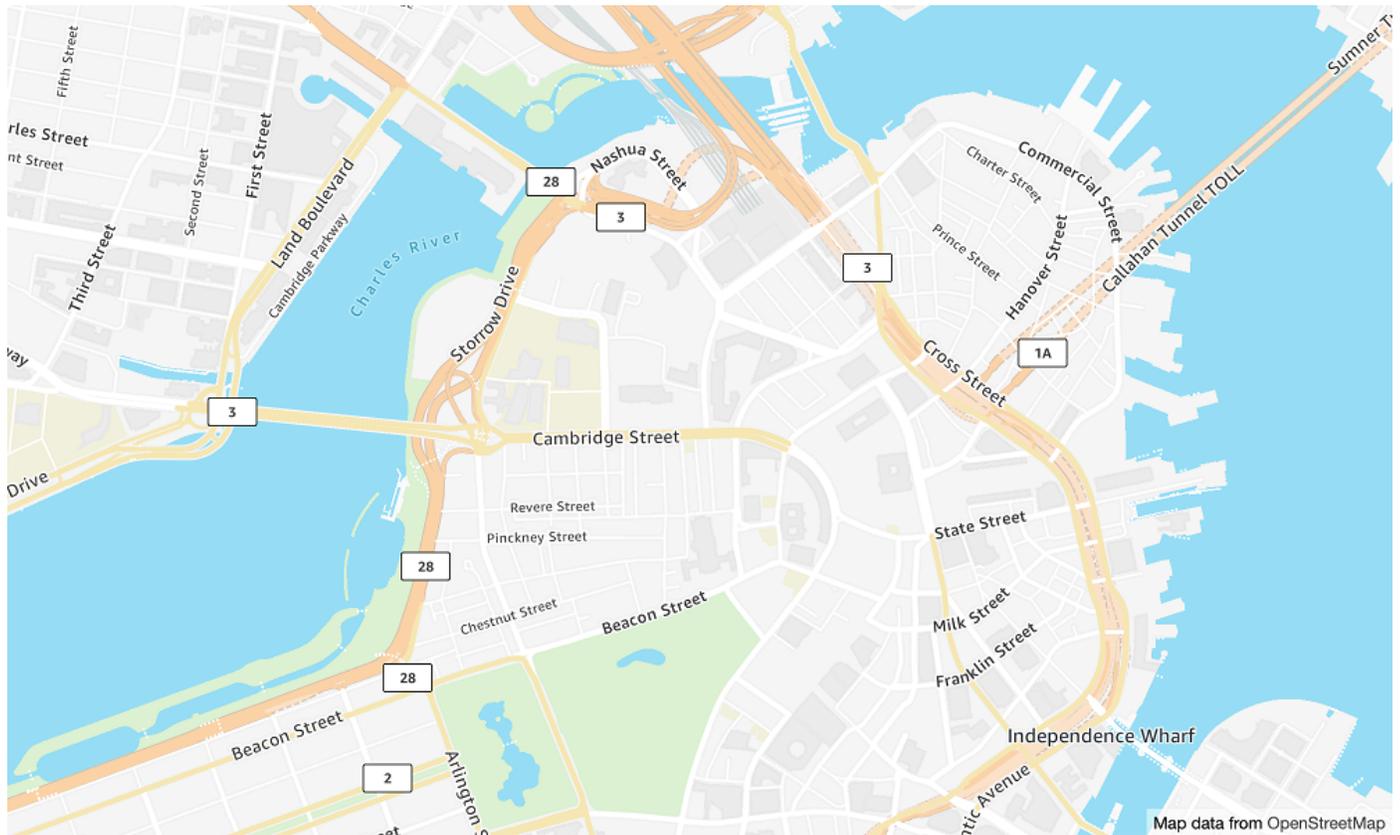
Open Data Map-Stile

Amazon Location Service unterstützt beim [Erstellen einer Kartenressource die](#) folgenden Kartenstile:

Open Data Map-Stile unterstützen alternative [Politische Ansichten](#).

Open Data Standard Light

Open Data Standard Light



Name des Kartenstils: `VectorOpenDataStandardLight`

Dies bietet eine detaillierte Basiskarte für die Welt in einem Lightmap-Stil, die für die Verwendung von Websites und mobilen Anwendungen geeignet ist. Dazu gehören Hoch-, Haupt-, Haupt-, Neben-, Land-, Wassergebiete, Städte, Parks, Kennzeichen, Gebäudegrundrisse und Verwaltungsgrenzen.

Diese Basiskarte basiert auf der OSM-[Daylight-Map-Verteilung](#), die von OpenStreetMap (OSM)-Mitwirkenden kompiliert wurde. Die OSM-Community umfasst über 1,8 Millionen Beitragende, die mehr als 500.000 Funktionen täglich aktualisieren. Amazon Location Service integriert diese Änderungen regelmäßig.

Schriftarten

Amazon Location stellt Schriftarten mithilfe von [GetMapGlyphs](#) bereit. Die folgenden Schrift-Stacks sind für diese Zuordnung verfügbar:

- Amazon Ember Bold, Noto Sans Bold
- Amazon Ember Condensed RC Bold, Noto Sans Bold
- Amazon Ember Condensed RC Regular, Noto Sans Regular

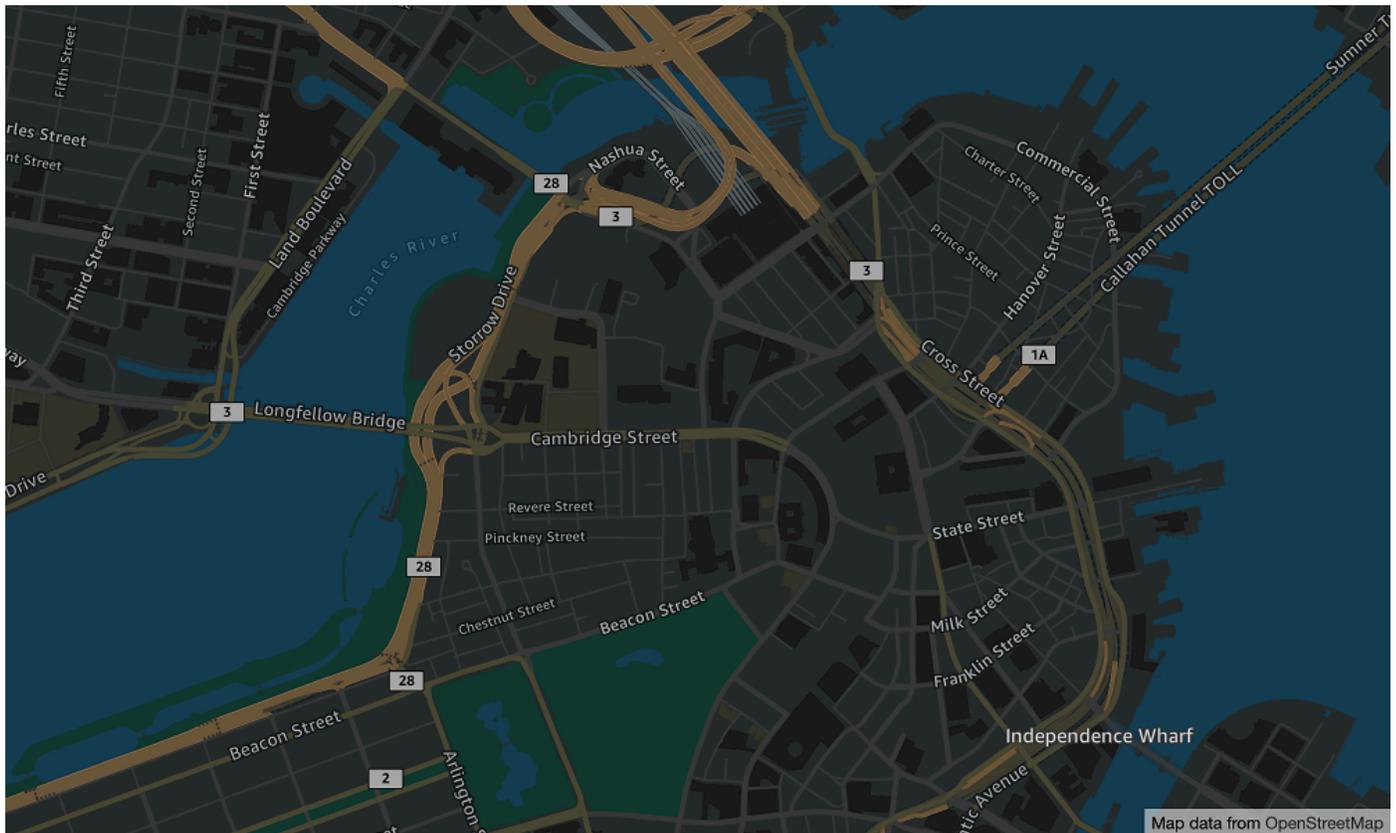
- Amazon Ember Medium, Noto Sans Medium
- Amazon Ember Normal Italic, Noto Sans Italic
- Amazon Ember Regular, Noto Sans Regular
- Amazon Ember Regular, Noto Sans Regular, Noto Sans Arabic Regular
- Amazon Ember Condensed RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold, Noto Sans Bold, Noto Sans Arabic Bold
- Amazon Ember Regular Italic, Noto Sans Italic, Noto Sans Arabic Regular
- Amazon Ember Condensed RC Regular, Noto Sans Regular, Noto Sans Arabic Condensed Regular
- Amazon Ember Medium, Noto Sans Medium, Noto Sans Arabic Medium

 Note

Die von verwendeten Schriftarten `VectorOpenDataStandardLight` sind kombinierte Schriftarten, die Amazon Ember für die meisten Glyphs verwenden, aber Noto Sans für Glyphs, die von nicht unterstützt werden Amazon Ember.

Open Data Standard Dark

Open-Data-Standard-Dunkel



Name des Kartenstils: `VectorOpenDataStandardDark`

Dies ist ein Kartenstil im dunklen Design, der eine detaillierte Basiskarte für die Welt bietet, die für die Verwendung von Websites und mobilen Anwendungen geeignet ist. Dazu gehören Hoch-, Haupt-, Haupt-, Neben-, Land-, Wassergebiet, Städte, Parks, Kennzeichen, Gebäudegrundrisse und Verwaltungsgrenzen.

Diese Basiskarte basiert auf der OSM-[Daylight-Map-Verteilung](#), die von OpenStreetMap (OSM)-Mitwirkenden kompiliert wurde. Die OSM-Community umfasst über 1,8 Millionen Beitragende, die mehr als 500.000 Funktionen täglich aktualisieren. Amazon Location Service integriert diese Änderungen regelmäßig.

Schriftarten

Amazon Location stellt Schriftarten mithilfe von [GetMapGlyphs](#) bereit. Die folgenden Schrift-Stacks sind für diese Zuordnung verfügbar:

- Amazon Ember Bold, Noto Sans Bold
- Amazon Ember Condensed RC Bold, Noto Sans Bold
- Amazon Ember Condensed RC Regular, Noto Sans Regular

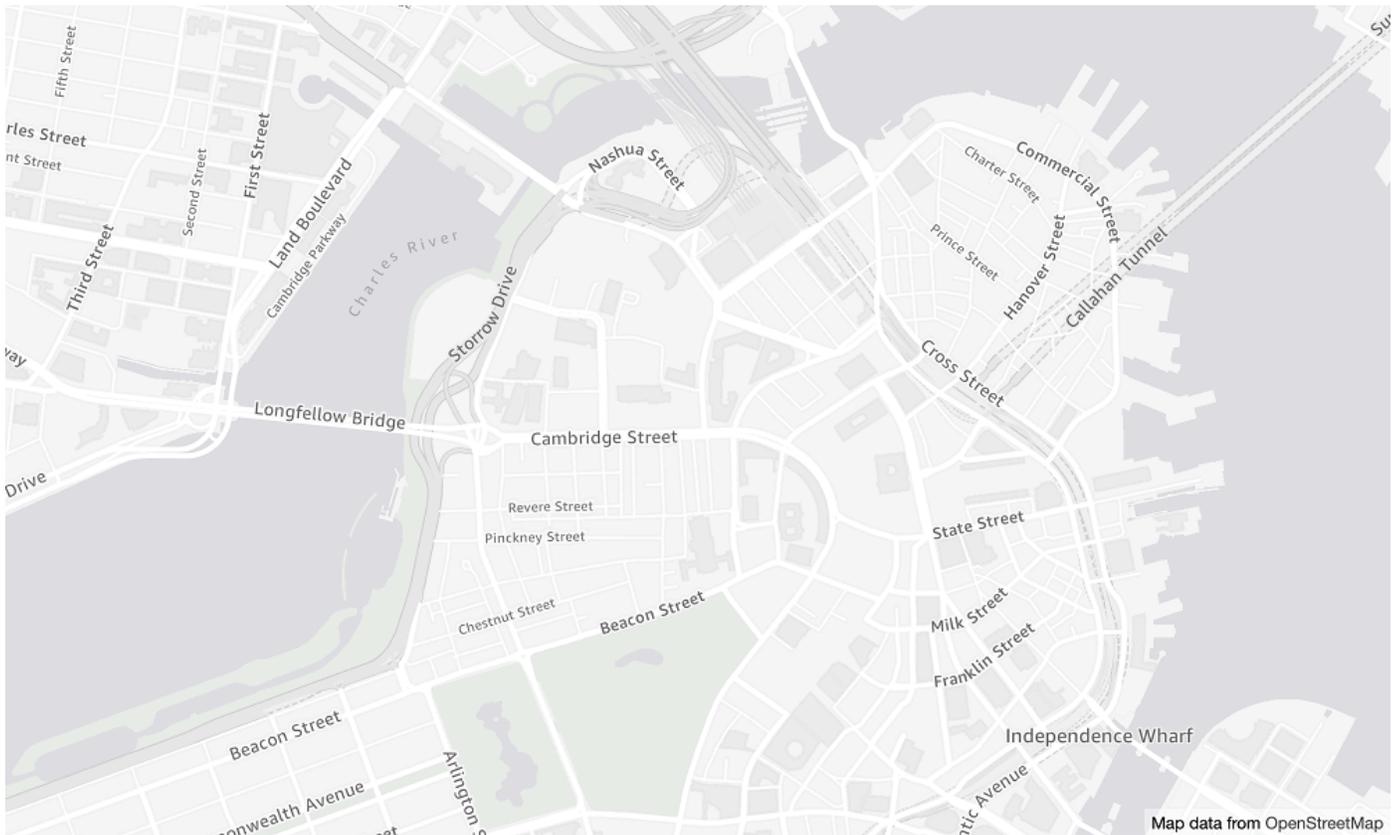
- Amazon Ember Medium, Noto Sans Medium
- Amazon Ember Normal Italic, Noto Sans Italic
- Amazon Ember Regular, Noto Sans Regular
- Amazon Ember Regular, Noto Sans Regular, Noto Sans Arabic Regular
- Amazon Ember Condensed RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold, Noto Sans Bold, Noto Sans Arabic Bold
- Amazon Ember Regular Italic, Noto Sans Italic, Noto Sans Arabic Regular
- Amazon Ember Condensed RC Regular, Noto Sans Regular, Noto Sans Arabic Condensed Regular
- Amazon Ember Medium, Noto Sans Medium, Noto Sans Arabic Medium

 Note

Die von verwendeten Schriftarten `VectorOpenDataStandardDark` sind kombinierte Schriftarten, die Amazon Ember für die meisten Glyphs verwenden, aber Noto Sans für Glyphs, die von nicht unterstützt werden Amazon Ember.

Open Data Visualization Light

Open Data Visualization Light



Name des Kartenstils: `VectorOpenDataVisualizationLight`

Dies ist ein hell gestalteter Stil mit stummgeschalteten Farben und weniger Features, die beim Verständnis überlagerter Daten helfen.

Diese Basiskarte basiert auf der OSM-[Daylight-Map-Verteilung](#), die von OpenStreetMap (OSM)-Mitwirkenden kompiliert wurde. Die OSM-Community umfasst über 1,8 Millionen Beitragende, die mehr als 500.000 Funktionen täglich aktualisieren. Amazon Location Service integriert diese Änderungen regelmäßig.

Schriftarten

Amazon Location stellt Schriftarten mithilfe von [GetMapGlyphs](#) bereit. Die folgenden Schrift-Stacks sind für diese Zuordnung verfügbar:

- Amazon Ember Bold, Noto Sans Bold
- Amazon Ember Condensed RC Bold, Noto Sans Bold
- Amazon Ember Condensed RC Regular, Noto Sans Regular
- Amazon Ember Medium, Noto Sans Medium

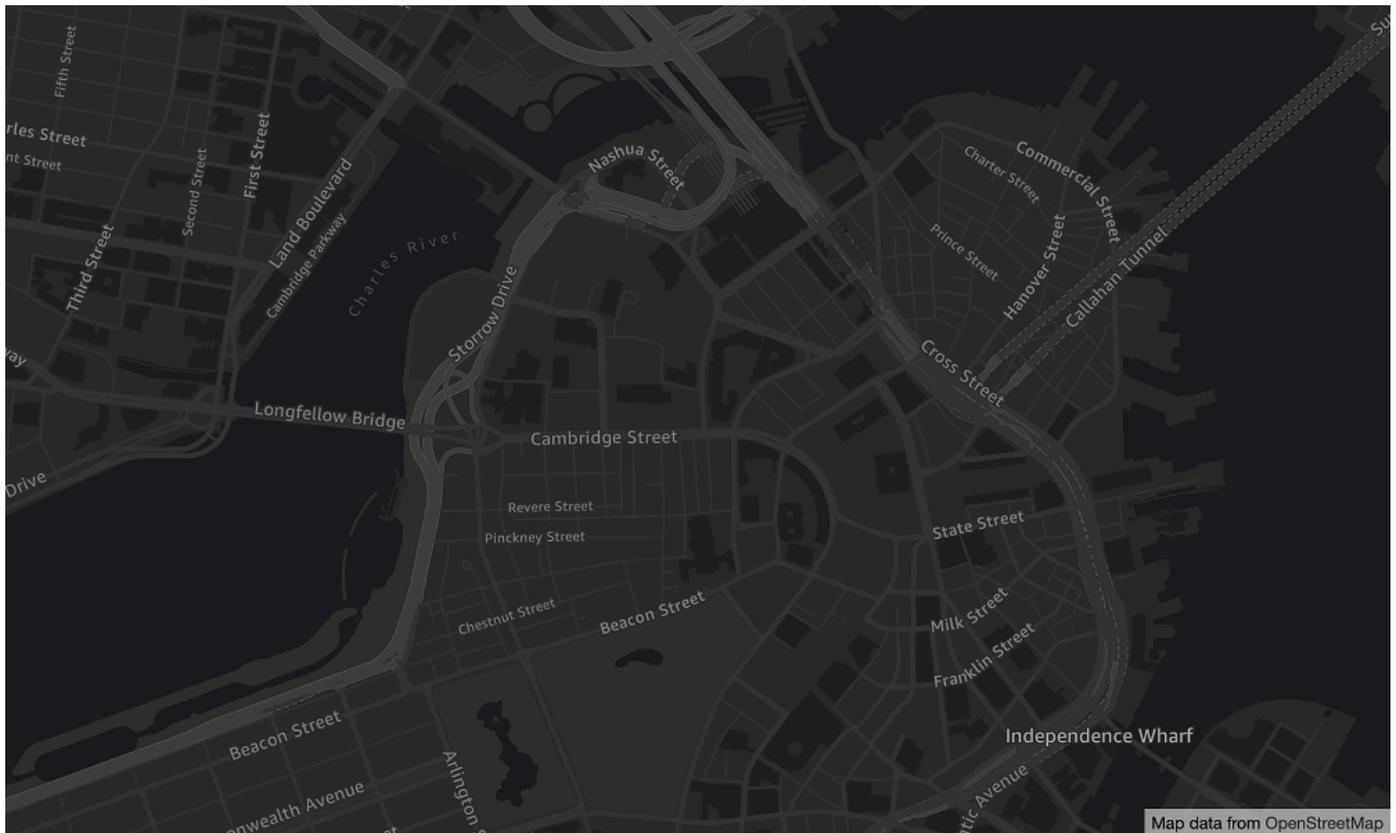
- Amazon Ember Normal Italic, Noto Sans Italic
- Amazon Ember Regular, Noto Sans Regular
- Amazon Ember Regular, Noto Sans Regular, Noto Sans Arabic Regular
- Amazon Ember Condensed RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold, Noto Sans Bold, Noto Sans Arabic Bold
- Amazon Ember Regular Italic, Noto Sans Italic, Noto Sans Arabic Regular
- Amazon Ember Condensed RC Regular, Noto Sans Regular, Noto Sans Arabic Condensed Regular
- Amazon Ember Medium, Noto Sans Medium, Noto Sans Arabic Medium

 Note

Die von verwendeten Schriftarten `VectorOpenDataVisualizationLight` sind kombinierte Schriftarten, die Amazon Ember für die meisten Glyphs verwenden, aber Noto Sans für Glyphs, die von nicht unterstützt werden Amazon Ember.

Open Data Visualization Dark

Open Data Visualization Dark



Name des Kartenstils: `VectorOpenDataVisualizationDark`

Dies ist ein Modell mit farbigem Design mit stummgeschalteten Farben und weniger Features, die beim Verständnis überlagerter Daten helfen.

Diese Basiskarte basiert auf der [OSM-Daylight-Map-Verteilung](#), die von OpenStreetMap (OSM)-Mitwirkenden kompiliert wurde. Die OSM-Community umfasst über 1,8 Millionen Beitragende, die mehr als 500.000 Funktionen täglich aktualisieren. Amazon Location Service integriert diese Änderungen regelmäßig.

Schriftarten

Amazon Location stellt Schriftarten mithilfe von [GetMapGlyphs](#) bereit. Die folgenden Schrift-Stacks sind für diese Zuordnung verfügbar:

- Amazon Ember Bold, Noto Sans Bold
- Amazon Ember Condensed RC Bold, Noto Sans Bold
- Amazon Ember Condensed RC Regular, Noto Sans Regular
- Amazon Ember Medium, Noto Sans Medium

- Amazon Ember Normal Italic, Noto Sans Italic
- Amazon Ember Regular, Noto Sans Regular
- Amazon Ember Regular, Noto Sans Regular, Noto Sans Arabic Regular
- Amazon Ember Condensed RC Bold, Noto Sans Bold, Noto Sans Arabic Condensed Bold
- Amazon Ember Bold, Noto Sans Bold, Noto Sans Arabic Bold
- Amazon Ember Regular Italic, Noto Sans Italic, Noto Sans Arabic Regular
- Amazon Ember Condensed RC Regular, Noto Sans Regular, Noto Sans Arabic Condensed Regular
- Amazon Ember Medium, Noto Sans Medium, Noto Sans Arabic Medium

Note

Die von verwendeten Schriftarten `VectorOpenDataVisualizationDark` sind kombinierte Schriftarten, die Amazon Ember für die meisten Glyphs verwenden, aber Noto Sans für Glyphs, die von nicht unterstützt werden Amazon Ember.

Abdeckung: Open Data

Open Data enthält Karten mit globaler Abdeckung für das Rendern mit einer [Kartenressource von Amazon Location Service](#).

Note

Open Data ist nur für die Verwendung mit Kartenressourcen von Amazon Location Service bestimmt. Sie können Open Data nicht als Datenanbieter verwenden, um Abfragen für Geokodierung, Reverse-Geokodierung und Suchen zu unterstützen oder Abfragen zur Berechnung einer Route zu unterstützen.

Nutzungsbedingungen und Datenzuordnung: Open Data

Bevor Sie Open Data verwenden, stellen Sie sicher, dass Sie alle geltenden gesetzlichen Anforderungen erfüllen können, einschließlich der für Open Data und geltenden Lizenzbedingungen AWS.

Weitere Informationen zu den AWS Anforderungen finden Sie unter [AWS-Servicebedingungen](#).

Informationen zu den Richtlinien für die Open-Data-Zuweisung finden Sie unter [OpenStreetMap Copyright und Lizenz](#) OpenStreetMap von und [unter Richtlinien für die Lizenz/Zuordnung von](#) .

Fehlerberichte und Beitrag zu Open Data

OpenStreetMap (OSM) und Naturerde sind Community-gesteuerte Open-Data-Projekte. Wenn Sie auf ein Problem mit den Daten stoßen, können Sie die Fehler melden oder direkt Korrekturen oder Vorschläge beitragen.

- Um einen Fehler zu melden oder einen Vorschlag in OSM zu unterbreiten, können Sie eine Notiz auf der Karte erstellen. Dies ist ein Kommentar auf der Karte, der Beitragenden hilft, Korrekturen an der Karte vorzunehmen. Sie erstellen Notizen über die [OpenStreetMap Website](#) . Weitere Informationen zu Notizen finden Sie unter [Hinweise](#) im OpenStreetMap Wiki .
- Weitere Informationen zum direkten Beitrag zu OpenStreetMap, einschließlich des Hinzufügens von Speicherorten und der Behebung von Fehlern, finden Sie unter [Contribute Map Data](#) im OpenStreetMap Wiki .
- Um eine Korrekturanfrage für Daten in der Naturerde einzureichen, können Sie ein Problem über die [Naturerde-Website](#) einreichen.

Note

Die Behebung von Fehlern in OpenStreetMap kann schnell auftreten, es kann jedoch einige Zeit dauern, bis Korrekturen in der Daylight-Map-Verteilung der OSM-Daten angezeigt werden, die vom Open Data-Anbieter verwendet werden. Auf der Website [Daylight Map Distribution](#) finden Sie weitere Informationen über den Prozess. Darüber hinaus aktualisiert Amazon Location Service die in Amazon Location Service verwendeten Kartendaten ungefähr monatlich.

Funktionen nach Datenanbieter

In diesem Abschnitt werden die in Amazon Location Service verfügbaren Funktionen beschrieben, kategorisiert nach Datenanbieter.

Die folgende Tabelle bietet einen allgemeinen Überblick über die Funktionen.

Datenanbieter	Geografische Abdeckung	Funktionsabdeckung	AWS-Region
Esri	Global	Karten, Orte, Routen	Alle Regionen , in denen Amazon Location verfügbar ist.
Grab	Südostasien	Karten, Orte, Routen	Nur Asien-Pazifik (Singapur), ap-southeast-1 .
Hier	Global	Karten, Orte, Routen	Alle Regionen , in denen Amazon Location verfügbar ist.
Daten öffnen	Global	Zuordnungen	Alle Regionen , in denen Amazon Location verfügbar ist.

Auf den folgenden Registerkarten werden Details zu den einzelnen Feature-Bereichen angezeigt.

Map Features

Die folgende Tabelle zeigt die Kartenfunktionen nach Datenanbieter. Weitere Informationen zu Kartenkonzepten finden Sie unter [Zuordnungen](#).

Datenanbieter	Unterstützte Kartentypen	Vektorzoomstufen	Raster-Zoomstufen
Esri	Vektor Raster (Imagery) Weitere Informationen finden Sie unter Esri-Kartenstile .	0-15	0-23
Grab	Vektor	0-14	Keine

Datenanbieter	Unterstützte Kartentypen	Vektorzoomstufen	Raster-Zoomstufen
	(nur Südostasien) Weitere Informationen finden Sie unter Grab-Kartenstile .		
Hier	Vektor Raster (Imagery) Hybrid Weitere Informationen finden Sie unter HERE-Kartenstile .	1-17	0-19
Daten öffnen	Vektor Weitere Informationen finden Sie unter Open Data Map-Stile .	0-15	Keine

 Note

Zoomstufen stellen die maximalen und minimalen Einstellungen dar, wie in den APIs der einzelnen Anbieter definiert. Verschiedene Bereiche der Karte können unterschiedliche Höchstwerte haben. Beispielsweise können Seekacheln weniger detaillierte Zoomstufen haben als Bereiche in großen Städten.

MapLibre (und andere Map-Rendering-Engines) ermöglichen es Ihnen, minimale und maximale Zoomstufen festzulegen, und berücksichtigt auch die Zoomstufen des Datenanbieters in einem Bereich, sodass Sie keinen Code schreiben müssen, um diese Abweichungen zu behandeln.

Places and Search

Die folgende Tabelle zeigt die Orts- und Suchfunktionen nach Datenanbieter. Weitere Informationen zu Ortskonzepten finden Sie unter [Ortssuche](#).

Datenanbieter	Geokodierung	Umgekehrte Geokodierung	AutoVervollständigen	GetPlace
Esri	Alle Funktionen, außer: PlaceId	Alle Funktionen, außer: TimeZone PlaceId	Alle Funktionen	Alle Funktionen
Grab	Alle Funktionen, außer: Einheitentyp Nicht unterstützte Kategorien	Alle Funktionen	Alle Funktionen	Alle Funktionen, außer: Einheitentyp SubMunicipality
Hier	Alle Funktionen, außer: Einheitennummer Einheitentyp Relevanz Zusätzliche Einschränkungen beim Filtern	Alle Funktionen	Alle Funktionen	Alle Funktionen, außer: Einheitennummer Einheitentyp SubMunicipality
Daten öffnen	Nicht unterstützt	Nicht unterstützt	Nicht unterstützt	Unterstützt nur:

Datenanbieter	Geokodierung	Umgekehrte Geokodierung	AutoVervollständigen	GetPlace
				SubMunicipality

Route features

Die folgende Tabelle zeigt die Routenfunktionen nach Datenanbieter. Weitere Informationen zu Routenkonzepten finden Sie unter [Routen](#). Ausführlichere Beschreibungen der Einschränkungen der Routenmatrix finden Sie unter [Einschränkungen bei der Ankunft und bei Zielpositionen](#).

Datenanbieter	Reisemodi	Berechnen der Route	Routenmatrix
Esri	Auto, Lkw, Walking	Die Zeitzone und das Ziel müssen innerhalb von 400 km voneinander liegen. Die Gesamtreisezeit darf nicht mehr als 400 Minuten betragen. ArrivalTime wird nicht unterstützt.	Bis zu 10 Ausgangs- und Zielpositionen. Wird in Korea nicht unterstützt. Alle Paare aus Hin- und Rückfahrt müssen innerhalb von 400 km voneinander liegen.
Grab	Auto, Motorcycle. Walking und Bicycle in ausgewählten Städten .	Keine Abstandsgrenzen.	Bis zu 350 Ausgangs- und Zielpositionen.
Hier	Auto, Lkw, Walking	Keine Entfernungsgrenze. Routen, die mehr als 10 km außerhalb eines Kreises liegen, um die Start- und	Bis zu 350 Ausgangs- und Zielpositionen. Alle Start- und Zielpositionen

Datenanbieter	Reisemodi	Berechnen der Route	Routenmatrix
		Zielpositionen zu umgehen, werden nicht berechnet.	müssen mit einem 180-km-Kreiskreis fallen. Längere Routen werden mit zusätzlichen Einschränkungen unterstützt.
Daten öffnen	Nicht unterstützt	Nicht unterstützt	Nicht unterstützt

Nutzungsbedingungen und Datenzuordnung für Datenanbieter

Bevor Sie einen Datenanbieter verwenden, stellen Sie sicher, dass Sie alle geltenden gesetzlichen Anforderungen erfüllen können, einschließlich der für die Nutzung des Anbieters geltenden Lizenzbedingungen.

Weitere Informationen zu den AWS Anforderungen finden Sie unter [AWS-Servicebedingungen](#).

Wenn Sie einen Datenanbieter mit Ihren Amazon Location-Ressourcen für Ihre Anwendung oder Dokumentation verwenden, stellen Sie sicher, dass Sie Zuordnungen für jeden von Ihnen verwendeten Datenanbieter angeben.

Weitere Informationen zur Compliance und Zuordnung der einzelnen Datenanbieter finden Sie in den folgenden Themen.

- Esri – [Nutzungsbedingungen und Datenzuordnung: Esri](#)
- Grab – [Nutzungsbedingungen und Datenzuordnung: Grab](#)
- HIER – [Nutzungsbedingungen und Datenzuordnung: HERE](#)
- Offene Daten – [Nutzungsbedingungen und Datenzuordnung: Open Data](#)

Regionen und Endpunkte von Amazon Location

Amazon Location ist in den folgenden AWS Regionen verfügbar:

Regionen

Name der Region	Region	Endpunkt	Protocol (Protokoll)
USA Ost (Ohio)	us-east-2	geo.us-east-2.amazonaws.com	HTTPS
USA Ost (Nord-Virginia)	us-east-1	geo.us-east-1.amazonaws.com	HTTPS
USA West (Oregon)	us-west-2	geo.us-west-2.amazonaws.com	HTTPS
Asien-Pazifik (Mumbai)	ap-south-1	geo.ap-south-1.amazonaws.com	HTTPS
Asien-Pazifik (Singapur)	ap-southeast-1	geo.ap-southeast-1.amazonaws.com	HTTPS
Asien-Pazifik (Sydney)	ap-southeast-2	geo.ap-southeast-2.amazonaws.com	HTTPS
Asien-Pazifik (Tokio)	ap-northeast-1	geo.ap-northeast-1.amazonaws.com	HTTPS
Kanada (Zentral)	ca-central-1	geo.ca-central-1.amazonaws.com	HTTPS

Name der Region	Region	Endpunkt	Protocol (Protokoll)
Europa (Frankfurt)	eu-central-1	geo.eu-central-1.amazonaws.com	HTTPS
Europa (Irland)	eu-west-1	geo.eu-west-1.amazonaws.com	HTTPS
Europa (London)	eu-west-2	geo.eu-west-2.amazonaws.com	HTTPS
Europa (Stockholm)	eu-north-1	geo.eu-north-1.amazonaws.com	HTTPS
Südamerika (São Paulo)	sa-east-1	geo.sa-east-1.amazonaws.com	HTTPS
AWS GovCloud (USA-West)	us-gov-west-1	geo.us-gov-west-1.amazonaws.com	HTTPS
		geo-fips.us-gov-west-1.amazonaws.com	HTTPS

Note

Weitere Informationen zur Verwendung der Endpunkte in dieser Tabelle finden Sie im folgenden Abschnitt.

Endpunkte

Die allgemeine Syntax eines Amazon-Location-Regionalendpunkts lautet wie folgt:

```
protocol://service-code.geo.region-code.amazonaws.com
```

Innerhalb dieser Syntax verwendet Amazon Location die folgenden Service-Codes:

Service	Service-Code
Amazon Location Maps	Maps
Amazon Location Places	Orte
Amazon Location Geofences	Geofencing
Amazon Location Trackers	Nachverfolgung
Amazon Location Routes	Routen

Der regionale Endpunkt für Amazon Location Maps für USA Ost (Nord-Virginia) wäre beispielsweise: `https://maps.geo.us-east-1.amazonaws.com`.

API-Operation Endpunkte

Die Syntax für einen Endpunkt der Steuerebene von Amazon Location Service lautet wie folgt:

```
protocol://cp.service-code.geo.region-code.amazonaws.com
```

Die Aktionen der Steuerebene für Amazon Location Service sind:

Service	Endpunkt	API-Operation
Amazon Location Maps	<code>https://cp.maps.geo.<i>Region</i>.amazonaws.com</code>	CreateMap DeleteMap DescribeMap ListMaps UpdateMap

Service	Endpoint	API-Operation
Amazon Location Places	https://cp.places.geo.Region.amazonaws.com	CreatePlaceIndex DeletePlaceIndex DescribePlaceIndex ListPlaceIndexes UpdatePlaceIndex
Amazon Location Geofences	https://cp.geofencing.geo.Region.amazonaws.com	CreateGeofenceCollection DeleteGeofenceCollection DescribeGeofenceCollection ListGeofenceCollections UpdateGeofenceCollection
Amazon Location Trackers	https://cp.tracking.geo.Region.amazonaws.com	CreateTracker DeleteTracker DescribeTracker UpdateTracker ListTrackers AssociateTrackerConsumer DisassociateTrackerConsumer ListTrackerConsumers

Service	Endpoint	API-Operation
Amazon Location Routes	<code>https://cp.routes.geo.<i>Region</i>.amazonaws.com</code>	CreateRouteCalculator DeleteRouteCalculator DescribeRouteCalculator ListRouteCalculators UpdateRouteCalculator
Amazon Location-Metadaten	<code>https://cp.metadat a.geo.<i>Region</i>.amazonaw s.com</code>	CreateKey DeleteKey DescribeKey ListKeys UpdateKey

Amazon Location Service-Kontingente

Dieses Thema enthält eine Zusammenfassung der Ratenlimits und Kontingente für Amazon Location Service.

Note

Wenn Sie ein höheres Kontingent benötigen, können Sie die Service Quotas-Konsole verwenden, um [Kontingenterhöhungen für anpassbare Kontingente anzufordern](#). Wenn Sie eine Kontingenterhöhung beantragen, wählen Sie die Region aus, in der Sie die Kontingenterhöhung benötigen, da die meisten Kontingente für die AWS Region spezifisch sind.

Service Quotas sind die maximale Anzahl von Ressourcen, die Sie pro AWS Konto und AWS Region haben können. Amazon Location Service lehnt zusätzliche Anfragen ab, die das Servicekontingent überschreiten.

Ratenlimits (Kontingente, die mit Rate von... beginnen) sind die maximale Anzahl von Anforderungen pro Sekunde, mit einer Burst-Rate von 80 Prozent des Limits innerhalb eines Teils der Sekunde, definiert für jede API-Operation. Amazon Location Service drosselt Anfragen, die das Ratenlimit des Vorgangs überschreiten.

Name	Standard	Anpas	Beschreibung
API-Schlüsselressourcen pro Konto	Jede unterstützte Region: 500	Nein	Die maximale Anzahl von API-Schlüsselressourcen (aktiv oder abgelaufen), die Sie pro Konto haben können.
Geofence-Sammlungsressourcen pro Konto	Jede unterstützte Region: 1 500	Ja	Die maximale Anzahl von Geofence-Sammlungsressourcen, die Sie pro Konto erstellen können.
Geofences pro Geofence-Sammlung	Jede unterstützte Region: 50 000	Nein	Die maximale Anzahl von Geofences, die Sie pro Geofence-Sammlung erstellen können.
Kartenressourcen pro Konto	Jede unterstützte Region: 40	Ja	Die maximale Anzahl von Kartenressourcen, die Sie pro Konto erstellen können.
Ortsindexressourcen pro Konto	Jede unterstützte Region: 40	Ja	Die maximale Anzahl von Ortsindexressourcen, die Sie pro Konto erstellen können.
Rate der AssociateTrackerConsumer API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von AssociateTrackerConsumer Anforderungen, die Sie pro Sekunde stellen können. Darüber

Name	Standard	Anpas	Beschreibung
			hinausgehende Anfragen werden gedrosselt.
Rate der BatchDeleteDevicePositionHistory API-Anforderungen	Jede unterstützte Region: 50 pro Sekunde	<u>Ja</u>	Die maximale Anzahl von BatchDeleteDevicePositionHistory Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der BatchDeleteGeofence API-Anforderungen	Jede unterstützte Region: 50 pro Sekunde	<u>Ja</u>	Die maximale Anzahl von BatchDeleteGeofence Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der BatchEvaluateGeofences API-Anforderungen	Jede unterstützte Region: 50 pro Sekunde	<u>Ja</u>	Die maximale Anzahl von BatchEvaluateGeofences Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der BatchGetDevicePosition API-Anforderungen	Jede unterstützte Region: 50 pro Sekunde	<u>Ja</u>	Die maximale Anzahl von BatchGetDevicePosition Anfragen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der BatchPutGeofence API-Anforderungen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von BatchPutGeofence Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der BatchUpdateDevicePosition API-Anforderungen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von BatchUpdateDevicePosition Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der CalculateRoute API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von CalculateRoute Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der CalculateRouteMatrix API-Anforderungen	Jede unterstützte Region: 5 pro Sekunde	Ja	Die maximale Anzahl von CalculateRouteMatrix Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der CreateGeofenceCollection API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von CreateGeofenceCollection Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der CreateKey API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von CreateKey Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der CreateMap API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von CreateMap Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der CreatePlaceIndex API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von CreatePlaceIndex Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der CreateRouteCalculator API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von CreateRouteCalculator Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der CreateTracker API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von CreateTracker Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der DeleteGeofenceCollection API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DeleteGeofenceCollection Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der DeleteKey API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DeleteKey Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der DeleteMap API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	<u>Ja</u>	Die maximale Anzahl von DeleteMap Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der DeletePlaceIndex API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	<u>Ja</u>	Die maximale Anzahl von DeletePlaceIndex Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der DeleteRouteCalculator API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	<u>Ja</u>	Die maximale Anzahl von DeleteRouteCalculator Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der DeleteTracker API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	<u>Ja</u>	Die maximale Anzahl von DeleteTracker Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der DescribeGeofenceCollection API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DescribeGeofenceCollection Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der DescribeKey API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DescribeKey Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der DescribeMap API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DescribeMap Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der DescribePlaceIndex API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DescribePlaceIndex Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der DescribeRouteCalculator API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DescribeRouteCalculator Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der DescribeTracker API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DescribeTracker Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der DisassociateTrackerConsumer API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von DisassociateTrackerConsumer Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der GetDevicePosition API-Anforderungen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von GetDevicePosition Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der GetDevicePositionHistory API-Anforderungen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von GetDevicePositionHistory Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der GetGeofence API-Anforderungen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von GetGeofence Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der GetMapGlyphs API-Anforderungen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von GetMapGlyphs Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der GetMapSprites API-Anforderungen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von GetMapSprites Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der GetMapStyleDescriptor API-Anforderungen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von GetMapStyleDescriptor Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der GetMapTile API-Anforderungen	Jede unterstützte Region: 500 pro Sekunde	Ja	Die maximale Anzahl von GetMapTile Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der GetPlace API-Anforderungen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von GetPlace Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der ListDevicePositions API-Anforderungen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von ListDevicePositions Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der ListGeofenceCollections API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von ListGeofenceCollections Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der ListGeofences API-Anforderungen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von ListGeofences Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der ListKeys API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von ListKeys Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der ListMaps API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von ListMaps Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der ListPlaceIndexes API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von ListPlaceIndexes Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der ListRouteCalculators API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von ListRouteCalculators Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der ListTagsForResource API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von ListTagsForResource Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der ListTrackerConsumers API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von ListTrackerConsumers Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der ListTrackers API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von ListTrackers Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der PutGeofence API-Anforderungen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von PutGeofence Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der SearchPlaceIndexForPosition API-Anforderungen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von SearchPlaceIndexForPosition Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der SearchPlaceIndexForSuggestions API-Anforderungen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von SearchPlaceIndexForSuggestions Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der SearchPlaceIndexForText API-Anforderungen	Jede unterstützte Region: 50 pro Sekunde	Ja	Die maximale Anzahl von SearchPlaceIndexForText Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der TagResource API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von TagResource Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der UntagResource API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von UntagResource Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der UpdateGeofenceCollection API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von UpdateGeofenceCollection Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der UpdateKey API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von UpdateKey Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der UpdateMap API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von UpdateMap Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der UpdatePlaceIndex API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von UpdatePlaceIndex Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Rate der UpdateRouteCalculator API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von UpdateRouteCalculator Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.

Name	Standard	Anpas	Beschreibung
Rate der UpdateTracker API-Anforderungen	Jede unterstützte Region: 10 pro Sekunde	Ja	Die maximale Anzahl von UpdateTracker Anforderungen, die Sie pro Sekunde stellen können. Darüber hinausgehende Anfragen werden gedrosselt.
Routenrechner-Ressourcen pro Konto	Jede unterstützte Region: 40	Ja	Die maximale Anzahl von Routenrechner-Ressourcen, die Sie pro Konto erstellen können.
Tracker-Verbraucher pro Tracker	Jede unterstützte Region: 5	Nein	Die maximale Anzahl der Geofence-Sammlungen, mit der die Tracker-Ressource verknüpft werden können.
Tracker-Ressourcen pro Konto	Jede unterstützte Region: 500	Ja	Die maximale Anzahl von Tracker-Ressourcen, die Sie pro Konto erstellen können.

 Note

Sie können Ihre Nutzung mit Cloudwatch anhand Ihrer Kontingente überwachen. Weitere Informationen finden Sie unter [Verwenden von CloudWatch zur Überwachung der Nutzung anhand von Kontingenten](#).

Verwalten Ihrer Amazon Location Service Quotas

Amazon Location Service ist in Service Quotas integriert, einen - AWS Service, mit dem Sie Ihre Kontingente von einem zentralen Ort aus anzeigen und verwalten können. Weitere Informationen zu Service Quotas finden Sie unter [Was sind Service Quotas](#) im Benutzerhandbuch für Service Quotas.

Service Quotas macht es einfach, den Wert Ihrer Amazon Location Service Quotas nachzuschlagen.

AWS Management Console

So zeigen Sie Amazon Location Service Quotas mit der Konsole an

1. Öffnen Sie die Service-Quotas-Konsole unter <https://console.aws.amazon.com/servicequotas/>.
2. Wählen Sie im Navigationsbereich AWS -Services.
3. Suchen Sie in der Liste AWS der Services nach Amazon Location und wählen Sie sie aus.

In der Liste Service Quotas wird der Name der Service Quota, der angewendete Wert (falls verfügbar) und das AWS -Standardkontingent angezeigt. Zudem wird angezeigt, ob der Kontingentwert anpassbar ist.

4. Wählen Sie den Kontingentnamen, um zusätzliche Informationen zu einem Service Quota anzuzeigen, z. B. seine Beschreibung.
5. (Optional) Um eine Kontingenterhöhung zu beantragen, wählen Sie das Kontingent, das Sie erhöhen möchten, und dann Request quota increase (Kontingenterhöhung beantragen) aus, geben Sie die erforderlichen Informationen ein, und wählen Sie dann Request (Beantragen) aus.

Weitere Informationen zur Verwendung der Konsole mit Service Quotas finden Sie im [Benutzerhandbuch zu Service Quotas](#). Informationen zur Erhöhung eines Kontingents finden Sie unter [Anfordern einer Kontingenterhöhung](#) im Benutzerhandbuch zu Service Quotas.

AWS CLI

So zeigen Sie Amazon Location Service Quotas mit der an AWS CLI

Führen Sie den folgenden Befehl aus, um die Standardkontingente für Amazon Location anzuzeigen.

```
aws service-quotas list-aws-default-service-quotas \
```

```
--query 'Quotas[*].  
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \  
--service-code geo \  
--output table
```

Weitere Informationen zum Arbeiten mit Service Quotas mithilfe der finden Sie in der AWS CLI-[Befehlsreferenz für Service Quotas AWS CLI](#). Informationen zum Beantragen einer Kontingenterhöhung finden Sie unter dem [request-service-quota-increase](#)-Befehl in der [AWS CLI -Befehlsreferenz](#).

Erste Schritte als Entwickler mit Amazon Location Service

Sie können Amazon Location Service verwenden, um Apps geografische Funktionen über viele verschiedene Formfaktoren und Systeme hinweg bereitzustellen, darunter Backend-Webservices, Webanwendungen und mobile Anwendungen. Es gibt viele Tools, die Sie beim Erstellen Ihrer Anwendungen unterstützen, darunter SDKs, Bibliotheken und Beispielcode.

Dieser Abschnitt enthält Informationen und Links, die Ihnen den Einstieg in Amazon Location erleichtern. Insbesondere die folgenden Themen enthalten Informationen, die für Sie am hilfreichsten sein können:

- [Szenarien und Anwendungsfälle](#) – Eine Liste von Entwicklungsszenarien und wie Amazon Location Service Ihnen dabei helfen kann, sie abzuschließen.
- [Amazon Location SDKs und Tools](#) – Die Software Development Kits (SDKs) und Bibliotheken, die Sie beim Programmieren mit Amazon Location unterstützen.
- [Amazon Location Service API Reference](#) – Ein Verweis auf die zentralen Amazon Location APIs, die mit dem AWS SDK geliefert werden.
- [Codebeispiele](#) – Dieser Abschnitt enthält Beispiele, die Ihnen beim Einstieg oder beim Hinzufügen von Funktionen zu Ihrer vorhandenen Anwendung helfen.
- [Schnellstart-Tutorial](#) – In diesem Tutorial erfahren Sie, wie Sie Ihre erste Anwendung erstellen. Es gibt Versionen des Tutorials zum Erstellen einer Webanwendung oder einer Android-basierten mobilen Anwendung.
- [Konzepte von Amazon Location Service](#) – In diesem Abschnitt dieses Handbuchs werden die grundlegenden Konzepte von Amazon Location beschrieben, einschließlich der Abschnitte zu Karten, Ortssuche, Routen und Geofences und Trackern.
- [Amplify](#) – Amplify ist eine Komplettlösung, die einen Großteil der Funktionen enthält, die zum Erstellen von Web- und mobilen Anwendungen mit der erforderlich sind AWS Cloud. Wenn Sie bereits Amplify verwenden oder Amplify verwenden möchten, verfügt es über eine Geobibliothek, die das integrierte Amazon Location Service verwendet und die Sie verwenden können. Informationen zu den ersten Schritten mit Amplify Geo finden Sie in der Dokumentation [hier](#).

Szenarien und Anwendungsfälle

Amazon Location Service ist ein Service, der in der ausgeführt wird AWS Cloud. Sie können sie von Ihren eigenen Amazon EC2-Instances in der Cloud aus aufrufen, aber viele

Zuweisungsanwendungen werden auf Geräten oder einer Kombination aus Geräten und der Cloud ausgeführt. Im Folgenden werden nur einige typische Szenarien aufgeführt und es wird beschrieben, wie Sie sie entwickeln können.

- Eine Backend-Anwendung, die Ihnen hilft, Routen für Treiber in Ihrer Flotte zu optimieren.

Sie können eine Anwendung schreiben, die auf [Amazon EC2](#) in der ausgeführt wird AWS Cloud , die den Amazon Location Service verwendet, um [Routenmatrizen als Eingabe für einen Routenoptimierer für Ihre Flotte zu berechnen](#). Verwenden Sie das [AWS SDK](#), um Amazon Location aufzurufen.

- Eine Webanwendung, mit der Ihre Kunden die Standorte Ihres Unternehmens finden können.

Sie können eine Website erstellen, die auf Amazon EC2-Instances ausgeführt wird, einschließlich einer standortbasierten Anwendung. Verwenden Sie das [AWS SDK für JavaScript](#) , um eine Webanwendung zu entwickeln, um Standorte mithilfe [der Ortssuche](#) nachzuschlagen und Ergebnisse mithilfe von auf einer [Karte](#) anzuzeigen MapLibre. Verwenden Sie das Amazon Location SDK, um die Programmierung mit einem -Standort zu vereinfachen.

- Fügen Sie einer vorhandenen iOS- oder Android-Anwendung Standortfunktionen hinzu.

Sie können das AWS SDK for Swift (iOS) oder [Kotlin](#) (Android) verwenden, um Amazon Location aufzurufen und Ihrer Anwendung [die Ortssuche hinzuzufügen und](#) ihr Funktionen zuzuordnen. Verwenden Sie MapLibre , um Karten zu rendern. Für andere Sprachen sind zusätzliche [AWS SDKs](#) verfügbar.

- Verfolgen Sie Komponenten (Geräte oder Fahrzeuge) und erhalten Sie Updates, wenn sie von Ihnen definierte Eingangs- oder Ausgangsbereiche sind.

Eine Anwendung zur Nachverfolgung von Geräten besteht aus mehreren Teilen.

- Für jedes Gerät, das Sie verfolgen, muss eine [Tracker](#)-Ressource erstellt werden, um sie zu verfolgen. Es muss Positionsaktualisierungen an Amazon Location Service senden, z. B. mithilfe von [MQTT](#) .
- Erstellen Sie [Geofences](#), um Bereiche zu definieren, in denen Sie Ein- und Ausgangsereignisse für Ihre Assets erhalten möchten.
- Sie können [Amazon EC2](#) oder verwenden [AWS Lambda](#), um auf Ihre Ereignisse zu reagieren, wenn Komponenten die Geofence-Bereiche betreten oder verlassen.
- Sie können dies erweitern, um Web- oder Geräteanwendungen zu erstellen, um Ihre Asset-Standorte auf Karten zu verfolgen und anzuzeigen.

Der folgende Abschnitt enthält Details zu Tools und Bibliotheken, die für jeden Aspekt des Amazon Location Service verfügbar sind.

SDKs und Tools für die Verwendung von Amazon Location Service

Es gibt mehrere Tools, die Ihnen bei der Verwendung von Amazon Location Service helfen.

- -AWS SDKs – Die AWS Software Development Kits (SDKs) sind in vielen gängigen Programmiersprachen verfügbar und bieten eine API, Codebeispiele und Dokumentation, die das Erstellen von Anwendungen in Ihrer bevorzugten Sprache erleichtern. Die AWS SDKs enthalten die wichtigsten Amazon Location APIs und Funktionen, einschließlich Zugriff auf Karten, Ortssuche, Routen, Geofence und Tracker. Weitere Informationen zu den SDKs, die für die Verwendung mit Amazon Location Service für verschiedene Anwendungen und Sprachen verfügbar sind, finden Sie unter [SDKs nach Sprache](#).
- MapLibre – Amazon Location Service empfiehlt, Karten mit der [MapLibre](#) Rendering-Engine zu rendern. MapLibre ist eine Engine zum Anzeigen von Karten in Web- oder mobilen Anwendungen. MapLibre verfügt auch über ein Plugin-Modell und unterstützt die Benutzeroberfläche für die Suche und Routen in einigen Sprachen und Plattformen. Weitere Informationen zur Verwendung MapLibre von und zu den darin enthaltenen Funktionen finden Sie unter [MapLibre](#).
- Amazon Location SDK – Das Amazon Location SDK besteht aus einer Reihe von Open-Source-Bibliotheken, die die Entwicklung von Anwendungen mit Amazon Location Service erleichtern. Die anfänglichen Bibliotheken bieten Funktionen in , JavaScript um die Authentifizierung und Konvertierung zwischen Amazon Location-Datentypen und [GeoJSON](#) zu unterstützen, sowie ein gehostetes Paket des Amazon Location-Clients für das AWS SDK v3. Weitere Informationen zum Amazon Location SDK finden Sie unter [Amazon Location SDK](#).

SDKs nach Sprache

Die folgenden Tabellen enthalten Informationen zu AWS SDKs und MapLibre Versionen für Sprachen und Frameworks nach Anwendungstyp: Web-, Mobil- oder Backend-Anwendung.

SDK-Versionen

Wir empfehlen Ihnen, den neuesten Build des AWS SDK und alle anderen SDKs zu verwenden, die Sie in Ihren Projekten verwenden, und die SDKs auf dem neuesten Stand zu halten. Das AWS SDK stellt Ihnen die neuesten Funktionen und Sicherheitsupdates

bereit. Den neuesten Build des AWS SDK für finden Sie JavaScriptbeispielsweise im [Thema Browserinstallation](#) in der Dokumentation zum AWS SDK für JavaScript .

Web frontend

Die folgenden AWS SDKs und MapLibre Versionen sind für die Entwicklung von Web-Frontend-Anwendungen verfügbar.

Sprache/Framework	AWS SDK	Rendering-Framework
Vollständig unterstützt		
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/
Teilweise unterstützt		
Flutter	https://docs.amplify.aws/start/q/integration/flutter/ Flutter wird von noch nicht vollständig unterstützt AWS, aber über Amplify wird eine begrenzte Unterstützung angeboten.	https://github.com/maplibre/flutter-maplibre-gl Die MapLibre Flutter-Bibliothek gilt als experimentell.
Node.js	https://aws.amazon.com/sdk-for-javascript/	Node.js wird nicht MapLibre unterstützt.
PHP	https://aws.amazon.com/sdk-for-php/	PHP wird nicht MapLibre unterstützt.

Mobile frontend

Die folgenden AWS SDKs und MapLibre Versionen sind für die Entwicklung mobiler Frontend-Anwendungen verfügbar.

Sprache/Framework	AWS SDK	Rendering-Framework
Vollständig unterstützt		
Java	https://aws.amazon.com/sdk-for-java/	https://maplibre.org/projects/maplibre-native/
Kotlin	https://aws.amazon.com/sdk-for-kotlin/ Amazon Location Service Mobile Authentication SDK für Android	https://maplibre.org/projects/maplibre-native/ Erfordert benutzerdefinierte Bindungen, ebenso wie MapLibre Java-basiert.
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/ Amazon Location Service Mobile Authentication SDK für iOS	https://maplibre.org/projects/maplibre-native/
Teilweise unterstützt		
Flutter	https://docs.amplify.aws/start/q/integration/flutter/ Flutter wird von noch nicht vollständig unterstützt AWS, aber über Amplify wird eine	https://github.com/maplibre/flutter-maplibre-gl Die MapLibre Flutter-Bibliothek gilt als experimentell.

Sprache/Framework	AWS SDK	Rendering-Framework
	begrenzte Unterstützung angeboten.	

Backend application

Die folgenden AWS SDKs sind für die Entwicklung von Backend-Anwendungen verfügbar. MapLibre ist hier nicht aufgeführt, da für Backend-Anwendungen normalerweise kein Karten-Rendering erforderlich ist.

Sprache	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/ Das AWS SDK für Rust befindet sich in der Entwicklervorschau.

Verwenden von MapLibre Tools und Bibliotheken mit Amazon Location

Eines der wichtigen Tools zum Erstellen interaktiver Anwendungen mit Amazon Location ist MapLibre. [MapLibre](#) ist in erster Linie eine Rendering-Engine zur Anzeige von Karten in einer Web- oder mobilen Anwendung. Es bietet jedoch auch Unterstützung für Plugins und Funktionen für die Arbeit mit anderen Aspekten von Amazon Location. Im Folgenden werden Tools beschrieben, die Sie verwenden können, basierend auf dem Standortbereich, mit dem Sie arbeiten möchten.

Note

Um einen beliebigen Aspekt von Amazon Location zu verwenden, installieren Sie das [AWS SDK für die Sprache, die Sie verwenden möchten](#).

- Karten

Um Karten in Ihrer Anwendung anzuzeigen, benötigen Sie eine Map-Rendering-Engine, die die von Amazon Location bereitgestellten Daten verwendet und zum Bildschirm zieht. Map-Rendering-Engines bieten auch Funktionen zum Verschieben und Zoomen der Karte oder zum Hinzufügen von Markierungen oder Pushpins und anderen Anmerkungen zur Karte.

Amazon Location Service empfiehlt, Karten mit der [MapLibre](#) Rendering-Engine zu rendern. MapLibre GL JS ist eine Engine zum Anzeigen von Karten in JavaScript, während MapLibre Native Karten entweder für iOS oder Android bereitstellt.

MapLibre verfügt auch über ein Plug-In-Ökosystem, um die Kernfunktionalität zu erweitern. Weitere Informationen finden Sie unter <https://maplibre.org/maplibre-gl-js-docs/plugins/>.

- Ortssuche

Um das Erstellen einer Suchbenutzeroberfläche zu vereinfachen, können Sie den [MapLibre Geocoder](#) für das Web verwenden (Android-Anwendungen können das [Android Places-Plug-In](#) verwenden).

- Routen

Um Routen auf der Karte anzuzeigen, verwenden Sie die [MapLibre Richtungen](#) .

- Geofences und Tracker

MapLibre verfügt über kein spezifisches Rendering oder Tools für Geofences und Tracking, aber Sie können die Rendering-Funktionalität und [Plug-Ins](#) verwenden, um die Geofences und verfolgten Geräte auf der Karte anzuzeigen.

Die nachverfolgten Geräte können [MQTT](#) verwenden oder Updates manuell an Amazon Location Service senden. Geofence-Ereignisse können mit beantwortet werden [AWS Lambda](#).

Viele Open-Source-Bibliotheken stehen zur Verfügung, um zusätzliche Funktionen für Amazon Location Service bereitzustellen, z. B. [Turf](#), die Geoanalysefunktionen bieten.

Viele Bibliotheken verwenden die offenen [GeoJSON](#)-Standarddaten. Amazon Location Service bietet eine Bibliothek zur Unterstützung der Verwendung von GeoJSON in JavaScript Anwendungen. Weitere Informationen finden Sie im folgenden Abschnitt, [Amazon Location SDK und Bibliotheken](#).

Amazon Location SDK und Bibliotheken

Das Amazon Location SDK besteht aus einer Reihe von Open-Source-Bibliotheken, die nützliche Funktionen für die Entwicklung von Amazon Location-Anwendungen bereitstellen. Die folgenden Funktionen sind enthalten:

- Amazon-Location-Client – Die Amazon-Location-Objekte im AWS SDK v3 sind gebündelt und verpackt, um die Verwendung in der Webentwicklung zu vereinfachen.
- Authentifizierung – Das Authentifizierungsprogramm vereinfacht die Authentifizierung (mit Amazon Cognito oder API-Schlüsseln) beim Erstellen einer Webseite oder JavaScript Anwendung für Amazon Location Service.
- Amazon Location GeoJSON-Funktionen – Die GeoJSON-Konvertierungsprogramme erleichtern die Konvertierung zwischen den in der Branche üblichen [GeoJSON](#)-formatierten Daten und den Amazon Location API-Formaten.

Themen

- [So beginnen Sie mit der Verwendung des Amazon Location SDK](#)
- [Amazon-Location-Client](#)
- [Authentifizierungshelfer](#)
- [GeoJSON-Konvertierungshelfer](#)

So beginnen Sie mit der Verwendung des Amazon Location SDK

Das Amazon Location SDK ist eine Reihe von Funktionen, die die Verwendung des Amazon Location Service in einer Anwendung vereinfachen können. Sie können diese Funktionen installieren und in Ihre JavaScript Anwendung importieren. In den folgenden Abschnitten werden der Amazon Location-Client sowie die Authentifizierungs- und GeoJSON-Hilfsbibliotheken beschrieben.

Amazon-Location-Client

Mit AWS SDK v3 ist das SDK durch den Service getrennt. Sie können nur die Teile installieren, die Sie benötigen. Verwenden Sie beispielsweise die folgenden Befehle, um den Amazon Location-Client und den Anmeldeinformationsanbieter für Amazon Cognito zu installieren.

```
npm install @aws-sdk/client-location
npm install @aws-sdk/credential-providers
```

Um die Verwendung von Amazon Location Service in JavaScript Web-Frontend-Anwendungen zu erleichtern, AWS stellt ein gehostetes Paket der Amazon-Location-Bibliothek und des Anmeldeinformationsanbieters bereit. Um den gebündelten Client zu verwenden, fügen Sie ihn wie folgt zu Ihrem HTML in einem Skript-Tag hinzu:

```
<script src="https://unpkg.com/@aws/amazon-location-client@1.x/dist/amazonLocationClient.js"></script>
```

Note

Das Paket wird aus Gründen der Benutzerfreundlichkeit auf dem neuesten Stand und abwärtskompatibel gehalten. Die Verwendung dieses Skript-Tags oder der NPM-Installation erhält immer die neueste Version.

Authentifizierungshelfer

Die Amazon Location Authentication Helpers erleichtern die Authentifizierung, wenn Amazon Location API-Aufrufe von Ihrer JavaScript Anwendung aus ausgeführt werden. Die Authentifizierungshelfer helfen Ihnen speziell bei der Verwendung von [Amazon Cognito](#) oder [API-Schlüsseln](#) als Authentifizierungsmethode. Dies ist eine Open-Source-Bibliothek, die auf verfügbar ist GitHub: <https://github.com/aws-geospatial/amazon-location-utilities-auth-helper-js>.

Note

Die Amazon Cognito-Unterstützung im Authentifizierungshelfer unterstützt nicht die Funktion für Verbundidentitäten von Amazon Cognito .

Installation

Sie können die Bibliotheken mit einer lokalen Installation verwenden, wenn Sie ein Build-System wie Webpack verwenden, oder indem Sie vorgefertigte JavaScript Pakete mit `<script>` Tags in Ihren HTML-Code aufnehmen.

- Verwenden Sie den folgenden Befehl, um die Bibliothek mit NPM zu installieren:

```
npm install @aws/amazon-location-utilities-auth-helper
```

- Verwenden Sie den folgenden Befehl in Ihrer HTML-Datei, um das Skript zu laden:

```
<script src="https://unpkg.com/@aws/amazon-location-utilities-auth-helper@1.x/dist/amazonLocationAuthHelper.js"></script>
```

Import

Um eine bestimmte Funktion in Ihrer JavaScript Anwendung zu verwenden, müssen Sie diese Funktion importieren. Der folgende Code wird verwendet, um die Funktion `withIdentityPoolId` in Ihre Anwendung zu importieren.

```
import { withIdentityPoolId } from '@aws/amazon-location-utilities-auth-helper';
```

Authentifizierungsfunktionen

Die Amazon Location Authentication Helpers enthalten die folgenden Funktionen, die ein `AuthHelper` Objekt zurückgeben:

- `async withIdentityPoolId(identityPoolId: string): AuthHelper` – Diese Funktion gibt ein `AuthHelper` Objekt zurück, das für die Arbeit mit Amazon Cognito initialisiert wurde
- `async withAPIKey(API_KEY: string): AuthHelper` – Diese Funktion gibt ein `AuthHelper` Objekt zurück, das für die Arbeit mit API-Schlüsseln initialisiert wurde.

Das `-AuthHelper` Objekt bietet die folgenden Funktionen:

- `AuthHelper.getMapAuthenticationOptions()` – Diese Funktion des `AuthHelper` Objekts gibt ein JavaScript Objekt mit der zurücktransformRequest, das mit den Kartenoptionen in MapLibre JS verwendet werden kann. Wird nur bei Initialisierung mit einem Identitäten-Pool bereitgestellt.
- `AuthHelper.getLocationClientConfig()` – Diese Funktion des `- AuthHelper` Objekts gibt ein JavaScript Objekt mit der zurückcredentials, das zur Initialisierung eines verwendet werden kann `LocationClient`.
- `AuthHelper.getCredentials()` – Diese Funktion des `AuthHelper` Objekts gibt die internen Anmeldeinformationen von Amazon Cognito zurück. Wird nur bei Initialisierung mit einem Identitäten-Pool bereitgestellt.

Beispiel: Initialisieren eines MapLibre Kartenobjekts mit Amazon Cognito unter Verwendung eines `AuthHelper`

```
import { withIdentityPoolId } from '@aws/amazon-location-utilities-auth-helper';

const authHelper = await withIdentityPoolId("identity-pool-id"); // use Cognito pool id
for credentials

const map = new maplibregl.Map({
  container: "map", // HTML element ID of map element
  center: [-123.1187, 49.2819], // initial map center point
  zoom: 16, // initial map zoom
  style: https://maps.geo.region.amazonaws.com/maps/v0/maps/mapName/style-
descriptor', // Defines the appearance of the map
  ...authHelper.getMapAuthenticationOptions(), // Provides credential options
  required for requests to Amazon Location
});
```

Beispiel: Initialisieren eines MapLibre Kartenobjekts mit einem API-Schlüssel (**AuthHelper** wird in diesem Fall nicht benötigt)

```
const map = new maplibregl.Map({
  container: "map", // HTML element ID of map element
  center: [-123.1187, 49.2819], // initial map center point
  zoom: 16, // initial map zoom
  style: https://maps.geo.region.amazonaws.com/maps/v0/maps/${mapName}/style-
descriptor?key=api-key-id',
```

```
});
```

Beispiel: Initialisieren Sie den Location-Client aus dem AWS SDK für JS mit Amazon Cognito und AuthHelper

In diesem Beispiel wird AWS SDK für JavaScript v3 verwendet.

```
import { withIdentityPoolId } from '@aws/amazon-location-utilities-auth-helper';

const authHelper = await withIdentityPoolId("identity-pool-id"); // use Cognito pool id
for credentials

//initialize the Location client:
const client = new LocationClient({
  region: "region",
  ...authHelper.getLocationClientConfig() // sets up the Location client to use the
  Cognito pool defined above
});

//call a search function with the location client:
const result = await client.send(new SearchPlaceIndexForPositionCommand({
  IndexName: "place-index", // Place index resource to use
  Position: [-123.1187, 49.2819], // position to search near
  MaxResults: 10 // number of results to return
}));
```

Beispiel: Initialisieren Sie den Location-Client aus dem AWS -SDK für JS mit einem API-Schlüssel und AuthHelper

In diesem Beispiel wird AWS SDK für JavaScript v3 verwendet.

```
import { withAPIKey } from '@aws/amazon-location-utilities-auth-helper';

const authHelper = await withAPIKey("api-key-id"); // use API Key id for credentials

//initialize the Location client:
const client = new LocationClient({
  region: "region",
  ...authHelper.getLocationClientConfig() // sets up the Location client to use the
  API Key defined above
});
```

```
//call a search function with the location client:
const result = await client.send(new SearchPlaceIndexForPositionCommand({
  IndexName: "place-index", // Place index resource to use
  Position: [-123.1187, 49.2819], // position to search near
  MaxResults: 10 // number of results to return
}));
```

Mobile Authentication Helpers

Die SDKs für die mobile Authentifizierung von Amazon Location Service sind Dienstprogramme, die Service-API-Aufrufe von Ihrer iOS- oder Android-Anwendung aus durchführen. Durch die Handhabung der Authentifizierung mit minimalem zusätzlichem Code und ohne Einführung neuer Abhängigkeiten ermöglichen die Bibliotheken eine schnellere Integration der Funktionen von Amazon Location Service in mobile Anwendungen. Die folgenden Dienstprogramme sind verfügbar:

- [Amazon Location Service Mobile Authentication SDK für iOS](#)
- [Amazon Location Service Mobile Authentication SDK für Android](#)

GeoJSON-Konvertierungshelfer

Die Amazon Location GeoJSON-Konvertierungshilfen bieten Tools zur Konvertierung von Amazon Location Service-Datentypen in das und aus dem branchenüblichen [GeoJSON](#)-Format. GeoJSON wird beispielsweise mit verwendet, MapLibre um geografische Daten auf der Karte zu rendern. Dies ist eine Open-Source-Bibliothek, die auf verfügbar ist GitHub: <https://github.com/aws-geospatial/amazon-location-utilities-datatypes-js>.

Installation

Sie können die Bibliotheken mit einer lokalen Installation verwenden, z. B. Webpack, oder indem Sie vorgefertigte JavaScript Pakete mit `<script>` Tags in Ihren HTML-Code aufnehmen.

- Verwenden Sie den folgenden Befehl, um die Bibliothek mit NPM zu installieren.

```
npm install @aws/amazon-location-utilities-datatypes
```

- Verwenden Sie den folgenden Befehl in Ihrer HTML-Datei, um das Skript zu laden:

```
<script src="https://unpkg.com/@aws/amazon-location-utilities-datatypes@1.x/dist/amazonLocationDataConverter.js"></script>
```

Import

Um eine bestimmte Funktion in Ihrer JavaScript Anwendung zu verwenden, müssen Sie diese Funktion importieren. Der folgende Code wird verwendet, um die Funktion `placeToFeatureCollection` in Ihre Anwendung zu importieren.

```
import { placeToFeatureCollection } from '@aws/amazon-location-utilities-datatypes';
```

GeoJSON-Konvertierungsfunktionen

Die Amazon Location GeoJSON-Konvertierungshelfer umfassen die folgenden Funktionen:

- `placeToFeatureCollection(place: GetPlaceResponse | searchPlaceIndexForPositionResponse | searchPlaceIndexForTextResponse, keepNull: boolean): Feature` – Diese Funktion konvertiert Antworten von den Ortssuchfunktionen in einen GeoJSON FeatureCollection mit mindestens 1 Punkt-Feature.
- `devicePositionToFeatureCollection(devicePositions: GetDevicePositionResponse | BatchGetDevicePositionResponse | GetDevicePositionHistoryResponse | ListDevicePositionsResponse, keepNull: boolean)` – Diese Funktion konvertiert Antworten von den Positionsfunktionen des Tracker-Geräts in einen GeoJSON FeatureCollection mit mindestens 1 Punkt-Feature.
- `routeToFeatureCollection(legs: CalculateRouteResponse): FeatureCollection` – Diese Funktion konvertiert Antworten von der Funktion zur Berechnung der Route in einen GeoJSON FeatureCollection -Code mit einem einzigen MultiStringLine Feature. Jeder Teil der Route wird durch einen LineString Eintrag im dargestellt MultiStringLine.
- `geofenceToFeatureCollection(geofences: GetGeofenceResponse | PutGeofenceRequest | BatchPutGeofenceRequest | ListGeofencesResponse): FeatureCollection` – Diese Funktion konvertiert die Anforderung oder Antwort von Geofence-Funktionen in einen GeoJSON FeatureCollection mit Polygon-Funktionen. Es kann Geofences sowohl in der Antwort als auch in der Anforderung konvertieren, sodass Sie Geofences auf einer Karte anzeigen können, bevor Sie sie mit `PutGeofence` oder hochladen `BatchPutGeofence`.

Diese Funktion konvertiert einen Kreis-Geofence in ein Feature mit einem ungefähren Polygon, hat aber auch die Eigenschaften „center“ und „radius“, um den Kreis-Geofence bei Bedarf neu zu erstellen (siehe nächste Funktion).

- `featureCollectionToGeofences(featureCollection: FeatureCollection): BatchPutGeofenceRequestEntry[]` – Diese Funktion konvertiert einen GeoJSON

FeatureCollection mit Polygon-Funktionen in ein Array von BatchPutGeofenceRequestEntry Objekten, sodass das Ergebnis verwendet werden kann, um eine Anforderung an zu erstellen BatchPutGeofence.

Wenn ein Feature in der die Eigenschaften „center“ und „radius“ FeatureCollection hat, wird es in einen Kreisgeofence-Anforderungseintrag konvertiert, wobei die Geometrie des Polygons ignoriert wird.

Beispiel: Konvertieren von Suchergebnissen in eine Punktebene in MapLibre

In diesem Beispiel wird AWS SDK für JavaScript v3 verwendet.

```
import { placeToFeatureCollection } from '@aws/amazon-location-utility-datatypes';

...

let map; // map here is an initialized MapLibre instance

const client = new LocationClient(config);
const input = { your_input };
const command = new searchPlaceIndexForTextCommand(input);
const response = await client.send(command);

// calling utility function to convert the response to GeoJSON
const featureCollection = placeToFeatureCollection(response);
map.addSource("search-result", featureCollection);
map.addLayer({
  id: "search-result",
  type: "circle",
  source: "search-result",
  paint: {
    "circle-radius": 6,
    "circle-color": "#B42222",
  },
});
```

Amazon Location APIs

Amazon Location Service bietet API-Operationen für den programmgesteuerten Zugriff auf die Standortfunktionalität. Dazu gehören APIs für Karten, Orte, Routen, Tracker, Geofences und das

Markieren Ihrer Ressourcen. Informationen zu den verfügbaren API-Aktionen finden Sie in der [API-Referenz zu Amazon Location Service](#).

Beispiele finden Sie im [Codebeispiele](#) Kapitel dieses Handbuchs.

Verwenden von Amazon Location mit einem AWS-SDK

AWS-Software Development Kits (SDKs) sind für viele gängige Programmiersprachen erhältlich. Jedes SDK bietet eine API, Codebeispiele und Dokumentation, die es Entwicklern erleichtert, AWS Anwendungen in ihrer bevorzugten Sprache zu erstellen.

Weitere Informationen zu den SDKs, die für die Verwendung mit Amazon Location Service nach Sprache verfügbar sind, finden Sie unter [SDKs nach Sprache](#) in diesem Handbuch.

SDK-Versionen

Wir empfehlen Ihnen, den neuesten Build des AWS SDK und alle anderen SDKs zu verwenden, die Sie in Ihren Projekten verwenden, und die SDKs auf dem neuesten Stand zu halten. Das AWS SDK stellt Ihnen die neuesten Funktionen und Sicherheitsupdates bereit. Den neuesten Build des AWS SDK für finden Sie JavaScriptbeispielsweise im [Thema Browserinstallation](#) in der Dokumentation zum AWS SDK für JavaScript.

Amazon Location API-Fehlermeldungsaktualisierungen

Ab dem 1. August 2023 ändert das Amazon Location-Team API-Fehlermeldungen, wie in den folgenden Tabellen beschrieben. Fehlercodes werden nicht geändert. Wenn Ihre Anwendungen von exakten Fehlermeldungszeichenfolgen abhängen, müssen Sie Ihre Anwendungen mit den neuen Zeichenfolgen aktualisieren. Wenden Sie sich an , um Hilfe bei Fragen oder Problemen zu erhaltenAWS Support.

Themen

- [Orte](#)
- [Zuordnungen](#)
- [Tracker](#)
- [Routen](#)
- [Metadaten](#)
- [Geofences](#)

Orte

Orte

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
500	InternalServerErrorException	Internal Server Exception	Internal server error. Try again later.
404	ResourceNotFoundException	resource <PlaceIndexName> not found, reason: <Reason> Resource '<PlaceIndexName>' not found placeIdx<PlaceIndexName> not found, reason: <Reason> no place index with name '%s' found	Place index not found: <PlaceIndexName>.
404	ResourceNotFoundException	place not found	Place not found: <PlaceId>.
400	ValidationException	PlaceIndex <PlaceIndexName> cannot be used for SearchPlaceIndexForSuggestions because it has IntendedUse <IntendedUse>	A place index with 'IntendedUse' set to Storage does not support 'SearchPlaceIndexForSuggestions' operation.
400	ValidationException	only one of 'BiasPosition' or 'FilterBBox' may be set	Only one of 'BiasPosition' or 'FilterBBox' may be set.
400	ValidationException	BiasPosition must have exactly 2 entries	'BiasPosition' must have exactly 2 entries.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	BiasPosition[0] must be between -180 and 180	'BiasPosition[0]' must be between -180 and 180.
400	ValidationException	BiasPosition[1] must be between -90 and 90	'BiasPosition[1]' must be between -90 and 90.
400	ValidationException	FilterBBox must have exactly 4 entries	'FilterBBox' must have exactly 4 entries.
400	ValidationException	FilterBBox[0] must be between -180 and 180	'FilterBBox[0]' must be between -180 and 180.
400	ValidationException	FilterBBox[1] must be between -90 and 90	'FilterBBox[1]' must be between -90 and 90.
400	ValidationException	FilterBBox[2] must be between -180 and 180	'FilterBBox[2]' must be between -180 and 180.
400	ValidationException	FilterBBox[3] must be between -90 and 90	'FilterBBox[3]' must be between -90 and 90.
400	ValidationException	FilterBBox must have more southwesterly point before more northeasterly point	'FilterBBox' must have more southwesterly position before more northeasterly position.
400	ValidationException	Position must have exactly 2 entries	'Position' must have exactly 2 entries.
400	ValidationException	Position[0] must be between -180 and 180	'Position[0]' must be between -180 and 180.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	Position[1] must be between -90 and 90	'Position[1]' must be between -90 and 90.
400	ValidationException	Language is not a valid BCP 47 language tag	'Language' must comply with the BCP 47 Language Tag standard, but was set to <GivenValue>. For more information, see https://wikipedia.org/wiki/IETF_language_tag .
400	ValidationException	'placeID' is invalid	'Placeld' must be a valid ID.
400	ValidationException	no customer account ID parameter found	'RequesterAccountID' is a required field.
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	unsupported price plan '<PricingPlan>'	'PricingPlan' must be set to RequestBasedUsage.
400	ValidationException	'DataSource' must be one of: Here, Esri	'DataSource' must be one of Esri, Grab, Here.
400	ValidationException	Grab is only supported in the ap-southeast-1 region	'DataSource' Grab must only be used in following regions: ap-southeast-1.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	'IntendedUse' and 'PricingPlan' must both be provided to update either property	'IntendedUse' and 'PricingPlan' must both be provided to update either attribute .
402	ServiceQuotaExceededException	Place resources per account exceeded quota limits. For more info, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/	Place index resources have exceeded the quota per account per region. For more information, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/ .
409	ConflictException	Resource already exists	Place index already exists: <PlaceIndexName>.

Zuordnungen

Zuordnungen

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
500	InternalServerErrorException	Internal Server Exception unable to find style template Error fetching style	Internal server error. Try again later.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
		was not able to serialize the map style file	
404	ResourceNotFoundException	Map not found	Map not found: <MapName>.
404	ResourceNotFoundException	Sprites are not supported for this resource	Sprite not found: <SpriteName>.
400	ValidationException	Resource name should be set	'MapName' is a required field.
400	ValidationException	Must provide a valid number for start and end of Range	Font Unicode range start and end numbers must both be provided.
400	ValidationException	Start of range is an invalid number: <StartValue>	Start of font Unicode range must be a valid number.
400	ValidationException	End of range is an invalid number: <StartValue>	End of font Unicode range must be a valid number.
400	ValidationException	End of range must be exactly 255 higher from start of range, difference found: <Difference>	The difference between the start and end of the font Unicode range must be exactly 255. Difference found: <Difference>.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	Start of range must be a multiple of 256, found <StartValue>	Start of font Unicode range must be a multiple of 256, but was set to: <StartValue>.
400	ValidationException	Request font is empty	'FontStack' is a required field.
400	ValidationException	Request font is not valid for the datasource <DataSource>	<FontStack> is not a supported font stack for data source <DataSource>. For more information about the list of supported font stacks, see https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapGlyphs.html .
400	ValidationException	Request font is not valid	<FontStack> is not a supported font stack for data source <DataSource>. For more information about the list of supported font stacks, see https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapGlyphs.html .

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	DataSource is invalid: <DataSource>	'DataSource' must be one of Esri, Grab, Here.
400	ValidationException	Request filename is empty	'FileName' is a required field.
400	ValidationException	Request filename is not valid	<SpriteFile> is not a supported sprite file name. For more information about the list of supported sprite file names, see https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html .
400	ValidationException	Filename is invalid: <FileName>	<SpriteFile> is not a supported sprite file name. For more information about the list of supported sprite file names, see https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html .

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	Filename is an invalid content type: <FileName>	<SpriteFile> is not a supported sprite file name. For more information about the list of supported sprite file names, see https://docs.aws.amazon.com/location/latest/APIReference/API_GetMapSprites.html .
400	ValidationException	Filename is invalid: <FileName>	'Filename' must not be empty.
400	ValidationException	y-coordinate part of 'Y' must be a valid integer	y- coordinate part of 'Y' must be an integer.
400	ValidationException	tile resolution part of 'Y' must be a valid integer followed by 'x'	Tile resolution part of 'Y' must be an integer followed by 'X'.
400	ValidationException	file type extension part of 'Y' must not be empty if a '.' is present	File type extension part of 'Y' must not be empty if a '.' is present.
400	ValidationException	'Z' must be a valid integer	'Z' must be an integer.
400	ValidationException	'X' must be a valid integer	'X' must be an integer.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	'Z' must not be less than minimum zoom of style '<Style>' (<MinimumValue>)	'Z' must not be less than minimum zoom of style <Style> (<MinimumValue>).
400	ValidationException	'Z' must not be greater than maximum zoom of style '<Style>' (<MaximumValue>)	'Z' must not be greater than maximum zoom of style Style (<MaximumValue>).
400	ValidationException	'Z' value not supported	'Z' must be between 0 and 63.
400	ValidationException	tile resolution part of 'Y' must be omitted because '<Style>' is a vector style	Tile resolution part of 'Y' must be omitted for style <Style>.
400	ValidationException	tile resolution part of 'Y' must be at least 1	Tile resolution part of 'Y' must be at least 1.
400	ValidationException	tile resolution part of 'Y' must not be greater than max resolution of style '<Style>' (<MaximumResolution>)	Tile resolution part of 'Y' must not be greater than maximum resolution of style <Style> (max <MaxResolution>).
400	ValidationException	file type extension part of 'Y' must be one of <SupportedFileFormats> (or may be omitted) for style '<Style>'	File type extension part of 'Y' must be one of <SupportedFileFormats> (or may be omitted) for style <Style>.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	file type extension part of 'Y' must be omitted for style '<Style>'	File type extension part of 'Y' must be omitted for style <Style>.
400	ValidationException	y-coordinate part of 'Y' must be an integer in the range 0..2^Zoom -1 (0..<MaxTileCoordinate>)	y-coordinate part of 'Y' must be an integer in the range 0..2^Zoom -1 (0..<MaxTileCoordinate>).
400	ValidationException	'DataSource' must be one of: Here, Esri	'DataSource' must be one of Esri, Grab, Here.
400	ValidationException	unsupported price plan '<PricingPlan>'	'PricingPlan' must be set to RequestBasedUsage.
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	Unsupported Map Style: <Style>	<Style> is not a supported map style. For more information about list of supported map styles, see https://docs.aws.amazon.com/location/latest/APIReference/API_MapConfiguration.html .

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
402	ServiceQuotaExceededException	Map resources per account exceeded quota limits. For more info, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/	Map resources have exceeded the quota per account per region. For more information, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/ .
409	ConflictException	Resource already exists	Map already exists: <MapName>.

Tracker

Tracker

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
500	InternalServerErrorException	Internal Server Exception internal server error unable to retrieve point from the storage unable to verify tracker Error processing List request	Internal server error. Try again later.
404	ResourceNotFoundException	tracker not found: <TrackerName>	Tracker not found: <TrackerName>.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
		Tracker with name <TrackerName> was not found	
404	ResourceNotFoundException	association not found: TrackerName <TrackerName>; and ConsumerArn <ConsumerArn >	Association between tracker <TrackerName> and consumer <ConsumerArn> is not found.
400	ValidationException	'ConsumerArn' must refer to a geofence collection resource	'ConsumerArn' must refer to a geofence collection resource.
400	ValidationException	'ConsumerArn' must refer to a resource in the same region as the tracker it is associated to	'ConsumerArn' must refer to a resource in the same region as the tracker it is associated with.
400	ValidationException	'ConsumerArn' must refer to a resource in the same AWS account as the tracker it is associated to	'ConsumerArn' must refer to a resource in the same AWS account as the tracker it is associated with.
400	ValidationException	'DataSource' must be one of: Here, Esri	'DataSource' must be one of Esri, Grab, Here.
400	ValidationException	Nothing to update.	At least one of the following fields must be set: 'Description', 'PositionFiltering'
400	ValidationException	Invalid token	'NextToken' must be a valid token.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	request.TrackerName not found on request	'TrackerName ' is a required field.
400	ValidationException	no deviceId parameter found	'DeviceId' is a required field.
400	ValidationException	unsupported price plan '<PricingPlan>'	'PricingPlan' must be set to RequestBasedUsage
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	provided start time is incorrect, should follow the format YYYY-MM-DDThh:mm:ss.sssZ“	'StartTimeInclusive' must follow the format YYYY-MM-DDThh:mm:ss.sssZ.
400	ValidationException	provided end time is incorrect, should follow the format YYYY-MM-DDThh:mm:ss.sssZ	'EndTimeExclusive' must follow the format YYYY-MM-DDThh:mm:ss.sssZ.
400	ValidationException	end time must be after start time	'EndTimeExclusive' must be after 'StartTimeInclusive'.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	invalid key state	KMS key must be a symmetric Customer Master Key (CMK). Invalid state found. For more information about how key state affects the use of a KMS key, see https://docs.aws.amazon.com/kms/latest/developerguide/key-state.html .
400	ValidationException	key not found	Invalid KMS key. '<KmsKeyId>' <KmsKeyIdValue> not found.
400	ValidationException	key is disabled	Symmetric Customer Master Key (CMK) must be enabled.
400	ValidationException	access denied	Symmetric Customer Master Key (CMK) must allow Amazon Location to create grants to its KMS key.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
402	ServiceQuotaExceededException	Tracker <TrackerName> may not have more than <Max> consumer associations	Tracker resource may not have more than <Max> consumer associations. For more information, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/ .
402	ServiceQuotaExceededException	Trackers per account exceeded quota limits. For more info, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/	Tracking resources have exceeded the quota per account per region. For more information, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/ .
409	ConflictException	association already exists: TrackerName <TrackerName>; and ConsumerArn <ConsumerArn>	An association already exists between tracker <TrackerName> and consumer <ConsumerArn>.
409	ConflictException	Tracker already exists: <TrackerName>	Tracker already exists: <TrackerName>.

Routen

Routen

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
500	InternalServerErrorException	Internal Server Exception	Internal server error. Try again later.
404	ResourceNotFoundException	Resource not found	Route calculator not found: <RouteCalculatorName>.
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	'DataSource' must be one of: Here, Esri, Grab	'DataSource' must be one of Esri, Grab, Here.
400	ValidationException	<PricingPlan> pricing plan is not supported	'PricingPlan' must be set to RequestBasedUsage
400	ValidationException	unsupported price plan '<PricingPlan>'	'PricingPlan' must be set to RequestBasedUsage
400	ValidationException	Grab is only supported in the ap-southeast-1 region	'DataSource' <DataSourceName> must only be used in following regions: ap-southeast-1.
400	ValidationException	PricingPlan must be 'RequestBasedUsage'	'PricingPlan' must be set to RequestBasedUsage.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	'DeparturePositions[0][0]' must be between -180 and 180	'DeparturePositions[0][0]' must be between -180 and 180.
400	ValidationException	'DeparturePositions[0][1]' must be between -90 and 90	'DeparturePositions[0][1]' must be between -90 and 90.
400	ValidationException	'DestinationPositions[0][0]' must be between -180 and 180	'DestinationPositions[0][0]' must be between -180 and 180.
400	ValidationException	'DestinationPositions[0][1]' must be between -90 and 90.	'DestinationPositions[0][1]' must be between -90 and 90
400	ValidationException	'DepartNow' may not be true if 'DepartureTime' is set	Only one of 'DepartNow' or 'DepartureTime' may be set.
400	ValidationException	'<TravelModeOption>' may not be set when 'TravelMode' has value <TravelModeOption>	'<TravelModeOption>' must not be set when 'TravelMode' has value <TravelModeOption>.
400	ValidationException	'CarModeOptions' may not be set when 'TravelMode' has value Walking	'CarModeOptions' must not be set when 'TravelMode' has value Walking.
400	ValidationException	'TruckModeOptions' may not be set when 'TravelMode' has value Walking	'TruckModeOptions' must not be set when 'TravelMode' has value Walking.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	'TruckModeOptions' may not be set when 'TravelMode' has value Car	'TruckModeOptions' must not be set when 'TravelMode' has value Car.
400	ValidationException	'CarModeOptions' may not be set when 'TravelMode' has value Truck	'CarModeOptions' must not be set when 'TravelMode' has value Truck.
400	ValidationException	At least one of [Height, Length, Width] must be set in 'TruckModeOptions.Dimensions'	At least one of the following attribute must be set in TruckModeOptions.Dimensions: Height, Length, Width.
400	ValidationException	At least one of [Total] must be set in 'TruckModeOptions.Weight'	At least one of the following attribute must be set in TruckModeOptions.Weight: Total.
400	ValidationException	'DeparturePositions' count must be 10 or less with DataSource set to Esri	'DeparturePositions' must have length at most 10 for 'DataSource' Esri.
400	ValidationException	'DestinationPositions' count must be 10 or less with DataSource set to Esri	'DestinationPositions' must have length at most 10 for 'DataSource' Esri.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	'DeparturePositions[0]' is more than 40km away from 'DestinationPositions[0]'	'DeparturePositions[0]' must not be more than 40 km away from 'DestinationPositions[0]'.
400	ValidationException	'DeparturePositions[0]' is more than 400km away from 'DestinationPositions[0]'	'DeparturePositions[0]' must not be more than 400 km away from 'DestinationPositions[0]'.
400	ValidationException	DeparturePositions[0] is contained within an unsupported region. Korea is not supported for CalculateRouteMatrix with the provider Esri.	DeparturePositions[0] is located in Korea, which is not supported when using CalculateRouteMatrix with data provider Esri.
400	ValidationException	'<HereTruckDimension>' must be between <Min> and <Max> <Unit>	'HereTruckDimension' must be between <Min> and <Max> <Unit>.
400	ValidationException	'WaypointPositions[0][0]' must be between -180 and 180	'WaypointPositions[0][0]' must be between -180 and 180.
400	ValidationException	'WaypointPositions[0][1]' must be between -90 and 90	'WaypointPositions[0][1]' must be between -90 and 90.
400	ValidationException	'WaypointPositions[1][0]' must be between -180 and 180	'WaypointPositions[1][0]' must be between -180 and 180.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	'WaypointPositions[1][1]' must be between -90 and 90	'WaypointPositions[1][1]' must be between -90 and 90.
400	ValidationException	No road segment could be matched for one or more coordinates within a radius (1km)	One or more provided positions are more than 1 km from the nearest road segment.
400	ValidationException	Some positions in the request are unreachable	Some positions in the request are unreachable.
400	ValidationException	Total distance between all waypoints must be not be greater than 40km for DataSource Esri when using TravelMode Walking	Total distance between all route positions must not be greater than 40 km for 'DataSource' Esri and 'TravelMode' Walking.
400	ValidationException	Total distance between all waypoints must be not be greater than 400km for DataSource Esri	Total distance between all route positions must not be greater than 400 km for 'DataSource' Esri.
400	ValidationException	Following positions in the request are unreachable: <UnreachablePositions>	The following positions are unreachable: <UnreachablePositions>.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	'DepartureTime' contains a badly-formatted timestamp	'DepartureTime' must follow the format YYYY-MM-DDThh:mm:ss.sssZ.
400	ValidationException	'TravelMode' <TravelMode> is not supported by <DataProvider>	'TravelMode' <TravelMode> not supported by data provider <DataProvider>.
400	ValidationException	'DeparturePositions' must be set	'DeparturePositions' must not be empty.
400	ValidationException	'DestinationPositions' must be set	'DestinationPositions' must not be empty.
400	ValidationException	Some inputs in the request are invalid	Some inputs in the request are invalid.
400	ValidationException	No route found between position <FirstPosition> and position <SecondPosition>	No route found between position <FirstPosition> and position <SecondPosition>.
400	ValidationException	No route found	No route found. For more information, see https://developer.amazon.com/documentation/routing-api/dev_guide/topics/notice.html .
400	ValidationException	No route found	No route found.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
402	ServiceQuotaExceededException	Route calculators per account exceeded quota limits. For more info, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/	Route calculator resources have exceeded the quota per account per region. For more information, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/ .
409	ConflictException	Resource already exists	Route calculator already exists: <RouteCalculatorName>.

Metadaten

Metadaten

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
500	InternalServerErrorException	Internal Server Error Error processing List request	Internal server error. Try again later.
404	ResourceNotFoundException	APIKey not found	Api key not found: <APIKeyName>.
404	ResourceNotFoundException	APIKeyID not found	ApiKeyId not found: <APIKeyID>.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	Either ExpireTime or NoExpiry must be provided	At least one of the following fields must be set: 'ExpireTime', 'NoExpiry'.
400	ValidationException	NoExpiry cannot be set to false if no ExpireTime is provided	'ExpireTime' must be set when 'NoExpiry' has value false.
400	ValidationException	ExpireTime cannot be set if NoExpiry is true	'ExpireTime' must not be set when 'NoExpiry' has value true.
400	ValidationException	Expire time '<ExpireTimeValue>' is not a valid time format	'ExpireTime' must follow the format YYYY-MM-DDThh:mm:ss.sssZ.
400	ValidationException	Expire time '<ExpireTimeValue>' cannot be in the past when creating a key	'ExpireTime' must not be in the past.
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	The API Key %s has been recently used and the requested update may impact current usage. Specify ForceUpdate=true to update the API Key configuration.	This update may cause some users to lose API access. Because this API Key has been used in the last 7 days, you must set 'ForceUpdate' to true to confirm this change.
400	ValidationException	Expire time '<ExpireTimeValue>' must not be more than 1 minute in the past	'ExpireTime' must not be more than 1 minute in the past.
400	ValidationException	Description, ExpireTime, NoExpiry and Restrictions can't all be empty	At least one of the following fields must be set: 'Description', 'ExpireTime', 'NoExpiry', 'Restrictions'.
400	ValidationException	API Key expired	'ApiKeyId' must not be expired.
409	ConflictException	API key named <APIKeyName> already exists	Api key already exists: <APIKeyName>.

Geofences

Geofences

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
500	InternalServerErrorException	<p>internal server error</p> <p>Internal server error</p> <p>Unsupported geofence geometry encountered</p> <p>geometry marshal error</p> <p>geometry load error</p> <p>unable to get geofence collection</p> <p>unable to delete geofences</p> <p>unable to retrieve geofence</p> <p>Error processing List request</p>	<p>Internal server error.</p> <p>Try again later.</p>
404	ResourceNotFoundException	<p>collection not found: <GeofenceCollectionName></p> <p><GeofenceCollectionName> geofence collection not found</p> <p>Resource not found error</p>	<p>Geofence Collection not found: <GeofenceCollectionName>.</p>

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
		no geofence with given name found	
400	ValidationException	unsupported price plan '<PricingPlan>'	'PricingPlan' must be set to RequestBasedUsage.
400	ValidationException	KMS key must be a symmetric CMK. Invalid usage type: <UsageType>	KMS key must be a symmetric Customer Master Key (CMK). Invalid usage type <UsageType>. For how to create a symmetric CMK, refer to https://docs.aws.amazon.com/kms/latest/developerguide/create-keys.html#create-symmetric-cmk .
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	PricingPlanDataSource cannot be updated without updating PricingPlan	'PricingPlan' must be provided to update 'PricingPlanDataSource'.
400	ValidationException	nothing to update	At least one of the following fields must be set: 'Description'

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	invalid key state	KMS key must be a symmetric Customer Master Key (CMK). Invalid state <InvalidState>. For more information about how key state affects the use of a KMS key, see https://docs.aws.amazon.com/kms/latest/developerguide/key-state.html .
400	ValidationException	key not found	Invalid KMS key. '<KmsKeyId>' <KmsKeyIdValue> not found.
400	ValidationException	key is disabled	Symmetric Customer Master Key (CMK) must be enabled.
400	ValidationException	access denied	Symmetric Customer Master Key (CMK) must allow Amazon Location to create grants to its KMS key.
400	ValidationException	duplicate geofence ID in batch	'GeofenceId' <DuplicatedGeofenceId> is duplicated in batch.
400	ValidationException	missing GeofenceId	'GeofenceId' must not be empty.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	Invalid token	'NextToken' must be a valid token.
400	ValidationException	Expired token	'NextToken' must not be expired.
400	ValidationException	Position[0] must be between -180 and 180	'Position[0]' must be between -180 and 180.
400	ValidationException	Position[1] must be between -90 and 90	'Position[1]' must be between -90 and 90.
400	ValidationException	radius must be less than or equal to 1000km	'Geometry.Circle.Radius' must be less than or equal to 1000km.
400	ValidationException	no geofence with given name found	Geofence not found: <CollectionName>.
400	ValidationException	Geometry must contain either a Circle or Polygon, not both	Only one of 'Circle' or 'Polygon' may be set within 'Geometry'.
400	ValidationException	Geometry must contain a Polygon or a Circle	One of 'Polygon' or 'Circle' must be set within 'Geometry'.
400	ValidationException	radius must be greater than 0m	'Geometry.Circle.Radius' must be greater than 0m.
400	ValidationException	empty polygon	'Geometry.Polygon' must not be empty.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	empty polygon ring	'Geometry.Polygon' must not be empty.
400	ValidationException	circle can not cross antimeridian	'Geometry.Circle' must not cross antimeridian. Cut it in two such that neither part's representation crosses the antimeridian.
400	ValidationException	polygon can not cross antimeridian	'Geometry.Polygon' must not cross antimeridian. Cut it in two such that neither part's representation crosses the antimeridian.
400	ValidationException	polygon can not have interior rings (holes), remove holes	'Geometry.Polygon' must not have interior rings (holes). For more information about interior rings see https://www.rfc-editor.org/rfc/rfc7946.html#appendix-A.3 .
400	ValidationException	polygon ring is not closed	'Geometry.Polygon' contains an open ring. Close the ring by ensuring the first and last positions are equal.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	polygon ring has more than 1000 vertices	'Geometry.Polygon' must not have more than 1000 vertices.
400	ValidationException	polygon ring has fewer than 4 positions	Number of vertices in 'Geometry.Polygon' must be greater or equal to 4.
400	ValidationException	invalid center	'Geometry.Circle.Center' must be a valid position (longitude/latitude pair).
400	ValidationException	radius must be greater than 0m	'Geometry.Circle.Radius' must be greater than 0 m.
400	ValidationException	longitude range should be between -180 and 180 degrees	Longitude must be between -180 and 180 degrees, but was set to <Provided Longitude>.
400	ValidationException	latitude range should be between -90 and 90 degrees	Latitude must be between -90 and 90 degrees, but was set to <Provided Longitude>.
400	ValidationException	polygon exterior ring is expected to be counter clockwise	'Geometry.Polygon' must be oriented counter-clockwise.

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	polygon interior ring should be clockwise oriented	'Geometry.Polygon' must be oriented clockwise.
400	ValidationException	radius must be less than or equal to 1000km	'Geometry.Circle.Radius' must be less than or equal to 1000 km.
400	ValidationException	timestamp.Parse() error	'SampleTime' must follow the format YYYY-MM-DDThh:mm:ss.sssZ.
400	ValidationException	invalid input	'SourceArn' must refer to a tracker resource.
400	ValidationException	arn: invalid prefix	'SourceArn' must be a valid ARN. For more information, see https://docs.aws.amazon.com/general/latest/gr/AWS-arns-and-namespaces.html .
400	ValidationException	arn: not enough sections	'SourceArn' must be a valid ARN. For more information, see https://docs.aws.amazon.com/general/latest/gr/AWS-arns-and-namespaces.html .

Fehlercode	Exception	Alte Fehlermeldung	Neue Fehlermeldung
400	ValidationException	invalid resource part	'SourceArn' must refer to a tracker resource.
402	ServiceQuotaExceededException	Geofence collections per account exceeded quota limits. For more info, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/	Geofence collection resources have exceeded the quota per account per region. For more information, see https://aws.amazon.com/premiumsupport/knowledge-center/manage-service-limits/ .
409	ConflictException	collection already exists: <Geofence CollectionName>	Geofence Collection already exists: <GeofenceCollectionName>.
409	ConflictException	Resource conflict error	Geofence already exists: <Geofence Name>.

Codebeispiele und Tutorials für die Arbeit mit Amazon Location Service

Dieses Thema enthält eine Liste von Codebeispielen, Tutorials und Blogbeiträgen, die Ihnen helfen, mehr über Amazon Location Service zu erfahren. Jedes Codebeispiel enthält eine Beschreibung seiner Funktionsweise.

Weitere Beispiele finden Sie auf der [AWS GitHub Geodatenseite](#), auf der [AWS GitHub Beispielseite für Amazon Location](#) und auf der [AWS Blog-Website](#).

Note

Es ist gut, den Unterschied zwischen der AWS GitHub Geodatenseite und der AWS GitHub Beispielseite zu verstehen.

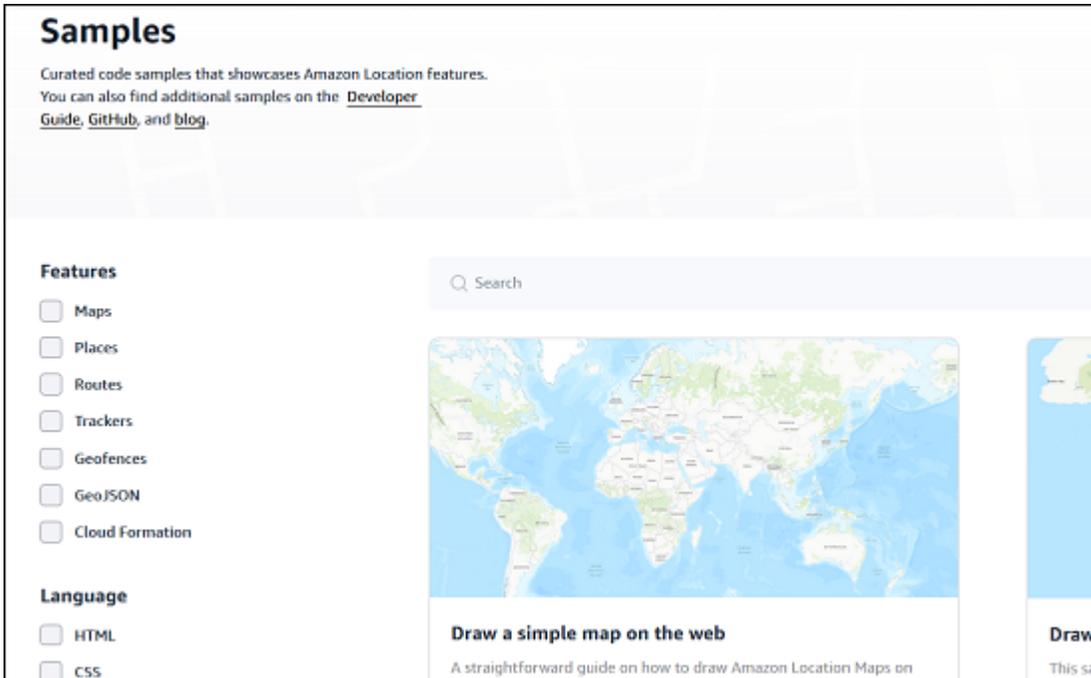
- Geodaten GitHub – Die [AWS GitHub Geodatenseite](#) enthält Beispiele, die vom Amazon Location Service-Team erstellt und verwaltet werden.
- Beispiele GitHub – Die [AWS GitHub Seite „Beispiele“ für Amazon Location](#) enthält Beispiele, die für Amazon Location erstellt wurden, aber aktiv verwaltet werden können oder nicht.

Das [Schnellstart](#)-Tutorial ist ein guter Ausgangspunkt, bevor Sie andere Beispiele verwenden, da es zeigt, wie die Voraussetzungen erfüllt werden, die für die meisten Beispiele nützlich sind.

Themen

- [Amazon-Standort-Demo-Website](#)
- [Tutorial: Schnellstart](#)
- [Tutorial: Datenbankanreicherung](#)
- [Beispiel: Erkunden der App](#)
- [Beispiel: Entwerfen einer Karte](#)
- [Beispiel: Grafikmarkierungen](#)
- [Beispiel: Zeichnen von gruppierten Punkten](#)
- [Beispiel: Zeichnen eines Polygons](#)
- [Beispiel: Ändern der Kartensprache](#)
- [Blog: Geschätzte Zustellungszeit-Benachrichtigungen](#)

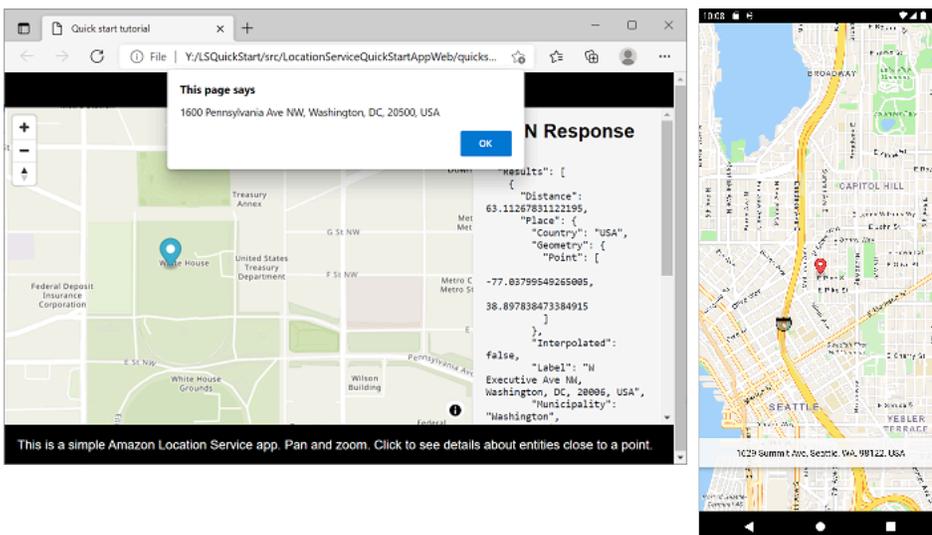
Amazon-Standort-Demo-Website



Demos mit dem Quellcode von Amazon Location Service in Aktion finden Sie auf der [Amazon Location Demo Site](#) . Diese Website enthält eine [gehostete Web-Demo](#) sowie eine Demo-App für [Android](#) .

Auf der Seite [Beispiele](#) der Website finden Sie auch eine Vielzahl von Beispielen, die nach Funktionen, Sprache und Plattform gefiltert werden können.

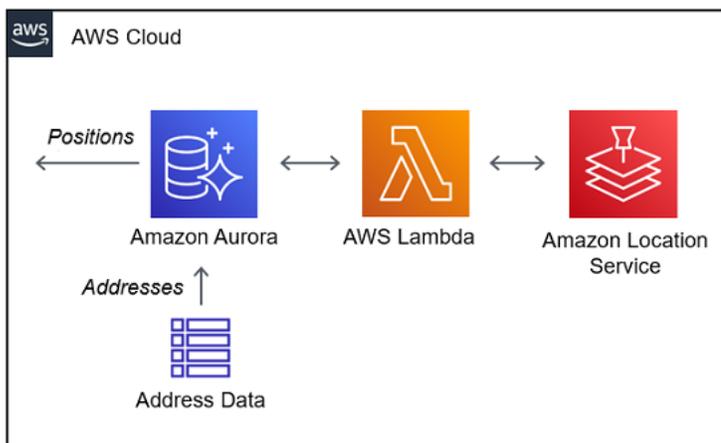
Tutorial: Schnellstart



Für Web-, iOS- und Android-Geräte sind Schnellstart-Tutorials verfügbar. Für jede Plattform zeigt Ihnen das Tutorial, wie Sie einer Anwendung eine interaktive Karte hinzufügen und Aufrufe an die Amazon Location Service-APIs von Ihrer Anwendung aus tätigen. Das Tutorial ist für JavaScript in einer statischen Webseite, Kotlin für eine Android-Telefonanwendung oder Swift für eine iOS-Anwendung verfügbar.

- JavaScript für einen statischen Link zur Webseitendokumentation: [Erstellen einer Webanwendung](#)
- Dokumentationslink für Kotlin für eine Android-Anwendung: [Schnellstart mit Amazon Location Service](#)
- Swift für einen iOS-App-Dokumentationslink: [Erstellen einer iOS-App](#)

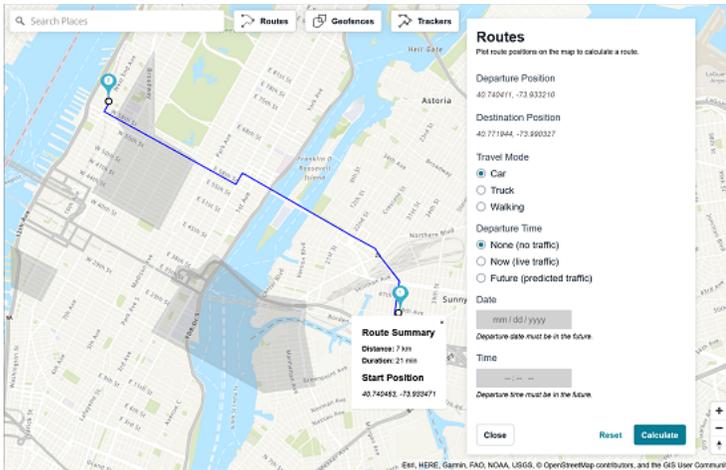
Tutorial: Datenbankanreicherung



Dieses Tutorial zeigt Ihnen, wie Sie den Amazon Location Service verwenden, der von aufgerufen wird, AWS Lambda um Adressen zu normalisieren und Datensätzen in einer Amazon-Aurora-Datenbank Breiten- und Längengrad hinzuzufügen. Verwendet Amazon Aurora und AWS Lambda.

Link zur Dokumentation: [Amazon Aurora PostgreSQL Benutzerdefinierte Funktionen für Amazon Location Service](#)

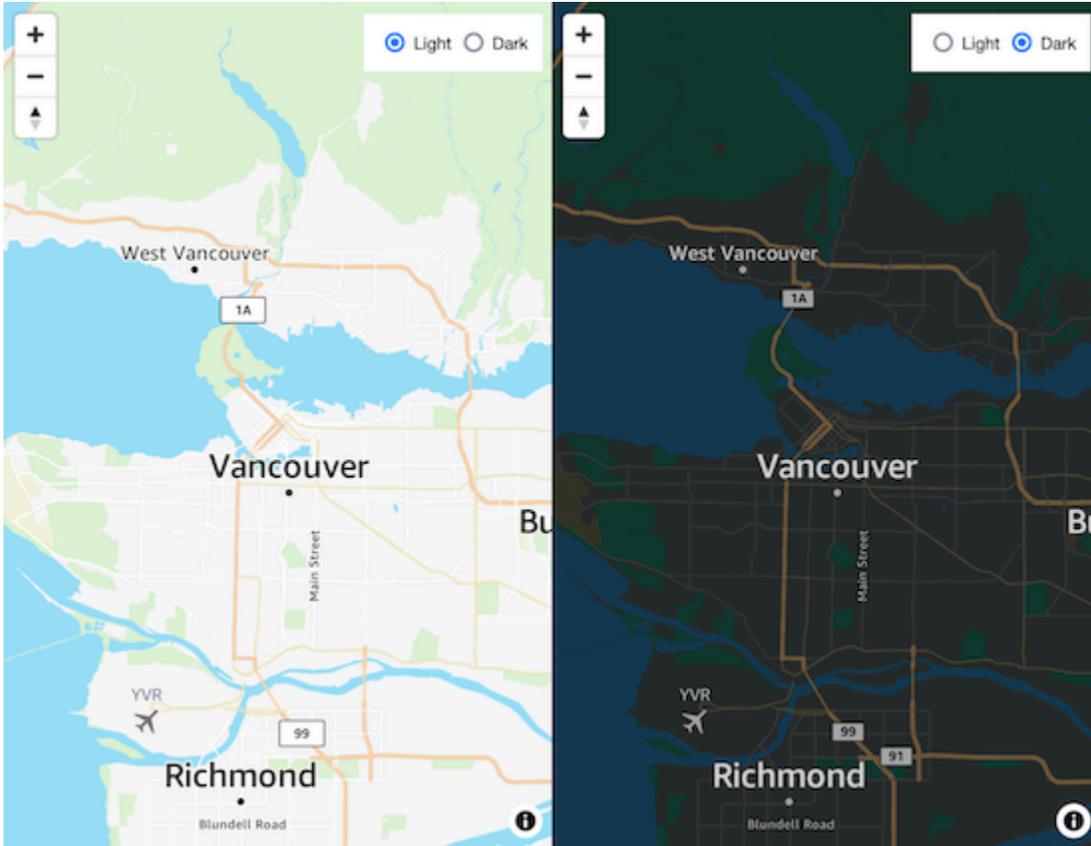
Beispiel: Erkunden der App



Eine der besten Möglichkeiten, mehr über die Funktionalität von Amazon Location Service zu erfahren, besteht darin, die [Funktionalität Erkunden](#) in der Amazon-Location-Konsole zu verwenden. Dieses vollständige Webanwendungsbeispiel ahmt die Karten, Orte, Routen, Geofences und Tracker-Funktionalität von der Konsole aus nach, um Ihnen zu zeigen, wie Sie diese Funktionen in Ihrer eigenen App neu erstellen können. Verwendet Amplify, React und JavaScript.

GitHub Beispiellink: [Erkunden einer Beispielanwendung](#)

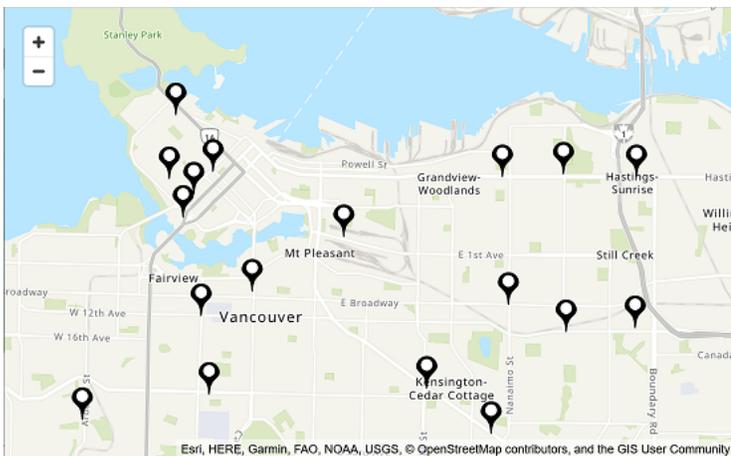
Beispiel: Entwerfen einer Karte



Dieses Codebeispiel zeigt, wie Sie mit MapLibre in zwischen einer Satellitenkarte und einer Vektor-Roadmap wechseln JavaScript. Verwendet MapLibre, das Amazon Location Authentication Helper und JavaScript.

GitHub Geodatenlink: [Interaktive Karte mit Stilwechsel](#)

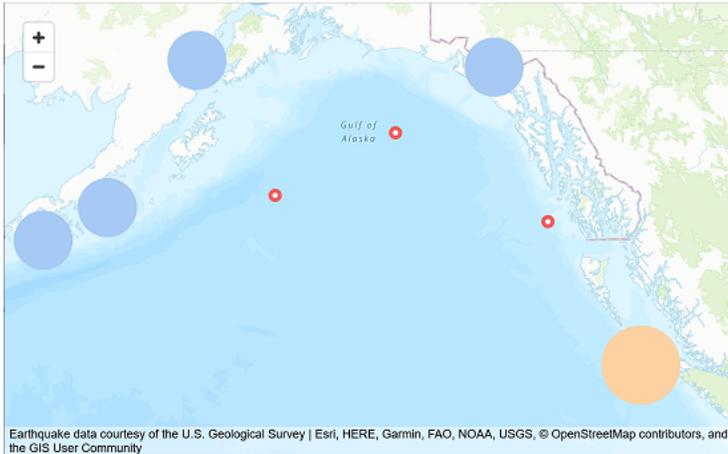
Beispiel: Grafikmarkierungen



Dieses Codebeispiel zeigt Amazon Locker-Standorte in Bol, BC, Kanada. Es zeigt, wie Sie Markierungen an Punktpositionen zeichnen. Verwendet MapLibre, Node.js, React, das Amazon Location Authentication Helper und JavaScript.

GitHub Geodatenlink: [Interaktive Karte mit Markierungen an Punkten](#)

Beispiel: Zeichnen von gruppierten Punkten



Anhand von USGS-Erdbebendaten zeigt dieses Codebeispiel, wie Punkte, die sich in der Nähe auf der Karte befinden, zusammengeführt werden. Verwendet MapLibre, Node.js, React, Amplify und JavaScript.

GitHub Link für Beispiele: [Interaktive Karte mit Punktclustern](#)

Beispiel: Zeichnen eines Polygons



Dieses Codebeispiel zeigt, wie Sie ein Polygon auf der Karte zeichnen. Verwendet MapLibre, Node.js, React, das Amazon Location Authentication Helper und JavaScript.

GitHub Geodatenlink: [Interaktive Karte mit Polygonen](#)

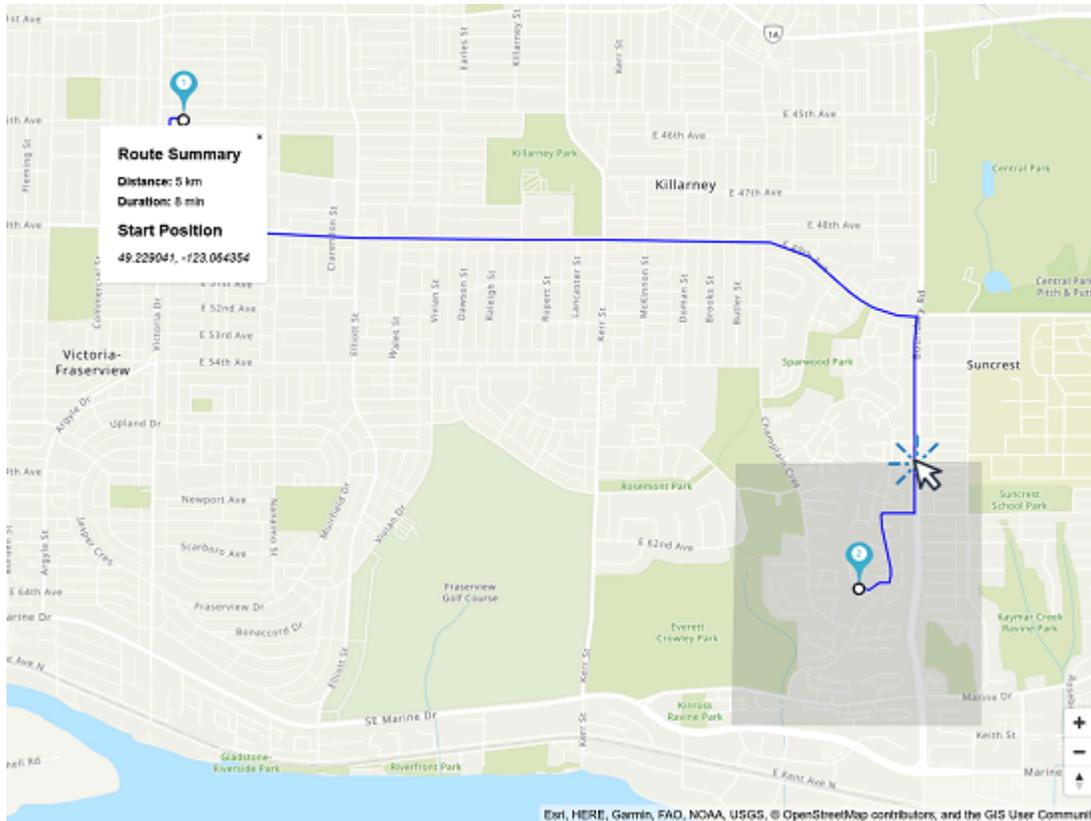
Beispiel: Ändern der Kartensprache



Dieses Codebeispiel zeigt, wie Sie die Anzeigesprache von Karten in Amazon Location ändern können. Verwendet Amplify, React und MapLibre.

GitHub Link für Beispiele: [Beispiel für eine Zuordnungssprache ändern](#)

Blog: Geschätzte Zustellungszeit-Benachrichtigungen



Dieser Blogbeitrag zeigt verschiedene Möglichkeiten, Kunden über geschätzte Lieferzeiten zu informieren. Es erläutert die Verwendung von Routen, um die geschätzte Verkehrszeit anzuzeigen, und die Verwendung von Trackern und Geofences, um zu benachrichtigen, wenn ein Treiber in der Nähe des Kunden ist. Verwendet Amplify, React, Amazon EventBridge und Amazon Simple Notification Service (Amazon SNS).

Blog-Link: [Geschätzte Zeit der Benachrichtigungen über Zeitzonen und Nähe](#)

So verwenden Sie Amazon Location Service

Sie können die Funktionen von Amazon Location Service verwenden, um geografische und standortbezogene Aufgaben auszuführen. Sie können diese Aufgaben dann kombinieren, um komplexere Anwendungsfälle wie GeoMarketing, Bereitstellung und Komponentenverfolgung zu bewältigen.

Wenn Sie bereit sind, Standortfunktionen in Ihre Anwendung zu integrieren, verwenden Sie die folgenden Methoden, um die Amazon Location Service-Funktionalität zu verwenden, abhängig von Ihren Zielen und Einschränkungen:

- **Nutzungstools** – Wenn Sie mit Amazon Location-Ressourcen experimentieren möchten, sind die folgenden Tools die schnellste Möglichkeit, auf die APIs zuzugreifen und diese auszuprobieren:
 - Die [Amazon Location-Konsole](#) bietet eine Vielzahl von Quick-Access-Tools. Sie können Ihre -Ressourcen erstellen und verwalten und die APIs auf [der Seite Erkunden](#) ausprobieren. Die Konsole ist auch nützlich, um Ressourcen (in der Regel eine einmalige Aufgabe) zu erstellen, um die Verwendung einer der anderen später beschriebenen Methoden vorzubereiten.
 - Mit der [-AWSBefehlszeilenschnittstelle](#) (CLI) können Sie Ressourcen erstellen und über ein Terminal auf die Amazon Location APIs zugreifen. Die AWS CLI übernimmt die Authentifizierung, wenn Sie sie mit Ihren -Anmeldeinformationen konfigurieren.
 - Sie können [Codebeispiele und Tutorials](#) sehen, die zeigen, wie Aufgaben mit den Amazon Location Service APIs ausgeführt werden. Dazu gehört [ein Beispiel](#), das einen Großteil der Funktionalität der Seite Erkunden in der -Konsole nachahmt.
- **Plattform-SDKs** – Wenn Sie keine Daten auf einer Karte visualisieren, können Sie jedes der [AWS Standard-Tools](#) verwenden, um auf zu bauenAWS.
 - Die folgenden SDKs sind verfügbar: C++, Go, Java JavaScript, .NET, Node.js, PHP, Python und Ruby.
- **Frontend-SDKs und Bibliotheken** – Wenn Sie Amazon Location verwenden möchten, um eine Anwendung auf einer mobilen Plattform zu erstellen oder Daten auf einer Karte auf einer beliebigen Plattform zu visualisieren, haben Sie die folgenden Optionen:
 - Die AWS Amplify Bibliotheken integrieren Amazon Location in [iOS-](#), [Android-](#) und -[JavaScript](#)Webanwendungen.
 - Mit den MapLibre Bibliotheken können Sie clientseitige Zuordnungen in [iOS-](#), [Android-](#) und [JavaScript](#) Webanwendungen mithilfe von Amazon Location rendern.

- Mit Tangram ES-Bibliotheken können Sie 2D- und 3D-Zuweisungen aus Vektordaten mithilfe von OpenGL ES in [iOS](#)- und [Android](#)-Webanwendungen rendern. Es gibt auch Tangram für [JavaScript](#) Webanwendungen.
- Senden von direkten HTTPS-Anfragen – Wenn Sie mit einer Programmiersprache arbeiten, für die kein SDK verfügbar ist, oder wenn Sie mehr Kontrolle darüber haben möchten, wie Sie eine Anfrage an sendenAWS, können Sie auf Amazon Location zugreifen, indem Sie direkte HTTPS-Anfragen senden, die durch den Signature Version 4-Signaturprozess authentifiziert wurden. Weitere Informationen zum [Signaturprozess mit Signature Version 4](#) finden Sie unter Allgemeine AWS-Referenz.

In diesem Kapitel werden viele Aufgaben beschrieben, die Anwendungen unter Verwendung von Standortdaten gemeinsam sind. Im Abschnitt [Häufige Anwendungsfälle](#) wird beschrieben, wie Sie diese mit anderen -AWSServices kombinieren, um komplexere Anwendungsfälle zu erzielen.

Themen

- [Voraussetzungen für die Verwendung von Amazon Location Service](#)
- [Verwenden von Amazon Location Maps in Ihrer Anwendung](#)
- [Suchen von Orts- und Geolocation-Daten mit Amazon Location](#)
- [Berechnen von Routen mit Amazon Location Service](#)
- [Geofencing eines Interessengebiets mit Amazon Location](#)
- [Markieren Ihrer Amazon Location Service-Ressourcen](#)
- [Gewähren des Zugriffs auf Amazon Location Service](#)
- [Überwachen von Amazon Location Service](#)
- [Erstellen von Amazon Location Service-Ressourcen mit AWS CloudFormation](#)

Voraussetzungen für die Verwendung von Amazon Location Service

In diesem Abschnitt wird beschrieben, wie Sie Amazon Location Service verwenden müssen. Sie müssen über ein verfügbares AWS-Konto und den Zugriff auf Amazon Location für Benutzer eingerichtet haben, die es verwenden möchten.

So melden Sie sich für ein AWS-Konto an

Wenn Sie kein AWS-Konto haben, führen Sie die folgenden Schritte zum Erstellen durch.

Anmeldung für ein AWS-Konto

1. Öffnen Sie <https://portal.aws.amazon.com/billing/signup>.
2. Folgen Sie den Online-Anweisungen.

Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

Wenn Sie sich für ein AWS-Konto anmelden, wird ein Root-Benutzer des AWS-Kontos erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem [Administratorbenutzer Administratorzugriff](#) zu und verwenden Sie nur den Root-Benutzer, um [Aufgaben auszuführen, die Root-Benutzerzugriff](#) erfordern.

AWS sendet Ihnen eine Bestätigungs-E-Mail, sobald die Anmeldung abgeschlossen ist. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu <https://aws.amazon.com/> auf und klicken Sie auf Mein Konto.

Erstellen eines Administratorbenutzers

Nachdem Sie sich für ein AWS-Konto angemeldet haben, sichern Sie Ihr Root-Benutzer des AWS-Kontos, aktivieren Sie AWS IAM Identity Center und erstellen Sie einen administrativen Benutzer, damit Sie nicht den Root-Benutzer für alltägliche Aufgaben verwenden.

Schützen Ihres Root-Benutzer des AWS-Kontos

1. Melden Sie sich bei der [AWS Management Console](#) als Kontobesitzer an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto-E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter [Anmelden als Root-Benutzer](#) im AWS-AnmeldungBenutzerhandbuch zu .

2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

Anweisungen dazu finden Sie unter [Aktivieren eines virtuellen MFA-Geräts für den Root-Benutzer Ihres AWS-Konto \(Konsole\)](#) im IAM-Benutzerhandbuch.

Erstellen eines Administratorbenutzers

1. Aktivieren von IAM Identity Center.

Anweisungen finden Sie unter [Aktivieren AWS IAM Identity Center](#) im AWS IAM Identity Center Benutzerhandbuch.

2. Im IAM Identity Center gewähren Sie einem administrativen Benutzer administrativen Zugriff.

Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie unter [Benutzerzugriff mit dem standardmäßigen IAM-Identity-Center-Verzeichnis konfigurieren](#) im AWS IAM Identity Center-Benutzerhandbuch.

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

Hilfe bei der Anmeldung mit einem IAM-Identity-Center-Benutzer finden Sie unter [Anmelden beim AWS-Zugangsportale](#) im AWS-Anmeldung Benutzerhandbuch zu.

Zugriff auf Amazon Location Service gewähren

Ihre Nicht-Admin-Benutzer haben standardmäßig keine Berechtigungen. Bevor sie auf Amazon Location zugreifen können, müssen Sie die Berechtigung erteilen, indem Sie eine IAM-Richtlinie mit bestimmten Berechtigungen anfügen. Befolgen Sie unbedingt das Prinzip der geringsten Berechtigung, wenn Sie Zugriff auf -Ressourcen gewähren.

Note

Informationen zum Gewähren von Zugriff auf die Funktionalität von Amazon Location Service für nicht authentifizierte Benutzer (z. B. in einer webbasierten Anwendung) finden Sie unter [Gewähren des Zugriffs auf Amazon Location Service](#).

Die folgende Beispielrichtlinie erteilt einem Benutzer die Berechtigung, auf alle Amazon Location-Operationen zuzugreifen. Weitere Beispiele finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Location Service](#).

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

Um Zugriff zu gewähren, fügen Sie Ihren Benutzern, Gruppen oder Rollen Berechtigungen hinzu:

- Benutzer und Gruppen in AWS IAM Identity Center:

Erstellen Sie einen Berechtigungssatz. Befolgen Sie die Anweisungen unter [Erstellen eines Berechtigungssatzes](#) im AWS IAM Identity Center-Benutzerhandbuch.

- Benutzer, die in IAM über einen Identitätsanbieter verwaltet werden:

Erstellen Sie eine Rolle für den Identitätsverbund. Befolgen Sie die Anweisungen unter [Erstellen einer Rolle für einen externen Identitätsanbieter \(Verbund\)](#) im IAM-Benutzerhandbuch.

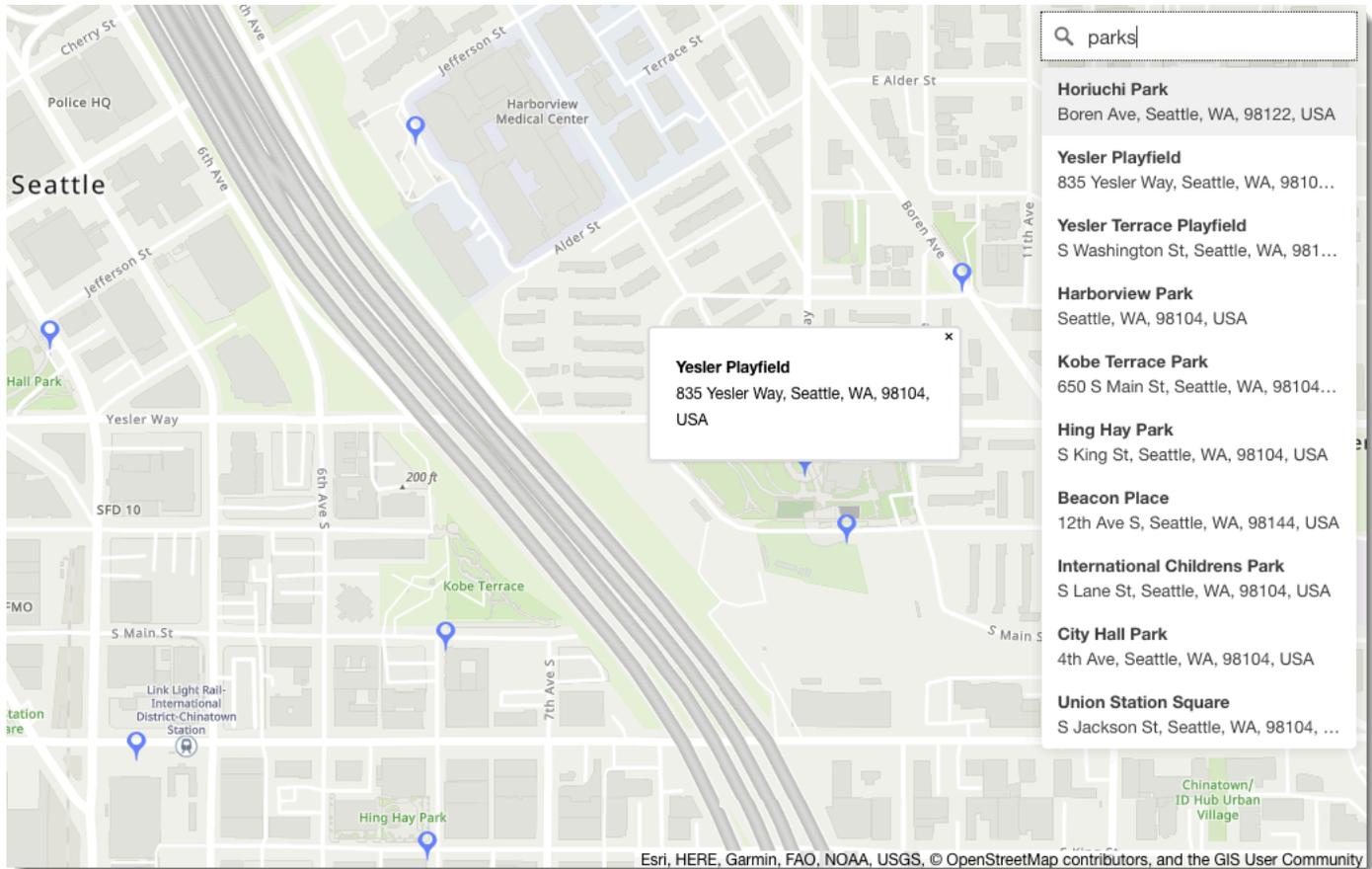
- IAM-Benutzer:

- Erstellen Sie eine Rolle, die Ihr Benutzer annehmen kann. Folgen Sie den Anweisungen unter [Erstellen einer Rolle für einen IAM-Benutzer](#) im IAM-Benutzerhandbuch.
- (Nicht empfohlen) Weisen Sie einem Benutzer eine Richtlinie direkt zu oder fügen Sie einen Benutzer zu einer Benutzergruppe hinzu. Befolgen Sie die Anweisungen unter [Hinzufügen von Berechtigungen zu einem Benutzer \(Konsole\)](#) im IAM-Benutzerhandbuch.

Wenn Sie Anwendungen erstellen, die Amazon Location Service verwenden, benötigen einige Benutzer möglicherweise nicht authentifizierten Zugriff. Informationen zu diesen Anwendungsfällen finden Sie unter [Aktivieren des nicht authentifizierten Zugriffs mit Amazon Cognito](#).

Verwenden von Amazon Location Maps in Ihrer Anwendung

Amazon Location Maps sind kostengünstig und interaktiv. Sie können eine vorhandene Karte in Ihrer Anwendung ersetzen, um Geld zu sparen, oder eine neue hinzufügen, um standortbasierte Daten visuell anzuzeigen, z. B. Ihren Standort.



Mit Amazon Location Service können Sie einen Datenanbieter für Kartenoperationen auswählen, indem Sie eine Kartenressource erstellen und konfigurieren. Die Kartenressource konfiguriert den Datenanbieter und den Stil, der zum Rendern der Karte verwendet wird.

Nachdem Sie Ihre Ressource erstellt haben, können Sie Anfragen direkt über das AWS SDK oder mithilfe einer Bibliothek senden, die speziell für das Rendern von Karten in Ihrer Umgebung erstellt wurde.

Note

Eine Übersicht über Kartenkonzepte finden Sie unter [Zuordnungen](#).

Themen

- [Voraussetzungen](#)
- [Anzeigen einer Karte in Ihrer Anwendung](#)
- [Zeichnen von Datenfunktionen auf einer Karte](#)
- [Festlegen von Erweiterungen für eine Zuordnung mit MapLibre](#)
- [Verwalten Ihrer Kartenressourcen](#)

Voraussetzungen

Bevor Sie eine Zuordnung in Ihrer Anwendung anzeigen, führen Sie die erforderlichen Schritte aus:

Themen

- [Erstellen einer Kartenressource](#)
- [Authentifizieren Ihrer Anforderungen](#)

Erstellen einer Kartenressource

Um eine Karte in Ihrer Anwendung zu verwenden, benötigen Sie eine Kartenressource, die den Kartenstil und den Datenanbieter angibt, der in Ihren Karten verwendet werden soll.

Note

Wenn Ihre Anwendung Assets, die Sie in Ihrem Unternehmen verwenden, wie z. B. Lieferfahrzeuge oder Mitarbeiter, verfolgt oder weiterleitet, dürfen Sie Esri nicht als Geolokalisierungsanbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS-Servicebedingungen](#).

Sie können eine Kartenressource mithilfe der Amazon Location Service-Konsole, der AWS CLI oder der Amazon Location APIs erstellen.

Console

So erstellen Sie eine Kartenressource mit der Amazon Location Service-Konsole

1. Wählen Sie in der Amazon-Standortkonsole auf der Seite [Karten](#) die Option Karte erstellen aus, um eine Vorschau der Kartenstile anzuzeigen.

2. Fügen Sie einen Namen und eine Beschreibung für die neue Kartenressource hinzu.
3. Wählen Sie einen Kartenstil aus.

 Note

Wenn Ihre Anwendung Assets, die Sie in Ihrem Unternehmen verwenden, wie z. B. Lieferfahrzeuge oder Mitarbeiter, verfolgt oder weiterleitet, dürfen Sie Esri nicht als Geolokalisierungsanbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS-Servicebedingungen](#).

4. Wählen Sie aus der zu [Politische Ansichten](#) verwendenden aus.
5. Stimmen Sie den Allgemeinen Geschäftsbedingungen für Amazon Location zu und wählen Sie dann Karte erstellen aus. Sie können mit der ausgewählten Karte interagieren: Vergrößern, Verkleinern oder Vergrößern in eine beliebige Richtung.
6. Damit Ihre Benutzer den Stil wechseln können (z. B. damit sie zwischen Satellitenbildern und Vektorstil wechseln können), müssen Sie für jeden Stil eine Kartenressource erstellen.

Sie können Ressourcen mit Kartenstilen löschen, die Sie nicht auf der [Karten-Startseite](#) in der -Konsole verwenden möchten.

API

So erstellen Sie eine Kartenressource mithilfe der Amazon Location APIs

Verwenden Sie die [-CreateMap](#)Operation aus den Amazon Location APIs .

Das folgende Beispiel ist eine API-Anforderung zum Erstellen einer Kartenressource namens *ExampleMap* unter Verwendung des *VectorEsriStreets* Kartenstils.

```
POST /maps/v0/maps HTTP/1.1
Content-type: application/json

{
  "Configuration": {
    "Style": "VectorEsriStreets"
  },
  "MapName": "ExampleMap"
}
```

Note

Wenn Ihre Anwendung Assets, die Sie in Ihrem Unternehmen verwenden, wie z. B. Lieferfahrzeuge oder Mitarbeiter, verfolgt oder weiterleitet, dürfen Sie Esri nicht als Geolokalisierungsanbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS-Servicebedingungen](#).

AWS CLI

So erstellen Sie eine Kartenressource mit -AWS CLIBefehlen

Verwenden Sie den [create-map](#)-Befehl.

Im folgenden Beispiel wird eine Kartenressource namens *ExampleMap* mit *VectorEsriStreets* als Kartenstil erstellt.

```
aws location \  
  create-map \  
    --configuration Style="VectorEsriStreets" \  
    --map-name "ExampleMap"
```

Note

Wenn Ihre Anwendung Assets, die Sie in Ihrem Unternehmen verwenden, wie z. B. Lieferfahrzeuge oder Mitarbeiter, verfolgt oder weiterleitet, dürfen Sie Esri nicht als Geolokalisierungsanbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS-Servicebedingungen](#).

Authentifizieren Ihrer Anforderungen

Sobald Sie eine Kartenressource erstellt haben und bereit sind, Standortfunktionen in Ihrer Anwendung zu erstellen, müssen Sie auswählen, wie Sie Ihre Anfragen authentifizieren würden.

Note

Die meisten Frontend-Anwendungen für Karten erfordern nicht authentifizierten Zugriff auf die Karten oder andere Funktionen von Amazon Location Service. Abhängig von Ihrer

Anwendung möchten Sie möglicherweise AWS Signature v4 verwenden, um Anfragen zu authentifizieren, oder Sie können Amazon Cognito- oder Amazon-Location-API-Schlüssel für die nicht authentifizierte Verwendung verwenden. Weitere Informationen zu all diesen Optionen finden Sie unter [Gewähren des Zugriffs auf Amazon Location Service](#).

Anzeigen einer Karte in Ihrer Anwendung

Dieser Abschnitt enthält Tutorials zur Verwendung von Karten-Rendering-Tools, um eine Karte in Ihrer mobilen oder Webanwendung anzuzeigen, wenn Sie Amazon Location APIs verwenden. Wie im [So verwenden Sie Amazon Location Service](#) Thema erwähnt, haben Sie eine Auswahl an Bibliotheken, die Sie beim Rendern von Karten mit Amazon Location verwenden können, einschließlich Amplify MapLibre, und Tangram.

Führen Sie einen der folgenden Schritte aus, um eine Zuordnung in Ihrer Anwendung anzuzeigen:

- Die direkteste Möglichkeit, eine Karte in Ihren Web- und mobilen Frontend-Anwendungen anzuzeigen, ist die Verwendung von MapLibre. Sie können den [MapLibre Tutorials](#) oder sogar dem [Schnellstart-Tutorial](#) folgen, um zu erfahren, wie Sie verwenden MapLibre.
- Wenn Sie bereits AWS Amplify Entwickler sind, sollten Sie das Amplify Geo SDK verwenden. Weitere Informationen finden Sie im [Amplify-Tutorial](#).
- Wenn Sie bereits ein Benutzer von Tangram sind und es weiterhin zum Rendern Ihrer Karte verwenden möchten, während Sie zu Amazon Location Service wechseln, folgen Sie dem [Tutorial zu Tangram](#).

Themen

- [Verwenden der MapLibre Bibliothek mit Amazon Location Service](#)
- [Verwenden der Amplify-Bibliothek mit Amazon Location Service](#)
- [Verwenden von Tangram mit Amazon Location Service](#)

Verwenden der MapLibre Bibliothek mit Amazon Location Service

Die folgenden Tutorials führen Sie durch die Verwendung der - MapLibre Bibliothek mit Amazon Location.

Themen

- [Verwenden von MapLibre microSD JS mit Amazon Location Service](#)
- [Verwenden des MapLibre nativen SDK für Android mit Amazon Location Service](#)
- [Verwenden des MapLibre nativen SDK für iOS mit Amazon Location Service](#)

Verwenden von MapLibre microSD JS mit Amazon Location Service

Verwenden Sie [MapLibre microSD JS](#), um clientseitige Maps in Webanwendungen einzubetten.

MapLibre microSD JS ist eine Open-Source- JavaScript Bibliothek, die mit den Stilen und Kacheln kompatibel ist, die von der Amazon Location Service Maps API bereitgestellt werden. Sie können MapLibre microSD JS in einen einfachen HTML oder eine einfache JavaScript Anwendung integrieren, um anpassbare und reaktionsschnelle clientseitige Karten einzubetten.

In diesem Tutorial wird beschrieben, wie MapLibre microSD JS in Amazon Location in einem einfachen HTML und einer einfachen JavaScript Anwendung integriert wird. Die gleichen Bibliotheken und Techniken, die in diesem Tutorial vorgestellt werden, gelten auch für Frameworks wie [React](#) und [Angular](#) .

Die Beispielanwendung für dieses Tutorial ist als Teil des Amazon Location Service-Beispiel-Repositoryys auf verfügbar [GitHub](#).

Erstellen der Anwendung: Gerüsterstellung

In diesem Tutorial wird eine Webanwendung erstellt, die verwendet, JavaScript um eine Karte auf einer HTML-Seite zu erstellen.

Erstellen Sie zunächst eine HTML-Seite (`index.html`), die den Container der Karte enthält:

- Geben Sie ein `-div`Element mit einem `id` von `einmap`, um die Dimensionen der Karte auf die Kartenansicht anzuwenden. Die Dimensionen werden vom Ansichtsfenster geerbt.

```
<html>
  <head>
    <style>
      body {
        margin: 0;
      }

      #map {
```

```
        height: 100vh; /* 100% of viewport height */
    }
</style>
</head>
<body>
    <!-- map container -->
    <div id="map" />
</body>
</html>
```

Erstellen der Anwendung: Hinzufügen von Abhängigkeiten

Fügen Sie Ihrer Anwendung die folgenden Abhängigkeiten hinzu:

- MapLibre microSD JS (v3.x) und das zugehörige CSS.
- Der Amazon-Standort [Authentifizierungshelfer](#).

```
<!-- CSS dependencies -->
<link
  href="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.css"
  rel="stylesheet"
/>
<!-- JavaScript dependencies -->
<script src="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.js"></script>
<script src="https://unpkg.com/@aws/amazon-location-authentication-helper.js"></script>
<script>
  // application-specific code
</script>
```

Dadurch wird eine leere Seite mit dem Container der Zuordnung erstellt.

Erstellen der Anwendung: Konfiguration

So konfigurieren Sie Ihre Anwendung mit JavaScript:

1. Geben Sie die Namen und Kennungen Ihrer Ressourcen ein.

```
// Cognito Identity Pool ID
const identityPoolId = "us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd";
// Amazon Location Service Map name
const mapName = "ExampleMap";
```

2. Instanzieren Sie einen Anmeldeinformationsanbieter mithilfe des nicht authentifizierten Identitätspools, den Sie unter [Verwenden von Karten – Schritt 2, Authentifizierung einrichten](#) erstellt haben. Wir werden dies in eine Funktion namens `initializeMap`, die auch anderen Karteninitialisierungscode enthält, der im nächsten Schritt hinzugefügt wird

```
// extract the Region from the Identity Pool ID; this will be used for both Amazon
  Cognito and Amazon Location
AWS.config.region = identityPoolId.split(":")[0];

async function initializeMap() {
  // Create an authentication helper instance using credentials from Cognito
  const authHelper = await
  amazonLocationAuthHelper.withIdentityPoolId(identityPoolId);

  // ... more here, later
}
```

Erstellen der Anwendung: Karteninitialisierung

Damit die Karte nach dem Laden der Seite angezeigt wird, müssen Sie die Karte initialisieren. Sie können die anfängliche Kartenposition anpassen, zusätzliche Steuerelemente hinzufügen und Daten überlagern.

```
async function initializeMap() {
  // Create an authentication helper instance using credentials from Cognito
  const authHelper = await amazonLocationAuthHelper.withIdentityPoolId(identityPoolId);

  // Initialize the map
  const map = new maplibregl.Map({
    container: "map",
    center: [-123.1187, 49.2819], // initial map centerpoint
    zoom: 10, // initial map zoom
    style: 'https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/style-
descriptor',
    ...authHelper.getMapAuthenticationOptions(), // authentication, using cognito
  });

  map.addControl(new maplibregl.NavigationControl(), "top-left");
}

initializeMap();
```

Note

Sie müssen für jeden Datenanbieter, den Sie verwenden, entweder in Ihrer Anwendung oder in Ihrer Dokumentation ein Wortzeichen oder eine Textzuweisung angeben. Attributionszeichenfolgen sind in der Style-Deskriptor-Antwort unter den `sources.grabmaptiles.attribution` Schlüsseln `sources.esri.attribution` `sources.here.attribution`, und enthalten. MapLibre GL JS stellt automatisch die Attribution bereit. Wenn Sie Amazon-Location-Ressourcen mit [Datenanbietern](#) verwenden, lesen Sie unbedingt die [Servicebedingungen](#).

Ausführen der Anwendung

Sie können diese Beispielanwendung ausführen, indem Sie sie auf einem lokalen Webserver verwenden oder sie in einem Browser öffnen.

Um einen lokalen Webserver zu verwenden, können Sie `npx` verwenden, da es als Teil von Node.js installiert ist. Sie können `npx serve` innerhalb desselben Verzeichnisses wie `index.html` verwenden. Dies dient der Anwendung auf `localhost:5000`.

Note

Wenn die Richtlinie, die Sie für Ihre nicht authentifizierte Amazon Cognito-Rolle erstellt haben, eine `referrer` Bedingung enthält, werden Sie möglicherweise daran gehindert, mit `localhost: URLs` zu testen. In diesem Fall können Sie mit einem Webserver testen, der eine URL bereitstellt, die sich in Ihrer Richtlinie befindet.

Nach Abschluss des Tutorials sieht die endgültige Anwendung wie im folgenden Beispiel aus.

```
<!-- index.html -->
<html>
  <head>
    <link href="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.css"
rel="stylesheet" />
    <style>
      body {
        margin: 0;
      }
      #map {
```

```
        height: 100vh;
    }
</style>
</head>

<body>
  <!-- map container -->
  <div id="map" />
  <!-- JavaScript dependencies -->
  <script src="https://unpkg.com/maplibre-gl@3.x/dist/maplibre-gl.js"></script>
  <script src="https://unpkg.com/@aws/amazon-location-authentication-helper.js"></
script>
  <script>
    // configuration
    const identityPoolId = "us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd"; //
Cognito Identity Pool ID
    const mapName = "ExampleMap"; // Amazon Location Service Map Name

    // extract the region from the Identity Pool ID
    const region = identityPoolId.split(":")[0];

    async function initializeMap() {
      // Create an authentication helper instance using credentials from Cognito
      const authHelper = await
amazonLocationAuthHelper.withIdentityPoolId(identityPoolId);

      // Initialize the map
      const map = new maplibregl.Map({
        container: "map",
        center: [-123.115898, 49.295868],
        zoom: 10,
        style: `https://maps.geo.${region}.amazonaws.com/maps/v0/maps/${mapName}/
style-descriptor`,
        ...authHelper.getMapAuthenticationOptions(),
      });
      map.addControl(new maplibregl.NavigationControl(), "top-left");
    }

    initializeMap();
  </script>
</body>
</html>
```

Wenn Sie diese Anwendung ausführen, wird eine Vollbildkarte mit dem von Ihnen gewählten Kartenstil angezeigt. Dieses Beispiel ist im Amazon Location Service-Beispiel-Repository auf verfügbar [GitHub](#).

Verwenden des MapLibre nativen SDK für Android mit Amazon Location Service

Verwenden Sie [MapLibre das native](#) SDK, um interaktive Karten in Ihre Android-Anwendungen einzubetten.

Das MapLibre native SDK für Android ist eine Bibliothek, die auf [Mapbox Native](#) basiert und mit den Stilen und Kacheln kompatibel ist, die von der Amazon Location Service Maps API bereitgestellt werden. Sie können MapLibre das native SDK für Android integrieren, um interaktive Kartenansichten in skalierbare, anpassbare Vektorkarten in Ihre Android-Anwendungen einzubetten.

In diesem Tutorial wird beschrieben, wie Sie das MapLibre native SDK für Android in Amazon Location integrieren. Die Beispielanwendung für dieses Tutorial ist als Teil des Amazon Location Service-Beispiel-Repositorys auf verfügbar [GitHub](#).

Erstellen der Anwendung: Initialisierung

So initialisieren Sie Ihre Anwendung:

1. Erstellen Sie ein neues Android Studio-Projekt aus der Vorlage Leere Aktivität.
2. Stellen Sie sicher, dass Kotlin für die Projektsprache ausgewählt ist.
3. Wählen Sie ein Mindest-SDK von API 14: Android 4.0 (IceSpeed Speed) oder höher aus.
4. Öffnen Sie Projektstruktur und navigieren Sie dann zu Datei > Projektstruktur..., um den Abschnitt Abhängigkeiten auszuwählen.
5. Wenn <Alle Module> ausgewählt ist, wählen Sie dann die Schaltfläche +, um eine neue Bibliotheksabhängigkeit hinzuzufügen.
6. Fügen Sie AWS Android SDK Version 2.20.0 oder höher hinzu. Beispiel:
`com.amazonaws:aws-android-sdk-core:2.20.0`
7. Fügen Sie das MapLibre native SDK für Android Version 9.4.0 oder höher hinzu. Beispiel:
`org.maplibre.gl:android-sdk:9.4.0`
8. Fügen Sie auf Projektebene Ihrer build.gradle-Datei das folgende Maven-Repository hinzu, um auf die MapLibre Pakete für Android zuzugreifen:

```
allprojects {
    repositories {
        // Retain your existing repositories
    }
}
```

```
        google()
        jcenter()

        // Declare the repositories for MapLibre
        mavenCentral()
    }
}
```

Erstellen der Anwendung: Konfiguration

So konfigurieren Sie Ihre Anwendung mit Ihren Ressourcen und Ihrer AWS Region:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="identityPoolId">us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd</string>
    <string name="mapName">ExampleMap</string>
    <string name="awsRegion">us-east-1</string>
</resources>
```

Erstellen der Anwendung: Aktivitätslayout

Bearbeiten von `app/src/main/res/layout/activity_main.xml`:

- Fügen Sie ein `hinzuMapView`, das die Zuordnung wiedergibt. Dadurch wird auch der anfängliche Mittelpunkt der Karte festgelegt.
- Fügen Sie ein `hinzuTextView`, das die Zuordnung anzeigt.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <com.mapbox.mapboxsdk.maps.MapView
        android:id="@+id/mapView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
```

```
app:mapbox_cameraTargetLat="49.2819"  
app:mapbox_cameraTargetLng="-123.1187"  
app:mapbox_cameraZoom="12"  
app:mapbox_uiAttribution="false"  
app:mapbox_uiLogo="false" />  
  
<TextView  
    android:id="@+id/attributionView"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:background="#80808080"  
    android:padding="5sp"  
    android:textColor="@android:color/black"  
    android:textSize="10sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    tools:ignore="SmallSp" />  
</androidx.constraintlayout.widget.ConstraintLayout>
```

Note

Sie müssen für jeden Datenanbieter, den Sie verwenden, entweder in Ihrer Anwendung oder in Ihrer Dokumentation ein Wortzeichen oder eine Textzuweisung angeben. Attributionszeichenfolgen sind in der Style-Deskriptor-Antwort unter den `source.grabmaptiles.attribution` Schlüsseln `sources.esri.attribution` `sources.here.attribution`, und enthalten. Wenn Sie Amazon-Location-Ressourcen mit [Datenanbietern](#) verwenden, lesen Sie unbedingt die [Nutzungsbedingungen für den Service](#).

Erstellen der Anwendung: Anforderungstransformation

Erstellen Sie eine Klasse mit dem Namen `SigV4Interceptor`, um AWS-Anforderungen abzufangen und sie mit [Signature Version 4](#) zu signieren. Dies wird bei dem HTTP-Client registriert, der zum Abrufen von Kartenressourcen verwendet wird, wenn die Hauptaktivität erstellt wird.

```
package aws.location.demo.okhttp  
  
import com.amazonaws.DefaultRequest  
import com.amazonaws.auth.AWS4Signer  
import com.amazonaws.auth.AWSCredentialsProvider
```

```
import com.amazonaws.http.HttpMethodName
import com.amazonaws.util.IOUtils
import okhttp3.HttpUrl
import okhttp3.Interceptor
import okhttp3.Request
import okhttp3.Response
import okio.Buffer
import java.io.ByteArrayInputStream
import java.net.URI

class SigV4Interceptor(
    private val credentialsProvider: AWSCredentialsProvider,
    private val serviceName: String
) : Interceptor {
    override fun intercept(chain: Interceptor.Chain): Response {
        val originalRequest = chain.request()

        if (originalRequest.url().host().contains("amazonaws.com")) {
            val signer = if (originalRequest.url().encodedPath().contains("@")) {
                // the presence of "@" indicates that it doesn't need to be double URL-
                encoded
                AWS4Signer(false)
            } else {
                AWS4Signer()
            }

            val awsRequest = toAWSRequest(originalRequest, serviceName)
            signer.setServiceName(serviceName)
            signer.sign(awsRequest, credentialsProvider.credentials)

            return chain.proceed(toSignedOkHttpRequest(awsRequest, originalRequest))
        }

        return chain.proceed(originalRequest)
    }

    companion object {
        fun toAWSRequest(request: Request, serviceName: String): DefaultRequest<Any> {
            // clone the request (AWS-style) so that it can be populated with
            credentials
            val dr = DefaultRequest<Any>(serviceName)

            // copy request info
            dr.httpMethod = HttpMethodName.valueOf(request.method())
        }
    }
}
```

```
with(request.url()) {
    dr.resourcePath = uri().path
    dr.endpoint = URI.create("${scheme()}://${host()}")

    // copy parameters
    for (p in queryParameterNames()) {
        if (p != "") {
            dr.addParameter(p, queryParameter(p))
        }
    }

    // copy headers
    for (h in request.headers().names()) {
        dr.addHeader(h, request.header(h))
    }

    // copy the request body
    val bodyBytes = request.body()?.let { body ->
        val buffer = Buffer()
        body.writeTo(buffer)
        IOUtils.toByteArray(buffer.inputStream())
    }

    dr.content = ByteArrayInputStream(bodyBytes ?: ByteArray(0))

    return dr
}

fun toSignedOkHttpRequest(
    awsRequest: DefaultRequest<Any>,
    originalRequest: Request
): Request {
    // copy signed request back into an OkHttp Request
    val builder = Request.Builder()

    // copy headers from the signed request
    for ((k, v) in awsRequest.headers) {
        builder.addHeader(k, v)
    }

    // start building an HttpUrl
    val urlBuilder = HttpUrl.Builder()
        .host(awsRequest.endpoint.host)
```

```
        .scheme(awsRequest.endpoint.scheme)
        .encodedPath(awsRequest.resourcePath)

        // copy parameters from the signed request
        for ((k, v) in awsRequest.parameters) {
            urlBuilder.addQueryParameter(k, v)
        }

        return builder.url(urlBuilder.build())
            .method(originalRequest.method(), originalRequest.body())
            .build()
    }
}
}
```

Erstellen der Anwendung: Hauptaktivität

Die Hauptaktivität ist für die Initialisierung der Ansichten verantwortlich, die Benutzern angezeigt werden. Dazu gehören:

- Instanzieren eines Amazon Cognito-CredentialsProvider.
- Registrieren des Signature Version 4-Interceptors.
- Konfigurieren der Karte, indem sie auf einen Deskriptor im Kartenstil verweist und die entsprechende Zuordnung anzeigt.

MainActivity ist auch für die Weiterleitung von Lebenszyklusereignissen an die Kartenansicht verantwortlich, sodass das aktive Ansichtsfenster zwischen Aufrufen beibehalten werden kann.

```
package aws.location.demo.maplibre

import android.os.Bundle
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import aws.location.demo.okhttp.SigV4Interceptor
import com.amazonaws.auth.CognitoCachingCredentialsProvider
import com.amazonaws.regions.Regions
import com.mapbox.mapboxsdk.Mapbox
import com.mapbox.mapboxsdk.maps.MapView
import com.mapbox.mapboxsdk.maps.Style
import com.mapbox.mapboxsdk.module.http.HttpRequestUtil
import okhttp3.OkHttpClient
```

```
private const val SERVICE_NAME = "geo"

class MainActivity : AppCompatActivity() {
    private var mapView: MapView? = null

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

        // configuration
        val identityPoolId = getString(R.string.identityPoolId)
        val region = getString(R.string.awsRegion)
        val mapName = getString(R.string.mapName)

        // Credential initialization
        val credentialProvider = CognitoCachingCredentialsProvider(
            applicationContext,
            identityPoolId,
            Regions.fromName(identityPoolId.split(":").first())
        )

        // initialize MapLibre
        Mapbox.getInstance(this, null)
        HttpRequestUtil.setOkHttpClient(
            OkHttpClient.Builder()
                .addInterceptor(SigV4Interceptor(credentialProvider, SERVICE_NAME))
                .build()
        )

        // initialize the view
        setContentView(R.layout.activity_main)

        // initialize the map view
        mapView = findViewById(R.id.mapView)
        mapView?.onCreate(savedInstanceState)
        mapView?.getMapAsync { map ->
            map.setStyle(
                Style.Builder()
                    .fromUri("https://maps.geo.${region}.amazonaws.com/maps/v0/maps/
${mapName}/style-descriptor")
            ) { style ->
                findViewById<TextView>(R.id.attributionView).text =
                    style.sources.first()?.attribution
            }
        }
    }
}
```

```
    }  
}  
  
override fun onStart() {  
    super.onStart()  
    mapView?.onStart()  
}  
  
override fun onResume() {  
    super.onResume()  
    mapView?.onResume()  
}  
  
override fun onPause() {  
    super.onPause()  
    mapView?.onPause()  
}  
  
override fun onStop() {  
    super.onStop()  
    mapView?.onStop()  
}  
  
override fun onSaveInstanceState(outState: Bundle) {  
    super.onSaveInstanceState(outState)  
    mapView?.onSaveInstanceState(outState)  
}  
  
override fun onLowMemory() {  
    super.onLowMemory()  
    mapView?.onLowMemory()  
}  
  
override fun onDestroy() {  
    super.onDestroy()  
    mapView?.onDestroy()  
}  
}
```

Wenn Sie diese Anwendung ausführen, wird eine Vollbildkarte im Stil Ihrer Wahl angezeigt. Dieses Beispiel ist als Teil des Amazon Location Service-Beispiel-Repositorys auf verfügbar [GitHub](#).

Verwenden des MapLibre nativen SDK für iOS mit Amazon Location Service

Verwenden Sie [MapLibre das native SDK für iOS](#), um clientseitige Karten in iOS-Anwendungen einzubetten.

Das MapLibre native SDK für iOS ist eine Bibliothek, die auf [Mapbox microSD Native](#) basiert und mit den Stilen und Kacheln kompatibel ist, die von der Amazon Location Service Maps API bereitgestellt werden. Sie können MapLibre das native SDK für iOS integrieren, um interaktive Kartenansichten mit skalierbaren, anpassbaren Vektorkarten in Ihre iOS-Anwendungen einzubetten.

In diesem Tutorial wird beschrieben, wie Sie das MapLibre native SDK für iOS in Amazon Location integrieren. Die Beispielanwendung für dieses Tutorial ist als Teil des Amazon Location Service-Beispiel-Repositorys auf verfügbar [GitHub](#).

Erstellen der Anwendung: Initialisierung

So initialisieren Sie Ihre Anwendung:

1. Erstellen Sie ein neues Xcode-Projekt aus der -App-Vorlage.
2. Wählen Sie SwiftUI als Schnittstelle aus.
3. Wählen Sie die SwiftUI-Anwendung für ihren Lebenszyklus aus.
4. Wählen Sie Swift als Sprache aus.

Hinzufügen MapLibre von Abhängigkeiten mit Swift Packages

So fügen Sie Ihrem Xcode-Projekt eine Paketabhängigkeit hinzu:

1. Navigieren Sie zu Datei > Pakete austauschen > Paketabhängigkeit hinzufügen.
2. Geben Sie die Repository-URL ein: **`https://github.com/maplibre/maplibre-gl-native-distribution`**

Note

Weitere Informationen zu Swift-Paketen finden Sie unter [Hinzufügen von Paketabhängigkeiten zu Ihrer App](#) unter Apple.com

3. Installieren Sie in Ihrem Terminal CocoaPods:

```
sudo gem install cocoapods
```

4. Navigieren Sie zum Projektverzeichnis Ihrer Anwendung und initialisieren Sie die Podfile mit dem CocoaPods Paketmanager:

```
pod init
```

5. Öffnen Sie die Podfile-Datei, die AWSCore Sie als Abhängigkeit hinzufügen möchten:

```
platform :ios, '12.0'

target 'Amazon Location Service Demo' do
  use_frameworks!

  pod 'AWSCore'
end
```

6. Abhängigkeiten herunterladen und installieren:

```
pod install --repo-update
```

7. Öffnen Sie den Xcode-Workspace, den CocoaPods erstellt hat:

```
xed .
```

Erstellen der Anwendung: Konfiguration

Fügen Sie die folgenden Schlüssel und Werte zu Info.plist hinzu, um die Anwendung zu konfigurieren:

Schlüssel	Wert
AWSRegion	us-east-1
IdentityPoolId	us-east-1:54f2ba88-9390-498d-aaaa5-0d97fb7ca3bd
MapName	ExampleMap

Erstellen der Anwendung: ContentView Layout

Um die Karte zu rendern, bearbeiten Sie `ContentView.swift`:

- Fügen Sie eine `hinzuMapView`, die die Zuordnung wiedergibt.
- Fügen Sie einen `hinzuTextField`, der die Zuordnung anzeigt.

Dadurch wird auch der anfängliche Mittelpunkt der Karte festgelegt.

```
import SwiftUI

struct ContentView: View {
    @State private var attribution = ""

    var body: some View {
        MapView(attribution: $attribution)
            .centerCoordinate(.init(latitude: 49.2819, longitude: -123.1187))
            .zoomLevel(12)
            .edgesIgnoringSafeArea(.all)
            .overlay(
                TextField("", text: $attribution)
                    .disabled(true)
                    .font(.system(size: 12, weight: .light, design: .default))
                    .foregroundColor(.black)
                    .background(Color.init(Color.RGBColorSpace.sRGB, white: 0.5,
opacity: 0.5))
                    .cornerRadius(1),
                alignment: .bottomTrailing)
            }
    }

    struct ContentView_Previews: PreviewProvider {
        static var previews: some View {
            ContentView()
        }
    }
}
```

Note

Sie müssen für jeden Datenanbieter, den Sie verwenden, entweder in Ihrer Anwendung oder in Ihrer Dokumentation ein Wortzeichen oder eine

Textzuweisung angeben. Attributionszeichenfolgen sind in der Style-Deskriptor-Antwort unter den `source.grabmaptiles.attribution` Schlüsseln `sources.esri.attributionsources.here.attribution`, und enthalten. Wenn Sie Amazon-Location-Ressourcen mit [Datenanbietern](#) verwenden, lesen Sie unbedingt die [Servicebedingungen](#).

Erstellen der Anwendung: Anforderungstransformation

Erstellen Sie eine neue Swift-Datei mit dem Namen `AWSSignatureV4Delegate.swift`, die die folgende Klassendefinition `AWSSignatureV4Delegate` enthält, um AWS-Anfragen abzufangen und sie mit [Signature Version 4](#) zu signieren. Eine Instance dieser Klasse wird als Offline-Speicherdelegierter zugewiesen, der auch für das Umschreiben von URLs in der Kartenansicht verantwortlich ist.

```
import AWSCore
import Mapbox

class AWSSignatureV4Delegate : NSObject, MGLOfflineStorageDelegate {
    private let region: AWSRegionType
    private let identityPoolId: String
    private let credentialsProvider: AWSCredentialsProvider

    init(region: AWSRegionType, identityPoolId: String) {
        self.region = region
        self.identityPoolId = identityPoolId
        self.credentialsProvider = AWSCognitoCredentialsProvider(regionType: region,
identityPoolId: identityPoolId)
        super.init()
    }

    class func doubleEncode(path: String) -> String? {
        return path.addingPercentEncoding(withAllowedCharacters: .urlPathAllowed)?
            .addingPercentEncoding(withAllowedCharacters: .urlPathAllowed)
    }

    func offlineStorage(_ storage: MGLOfflineStorage, urlForResourceOf kind:
MGLResourceKind, with url: URL) -> URL {
        if url.host?.contains("amazonaws.com") != true {
            // not an AWS URL
            return url
        }
    }
}
```

```
// URL-encode spaces, etc.
let keyPath = String(url.path.dropFirst())
guard let percentEncodedKeyPath =
keyPath.addingPercentEncoding(withAllowedCharacters: .urlPathAllowed) else {
    print("Invalid characters in path '\(keyPath)'; unsafe to sign")
    return url
}

let endpoint = AWSEndpoint(region: region, serviceName: "geo", url: url)
let requestHeaders: [String: String] = ["host": endpoint!.hostName]

// sign the URL
let task = AWSSignatureV4Signer
    .generateQueryStringForSignatureV4(
        withCredentialProvider: credentialsProvider,
        httpMethod: .GET,
        expireDuration: 60,
        endpoint: endpoint!,
        // workaround for https://github.com/aws-amplify/aws-sdk-ios/
issues/3215
        keyPath: AWSSignatureV4Delegate.doubleEncode(path:
percentEncodedKeyPath),
        requestHeaders: requestHeaders,
        requestParameters: .none,
        signBody: true)
task.waitUntilFinished()

if let error = task.error as NSError? {
    print("Error occurred: \(error)")
}

if let result = task.result {
    var urlComponents = URLComponents(url: (result as URL),
resolvingAgainstBaseURL: false)!
    // re-use the original path; workaround for https://github.com/aws-amplify/
aws-sdk-ios/issues/3215
    urlComponents.path = url.path

    // have Mapbox GL fetch the signed URL
    return (urlComponents.url)!
}

// fall back to an unsigned URL
return url
```

```
}  
}
```

Erstellen der Anwendung: Kartenansicht

Die Kartenansicht ist für die Initialisierung einer Instance von `AWSSignatureV4Delegate` und die Konfiguration der zugrunde liegenden verantwortlich `MGLMapView`, die Ressourcen abrufen und die Karte rendert. Außerdem übernimmt es die Weitergabe von Attributionszeichenfolgen aus der Quelle des Stildeskriptors zurück an die `ContentView`.

Erstellen Sie eine neue Swift-Datei mit dem Namen `MapView.swift`, die die folgende `struct` Definition enthält:

```
import SwiftUI  
import AWSCore  
import Mapbox  
  
struct MapView: UIViewRepresentable {  
    @Binding var attribution: String  
  
    private var mapView: MGLMapView  
    private var signingDelegate: MGLOfflineStorageDelegate  
  
    init(attribution: Binding<String>) {  
        let regionName = Bundle.main.object(forKey: "AWSRegion") as!  
String  
        let identityPoolId = Bundle.main.object(forKey: "IdentityPoolId")  
as! String  
        let mapName = Bundle.main.object(forKey: "MapName") as! String  
  
        let region = (regionName as NSString).aws_regionTypeValue()  
  
        // MGLOfflineStorage doesn't take ownership, so this needs to be a member here  
        signingDelegate = AWSSignatureV4Delegate(region: region, identityPoolId:  
identityPoolId)  
  
        // register a delegate that will handle SigV4 signing  
        MGLOfflineStorage.shared.delegate = signingDelegate  
  
        mapView = MGLMapView(  
            frame: .zero,  
            styleURL: URL(string: "https://maps.geo.\(regionName).amazonaws.com/maps/  
v0/maps/\(mapName)/style-descriptor"))
```

```
        _attribution = attribution
    }

    func makeCoordinator() -> Coordinator {
        Coordinator($attribution)
    }

    class Coordinator: NSObject, MGLMapViewDelegate {
        var attribution: Binding<String>

        init(_ attribution: Binding<String>) {
            self.attribution = attribution
        }

        func mapView(_ mapView: MGLMapView, didFinishLoading style: MGLStyle) {
            let source = style.sources.first as? MGLVectorTileSource
            let attribution = source?.attributionInfos.first
            self.attribution.wrappedValue = attribution?.title.string ?? ""
        }
    }

    // MARK: - UIViewRepresentable protocol

    func makeUIView(context: UIViewRepresentableContext<MapView>) -> MGLMapView {
        mapView.delegate = context.coordinator

        mapView.logoView.isHidden = true
        mapView.attributionButton.isHidden = true
        return mapView
    }

    func updateUIView(_ uiView: MGLMapView, context:
    UIViewRepresentableContext<MapView>) {
    }

    // MARK: - MGLMapView proxy

    func centerCoordinate(_ centerCoordinate: CLLocationCoordinate2D) -> MapView {
        mapView.centerCoordinate = centerCoordinate
        return self
    }

    func zoomLevel(_ zoomLevel: Double) -> MapView {
```

```
        mapView.zoomLevel = zoomLevel
        return self
    }
}
```

Wenn Sie diese Anwendung ausführen, wird eine Vollbildkarte im Stil Ihrer Wahl angezeigt. Dieses Beispiel ist als Teil des Amazon Location Service-Beispiel-Repositorys auf verfügbar [GitHub](#).

Verwenden der Amplify-Bibliothek mit Amazon Location Service

Das folgende Tutorial führt Sie durch die Verwendung von AWS Amplify mit Amazon Location. Amplify verwendet MapLibre microSD JS, um Karten in Ihrer JavaScript-basierten Anwendung zu rendern.

Amplify ist eine Reihe von Open-Source-Clientbibliotheken, die Schnittstellen für verschiedene Kategorien von -Services bereitstellen, einschließlich Amplify Geo, das von Amazon Location Service unterstützt wird. [Erfahren Sie mehr über die AWS Amplify Geo- JavaScript Bibliothek](#) .

Note

In diesem Tutorial wird davon ausgegangen, dass Sie die Schritte unter [Verwenden von Karten – So fügen Sie Ihrer Anwendung eine Zuordnung hinzu](#) bereits befolgt haben.

Erstellen der Anwendung: Gerüsterstellung

In diesem Tutorial wird eine Webanwendung erstellt, die verwendet, JavaScript um eine Karte auf einer HTML-Seite zu erstellen.

Erstellen Sie zunächst eine HTML-Seite (`index.html`), die den Container der Karte enthält:

- Geben Sie ein `-divElement` mit einem `id` von `einmap`, um die Dimensionen der Karte auf die Kartenansicht anzuwenden. Die Dimensionen werden vom Ansichtsfenster geerbt.

```
<html>
  <head>
    <style>
      body { margin: 0; }
      #map { height: 100vh; } /* 100% of viewport height */
    </style>
```

```

</head>

<body>
  <!-- map container -->
  <div id="map" />
</body>
</html>

```

Erstellen der Anwendung: Hinzufügen von Abhängigkeiten

Fügen Sie Ihrer Anwendung die folgenden Abhängigkeiten hinzu:

- AWS Amplify -Map- und Geobibliotheken.
- AWS Amplify -Core-Bibliothek.
- AWS Amplify -Authentifizierungsbibliothek.
- AWS Amplify Stylesheet.

```

<!-- CSS dependencies -->
  <link href="https://cdn.amplify.aws/packages/maplibre-
gl/1.15.2/maplibre-gl.css" rel="stylesheet" integrity="sha384-
DrPVD9GufrxGb7kWwRv0CywpxTmfvbK0Z5i5pN7urmIThew0zXKTME+gutUgtpeD"
  crossorigin="anonymous" referrerpolicy="no-referrer"></link>

<!-- JavaScript dependencies -->
  <script src="https://cdn.amplify.aws/packages/maplibre-gl/1.15.2/maplibre-gl.js"
  integrity="sha384-rwYfkmA0pciZS2bDuwZ/Xa/Gog6jXem8D/whm3wnsZSVFemDDlprcUXHnDDUcrNU"
  crossorigin="anonymous" referrerpolicy="no-referrer"></script>
  <script src="https://cdn.amplify.aws/packages/core/4.3.0/aws-amplify-core.min.js"
  integrity="sha384-70h+5w0l7XGyYvSqBKi2Q7SA5K640V5nyW2/LEbevDQEV1HMJqJLA1A00z2hu8fJ"
  crossorigin="anonymous" referrerpolicy="no-referrer"></script>
  <script src="https://cdn.amplify.aws/packages/auth/4.3.8/aws-amplify-auth.min.js"
  integrity="sha384-jfkXCEfYyVmDXyKlgWNwv54xRaZgk14m7sjeb2jLVBtUXCD2p+WU8YZ2mPZ9Xbdw"
  crossorigin="anonymous" referrerpolicy="no-referrer"></script>
  <script src="https://cdn.amplify.aws/packages/geo/1.1.0/aws-amplify-geo.min.js"
  integrity="sha384-TFMTyWuCbipXTzv0gzJbV8TPUupG1rA1AVrznAhCSpXTIdGw82bGd8RTk5rr3nP"
  crossorigin="anonymous" referrerpolicy="no-referrer"></script>
  <script src="https://cdn.amplify.aws/packages/maplibre-gl-js-
amplify/1.1.0/maplibre-gl-js-amplify.umd.min.js" integrity="sha384-7/
RxWonKW1nM9zCKiWU9x6bkQTjldosg0D1vZYm0Zj+K/vUSnA3s0Mh1RRWAtHPi" crossorigin="anonymous"
  referrerpolicy="no-referrer"></script>
<script>

```

```
// application-specific code
</script>
```

Dadurch wird eine leere Seite mit dem Container der Zuordnung erstellt.

Erstellen der Anwendung: Konfiguration

So konfigurieren Sie Ihre Anwendung mit JavaScript:

1. Geben Sie die Kennungen des nicht authentifizierten Identitätspools ein, den Sie unter [Verwenden von Karten – Schritt 2, Authentifizierung einrichten erstellt haben](#).

```
// Cognito Identity Pool ID
const identityPoolId = "region:identityPoolID"; // for example: us-
east-1:123example-1234-5678
// extract the Region from the Identity Pool ID
const region = identityPoolId.split(":")[0];
```

2. Konfigurieren Sie AWS Amplify für die Verwendung der von Ihnen erstellten Ressourcen, einschließlich des Identitätspools und der Map-Ressource (hier mit dem Standardnamen `dargestelltexplore.map`).

```
// Configure Amplify
const { Amplify } = aws_amplify_core;
const { createMap } = AmplifyMapLibre;

Amplify.configure({
  Auth: {
    identityPoolId,
    region,
  },
  geo: {
    AmazonLocationService: {
      maps: {
        items: {
          "explore.map": {
            style: "Default style"
          },
        },
        default: "explore.map",
      },
      region,
    },
  },
});
```

```
}  
});
```

Erstellen der Anwendung: Karteninitialisierung

Damit die Karte nach dem Laden der Seite angezeigt wird, müssen Sie die Karte initialisieren. Sie können die anfängliche Kartenposition anpassen, zusätzliche Steuerelemente hinzufügen und Daten überlagern.

```
async function initializeMap() {  
  const map = await createMap(  
    {  
      container: "map",  
      center: [-123.1187, 49.2819],  
      zoom: 10,  
      hash: true,  
    }  
  );  
  
  map.addControl(new maplibregl.NavigationControl(), "top-left");  
}  
  
initializeMap();
```

Note

Sie müssen für jeden Datenanbieter, den Sie verwenden, entweder in Ihrer Anwendung oder in Ihrer Dokumentation ein Wortzeichen oder eine Textzuweisung angeben. Attributionszeichenfolgen sind in der Style-Deskriptor-Antwort unter den `sources.grabmaptiles.attribution` Schlüsseln `sources.esri.attribution` `sources.here.attribution`, und enthalten. Amplify stellt automatisch eine Zuordnung bereit. Wenn Sie Amazon-Location-Ressourcen mit [Datenanbietern](#) verwenden, lesen Sie unbedingt die [Servicebedingungen](#).

Ausführen der Anwendung

Sie können diese Beispielanwendung ausführen, indem Sie sie auf einem lokalen Webserver verwenden oder sie in einem Browser öffnen.

Um einen lokalen Webserver zu verwenden, können Sie npx verwenden, das als Teil von Node.js installiert ist, oder einen anderen Webserver Ihrer Wahl. Um npx zu verwenden, geben Sie die URL `localhost:5000` ein.

Note

Wenn die Richtlinie, die Sie für Ihre nicht authentifizierte Amazon Cognito-Rolle erstellt haben, eine `referrer` Bedingung enthält, werden Sie möglicherweise daran gehindert, mit `localhost:5000` URLs zu testen. In diesem Fall können Sie mit einem Webserver testen, der eine URL bereitstellt, die sich in Ihrer Richtlinie befindet.

Nach Abschluss des Tutorials sieht die endgültige Anwendung wie im folgenden Beispiel aus.

```
<html>
  <head>
    <!-- CSS dependencies -->
    <link href="https://cdn.amplify.aws/packages/maplibre-gl/1.15.2/maplibre-gl.css" rel="stylesheet" integrity="sha384-DrPVD9GufrixGb7kWwRv0CywpXTmfvbK0Z5i5pN7urmIThew0zXKTME+gutUgtpeD" crossorigin="anonymous" referrerpolicy="no-referrer"></link>

    <!-- JavaScript dependencies -->
    <script src="https://cdn.amplify.aws/packages/maplibre-gl/1.15.2/maplibre-gl.js" integrity="sha384-rwYfkmA0pciZS2bDuwZ/Xa/Gog6jXem8D/whm3wnsZSVFemDDlprcUXHnDDUcrNU" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    <script src="https://cdn.amplify.aws/packages/core/4.3.0/aws-amplify-core.min.js" integrity="sha384-70h+5w0l7XGyYvSqBki2Q7SA5K640V5nyW2/LEbevDQEV1HMJqJLA1A00z2hu8fJ" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    <script src="https://cdn.amplify.aws/packages/auth/4.3.8/aws-amplify-auth.min.js" integrity="sha384-jfkXCEfYyVmDXyKlgWNwv54xRaZgk14m7sJeb2jLVBtUXCD2p+WU8YZ2mPZ9Xbdw" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    <script src="https://cdn.amplify.aws/packages/geo/1.1.0/aws-amplify-geo.min.js" integrity="sha384-TFMTyWuCbiptXTzv0gzJbV8TPUupG1rA1AVrznAhCSpXTIdGw82bGd8RTk5rr3nP" crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    <script src="https://cdn.amplify.aws/packages/maplibre-gl-js-amplify/1.1.0/maplibre-gl-js-amplify.umd.min.js" integrity="sha384-7/RxWonKW1nM9zCKiwU9x6bkQTjldosg0D1vZYm0Zj+K/vUSnA3s0MhlRRWAtHPi" crossorigin="anonymous" referrerpolicy="no-referrer"></script>

  </head>
  <body>
```

```
    body { margin: 0; }
    #map { height: 100vh; }
  </style>
</head>

<body>
  <div id="map" />
  <script type="module">
    // Cognito Identity Pool ID
    const identityPoolId = "region:identityPoolId"; // for example: us-
east-1:123example-1234-5678
    // extract the Region from the Identity Pool ID
    const region = identityPoolId.split(":")[0];

    // Configure Amplify
    const { Amplify } = aws_amplify_core;
    const { createMap } = AmplifyMapLibre;

    Amplify.configure({
      Auth: {
        identityPoolId,
        region,
      },
      geo: {
        AmazonLocationService: {
          maps: {
            items: {
              "explore.map": {
                style: "Default style"
              },
            },
            default: "explore.map",
          },
          region,
        },
      },
    });

    async function initializeMap() {
      const map = await createMap(
        {
          container: "map",
          center: [-123.1187, 49.2819],
          zoom: 10,
```

```
        hash: true,
      }
    );

    map.addControl(new maplibregl.NavigationControl(), "top-left");
  }

  initializeMap();
</script>
</body>
</html>
```

Wenn Sie diese Anwendung ausführen, wird eine Vollbildkarte mit dem von Ihnen gewählten Kartenstil angezeigt. Dieses Beispiel wird auch auf der Registerkarte Karte einbetten einer beliebigen Seite mit Kartenressourcen in der [Amazon Location Service-Konsole](#) beschrieben.

Nachdem Sie dieses Tutorial abgeschlossen haben, gehen Sie zum Thema [Eine Karte anzeigen](#) in der -AWS AmplifyDokumentation, um mehr zu erfahren, einschließlich der Anzeige von Markierungen auf der Karte.

Verwenden von Tangram mit Amazon Location Service

Dieser Abschnitt enthält die folgenden Tutorials zur Integration von Tangram in Amazon Location.

Important

Die Tangram-Stile in den folgenden Tutorials sind nur mit Amazon Location Map-Ressourcen kompatibel, die mit dem `-VectorHereContrast`Stil konfiguriert sind.

Im Folgenden finden Sie ein Beispiel für einen AWS -CLI-Befehl zum Erstellen einer neuen Kartenressource mit dem Namen *TangramExampleMap* unter Verwendung des *-VectorHereContrast*Stils:

```
aws --region us-east-1 \
  location \
  create-map \
  --map-name "TangramExampleMap" \
  --configuration "Style=VectorHereContrast"
```

Note

Die Fakturierung wird durch Ihre -Nutzung bestimmt. Möglicherweise fallen Gebühren für die Nutzung anderer -AWS-Services an. Weitere Informationen finden Sie unter [Amazon Location Service – Preise](#).

Themen

- [Verwenden von Tangram mit Amazon Location Service](#)
- [Verwenden von Tangram ES für Android mit Amazon Location Service](#)
- [Verwenden von Tangram ES für iOS mit Amazon Location Service](#)

Verwenden von Tangram mit Amazon Location Service

[Tangram](#) ist eine flexible Mapping-Engine, die für das Echtzeit-Rendering von 2D- und 3D-Karten aus Vektorkacheln entwickelt wurde. Es kann mit von Mapzen gestalteten Stilen und den HERE-Kacheln verwendet werden, die von der Amazon Location Service Maps API bereitgestellt werden. In diesem Handbuch wird beschrieben, wie Sie Tangram in Amazon Location in eine einfache HTML/JavaScript-Anwendung integrieren, obwohl dieselben Bibliotheken und Techniken auch bei der Verwendung von Frameworks wie React und Angular gelten.

Tangram wurde auf [Leaflet](#) erstellt, einer Open-Source- JavaScript Bibliothek für mobile interaktive Karten. Das bedeutet, dass viele Leaflet-kompatible Plug-Ins und Kontrollen auch mit Tangram funktionieren.

Tangram-Stile, die für die Arbeit mit dem [Tilezen-Schema](#) entwickelt wurden, sind weitgehend mit Amazon Location kompatibel, wenn Karten von HERE verwendet werden. Dazu zählen:

- [Blasenumbruch](#) – Ein voll funktionsfähiger Stil mit hilfreichen Symbolen für wichtige Punkte
- [Cinnabar](#) – Ein klassisches Erscheinungsbild für allgemeine Mapping-Anwendungen
- [Auffüllen](#) – Ein microSD-Kartenstil, der für Datenvisualisierungs-Überlagerungen entwickelt wurde und dem halben Toner-Stil von Stamen Design entspricht
- [Boln](#) – Eine Untersuchung von Skalierungstransformationen in der visuellen Sprache von TRON
- [Walkaround](#) – Ein auf Telefonie ausgerichteter Stil, der sich perfekt für Fußweg oder Fortfahren eignet

In diesem Handbuch wird beschrieben, wie Sie Tangram mit Amazon Location in einem einfachen HTML/einer einfachen JavaScript Anwendung integrieren, indem Sie den Tangram-Stil verwenden, der als „[Bulk Wrap](#)“ bezeichnet wird. Dieses Beispiel ist als Teil des Amazon Location Service-Beispiel-Repositoryys auf verfügbar [GitHub](#).

Obwohl andere Tangram-Stile am besten von Rasterkacheln begleitet werden, die Telefonieinformationen codieren, wird diese Funktion von Amazon Location noch nicht unterstützt.

Important

Die Tangram-Stile im folgenden Tutorial sind nur mit Amazon Location Map-Ressourcen kompatibel, die mit dem `-VectorHereContrastStil` konfiguriert sind.

Erstellen der Anwendung: Gerüsterstellung

Die Anwendung ist eine HTML-Seite mit JavaScript , um die Karte auf Ihrer Webanwendung zu erstellen. Erstellen Sie eine HTML-Seite (`index.html`) und erstellen Sie den Container der Zuordnung:

- Geben Sie ein `-divElement` mit einem `id` der Karte ein, um die Dimensionen der Karte auf die Kartenansicht anzuwenden.
- Die Dimensionen werden vom Ansichtsfenster geerbt.

```
<html>
  <head>
    <style>
      body {
        margin: 0;
      }

      #map {
        height: 100vh; /* 100% of viewport height */
      }
    </style>
  </head>
  <body>
    <!-- map container -->
    <div id="map" />
  </body>
```

```
</html>
```

Erstellen der Anwendung: Hinzufügen von Abhängigkeiten

Fügen Sie die folgenden Abhängigkeiten hinzu:

- Leaflet und das zugehörige CSS.
- Tangram.
- AWS SDK für JavaScript.

```
<!-- CSS dependencies -->
<link
  rel="stylesheet"
  href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"
  integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAsh0MAS6/keqq/
sMzMZ19scR4PsZChSR7A=="
  crossorigin=""
/>
<!-- JavaScript dependencies -->
<script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>
<script src="https://unpkg.com/tangram"></script>
<script src="https://sdk.amazonaws.com/js/aws-sdk-2.784.0.min.js"></script>
<script>
  // application-specific code
</script>
```

Dadurch wird eine leere Seite mit den erforderlichen Voraussetzungen erstellt. Der nächste Schritt führt Sie durch das Schreiben des JavaScript Codes für Ihre Anwendung.

Erstellen der Anwendung: Konfiguration

So konfigurieren Sie Ihre Anwendung mit Ihren Ressourcen und Anmeldeinformationen:

1. Geben Sie die Namen und Kennungen Ihrer Ressourcen ein.

```
// Cognito Identity Pool ID
const identityPoolId = "us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd";
// Amazon Location Service map name; must be HERE-backed
const mapName = "TangramExampleMap";
```

2. Instanzieren Sie einen Anmeldeinformationsanbieter mithilfe des nicht authentifizierten Identitätspools, den Sie unter [Verwenden von Karten – Schritt 2, Authentifizierung einrichten](#) erstellt haben. Da Anmeldeinformationen außerhalb des normalen AWS SDK-Workflows verwendet werden, laufen Sitzungen nach einer Stunde ab.

```
// extract the region from the Identity Pool ID; this will be used for both Amazon
  Cognito and Amazon Location
AWS.config.region = identityPoolId.split(":", 1)[0];

// instantiate a Cognito-backed credential provider
const credentials = new AWS.CognitoIdentityCredentials({
  IdentityPoolId: identityPoolId,
});
```

3. Mit Tangram können Sie zwar die URL(s) überschreiben, die zum Abrufen von Kacheln verwendet werden, aber es beinhaltet nicht die Möglichkeit, Anfragen abzufangen, damit sie signiert werden können.

Um dies zu umgehen, überschreiben Sie `sources.mapzen.url` um auf Amazon Location zu verweisen, indem Sie einen synthetischen Hostnamen verwenden `amazon.location`, der von einem [Service-Worker](#) bearbeitet wird. Im Folgenden finden Sie ein Beispiel für eine Szenenkonfiguration mit [dem Blasen-Wrap](#):

```
const scene = {
  import: [
    // Bubble Wrap style
    "https://www.nextzen.org/carto/bubble-wrap-style/10/bubble-wrap-style.zip",
    "https://www.nextzen.org/carto/bubble-wrap-style/10/themes/label-7.zip",
    "https://www.nextzen.org/carto/bubble-wrap-style/10/themes/bubble-wrap-road-shields-usa.zip",
    "https://www.nextzen.org/carto/bubble-wrap-style/10/themes/bubble-wrap-road-shields-international.zip",
  ],
  // override values beneath the `sources` key in the style above
  sources: {
    mapzen: {
      // point at Amazon Location using a synthetic URL, which will be handled by
      the service
      // worker
      url: `https://amazon.location/${mapName}/{z}/{x}/{y}`,
    },
  },
  // effectively disable raster tiles containing encoded normals
```

```
    normals: {
      max_zoom: 0,
    },
    "normals-elevation": {
      max_zoom: 0,
    },
  },
};
```

Erstellen der Anwendung: Anforderungstransformation

Um den Service-Worker zu registrieren und zu initialisieren, erstellen Sie eine `registerServiceWorker` Funktion, die aufgerufen werden soll, bevor die Zuordnung initialisiert wird. Dadurch wird der JavaScript Code registriert, der in einer separaten Datei mit dem Namen `sw.js` als Service-Worker bereitgestellt wird, der `steuertindex.html`.

Anmeldeinformationen werden aus Amazon Cognito geladen und zusammen mit der Region an den Service-Worker übergeben, um Informationen zum Signieren von Kachelanforderungen mit [Signature Version 4](#) bereitzustellen.

```
/**
 * Register a service worker that will rewrite and sign requests using Signature
 * Version 4.
 */
async function registerServiceWorker() {
  if ("serviceWorker" in navigator) {
    try {
      const reg = await navigator.serviceWorker.register("./sw.js");

      // refresh credentials from Amazon Cognito
      await credentials.refreshPromise();

      await reg.active.ready;

      if (navigator.serviceWorker.controller == null) {
        // trigger a navigate event to active the controller for this page
        window.location.reload();
      }

      // pass credentials to the service worker
      reg.active.postMessage({
        credentials: {
```

```

        accessKeyId: credentials.accessKeyId,
        secretAccessKey: credentials.secretAccessKey,
        sessionToken: credentials.sessionToken,
    },
    region: AWS.config.region,
  });
} catch (error) {
  console.error("Service worker registration failed:", error);
}
} else {
  console.warn("Service worker support is required for this example");
}
}

```

Die Service-Worker-Implementierung in `sw.js` lauscht auf `message` Ereignisse, um Konfigurationsänderungen an Anmeldeinformationen und Regionen aufzunehmen. Es fungiert auch als Proxy-Server, indem es auf `fetch` Ereignisse wartet. `fetch` Ereignisse, die auf den `amazon.location` synthetischen Hostnamen abzielen, werden neu geschrieben, um auf die entsprechende Amazon Location API abzielen und mit dem von Amplify Core signierten `Signer`.

```

// sw.js
self.importScripts(
  "https://unpkg.com/@aws-amplify/core@3.7.0/dist/aws-amplify-core.min.js"
);

const { Signer } = aws_amplify_core;

let credentials;
let region;

self.addEventListener("install", (event) => {
  // install immediately
  event.waitUntil(self.skipWaiting());
});

self.addEventListener("activate", (event) => {
  // control clients ASAP
  event.waitUntil(self.clients.claim());
});

self.addEventListener("message", (event) => {
  const {
    data: { credentials: newCredentials, region: newRegion },
  }

```

```
    } = event;

    if (newCredentials !== null) {
      credentials = newCredentials;
    }

    if (newRegion !== null) {
      region = newRegion;
    }
  });

  async function signedFetch(request) {
    const url = new URL(request.url);
    const path = url.pathname.slice(1).split("/");

    // update URL to point to Amazon Location
    url.pathname = `/maps/v0/maps/${path[0]}/tiles/${path.slice(1).join("/")}`;
    url.host = `maps.geo.${region}.amazonaws.com`;
    // strip params (Tangram generates an empty api_key param)
    url.search = "";

    const signed = Signer.signUrl(url.toString(), {
      access_key: credentials.accessKeyId,
      secret_key: credentials.secretAccessKey,
      session_token: credentials.sessionToken,
    });

    return fetch(signed);
  }

  self.addEventListener("fetch", (event) => {
    const { request } = event;

    // match the synthetic hostname we're telling Tangram to use
    if (request.url.includes("amazon.location")) {
      return event.respondWith(signedFetch(request));
    }

    // fetch normally
    return event.respondWith(fetch(request));
  });
```

Verwenden Sie die folgende Funktion in `index.html`, um Anmeldeinformationen automatisch zu erneuern und sie vor Ablauf an den Service-Worker zu `sendenindex.html`:

```
async function refreshCredentials() {
  await credentials.refreshPromise();

  if ("serviceWorker" in navigator) {
    const controller = navigator.serviceWorker.controller;

    controller.postMessage({
      credentials: {
        accessKeyId: credentials.accessKeyId,
        secretAccessKey: credentials.secretAccessKey,
        sessionToken: credentials.sessionToken,
      },
    });
  } else {
    console.warn("Service worker support is required for this example.");
  }

  // schedule the next credential refresh when they're about to expire
  setTimeout(refreshCredentials, credentials.expireTime - new Date());
}
```

Erstellen der Anwendung: Karteninitialisierung

Damit die Karte nach dem Laden der Seite angezeigt wird, müssen Sie die Karte initialisieren. Sie haben die Möglichkeit, die anfängliche Kartenposition anzupassen, zusätzliche Steuerelemente hinzuzufügen und Daten zu überlagern.

Note

Sie müssen für jeden Datenanbieter, den Sie verwenden, entweder in Ihrer Anwendung oder in Ihrer Dokumentation ein Wortzeichen oder eine Textzuweisung angeben. Attributionszeichenfolgen sind in der Style-Deskriptor-Antwort unter den `source.grabmaptiles.attribution` Schlüsseln `sources.esri.attributionsources.here.attribution`, und enthalten. Da Tangram diese Ressourcen nicht anfordert und nur mit Karten von HERE kompatibel ist, verwenden Sie „here 2020“. Wenn Sie Amazon-Location-Ressourcen mit [Datenanbietern](#) verwenden, lesen Sie unbedingt die [Servicebedingungen](#).

```
/**
 * Initialize a map.
 */
async function initializeMap() {
  // register the service worker to handle requests to https://amazon.location
  await registerServiceWorker();

  // Initialize the map
  const map = L.map("map").setView([49.2819, -123.1187], 10);
  Tangram.leafletLayer({
    scene,
  }).addTo(map);
  map.attributionControl.setPrefix("");
  map.attributionControl.addAttribution("© 2020 HERE");
}

initializeMap();
```

Ausführen der Anwendung

Um dieses Beispiel auszuführen, können Sie:

- Verwenden Sie einen Host, der HTTPS unterstützt,
- Verwenden Sie einen lokalen Webserver, um die Sicherheitseinschränkungen von Service-Workern einzuhalten.

Um einen lokalen Webserver zu verwenden, können Sie `npx` verwenden, da es als Teil von Node.js installiert ist. Sie können `npx serve` innerhalb desselben Verzeichnisses wie `index.html` und `verwendensw.js`. Dies dient der Anwendung auf localhost:5000.

Im Folgenden finden Sie die `index.html` Datei :

```
<!-- index.html -->
<html>
  <head>
    <link
      rel="stylesheet"
      href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"
      integrity="sha512-xodZBNTC5n17Xt2atTPuE1HxjVMSvLVW9ocqUKLsCC5CXdbqCmblAsh0MAS6/
      keqq/sMzMZ19scR4PsZChSR7A=="
      crossorigin=""
```

```
    />
  <style>
    body {
      margin: 0;
    }

    #map {
      height: 100vh;
    }
  </style>
</head>

<body>
  <div id="map" />
  <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>
  <script src="https://unpkg.com/tangram"></script>
  <script src="https://sdk.amazonaws.com/js/aws-sdk-2.784.0.min.js"></script>
  <script>
    // configuration
    // Cognito Identity Pool ID
    const identityPoolId = "<Identity Pool ID>";
    // Amazon Location Service Map name; must be HERE-backed
    const mapName = "<Map name>";

    AWS.config.region = identityPoolId.split(":")[0];

    // instantiate a credential provider
    credentials = new AWS.CognitoIdentityCredentials({
      IdentityPoolId: identityPoolId,
    });

    const scene = {
      import: [
        // Bubble Wrap style
        "https://www.nextzen.org/carto/bubble-wrap-style/10/bubble-wrap-style.zip",
        "https://www.nextzen.org/carto/bubble-wrap-style/10/themes/label-7.zip",
        "https://www.nextzen.org/carto/bubble-wrap-style/10/themes/bubble-wrap-road-shields-usa.zip",
        "https://www.nextzen.org/carto/bubble-wrap-style/10/themes/bubble-wrap-road-shields-international.zip",
      ],
      // override values beneath the `sources` key in the style above
      sources: {
        mapzen: {
```

```
    // point at Amazon Location using a synthetic URL, which will be handled by
the service
    // worker
    url: `https://amazon.location/${mapName}/{z}/{x}/{y}`,
  },
  // effectively disable raster tiles containing encoded normals
  normals: {
    max_zoom: 0,
  },
  "normals-elevation": {
    max_zoom: 0,
  },
},
};

/**
 * Register a service worker that will rewrite and sign requests using Signature
Version 4.
 */
async function registerServiceWorker() {
  if ("serviceWorker" in navigator) {
    try {
      const reg = await navigator.serviceWorker.register("./sw.js");

      // refresh credentials from Amazon Cognito
      await credentials.refreshPromise();

      await reg.active.ready;

      if (navigator.serviceWorker.controller == null) {
        // trigger a navigate event to active the controller for this page
        window.location.reload();
      }

      // pass credentials to the service worker
      reg.active.postMessage({
        credentials: {
          accessKeyId: credentials.accessKeyId,
          secretAccessKey: credentials.secretAccessKey,
          sessionToken: credentials.sessionToken,
        },
        region: AWS.config.region,
      });
    } catch (error) {
```

```
        console.error("Service worker registration failed:", error);
    }
    } else {
        console.warn("Service Worker support is required for this example");
    }
}

/**
 * Initialize a map.
 */
async function initializeMap() {
    // register the service worker to handle requests to https://amazon.location
    await registerServiceWorker();

    // Initialize the map
    const map = L.map("map").setView([49.2819, -123.1187], 10);
    Tangram.leafletLayer({
        scene,
    }).addTo(map);
    map.attributionControl.setPrefix("");
    map.attributionControl.addAttribution("© 2020 HERE");
}

initializeMap();
</script>
</body>
</html>
```

Im Folgenden finden Sie die `-sw.js` Datei:

```
// sw.js
self.importScripts(
    "https://unpkg.com/@aws-amplify/core@3.7.0/dist/aws-amplify-core.min.js"
);

const { Signer } = aws_amplify_core;

let credentials;
let region;

self.addEventListener("install", (event) => {
    // install immediately
    event.waitUntil(self.skipWaiting());
});
```

```
});

self.addEventListener("activate", (event) => {
  // control clients ASAP
  event.waitUntil(self.clients.claim());
});

self.addEventListener("message", (event) => {
  const {
    data: { credentials: newCredentials, region: newRegion },
  } = event;

  if (newCredentials !== null) {
    credentials = newCredentials;
  }

  if (newRegion !== null) {
    region = newRegion;
  }
});

async function signedFetch(request) {
  const url = new URL(request.url);
  const path = url.pathname.slice(1).split("/");

  // update URL to point to Amazon Location
  url.pathname = `/maps/v0/maps/${path[0]}/tiles/${path.slice(1).join("/")}`;
  url.host = `maps.geo.${region}.amazonaws.com`;
  // strip params (Tangram generates an empty api_key param)
  url.search = "";

  const signed = Signer.signUrl(url.toString(), {
    access_key: credentials.accessKeyId,
    secret_key: credentials.secretAccessKey,
    session_token: credentials.sessionToken,
  });

  return fetch(signed);
}

self.addEventListener("fetch", (event) => {
  const { request } = event;

  // match the synthetic hostname we're telling Tangram to use
```

```
if (request.url.includes("amazon.location")) {
    return event.respondWith(signedFetch(request));
}

// fetch normally
return event.respondWith(fetch(request));
});
```

Dieses Beispiel ist als Teil des Amazon Location Service-Beispiel-Repositorys auf verfügbar [GitHub](#).

Verwenden von Tangram ES für Android mit Amazon Location Service

[Tangram ES](#) ist eine C++-Bibliothek zum Rendern von 2D- und 3D-Zuweisungen aus Vektordaten mit OpenGL ES. Es ist das native Gegenstück zu [Tangram](#).

Tangram-Stile, die für die Arbeit mit dem [Tilezen-Schema](#) entwickelt wurden, sind weitgehend mit Amazon Location kompatibel, wenn Karten von HERE verwendet werden. Dazu zählen:

- [Blasen-Wrap](#) – Ein voll funktionsfähiger Stil mit hilfreichen Symbolen für wichtige Punkte.
- [Cinnabar](#) – Ein klassischer Look und Go-To für allgemeine Mapping-Anwendungen.
- [Auffüllen](#) – Ein microSD-Kartenstil, der für Datenvisualisierungs-Überlagerungen entwickelt wurde und vom halben Toner-Stil von Stamen Design passt.
- [Boln](#) – Eine Erkundung von Skalierungstransformationen in der visuellen Sprache von TRON.
- [Walkthrough](#) – Ein auf den Fuß gelegter Stil, der sich perfekt für Walk oder Fortfahren eignet.

In diesem Handbuch wird beschrieben, wie Sie Tangram ES für Android mit Amazon Location integrieren, indem Sie den Tangram-Stil namens Cinnabar verwenden. Dieses Beispiel ist als Teil des Amazon Location Service-Beispiel-Repositorys auf verfügbar [GitHub](#).

Obwohl andere Tangram-Stile am besten von Rasterkacheln begleitet werden, die Telefonieinformationen codieren, wird diese Funktion von Amazon Location noch nicht unterstützt.

Important

Die Tangram-Stile im folgenden Tutorial sind nur mit Amazon Location Map-Ressourcen kompatibel, die mit dem `-VectorHereContrast` Stil konfiguriert sind.

Erstellen der Anwendung: Initialisierung

So initialisieren Sie Ihre Anwendung:

1. Erstellen Sie ein neues Android Studio-Projekt aus der Vorlage Leere Aktivität.
2. Stellen Sie sicher, dass Kotlin für die Projektsprache ausgewählt ist.
3. Wählen Sie ein Mindest-SDK von API 16 aus: Android 4.1 (Jelly Bean) oder höher.
4. Öffnen Sie Projektstruktur, um Datei , Projektstruktur... auszuwählen, und wählen Sie den Abschnitt Abhängigkeiten aus.
5. Wenn <Alle Module> ausgewählt ist, wählen Sie die Schaltfläche +, um eine neue Bibliotheksabhängigkeit hinzuzufügen.
6. Fügen Sie AWS Android SDK Version 2.19.1 oder höher hinzu. Beispiel:
`com.amazonaws:aws-android-sdk-core:2.19.1`
7. Fügen Sie die Tangram-Version 0.13.0 oder höher hinzu. Zum Beispiel:
`com.mapzen.tangram:tangram:0.13.0`.

Note

Suche nach Tangram : `com.mapzen.tangram:tangram:0.13.0` generiert eine Meldung, dass sie „nicht gefunden“ ist, aber wenn Sie OK auswählen, kann sie hinzugefügt werden.

Erstellen der Anwendung: Konfiguration

So konfigurieren Sie Ihre Anwendung mit Ihren Ressourcen und Ihrer AWS Region:

1. Geben Sie einen Namen für den Benutzer ein und klicken Sie dann auf `app/src/main/res/values/configuration.xml`.
2. Geben Sie die Namen und Kennungen Ihrer Ressourcen sowie die AWS Region ein, in der sie erstellt wurden:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="identityPoolId">us-east-1:54f2ba88-9390-498d-aaa5-0d97fb7ca3bd</string>
  <string name="mapName">TangramExampleMap</string>
```

```
<string name="awsRegion">us-east-1</string>
<string name="sceneUrl">https://www.nextzen.org/carto/cinnabar-style/9/cinnabar-
style.zip</string>
<string name="attribution">© 2020 HERE</string>
</resources>
```

Erstellen der Anwendung: Aktivitätslayout

Bearbeiten von `app/src/main/res/layout/activity_main.xml`:

- Fügen Sie ein `hinzuMapView`, das die Zuordnung wiedergibt. Dadurch wird auch der anfängliche Mittelpunkt der Karte festgelegt.
- Fügen Sie ein `hinzuTextView`, das die Zuordnung anzeigt.

Dadurch wird auch der anfängliche Mittelpunkt der Karte festgelegt.

Note

Sie müssen für jeden Datenanbieter, den Sie verwenden, entweder in Ihrer Anwendung oder in Ihrer Dokumentation ein Wortzeichen oder eine Textzuweisung angeben. Attributionszeichenfolgen sind in der Style-Deskriptor-Antwort unter den `source.grabmaptiles.attribution` Schlüsselnamen `sources.esri.attribution`, `sources.here.attribution`, und enthalten. Da Tangram diese Ressourcen nicht anfordert und nur mit Karten von HERE kompatibel ist, verwenden Sie „here 2020“. Wenn Sie Amazon-Location-Ressourcen mit [Datenanbietern](#) verwenden, lesen Sie unbedingt die [Servicebedingungen](#).

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <com.mapzen.tangram.MapView
        android:id="@+id/map"
        android:layout_height="match_parent"
```

```
        android:layout_width="match_parent" />

    <TextView
        android:id="@+id/attributionView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#80808080"
        android:padding="5sp"
        android:textColor="@android:color/black"
        android:textSize="10sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        tools:ignore="SmallSp" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Erstellen der Anwendung: Anforderungstransformation

Erstellen Sie eine Klasse mit dem Namen `SigV4Interceptor`, um AWS Anfragen abzufangen und sie mit [Signature Version 4](#) zu signieren. Dies wird bei dem HTTP-Client registriert, der zum Abrufen von Kartenressourcen verwendet wird, wenn die Hauptaktivität erstellt wird.

```
package aws.location.demo.okhttp

import com.amazonaws.DefaultRequest
import com.amazonaws.auth.AWS4Signer
import com.amazonaws.auth.AWSCredentialsProvider
import com.amazonaws.http.HttpMethodName
import com.amazonaws.util.IOUtils
import okhttp3.HttpUrl
import okhttp3.Interceptor
import okhttp3.Request
import okhttp3.Response
import okio.Buffer
import java.io.ByteArrayInputStream
import java.net.URI

class SigV4Interceptor(
    private val credentialsProvider: AWSCredentialsProvider,
    private val serviceName: String
) : Interceptor {
    override fun intercept(chain: Interceptor.Chain): Response {
        val originalRequest = chain.request()
```

```
    if (originalRequest.url().host().contains("amazonaws.com")) {
        val signer = if (originalRequest.url().encodedPath().contains("@")) {
            // the presence of "@" indicates that it doesn't need to be double URL-
encoded
            AWS4Signer(false)
        } else {
            AWS4Signer()
        }

        val awsRequest = toAWSRequest(originalRequest, serviceName)
        signer.setServiceName(serviceName)
        signer.sign(awsRequest, credentialsProvider.credentials)

        return chain.proceed(toSignedOkHttpRequest(awsRequest, originalRequest))
    }

    return chain.proceed(originalRequest)
}

companion object {
    fun toAWSRequest(request: Request, serviceName: String): DefaultRequest<Any> {
        // clone the request (AWS-style) so that it can be populated with
credentials
        val dr = DefaultRequest<Any>(serviceName)

        // copy request info
        dr.httpMethod = HttpMethodName.valueOf(request.method())
        with(request.url()) {
            dr.resourcePath = uri().path
            dr.endpoint = URI.create("${scheme()}://${host()}")

            // copy parameters
            for (p in queryParameterNames()) {
                if (p != "") {
                    dr.addParameter(p, queryParameter(p))
                }
            }
        }

        // copy headers
        for (h in request.headers().names()) {
            dr.addHeader(h, request.header(h))
        }
    }
}
```

```
// copy the request body
val bodyBytes = request.body()?.let { body ->
    val buffer = Buffer()
    body.writeTo(buffer)
    IOUtils.toByteArray(buffer.inputStream())
}

dr.content = ByteArrayInputStream(bodyBytes ?: ByteArray(0))

return dr
}

fun toSignedOkHttpRequest(
    awsRequest: DefaultRequest<Any>,
    originalRequest: Request
): Request {
    // copy signed request back into an OkHttp Request
    val builder = Request.Builder()

    // copy headers from the signed request
    for ((k, v) in awsRequest.headers) {
        builder.addHeader(k, v)
    }

    // start building an HttpUrl
    val urlBuilder = HttpUrl.Builder()
        .host(awsRequest.endpoint.host)
        .scheme(awsRequest.endpoint.scheme)
        .encodedPath(awsRequest.resourcePath)

    // copy parameters from the signed request
    for ((k, v) in awsRequest.parameters) {
        urlBuilder.addQueryParameter(k, v)
    }

    return builder.url(urlBuilder.build())
        .method(originalRequest.method(), originalRequest.body())
        .build()
    }
}
```

Erstellen der Anwendung: Hauptaktivität

Die Hauptaktivität ist für die Initialisierung der Ansichten verantwortlich, die Benutzern angezeigt werden. Dazu gehören:

- Instanzieren eines Amazon Cognito-CredentialsProvider.
- Registrieren des Signature Version 4-Interceptors.
- Konfigurieren der Karte, indem sie auf einen Kartenstil verweist, Kachel-URLs überschreibt und die entsprechende Zuordnung anzeigt.

MainActivity ist auch für die Weiterleitung von Lebenszyklusereignissen an die Kartenansicht verantwortlich.

```
package aws.location.demo.tangram

import android.os.Bundle
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import aws.location.demo.okhttp.SigV4Interceptor
import com.amazonaws.auth.CognitoCachingCredentialsProvider
import com.amazonaws.regions.Regions
import com.mapzen.tangram.*
import com.mapzen.tangram.networking.DefaultHttpHandler
import com.mapzen.tangram.networking.HttpHandler

private const val SERVICE_NAME = "geo"

class MainActivity : AppCompatActivity(), MapView.MapReadyCallback {
    private var mapView: MapView? = null

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

        setContentView(R.layout.activity_main)

        mapView = findViewById(R.id.map)

        mapView?.getMapAsync(this, getHttpHandler())
        findViewById<TextView>(R.id.attributionView).text =
            getString(R.string.attribution)
    }
}
```

```
override fun onMapReady(mapController: MapController?) {
    val sceneUpdates = arrayListOf(
        SceneUpdate(
            "sources.mapzen.url",
            "https://maps.geo.#{getString(R.string.awsRegion)}.amazonaws.com/maps/
v0/maps/${
                getString(
                    R.string.mapName
                )
            }/tiles/{z}/{x}/{y}"
        )
    )

    mapController?.let { map ->
        map.updateCameraPosition(
            CameraUpdateFactory.newLngLatZoom(
                LngLat(-123.1187, 49.2819),
                12F
            )
        )
        map.loadSceneFileAsync(
            getString(R.string.sceneUrl),
            sceneUpdates
        )
    }
}

private fun getHttpHandler(): HttpHandler {
    val builder = DefaultHttpHandler.getClientBuilder()

    val credentialsProvider = CognitoCachingCredentialsProvider(
        applicationContext,
        getString(R.string.identityPoolId),
        Regions.US_EAST_1
    )

    return DefaultHttpHandler(
        builder.addInterceptor(
            SigV4Interceptor(
                credentialsProvider,
                SERVICE_NAME
            )
        )
    )
}
```

```
    )
}

override fun onResume() {
    super.onResume()
    mapView?.onResume()
}

override fun onPause() {
    super.onPause()
    mapView?.onPause()
}

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

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

Wenn Sie diese Anwendung ausführen, wird eine Vollbildkarte im Stil Ihrer Wahl angezeigt. Dieses Beispiel ist als Teil des Amazon Location Service-Beispiel-Repositorys auf verfügbar [GitHub](#).

Verwenden von Tangram ES für iOS mit Amazon Location Service

[Tangram ES](#) ist eine C++-Bibliothek zum Rendern von 2D- und 3D-Zuweisungen aus Vektordaten mit OpenGL ES. Es ist das native Gegenstück zu [Tangram](#).

Tangram-Stile, die für die Arbeit mit dem [Tilezen-Schema](#) entwickelt wurden, sind weitgehend mit Amazon Location kompatibel, wenn Karten von HERE verwendet werden. Dazu zählen:

- [Blasen-Wrap](#) – Ein voll funktionsfähiger Stil mit hilfreichen Symbolen für wichtige Punkte
- [Cinnabar](#) – Ein klassisches Erscheinungsbild für allgemeine Mapping-Anwendungen
- [Auffüllen](#) – Ein microSD-Kartenstil, der für Datenvisualisierungs-Überlagerungen entwickelt wurde und dem halben Toner-Stil von Stamen Design entspricht
- [Boln](#) – Eine Untersuchung von Skalierungstransformationen in der visuellen Sprache von TRON

- [Walkaround](#) – Ein auf den Fußabdruck ausgerichteter Stil, der sich perfekt zum Erkunden oder Verlassen von eignet

In diesem Handbuch wird beschrieben, wie Sie Tangram ES für iOS mit Amazon Location integrieren, indem Sie den Tangram-Stil namens Cinnabar verwenden. Dieses Beispiel ist als Teil des Amazon Location Service-Beispiel-Repositorys auf verfügbar [GitHub](#).

Obwohl andere Tangram-Stile am besten von Rasterkacheln begleitet werden, die Telefonieinformationen codieren, wird diese Funktion von Amazon Location noch nicht unterstützt.

 **Important**

Die Tangram-Stile im folgenden Tutorial sind nur mit Amazon Location Map-Ressourcen kompatibel, die mit dem `-VectorHereContrastStil` konfiguriert sind.

Erstellen der Anwendung: Initialisierung

So initialisieren Sie die Anwendung:

1. Erstellen Sie ein neues Xcode-Projekt aus der -App-Vorlage.
2. Wählen Sie SwiftUI als Schnittstelle aus.
3. Wählen Sie SwiftUI-Anwendung für ihre Lebensdauer aus.
4. Wählen Sie Swift als Sprache aus.

Erstellen der Anwendung: Abhängigkeiten hinzufügen

Um Abhängigkeiten hinzuzufügen, können Sie einen Abhängigkeitsmanager verwenden, z. B.

[CocoaPods](#):

1. Installieren Sie in Ihrem Terminal CocoaPods:

```
sudo gem install cocoapods
```

2. Navigieren Sie zum Projektverzeichnis Ihrer Anwendung und initialisieren Sie die Podfile mit dem CocoaPods Paketmanager:

```
pod init
```

- Öffnen Sie das Podfile, um AWSCore und Tangram-es als Abhängigkeiten hinzuzufügen:

```
platform :ios, '12.0'

target 'Amazon Location Service Demo' do
  use_frameworks!

  pod 'AWSCore'
  pod 'Tangram-es'
end
```

- Abhängigkeiten herunterladen und installieren:

```
pod install --repo-update
```

- Öffnen Sie den Xcode-Workspace, den CocoaPods erstellt hat:

```
xed .
```

Erstellen der Anwendung: Konfiguration

Fügen Sie die folgenden Schlüssel und Werte zu Info.plist hinzu, um die Anwendung zu konfigurieren und Telemetrie zu deaktivieren:

Schlüssel	Wert
AWSRegion	us-east-1
IdentityPoolId	us-east-1:54f2ba88-9390-498d-aaaa5-0d97fb7ca3bd
MapName	ExampleMap
SceneURL	https://www.nextzen.org/carto/cinnabar-style/9/cinnabar-style.zip

Erstellen der Anwendung: ContentView Layout

Um die Karte zu rendern, bearbeiten Sie ContentView.swift:

- Fügen Sie eine `hinzuMapView`, die die Zuordnung wiedergibt.
- Fügen Sie einen `hinzuTextField`, der die Zuordnung anzeigt.

Dadurch wird auch der anfängliche Mittelpunkt der Karte festgelegt.

Note

Sie müssen für jeden Datenanbieter, den Sie verwenden, entweder in Ihrer Anwendung oder in Ihrer Dokumentation ein Wortzeichen oder eine Textzuweisung angeben. Attributionszeichenfolgen sind in der Style-Deskriptor-Antwort unter den `source.grabmaptiles.attribution` Schlüsselnamen `sources.esri.attribution` und `sources.here.attribution`, und enthalten. Wenn Sie Amazon-Location-Ressourcen mit [Datenanbietern](#) verwenden, lesen Sie unbedingt die [Nutzungsbedingungen für -Services](#).

```
import SwiftUI
import TangramMap

struct ContentView: View {
    var body: some View {
        MapView()
            .cameraPosition(TGCameraPosition(
                center: CLLocationCoordinate2DMake(49.2819, -123.1187),
                zoom: 10,
                bearing: 0,
                pitch: 0))
            .edgesIgnoringSafeArea(.all)
            .overlay(
                Text("© 2020 HERE")
                    .disabled(true)
                    .font(.system(size: 12, weight: .light, design: .default))
                    .foregroundColor(.black)
                    .background(Color.init(Color.RGBColorSpace.sRGB, white: 0.5,
opacity: 0.5))
                    .cornerRadius(1),
                alignment: .bottomTrailing)
    }
}
```

```
struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```

Erstellen der Anwendung: Anforderungstransformation

Erstellen Sie eine neue Swift-Datei mit dem Namen `AWSSignatureV4URLHandler.swift`, die die folgende Klassendefinition enthält, um AWS Anforderungen abzufangen und sie mit [Signature Version 4](#) zu signieren. Dies wird als URL-Handler innerhalb des `Tangram` registriert `MapView`.

```
import AWSCore
import TangramMap

class AWSSignatureV4URLHandler: TGDefaultURLHandler {
    private let region: AWSRegionType
    private let identityPoolId: String
    private let credentialsProvider: AWSCredentialsProvider

    init(region: AWSRegionType, identityPoolId: String) {
        self.region = region
        self.identityPoolId = identityPoolId
        self.credentialsProvider = AWSCognitoCredentialsProvider(regionType: region,
identityPoolId: identityPoolId)
        super.init()
    }

    override func downloadRequestAsync(_ url: URL, completionHandler: @escaping
TGDownloadCompletionHandler) -> UInt {
        if url.host?.contains("amazonaws.com") != true {
            // not an AWS URL
            return super.downloadRequestAsync(url, completionHandler:
completionHandler)
        }

        // URL-encode spaces, etc.
        let keyPath = String(url.path.dropFirst())
        guard let keyPathSafe =
keyPath.addingPercentEncoding(withAllowedCharacters: .urlPathAllowed) else {
            print("Invalid characters in path '\(keyPath)'; unsafe to sign")
        }
    }
}
```

```

        return super.downloadRequestAsync(url, completionHandler:
completionHandler)
    }

    // sign the URL
    let endpoint = AWSEndpoint(region: region, serviceName: "geo", url: url)
    let requestHeaders: [String: String] = ["host": endpoint!.hostName]
    let task = AWSSignatureV4Signer
        .generateQueryStringForSignatureV4(
            withCredentialProvider: credentialsProvider,
            httpMethod: .GET,
            expireDuration: 60,
            endpoint: endpoint!,
            keyPath: keyPathSafe,
            requestHeaders: requestHeaders,
            requestParameters: .none,
            signBody: true)
    task.waitUntilFinished()

    if let error = task.error as NSError? {
        print("Error occurred: \(error)")
    }

    if let result = task.result {
        // have Tangram fetch the signed URL
        return super.downloadRequestAsync(result as URL, completionHandler:
completionHandler)
    }

    // fall back to an unsigned URL
    return super.downloadRequestAsync(url, completionHandler: completionHandler)
}
}

```

Erstellen der Anwendung: Kartenansicht

Die Kartenansicht ist für die Initialisierung einer Instance von `AWSSignatureV4Delegate` und die Konfiguration der zugrunde liegenden verantwortlich `MGLMapView`, die Ressourcen abrufen und die Karte rendert. Außerdem übernimmt es die Weitergabe von Attributionszeichenfolgen aus der Quelle des Stildeskriptors zurück an die `ContentView`.

Erstellen Sie eine neue Swift-Datei mit dem Namen `MapView.swift`, die die folgende `struct` Definition enthält:

```
import AWSCore
import TangramMap
import SwiftUI

struct MapView: UIViewRepresentable {
    private let mapView: TGMapView

    init() {
        let regionName = Bundle.main.object(forKey: "AWSRegion") as!
String
        let identityPoolId = Bundle.main.object(forKey: "IdentityPoolId")
as! String
        let mapName = Bundle.main.object(forKey: "MapName") as! String
        let sceneURL = URL(string: Bundle.main.object(forKey: "SceneURL")
as! String)!

        let region = (regionName as NSString).aws_regionTypeValue()

        // rewrite tile URLs to point at AWS resources
        let sceneUpdates = [
            TGSceneUpdate(path: "sources.mapzen.url",
                value: "https://maps.geo.\(regionName).amazonaws.com/maps/v0/
maps/\(mapName)/tiles/{z}/{x}/{y}")]

        // instantiate a TGURLHandler that will sign AWS requests
        let urlHandler = AWSSignatureV4URLHandler(region: region, identityPoolId:
identityPoolId)

        // instantiate the map view and attach the URL handler
        mapView = TGMapView(frame: .zero, urlHandler: urlHandler)

        // load the map style and apply scene updates (properties modified at runtime)
        mapView.loadScene(from: sceneURL, with: sceneUpdates)
    }

    func cameraPosition(_ cameraPosition: TGCameraPosition) -> MapView {
        mapView.cameraPosition = cameraPosition

        return self
    }

    // MARK: - UIViewRepresentable protocol
```

```
func makeUIView(context: Context) -> TGMMapView {
    return mapView
}

func updateUIView(_ uiView: TGMMapView, context: Context) {
}
}
```

Wenn Sie diese Anwendung ausführen, wird eine Vollbildkarte im Stil Ihrer Wahl angezeigt. Dieses Beispiel ist als Teil des Amazon Location Service-Beispiel-Repositorys auf verfügbar [GitHub](#).

Zeichnen von Datenfunktionen auf einer Karte

Nachdem Sie über eine Anwendung verfügen, die eine Karte mithilfe von Amplify oder Tangram rendert MapLibre, besteht ein natürlicher nächster Schritt darin, Features über die Karte zu zeichnen. Möglicherweise möchten Sie Ihre Kundenstandorte als Markierungen auf der Karte rendern.

Im Allgemeinen können Sie die [Ortssuchfunktionen](#) verwenden, um Standorte aus Ihren Daten zu finden, und dann die Funktionalität von Amplify oder Tangram verwenden MapLibre, um die Standorte zu rendern.

Beispiele für das Rendern verschiedener Objekttypen auf der Karte finden Sie in den folgenden MapLibre Beispielen:

- [Beispiel: Grafikmarkierungen](#)
- [Beispiel: Zeichnen von gruppierten Punkten](#)
- [Beispiel: Zeichnen eines Polygons](#)

Weitere Beispiele und Tutorials finden Sie unter [Codebeispiele und Tutorials für die Arbeit mit Amazon Location Service](#).

Festlegen von Erweiterungen für eine Zuordnung mit MapLibre

Es gibt Situationen, in denen Sie nicht möchten, dass Ihre Benutzer die gesamte Welt schwenken oder zoomen können. Wenn Sie MapLibredas Kartensteuerelement von verwenden, können Sie die Ausmaße des Kartensteuerelements mit der `maxBounds` Option einschränken und den Zoom mit den `maxZoom` Optionen `minZoom` und einschränken.

Das folgende Codebeispiel zeigt, wie das Kartensteuerelement initialisiert wird, um das Panning auf eine bestimmte Grenze zu beschränken (in diesem Fall die Ausmaße der Grab-Datenquelle).

Note

Diese Beispiele befinden sich in JavaScript und funktionieren im Kontext des [Erstellen einer Webanwendung](#) Tutorials.

```
// Set bounds to Grab data provider region
var bounds = [
    [90.0, -21.943045533438166], // Southwest coordinates
    [146.25, 31.952162238024968] // Northeast coordinates
];

var mglMap = new maplibregl.Map(
    {
        container: 'map',
        style: mapName,
        maxBounds: bounds // Sets bounds as max
        transformRequest,
    }
);
```

Ebenso können Sie eine minimale und maximale Zoomstufe für die Karte festlegen. Die Werte für beide Werte können zwischen 0 und 24 liegen, obwohl die Standardwerte für den minimalen Zoom 0 und für den maximalen Zoom 22 sind (Datenanbieter stellen möglicherweise nicht auf allen Zoomstufen Daten bereit. Die meisten Kartenbibliotheken behandeln dies automatisch). Im folgenden Beispiel werden die maxZoom Optionen minZoom und für das Map MapLibre -Steuerelement initialisiert.

```
// Set the minimum and maximum zoom levels
var mglMap = new maplibregl.Map(
    {
        container: 'map',
        style: mapName,
        maxZoom: 12,
        minZoom: 5,
        transformRequest,
    }
);
```

i Tip

Die MapLibre Kartensteuerung ermöglicht auch das Festlegen dieser Optionen zur Laufzeit und nicht während der Initialisierung mit den `set . . .` Funktionen `get . . .` und `.` Verwenden Sie beispielsweise `getMaxBounds` und `, setMaxBounds` um die Kartengrenzen zur Laufzeit zu ändern.

Verwalten Ihrer Kartenressourcen

Sie können Ihre Kartenressourcen mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs verwalten.

Auflisten von Kartenressourcen

Sie können eine Liste Ihrer Kartenressourcen über die Amazon-Standortkonsole, die AWS CLI oder die Amazon-Standort-APIs anzeigen.

Console

So zeigen Sie eine Liste der vorhandenen Kartenressourcen mit der Amazon-Standortkonsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich `Maps` aus.
3. Zeigen Sie eine Liste Ihrer Kartenressourcen unter `Meine Karten` an.

API

Verwenden Sie die [-ListMaps](#) Operation aus den Amazon Location Maps APIs.

Das folgende Beispiel ist eine API-Anforderung zum Abrufen einer Liste von Kartenressourcen im AWS Konto.

```
POST /maps/v0/list-maps
```

Im Folgenden finden Sie eine Beispielantwort für [ListMaps](#):

```
{
  "Entries": [
    {
```

```
    "CreateTime": 2020-10-30T01:38:36Z,  
    "DataSource": "Esri",  
    "Description": "string",  
    "MapName": "ExampleMap",  
    "UpdateTime": 2020-10-30T01:38:36Z  
  }  
],  
"NextToken": "1234-5678-9012"  
}
```

CLI

Verwenden Sie den [list-map](#)-Befehl.

Das folgende Beispiel ist ein AWS CLI, um eine Liste von Kartenressourcen im AWS Konto abzurufen.

```
aws location list-maps
```

Abrufen von Kartenressourcendetails

Sie können Details zu jeder Kartenressource in Ihrem AWS-Konto über die Amazon Location-Konsole, die AWS CLI oder die Amazon Location APIs abrufen.

Console

So zeigen Sie die Details einer Kartenressource mit der Amazon Location-Konsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich **Maps** aus.
3. Wählen Sie unter **Meine Karten** den Namenslink der Zielkartenressource aus.

API

Verwenden Sie die [DescribeMap](#) Operation aus den Amazon Location Maps APIs .

Das folgende Beispiel ist eine API-Anforderung zum Abrufen der Kartenressourcendetails für *ExampleMap*.

```
GET /maps/v0/maps/ExampleMap
```

Im Folgenden finden Sie eine Beispielantwort für [DescribeMap](#):

```
{
  "Configuration": {
    "Style": "VectorEsriNavigation"
  },
  "CreateTime": "2020-10-30T01:38:36Z",
  "DataSource": "Esri",
  "Description": "string",
  "MapArn": "arn:aws:geo:us-west-2:123456789012:maps/ExampleMap",
  "MapName": "ExampleMap",
  "Tags": {
    "Tag1" : "Value1"
  },
  "UpdateTime": "2020-10-30T01:40:36Z"
}
```

CLI

Verwenden Sie den [describe-map](#)-Befehl.

Das folgende Beispiel ist ein AWS CLI, um die Kartenressourcendetails für *ExampleMap* abzurufen.

```
aws location describe-map \
  --map-name "ExampleMap"
```

Löschen einer Kartenressource

Sie können eine Kartenressource aus Ihrem AWS Konto mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs löschen.

Warning

Dieser Vorgang löscht die Ressource dauerhaft.

Console

So löschen Sie eine vorhandene Kartenressource mithilfe der Amazon Location-Konsole

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Maps aus.
3. Wählen Sie unter Liste Meine Karten die Zielkarte aus der Liste aus.
4. Wählen Sie Zuordnung löschen aus.

API

Verwenden Sie die [DeleteMap](#) Operation aus den Amazon Location Maps APIs .

Das folgende Beispiel ist eine API-Anforderung zum Löschen der Kartenressource *ExampleMap*.

```
DELETE /maps/v0/maps/ExampleMap
```

Im Folgenden finden Sie ein Beispiel für eine Erfolgsantwort für [DeleteMap](#):

```
HTTP/1.1 200
```

CLI

Verwenden Sie den [delete-map](#)-Befehl.

Das folgende Beispiel ist ein -AWS CLIBefehl zum Löschen der Kartenressource *ExampleMap*.

```
aws location delete-map \  
  --map-name "ExampleMap"
```

Suchen von Orts- und Geolocation-Daten mit Amazon Location

Amazon Location bietet die Möglichkeit, die Geolocation-Daten des von Ihnen ausgewählten Anbieters zu durchsuchen oder zu platzieren. Es gibt verschiedene Arten von Suchen.

- Geokodierung – Geokodierung ist der Prozess der Suche nach Adressen, Regionen, Geschäftsnamen oder anderen Interessenpunkten, basierend auf Texteingabe. Es gibt Details und die Position (in Breiten- und Längengrad) der gefundenen Ergebnisse zurück.
- Umgekehrte Geokodierung – Mit der umgekehrten Geokodierung können Sie Stellen in der Nähe eines bestimmten Standorts finden.

- **Autocomplete** – Autocomplete ist der Prozess, bei dem automatische Vorschläge gemacht werden, wie die Benutzer in einer Abfrage eingeben. Wenn sie beispielsweise **Par** einen Vorschlag eingeben, könnte er sein **Paris, France**.

Mit Amazon Location können Sie einen Datenanbieter für Ortssuchvorgänge auswählen, indem Sie eine Ortsindexressource erstellen und konfigurieren.

Sobald Sie Ihre Ressource erstellt haben, können Sie Anfragen mit dem AWS SDK für Ihre bevorzugte Sprache, Amplify oder die REST-API-Endpunkte senden. Sie können Daten aus der Antwort verwenden, um Standorte auf einer Karte zu markieren, Positionsdaten anzureichern und Positionen in für Menschen lesbaren Text umzuwandeln.

Note

Eine Übersicht über die Konzepte für die Suche nach Orten finden Sie unter [Ortssuche](#).

Themen

- [Voraussetzungen](#)
- [Geokodierung](#)
- [Geokodierung umkehren](#)
- [AutoVervollständigen](#)
- [Verwenden von Orts-IDs](#)
- [Kategorien und Filterergebnisse platzieren](#)
- [Amazon Aurora PostgreSQL Benutzerdefinierte Funktionen für Amazon Location Service](#)
- [Verwalten Ihrer Ortsindexressourcen](#)

Voraussetzungen

Bevor Sie mit der Geokodierung, der umgekehrten Geokodierung oder der Suche nach Orten beginnen, führen Sie die erforderlichen Schritte aus:

Themen

- [Erstellen einer Ortsindexressource](#)
- [Authentifizieren Ihrer Anforderungen](#)

Erstellen einer Ortsindexressource

Erstellen Sie zunächst eine Ortsindex-Ressource in Ihrem AWS Konto.

Wenn Sie eine Ortsindexressource erstellen, können Sie aus den Datenanbietern wählen, die zur Unterstützung von Abfragen für Geokodierung, Reverse-Geokodierung und Suchen verfügbar sind:

1. Esri – Weitere Informationen zur Abdeckung von Esri in Ihrer relevanten Region finden Sie unter [Abdeckung der Esri-Geocodierung](#) in der Esri-Dokumentation.
2. HERE Technologies – Weitere Informationen zur Abdeckung von HERE in Ihrer relevanten Region finden Sie unter [Abdeckung der HERE-Geokodierung](#) in der HERE-Dokumentation.
3. Grab – Grab stellt Daten nur für Südostasien bereit. Weitere Informationen zur Abdeckung von Grab finden Sie unter [Abgedeckte Länder/Regionen und Gebiete](#) in diesem Leitfaden.

Sie können dies über die Amazon Location Service-Konsole, die AWS CLI oder die Amazon Location APIs tun.

Console

So erstellen Sie eine Ortsindexressource mit der Amazon Location Service-Konsole

1. Öffnen Sie die Amazon Location Service-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Indizes platzieren aus.
3. Wählen Sie Ortsindex erstellen aus.
4. Füllen Sie die folgenden Felder aus:
 - Name – Geben Sie einen Namen für die Ortsindex-Ressource ein. Zum Beispiel *ExamplePlaceIndex*. Maximal 100 Zeichen. Gültige Einträge umfassen alphanumerische Zeichen, Bindestriche, Punkte und Unterstriche.
 - Beschreibung – Geben Sie eine optionale Beschreibung ein.
5. Wählen Sie unter Datenanbieter einen verfügbaren [Datenanbieter](#) aus, der mit Ihrer Ortsindexressource verwendet werden soll.

Note

Wenn Ihre Anwendung Assets, die Sie in Ihrem Unternehmen verwenden, wie z. B. Lieferfahrzeuge oder Mitarbeiter, verfolgt oder weiterleitet, dürfen Sie Esri nicht als

Geolokalisierungsanbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS -Servicebedingungen](#).

6. Geben Sie unter Datenspeicheroptionen an, ob Sie Suchergebnisse aus Ihrer Ortsindexressource speichern möchten.
7. (Optional) Geben Sie unter Markierungen einen Tag Key (Schlüssel) und einen Value (Wert) ein. Dadurch wird ein Tag für Ihre neue Ortsindexressource hinzugefügt. Weitere Informationen finden Sie unter [Markieren Ihrer Ressourcen](#).
8. Wählen Sie Ortsindex erstellen aus.

API

So erstellen Sie eine Ortsindex-Ressource mithilfe der Amazon Location APIs

Verwenden Sie die [CreatePlaceIndex](#) Operation aus den Amazon Location Places APIs .

Das folgende Beispiel ist eine API-Anforderung zum Erstellen einer Ortsindexressource mit dem Namen *ExamplePlaceIndex* unter Verwendung des Datenanbieters *Esri* .

```
POST /places/v0/indexes
Content-type: application/json

{
  "DataSource": "Esri",
  "DataSourceConfiguration": {
    "IntendedUse": "SingleUse"
  },
  "Description": "string",
  "IndexName": "ExamplePlaceIndex",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

AWS CLI

So erstellen Sie eine Ortsindex-Ressource mit AWS CLI -Befehlen

Verwenden Sie den [create-place-index](#)-Befehl.

Im folgenden Beispiel wird eine Ortsindexressource namens *ExamplePlaceIndex* mit *Esri* als Datenanbieter erstellt.

```
aws location \
  create-place-index \
  --data-source "Esri" \
  --description "Example place index" \
  --index-name "ExamplePlaceIndex" \
  --tags Tag1=Value1
```

Note

Die Fakturierung hängt von Ihrer Nutzung ab. Möglicherweise fallen Gebühren für die Nutzung anderer AWS -Services an. Weitere Informationen finden Sie unter [Amazon Location Service – Preise](#).

Authentifizieren Ihrer Anforderungen

Sobald Sie eine Ortsindexressource erstellt haben und bereit sind, Standortfunktionen in Ihrer Anwendung zu erstellen, wählen Sie aus, wie Sie Ihre Anfragen authentifizieren würden:

- Informationen dazu, wie Sie auf die Services zugreifen können, finden Sie unter [Zugriff auf Amazon Location Service](#).
- Wenn Sie eine Website mit anonymen Benutzern haben, können Sie API-Schlüssel oder Amazon Cognito verwenden.

Beispiel

Das folgende Beispiel zeigt die Verwendung eines API-Schlüssels für die Autorisierung mit [AWS JavaScript SDK v3](#) und dem Amazon Location [Authentifizierungshelfer](#).

```
import { LocationClient, SearchPlaceIndexForTextCommand } from "@aws-sdk/client-location";
import { withAPIKey } from "@aws/amazon-location-utilities-auth-helper";

const apiKey = "v1.public.your-api-key-value"; // API key

// Create an authentication helper instance using an API key
```

```
const authHelper = await withAPIKey(apiKey);

const client = new LocationClient({
  region: "<region>", // region containing Cognito pool
  ...authHelper.getLocationClientConfig(), // Provides configuration required to make
  requests to Amazon Location
});

const input = {
  IndexName: "ExamplePlaceIndex",
  Text: "Anyplace",
  BiasPosition: [-123.4567, 45.6789]
};

const command = new SearchPlaceIndexForTextCommand(input);

const response = await client.send(command);
```

Geokodierung

Geokodierung ist ein Prozess, der Text, z. B. eine Adresse, eine Region, einen Geschäftsnamen oder einen Interessenpunkt, in eine Reihe geografischer Koordinaten umwandelt. Sie können Ortsindexressourcen verwenden, um Geocodierungsanforderungen zu senden und Daten, die aus der Geocodierung abgerufen wurden, zu integrieren, um Daten auf einer Karte für Ihre Web- oder mobile Anwendung anzuzeigen.

Dieser Abschnitt führt Sie durch das Senden einer einfachen Geocodierungsanforderung und das Senden von Geocodierungsanfragen mit optionalen Spezifikationen.

Geokodierung

Sie können eine einfache Anfrage an Geocode senden, indem Sie die [-SearchPlaceIndexForText](#) Operation verwenden, um eine Adresse in eine Reihe von Koordinaten zu konvertieren. Eine einfache -Anforderung enthält den folgenden erforderlichen Parameter:

- Text – Eine Adresse, ein Name, eine Stadt oder eine Region, die in eine Reihe von Koordinaten konvertiert werden soll. Zum Beispiel die Zeichenfolge Any Town.

Verwenden Sie den folgenden optionalen Parameter, um eine maximale Anzahl von Ergebnissen pro Seite anzugeben:

- `MaxResults` – Beschränkt die maximale Anzahl der Ergebnisse, die in der Abfrageantwort zurückgegeben werden.

Sie können die AWS CLI oder die Amazon Location APIs verwenden.

API

Das folgende Beispiel ist eine [SearchPlaceIndexForText](#) Anforderung, die Ortsindex-Ressource *ExamplePlaceIndex* nach einer Adresse, einem Namen, einer Stadt oder einer Region namens *Beliebige Stadt zu* durchsuchen.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Any Town",
  "MaxResults": 10
}
```

AWS CLI

Das folgende Beispiel ist ein [search-place-index-for-text](#) Befehl, um die Ortsindex-Ressource *ExamplePlaceIndex* nach einer Adresse, einem Namen, einer Stadt oder einer Region namens *Beliebige Stadt zu* durchsuchen.

```
aws location \
  search-place-index-for-text \
    --index-name ExamplePlaceIndex \
    --text "Any Town" \
    --max-results 10
```

Geocode nahe einer Position

Bei der Geokodierung können Sie eine bestimmte Position mit dem folgenden optionalen Parameter geocodieren:

- **BiasPosition** – Die Position, an der Sie in der Nähe suchen möchten. Dadurch wird Ihre Suche eingeschränkt, indem nach Ergebnissen gesucht wird, die der angegebenen Position am nächsten liegen. Definiert als [*longitude*, *latitude*]

Das folgende Beispiel ist eine [SearchPlaceIndexForText](#) Anforderung, um die Ortsindexressource nach einer Adresse, einem Namen, einer Stadt oder einer Region namens *Any Town* in der Nähe der Position [*-123.4567,45,6789*] zu durchsuchen.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Any Town",
  "BiasPosition": [-123.4567,45.6789]
}
```

Geocode innerhalb eines Begrenzungsrahmens

Mit dem folgenden optionalen Parameter können Sie innerhalb eines Begrenzungsrahmens Geocodieren, um Ihre Ergebnisse auf Koordinaten innerhalb einer bestimmten Grenze einzugrenzen:

- **FilterBBox** – Ein Begrenzungsrahmen, den Sie angeben, um Ihre Ergebnisse nach Koordinaten innerhalb der Grenzen des Rahmens zu filtern. Definiert als [*LongitudeSW*, *LatitudeSW*, *LongitudeNE*, *LatitudeNE*]

Note

Eine Anforderung darf nicht sowohl die **FilterBBox** **BiasPosition** Parameter als auch enthalten. Die Angabe beider Parameter in der Anforderung gibt einen **ValidationException** Fehler zurück.

Das folgende Beispiel ist eine [SearchPlaceIndexForText](#) Anforderung, innerhalb eines Begrenzungsrahmens nach einer Adresse, einem Namen, einer Stadt oder einer Region namens *Beliebige Stadt* zu suchen. Der Begrenzungsrahmen folgt:

- Der Längengrad der Bolwest-Ecke beträgt *-124.1450*.
- Der Breitengrad der Westecke beträgt *41,7045*.

- Der Längengrad der nordöstlichen Ecke beträgt **-124.1387**.
- Der Breitengrad der nordöstlichen Ecke beträgt **41,7096**.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json
```

```
{
  "Text": "Any Town",
  "FilterBBox": [
    -124.1450, 41.7045,
    -124.1387, 41.7096
  ]
}
```

Geocode innerhalb eines Landes

Sie können mit dem folgenden optionalen Parameter innerhalb eines oder mehrerer Länder, die Sie angeben, geokodieren:

- **FilterCountries** – Das Land oder die Region, in dem bzw. der Sie Geocode erstellen möchten. Sie können bis zu 100 Länder in einer Anfrage mit einem dreistelligen [ISO 3166](#)-Ländercode definieren. Verwenden Sie beispielsweise AUS für Australien.

Das folgende Beispiel ist eine [SearchPlaceIndexForText](#) Anfrage zur Suche nach einer Adresse, einem Namen, einer Stadt oder einer Region namens *Beliebige Stadt* in Deutschland und Frankreich.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json
```

```
{
  "Text": "Any Town",
  "FilterCountries": ["DEU", "FRA"]
}
```

Filtern nach Kategorien

Sie können die Kategorien filtern, die in Ihrer Geocode-Anforderung zurückgegeben werden, indem Sie den folgenden optionalen Parameter verwenden:

- `FilterCategories` – Die Ergebniskategorien, die in Ihrer Abfrage zurückgegeben werden sollen. Sie können bis zu 5 Kategorien in einer Anforderung angeben. Die Liste der Amazon Location Service-Kategorien finden Sie im Abschnitt [Kategorien](#). Sie können beispielsweise angeben, `Hotel` um nur Rückgabe-Ensembles in Ihrer Abfrage anzugeben.

Das folgende Beispiel ist eine [SearchPlaceIndexForText](#) Anfrage, in den USA nach einem Lebensmittelgeschäft namens *Home* zu suchen.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Hometown Coffee",
  "FilterCategories": ["Coffee Shop"],
  "FilterCountries": ["USA"]
}
```

Weitere Informationen zum Filtern nach Kategorien finden Sie unter [Kategorien und Filterergebnisse platzieren](#)

Geocode in einer bevorzugten Sprache

Sie können eine Sprachpräferenz für Ergebnisse Ihrer Suche festlegen, indem Sie den optionalen `Language` Parameter verwenden. Beispielsweise **100 Main St, Anytown, USA** kann eine Suche nach *100 Main St, Any Town, USA* standardmäßig zurückgeben. Wenn Sie jedoch für `as` als `auswählenLanguage`, können die Ergebnisse *100 Rue Principale, Any Town, États-Unis* stattdessen zurückgegeben werden.

- `Language` – Ein Sprachcode, der zum Rendern der Ergebnisse Ihrer Abfrage verwendet werden soll. Der Wert muss ein gültiger [BCP-47](#)-Sprachcode sein. Zum Beispiel `en` für Englisch.

Note

Wenn nicht angegeben `Language` ist oder die angegebene Sprache für ein Ergebnis nicht unterstützt wird, wird die Standardsprache des Partners für dieses Ergebnis verwendet.

Das folgende Beispiel ist eine `SearchPlaceIndexForText` Anforderung zur Suche nach einem Ort namens **Any Town** mit der bevorzugten Sprache, die als angegeben istde.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json
{
  "Text": "Any Town",
  "Language": "de"
}
```

Beispielantwort

Example

Im Folgenden finden Sie eine Beispielantwort, wenn Sie die [-SearchPlaceIndexForText](#) Operation über die Amazon Location Places APIs aufrufen. Die Ergebnisse umfassen relevante [Stellen](#) und die [Anforderungszusammenfassung](#) . Basierend auf der Auswahl von Esri oder HERE als Partner werden zwei Antworten angezeigt.

Example request

```
POST /places/v0/indexes/ExamplePlaceIndex/search/text
Content-type: application/json

{
  "Text": "Amazon",
  "MaxResults": 1,
  "FilterCountries": ["USA"],
  "BiasPosition": [-112.10, 46.32]
}
```

Example response (Esri)

```
{
  "Results": [
    {
      "Place": {
        "Country": "USA",
        "Geometry": {
          "Point": [
            -112.10667999999998,
```

```

        46.319090000000074
      ]
    },
    "Interpolated": false,
    "Label": "Amazon, MT, USA",
    "Municipality": "Amazon",
    "Region": "Montana",
    "SubRegion": "Jefferson County"
  },
  "Distance": 523.4619749879726,
  "Relevance": 1
}
],
"Summary": {
  "BiasPosition": [
    -112.1,
    46.32
  ],
  "DataSource": "Esri",
  "FilterCountries": [
    "USA"
  ],
  "MaxResults": 1,
  "ResultBBox": [
    -112.10667999999998,
    46.319090000000074,
    -112.10667999999998,
    46.319090000000074
  ],
  "Text": "Amazon"
}
}

```

Example response (HERE)

```

{
  "Summary": {
    "Text": "Amazon",
    "BiasPosition": [
      -112.1,
      46.32
    ],
    "FilterCountries": [

```

```

    "USA"
  ],
  "MaxResults": 1,
  "ResultBBox": [
    -112.10668,
    46.31909,
    -112.10668,
    46.31909
  ],
  "DataSource": "Here"
},
"Results": [
  {
    "Place": {
      "Label": "Amazon, Jefferson City, MT, United States",
      "Geometry": {
        "Point": [
          -112.10668,
          46.31909
        ]
      },
      "Neighborhood": "Amazon",
      "Municipality": "Jefferson City",
      "SubRegion": "Jefferson",
      "Region": "Montana",
      "Country": "USA",
      "Interpolated": false,
      "TimeZone": {
        "Name": "America/Denver",
        "Offset": -25200
      }
    },
    "PlaceId": "AQAAIAADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqdlJJZAdgcT2oWi1w9pS4wXX0k301vsKlGsPyHjV4EJxsu289i3hV0_BUPgP7SFoWai8BW2v7LvAjQ5NfUPy7a1v9a
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ",
    "Distance":
523.4619749905755
  }
]
}

```

Geokodierung umkehren

Reverse Geocoding ist ein Prozess, der eine Reihe von Koordinaten in aussagekräftigen Text umwandelt, z. B. eine Adresse, eine Region, einen Geschäftsnamen oder einen Interessenpunkt. Sie können Ortsindexressourcen verwenden, um Anforderungen zur Reverse-Geocodierung zu senden und Daten, die aus Reverse-Geocodierung abgerufen wurden, zu integrieren, um Daten auf einer Karte für Ihre Web- oder mobile Anwendung anzuzeigen.

Dieser Abschnitt führt Sie durch das Senden einer einfachen Reverse-Geocodierungsanforderung.

Geokodierung umkehren

Sie können eine einfache Anfrage einreichen, um eine Reihe von Koordinaten umzukehren und sie mithilfe der [SearchPlaceIndexForPosition](#) Operation in eine aussagekräftige Adresse, einen Interessenpunkt oder einen allgemeinen Ort ohne Adresse zu konvertieren. Eine einfache -Anforderung enthält den folgenden erforderlichen Parameter:

- **Position** – Eine Reihe von Koordinaten, die Sie in eine Adresse, einen Interessenpunkt oder eine allgemeine Position konvertieren möchten. Definiert im Format `[longitude,latitude]`.

Um eine maximale Anzahl von Ergebnissen pro Seite anzugeben, fügen Sie den folgenden optionalen Parameter hinzu:

- **MaxResults** – Beschränkt die maximale Anzahl der Ergebnisse, die in der Abfrageantwort zurückgegeben werden.

Wenn Sie eine bevorzugte Sprache für die Ergebnisse Ihrer Abfrage angeben möchten, verwenden Sie den folgenden optionalen Parameter:

- **Language** – Ein Sprachcode, der zum Rendern von Ergebnissen verwendet werden soll. Der Wert muss ein gültiger [BCP-47](#)-Sprachcode sein. Zum Beispiel en für Englisch.

Note

Wenn nicht angegeben Language ist oder die angegebene Sprache für ein Ergebnis nicht unterstützt wird, wird die Standardsprache des Partners für dieses Ergebnis verwendet.

Sie können die AWS CLI oder die Amazon Location APIs verwenden.

API

Das folgende Beispiel ist eine [SearchPlaceIndexForPosition](#) Anforderung, die Ortsindex-Ressource *ExamplePlaceIndex* nach einer aussagekräftigen Adresse, einem Interessenpunkt oder einem allgemeinen Standort in der Nähe der Position *[122.3394,47,6159]* zu durchsuchen.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/position
Content-type: application/json

{
  "Position": [-122.3394,47.6159],
  "MaxResults": 5,
  "Language": "de"
}
```

AWS CLI

Das folgende Beispiel ist ein [search-place-index-for-position](#) Befehl, um die Ortsindex-Ressource *ExamplePlaceIndex* nach einer aussagekräftigen Adresse, einem Interessenpunkt oder einer allgemeinen Position in der Nähe der Position *[122.3394,47,6159]* zu durchsuchen.

```
aws location \
  search-place-index-for-position \
    --index-name ExamplePlaceIndex \
    --position -122.3394 47.6159 \
    --max-results 5 \
    --language de
```

Beispielantwort

Example

Im Folgenden finden Sie eine Beispielantwort beim Aufrufen der [-SearchPlaceIndexForPosition](#) Operation über die Amazon Location Places APIs . Die Ergebnisse geben relevante [Stellen](#) und die [Anforderungszusammenfassung](#) zurück. Basierend auf der Auswahl von Esri oder Here als Partner werden zwei Antworten angezeigt.

Example request

```
POST /places/v0/indexes/ExamplePlaceIndex/search/position
Content-type: application/json

{
  "Position": [-122.3394,47.6159],
  "MaxResults": 1
}
```

Example response (Esri)

```
{
  "Results": [
    {
      "Place": {
        "AddressNumber": "2111",
        "Country": "USA",
        "Geometry": {
          "Point": [
            -122.33937999999995,
            47.615910000000004
          ]
        },
        "Interpolated": false,
        "Label": "The Spheres, 2111 7th Ave, Seattle, WA, 98121, USA",
        "Municipality": "Seattle",
        "Neighborhood": "Belltown",
        "PostalCode": "98121",
        "Region": "Washington",
        "SubRegion": "King County"
      },
      "Distance": 1.8685861313438727
    }
  ],
  "Summary": {
    "DataSource": "Esri",
    "MaxResults": 1,
    "Position": [
      -122.3394,
      47.6159
    ]
  }
}
```

```
}
```

Example response (HERE)

```
{
  "Summary": {
    "Position": [
      -122.3394,
      47.6159
    ],
    "MaxResults": 1,
    "DataSource": "Here"
  },
  "Results": [
    {
      "Place": {
        "Label": "2111 7th Ave, Seattle, WA 98121-5114, United States",
        "Geometry": {
          "Point": [
            -122.33938,
            47.61591
          ]
        },
        "AddressNumber": "2111",
        "Street": "7th Ave",
        "Neighborhood": "Belltown",
        "Municipality": "Seattle",
        "SubRegion": "King",
        "Region": "Washington",
        "Country": "USA",
        "PostalCode": "98121-5114",
        "Interpolated": false,
        "TimeZone": {
          "Name": "America/Los_Angeles",
          "Offset": -28800
        }
      },
      "PlaceId": "AQAIIAAdsn2T3KdRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqd1JZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hVO_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9a
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ",
      "Distance": 1.868586125090601
    }
  ]
}
```

}

AutoVervollständigen

Die automatische Vervollständigung bietet Endbenutzern reaktives Feedback, während sie ihre Suchabfrage eingeben. Es enthält Vorschläge für Adressen und Interessenpunkte, die auf teilweise oder falsch geschriebenem Freiformtext basieren. Sie können Ortsindexressourcen verwenden, um Vorschläge zur automatischen Vervollständigung anzufordern und die resultierenden Vorschläge in Ihrer Anwendung anzuzeigen.

Amazon Location unterstützt keine Speicherung von Vorschlägen zur automatischen Vervollständigung. Ein Fehler wird zurückgegeben, wenn der für einen Autocomplete-Aufruf verwendete Ortsindex für die Verwendung mit gespeicherten Geocodes konfiguriert ist. Um gespeicherte Geocodes zu verwenden und Vorschläge abzufragen, erstellen und konfigurieren Sie mehrere Ortsindizes.

In diesem Abschnitt wird beschrieben, wie Sie eine automatische Vervollständigungsanforderung senden. Es beginnt mit der einfachsten Form der Anforderung und zeigt dann optionale Parameter an, mit denen Sie die Relevanz von Suchergebnissen zur automatischen Vervollständigung erhöhen können.

Verwenden der automatischen Vervollständigung

Sie können eine einfache Anfrage für Vorschläge zur automatischen Vervollständigung über die [SearchPlaceIndexForSuggestions](#) Operation einreichen. Die einfachste Form der Anforderung hat einen einzelnen erforderlichen Parameter, die Abfrage Text:

- **Text** – Der Freiform-Teiltext, der zum Generieren von Ortsvorschlägen verwendet werden soll. Zum Beispiel die Zeichenfolge `eiffel tow`.

Um die Anzahl der zurückgegebenen Ergebnisse zu begrenzen, fügen Sie den optionalen `MaxResults` Parameter hinzu:

- **MaxResults** – Beschränkt die Anzahl der Ergebnisse, die in der Abfrageantwort zurückgegeben werden.

Sie können die Amazon Location APIs oder die verwenden AWS CLI.

API

Das folgende Beispiel ist eine [SearchPlaceIndexForSuggestions](#) Anforderung, die Ortsindex-Ressource nach bis zu **5** Vorschlägen zu durchsuchen *ExamplePlaceIndex*, die auf dem partiellen Ortsnamen .

```
POST /places/v0/indices/ExamplePlaceIndex/search/suggestions
Content-type: application/json

{
  "Text": "kamp",
  "MaxResults": 5
}
```

AWS CLI

Das folgende Beispiel ist ein [search-place-index-for-suggestions](#) Befehl, um die Ortsindex-Ressource nach bis zu **5** Vorschlägen zu durchsuchen *ExamplePlaceIndex*, die auf dem partiellen Ortsnamen .

```
aws location \
  search-place-index-for-suggestions \
  --index-name ExamplePlaceIndex \
  --text kamp \
  --max-results 5
```

Der Aufruf von `SearchPlaceIndexForSuggestions` führt zu einer Liste von Orten mit jeweils einem Namen und einer ID. Sie können diese Ergebnisse verwenden, um Vorschläge darüber zu präsentieren, wonach der Benutzer während der Eingabe suchen könnte, z. B. eine Dropdownliste mit Auswahlmöglichkeiten unter einem Textfeld. Im Folgenden finden Sie beispielsweise die Ergebnisse für Vorschläge, die auf einem Benutzer basieren, der *eingibt*.

```
{
  "Summary": {
    "Text": "kamp",
    "MaxResults": 5,
    "DataSource": "Esri"
  },
  "Results": [
    {
```

```

        "Text": "Kampuchea",
        "PlaceId": "AQAAAIAADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqd1JZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hV0_BUPgP7SFoWai8BW2v7LvAjQ5NfUPy7a1v9ajT3
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ"
    },
    {
        "Text": "Kampoul, Kabul, AFG",
        "PlaceId":
"AQAAAIAAA1mx1_-9ffzXD07rBgo9fh6E01Pd1YKvuT5rz2qBDxqBkhTlgkei0PR2s5sa3YBLxUqQI8bhymYcu9R-
DkX3L9QSi3CB5LhNPu160iSFJo6H8S1CrX03QsJALhrr9mdbg0R4R4YDywkHkeBlbn7g5C5LI_wYx873WeQZuilwtsGm8j
UeXcb_bg"
    },
    {
        "Text": "Kampala, UGA",
        "PlaceId":
"AQAAAIAAzZfZt3qMrUkG0byhP6MM0pqy2L8SULLVWT7a3ertLBRS6Q5n7I4s9D7E0nRHADaj7mL7kvX1Q8HD-
mpuiATXNJ1Ix4_V_1B15zHe8j1YKMWvXbgb08cMpgR2fqYqZMR1x-
dfB0080oqujKZldvPIDK1kNe3GwcaqvMWWPMeaGd203brFynubAe-MmFF-Gjz-WBMfUy9og6MV7bkk6NGCA"
    },
    {
        "Text": "Kampar, Riau, IDN",
        "PlaceId": "AQAAAIAAvbXXx-
sr0i111tH0kPdao0GF7WQ_KaZ444SEnevycp6Gtf_2JWgPfCE5bIQCYwya1uZQpX2a8YJoFm2K7Co14fLu7IK0yYOLhZx4k
    },
    {
        "Text": "Kampung Pasir Gudang Baru, Johor, MYS",
        "PlaceId":
"AQAAAIAA4HLQHdjUDcaaXLE9wtNIT1cjQYLgkbnMoG2eNN0AaQ8PJowabLRXmmPUaAj8MAD6vT0i6zqaun5Mixyj7vnYX
    }
}
}

```

Im nächsten Abschnitt wird erläutert, wie Sie die PlaceID aus diesen Ergebnissen verwenden.

Verwenden der Ergebnisse der automatischen Vervollständigung

Der Aufruf von `SearchPlaceIndexForSuggestions` führt zu einer Liste von Orten mit jeweils einem Namen und einer ID. Sie können diese Ergebnisse verwenden, um Vorschläge darüber zu präsentieren, wonach der Benutzer während der Eingabe suchen könnte, z. B. eine Dropdownliste mit Auswahlmöglichkeiten unter einem Textfeld. Wenn der Benutzer eines der Ergebnisse auswählt, können Sie die [GetPlace](#) Operation mit der ID seiner Auswahl aufrufen, um die Details dieses Orts zurückzugeben, einschließlich Standort, Adresse oder anderer Details.

Note

Eine PlaceId ist nur gültig, wenn alle der folgenden Bedingungen in der ursprünglichen Suchanfrage und im Aufruf von `identisch sindGetPlace`.

- Kunde AWS-Konto
- AWS-Region
- Der in der Ortsindex-Ressource angegebene Datenanbieter

In der Regel verwenden Sie `GetPlace` mit den Amazon Location APIs . Das folgende Beispiel ist eine [GetPlace](#) Anforderung, einen der Vorschläge aus dem vorherigen Abschnitt zu finden. Dieses Beispiel basiert auf dem partiellen Ortsnamen .

```
POST /places/v0/indexes/ExamplePlaceIndex/
places/AQAAAIAADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqdlJZAdgcT2oWi1w9pS4wXX0k301vsKlGsPyHjV4EJxsu289i3hV0_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9ajT3-
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ
```

Automatische Vervollständigung nahe einer Position

Wenn Sie mithilfe von nach Vorschlägen zur automatischen Vervollständigung von Orten suchen [SearchPlaceIndexForSuggestions](#), können Sie lokal relevantere Vorschläge erhalten, indem Sie den folgenden optionalen Parameter hinzufügen:

- `BiasPosition` – Die Position, an der Sie in der Nähe suchen möchten. Definiert als `[longitude, latitude]`.

Im folgenden Beispiel wird eine [SearchPlaceIndexForSuggestions](#) Anforderung verwendet, um die Ortsindex-Ressource *ExamplePlaceIndex* nach Ortsvorschlägen zu durchsuchen, die mit dem partiellen *Abfrageblock* nahe der Position `[32.5827,0.3169]` übereinstimmen.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
Content-type: application/json

{
  "Text": "kamp",
  "BiasPosition": [32.5827,0.3169]
```

```
}
```

Die für dasselbe zurückgegebenen Vorschläge Text können unterschiedlich sein, wenn ein anderes ausgewählt `BiasPosition` wird, z. B. `[-96.7977, 32.776]`.

Automatische Vervollständigung innerhalb eines Begrenzungsrahmens

Sie können Ihre Suche nach automatischer Vervollständigung einschränken, um nur Vorschläge für Stellen zu erhalten, die sich innerhalb einer bestimmten Grenze befinden, indem Sie den folgenden optionalen Parameter hinzufügen:

- `FilterBBox` – Ein Begrenzungsrahmen, den Sie angeben, um Ihre Ergebnisse nach Koordinaten innerhalb der Grenzen des Rahmens zu filtern. Definiert als `[LongitudeSW, LatitudeSW, LongitudeNE, LatitudeNE]`

Note

Eine Anforderung darf nicht sowohl die `FilterBBox` `BiasPosition` Parameter als auch enthalten. Die Angabe beider Parameter in der Anforderung gibt einen `ValidationException` Fehler zurück.

Im folgenden Beispiel wird eine [-SearchPlaceIndexForSuggestions](#)Anforderung verwendet, um die Ortsindex-Ressource `ExamplePlaceIndex` nach Ortsvorschlägen zu durchsuchen, die mit der partiellen Abfrage „“ übereinstimmen und die im Begrenzungsrahmen enthalten sind, wobei:

- Der Längengrad der Westecke des Begrenzungsrahmens beträgt `32,5020`.
- Der Breitengrad der Bolwesten Ecke des Begrenzungsrahmens beträgt `0,2678`.
- Der Längengrad der nordöstlichen Ecke des Begrenzungsrahmens beträgt `32,6129`.
- Der Breitengrad der nordöstlichen Ecke des Begrenzungsrahmens beträgt `0,3502`.

```
POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
```

```
Content-type: application/json
```

```
{
  "Text": "kamp",
  "FilterBBox": [
    32.5020, 0.2678,
```

```

    32.6129, 0.3502
  ]
}

```

Die für dasselbe zurückgegebenen Vorschläge Text unterscheiden sich, wenn ein anderes ausgewählt FilterBBox wird, z. B. [-97.9651, 32.0640, -95.1196, 34.0436].

Automatische Vervollständigung innerhalb eines Landes

Sie können Ihre Suche nach automatischer Vervollständigung einschränken, um nur Vorschläge für Orte zu erhalten, die sich in einem bestimmten Land oder einer bestimmten Gruppe von Ländern befinden, indem Sie den folgenden optionalen Parameter hinzufügen:

- **FilterCountries** – Die Länder, in denen Sie nach Ortsvorschlägen suchen möchten. Sie können bis zu 100 Länder in einer Anfrage mit einem dreistelligen [ISO-3166](#)-Ländercode angeben. Verwenden Sie beispielsweise AUS für Australien.

Im folgenden Beispiel wird eine [SearchPlaceIndexForSuggestions](#)Anforderung verwendet, um die Ortsindex-Ressource *ExamplePlaceIndex* nach Ortsvorschlägen zu durchsuchen, die mit der partiellen Abfrage „Camp“ übereinstimmen und in Ug Bol, Kenya oderania enthalten sind:

```

POST /places/v0/indexes/ExamplePlaceIndex/search/suggestions
Content-type: application/json

{
  "Text": "kamp",
  "FilterCountries": ["UGA", "KEN", "TZA"]
}

```

Die für dasselbe zurückgegebenen Vorschläge Text unterscheiden sich, wenn eine andere FilterCountries Liste ausgewählt wird, z. B. ["USA "].

Beispielantwort

Im Folgenden finden Sie eine Beispielantwort vorgeschlagener automatischer Vervollständigungen für die [SearchPlaceIndexForSuggestions](#)Operation unter Verwendung des Text-*Zumpfes* .

```

{
  "Summary": {
    "Text": "kamp",

```

```

    "MaxResults": 5,
    "DataSource": "Esri"
  },
  "Results": [
    {
      "Text": "Kampuchea",
      "PlaceId": "AQAAAIAADsn2T3KdrRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-
o3nqdlJZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hV0_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9ajT3
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ"
    },
    {
      "Text": "Kampoul, Kabul, AFG",
      "PlaceId":
"AQAAAIAAAA1mx1_-9ffzXD07rBgo9fh6E01Pd1YKvuT5rz2qBDxqBkhTlqkei0PR2s5sa3YBLxUqQI8bhymSycu9R-
DkX3L9QSi3CB5LhNPu160iSFJo6H8S1Crx03QsJALhrr9mdbg0R4R4YDywkHkeBlbnbn7g5C5LI_wYx873WeQZuilwtsGm8j
UeXcb_bg"
    },
    {
      "Text": "Kampala, UGA",
      "PlaceId":
"AQAAAIAAzZfZt3qMrUkG0byhP6MM0pqy2L8SUL1VWT7a3ertLBRS6Q5n7I4s9D7E0nRHADaj7mL7kvX1Q8HD-
mpuiATXNJ1Ix4_V_1B15zHe8j1YKMWvXbgb08cMpgR2fqYqZMR1x-
dfB0080oqujKZ1dvPIDK1kNe3GwcaqvMWWPMeaGd203brFynubAe-MmFF-Gjz-WBMfUy9og6MV7bkk6NGCA"
    },
    {
      "Text": "Kampar, Riau, IDN",
      "PlaceId": "AQAAAIAAvbXXx-
sr0i111tH0kPdao0GF7WQ_KaZ444SEnevycp6Gtf_2JWgPfCE5bIQCYwya1uZQpX2a8YJoFm2K7Col4fLu7IK0yY0LhZx4k
",
    },
    {
      "Text": "Kampung Pasir Gudang Baru, Johor, MYS",
      "PlaceId":
"AQAAAIAA4HLQHdjUDcaaXLE9wtNIT1cjQYLgkBnMoG2eNN0AaQ8PJoWabLRXmmPUaAj8MAD6vT0i6zqaun5Mixyj7vnYX
"
    }
  ]
}

```

Verwenden von Orts-IDs

Die Suche nach Orten gibt eine Liste von Ergebnissen zurück. Die meisten Ergebnisse enthalten einen PlaceId für dieses Ergebnis. Sie können einen PlaceId in einer [GetPlace](#) Operation verwenden, um die Informationen über diesen Ort (einschließlich Name, Adresse, Standort oder anderer Details) zurückzugeben.

Note

Die Verwendung von [SearchPlaceIndexForSuggestions](#) gibt PlaceId Ergebnisse für alle Ortsindizes zurück, die mit einer beliebigen Datenquelle erstellt wurden. Die Verwendung von [SearchPlaceIndexForText](#) oder [SearchPlaceIndexForPosition](#) gibt PlaceId nur dann einen zurück, wenn die verwendete Datenquelle HERE ist.

Jede definiert PlaceId eindeutig den Ort, auf den sie sich bezieht, aber ein einzelner Ort kann im PlaceId Laufe der Zeit mehr als einen haben und basiert auf dem Kontext. Die folgenden Regeln beschreiben die Eindeutigkeit und Lebensdauer eines PlaceId.

- Die in Aufrufen PlaceId zurückgegebenen sind spezifisch für Ihr AWS-Konto, die AWS Region und den Datenanbieter in Ihrer PlaceIndex Ressource. GetPlace findet Ergebnisse nur, wenn diese drei Attribute mit dem ursprünglichen Aufruf übereinstimmen, der das erstellt hatPlaceId.
- Die PlaceId für einen Ort ändert sich, wenn sich die Daten zu diesem Ort ändern. Wenn beispielsweise das Unternehmen, auf das es sich bezieht, den Standort verschiebt oder Namen ändert.
- Der von einem wiederholten Suchaufruf PlaceId zurückgegebene kann sich ändern, wenn der Backend-Service eine Aktualisierung vornimmt. Das ältere PlaceId wird weiterhin gefunden, aber neue Suchaufrufe geben möglicherweise eine andere ID zurück.

ist PlaceId eine Zeichenfolge. Es gibt keine spezifische Begrenzung für die Länge eines PlaceId. Im Folgenden finden Sie ein Beispiel für einen gültigen PlaceId.

```
AQAAAIAADsn2T3KdRWeaXLeVEyjNx_JfeTsMB0NVCEAnAZoJ-  
o3nqdLJZAdgcT2oWi1w9pS4wXX0k301vsK1GsPyHjV4EJxsu289i3hV0_BUPgP7SFoWAI8BW2v7LvAjQ5NfUPy7a1v9ajT3  
et39ZQDWSPLZUzgcjN-6VD2gyKkH0Po7gSm8YSJNSQ
```

Wenn Sie GetPlace mit einem PlaceId für einen Ort aufrufen, dessen Daten sich geändert haben (z. B. einen Geschäftsstandort, der nicht mehr funktioniert), führt dies zu einem 404-, -ResourceNotFoundFehler. Wenn Sie GetPlace mit einer aufrufen, PlaceId die nicht gültig ist, oder einer, die nicht im Kontext ist, z. B. von einer anderen AWS-Konto, wird ein 400-ValidationExceptionFehler zurückgegeben.

Sie können PlacelD zwar in nachfolgenden Anforderungen verwenden, PlacelD ist jedoch nicht als permanente Kennung gedacht und die ID kann sich zwischen aufeinanderfolgenden API-Aufrufen ändern. Bitte beachten Sie das folgende PlacelD-Verhalten für jeden Datenanbieter:

- Esri : Orts-IDs ändern sich mindestens jedes Quartal. Der typische Zeitraum für diese Änderungen wäre März, Juni, September und Dezember. Orts-IDs können sich auch zwischen der typischen vierteljährlichen Änderung ändern, aber das wird viel seltener sein.
- HERE : Wir empfehlen, dass Sie Daten nicht länger als eine Woche zwischenspeichern, um Ihre Daten auf dem neuesten Stand zu halten. Sie können davon ausgehen, dass weniger als 1 % ID-Schichten gegenüber der Veröffentlichung freigegeben werden, was etwa 1–2 Mal pro Woche entspricht.
- Grab : Orts-IDs können in den folgenden Situationen ablaufen oder ungültig werden.
 - Datenoperationen: Das POI kann von Grab Map Ops auf der Grundlage der Grundwahrheit aus der Grab-POI-Datenbank entfernt werden, z. B. in der realen Welt geschlossen, als doppeltes POI erkannt oder falsche Informationen vorliegen. Grab synchronisiert Daten wöchentlich mit der Waypoint-Umgebung.
 - Interpoliertes POI: Interpoliertes POI ist ein temporäres POI, das bei der Verarbeitung einer Anforderung in Echtzeit generiert wird, und es wird im `place.result_type` Feld in der Antwort als abgeleitet markiert. Die Informationen zu interpolierten POIs werden mindestens 30 Tage lang aufbewahrt, was bedeutet, dass Sie innerhalb von 30 Tagen POI-Details nach Orts-ID von der API für Ortsdetails abrufen können. Nach 30 Tagen können die interpolierten POIs (sowohl Orts-ID als auch Details) ablaufen und über die API für Ortsdetails nicht mehr zugänglich sein.

Kategorien und Filterergebnisse platzieren

Standorte werden kategorisiert. Wenn Sie nach einem Unternehmen suchen, könnte das Unternehmen Restaurantbeispielsweise ein sein. Selbst die Ergebnisse einer Suche nach einer Adresse können danach kategorisiert werden, ob sie mit einer Adresse , Straßen oder Schnittmenge übereinstimmt.

Im Allgemeinen kategorisiert Amazon Location Service Platzierungen in Ortstypen . Die Interessenpunkte werden weiter in die Interessenpunkte unterteilt.

Note

Nicht alle Ergebnisse haben Kategorien.

Sie können die Kategorien verwenden, um Ihre Geocodierungssuchen zu filtern.

Filtern von Ergebnissen

Wenn Sie verwenden `SearchPlaceIndexForText`, können Sie die Ergebnisse filtern, die von den Kategorien zurückgegeben werden, die Sie verwenden möchten. Beispielsweise:

- Wenn Sie nach einem Ort mit dem Namen „HomeSpeed“ suchen und nur Ergebnisse zurückgeben möchten, die als Fußbälle kategorisiert sind, können Sie dies tun, indem Sie aufrufen `SearchPlaceIndexForText` und die Kategorie `Interessenpunkt Coffee Shop` in den `-FilterCategoriesParameter` aufnehmen.
- Wenn Sie nach „123 Main St, Any, WA, 98123, USA“ suchen, können Sie das Ergebnis nur nach Adressen filtern, sodass Sie keine Übereinstimmungen erhalten, z. B. nach der Postleitzahl. Filtern Sie nur nach Adressen, indem Sie den Ortstyp `AddressType` in den `FilterCategories` Parameter einschließen.

Note

Nicht alle Datenanbieter unterstützen Filterung oder sie auf die gleiche Weise. Weitere Informationen finden Sie unter [Filtern von Einschränkungen nach Datenanbieter](#).

Im nächsten Abschnitt werden die Kategorien aufgeführt, nach denen Sie filtern können.

Kategorien

Die folgenden Listen zeigen die Kategorien, die Amazon Location Service zur Kategorisierung und Filterung verwendet. Diese Kategorien werden in allen Sprachen verwendet, unabhängig davon, ob der Sprachparameter auf eine andere Sprache festgelegt ist.

Note

Amazon Location Service ordnet Datenanbieterkategorien dieser Gruppe von Kategorien zu. Wenn ein Datenanbieter eine Stelle in eine Kategorie einfügt, die nicht Teil der Kategorieliste von Amazon Location Service ist, wird die Anbieterkategorie als zusätzliche Kategorie in die Ergebnisse aufgenommen.

Ortstypen – Diese Typen werden verwendet, um die Art der Übereinstimmung anzugeben, die zur Suche des Ergebnisses verwendet wurde.

- `AddressType` – Wird zurückgegeben, wenn das Ergebnis einer Adresse zugeordnet wurde.
- `StreetType` – Wird zurückgegeben, wenn das Ergebnis einer Straßen zugeordnet wurde.
- `IntersectionType` – Wird zurückgegeben, wenn das Ergebnis mit der Schnittmenge von zwei Straßen übereinstimmt.
- `PointOfInterestType` – Wird zurückgegeben, wenn das Ergebnis mit einem Interessenpunkt übereinstimmte, z. B. einem Unternehmen oder einem bivischen Standort.
- `CountryType` – Wird zurückgegeben, wenn das Ergebnis mit einem Land oder einer Hauptregion übereinstimmt.
- `RegionType` – Wird zurückgegeben, wenn das Ergebnis mit einer Region innerhalb eines Landes abgeglichen wurde, z. B. einem Bundesstaat oder einer Provinz.
- `SubRegionType` – Wird zurückgegeben, wenn das Ergebnis mit einer Unterregion innerhalb eines Landes abgeglichen wurde, z. B. einem Landkreis oder einem Stadtgebiet.
- `MunicipalityType` – Wird zurückgegeben, wenn das Ergebnis mit einer Stadt oder einem Arzt übereinstimmt.
- `NeighborhoodType` – Wird zurückgegeben, wenn das Ergebnis mit einer Nachbarschaft oder einem Gebiet innerhalb einer Stadt übereinstimmt.
- `PostalCodeType` – Wird zurückgegeben, wenn das Ergebnis mit einer Postleitzahl abgeglichen wurde.

Kategorien von Interessenpunkten – Diese Kategorien werden verwendet, um die Art des Unternehmens oder Standorts für die Ergebnisse von Interessenpunkten anzugeben.

- `Airport`
- `Amusement Park`
- `Aquarium`
- `Art Gallery`
- `ATM`
- `Bakery`
- `Bank`
- `Bar`

- Beauty Salon
- Bus Station
- Car Dealer
- Car Rental
- Car Repair
- Car Wash
- Cemetery
- Cinema
- City Hall
- Clothing Store
- Coffee Shop
- Consumer Electronics Store
- Convenience Store
- Court House
- Dentist
- Embassy
- Fire Station
- Fitness Center
- Gas Station
- Government Office
- Grocery
- Higher Education
- Hospital
- Hotel
- Laundry
- Library
- Liquor Store
- Lodging

- Market
- Medical Clinic
- Motel
- Museum
- Nightlife
- Nursing Home
- Park
- Parking
- Pet Store
- Pharmacy
- Plumbing
- Police Station
- Post Office
- Religious Place
- Restaurant
- School
- Shopping Mall
- Sports Center
- Storage
- Taxi Stand
- Tourist Attraction
- Train Station
- Veterinary Care
- Zoo

Filtern von Einschränkungen nach Datenanbieter

Nicht alle Anbieter verfügen über dieselbe Filterfunktionalität. In der folgenden Tabelle werden die Unterschiede beschrieben.

Anbieter	APIs mit Filterunterstützung	Für die Filterung unterstützte Kategorien	Rückgabewerte
Esri	SearchPlaceIndexForText , SearchPlaceIndexForSuggestions	Filtern Sie nach Ortstypen und Interessenpunktkategorien.	Kategorien werden von SearchPlaceIndexForText , SearchPlaceIndexForPosition und zurückgegeben GetPlace
Hier	SearchPlaceIndexForText , SearchPlaceIndexForSuggestions	Nur nach Ortstypen filtern.	Kategorien werden von SearchPlaceIndexForText und SearchPlaceIndexForSuggestions , SearchPlaceIndexForPosition und zurückgegeben GetPlace
Grab	Nicht unterstützt	Nicht unterstützt	Nicht unterstützt
Daten öffnen	n/a (Suche nach Orten wird nicht unterstützt)	–	–

Amazon Aurora PostgreSQL Benutzerdefinierte Funktionen für Amazon Location Service

Sie können Amazon Location Service verwenden, um mit Koordinaten und Adressen zu arbeiten, die in Datenbanktabellen gespeichert sind, um Ihre Geodaten zu bereinigen und zu ergänzen.

Beispielsweise:

- Sie können Geokodierung verwenden, um Adressen in Koordinaten zu konvertieren, um Datenlücken für Adressen zu normalisieren und zu füllen, die in einer Datenbanktabelle gespeichert sind.
- Sie können Adressen geocodieren, um ihre Positionen zu erhalten, und die Koordinaten mit räumlichen Datenbankfunktionen verwenden, z. B. mit einer Funktion, die Zeilen in einem bestimmten Bereich anzeigt.
- Sie können mit erweiterten Daten automatisierte Berichte erstellen, z. B. einen automatisierten Bericht, der alle Geräte in einem bestimmten Bereich veranschaulicht, oder einen automatisierten Bericht für Machine Learning, der Bereiche mit höheren Fehlerraten beim Senden von Standortaktualisierungen veranschaulicht.

Dieses Tutorial zeigt, wie Adressen formatiert und anreicht werden, die in einer -Amazon Aurora PostgreSQL-Datenbanktabelle mit Amazon Location Service gespeichert sind.

- Amazon Aurora PostgreSQL – Eine vollständig verwaltete, mit MySQL und PostgreSQL kompatible relationale Datenbank-Engine, die einen bis zu fünffachen Durchsatz von MySQL und einen bis zu dreimal höheren Durchsatz von PostgreSQL ausgibt, ohne den Großteil Ihrer vorhandenen Anwendung zu ändern. Weitere Informationen finden Sie unter [Was ist Amazon Aurora?](#) im Amazon-Aurora-Benutzerhandbuch.

Important

Die resultierende Anwendung in diesem Tutorial verwendet einen Ortsindex, der Geocodierungsergebnisse speichert. Weitere Informationen zu den Gebühren für die Speicherung von Geocodierungsergebnissen finden Sie unter [Amazon Location Service – Preise](#).

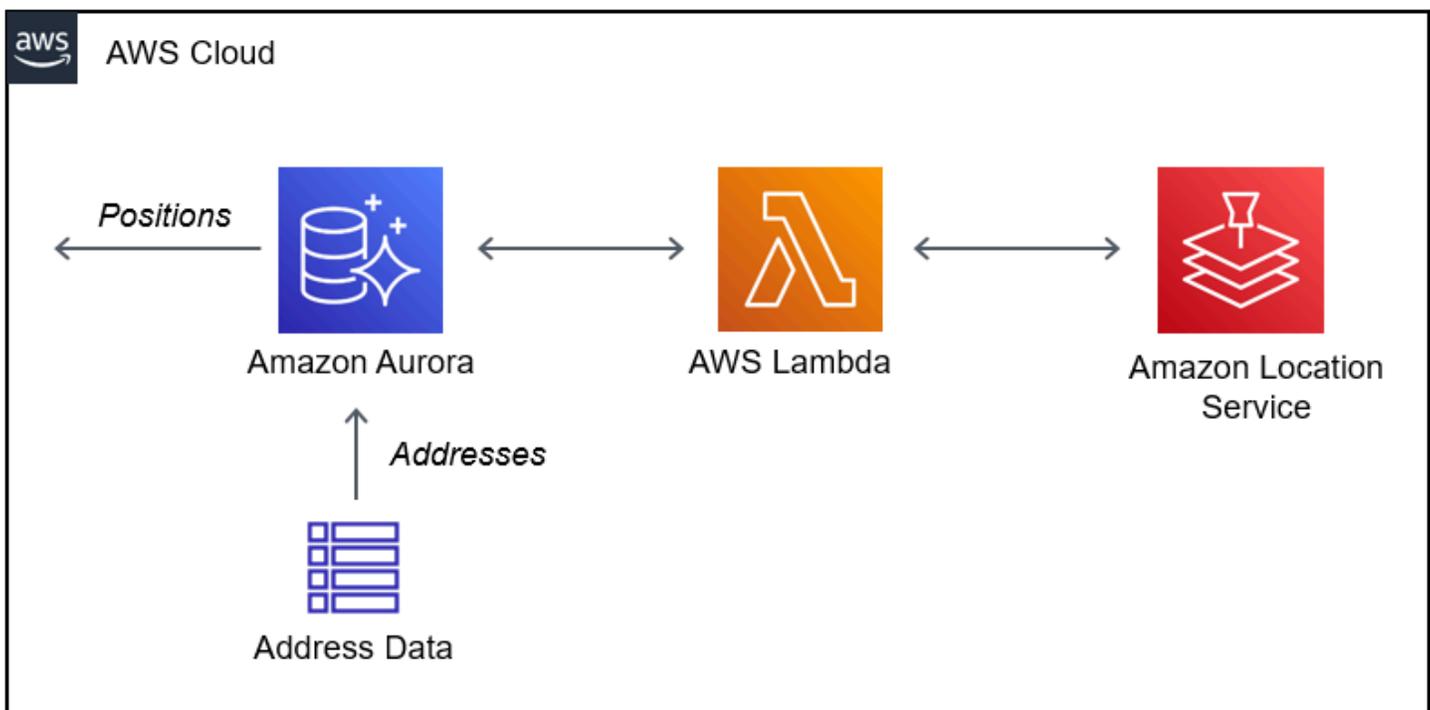
Beispielcode ist im Amazon Location Service-Beispiel-Repository auf verfügbar [GitHub](#), das [eine - AWS CloudFormation Vorlage](#) enthält.

Themen

- [Übersicht](#)
- [Voraussetzungen](#)

- [Schnellstart](#)
- [Erstellen einer Ortsindex-Ressource](#)
- [Erstellen einer -AWS LambdaFunktion für die Geocodierung](#)
- [Gewähren von Amazon Aurora PostgreSQL Zugriff auf AWS Lambda](#)
- [Aufrufen der AWS Lambda-Funktion](#)
- [Anreicherung einer Datenbank mit Adressdaten](#)
- [Nächste Schritte](#)

Übersicht



Die Architektur umfasst die folgenden Integrationen:

- Diese Lösung verwendet eine Amazon-Location-Place-Indexressource, um Geocodierungsabfragen mit der Operation `SearchPlaceIndexForText` zu unterstützen.
- AWS Lambda verwendet ein Python-Lambda, das Adressen geocodiert, wenn eine IAM-Richtlinie die Berechtigung erteilt, AWS Lambda die Geocodierungsoperation von Amazon Location aufzurufen, `SearchPlaceIndexForText`.
- Erteilen Sie die Berechtigung Amazon Aurora PostgreSQL zum Aufrufen der Geocodierungs-Lambda-Funktion mithilfe einer benutzerdefinierten SQL-Funktion.

Voraussetzungen

Bevor Sie beginnen, benötigen Sie die folgenden Voraussetzungen:

- Ein -Amazon Aurora PostgreSQLCluster. Weitere Informationen zum [Erstellen eines Amazon-Aurora-DB-Clusters](#) finden Sie im Amazon-Aurora-Benutzerhandbuch.

Note

Wenn Ihr Amazon-Aurora-Cluster nicht öffentlich verfügbar ist, müssen Sie Amazon Aurora auch so konfigurieren, dass eine Verbindung zu AWS Lambda in einer Virtual Private Cloud (VPC) in Ihrem AWS Konto hergestellt wird. Weitere Informationen finden Sie unter [Gewähren von Amazon Aurora PostgreSQL Zugriff auf AWS Lambda](#).

- Ein SQL-Entwicklertool zum Herstellen einer Verbindung mit dem Amazon Aurora PostgreSQL Cluster.

Schnellstart

Als Alternative zu den Schritten in diesem Tutorial können Sie einen Quick Stack starten, um eine -AWS LambdaFunktion bereitzustellen, die den Amazon Location-Vorgang unterstützt [SearchPlaceIndexForText](#). Dadurch wird Ihr AWS Konto automatisch so konfiguriert, dass Amazon Aurora aufrufen kannAWS Lambda.

Sobald Sie Ihr AWS Konto konfiguriert haben, müssen Sie:

- Fügen Sie das Lambda-Feature zu Amazon Aurora hinzu. Weitere Informationen finden Sie unter Hinzufügen der IAM-Rolle zu einem Amazon-Aurora-DB-Cluster in [Gewähren von Amazon Aurora PostgreSQL Zugriff auf AWS Lambda](#).
- Laden Sie die benutzerdefinierte Funktion in Ihre Datenbank. Siehe [Aufrufen der AWS Lambda-Funktion](#).

Launch Stack 

Erstellen einer Ortsindex-Ressource

Erstellen Sie zunächst eine Ortsindexressource zur Unterstützung von Geocodierungsabfragen.

1. Öffnen Sie die Amazon Location Service-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Indizes platzieren aus.
3. Füllen Sie die folgenden Felder aus:
 - Name – Geben Sie einen Namen für die Ortsindex-Ressource ein. Zum Beispiel *AuroraPlaceIndex*. Maximal 100 Zeichen. Gültige Einträge umfassen alphanumerische Zeichen, Bindestriche, Punkte und Unterstriche.
 - Beschreibung – Geben Sie eine optionale Beschreibung ein. Zum Beispiel den *Ortsindex für Amazon Aurora* .
4. Wählen Sie unter Datenanbieter einen verfügbaren [Datenanbieter](#) aus, der mit Ihrer Ortsindexressource verwendet werden soll. Wenn Sie keine Präferenz haben, empfehlen wir, mit *Esri* zu beginnen.
5. Geben Sie unter Datenspeicheroptionen die Option Ja an, Ergebnisse werden gespeichert. Dies weist darauf hin, dass Sie beabsichtigen, die Geocodierungsergebnisse in einer Datenbank zu speichern.
6. (Optional) Geben Sie unter Markierungen einen Tag Key (Schlüssel) und einen Value (Wert) ein. Dadurch wird ein Tag für Ihre neue Ortsindexressource hinzugefügt. Weitere Informationen finden Sie unter [Markieren Ihrer Ressourcen](#).
7. Wählen Sie Ortsindex erstellen aus.

Erstellen einer -AWS LambdaFunktion für die Geocodierung

Um eine Verbindung zwischen Amazon Aurora PostgreSQL und Amazon Location Service herzustellen, benötigen Sie eine -AWS LambdaFunktion, um Anfragen von der Datenbank-Engine zu verarbeiten. Diese Funktion übersetzt das benutzerdefinierte Lambda-Funktionsereignis und ruft die Amazon Location-Operation aufSearchPlaceIndexForText.

Sie können die Funktion mithilfe der AWS LambdaKonsole, der AWS Command Line Interfaceoder der AWS Lambda APIs erstellen.

So erstellen Sie eine benutzerdefinierte Lambda-Funktion mithilfe der Konsole

1. Öffnen Sie die AWS Lambda-Konsole unter <https://console.aws.amazon.com/lambda/>.
2. Wählen Sie in der linken Navigation Funktionen aus.

3. Wählen Sie Funktion erstellen und stellen Sie sicher, dass Von Grund auf neu erstellen ausgewählt ist.
4. Füllen Sie die folgenden Felder aus:
 - Funktionsname – Geben Sie einen eindeutigen Namen für Ihre Funktion ein. Gültige Einträge umfassen alphanumerische Zeichen, Bindestriche und Unterstriche ohne Leerzeichen. Zum Beispiel *AuroraGeocoder*.
 - Laufzeit – Wählen Sie *Python 3.8* aus.
5. Wählen Sie Funktion erstellen.
6. Wählen Sie die Registerkarte Code, um den Editor zu öffnen.
7. Überschreiben Sie den Platzhaltercode in `lambda_function.py` wie folgt:

```
from os import environ

import boto3
from botocore.config import Config

# load the place index name from the environment, falling back to a default
PLACE_INDEX_NAME = environ.get("PLACE_INDEX_NAME", "AuroraPlaceIndex")

location = boto3.client("location", config=Config(user_agent="Amazon Aurora
  PostgreSQL"))

"""
This Lambda function receives a payload from Amazon Aurora and translates it to
an Amazon Location `SearchPlaceIndex` call and returns the results as-is, to be
post-processed by a PL/pgSQL function.
"""
def lambda_handler(event, context):
    kwargs = {}

    if event.get("biasPosition") is not None:
        kwargs["BiasPosition"] = event["biasPosition"]

    if event.get("filterBBox") is not None:
        kwargs["FilterBBox"] = event["filterBBox"]

    if event.get("filterCountries") is not None:
        kwargs["FilterCountries"] = event["filterCountries"]

    if event.get("maxResults") is not None:
```

```
kwargs["MaxResults"] = event["maxResults"]

return location.search_place_index_for_text(
    IndexName=PLACE_INDEX_NAME,
    Text=event["text"],
    **kwargs)["Results"]
```

8. Wenn Sie Ihren Ortsindex anders als benannt haben *AuroraPlaceIndex*, erstellen Sie eine Umgebungsvariable mit dem Namen `PLACE_INDEX_NAME`, um den Ressourcennamen zuzuweisen:
 - Wählen Sie auf der Registerkarte Konfiguration die Option Umgebungsvariablen aus.
 - Wählen Sie Bearbeiten und dann Umgebungsvariable hinzufügen aus.
 - Geben Sie für Schlüssel ein `PLACE_INDEX_NAME`.
 - Geben Sie für Wert den Namen Ihrer Ortsindex-Ressource ein.
9. Wählen Sie Bereitstellen, um die aktualisierte Funktion zu speichern.
10. Wählen Sie im Dropdown-Menü Test die Option Testereignis konfigurieren aus.
11. Wählen Sie Create new test event aus.
12. Geben Sie das folgende Testereignis ein:

```
{
  "text": "Baker Beach",
  "biasPosition": [-122.483, 37.790],
  "filterCountries": ["USA"]
}
```

13. Wählen Sie Test, um die Lambda-Funktion zu testen.
14. Wählen Sie die Registerkarte Konfiguration aus.
15. Wählen Sie unter Allgemeine Konfiguration die Option Berechtigungen aus.
16. Wählen Sie unter Ausführungsrolle den Hyperlink-Rollennamen aus, um Amazon Location Service Berechtigungen für Ihre Lambda-Funktion zu erteilen.
17. Wählen Sie auf der Registerkarte Berechtigungen: Wählen Sie das Dropdown-Menü Berechtigungen hinzufügen und dann Inline-Richtlinie erstellen aus.
18. Wählen Sie den Tab JSON.
19. Fügen Sie die folgende IAM-Richtlinie hinzu:

- Die folgende Richtlinie erteilt die Berechtigung zum Senden `SearchPlaceIndexForText` an die Ortsindex-Ressource *AuroraPlaceIndex*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:SearchPlaceIndexForText",
      "Resource": "arn:aws:geo:<Region>:<AccountId>:place-index/AuroraPlaceIndex"
    }
  ]
}
```

20. Wählen Sie Richtlinie prüfen.
21. Geben Sie den Namen einer Richtlinie ein. Zum Beispiel *AuroraPlaceIndexReadOnly*.
22. Wählen Sie Richtlinie erstellen aus.

Gewähren von Amazon Aurora PostgreSQL Zugriff auf AWS Lambda

Bevor eine -AWS LambdaFunktion aufrufen Amazon Aurora PostgreSQL kann, müssen Sie Zugriffsberechtigungen erteilen.

Wenn Ihr Amazon Aurora PostgreSQL Cluster nicht öffentlich zugänglich ist, müssen Sie zunächst einen VPC-Endpunkt für erstellenAWS Lambda, damit Amazon Aurora Ihre Lambda-Funktion aufrufen kann.

Erstellen eines VPC-Endpunkts für AWS Lambda

Note

Dieser Schritt ist nur erforderlich, wenn Ihr Amazon Aurora PostgreSQL Cluster nicht öffentlich zugänglich ist.

1. Öffnen Sie die [Amazon Virtual Private Cloud Console](#).
2. Wählen Sie in der linken Navigation Endpunkte aus.
3. Wählen Sie Endpunkt erstellen aus.

4. Geben Sie im Filter Service Name „lambda“ ein und wählen Sie dann `auscom.amazonaws.<region>.lambda`.
5. Wählen Sie die VPC aus, die Ihren Aurora-Cluster enthält.
6. Wählen Sie für jede Availability Zone ein Subnetz aus.
7. Geben Sie im Filter Sicherheitsgruppe „Standard“ oder den Namen der Sicherheitsgruppe ein, der Ihr Aurora-Cluster angehört, und wählen Sie dann die Sicherheitsgruppe aus.
8. Wählen Sie Endpunkt erstellen aus.

Erstellen Sie eine IAM-Richtlinie, um die Berechtigung zum Aufrufen Ihrer AWS Lambda Funktion zu erteilen

1. Öffnen Sie die [IAM-Konsole](#).
2. Erweitern Sie in der linken Navigation die Option Zugriffsverwaltung, um Richtlinien auszuwählen.
3. Wählen Sie Richtlinie erstellen aus.
4. Geben Sie auf der Registerkarte JSON die folgende Richtlinie ein:
 - Im Folgenden finden Sie ein Beispiel für eine IAM-Richtlinie, die die Amazon Aurora PostgreSQL Berechtigung zum Aufrufen der AuroraGeocoder AWS Lambda Funktion erteilt.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "lambda:InvokeFunction",
      "Resource": [
        "arn:aws:lambda:<Region>:<AccountId>:function:AuroraGeocoder"
      ]
    }
  ]
}
```

5. Wählen Sie Weiter: Tags, um optionale Tags hinzuzufügen.
6. Wählen Sie Weiter: Prüfen aus.
7. Überprüfen Sie Ihre Richtlinie und geben Sie die folgenden Details für die Richtlinie ein:

- Name – Verwenden Sie alphanumerische Zeichen und „+=,@-“. Maximal 128 Zeichen. Zum Beispiel *AuroraGeocoderInvoke*.
 - Beschreibung – Geben Sie eine optionale Beschreibung ein. Verwenden Sie alphanumerische Zeichen und '+=,@-'. Maximal 1000 Zeichen.
8. Wählen Sie Richtlinie erstellen aus. Notieren Sie sich den ARN für diese Richtlinie, mit dem Sie die Richtlinie an eine IAM-Rolle anfügen.

Erstellen einer IAM-Rolle, um Amazon Relational Database Service (Amazon RDS) die Berechtigung zu erteilen

Durch Erstellen einer IAM-Rolle Amazon Aurora PostgreSQL kann die Rolle in Ihrem Namen übernehmen, um auf Ihre Lambda-Funktion zuzugreifen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen IAM-Benutzer](#) im IAM-Benutzerhandbuch.

Das folgende Beispiel ist ein -AWS CLIBefehl, der eine Rolle mit dem Namen erstellt *AuroraGeocoderInvokeRole*:

```
aws iam create-role --role-name rds-lambda-role --assume-role-policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}'
```

Hängen Sie Ihre IAM-Richtlinie an die IAM-Rolle an

Wenn Sie über eine IAM-Rolle verfügen, fügen Sie die von Ihnen erstellte IAM-Richtlinie an.

Das folgende Beispiel ist ein -AWS CLIBefehl, der die Richtlinie *AuroraGeocoderInvoke* an die Rolle anfügt *AuroraGeocoderInvokeRole*.

```
aws iam attach-role-policy --policy-arn AuroraGeocoderInvoke --role-  
name AuroraGeocoderInvokeRole
```

Hinzufügen der IAM-Rolle zu einem Amazon Aurora-DB-Cluster

Das folgende Beispiel ist ein -AWS CLIBefehl zum Hinzufügen einer IAM-Rolle zu einem Amazon Aurora PostgreSQL DB-Cluster mit dem Namen *MyAuroraCluster*.

```
aws rds add-role-to-db-cluster \  
--db-cluster-identifier MyAuroraCluster \  
--feature-name Lambda \  
--role-arn AuroraGeocoderInvokeRole \  
--region your-region
```

Aufrufen der AWS Lambda-Funktion

Nachdem Sie die Berechtigung zum Aufrufen Ihrer Lambda-Funktion Amazon Aurora PostgreSQL für die Geocodierung erteilt haben, können Sie eine Amazon Aurora PostgreSQL benutzerdefinierte Funktion erstellen, um die AWS Lambda Geocodierungsfunktion aufzurufen. Weitere Informationen finden Sie unter [Aufrufen einer -AWS Lambda-Funktion aus einem Amazon Aurora PostgreSQL-DB-Cluster](#) im Amazon-Aurora-Benutzerhandbuch.

Installieren der erforderlichen PostgreSQL-Erweiterungen

Informationen zum Installieren der erforderlichen PostgreSQL-Erweiterungen `aws_lambda` und `aws_commons` Erweiterungen finden Sie unter [Übersicht über die Verwendung einer Lambda-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.

```
CREATE EXTENSION IF NOT EXISTS aws_lambda CASCADE;
```

Installieren der erforderlichen PostGIS-Erweiterungen

PostGIS ist eine Erweiterung von PostgreSQL zur Speicherung und Verwaltung von Geodaten. Weitere Informationen finden Sie unter [Arbeiten mit der PostGIS-Erweiterung](#) im Amazon Relational Database Service-Benutzerhandbuch.

```
CREATE EXTENSION IF NOT EXISTS postgis;
```

Erstellen einer benutzerdefinierten SQL-Funktion, die die Lambda-Funktion aufruft

Erstellen Sie in einem SQL-Editor eine neue benutzerdefinierte Funktion, `f_SearchPlaceIndexForText` um die Funktion aufzurufen *AuroraGeocoder*:

```
CREATE OR REPLACE FUNCTION f_SearchPlaceIndexForText(
  text text,
  bias_position geometry(Point, 4326) DEFAULT NULL,
  filter_bbox box2d DEFAULT NULL,
  filter_countries text[] DEFAULT NULL,
  max_results int DEFAULT 1
)
RETURNS TABLE (
  label text,
  address_number text,
  street text,
  municipality text,
  postal_code text,
  sub_region text,
  region text,
  country text,
  geom geometry(Point, 4326)
)
LANGUAGE plpgsql
IMMUTABLE
AS $function$
begin
  RETURN QUERY
  WITH results AS (
    SELECT json_array_elements(payload) rsp
    FROM aws_lambda.invoke(
      aws_commons.create_lambda_function_arn('AuroraGeocoder'),
      json_build_object(
        'text', text,
        'biasPosition',
        CASE WHEN bias_position IS NOT NULL THEN
          array_to_json(ARRAY[ST_X(bias_position), ST_Y(bias_position)])
        END,
        'filterBBox',
        CASE WHEN filter_bbox IS NOT NULL THEN
          array_to_json(ARRAY[ST_XMin(filter_bbox), ST_YMin(filter_bbox),
ST_XMax(filter_bbox), ST_YMax(filter_bbox)])
        END,
        'filterCountries', filter_countries,
        'maxResults', max_results
```

```
    )
  )
)
SELECT
  rsp->'Place'->>'Label' AS label,
  rsp->'Place'->>'AddressNumber' AS address_number,
  rsp->'Place'->>'Street' AS street,
  rsp->'Place'->>'Municipality' AS municipality,
  rsp->'Place'->>'PostalCode' AS postal_code,
  rsp->'Place'->>'SubRegion' AS sub_region,
  rsp->'Place'->>'Region' AS region,
  rsp->'Place'->>'Country' AS country,
  ST_GeomFromGeoJSON(
    json_build_object(
      'type', 'Point',
      'coordinates', rsp->'Place'->'Geometry'->'Point'
    )
  ) geom
FROM results;
end;
$function$;
```

Rufen Sie die SQL-Funktion auf, um von Aurora aus zu geocodieren

Wenn Sie die SQL-Anweisung ausführen, wird die Lambda-Funktion aufgerufen *AuroraGeocoder*, die Adresdatensätze aus der Datenbanktabelle in der Amazon Aurora PostgreSQL Datenbank übernimmt und sie mithilfe einer Ortsindexressource geocodiert.

Note

Amazon Aurora PostgreSQL ruft die Lambda-Funktion für jeden Aufruf der benutzerdefinierten SQL-Funktion auf.

Wenn Sie 50 Zeilen geocodieren, Amazon Aurora PostgreSQL ruft die Lambda-Funktion 50 Mal auf. Ein Aufruf für jede Zeile.

Die folgende `f_SearchPlaceIndexForText` SQL-Funktion stellt Anforderungen an die [SearchPlaceIndexForText](#) API von Amazon Location über die *AuroraGeocoder* Lambda-Funktion. Die Funktion gibt eine `geom` Spalte zurück, die eine PostGIS-Geometrie ist, die in `Text_ST_AsText(geom)` konvertiert wird.

```
SELECT *, ST_AsText(geom)
FROM f_SearchPlaceIndexForText('Vancouver, BC');
```

Standardmäßig enthält die Rückgabe eine Zeile. Um zusätzliche Zeilen bis zum `MaxResults` Limit anzufordern, führen Sie die folgende SQL-Anweisung aus, während Sie ein `BiasPosition` und eine Einschränkung auf die Ergebnisse in Kanada angeben.

```
SELECT *
FROM f_SearchPlaceIndexForText('Mount Pleasant', ST_MakePoint(-123.113, 49.260), null,
'{"CAN"}', 5);
```

Um Ergebnisse mithilfe eines Begrenzungsrahmens zu filtern, übergeben Sie einen [Box2D](#) als `filter_bbox`:

- [FilterBBox](#) – Filtert die Ergebnisse, indem es Stellen innerhalb eines Begrenzungsrahmens zurückgibt. Dieser Parameter ist optional.

```
SELECT *
FROM f_SearchPlaceIndexForText('Mount Pleasant', null, 'BOX(-139.06 48.30, -114.03
60.00)::box2d, '{"CAN"}', 5);
```

Weitere Informationen zu PostGIS-Typen und -Funktionen finden Sie in der [PostGIS-Referenz](#).

Anreicherung einer Datenbank mit Adressdaten

Sie können eine formatierte Adresse erstellen und gleichzeitig mithilfe der Amazon Location-Operation normalisieren und geocodieren, `SearchPlaceIndexForText` wenn eine Datenbanktabelle mit den folgenden Daten in die folgenden Spalten unterteilt ist:

- `id`
- `address`
- `city`
- `state`
- `zip`

```
WITH source_data AS (
```

```

SELECT
  id,
  address || ', ' || city || ', ' || state || ', ' || zip AS formatted_address
FROM addresses
),
geocoded_data AS (
  SELECT
    *,
    (f_SearchPlaceIndexForText(formatted_address)).*
  FROM source_data
)
SELECT
  id,
  formatted_address,
  label normalized_address,
  ST_Y(geom) latitude,
  ST_X(geom) longitude
FROM geocoded_data
-- limit the number of rows that will be geocoded; remove this to geocode the entire
table
LIMIT 1;

```

Das folgende Beispiel veranschaulicht eine resultierende Datentabellenzeile:

```

id |          formatted_address          |          normalized_address          |
latitude | longitude |
-----+-----+-----
42 | 123 Anytown Ave N, Seattle, WA | 123 Anytown Ave N, Seattle, WA, 12345, USA |
47.6223000127926 | -122.336745971039
(1 row)

```

Aktualisieren der Datenbanktabelle und Auffüllen von Spalten

Im folgenden Beispiel wird die Tabelle aktualisiert und Spalten mit Ergebnissen von SearchPlaceIndexForText Abfragen gefüllt:

```

WITH source_data AS (
  -- select rows that have not been geocoded and created a formatted address for each
  SELECT
    id,
    address || ', ' || city || ', ' || state || ', ' || zip AS formatted_address
  FROM addresses

```

```
WHERE label IS NULL
-- limit the number of rows that will be geocoded; remove this to geocode the entire
table
LIMIT 1
),
geocoded_data AS (
-- geocode each row and keep it linked to the source's ID
SELECT
    id,
    (f_SearchPlaceIndexForText(formatted_address)).*
FROM source_data
)
UPDATE addresses
-- populate columns
SET
    normalized_address = geocoded_data.label,
    latitude = ST_Y(geocoded_data.geom),
    longitude = ST_X(geocoded_data.geom)
FROM geocoded_data
-- ensure that rows match
WHERE addresses.id = geocoded_data.id;
```

Nächste Schritte

Beispielcode ist im Amazon Location Service-Beispiel-Repository auf verfügbar [GitHub](#), das [eine - AWS CloudFormationVorlage](#) enthält.

Verwalten Ihrer Ortsindexressourcen

Sie können Ihre Ortsindexressourcen mithilfe der Amazon-Standortkonsole, der AWS CLI oder der Amazon-Standort-APIs verwalten.

Auflisten Ihrer Ortsindexressourcen

Sie können Ihre Ortsindex-Ressourcenliste über die Amazon-Standortkonsole, die AWS CLI, die oder die Amazon-Standort-APIs anzeigen:

Console

So zeigen Sie eine Liste der Ortsindexressourcen mit der Amazon-Standortkonsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.

2. Wählen Sie im linken Navigationsbereich Indizes platzieren aus.
3. Zeigen Sie eine Liste Ihrer Ortsindexressourcen unter Meine Ortsindizes an.

API

Verwenden Sie die [ListPlaceIndexes](#) Operation aus den Amazon Location Places APIs .

Das folgende Beispiel ist eine API-Anforderung zum Abrufen einer Liste von Ortsindexressourcen im AWS Konto.

```
POST /places/v0/list-indexes
```

Im Folgenden finden Sie eine Beispielantwort für [ListPlaceIndexes](#):

```
{
  "Entries": [
    {
      "CreateTime": 2020-10-30T01:38:36Z,
      "DataSource": "Esri",
      "Description": "string",
      "IndexName": "ExamplePlaceIndex",
      "UpdateTime": 2020-10-30T01:40:36Z
    }
  ],
  "NextToken": "1234-5678-9012"
}
```

CLI

Verwenden Sie den [list-place-indexes](#)-Befehl.

Das folgende Beispiel ist ein AWS CLI, um eine Liste der Ortsindexressourcen im AWS Konto abzurufen.

```
aws location list-place-indexes
```

Abrufen von Details zu Ortsindexressourcen

Sie können Details zu jeder Ortsindexressource in Ihrem AWS Konto mithilfe der Amazon Location-Konsole AWS CLI, der oder der Amazon Location APIs abrufen:

Console

So zeigen Sie die Details einer Ortsindexressource mit der Amazon Location-Konsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Indizes platzieren aus.
3. Wählen Sie unter Meine Ortsindizes den Namenslink der Zielort-Indexressource aus.

API

Verwenden Sie die [DescribePlaceIndex](#) Operation aus den Amazon Location Place APIs .

Das folgende Beispiel ist eine API-Anforderung zum Abrufen der Ortsindex-Ressourcendetails für *ExamplePlaceIndex*.

```
GET /places/v0/indexes/ExamplePlaceIndex
```

Im Folgenden finden Sie eine Beispielantwort für [DescribePlaceIndex](#):

```
{
  "CreateTime": 2020-10-30T01:38:36Z,
  "DataSource": "Esri",
  "DataSourceConfiguration": {
    "IntendedUse": "SingleUse"
  },
  "Description": "string",
  "IndexArn": "arn:aws:geo:us-west-2:123456789012:place-indexes/ExamplePlaceIndex",
  "IndexName": "ExamplePlaceIndex",
  "Tags": {
    "string" : "string"
  },
  "UpdateTime": 2020-10-30T01:40:36Z
}
```

CLI

Verwenden Sie den [describe-place-index](#)-Befehl.

Das folgende Beispiel ist ein AWS CLI, um die Details zur Ortsindex-Ressource für *abzurufenExamplePlaceIndex*.

```
aws location describe-place-index \  
  --index-name "ExamplePlaceIndex"
```

Löschen einer Ortsindex-Ressource

Sie können eine Ortsindexressource aus Ihrem AWS-Konto mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs löschen:

Console

So löschen Sie eine Ortsindexressource mithilfe der Amazon Location-Konsole

Warning

Dieser Vorgang löscht die Ressource dauerhaft.

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Indizes platzieren aus.
3. Wählen Sie unter Mein Ortsindex die Zielortindexressource aus.
4. Wählen Sie Ortsindex löschen aus.

API

Verwenden Sie die [DeletePlaceIndex](#) Operation aus den Amazon Location Places APIs .

Das folgende Beispiel ist eine API-Anforderung zum Löschen der Ortsindex-Ressource *ExamplePlaceIndex*.

```
DELETE /places/v0/indexes/ExamplePlaceIndex
```

Im Folgenden finden Sie ein Beispiel für eine Erfolgsantwort für [DeletePlaceIndex](#):

```
HTTP/1.1 200
```

CLI

Verwenden Sie den [delete-place-index](#)-Befehl.

Das folgende Beispiel ist ein -AWS CLIBefehl zum Löschen der Ortsindex-Ressource *ExamplePlaceIndex*.

```
aws location delete-place-index \  
  --index-name "ExamplePlaceIndex"
```

Berechnen von Routen mit Amazon Location Service

Mit Amazon Location können Sie einen Datenanbieter für die Berechnung einer Route auswählen, indem Sie eine Routenrechner-Ressource erstellen und konfigurieren.

Sie können die Routenrechner-Ressource verwenden, um [eine Route für bestimmte Parameter mithilfe des SDK oder der REST-API-Endpunkte zu berechnen](#). AWS Verwenden Sie diese Routenrechner-Ressource, um Routen zwischen einem Ursprung, einem Ziel und bis zu 23 Routenpunkten für verschiedene Transportarten, Abstände und Verkehrsbedingungen zu berechnen.

Sie können auch die Routenrechner-Ressource verwenden, um Eingaben für Ihre Routenplanungsalgorithmen oder [-produkte zu erstellen, indem Sie eine Routenmatrix berechnen](#). Berechnen Sie die Reisezeit und die Entfernung zwischen einer Reihe von Verlassenspositionen und einer Reihe von Zielpositionen. Routenplanungssoftware kann diese Zeit- und Entfernungsdaten verwenden, um eine Route oder eine Reihe von Routen zu optimieren, z. B. wenn Sie mehrere Zustellrouten planen und die beste Route und Zeit für jeden Stopp finden möchten. Sie können eine Matrix von Routen für verschiedene Arten von Transporten, Vermeidungen und Verkehrsbedingungen berechnen.

Note

Eine Übersicht über Routing-Konzepte finden Sie unter [Routen](#).

Themen

- [Voraussetzungen](#)
- [Berechnen einer Route](#)
- [Routenplanung mit einer Routenmatrix](#)
- [Positionen, die sich nicht auf einer Straßen befinden](#)
- [Zeit für die Warteschlange](#)

- [Reisemodus](#)
- [Verwalten Ihrer Routenrechner-Ressourcen](#)

Voraussetzungen

Bevor Sie mit der Berechnung von Routen beginnen, führen Sie die erforderlichen Schritte aus:

Themen

- [Erstellen einer Routenrechner-Ressource](#)
- [Authentifizieren Ihrer Anforderungen](#)

Erstellen einer Routenrechner-Ressource

Bevor Sie eine Route berechnen können, erstellen Sie eine Routenrechner-Ressource in Ihrem AWS Konto.

Wenn Sie eine Routenrechner-Ressource erstellen, können Sie aus den verfügbaren Datenanbietern wählen:

1. Esri – Weitere Informationen zur Abdeckung von Esri in Ihrer relevanten Region finden Sie unter [Esri-Details zu Straßennetzen und Verkehrsabdeckung](#).
2. HERE Technologies – Weitere Informationen zur Abdeckung von HERE in Ihrer relevanten Region finden Sie unter [HERE-Abdeckung für das Auto-Routing](#) und [HERE-Abdeckung für das Lkw-Routing](#).
3. Grab – Weitere Informationen zur Abdeckung von Grab finden Sie unter [Abgedeckte Länder/Regionen und Gebiete](#).

Note

Wenn Ihre Anwendung Assets, die Sie in Ihrem Unternehmen verwenden, wie z. B. Lieferfahrzeuge oder Mitarbeiter, verfolgt oder weiterleitet, dürfen Sie Esri nicht als Geolokalisierungsanbieter verwenden. Weitere Informationen finden Sie in Abschnitt 82 der [AWS-Servicebedingungen](#).

Sie können dies über die Amazon Location Service-Konsole, die AWS CLI oder die Amazon Location APIs tun.

Console

So erstellen Sie eine Routenrechner-Ressource mit der Amazon Location-Konsole

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Routenrechner aus.
3. Wählen Sie Routenrechner erstellen aus.
4. Füllen Sie die folgenden Felder aus:
 - Name – Geben Sie einen Namen für die Routenrechner-Ressource ein. Zum Beispiel *ExampleCalculator*. Maximal 100 Zeichen. Gültige Einträge umfassen alphanumerische Zeichen, Bindestriche, Punkte und Unterstriche.
 - Beschreibung – Geben Sie eine optionale Beschreibung ein.
5. Wählen Sie für Datenanbieter einen [Datenanbieter](#) aus, der als Routenrechner verwendet werden soll.
6. (Optional) Geben Sie unter Markierungen einen Tag Key (Schlüssel) und einen Value (Wert) ein. Dadurch wird ein Tag zu Ihrer neuen Routenrechner-Ressource hinzugefügt. Weitere Informationen finden Sie unter [Markieren Ihrer Ressourcen](#).
7. Wählen Sie Routenrechner erstellen aus.

API

So erstellen Sie eine Routenrechner-Ressource mithilfe der Amazon Location APIs

Verwenden Sie die [CreateRouteCalculator](#) Operation aus den Amazon Location Places APIs .

Das folgende Beispiel ist eine API-Anforderung zum Erstellen einer Routenrechner-Ressource mit dem Namen *ExampleCalculator* unter Verwendung des Datenanbieters *Esri* .

```
POST /routes/v0/calculators
Content-type: application/json

{
  "CalculatorName": "ExampleCalculator",
```

```
"DataSource": "Esri",  
"Description": "string",  
"Tags": {  
  "Tag1" : "Value1"  
}  
}
```

AWS CLI

So erstellen Sie eine Routenrechner-Ressource mit -AWS CLIBefehlen

Verwenden Sie den `create-route-calculator`-Befehl.

Im folgenden Beispiel wird eine Routenrechner-Ressource namens *ExampleCalculator* mit *Esri* als Datenanbieter erstellt.

```
aws location \  
  create-route-calculator \  
  --calculator-name "ExampleCalculator" \  
  --data-source "Esri" \  
  --tags Tag1=Value1
```

Note

Die Fakturierung hängt von Ihrer Nutzung ab. Möglicherweise fallen Gebühren für die Nutzung anderer -AWSServices an. Weitere Informationen finden Sie unter [Amazon Location Service – Preise](#).

Authentifizieren Ihrer Anforderungen

Sobald Sie eine Routenrechner-Ressource erstellt haben und bereit sind, Standortfunktionen in Ihrer Anwendung zu erstellen, wählen Sie aus, wie Sie Ihre Anforderungen authentifizieren würden:

- Informationen dazu, wie Sie auf die Services zugreifen können, finden Sie unter [Zugriff auf Amazon Location Service](#).
- Wenn Sie eine Website mit anonymen Benutzern haben, können Sie API-Schlüssel oder Amazon Cognito verwenden.

Beispiel

Das folgende Beispiel zeigt die Verwendung eines API-Schlüssels für die Autorisierung mit [AWS JavaScript SDK v3](#) und dem Amazon Location [Authentifizierungshelfer](#).

```
import { LocationClient, CalculateRouteCommand } from "@aws-sdk/client-location";
import { withAPIKey } from "@aws/amazon-location-utilities-auth-helper";

const apiKey = "v1.public.your-api-key-value"; // API key

// Create an authentication helper instance using an API key
const authHelper = await withAPIKey(apiKey);

const client = new LocationClient({
  region: "<region>", // region containing Cognito pool
  ...authHelper.getLocationClientConfig(), // Provides configuration required to make
  requests to Amazon Location
});

const input = {
  CalculatorName: "ExampleCalculator",
  DeparturePosition: [-123.4567, 45.6789],
  DestinationPosition: [-123.123, 45.234],
};

const command = new CalculateRouteCommand(input);

const response = await client.send(command);
```

Berechnen einer Route

Sie können Amazon Location Service verwenden, um Routen zwischen einem Ursprung und einem Ziel mit bis zu 23 Routenpunkten entlang der Route für verschiedene Arten von Transporten, Abwägungen und Verkehrsbedingungen zu berechnen.

Note

Sie müssen zunächst eine Routenrechner-Ressource erstellen und die Authentifizierung für Ihre Anfragen an Amazon Location einrichten. Weitere Informationen finden Sie unter [Voraussetzungen](#).

Beginnen Sie mit der Berechnung von Routen

Senden Sie eine einfache Anfrage mithilfe der [-CalculateRoute](#) Operation. Eine einfache -Anforderung enthält die folgenden Pflichtfelder:

- **DeparturePosition** – Die Startposition, aus der die Route berechnet werden soll. Definiert als [longitude, latitude]
- **DestinationPosition** – Die Endposition, an der die Route berechnet werden soll. Definiert als [longitude, latitude].

Note

Wenn Sie eine Verlassen- oder Zielposition angeben, die sich nicht auf einer Straßen befindet, [verschiebt Amazon Location die Position auf die nächste Straßen](#) .

Optional können Sie [in Ihrer Anfrage Wegpunkte](#) , eine [Verlassenszeit](#) und einen [Reisemodus](#) angeben.

Sie können die AWS CLI oder die Amazon Location APIs verwenden.

API

Das folgende Beispiel ist eine `-CalculateRoute` Anforderung mit der Routenrechner-Ressource *ExampleCalculator*. Die Anforderung gibt die Berechnung einer Route von einer Ausgangsposition `[-122.7565, 49.0021]` zu einer Zielposition `[-122.3394, 47.6159]` an.

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-122.7565,49.0021],
  "DestinationPosition": [-122.3394, 47.6159]
}
```

AWS CLI

Das folgende Beispiel ist ein `calculate-route` Befehl, der die Routenrechner-Ressource verwendet *ExampleCalculator*. Die Anforderung gibt die Berechnung einer Route von einer Ausgangsposition `[-122.7565, 49.0021]` zu einer Zielposition `[-122.3394, 47.6159]` an.

```
aws location \
  calculate-route \
    --calculator-name ExampleCalculator \
    --departure-position -122.7565 49.0021 \
    --destination-position -122.3394 47.6159
```

Standardmäßig gibt die Antwort `Distance` in Kilometern zurück. Sie können die Maßeinheit mit dem folgenden optionalen Parameter in Fuß ändern:

- `DistanceUnit` – Gibt das Einheitensystem an, das für die Abstandsergebnisse verwendet werden soll.

Example

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-122.7565,49.0021],
  "DestinationPosition": [-122.3394, 47.6159],
  "DistanceUnit": "Miles"
}
```

Festlegen von Wegpunkten

Bei der Berechnung einer Route können Sie mithilfe von Wegpunktpositionen bis zu 23 Zwischen-Stopover-Punkten zwischen der Ausgangsposition und der Zielposition angeben.

- `WaypointPositions` – Gibt eine geordnete Liste von Zwischenpositionen an, die entlang einer Route zwischen der Ausgangsposition und der Zielposition aufgenommen werden sollen.

Note

Wenn Sie eine Wegpunktposition angeben, die sich nicht auf einer Straße befindet, verschiebt Amazon Location die Position auf die nächste Straße.

Example

Die folgende [CalculateRoute](#) Anforderung berechnet eine Route mit 2 Wegpunkten:

- Die Startposition ist [-122.7565, 49.0021], und die Zielposition ist [-122.3394, 47.6159].
- Für den Anforderungsparameter `WaypointPositions`:
 - Der erste Stopp über Position ist [-122.1884, 48,0936].
 - Der zweite Stopp über Position ist [-122.3493, 47,6205].
- Setzen Sie den folgenden optionalen Parameter auf „*true*“, um die Geometrie des Trichter-Linestrings zwischen diesen beiden Wegpunkten einzuschließen:
 - `IncludeLegGeometry` – Schließt die Geometrie jedes Pfads zwischen einem Paar von Positionen in die Antwort ein.

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-122.7565,49.0021],
  "DestinationPosition": [-122.3394, 47.6159],
  "WaypointPositions":[
    [-122.1884,48.0936],
    [-122.3493,47.6205]
  ],
  "IncludeLegGeometry": true
}
```

Beispielantwort

Im Folgenden finden Sie eine Beispielanforderung mit der entsprechenden Antwort beim Aufrufen der [-CalculateRoute](#) Operation von der Amazon Location Routes API mit dem `IncludeLegGeometry` Wert „*true*“, der die Linestring-Geometrie jedes Pfads zwischen einem Paar von Positionen in der Antwort enthält.

Example request

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-122.7565,49.0021],
  "DestinationPosition": [-122.3394, 47.6159],
  "IncludeLegGeometry": true
}
```

Example response

```
{
  "Legs": [
    {
      "Distance": 178.5,
      "DurationSeconds": 6480,
      "EndPosition": [-122.3394,47.6159],
      "Geometry": {
        "LineString": [
          [-122.7565,49.0021],
          [-122.3394,47.6159]
        ]
      },
      "StartPosition": [-122.7565,49.0021],
      "Steps": [
        {
          "Distance": 178.5,
          "DurationSeconds": 6480,
          "EndPosition": [-122.3394,47.6159],
          "GeometryOffset": 0,
          "StartPosition": [-122.7565,49.0021]
        }
      ]
    }
  ],
  "Summary": {
    "DataSource": "Esri",
    "Distance": 178.5,
    "DistanceUnit": "Kilometers",
    "DurationSeconds": 6480,
    "RouteBBox": [
      -122.7565,49.0021,
      -122.3394,47.6159
    ]
  }
}
```

Routenplanung mit einer Routenmatrix

Sie können Amazon Location Service verwenden, um Eingaben für Ihre Routenplanungs- und Optimierungssoftware zu erstellen. Sie können Routenergebnisse, einschließlich der Zeit und

Entfernung, für Routen zwischen einer Reihe von Abstiegspositionen und einer Reihe von Zielpositionen erstellen.

Angenommen, es handelt sich um die Abgangspositionen A und B sowie die Zielpositionen X und Y. Amazon Location Service gibt die Zeit und Entfernung für Routen von A zu X, A zu Y, B zu X und B zu Y zurück.

Sie können die Routen mit unterschiedlichen Transportarten, Abstammungen und Verkehrsbedingungen berechnen. Sie können beispielsweise angeben, dass das Fahrzeug ein Lkw mit einer Länge von 35 Fuß ist, und die berechnete Route verwendet diese Einschränkungen, um die Fahrtzeit und die Entfernung zu bestimmen.

Die Anzahl der zurückgegebenen Ergebnisse (und berechneten Routen) ist die Anzahl der Abstiegspositionen multipliziert mit der Anzahl der Zielpositionen. Ihnen wird jede berechnete Route in Rechnung gestellt, nicht jede Anforderung an den Service. Daher wird eine Routenmatrix mit 10 Abgängen und 10 Zielen als 100 Routen abgerechnet.

Berechnen einer Routenmatrix

Sie können eine Matrix von Routen zwischen einer Reihe von Abstiegspositionen und einer Reihe von Zielpositionen berechnen. Die Routenergebnisse beinhalten die Reisezeit und die Entfernung.

Voraussetzung

- Sie müssen zunächst eine Routenrechner-Ressource erstellen und die Authentifizierung für Ihre Anfragen an Amazon Location einrichten. Weitere Informationen finden Sie unter [Voraussetzungen](#).

Senden Sie eine -Anforderung mithilfe der [-CalculateRouteMatrix](#) Operation. Eine minimale Anforderung enthält die folgenden Pflichtfelder:

- `DeparturePositions` – Der Satz von Startpositionen, für die die Routen berechnet werden sollen. Definiert als Array von `[longitude, latitude]`
- `DestinationPositions` – Der Satz von Endpositionen, für die die Routen berechnet werden sollen. Definiert als Array von `[longitude, latitude]`.

Note

Wenn Sie eine Verlassen- oder Zielposition angeben, die sich nicht auf einer Straßen befindet, [verschiebt Amazon Location die Position auf die nächste Straßen](#).

Sie können optional eine [Ankunftszeit](#) und einen [Reisemodus](#) in Ihrer Anfrage angeben.

Sie können die AWS CLI oder die Amazon Location APIs verwenden.

API

Das folgende Beispiel ist eine `-CalculateRouteMatrix`Anforderung mit der Routenrechner-Ressource `ExampleCalculator`. Die Anforderung gibt die Berechnung der Matrix der Routen von den Abgangspositionen `[-122.7565, 49.0021]` und `[-122.2014, 47.6101]` zu den Zielpositionen `[-122.3394, 47.6159]` und `[-122.4813, 48.7511]` an.

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route-matrix
Content-type: application/json
{
  "DeparturePositions": [
    [-122.7565, 49.0021],
    [-122.2014, 47.6101]
  ],
  "DestinationPositions": [
    [-122.3394, 47.6159],
    [-122.4813, 48.7511]
  ]
}
```

AWS CLI

Das folgende Beispiel ist ein `calculate-route-matrix` Befehl, der die Routenrechner-Ressource verwendet `ExampleCalculator`. Die Anforderung gibt die Berechnung der Matrix der Routen von den Abgangspositionen `[-122.7565, 49.0021]` und `[-122.2014, 47.6101]` zu den Zielpositionen `[-122.3394, 47.6159]` und `[-122.4813, 48.7511]` an.

```
aws location \
  calculate-route-matrix \
    --calculator-name ExampleCalculator \
    --departure-positions "[[-122.7565, 49.0021], [-122.2014, 47.6101]]" \
```

```
--destination-positions "[[-122.3394,47.6159],[-122.4813,48.7511]]"
```

Standardmäßig gibt die Antwort Distance in Kilometern zurück. Mit dem folgenden optionalen Parameter können Sie die Maßeinheit in Fuß ändern:

- `DistanceUnit` – Gibt das Einheitensystem an, das für die Abstandsergebnisse verwendet werden soll.

Example

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route-matrix
Content-type: application/json
{
  "DeparturePositions": [
    [-122.7565,49.0021],
    [-122.2014,47.6101]
  ],
  "DestinationPositions": [
    [-122.3394, 47.6159],
    [-122.4813,48.7511]
  ],
  "DistanceUnit": "Miles"
}
```

Einschränkungen bei der Ankunft und bei Zielpositionen

Bei der Berechnung einer Routenmatrix gibt es Einschränkungen in Bezug auf die Ausgangs- und Zielpositionen. Diese Einschränkungen variieren je nach Anbieter, der von der `RouteCalculator` Ressource verwendet wird.

Einschränkung	Esri	Grab	Hier
Anzahl der Positionen	Bis zu 10 Ausgangspositionen und 10 Zielpositionen.	Bis zu 350 Ausgangspositionen und 350 Zielpositionen.	Bis zu 350 Ausgangspositionen und 350 Zielpositionen. Für längere Routen gelten zusätzliche Einschränkungen.

Einschränkung	Esri	Grab	Hier
			<p>Weitere Informationen finden Sie im Abschnitt.</p>
Abstand zwischen Positionen	<p>Jedes Paar von Verlassen- und Zielpositionen muss sich innerhalb von 400 km voneinander befinden (40 km für Walking-Routen).</p>		<p>Alle Eingangs- und Zielpositionen müssen innerhalb eines 180 km langen Trichterkreises liegen.</p> <p>Für längere Routen gelten zusätzliche Einschränkungen. Weitere Informationen finden Sie im Abschnitt.</p>
Routenlänge	<p>Routen werden nicht abgeschlossen, wenn die Gesamtreisezeit für die Route mehr als 400 Minuten beträgt.</p>		<p>Routen, die mehr als 10 km außerhalb eines Kreises um die Ausgangspunkte und Zielpunkte herum abweichen, werden nicht berechnet.</p> <p>Für längere Routen gelten zusätzliche Einschränkungen. Weitere Informationen finden Sie im Abschnitt.</p>

Einschränkung	Esri	Grab	Hier
Regionen	Die Berechnung einer Routenmatrix wird in Korea nicht unterstützt.	Verfügbar in Südostasien. Eine Liste der unterstützten Länder/Regionen und weitere Informationen finden Sie unter Abgedeckte Länder/Regionen und Gebiete .	Keine zusätzlichen Einschränkungen.

Längere Routenplanung

Die Berechnung einer Matrix von Routenergebnissen ist für eine effiziente Routenplanung nützlich, aber die Berechnung kann einige Zeit in Anspruch nehmen. Alle Amazon Location Service-Datenanbieter setzen Einschränkungen für die Anzahl der Routen oder die Entfernung der Routen fest, die berechnet werden können. HERE ermöglicht beispielsweise das Erstellen von Routen zwischen 350 Eingangs- und Zielpositionen, diese Positionen müssen jedoch innerhalb eines 180-km-Bereichs liegen. Was ist, wenn Sie mit längeren Routen planen möchten?

Sie können eine Matrix von Routen mit uneingeschränkten Längen für eine geringere Anzahl von Routen berechnen, indem Sie einen `RouteCalculator` mit HERE als Datenanbieter verwenden. Dies ändert nichts daran, wie Sie die [CalculateRouteMatrix](#) API aufrufen. Amazon Location lässt einfach längere Routen zu, wenn Sie die Anforderungen erfüllen.

Die Anforderungen für Routenberechnungen mit längerer Länge sind:

- Der `RouteCalculator` muss den HERE-Datenanbieter verwenden.
- Die Anzahl der Abstiegspositionen darf nicht größer als 15 sein.
- Die Gesamtzahl der zu berechnenden Routen darf nicht größer als 100 sein.
- Langdistanz-Routing ist für Lkw-Routing mit Gebührenvermeidungen nicht zulässig, wenn die Routen größer als 1.000 km sind. Diese Kombination ist langsamer zu berechnen und kann dazu führen, dass der Aufruf eine Zeitüberschreitung hat. Sie können diese Routen mit der [-CalculateRoute](#) Operation einzeln berechnen.

Wenn Ihr Aufruf diese Anforderungen nicht erfüllt (z. B. wenn Sie 150 Routenberechnungen in einem einzigen Aufruf anfordern), `CalculateRouteMatrix` kehrt dazu zurück, nur die kürzeren Routenregeln zuzulassen. Anschließend können Sie die Routen berechnen, solange sich die Positionen in einem 180-km-Kreis befinden.

Beachten Sie bei der Berechnung längerer Routen die folgenden Punkte:

- Längere Routen können länger dauern als die maximale Zeit für Amazon Location APIs . Wenn Sie häufige Timeouts mit bestimmten Routen erhalten, können Sie bei jedem Aufruf von eine geringere Anzahl von Routen ausprobieren `CalculateRouteMatrix`.
- Wenn Sie Ihrer `CalculateRouteMatrix` Anfrage weitere Ziel- oder Ausgangspositionen hinzufügen, kann der Vorgang in den eingeschränkteren Modus wechseln und Sie können eine Fehlermeldung für eine Route erhalten, die problemlos berechnet werden kann, wenn weniger Routen erstellt werden müssen. Reduzieren Sie in diesem Fall die Anzahl der Ziel- oder Ausgangspositionen und stellen Sie mehrere Anfragen, um den gesamten Satz von Routenberechnungen zu erhalten, die Sie benötigen.

Beispielantwort

Im Folgenden finden Sie eine Beispielanforderung mit der entsprechenden Antwort beim Aufrufen der [-CalculateRouteMatrix](#) Operation von der Amazon Location Routes API aus.

Example request

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route-matrix
Content-type: application/json
{
  "DeparturePositions": [
    [-122.7565,49.0021],
    [-122.2014,47.6101]
  ],
  "DestinationPositions": [
    [-122.3394, 47.6159],
    [-122.4813,48.7511]
  ]
}
```

Example response

```
{
```

```
"RouteMatrix": [
  [
    {
      "Distance": 178.764,
      "DurationSeconds": 7565
    },
    {
      "Distance": 39.795,
      "DurationSeconds": 1955
    }
  ],
  [
    {
      "Distance": 15.31,
      "DurationSeconds": 1217
    },
    {
      "Distance": 142.506,
      "DurationSeconds": 6279
    }
  ]
],
"Summary": {
  "DataSource": "Here",
  "RouteCount": 4,
  "ErrorCount": 0,
  "DistanceUnit": "Kilometers"
}
}
```

Positionen, die sich nicht auf einer Straßen befinden

Wenn Sie `CalculateRoute` oder `verwendenCalculateRouteMatrix`, verschiebt Amazon Location die Position in eine nahegelegene Straße, wenn Sie eine Position für Hin- und Rückfahrt, Ziel oder Wegpunkt angeben, die sich nicht auf einer Straßen befindet.

Die folgende [CalculateRoute](#) Anforderung gibt eine Verlassen- und Zielposition an, die sich nicht auf einer Straßen befindet:

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
```

```
"DeparturePosition": [-123.128014, 49.298472],
"DestinationPosition": [-123.134701, 49.294315]
}
```

Die resultierende Antwort gibt eine Position zurück, die auf eine nahegelegene Straßen eingefangen wurde:

```
{
  "Legs": [
    {
      "StartPosition": [-123.12815, 49.29717],
      "EndPosition": [-123.13375, 49.2926],
      "Distance": 4.223,
      "DurationSeconds": 697,
      "Steps": [
        {
          "StartPosition": [ -123.12815, 49.29717 ],
          "EndPosition": [ -123.12806, 49.29707 ],
          "Distance": 0.013,
          "DurationSeconds": 8
        },
        {
          "StartPosition": [ -123.12806, 49.29707 ],
          "EndPosition": [ -123.1288, 49.29659 ],
          "Distance": 0.082,
          "DurationSeconds": 36
        },
        {
          "StartPosition": [ -123.1288, 49.29659 ],
          "EndPosition": [ -123.12021, 49.29853 ],
          "Distance": 0.742,
          "DurationSeconds": 128
        },
        {
          "StartPosition": [ -123.12021, 49.29853 ],
          "EndPosition": [ -123.1201, 49.29959 ],
          "Distance": 0.131,
          "DurationSeconds": 26
        },
        {
          "StartPosition": [ -123.1201, 49.29959 ],
          "EndPosition": [ -123.13562, 49.30681 ],
          "Distance": 1.47,

```

```

        "DurationSeconds": 238
    },
    {
        "StartPosition": [ -123.13562, 49.30681 ],
        "EndPosition": [ -123.13693, 49.30615 ],
        "Distance": 0.121,
        "DurationSeconds": 28
    },
    {
        "StartPosition": [ -123.13693, 49.30615 ],
        "EndPosition": [ -123.13598, 49.29755 ],
        "Distance": 0.97,
        "DurationSeconds": 156
    },
    {
        "StartPosition": [ -123.13598, 49.29755 ],
        "EndPosition": [ -123.13688, 49.29717 ],
        "Distance": 0.085,
        "DurationSeconds": 15
    },
    {
        "StartPosition": [ -123.13688, 49.29717 ],
        "EndPosition": [ -123.13375, 49.2926 ],
        "Distance": 0.609,
        "DurationSeconds": 62
    }
]
}
],
"Summary": {
    "RouteBBox": [ -123.13693, 49.2926, -123.1201, 49.30681 ],
    "DataSource": "Here",
    "Distance": 4.223,
    "DurationSeconds": 697,
    "DistanceUnit": "Kilometers"
}
}

```

Zeit für die Warteschlange

Wenn Sie `CalculateRoute` oder `CalculateRouteMatrix` aufrufen und in der Anforderung keine Abgangszeit angeben, spiegeln die berechneten Routen standardmäßig die optimalen Datenverkehrsbedingungen wider.

Sie können eine bestimmte Abgangszeit festlegen, um Live- und prädiktive Datenverkehrsbedingungen von Ihrem ausgewählten Datenanbieter zu verwenden, indem Sie eine der folgenden Optionen verwenden:

- `DepartNow` – Wenn der Wert auf „*true*“ gesetzt ist, werden Live-Datenverkehrsbedingungen verwendet, um den schnellsten Weg zu berechnen.
- `DepartureTime` – Wenn angegeben, werden prädiktive und bekannte Datenverkehrsbedingungen für die angeforderte Zeit verwendet. Definiert im folgenden [Format](#): `YYYY-MM-DDThh:mm:ss.sssZ`.

Example

Die folgende [CalculateRoute](#) Anfrage legt die Abreisezeit auf den 2. Juli 2024 um 12:15:20 Uhr UTC fest.

```
POST /routes/v0/calculators/ExampleCalculator/calculate/route
Content-type: application/json
{
  "DeparturePosition": [-122.7565,49.0021],
  "DestinationPosition": [-122.3394, 47.6159],
  "WaypointPositions":[
    [-122.1884,48.0936],
    [-122.3493,47.6205]
  ]
  "IncludeLegGeometry": true,
  "DepartureTime": 2024-07-02T12:15:20.000Z,
}
```

Reisemodus

Sie können einen Reisemodus festlegen, wenn Sie `CalculateRoute` oder `CalculateRouteMatrix` verwenden. Die Art der Reise wirkt sich auf die Geschwindigkeit der Reise und die Straßenkompatibilität aus. Während der Standardreisemodus auf dem Auto basiert, können Sie mit dem folgenden optionalen Parameter angeben, welchen Reisemodus Sie während der Reise entlang einer Route verwenden:

- `TravelMode` – Gibt den Transportmodus bei der Berechnung einer Route an, z. B.: *Bicycle*, *Car*, *MotorcycleTruck*, oder *Walking*.

Einschränkungen

- Wenn Sie `Walking` für den Reisemodus angeben und Ihr Datenanbieter Esri ist, müssen Start und Ziel innerhalb von 40 km liegen.
- `Bicycle` oder `Motorcycle` sind nur verfügbar, wenn Grab als Datenanbieter verwendet wird.
- Grab bietet nur `-Bicycle` und `-Walking`Routen in bestimmten Städten. Weitere Informationen finden Sie unter [Abgedeckte Länder/Regionen und Gebiete](#).
- `Truck` ist nicht verfügbar, wenn Grab als Datenanbieter verwendet wird.

Zusätzliche Einstellungen

Wenn Sie einen `TravelMode` von `Car` angeben, können Sie zusätzliche Routeneinstellungen mit dem folgenden optionalen Parameter angeben:

- `CarModeOptions` – Gibt Routeneinstellungen an, wenn Sie in einem Auto unterwegs sind, z. B. `AvoidFerries` oder `AvoidTolls`.

Wenn Sie einen `TravelMode` von `Truck` angeben, können Sie zusätzliche Routeneinstellungen mit dem folgenden optionalen Parameter angeben:

- `TruckModeOptions` – Gibt Routeneinstellungen bei Fahrten in einem Lkw an, z. B. `AvoidFerries` oder `AvoidTolls`, zusätzlich zur Angabe von Routen, die für die `TruckDimensions` und geeignet sind `TruckWeight`.

Example

Die folgende [CalculateRoute](#) Anforderung gibt `Truck` als Reisemodus an. Zu den zusätzlichen Routenbeschränkungen gehören: Vermeidung von Routen, die Seeflotten verwenden, und Vermeidung von Straßen, die die Lkw-Dimensionen und das Gewicht nicht berücksichtigen können.

```
{
  "DeparturePosition": [-122.7565,49.0021],
  "DestinationPosition": [-122.3394, 47.6159],
  "DepartNow": true,
  "TravelMode": "Truck",
  "TruckModeOptions": {
    "AvoidFerries": true,
    "AvoidTolls": false,
```

```
    "Dimensions": {
      "Height": 4.5,
      "Length": 15.5,
      "Unit": "Meters",
      "Width": 4.5
    },
    "Weight": {
      "Total": 7500,
      "Unit": "Pounds"
    }
  }
}
```

Verwalten Ihrer Routenrechner-Ressourcen

Sie können Ihre Routenrechner-Ressourcen mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs verwalten.

Auflisten Ihrer Routenrechner-Ressourcen

Sie können Ihre Routenrechnerliste über die Amazon-Standortkonsole, die AWS CLI oder die Amazon-Standort-APIs anzeigen:

Console

So zeigen Sie eine Liste der Routenrechner mit der Amazon-Standortkonsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Routenrechner aus.
3. Zeigen Sie die Details des Routenrechners unter Meine Routenrechner an.

API

Verwenden Sie die [ListRouteCalculators](#) Operation aus den Amazon Location Routes APIs .

Das folgende Beispiel ist eine API-Anforderung zum Abrufen einer Liste von Routenrechnern im AWS Konto.

```
POST /routes/v0/list-calculators
```

Im Folgenden finden Sie eine Beispielantwort für [ListRouteCalculators](#):

```
{
  "Entries": [
    {
      "CalculatorName": "ExampleCalculator",
      "CreateTime": 2020-09-30T22:59:34.142Z,
      "DataSource": "Esri",
      "Description": "string",
      "UpdateTime": 2020-09-30T23:59:34.142Z
    }
  ],
  "NextToken": "1234-5678-9012"
}
```

CLI

Verwenden Sie den `list-route-calculators`-Befehl.

Das folgende Beispiel ist ein AWS CLI, um eine Liste der Routenrechner im AWS Konto abzurufen.

```
aws location list-route-calculators
```

Abrufen von Routenrechnerdetails

Sie können Details zu jeder Routenrechner-Ressource in Ihrem AWS Konto mithilfe der Amazon Location-Konsole, der AWS CLI, oder der Amazon Location APIs abrufen:

Console

So zeigen Sie die Details eines Routenrechners mit der Amazon Location-Konsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Routenrechner aus.
3. Wählen Sie unter Meine Routenrechner den Namenslink des Zielroutenrechners aus.

API

Verwenden Sie die [-DescribeRouteCalculator](#) Operation aus den Amazon Location Routes APIs .

Das folgende Beispiel ist eine API-Anforderung zum Abrufen der Routenrechner-Details für *ExampleCalculator*.

```
GET /routes/v0/calculators/ExampleCalculator
```

Im Folgenden finden Sie eine Beispielantwort für [DescribeRouteCalculator](#):

```
{
  "CalculatorArn": "arn:aws:geo:us-west-2:123456789012:route-
calculator/ExampleCalculator",
  "CalculatorName": "ExampleCalculator",
  "CreateTime": 2020-09-30T22:59:34.142Z,
  "DataSource": "Esri",
  "Description": "string",
  "Tags": {
    "Tag1" : "Value1"
  },
  "UpdateTime": 2020-09-30T23:59:34.142Z
}
```

CLI

Verwenden Sie den `describe-route-calculator`-Befehl.

Das folgende Beispiel ist ein AWS CLI, um die Routenrechner-Details für *ExampleCalculator* abzurufen.

```
aws location describe-route-calculator \
  --calculator-name "ExampleCalculator"
```

Löschen eines Routenrechners

Sie können einen Routenrechner aus Ihrem AWS Konto mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs löschen:

Console

So löschen Sie einen Routenrechner mithilfe der Amazon Location-Konsole

Warning

Dieser Vorgang löscht die Ressource dauerhaft.

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Routenrechner aus.
3. Wählen Sie unter Meine Routenrechner den Zielroutenrechner aus.
4. Wählen Sie Routenrechner löschen aus.

API

Verwenden Sie die [DeleteRouteCalculator](#) Operation aus den Amazon Location Routes APIs .

Das folgende Beispiel ist eine API-Anforderung zum Löschen der Geofence-Sammlung *ExampleCalculator*.

```
DELETE /routes/v0/calculators/ExampleCalculator
```

Im Folgenden finden Sie eine Beispielantwort für [DeleteRouteCalculator](#):

```
HTTP/1.1 200
```

CLI

Verwenden Sie den `delete-route-calculator`-Befehl.

Das folgende Beispiel ist ein `-AWS CLI` Befehl zum Löschen der Geofence-Sammlung *ExampleCalculator*.

```
aws location delete-route-calculator \  
    --calculator-name "ExampleCalculator"
```

Geofencing eines Interessengebiets mit Amazon Location

Eine Geofencing-Anwendung bewertet die Position eines verfolgten Geräts im Verhältnis zu zuvor registrierten Interessenbereichen. Auf diese Weise können Aktionen basierend auf Positionsaktualisierungen durchgeführt werden. Sie können beispielsweise ein Ereignis initiieren, das eine Benachrichtigung auslöst, wenn sich ein Kunde, der in seiner mobilen App bestellt hat, in der Nähe eines Geschäfts befindet.

Note

Eine Übersicht über Geofencing und Tracker-Konzepte finden Sie unter [Geofences und Tracker](#).

Dieser Abschnitt des Handbuchs enthält step-by-step Anweisungen zum Erstellen einer Geofencing-Anwendung mit Amazon Location Service.

Übersicht über die Schritte

1. Fügen Sie Geofences um Interessenbereiche hinzu und speichern Sie sie in einer Geofence-Sammlungsressource.
2. Beginnen Sie mit der Nachverfolgung Ihrer Zielgeräte und speichern Sie den Gerätestandortverlauf in einer Tracker-Ressource.
3. Verknüpfen Sie Ihre Tracker-Ressource mit Ihrer Geofence-Erfassungsressource, sodass Gerätestandortaktualisierungen automatisch anhand all Ihrer Geofences ausgewertet werden.
4. Sie können die Gerätepositionen direkt anhand Ihrer Geofence-Sammlungsressourcen auswerten, wenn Sie Amazon Location Trackers nicht verwenden möchten, um den Standortverlauf Ihrer Geräte beizubehalten.

Nachdem Sie Ihre Geofencing-Lösung implementiert haben, gibt Ihre Geofence-Sammlungsressource die folgenden Ereignisse aus:

- ENTER – Ein verfolgtes Gerät gelangt in einen Geofence innerhalb einer Geofence-Sammlung.
- EXIT – Ein verfolgtes Gerät beendet einen Geofence innerhalb einer Geofence-Sammlung.

Sie können Amazon verwenden EventBridge , um auf Ereignisse zu reagieren, indem Sie sie an eine andere Stelle weiterleiten.

Alternativ zum Senden von Updates über die Amazon Location Service APIs von jedem Gerät aus können Sie MQTT verwenden, um Geräteaktualisierungen zu senden.

In den folgenden Themen werden diese Schritte und Alternativen ausführlich beschrieben.

Themen

- [Geofences hinzufügen](#)
- [Starten der Nachverfolgung](#)
- [Verknüpfen eines Trackers mit einer Geofence-Sammlung](#)
- [Gerätepositionen anhand von Geofences bewerten](#)
- [Reagieren auf Amazon Location Service-Ereignisse mit Amazon EventBridge](#)
- [Nachverfolgung mit AWS IoT und MQTT mit Amazon Location Service](#)
- [Verwalten Ihrer Geofence-Sammlungsressourcen](#)
- [Verwalten Ihrer Tracker-Ressourcen](#)

Geofences hinzufügen

Geofences enthalten Punkte und Scheitelpunkte, die eine geschlossene Grenze bilden, die einen Interessenbereich definiert. Geofence-Sammlungen speichern und verwalten einen oder mehrere Geofences.

Amazon Location Geofence Collections speichert Geofences, die mithilfe eines standardmäßigen Geodatenformats namens [GeoJSON \(RFC 7946\)](#) definiert wurden. Sie können Tools wie [geojson.io](#) kostenlos verwenden, um Ihre Geofences grafisch zu zeichnen und die GeoJSON-Ausgabedatei zu speichern.

Note

Amazon Location unterstützt keine Polygone mit Lücken, Multipolygonen, Uhrzeigersinn-Polygonen und Geofences, die den Antimeridian durchqueren.

Erstellen einer Geofence-Sammlung

Erstellen Sie eine Geofence-Sammlung zum Speichern und Verwalten von Geofences mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs.

Console

So erstellen Sie eine Geofence-Sammlung mit der Amazon Location-Konsole

1. Öffnen Sie die Amazon Location Service-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Geofence-Sammlungen aus.
3. Wählen Sie Geofence-Sammlung erstellen aus.
4. Füllen Sie die folgenden Felder aus:
 - Name – Geben Sie einen eindeutigen Namen ein. Zum Beispiel *ExampleGeofenceCollection*. Maximal 100 Zeichen. Gültige Einträge umfassen alphanumerische Zeichen, Bindestriche, Punkte und Unterstriche.
 - Beschreibung – Geben Sie eine optionale Beschreibung ein, um Ihre Ressourcen zu unterscheiden.
5. Unter EventBridge Regel mit CloudWatch als Ziel können Sie eine optionale EventBridge Regel erstellen, um auf [Geofence-Ereignisse zu reagieren](#). Auf diese Weise kann Amazon Location Ereignisse in Amazon CloudWatch Logs veröffentlichen.
6. (Optional) Geben Sie unter Markierungen einen Tag Key (Schlüssel) und einen Value (Wert) ein. Dadurch wird Ihrer neuen Geofence-Sammlung ein Tag hinzugefügt. Weitere Informationen finden Sie unter [Markieren Ihrer Amazon Location Service-Ressourcen](#).
7. (Optional) Unter Kundenverwaltete Schlüsselverschlüsselung können Sie wählen, ob Sie einen vom Kunden verwalteten Schlüssel hinzufügen möchten. Dadurch wird ein symmetrischer kundenverwalteter Schlüssel hinzugefügt, den Sie über die standardmäßige AWS-eigene Verschlüsselung erstellen, besitzen und verwalten. Weitere Informationen finden Sie unter [Verschlüsseln von Daten im Ruhezustand](#).
8. Wählen Sie Geofence-Sammlung erstellen aus.

API

So erstellen Sie eine Geofence-Sammlung mithilfe der Amazon Location APIs

Verwenden Sie die [CreateGeofenceCollection](#) Operation aus den Amazon Location Geofences APIs .

Im folgenden Beispiel wird eine API-Anfrage verwendet, um eine Geofence-Sammlung namens *ExampleGeofenceCollection* zu erstellen. Die Geofence-Sammlung ist einem [vom Kunden verwalteten AWS KMS Schlüssel](#) zugeordnet, um Kundendaten zu verschlüsseln.

```
POST /geofencing/v0/collections
Content-type: application/json

{
  "CollectionName": "ExampleGeofenceCollection",
  "Description": "Geofence collection 1 for shopping center",
  "KmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
  "Tags": {
    "Tag1" : "Value1"
  }
}
```

AWS CLI

So erstellen Sie eine Geofence-Sammlung mit -AWS CLIBefehlen

Verwenden Sie den [create-geofence-collection](#)-Befehl.

Im folgenden Beispiel wird ein verwendet AWS CLI, um eine Geofence-Sammlung namens zu erstellen *ExampleGeofenceCollection*. Die Geofence-Sammlung ist einem [vom Kunden verwalteten AWS KMS Schlüssel](#) zugeordnet, um Kundendaten zu verschlüsseln.

```
aws location \
  create-geofence-collection \
  --collection-name "ExampleGeofenceCollection" \
  --description "Shopping center geofence collection" \
  --kms-key-id "1234abcd-12ab-34cd-56ef-1234567890ab" \
  --tags Tag1=Value1
```

Note

Die Fakturierung hängt von Ihrer Nutzung ab. Möglicherweise fallen Gebühren für die Nutzung anderer -AWSServices an. Weitere Informationen finden Sie unter [Amazon Location Service – Preise](#).

Geofences entwerfen

Nachdem Sie Ihre Geofence-Sammlung erstellt haben, können Sie Ihre Geofences definieren. Geofences sind entweder als Polygon oder als Kreis definiert. Um ein Polygon-Geofence zu zeichnen, können Sie ein GeoJSON-bearbeitungstool wie geojson.io verwenden.

Um einen Geofence als Kreis zu erstellen, müssen Sie den Mittelpunkt des Kreises und den Radius definieren. Wenn Sie beispielsweise ein Geofence erstellen möchten, das benachrichtigt wird, wenn ein Gerät innerhalb von 50 Metern von einem bestimmten Standort kommt, würden Sie den Breiten- und Längengrad dieses Standorts verwenden und den Radius als 50 Meter angeben.

Mit den Amazon Location Service APIs können Sie Ihrem Geofence auch Metadaten in Form von Schlüssel-Wert-Paaren hinzufügen. Diese können nützlich sein, um Informationen über den Geofence zu speichern, z. B. seinen Typ oder andere Informationen, die für Ihre Anwendung spezifisch sind. Sie können diese Metadaten verwenden, wenn [Reagieren auf Amazon Location Service-Ereignisse mit Amazon EventBridge](#).

Hinzufügen von Polygon-Geofences

In diesem Abschnitt wird das Erstellen von Polygon-Geofences beschrieben

Zeichnen von Geofences mit einem GeoJSON-Tool

Nachdem Sie Ihre Geofence-Sammlung erstellt haben, können Sie Ihre Geofences mithilfe eines GeoJSON-Editor-Tools wie geojson.io definieren.

So erstellen Sie eine GeoJSON-Datei

1. Öffnen Sie ein GeoJSON-bearbeitungstool. Zum Beispiel geojson.io.
2. Wählen Sie das Symbol Polygon zeichnen und ziehen Sie Ihren Interessenbereich.
3. Wählen Sie Speichern und dann GeoJSON aus dem Dropdown-Menü aus.

GeoJSON-Geofences in eine Geofence-Sammlung einfügen

Sie können die resultierende GeoJSON-Datei verwenden, um Ihre Geofences über die Amazon Location Service-Konsole, die AWS CLI oder die Amazon Location APIs hochzuladen:

Console

So fügen Sie einer Geofence-Sammlung mithilfe der Amazon Location Service-Konsole ein Geofence hinzu

1. Öffnen Sie die Amazon Location Service-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Geofence-Sammlungen aus.
3. Wählen Sie in der Liste Geofence-Sammlungen den Namenslink für die Ziel-Geofence-Sammlung aus.
4. Wählen Sie unter Geofences die Option Geofences erstellen aus.
5. Ziehen Sie im Fenster Geofences hinzufügen Ihre GeoJSON in das Fenster.
6. Wählen Sie Geofences hinzufügen aus.

API

So fügen Sie Geofences mithilfe der Amazon Location APIs hinzu

Verwenden Sie die [PutGeofence](#) Operation aus den Amazon Location Geofences APIs .

Im folgenden Beispiel wird eine API-Anfrage verwendet, um ein Geofence mit der ID *GEOFENCE-EXAMPLE1* zu einer Geofence-Sammlung namens *ExampleGeofenceCollection* hinzuzufügen. Sie gibt auch eine einzelne Geofence-Metadateneigenschaft mit dem Schlüssel `Type` und dem Wert `loadingArea`.

```
PUT /geofencing/v0/collections/ExampleGeofenceCollection/geofence/GEOFENCE-EXAMPLE1
Content-type: application/json

{
  "GeofenceProperties": {
    "Type" : "loadingArea"
  },
  "Geometry": {
    "Polygon": [
      [
        [-5.716667, -15.933333],
        [-14.416667, -7.933333],
        [-12.316667, -37.066667],
        [-5.716667, -15.933333]
      ]
    ]
  }
}
```

Alternativ können Sie mithilfe der [BatchPutGeofence](#) Operation mehr als einen Geofence hinzufügen.

```
POST /geofencing/v0/collections/ExampleGeofenceCollection/put-geofences
Content-type: application/json

{
  "Entries": [
    {
      "GeofenceProperties": {
        "Type" : "loadingArea"
      },
      "GeofenceId": "GEOFENCE-EXAMPLE1",
      "Geometry": {
        "Polygon": [
          [
            [-5.716667, -15.933333],
            [-14.416667, -7.933333],
            [-12.316667, -37.066667],
            [-5.716667, -15.933333]
          ]
        ]
      }
    }
  ]
}
```

AWS CLI

So fügen Sie einer Geofence-Sammlung mithilfe von AWS CLI Befehlen einen Geofence hinzu

Verwenden Sie den [put-geofence](#)-Befehl.

Im folgenden Beispiel wird ein verwendet AWS CLI, um einen Geofence zu einer Geofence-Sammlung namens hinzuzufügen *ExampleGeofenceCollection*.

```
$ aws location \
  put-geofence \
    --collection-name ExampleGeofenceCollection \
    --geofence-id ExampleGeofenceTriangle \
    --geofence-properties '{"Type": "loadingArea"}' \
    --geometry 'Polygon=[[[[-5.716667, -15.933333],[ -14.416667, -7.933333],
[-12.316667, -37.066667],[ -5.716667, -15.933333]]]]'
```

```
{
  "CreateTime": "2020-11-11T00:16:14.487000+00:00",
  "GeofenceId": "ExampleGeofenceTriangle",
  "UpdateTime": "2020-11-11T00:19:59.894000+00:00"
}
```

Hinzufügen von Zirkelgeofences

In diesem Abschnitt wird das Erstellen von Zirkelgeofences beschrieben. Sie müssen den Breiten- und Längengrad des Punktes kennen, der das Zentrum des Kreises sein soll, sowie den Radius in Metern des Kreises. Sie können Zirkelgeofences mit den Amazon Location APIs oder erstellenAWS CLI.

API

So fügen Sie Zirkelgeofences mithilfe der Amazon Location APIs hinzu

Verwenden Sie die [PutGeofence](#) Operation aus den Amazon Location Geofences APIs .

Im folgenden Beispiel wird eine API-Anfrage verwendet, um ein Geofence mit der ID *GEOFENCE-EXAMPLE2* zu einer Geofence-Sammlung namens *ExampleGeofenceCollection* hinzuzufügen:

```
PUT /geofencing/v0/collections/ExampleGeofenceCollection/geofence/GEOFENCE-EXAMPLE2
Content-type: application/json

{
  "Geometry": {
    "Circle": {
      "Center": [-5.716667, -15.933333],
      "Radius": 50
    }
  }
}
```

AWS CLI

So fügen Sie einer Geofence-Sammlung mithilfe von AWS CLI Befehlen ein Zirkelgeofence hinzu

Verwenden Sie den [put-geofence](#)-Befehl.

Im folgenden Beispiel wird ein verwendet AWS CLI, um einen Geofence zu einer Geofence-Sammlung namens hinzuzufügen *ExampleGeofenceCollection*.

```
$ aws location \
  put-geofence \
    --collection-name ExampleGeofenceCollection \
    --geofence-id ExampleGeofenceCircle \
    --geometry 'Circle={Center=[-5.716667, -15.933333], Radius=50}'
```

Note

Sie können JSON für komplexe Geometrien auch in eine eigene Datei einfügen, wie im folgenden Beispiel.

```
$ aws location \
  put-geofence \
    --collection-name ExampleGeofenceCollection \
    --geofence-id ExampleGeofenceCircle \
    --geometry file:circle.json
```

In dem Beispiel enthält die Datei „circ.json“ JSON für die Kreisgeometrie.

```
{
  "Circle": {
    "Center": [-74.006975, 40.717127],
    "Radius": 287.7897969218057
  }
}
```

Starten der Nachverfolgung

Dieser Abschnitt führt Sie durch die Erstellung einer Nachverfolgungsanwendung, die Gerätestandorte erfasst.

Erstellen eines Trackers

Erstellen Sie eine Tracker-Ressource, um Positionsaktualisierungen von Ihren Geräten zu speichern und zu verarbeiten. Sie können die Amazon Location Service-Konsole, die AWS CLI oder die Amazon Location APIs verwenden.

Jede in Ihren Tracker-Ressourcen gespeicherte Positionsaktualisierung kann ein Maß für die Positionsgenauigkeit und bis zu drei Metadatenfelder über die Position oder das Gerät enthalten, die Sie speichern möchten. Die Metadaten werden als Schlüssel-Wert-Paare gespeichert und können Informationen wie Geschwindigkeit, Richtung, Temperaturdruck oder Motortemperatur speichern.

Tracker filtern Positionsaktualisierungen, sobald sie empfangen werden. Dies reduziert das visuelle Rauschen in Ihren Gerätepfaden (Jitter genannt) und reduziert die Anzahl der falschen Geofence-Ein- und -Ausgangsereignisse. Dies trägt auch zur Kostenverwaltung bei, indem die Anzahl der initiierten Geofence-Bewertungen reduziert wird.

Tracker bieten drei Optionen zur Positionsfilterung, um die Kosten zu verwalten und Jitter in Ihren Standortaktualisierungen zu reduzieren.

- Genauigkeitsbasiert – wird mit jedem Gerät verwendet, das eine Genauigkeitsmessung bietet. Die meisten Mobilgeräte stellen diese Informationen bereit. Die Genauigkeit jeder Positionsmessung wird durch viele Umgebungsfaktoren beeinflusst, darunter GPS-Satellitenempfang, Landschaft und Nähe von WLAN- und Bluetooth-Geräten. Die meisten Geräte, einschließlich der meisten Mobilgeräte, können eine Schätzung der Genauigkeit der Messung zusammen mit der Messung liefern. Beim AccuracyBased Filtern ignoriert Amazon Location Standortaktualisierungen, wenn sich das Gerät weniger als die gemessene Genauigkeit bewegt hat. Wenn beispielsweise zwei aufeinanderfolgende Updates von einem Gerät einen Genauigkeitsbereich von 5 m und 10 m haben, ignoriert Amazon Location die zweite Aktualisierung, wenn sich das Gerät weniger als 15 m bewegt hat. Amazon Location wertet weder ignorierte Updates anhand von Geofences aus noch speichert es sie.

Wenn keine Genauigkeit angegeben wird, wird sie als Null behandelt und die Messung wird als völlig genau angesehen.

Note

Sie können auch eine genauigkeitsbasierte Filterung verwenden, um die gesamte Filterung zu entfernen. Wenn Sie eine genauigkeitsbasierte Filterung auswählen, aber alle Genauigkeitsdaten auf Null überschreiben oder die Genauigkeit vollständig weglassen, filtert Amazon Location keine Aktualisierungen heraus.

- Entfernungsbasiert – Wird verwendet, wenn Ihre Geräte keine Genauigkeitsmessung bieten, Sie aber dennoch die Vorteile der Filterung nutzen möchten, um Jitter zu reduzieren und Kosten zu verwalten. -DistanceBasedFilterung ignoriert Standortaktualisierungen, bei denen sich Geräte weniger als 30 m (98,4 Fuß) bewegt haben. Wenn Sie die DistanceBased Positionsfilterung

verwenden, wertet Amazon Location diese ignorierten Updates weder anhand von Geofences aus noch speichert es die Updates.

Die Genauigkeit der meisten Mobilgeräte, einschließlich der durchschnittlichen Genauigkeit von iOS- und Android-Geräten, liegt innerhalb von 15 m. In den meisten Anwendungen kann die `DistanceBased` Filterung die Auswirkungen von Standortungenauigkeiten reduzieren, wenn Gerätekonzentrierungen auf einer Karte angezeigt werden, und die Kloneffekte mehrerer aufeinanderfolgender Ein- und Ausgangsereignisse, wenn sich Geräte nahe der Grenze eines Geofence befinden. Es kann auch dazu beitragen, die Kosten Ihrer Anwendung zu senken, indem weniger Aufrufe getätigt werden, um anhand verknüpfter Geofences zu bewerten oder Gerätepositionen abzurufen.

- **Zeitbasiert – (Standard)** Wird verwendet, wenn Ihre Geräte sehr häufig (mehr als einmal alle 30 Sekunden) Positionsaktualisierungen senden und Sie Geofence-Bewertungen nahezu in Echtzeit vornehmen möchten, ohne jede Aktualisierung zu speichern. Beim `TimeBased` Filtern wird jede Standortaktualisierung anhand verknüpfter Geofence-Sammlungen ausgewertet, aber nicht jede Standortaktualisierung wird gespeichert. Wenn Ihre Aktualisierungs-Häufigkeit häufiger als 30 Sekunden beträgt, wird nur ein Update pro 30 Sekunden für jede eindeutige Geräte-ID gespeichert.

Note

Beachten Sie bei der Entscheidung für Ihre Filtermethode die Kosten Ihrer Tracking-Anwendung und die Häufigkeit von Positionsaktualisierungen. Ihnen werden jede Standortaktualisierung und einmal die Bewertung der Positionsaktualisierung anhand jeder verknüpften Geofence-Sammlung in Rechnung gestellt. Wenn Sie beispielsweise eine zeitbasierte Filterung verwenden und Ihr Tracker mit zwei Geofence-Sammlungen verknüpft ist, zählt jede Positionsaktualisierung als eine Standortaktualisierungsanforderung und zwei Geofence-Sammlungsauswertungen. Wenn Sie alle 5 Sekunden Positionsaktualisierungen für Ihre Geräte melden und zeitbasierte Filterung verwenden, werden Ihnen 720 Standortaktualisierungen und 1 440 Geofence-Bewertungen pro Stunde für jedes Gerät in Rechnung gestellt.

Ihre Rechnung wird nicht von der Anzahl der Geofences in jeder Sammlung beeinflusst. Da jede Geofence-Sammlung bis zu 50.000 Geofences enthalten kann, sollten Sie Ihre Geofences nach Möglichkeit zu weniger Sammlungen kombinieren, um Ihre Kosten für Geofence-Bewertungen zu senken.

Standardmäßig erhalten Sie EventBridge Ereignisse jedes Mal, wenn ein verfolgtes Gerät einen verknüpften Geofence betritt oder beendet. Weitere Informationen finden Sie unter [Verknüpfen eines Trackers mit einer Geofence-Sammlung](#).

Sie können Ereignisse für alle gefilterten Positionsaktualisierungen für eine Tracker-Ressource aktivieren. Weitere Informationen finden Sie unter [Aktivieren von Aktualisierungsereignissen für einen Tracker](#).

Note

Wenn Sie Ihre Daten mit Ihrem eigenen AWS KMSkundenverwalteten Schlüssel verschlüsseln möchten, ist die Funktion Bounding Polygon Queries standardmäßig deaktiviert. Dies liegt daran, dass durch die Verwendung dieser Funktion Bounding Polygon Queries eine Darstellung Ihrer Gerätepositionen nicht mit Ihrem AWS KMS verwalteten Schlüssel verschlüsselt wird. Die genaue Geräteposition wird jedoch weiterhin mit Ihrem verwalteten Schlüssel verschlüsselt.

Sie können sich für die Funktion Bounding Polygon Queries entscheiden, indem Sie den `KmsKeyEnableGeospatialQueries` Parameter beim Erstellen oder Aktualisieren eines Trackers auf „true“ setzen.

Console

So erstellen Sie einen Tracker mit der Amazon Location-Konsole

1. Öffnen Sie die Amazon Location Service-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Tracker aus.
3. Wählen Sie Tracker erstellen aus.
4. Füllen Sie die folgenden Felder aus:
 - Name – Geben Sie einen eindeutigen Namen ein. Zum Beispiel *ExampleTracker*. Maximal 100 Zeichen. Gültige Einträge umfassen alphanumerische Zeichen, Bindestriche, Punkte und Unterstriche.
 - Beschreibung – Geben Sie eine optionale Beschreibung ein.
5. Wählen Sie unter Positionsfilterung die Option aus, die am besten zu Ihrer Tracker-Ressource passt. Wenn Sie die Positionsfilterung nicht festlegen, ist die Standardeinstellung

TimeBased. Weitere Informationen finden Sie unter [Tracker](#) in diesem Handbuch und [PositionFiltering](#) in der API-Referenz zu Amazon Location Service Trackers.

6. (Optional) Geben Sie unter Markierungen einen Tag Key (Schlüssel) und einen Value (Wert) ein. Dadurch wird Ihrer neuen Geofence-Sammlung ein Tag hinzugefügt. Weitere Informationen finden Sie unter [Markieren Ihrer Ressourcen](#).
7. (Optional) Unter Kundenverwaltete Schlüsselverschlüsselung können Sie wählen, ob Sie einen vom Kunden verwalteten Schlüssel hinzufügen möchten. Dadurch wird ein symmetrischer kundenverwalteter Schlüssel hinzugefügt, den Sie über die standardmäßige AWS-eigene Verschlüsselung erstellen, besitzen und verwalten. Weitere Informationen finden Sie unter [Verschlüsseln von Daten im Ruhezustand](#).
8. (Optional) Unter können Sie wählen `KmsKeyEnableGeospatialQueries`, ob Sie Geodatenabfragen aktivieren möchten. Auf diese Weise können Sie die Funktion Bounding Polygon Queries verwenden und gleichzeitig Ihre Daten mit einem vom Kunden AWS KMS verwalteten Schlüssel verschlüsseln.

 Note

Wenn Sie die Funktion Bounding Polygon Queries verwenden, wird eine Darstellung Ihrer Gerätepositionen nicht mit Ihrem AWS KMS verwalteten Schlüssel verschlüsselt. Die genaue Geräteposition wird jedoch weiterhin mit Ihrem verwalteten Schlüssel verschlüsselt.

9. (Optional) Unter EventBridge Konfiguration können Sie Ereignisse für gefilterte EventBridge Positionsaktualisierungen aktivieren. Dadurch wird jedes Mal ein Ereignis gesendet, wenn eine Positionsaktualisierung für ein Gerät in diesem Tracker der Bewertung der Positionsfilterung entspricht.
10. Wählen Sie Tracker erstellen aus.

API

So erstellen Sie einen Tracker mithilfe der Amazon Location APIs

Verwenden Sie die [CreateTracker](#) Operation aus den APIs von Amazon Location Trackers APIs.

Im folgenden Beispiel wird eine API-Anfrage verwendet, um einen Tracker namens zu erstellen *ExampleTracker*. Die Tracker-Ressource ist einem [vom Kunden verwalteten](#)

[AWS KMS Schlüssel zugeordnet, um Kundendaten zu verschlüsseln, und aktiviert keine Positionsaktualisierungen in 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"
  }
}
```

Erstellen eines Trackers mit **KmsKeyEnableGeospatialQueries** aktivierter

Im folgenden Beispiel ist der Parameter auf „true“ **KmsKeyEnableGeospatialQueries** gesetzt. Auf diese Weise können Sie die Funktion Bounding Polygon Queries verwenden und gleichzeitig Ihre Daten mit einem vom Kunden AWS KMS verwalteten Schlüssel verschlüsseln.

Informationen zur Verwendung der Funktion Bounding Polygon Queries finden Sie unter . [???](#)

Note

Wenn Sie die Funktion Bounding Polygon Queries verwenden, wird eine Darstellung Ihrer Gerätepositionen nicht mit Ihrem AWS KMS verwalteten Schlüssel verschlüsselt. Die genaue Geräteposition wird jedoch weiterhin mit Ihrem verwalteten Schlüssel verschlüsselt.

```
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

So erstellen Sie einen Tracker mithilfe von AWS CLI Befehlen

Verwenden Sie den [create-tracker](#)-Befehl.

Im folgenden Beispiel wird verwendet AWS CLI, um einen Tracker namens zu erstellen *ExampleTracker*. Die Tracker-Ressource ist einem [vom Kunden verwalteten AWS KMS Schlüssel zugeordnet, um Kundendaten zu verschlüsseln](#), und [aktiviert keine Positionsaktualisierungen in 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"
```

Erstellen eines Trackers mit **KmsKeyEnableGeospatialQueries** aktivierter

Im folgenden Beispiel ist der Parameter auf „true“ **KmsKeyEnableGeospatialQueries** gesetzt. Auf diese Weise können Sie die Funktion Bounding Polygon Queries verwenden und gleichzeitig Ihre Daten mit einem vom Kunden AWS KMS verwalteten Schlüssel verschlüsseln.

Informationen zur Verwendung der Funktion Bounding Polygon Queries finden Sie unter [. ???](#)

Note

Wenn Sie die Funktion Bounding Polygon Queries verwenden, wird eine Darstellung Ihrer Gerätepositionen nicht mit Ihrem AWS KMS verwalteten Schlüssel verschlüsselt.

Die genaue Geräteposition wird jedoch weiterhin mit Ihrem verwalteten Schlüssel verschlüsselt.

```
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

Die Fakturierung hängt von Ihrer Nutzung ab. Möglicherweise fallen Gebühren für die Nutzung anderer -AWS-Services an. Weitere Informationen finden Sie unter [Amazon Location Service – Preise](#).

Sie können die EventBridge Konfiguration Beschreibung , Positionsfilterung und bearbeiten, nachdem der Tracker erstellt wurde, indem Sie Tracker bearbeiten auswählen.

Authentifizieren Ihrer Anforderungen

Sobald Sie eine Tracker-Ressource erstellt haben und bereit sind, Gerätepositionen anhand von Geofences auszuwerten, wählen Sie aus, wie Sie Ihre Anfragen authentifizieren würden:

- Informationen dazu, wie Sie auf die Services zugreifen können, finden Sie unter [Zugriff auf Amazon Location Service](#).
- Wenn Sie Gerätepositionen mit nicht authentifizierten Anforderungen veröffentlichen möchten, sollten Sie Amazon Cognito verwenden.

Beispiel

Das folgende Beispiel zeigt die Verwendung eines Amazon Cognito-Identitäten-Pools für die Autorisierung mit [AWS JavaScript SDK v3](#) und dem Amazon Location [Authentifizierungshelfer](#).

```
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);
```

Aktualisieren Ihres Trackers mit einer Geräteposition

Um Ihre Geräte zu verfolgen, können Sie Updates zur Geräteposition an Ihren Tracker senden. Sie können diese Gerätepositionen oder den Gerätepositionsverlauf später von Ihrer Tracker-Ressource abrufen.

Jede Positionsaktualisierung muss die Geräte-ID, einen Zeitstempel und eine Position enthalten. Sie können optional andere Metadaten einschließen, einschließlich Genauigkeit und bis zu 3 Schlüssel-Wert-Paare für Ihren eigenen Gebrauch.

Wenn Ihr Tracker mit einer oder mehreren Geofence-Sammlungen verknüpft ist, werden Aktualisierungen anhand dieser Geofences ausgewertet (gemäß den Filterregeln, die Sie für den Tracker angegeben haben). Wenn ein Gerät einen Geofenced-Bereich überschreitet (indem es von innerhalb des Bereichs nach außen oder umgekehrt wechselt), erhalten Sie Ereignisse in EventBridge. Zu diesen - ENTER oder -EXIT Ereignissen gehören die Details zur Positionsaktualisierung, einschließlich der Geräte-ID, des Zeitstempels und der zugehörigen Metadaten.

Note

Weitere Informationen zur Positionsfilterung finden Sie unter [Erstellen eines Trackers](#). Weitere Informationen zu Geofence-Ereignissen finden Sie unter [Reagieren auf Amazon Location Service-Ereignisse mit Amazon EventBridge](#).

Verwenden Sie eine der folgenden Methoden, um Geräteaktualisierungen zu senden:

- [Senden Sie MQTT-Updates](#) an eine AWS IoT Core-Ressource und verknüpfen Sie sie mit Ihrer Tracker-Ressource.
- Senden Sie Standortaktualisierungen mithilfe der Amazon Location Trackers API, mithilfe der AWS CLI oder der Amazon Location APIs. Sie können die [AWS -SDKs](#) verwenden, um die APIs von Ihrer iOS- oder Android-Anwendung aus aufzurufen.

API

So senden Sie eine Positionsaktualisierung mit den Amazon Location APIs

Verwenden Sie die [BatchUpdateDevicePosition](#) Operation aus den Amazon Location Trackers APIs.

Im folgenden Beispiel wird eine API-Anfrage verwendet, um eine Aktualisierung der Geräteposition für an einen Tracker *ExampleDevice* zu senden *ExampleTracker*.

```
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

So senden Sie eine Positionsaktualisierung mithilfe von AWS CLI Befehlen

Verwenden Sie den [batch-update-device-position](#)-Befehl.

Im folgenden Beispiel wird ein verwendet AWS CLI, um eine Aktualisierung der Geräteposition für *ExampleDevice-1* und *ExampleDevice-2* an einen Tracker zu senden *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"}}]'
```

Abrufen des Standortverlaufs eines Geräts von einem Tracker

Ihre Amazon Location Tracker-Ressource speichert den Standortverlauf all Ihrer verfolgten Geräte für einen Zeitraum von 30 Tagen. Sie können den Gerätestandortverlauf, einschließlich aller zugehörigen Metadaten, aus Ihrer Tracker-Ressource abrufen. In den folgenden Beispielen wird die AWS CLI oder die Amazon Location APIs verwendet.

API

So rufen Sie den Gerätestandortverlauf von einem Tracker mithilfe der Amazon Location APIs ab

Verwenden Sie die [GetDevicePositionHistory](#) Operation aus den Amazon Location Trackers APIs .

Im folgenden Beispiel wird eine API-URI-Anfrage verwendet, um den Gerätestandortverlauf von *ExampleDevice* von einem Tracker mit dem Namen abzurufen, der *ExampleTracker* von 19:05:07 (inklusive) beginnt und auf 19:20:07 (exklusiv) endet 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

So rufen Sie den Gerätestandortverlauf von einem Tracker mithilfe von AWS CLI Befehlen ab

Verwenden Sie den [get-device-position-history](#)-Befehl.

Im folgenden Beispiel wird ein verwendet AWS CLI, um den Gerätestandortverlauf von *ExampleDevice* einem Tracker namens abzurufen, *ExampleTracker* der von 19:05:07 (inklusive) aus beginnt und bei 19:20:07 (exklusiv) auf endet 2020-10-02.

```
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"
```

Auflisten Ihrer Gerätepositionen

Sie können eine Liste der Gerätepositionen für einen Tracker mithilfe der AWS CLI oder der Amazon Location APIs mit der ListDevicePositions API anzeigen. Wenn Sie die ListDevicePositions API aufrufen, wird eine Liste der neuesten Positionen für alle Geräte zurückgegeben, die

einem bestimmten Tracker zugeordnet sind. Standardmäßig gibt diese API 100 der neuesten Gerätepositionen pro Ergebnisseite für einen bestimmten Tracker zurück. Um nur Geräte innerhalb einer bestimmten Region zurückzugeben, verwenden Sie den `FilterGeometry` Parameter, um eine Bounding Polygon Query zu erstellen. Wenn Sie aufrufen `ListDevicePositions`, werden nur Geräte innerhalb des Polygons zurückgegeben.

Note

Wenn Sie Ihre Daten mit Ihrem eigenen AWS KMSkundenverwalteten Schlüssel verschlüsseln möchten, ist die Funktion Bounding Polygon Queries standardmäßig deaktiviert. Dies liegt daran, dass durch die Verwendung dieser Funktion eine Darstellung Ihrer Gerätepositionen nicht mit dem von Ihnen AWS KMS verwalteten Schlüssel verschlüsselt wird. Die genaue Geräteposition ist jedoch weiterhin mit Ihrem verwalteten Schlüssel verschlüsselt.

Sie können sich für die Funktion Bounding Polygon Queries entscheiden. Dies geschieht, indem der `KmsKeyEnableGeospatialQueries` Parameter beim Erstellen oder Aktualisieren eines Trackers auf „true“ gesetzt wird.

API

Verwenden Sie die [-ListDevicePositions](#) Operation aus den Amazon Location Trackers APIs .

Das folgende Beispiel ist eine API-Anforderung zum Abrufen einer Liste von Gerätepositionen im polygonalen Bereich unter Verwendung des optionalen Parameters [FilterGeometry](#). Das Beispiel gibt 3 Gerätestandorte zurück, die in dem durch das `PolygonArray` definierten Bereich vorhanden sind.

```
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"
}

```

Im Folgenden finden Sie eine Beispielantwort für [ListDevicePositions](#):

```

{
  "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

Verwenden Sie den [list-trackers](#)-Befehl.

Das folgende Beispiel ist ein AWS CLI, um eine Liste von Geräten in einem polygonalen Bereich abzurufen.

```
aws location list-device-positions TODO: add arguments add props for filter geo
```

Verknüpfen eines Trackers mit einer Geofence-Sammlung

Nachdem Sie nun eine Geofence-Sammlung und einen Tracker haben, können Sie sie miteinander verknüpfen, sodass Standortaktualisierungen automatisch anhand all Ihrer Geofences ausgewertet werden. Wenn Sie nicht alle Standortaktualisierungen auswerten möchten oder alternativ einige Ihrer Standorte nicht in einer Tracker-Ressource speichern, können Sie die [Gerätepositionen bei Bedarf anhand von Geofences auswerten](#).

Wenn Gerätepositionen anhand von Geofences ausgewertet werden, werden Ereignisse generiert. Sie können eine Aktion für diese Ereignisse festlegen. Weitere Informationen zu Aktionen, die Sie für

Geofence-Ereignisse festlegen können, finden Sie unter [Reagieren auf Amazon Location Service-Ereignisse mit Amazon EventBridge](#).

Ein Amazon Location-Ereignis enthält die Attribute der Gerätepositionsaktualisierung, die es generiert, und einige Attribute des eingegebenen oder beendeten Geofences. Weitere Informationen zu den Daten, die in einem Geofence-Ereignis enthalten sind, finden Sie unter [Amazon- EventBridge Ereignisbeispiele für Amazon Location Service](#).

Die folgenden Beispiele verknüpfen eine Tracker-Ressource mit einer Geofence-Sammlung mithilfe der Konsole AWS CLI, der oder der Amazon Location APIs .

Console

So verknüpfen Sie eine Tracker-Ressource mit einer Geofence-Sammlung mithilfe der Amazon Location Service-Konsole

1. Öffnen Sie die Amazon Location Service-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Tracker aus.
3. Wählen Sie unter Geräte-Tracker den Namenslink des Ziel-Trackers aus.
4. Wählen Sie unter Verknüpfte Geofence-Sammlungen die Option Geofence-Sammlung verknüpfen aus.
5. Wählen Sie im Fenster Verknüpfte Geofence-Sammlung eine Geofence-Sammlung aus dem Dropdown-Menü aus.
6. Wählen Sie Verknüpfen.

Nachdem Sie die Tracker-Ressource verknüpft haben, wird ihr ein Aktiv-Status zugewiesen.

API

So verknüpfen Sie eine Tracker-Ressource mit einer Geofence-Sammlung mithilfe der Amazon Location APIs

Verwenden Sie die [AssociateTrackerConsumer](#) Operation aus den Amazon Location Trackers APIs .

Im folgenden Beispiel wird eine API-Anfrage verwendet, die mithilfe ihres [Amazon-Ressourcennamens](#) (ARN) *ExampleTracker* mit einer Geofence-Sammlung verknüpft.

```
POST /tracking/v0/trackers/ExampleTracker/consumers
```

```
Content-type: application/json

{
  "ConsumerArn": "arn:aws:geo:us-west-2:123456789012:geofence-
collection/ExampleGeofenceCollection"
}
```

AWS CLI

So verknüpfen Sie eine Tracker-Ressource mit einer Geofence-Sammlung mithilfe von AWS CLI Befehlen

Verwenden Sie den [associate-tracker-consumer](#)-Befehl.

Im folgenden Beispiel wird ein verwendet AWS CLI, um eine Geofence-Sammlung namens zu erstellen *ExampleGeofenceCollection*.

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

Gerätepositionen anhand von Geofences bewerten

Es gibt zwei Möglichkeiten, Positionen anhand von Geofences zu bewerten, um Geofence-Ereignisse zu generieren:

- Sie können Tracker und Geofence-Sammlungen verknüpfen. Weitere Informationen finden Sie im Abschnitt: [Verknüpfen eines Trackers mit einer Geofence-Sammlung](#).
- Sie können eine direkte Anforderung an die Geofence-Sammlungsressource stellen, um eine oder mehrere Positionen auszuwerten.

Wenn Sie auch den Standort Ihres Geräts verfolgen oder Standorte auf einer Karte anzeigen möchten, verknüpfen Sie den Tracker mit einer Geofence-Sammlung. Alternativ möchten Sie möglicherweise nicht alle Standortaktualisierungen auswerten oder Sie beabsichtigen nicht, Standortdaten in einer Tracker-Ressource zu speichern. Wenn einer dieser Fälle zutrifft, können Sie eine direkte Anfrage an die Geofence-Sammlung stellen und eine oder mehrere Gerätepositionen anhand ihrer Geofences auswerten.

Die Bewertung von Gerätepositionen anhand von Geofences generiert Ereignisse. Sie können auf diese Ereignisse reagieren und sie an andere -AWS-Services weiterleiten. Weitere Informationen zu Aktionen, die Sie beim Empfang von Geofence-Ereignissen ausführen können, finden Sie unter [Reagieren auf Amazon Location Service-Ereignisse mit Amazon EventBridge](#).

Ein Amazon Location-Ereignis enthält die Attribute der Gerätepositionsaktualisierung, die es generiert, einschließlich Zeit-, Positions-, Genauigkeits- und Schlüsselwert-Metadaten sowie einige Attribute des eingegebenen oder beendeten Geofence. Weitere Informationen zu den Daten, die in einem Geofence-Ereignis enthalten sind, finden Sie unter [Amazon- EventBridge Ereignisbeispiele für Amazon Location Service](#).

In den folgenden Beispielen wird die AWS CLI oder die Amazon Location APIs verwendet.

API

So bewerten Sie die Gerätepositionen anhand der Position von Geofences mithilfe der Amazon Location APIs

Verwenden Sie die [BatchEvaluateGeofences](#)-Operation aus den Amazon Location Geofences APIs .

Im folgenden Beispiel wird eine API-Anfrage verwendet, um die Position des Geräts in einer zugeordneten Geofence-Sammlung *ExampleDevice* auszuwerten *ExampleGeofenceCollection*. Ersetzen Sie diese Werte durch Ihre eigenen Geofence- und Geräte-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

So bewerten Sie die Gerätepositionen anhand der Position von Geofences mithilfe von AWS CLI Befehlen

Verwenden Sie den [batch-evaluate-geofences](#)-Befehl.

Im folgenden Beispiel wird ein verwendet AWS CLI, um die Position von *ExampleDevice* anhand einer zugeordneten Geofence-Sammlung zu bewerten *ExampleGeofenceCollection*. Ersetzen Sie diese Werte durch Ihre eigenen Geofence- und Geräte-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"} } ] '
```

Reagieren auf Amazon Location Service-Ereignisse mit Amazon EventBridge

Amazon EventBridge ist ein Serverless Event Bus, der Anwendungen mithilfe von Daten aus - AWS Services wie Amazon Location. EventBridge receives Events von Amazon Location effizient miteinander verbindet und diese Daten an Ziele wie weiterleitet AWS Lambda. Sie können Routing-Regeln einrichten, um zu bestimmen, wohin Ihre Daten gesendet werden sollen, um Anwendungsarchitekturen zu erstellen, die in Echtzeit reagieren.

Standardmäßig werden nur Geofence-Ereignisse (ENTER- und -EXIT Ereignisse, wenn Geräte in die gefensterten Bereiche gelangen oder diese verlassen) an gesendet EventBridge . Sie können auch alle gefilterten Positionsaktualisierungsereignisse für eine Tracker-Ressource aktivieren. Weitere Informationen finden Sie unter [Aktivieren von Aktualisierungsereignissen für einen Tracker](#).

Weitere Informationen finden Sie [unter Ereignisse und Ereignismuster](#) im Amazon- EventBridge Benutzerhandbuch.

Themen

- [Aktivieren von Aktualisierungsereignissen für einen Tracker](#)
- [Erstellen von Ereignisregeln für Amazon Location](#)
- [Amazon- EventBridge Ereignisbeispiele für Amazon Location Service](#)

Aktivieren von Aktualisierungsereignissen für einen Tracker

Standardmäßig sendet Amazon Location nur - ENTER und EXIT Geofence-Ereignisse an EventBridge. Sie können aktivieren, dass alle gefilterten UPDATE Positionereignisse für einen Tracker an gesendet werden EventBridge. Sie können dies tun, wenn Sie einen Tracker [erstellen](#) oder [aktualisieren](#).

Um beispielsweise einen vorhandenen Tracker mithilfe der zu aktualisierenAWS CLI, können Sie den folgenden Befehl verwenden (verwenden Sie den Namen Ihrer Tracker-Ressource anstelle von *MyTracker*).

```
aws location update-tracker --tracker-name MyTracker --event-bridge-enabled
```

Um Positionereignisse für einen Tracker zu deaktivieren, müssen Sie die -API oder die Amazon Location Service-Konsole verwenden.

Erstellen von Ereignisregeln für Amazon Location

Sie können [bis zu 300 Regeln pro Event Bus](#) in erstellen EventBridge , um Aktionen zu konfigurieren, die als Reaktion auf ein Amazon Location-Ereignis ausgeführt werden.

Sie können beispielsweise eine Regel für Geofence-Ereignisse erstellen, bei denen eine Push-Benachrichtigung gesendet wird, wenn ein Telefon innerhalb einer Geofenced-Grenze erkannt wird.

So erstellen Sie eine Regel für Amazon-Location-Ereignisse

Erstellen Sie mit den folgenden Werten [eine - EventBridge Regel](#) basierend auf Amazon Location-Ereignissen:

- Bei Rule type (Regeltyp) wählen Sie Rule with an event pattern (Regel mit einem Ereignismuster) aus.
- Fügen Sie im Feld Ereignismuster das folgende Muster hinzu:

```
{  
  "source": ["aws.geo"],
```

```
"detail-type": ["Location Geofence Event"]
}
```

Um eine Regel für Tracker-Positionsaktualisierungen zu erstellen, können Sie stattdessen das folgende Muster verwenden:

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Device Position Event"]
}
```

Sie können optional nur - ENTER oder -EXIT Ereignisse angeben, indem Sie ein `detail` Tag hinzufügen (wenn Ihre Regel für Tracker-Positionsaktualisierungen bestimmt ist, gibt es nur ein einziges `EventType`, sodass Sie es nicht filtern müssen):

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Geofence Event"],
  "detail": {
    "EventType": ["ENTER"]
  }
}
```

Sie können optional auch nach Eigenschaften der Position oder des Geofence filtern:

```
{
  "source": ["aws.geo"],
  "detail-type": ["Location Geofence Event"],
  "detail": {
    "EventType": ["ENTER"],
    "GeofenceProperties": {
      "Type": "LoadingDock"
    },
    "PositionProperties": {
      "VehicleType": "Truck"
    }
  }
}
```

- Wählen Sie unter Ziele auswählen die Zielaktion aus, die ausgeführt werden soll, wenn ein Ereignis vom Amazon Location Service empfangen wird.

Verwenden Sie beispielsweise ein Amazon Simple Notification Service (SNS)-Thema, um eine E-Mail oder Textnachricht zu senden, wenn ein Ereignis eintritt. Sie müssen zuerst mit der Amazon-SNS-Konsole ein Amazon-SNS-Thema erstellen. Weitere Informationen finden Sie unter [Verwenden von Amazon SNS für Benutzerbenachrichtigungen](#).

Warning

Es hat sich bewährt zu bestätigen, dass die Ereignisregel erfolgreich angewendet wurde oder Ihre automatisierte Aktion möglicherweise nicht wie erwartet ausgelöst wird. Um Ihre Ereignisregel zu überprüfen, initiieren Sie Bedingungen für die Ereignisregel. Simulieren Sie beispielsweise ein Gerät, das in einen Geofenced-Bereich gelangt.

Sie können auch alle Ereignisse von Amazon Location erfassen, indem Sie einfach den `detail-type` Abschnitt ausschließen. Beispielsweise:

```
{
  "source": [
    "aws.geo"
  ]
}
```

Note

Das gleiche Ereignis kann mehr als einmal zugestellt werden. Sie können die Ereignis-ID verwenden, um die Ereignisse, die Sie erhalten, zu deduplizieren.

Amazon- EventBridge Ereignisbeispiele für Amazon Location Service

Im Folgenden finden Sie ein Beispiel für ein Ereignis für die Eingabe eines Geofence, das durch Aufrufen von `initiiert wirdBatchUpdateDevicePosition`.

```
{
  "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"
  }
}
}

```

Im Folgenden finden Sie ein Beispiel für ein Ereignis zum Beenden eines Geofence, das durch Aufrufen von `initiiert` wird `BatchUpdateDevicePosition`.

```

{
  "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"
    }
  }
}

```

Im Folgenden finden Sie ein Beispiel für ein Ereignis für eine Positionsaktualisierung, das durch Aufrufen von `initiatedBatchUpdateDevicePosition` wird.

```

{
  "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"
}
}
```

Nachverfolgung mit AWS IoT und MQTT mit Amazon Location Service

[MQTT](#) ist ein leichtgewichtiges und weit verbreitetes Messaging-Protokoll, das für eingeschränkte Geräte entwickelt wurde. AWS IoT Core unterstützt Geräteverbindungen, die das MQTT-Protokoll und das MQTT over WebSocket Secure (WSS)-Protokoll verwenden.

[AWS IoT Core](#) verbindet Geräte mit AWS und ermöglicht es Ihnen, Nachrichten zwischen ihnen zu senden und zu empfangen. Die AWS IoT Core Regel-Engine speichert Abfragen zu den Nachrichtenthemen Ihrer Geräte und ermöglicht es Ihnen, Aktionen zum Senden von Nachrichten an andere -AWS-Services wie Amazon Location Service zu definieren. Geräte, die ihren Standort als Koordinaten kennen, können ihre Standorte über die Regel-Engine an Amazon Location weiterleiten.

Note

Geräte können ihre eigene Position kennen, z. B. über integriertes GPS. unterstützt AWS IoT auch die Standortverfolgung von Geräten von Drittanbietern. Weitere Informationen finden Sie unter [AWS IoT Core Device Location](#) im AWS IoT Core-Entwicklerhandbuch.

In der folgenden Anleitung wird die Nachverfolgung mithilfe von -AWS IoT Core-Regeln beschrieben. Sie können die Geräteinformationen auch an Ihre eigene AWS Lambda Funktion senden, wenn Sie sie vor dem Senden an Amazon Location verarbeiten müssen. Weitere Informationen zur

Verwendung von Lambda zur Verarbeitung Ihrer Gerätestandorte finden Sie unter [Verwenden von AWS Lambda mit MQTT](#).

Themen

- [Voraussetzung](#)
- [Erstellen einer AWS IoT Core-Regel](#)
- [Testen Ihrer AWS IoT Core Regel in der Konsole](#)
- [Verwenden von AWS Lambda mit MQTT](#)

Voraussetzung

Bevor Sie mit der Nachverfolgung beginnen können, müssen Sie die folgenden Voraussetzungen erfüllen:

- [Erstellen Sie eine Tracker-Ressource](#), an die Sie die Gerätestandortdaten senden.
- [Erstellen Sie eine IAM-Rolle](#), um AWS IoT Core Zugriff auf Ihren Tracker zu gewähren.

Wenn Sie diese Schritte ausführen, verwenden Sie die folgende Richtlinie, um Zugriff auf Ihren Tracker zu gewähren:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "WriteDevicePosition",
      "Effect": "Allow",
      "Action": "geo:BatchUpdateDevicePosition",
      "Resource": "arn:aws:geo:*:*:tracker/*"
    }
  ]
}
```

Erstellen einer AWS IoT Core-Regel

Erstellen Sie als Nächstes eine -AWS IoT Core-Regel, um die Positionstelemetrie Ihrer Geräte an Amazon Location Service weiterzuleiten. Weitere Informationen zum Erstellen von Regeln finden Sie in den folgenden Themen im AWS IoT Core Entwicklerhandbuch für :

- [Erstellen einer -AWS IoT Regel](#) für Informationen zum Erstellen einer neuen Regel.
- [Standortaktion](#) für Informationen zum Erstellen einer Regel für die Veröffentlichung in Amazon Location

Testen Ihrer AWS IoT Core Regel in der Konsole

Wenn derzeit keine Geräte Telemetrie veröffentlichen, die den Standort enthält, können Sie Ihre Regel mit der AWS IoT Core Konsole testen. Die Konsole verfügt über einen Testclient, über den Sie eine Beispielnachricht veröffentlichen können, um die Ergebnisse der Lösung zu überprüfen.

1. Melden Sie sich bei der -AWS IoT Core Konsole unter <https://console.aws.amazon.com/iot/> an.
2. Erweitern Sie in der linken Navigation Test und wählen Sie MQTT-Testclient aus.
3. Legen Sie unter In einem Thema veröffentlichen den Themennamen auf *iot/topic* (oder den Namen des Themas, das Sie in Ihrer AWS IoT Core Regel eingerichtet haben, falls unterschiedlich) fest und geben Sie Folgendes für die Nachrichtennutzlast an. Ersetzen Sie den Zeitstempel *1604940328* durch einen gültigen Zeitstempel innerhalb der letzten 30 Tage (alle Zeitstempel, die älter als 30 Tage sind, werden von Amazon Location Service-Trackern ignoriert).

```
{
  "payload": {
    "deviceid": "thing123",
    "timestamp": 1604940328,
    "location": { "lat": 49.2819, "long": -123.1187 },
    "accuracy": { "Horizontal": 20.5 },
    "positionProperties": { "field1": "value1", "field2": "value2" }
  }
}
```

4. Wählen Sie Im Thema veröffentlichen, um die Testnachricht zu senden.
5. Verwenden Sie den folgenden AWS CLI Befehl, um zu überprüfen, ob die Nachricht vom Amazon Location Service empfangen wurde. Wenn Sie ihn während der Einrichtung geändert haben, ersetzen Sie den Tracker-Namen durch den von Ihnen verwendeten.

```
aws location batch-get-device-position --tracker-name MyTracker --device-ids
thing123
```

Verwenden von AWS Lambda mit MQTT

Während die Verwendung von nicht mehr erforderlich AWS Lambda ist, wenn Sie Gerätestandortdaten zur Nachverfolgung an Amazon Location senden, sollten Sie in einigen Fällen trotzdem Lambda verwenden. Wenn Sie beispielsweise Ihre Gerätestandortdaten selbst verarbeiten möchten, bevor Sie sie an Amazon Location senden. In den folgenden Themen wird beschrieben, wie Sie mit Lambda Nachrichten verarbeiten, bevor Sie sie an Ihren Tracker senden. Weitere Informationen zu diesem Muster finden Sie in der [Referenzarchitektur](#).

Themen

- [Voraussetzung](#)
- [Erstellen einer Lambda-Funktion](#)
- [Erstellen einer AWS IoT Core-Regel](#)
- [Testen Ihrer AWS IoT Core Regel in der Konsole](#)

Voraussetzung

Bevor Sie mit der Nachverfolgung beginnen können, müssen Sie [eine Tracker-Ressource erstellen](#). Um eine Tracker-Ressource zu erstellen, können Sie die Amazon Location-Konsole, die AWS CLI oder die Amazon Location APIs verwenden.

Im folgenden Beispiel wird die Amazon Location Service-Konsole verwendet, um die Tracker-Ressource zu erstellen:

1. Öffnen Sie die Amazon Location Service-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Tracker aus.
3. Wählen Sie Tracker erstellen aus.
4. Füllen Sie die folgenden Felder aus:
 - Name – Geben Sie einen eindeutigen Namen mit maximal 100 Zeichen ein. Gültige Einträge sind alphanumerische Zeichen, Bindestriche und Unterstriche. Zum Beispiel *MyTracker*.
 - Beschreibung – Geben Sie eine optionale Beschreibung ein. Zum Beispiel *Tracker zum Speichern von AWS IoT Core Gerätepositionen*.
 - Positionsfilterung – Wählen Sie die Filterung aus, die Sie für Positionsaktualisierungen verwenden möchten. Zum Beispiel Genauigkeitsbasierte Filterung.

5. Wählen Sie Tracker erstellen aus.

Erstellen einer Lambda-Funktion

Um eine Verbindung zwischen AWS IoT Core und Amazon Location Service herzustellen, benötigen Sie eine AWS Lambda-Funktion, um Nachrichten zu verarbeiten, die von weitergeleitet werden AWS IoT Core. Diese Funktion extrahiert alle Positionsdaten, formatiert sie für Amazon Location Service und sendet sie über die Amazon Location Tracker API. Sie können diese Funktion über die AWS Lambda Konsole erstellen oder die AWS Command Line Interface (AWS CLI) oder die AWS Lambda APIs verwenden.

So erstellen Sie eine Lambda-Funktion, die Positionsaktualisierungen in Amazon Location über die Konsole veröffentlicht:

1. Öffnen Sie die AWS Lambda-Konsole unter <https://console.aws.amazon.com/lambda/>.
2. Wählen Sie in der linken Navigation Funktionen aus.
3. Wählen Sie Funktion erstellen und stellen Sie sicher, dass Von Grund auf neu erstellen ausgewählt ist.
4. Füllen Sie die folgenden Felder aus:
 - Funktionsname – Geben Sie einen eindeutigen Namen für Ihre Funktion ein. Gültige Einträge umfassen alphanumerische Zeichen, Bindestriche und Unterstriche ohne Leerzeichen. Zum Beispiel *MyLambda*.
 - Laufzeit – Wählen Sie *Python 3.8* aus.
5. Wählen Sie Funktion erstellen.
6. Wählen Sie die Registerkarte Code, um den Editor zu öffnen.
7. Überschreiben Sie den Platzhaltercode in `lambda_function.py` mit dem folgenden und ersetzen Sie den Wert, der dem zugewiesen ist, durch `TRACKER_NAME` den Namen des Trackers, den Sie als [Voraussetzung](#) erstellt haben.

```
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. Wählen Sie Bereitstellen, um die aktualisierte Funktion zu speichern.
9. Wählen Sie die Registerkarte Konfiguration aus.
10. Wählen Sie im Abschnitt Berechtigungen den Hyperlink Rollennamen aus, um Amazon Location Service Berechtigungen für Ihre Lambda-Funktion zu erteilen.
11. Wählen Sie auf der Übersichtsseite Ihrer Rolle die Option Berechtigungen hinzufügen und dann aus der Dropdown-Liste die Option Inline-Richtlinie erstellen aus.
12. Wählen Sie die Registerkarte JSON und überschreiben Sie die Richtlinie mit dem folgenden Dokument. Auf diese Weise kann Ihre Lambda-Funktion Gerätepositionen aktualisieren, die von allen Tracker-Ressourcen in allen Regionen verwaltet werden.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "WriteDevicePosition",
      "Effect": "Allow",
      "Action": "geo:BatchUpdateDevicePosition",
      "Resource": "arn:aws:geo:*:*:tracker/*"
    }
  ]
}
```

13. Wählen Sie Richtlinie prüfen.
14. Geben Sie den Namen einer Richtlinie ein. Zum Beispiel *AmazonLocationTrackerWriteOnly*.
15. Wählen Sie Richtlinie erstellen aus.

Sie können diesen Funktionscode nach Bedarf ändern, um ihn an Ihr eigenes Gerätenachrichtenschema anzupassen.

Erstellen einer AWS IoT Core-Regel

Erstellen Sie als Nächstes eine AWS IoT Core Regel, um die Positionstelemetrie Ihrer Geräte zur Transformation und Veröffentlichung an Amazon Location Service an die AWS Lambda Funktion weiterzuleiten. Die bereitgestellte Beispielregel geht davon aus, dass jede erforderliche Transformation von Gerätenutzlasten von Ihrer Lambda-Funktion durchgeführt wird. Sie können diese Regel über die AWS IoT CoreKonsole, die AWS Command Line Interface (AWS CLI) oder die AWS IoT Core APIs erstellen.

Note

Während die AWS IoT Konsole die Berechtigung übernimmt AWS IoT Core, die zum Aufrufen Ihrer Lambda-Funktion erforderlich ist, müssen Sie eine Richtlinie konfigurieren AWS CLI, um die Berechtigung zu erteilen. [AWS IoT](#)

So erstellen Sie eine AWS IoT Core mithilfe der Konsole

1. Melden Sie sich bei der -AWS IoT CoreKonsole unter <https://console.aws.amazon.com/iot/> an.
2. Erweitern Sie in der linken Navigation die Option Act und wählen Sie Rules aus.
3. Wählen Sie Regel erstellen, um den neuen Regelassistenten zu starten.
4. Geben Sie für Ihre Regel einen Namen und eine Beschreibung ein.
5. Aktualisieren Sie für die Regelabfrageanweisung das Attribut so, dass es auf ein Thema verweist, in dem mindestens ein Gerät Telemetrie veröffentlicht, die den Standort enthält. FROM Wenn Sie die Lösung testen, sind keine Änderungen erforderlich.

```
SELECT * FROM 'iot/topic'
```

6. Wählen Sie unter Eine oder mehrere Aktionen festlegen die Option Aktion hinzufügen aus.
7. Wählen Sie Nachricht an eine Lambda-Funktion senden aus.
8. Wählen Sie Configure action.
9. Suchen und wählen Sie Ihre Lambda-Funktion aus der Liste aus.
10. Wählen Sie Aktion hinzufügen aus.
11. Wählen Sie Regel erstellen aus.

Testen Ihrer AWS IoT Core Regel in der Konsole

Wenn derzeit keine Geräte Telemetrie veröffentlichen, die den Standort enthält, können Sie Ihre Regel und diese Lösung mithilfe der AWS IoT Core Konsole testen. Die Konsole verfügt über einen Testclient, über den Sie eine Beispielnachricht veröffentlichen können, um die Ergebnisse der Lösung zu überprüfen.

1. Melden Sie sich bei der -AWS IoT CoreKonsole unter <https://console.aws.amazon.com/iot/> an.
2. Erweitern Sie in der linken Navigation Test und wählen Sie MQTT-Testclient aus.
3. Legen Sie unter In einem Thema veröffentlichen den Themennamen auf *iot/topic* (oder den Namen des Themas, das Sie in Ihrer AWS IoT Core Regel eingerichtet haben, falls unterschiedlich) fest und geben Sie Folgendes für die Nachrichtennutzlast an. Ersetzen Sie den Zeitstempel *1604940328* durch einen gültigen Zeitstempel innerhalb der letzten 30 Tage (alle Zeitstempel, die älter als 30 Tage sind, werden ignoriert).

```
{
  "payload": {
    "deviceid": "thing123",
    "timestamp": 1604940328,
    "location": { "lat": 49.2819, "long": -123.1187 },
    "accuracy": { "Horizontal": 20.5 },
    "positionProperties": { "field1": "value1", "field2": "value2" }
  }
}
```

4. Wählen Sie Im Thema veröffentlichen, um die Testnachricht zu senden.
5. Verwenden Sie den folgenden AWS CLI Befehl, um zu überprüfen, ob die Nachricht vom Amazon Location Service empfangen wurde. Wenn Sie sie während der Einrichtung geändert haben, ersetzen Sie den Tracker-Namen und die Geräte-ID durch die von Ihnen verwendeten.

```
aws location batch-get-device-position --tracker-name MyTracker --device-ids
thing123
```

Verwalten Ihrer Geofence-Sammlungsressourcen

Verwalten Sie Ihre Geofence-Sammlungen mithilfe der Amazon Location-KonsoleAWS CLI, der oder der Amazon Location APIs.

Auflisten Ihrer Geofence-Sammlungsressourcen

Sie können Ihre Geofence-Sammlungsliste über die Amazon-Standortkonsole, die AWS CLI oder die Amazon-Standort-APIs anzeigen:

Console

So zeigen Sie eine Liste von Geofence-Sammlungen mit der Amazon Location-Konsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Geofence-Sammlungen aus.
3. Zeigen Sie eine Liste Ihrer Geofence-Sammlungen unter Meine Geofence-Sammlungen an.

API

Verwenden Sie die [ListGeofenceCollections](#) Operation aus den Amazon Location Geofences APIs .

Das folgende Beispiel ist eine API-Anforderung zum Abrufen einer Liste von Geofence-Sammlungen im AWS Konto.

```
POST /geofencing/v0/list-collections
```

Im Folgenden finden Sie eine Beispielantwort für ListGeofenceCollections:

```
{
  "Entries": [
    {
      "CollectionName": "ExampleCollection",
      "CreateTime": 2020-09-30T22:59:34.142Z,
      "Description": "string",
      "UpdateTime": 2020-09-30T23:59:34.142Z
    },
    "NextToken": "1234-5678-9012"
  ]
}
```

CLI

Verwenden Sie den [list-geofence-collections](#)-Befehl.

Das folgende Beispiel ist ein AWS CLI, um eine Liste von Geofence-Sammlungen im AWS Konto abzurufen.

```
aws location list-geofence-collections
```

Details zur Geofence-Sammlung abrufen

Sie können Details zu jeder Geofence-Sammlungsressource in Ihrem AWS Konto über die Amazon Location-Konsole, die AWS CLI oder die Amazon Location APIs abrufen:

Console

So zeigen Sie die Details einer Geofence-Sammlung mit der Amazon Location-Konsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Geofence-Sammlungen aus.
3. Wählen Sie unter Meine Geofence-Sammlungen den Namenslink der Ziel-Geofence-Sammlung aus.

API

Verwenden Sie die [DescribeGeofenceCollection](#) Operation aus den Amazon Location Geofences APIs .

Das folgende Beispiel ist eine API-Anforderung zum Abrufen der Details der Geofence-Sammlung für *ExampleCollection*.

```
GET /geofencing/v0/collections/ExampleCollection
```

Im Folgenden finden Sie eine Beispielantwort für DescribeGeofenceCollection:

```
{
  "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

Verwenden Sie den [describe-geofence-collection](#)-Befehl.

Das folgende Beispiel ist ein AWS CLI, um die Details der Geofence-Sammlung für *ExampleCollection* abzurufen.

```
aws location describe-geofence-collection \
  --collection-name "ExampleCollection"
```

Löschen einer Geofence-Sammlung

Sie können eine Geofence-Sammlung aus Ihrem AWS Konto mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs löschen.

Console

So löschen Sie eine Geofence-Sammlung mit der Amazon Location-Konsole

 **Warning**
Dieser Vorgang löscht die Ressource dauerhaft.

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Geofence-Sammlungen aus.
3. Wählen Sie unter Meine Geofence-Sammlung die Ziel-Geofence-Sammlung aus.
4. Wählen Sie Geofence-Sammlung löschen aus.

API

Verwenden Sie die [DeleteGeofenceCollection](#) Operation aus den Amazon Location APIs .

Das folgende Beispiel ist eine API-Anforderung zum Löschen der Geofence-Sammlung *ExampleCollection*.

```
DELETE /geofencing/v0/collections/ExampleCollection
```

Im Folgenden finden Sie eine Beispielantwort für DeleteGeofenceCollection:

```
HTTP/1.1 200
```

CLI

Verwenden Sie den [delete-geofence-collection](#)-Befehl.

Das folgende Beispiel ist ein -AWS CLIBefehl zum Löschen der Geofence-Sammlung *ExampleCollection*.

```
aws location delete-geofence-collection \  
  --collection-name "ExampleCollection"
```

Gespeicherte Geofences auflisten

Sie können Geofences auflisten, die in einer bestimmten Geofence-Sammlung gespeichert sind, indem Sie die Amazon Location-Konsole, die AWS CLI, oder die Amazon Location APIs verwenden.

Console

So zeigen Sie eine Liste von Geofences mit der Amazon Location-Konsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Geofence-Sammlungen aus.
3. Wählen Sie unter Meine Geofence-Sammlung den Namenslink der Ziel-Geofence-Sammlung aus.
4. Geofences in der Geofence-Sammlung unter Geofences anzeigen

API

Verwenden Sie die [ListGeofences](#) Operation aus den Amazon Location Geofences APIs .

Das folgende Beispiel ist eine API-Anforderung zum Abrufen einer Liste von Geofences, die in der Geofence-Sammlung gespeichert sind *ExampleCollection*.

```
POST /geofencing/v0/collections/ExampleCollection/list-geofences
```

Im Folgenden finden Sie eine Beispielantwort für ListGeofences:

```
{
  "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

Verwenden Sie den [list-geofences](#)-Befehl.

Das folgende Beispiel ist ein AWS CLI, um eine Liste von Geofences abzurufen, die in der Geofence-Sammlung gespeichert sind *ExampleCollection*.

```
aws location list-geofences \
  --collection-name "ExampleCollection"
```

Geofence-Details abrufen

Sie können die Details eines bestimmten Geofence, z. B. Erstellungszeit, Aktualisierungszeit, Geometrie und Status, aus einer Geofence-Sammlung mithilfe der Amazon Location-Konsole, AWS CLI oder der Amazon Location APIs abrufen.

Console

So zeigen Sie den Status eines Geofence mit der Amazon Location-Konsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Geofence-Sammlungen aus.
3. Wählen Sie unter Meine Geofence-Sammlung den Namenslink der Ziel-Geofence-Sammlung aus.
4. Unter Geofences können Sie den Status Ihrer Geofences anzeigen.

API

Verwenden Sie die `-GetGeofence` Operation aus den Amazon Location Geofences APIs .

Das folgende Beispiel ist eine API-Anforderung zum Abrufen der Geofence-Details aus einer Geofence-Sammlung *ExampleCollection*.

```
GET /geofencing/v0/collections/ExampleCollection/geofences/ExampleGeofence1
```

Im Folgenden finden Sie eine Beispielantwort für `GetGeofence`:

```
{
  "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

Verwenden Sie den [get-geofence](#)-Befehl.

Das folgende Beispiel ist ein AWS CLI, um die Details der Geofence-Sammlung für *ExampleCollection* abzurufen.

```
aws location get-geofence \
  --collection-name "ExampleCollection" \
  --geofence-id "ExampleGeofence1"
```

Geofences löschen

Sie können Geofences aus einer Geofence-Sammlung mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs löschen.

Console

So löschen Sie ein Geofence mithilfe der Amazon Location-Konsole

Warning

Dieser Vorgang löscht die Ressource dauerhaft.

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Geofence-Sammlungen aus.
3. Wählen Sie unter Meine Geofence-Sammlung den Namenslink der Ziel-Geofence-Sammlung aus.
4. Wählen Sie unter Geofences den Ziel-Geofence aus.
5. Wählen Sie Geofence löschen aus.

API

Verwenden Sie die [BatchDeleteGeofence](#) Operation aus den Amazon Location Geofences APIs .

Das folgende Beispiel ist eine API-Anforderung zum Löschen von Geofences aus der Geofence-Sammlung *ExampleCollection*.

```
POST /geofencing/v0/collections/ExampleCollection/delete-geofences
Content-type: application/json

{
  "GeofenceIds": [ "ExampleGeofence11" ]
}
```

Im Folgenden finden Sie ein Beispiel für eine Erfolgsantwort für [BatchDeleteGeofence](#).

```
HTTP/1.1 200
```

CLI

Verwenden Sie den [batch-delete-geofence](#)-Befehl.

Das folgende Beispiel ist ein -AWS CLIBefehl zum Löschen von Geofences aus der Geofence-Sammlung *ExampleCollection*.

```
aws location batch-delete-geofence \
  --collection-name "ExampleCollection" \
  --geofence-ids "ExampleGeofence11"
```

Verwalten Ihrer Tracker-Ressourcen

Sie können Ihre Tracker mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs verwalten.

Auflisten Ihrer Tracker

Sie können Ihre Tracker-Liste über die Amazon-Standortkonsole, die AWS CLI oder die Amazon-Standort-APIs anzeigen:

Console

So zeigen Sie eine Liste der vorhandenen Tracker mithilfe der Amazon Location-Konsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.

2. Wählen Sie in der linken Navigation Tracker aus.
3. Zeigen Sie eine Liste Ihrer Tracker-Ressourcen unter Meine Tracker an.

API

Verwenden Sie die [ListTrackers](#) Operation aus den Amazon Location Trackers APIs .

Das folgende Beispiel ist eine API-Anfrage zum Abrufen einer Liste von Trackern in Ihrem AWS Konto.

```
POST /tracking/v0/list-trackers
```

Im Folgenden finden Sie eine Beispielantwort für [ListTrackers](#):

```
{
  "Entries": [
    {
      "CreateTime": 2020-10-02T19:09:07.327Z,
      "Description": "string",
      "TrackerName": "ExampleTracker",
      "UpdateTime": 2020-10-02T19:10:07.327Z
    }
  ],
  "NextToken": "1234-5678-9012"
}
```

CLI

Verwenden Sie den [list-trackers](#)-Befehl.

Das folgende Beispiel ist ein AWS CLI, um eine Liste von Trackern in Ihrem AWS Konto abzurufen.

```
aws location list-trackers
```

Trennen eines Trackers von einer Geofence-Sammlung

Sie können einen Tracker von einer Geofence-Sammlung trennen, indem Sie die Amazon Location-Konsole AWS CLI, die oder die Amazon Location APIs verwenden:

Console

So heben Sie die Zuordnung eines Trackers zu einer zugeordneten Geofence-Sammlung mithilfe der Amazon-Standortkonsole auf

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Tracker aus.
3. Wählen Sie unter Meine Tracker den Namenslink des Ziel-Trackers aus.
4. Wählen Sie unter Verknüpfte Geofence-Sammlungen eine Geofence-Sammlung mit dem Status Verknüpft aus.
5. Wählen Sie Verknüpfung aufheben aus.

API

Verwenden Sie die [-DisassociateTrackerConsumer](#) Operation aus den Amazon Location Trackers APIs .

Das folgende Beispiel ist eine API-Anforderung zum Trennen eines Trackers von einer zugeordneten Geofence-Sammlung.

```
DELETE /tracking/v0/trackers/ExampleTracker/consumers/arn:aws:geo:us-west-2:123456789012:geofence-collection/ExampleCollection
```

Im Folgenden finden Sie eine Beispielantwort für [DisassociateTrackerConsumer](#):

```
HTTP/1.1 200
```

CLI

Verwenden Sie den [disassociate-tracker-consumer](#)-Befehl.

Das folgende Beispiel ist ein -AWS CLIBefehl, um die Zuordnung eines Trackers zu einer zugeordneten Geofence-Sammlung aufzuheben.

```
aws location disassociate-tracker-consumer \  
  --consumer-arn "arn:aws:geo:us-west-2:123456789012:geofence-collection/  
  ExampleCollection" \  
  --
```

```
--tracker-name "ExampleTracker"
```

Abrufen von Tracker-Details

Sie können Details zu jedem Tracker in Ihrem AWS Konto mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs abrufen.

Console

So zeigen Sie Tracker-Details mithilfe der Amazon Location-Konsole an

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie in der linken Navigation Tracker aus.
3. Wählen Sie unter Meine Tracker den Namenslink des Ziel-Trackers aus.
4. Zeigen Sie die Tracker-Details unter Informationen an.

API

Verwenden Sie die [DescribeTracker](#) Operation aus den Amazon Location Tracker-APIs.

Das folgende Beispiel ist eine API-Anforderung zum Abrufen der Tracker-Details für *ExampleTracker*.

```
GET /tracking/v0/trackers/ExampleTracker
```

Im Folgenden finden Sie eine Beispielantwort für [DescribeTracker](#):

```
{
  "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

Verwenden Sie den [describe-tracker](#)-Befehl.

Das folgende Beispiel ist ein -AWS CLIBefehl zum Abrufen von Tracker-Details für *ExampleTracker*.

```
aws location describe-tracker \  
  --tracker-name "ExampleTracker"
```

Löschen eines Trackers

Sie können einen Tracker aus Ihrem AWS Konto mithilfe der Amazon Location-Konsole, der AWS CLloder der Amazon Location APIs löschen:

Console

So löschen Sie eine vorhandene Kartenressource mithilfe der Amazon Location-Konsole

Warning

Dieser Vorgang löscht die Ressource dauerhaft. Wenn die Tracker-Ressource verwendet wird, kann ein Fehler auftreten. Stellen Sie sicher, dass die Zielressource keine Abhängigkeit für Ihre Anwendungen ist.

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich Tracker aus.
3. Wählen Sie unter Meine Tracker den Ziel-Tracker aus.
4. Wählen Sie Tracker löschen aus.

API

Verwenden Sie die [DeleteTracker](#)Operation aus den Amazon Location Tracker-APIs .

Das folgende Beispiel ist eine API-Anforderung zum Löschen des Trackers *ExampleTracker*.

```
DELETE /tracking/v0/trackers/ExampleTracker
```

Im Folgenden finden Sie eine Beispielantwort für [DeleteTracker](#):

```
HTTP/1.1 200
```

CLI

Verwenden Sie den [delete-tracker](#)-Befehl.

Das folgende Beispiel ist ein -AWS CLIBefehl zum Löschen des Trackers *ExampleTracker*.

```
aws location delete-tracker \  
  --tracker-name "ExampleTracker"
```

Markieren Ihrer Amazon Location Service-Ressourcen

Verwenden Sie die Ressourcenmarkierung in Amazon Location, um Tags zu erstellen, um Ihre Ressourcen nach Zweck, Eigentümer, Umgebung oder Kriterien zu kategorisieren. Das Markieren Ihrer Ressourcen hilft Ihnen, Ihre Ressourcen zu verwalten, zu identifizieren, zu organisieren, zu suchen und zu filtern.

Mit können Sie beispielsweise Gruppen von AWS Ressourcen erstellen AWS Resource Groups, die auf einem oder mehreren Tags oder Teilen von Tags basieren. Sie können auch Gruppen auf der Grundlage ihres Vorkommens in einem AWS CloudFormation-Batch erstellen. Mit Ressourcengruppen und dem Tag-Editor können Sie Daten für Anwendungen konsolidieren und anzeigen, die aus mehreren Services, Ressourcen und Regionen bestehen. Weitere Informationen zu [allgemeinen Tagging-Strategien](#) finden Sie in der Allgemeinen AWS-Referenz.

Jedes Tag ist eine Bezeichnung, die aus einem Schlüssel und einem Wert besteht, den Sie definieren:

- Tag-Schlüssel – Eine allgemeine Bezeichnung, die die Tag-Werte kategorisiert. Beispiel: `CostCenter`
- Tag-Wert – Eine optionale Beschreibung für die Tag-Schlüsselkategorie. Beispiel: `MobileAssetTrackingResourcesProd`

Dieses Thema hilft Ihnen beim Einstieg in das Tagging, indem Sie die Tagging-Einschränkungen überprüfen. Es zeigt Ihnen auch, wie Sie Tags erstellen und Tags verwenden, um Ihre AWS Kosten für jedes aktive Tag mithilfe von Kostenzuordnungsberichten zu verfolgen.

Themen

- [Tagging-Einschränkungen](#)
- [Gewährt die Berechtigung zum Markieren von Ressourcen](#)
- [Hinzufügen eines Tags zu einer Amazon Location Service-Ressource](#)
- [Verfolgen der Ressourcenkosten nach Tag](#)
- [Steuern des Zugriffs auf Amazon Location Service-Ressourcen mithilfe von Tags](#)
- [Weitere Informationen](#)

Tagging-Einschränkungen

Die folgenden grundlegenden Einschränkungen gelten für Tags (Markierungen):

- Maximale Tags pro Ressource – 50
- Jeder Tag (Markierung) muss für jede Ressource eindeutig sein. Jeder Tag (Markierung) kann nur einen Wert haben.

Note

Wenn Sie einen neuen Tag (Markierung) mit demselben Tag (Markierung)-Schlüssel wie ein bestehender Tag (Markierung) hinzufügen, wird der bestehende Tag (Markierung) vom neuen überschrieben.

- Maximale Schlüssellänge: 128 Unicode-Zeichen in UTF-8
- Maximale Wertlänge: 256 Unicode-Zeichen in UTF-8
- Erlaubte Zeichen in Services sind: Buchstaben, Zahlen und Leerzeichen, die in UTF-8 darstellbar sind, und die folgenden Sonderzeichen: + - = . _ : / @.
- Bei Tag (Markierung)-Schlüsseln und -Werten muss die Groß-/Kleinschreibung beachtet werden.
- Das Präfix `aws :` ist zur Verwendung in AWS reserviert. Wenn der Tag (Markierung) über einen Tag (Markierung)-Schlüssel mit diesem Präfix verfügt, können Sie den Schlüssel oder Wert des Tags (Markierung) nicht bearbeiten oder löschen. Tags mit dem `aws :` Präfix werden nicht auf Ihre Tags pro Ressourcenlimit angerechnet.

Gewährt die Berechtigung zum Markieren von Ressourcen

Sie können IAM-Richtlinien verwenden, um den Zugriff auf Ihre Amazon Location-Ressourcen zu steuern und die Berechtigung zum Markieren einer Ressource bei der Erstellung zu erteilen. Die Richtlinie kann nicht nur die Berechtigung zum Erstellen von Ressourcen erteilen, sondern auch Action Berechtigungen zum Zulassen von Tagging-Operationen enthalten:

- `geo:TagResource` – Ermöglicht es einem Benutzer, einer angegebenen Amazon Location-Ressource ein oder mehrere Tags zuzuweisen.
- `geo:UntagResource` – Ermöglicht es einem Benutzer, ein oder mehrere Tags aus einer angegebenen Amazon Location-Ressource zu entfernen.
- `geo:ListTagsForResource` – Ermöglicht einem Benutzer das Auflisten aller Tags, die einer Amazon Location-Ressource zugewiesen sind.

Im Folgenden finden Sie ein Richtlinienbeispiel, mit dem ein Benutzer eine Geofence-Sammlung erstellen und Ressourcen markieren kann:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowTaggingForGeofenceCollectionOnCreation",
      "Effect": "Allow",
      "Action": [
        "geo:CreateGeofenceCollection",
        "geo:TagResource"
      ],
      "Resource": "arn:aws:geo:region:accountID:geofence-collection/*"
    }
  ]
}
```

Hinzufügen eines Tags zu einer Amazon Location Service-Ressource

Sie können Tags hinzufügen, wenn Sie Ihre Ressourcen mithilfe der Amazon Location-Konsole, der AWS CLI oder der Amazon Location APIs erstellen:

- [Erstellen einer Kartenressource](#)
- [Erstellen einer Ortsindex-Ressource](#)

- [Erstellen einer Routenrechner-Ressource](#)
- [Erstellen einer Geofence-Sammlung](#)
- [Erstellen einer Tracker-Ressource](#)

Um vorhandene Ressourcen zu markieren, bearbeiten oder löschen Sie Tags

1. Öffnen Sie die Amazon Location-Konsole unter <https://console.aws.amazon.com/location/>.
2. Wählen Sie im linken Navigationsbereich die Ressource aus, die Sie markieren möchten. Zum Beispiel Maps .
3. Wählen Sie eine Ressource aus der Liste aus.
4. Wählen Sie Tags verwalten, um Ihre Tags hinzuzufügen, zu bearbeiten oder zu löschen.

Verfolgen der Ressourcenkosten nach Tag

Sie können Tags für die Kostenzuordnung verwenden, um Ihre AWS Kosten im Detail zu verfolgen. Nachdem Sie die Kostenzuordnungs-Tags aktiviert haben, AWS verwendet die Kostenzuordnungs-Tags, um Ihre Ressourcenfakturierung in Ihrem Kostenzuordnungsbericht zu organisieren. Auf diese Weise können Sie Ihre Nutzungskosten kategorisieren und verfolgen.

Es gibt zwei Arten von Kostenzuordnungs-Tags, die Sie aktivieren können:

- [AWS-generiert](#) – Diese Tags werden von generiertAWS. AWS -Tags verwenden das `aws :` Präfix , z. B. `aws :createdBy`.
- [Benutzerdefiniert](#) – Dies sind benutzerdefinierte Tags, die Sie erstellen. Die benutzerdefinierten Tags verwenden das `user :` Präfix , z. B. `user :CostCenter`.

Sie müssen jeden Tag-Typ einzeln aktivieren. Nachdem Ihre Tags aktiviert wurden, können Sie Ihren monatlichen Kostenzuordnungsbericht [aktivieren AWS Cost Explorer](#) oder anzeigen.

AWS-generated tags

So aktivieren Sie von AWS generierte Tags

1. Öffnen Sie die Konsole für Fakturierung und Kostenmanagement unter <https://console.aws.amazon.com/billing/>.
2. Wählen Sie im linken Navigationsbereich Kostenzuordnungs-Tags aus.

3. Wählen Sie auf der Registerkarte Von AWS generierte Kostenzuordnungs-Tags die Tag-Schlüssel aus, die Sie aktivieren möchten.
4. Wählen Sie Activate.

User-defined tags

So aktivieren Sie benutzerdefinierte Tags

1. Öffnen Sie die Konsole für Fakturierung und Kostenmanagement unter <https://console.aws.amazon.com/billing/>.
2. Wählen Sie im linken Navigationsbereich Kostenzuordnungs-Tags aus.
3. Wählen Sie auf der Registerkarte Benutzerdefinierte Kostenzuordnungs-Tags die Tag-Schlüssel aus, die Sie aktivieren möchten.
4. Wählen Sie Activate.

Nachdem Sie Ihre Tags aktiviert haben, AWS generiert einen [monatlichen Kostenzuordnungsbericht](#) für Ihre Ressourcennutzung und -kosten. Dieser Kostenzuordnungsbericht enthält alle Ihre AWS Kosten für jeden Abrechnungszeitraum, einschließlich markierter und nicht markierter Ressourcen. Weitere Informationen finden Sie unter [Verwendung von Kostenzuordnungs-Tags](#) im AWS Billing and Cost Management Leitfadens.

Steuern des Zugriffs auf Amazon Location Service-Ressourcen mithilfe von Tags

AWS Identity and Access Management (IAM)-Richtlinien unterstützen Tag-basierte Bedingungen, mit denen Sie die Autorisierung für Ihre Ressourcen basierend auf bestimmten Tag-Schlüsseln und -Werten verwalten können. Beispielsweise kann eine IAM-Rollenrichtlinie Bedingungen enthalten, um den Zugriff auf bestimmte Umgebungen, wie Entwicklung, Test oder Produktion, basierend auf Tags zu beschränken.

Weitere Informationen finden Sie im Thema zum [Steuern des Ressourcenzugriffs basierend auf Tags](#).

Weitere Informationen

Weitere Informationen über:

- Bewährte Methoden für die Markierung finden Sie unter [Markieren von AWS-Ressourcen](#) in der Allgemeinen AWS-Referenz.
- Informationen zum Verwenden von Tags zum Steuern des Zugriffs auf -AWSRessourcen finden Sie unter [Steuern des Zugriffs auf -AWSRessourcen mithilfe von Tags](#) im AWS Identity and Access Management -Benutzerhandbuch.

Gewähren des Zugriffs auf Amazon Location Service

Um Amazon Location Service verwenden zu können, muss einem Benutzer Zugriff auf die Ressourcen und APIs gewährt werden, aus denen Amazon Location besteht. Es gibt drei Strategien, mit denen Sie Zugriff auf Ihre -Ressourcen gewähren können.

- Verwenden von IAM – Um Benutzern Zugriff zu gewähren, die mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) authentifiziert sind, erstellen Sie eine IAM-Richtlinie, die den Zugriff auf die gewünschten Ressourcen ermöglicht. Weitere Informationen zu IAM und Amazon Location finden Sie unter [Identity and Access Management für Amazon Location Service](#).
- API-Schlüssel verwenden – Um nicht authentifizierten Benutzern Zugriff zu gewähren, können Sie API-Schlüssel erstellen, die schreibgeschützten Zugriff auf Ihre Amazon Location Service-Ressourcen gewähren. Dies ist in Fällen nützlich, in denen Sie nicht jeden Benutzer authentifizieren möchten. Zum Beispiel eine Webanwendung. Weitere Informationen zu API-Schlüsseln finden Sie unter [Erlauben des nicht authentifizierten Gastzugriffs auf Ihre Anwendung mithilfe von API-Schlüsseln](#).
- Amazon Cognito verwenden – Eine Alternative zu API-Schlüsseln ist die Verwendung von Amazon Cognito, um anonymen Zugriff zu gewähren. Mit Amazon Cognito können Sie eine umfassendere Autorisierung mit einer Richtlinie erstellen, um zu definieren, was von nicht authentifizierten Benutzern getan werden kann. Weitere Informationen zur Verwendung von Amazon Cognito finden Sie unter [Erlauben des nicht authentifizierten Gastzugriffs auf Ihre Anwendung mit Amazon Cognito](#).

Note

Sie können Amazon Cognito auch verwenden, um Ihren eigenen Authentifizierungsprozess zu verwenden oder mehrere Authentifizierungsmethoden mithilfe von Amazon Cognito Federated Identities zu kombinieren. Weitere Informationen finden Sie unter [Erste Schritte mit Verbundidentitäten](#) im Amazon Cognito-Entwicklerhandbuch.

Themen

- [Erlauben des nicht authentifizierten Gastzugriffs auf Ihre Anwendung mithilfe von API-Schlüsseln](#)
- [Erlauben des nicht authentifizierten Gastzugriffs auf Ihre Anwendung mit Amazon Cognito](#)

Erlauben des nicht authentifizierten Gastzugriffs auf Ihre Anwendung mithilfe von API-Schlüsseln

Wenn Sie Amazon Location Service APIs in Ihren Anwendungen aufrufen, führen Sie diesen Aufruf normalerweise als authentifizierter Benutzer aus, der zum Ausführen der API-Aufrufe autorisiert ist. Es gibt jedoch einige Fälle, in denen Sie nicht jeden Benutzer Ihrer Anwendung authentifizieren möchten. Beispielsweise möchten Sie vielleicht, dass eine Webanwendung, die Ihren Standort anzeigt, jedem, der die Website verwendet, zur Verfügung steht, unabhängig davon, ob er angemeldet ist oder nicht. In diesem Fall besteht eine Alternative darin, API-Schlüssel für die API-Aufrufe zu verwenden.

API-Schlüssel sind ein Schlüsselwert, der bestimmten Amazon Location Service-Ressourcen in Ihrem zugeordnet ist AWS-Konto, sowie bestimmten Aktionen, die Sie für diese Ressourcen ausführen können. Sie können einen API-Schlüssel in Ihrer Anwendung verwenden, um nicht authentifizierte Aufrufe an die Amazon Location APIs für diese Ressourcen zu tätigen. Wenn Sie beispielsweise einen API-Schlüssel mit der Kartenressource myMap und den GetMap* Aktionen verknüpfen, kann eine Anwendung, die diesen API-Schlüssel verwendet, Karten anzeigen, die mit dieser Ressource erstellt wurden, und Ihr Konto wird als jede andere Nutzung von Ihrem Konto belastet. Derselbe API-Schlüssel würde keine Berechtigungen zum Ändern oder Aktualisieren der Kartenressource erteilen – nur die Verwendung der Ressource ist zulässig.

Note

API-Schlüssel können nur mit Karten-, Orts- und Routenressourcen verwendet werden, und Sie können diese Ressourcen nicht ändern oder erstellen. Wenn Ihre Anwendung Zugriff auf andere Ressourcen oder Aktionen für nicht authentifizierte Benutzer benötigt, können Sie Amazon Cognito verwenden, um Zugriff zusammen mit oder anstelle von API-Schlüsseln zu gewähren. Weitere Informationen finden Sie unter [Erlauben des nicht authentifizierten Gastzugriffs auf Ihre Anwendung mit Amazon Cognito](#).

API-Schlüssel enthalten einen Klartextwert, der Zugriff auf eine oder mehrere Ressourcen in Ihrem gewährt AWS-Konto. Wenn jemand Ihren API-Schlüssel kopiert, kann er auf dieselben Ressourcen

zugreifen. Um dies zu vermeiden, können Sie die Domänen angeben, in denen der API-Schlüssel beim Erstellen des Schlüssels verwendet werden kann. Diese Domains werden als Referenten bezeichnet. Bei Bedarf können Sie auch kurzfristige API-Schlüssel erstellen, indem Sie Ablaufzeiten für Ihre API-Schlüssel festlegen.

Themen

- [API-Schlüssel im Vergleich zu Amazon Cognito](#)
- [Erstellen von API-Schlüsseln](#)
- [Verwenden eines API-Schlüssels zum Aufrufen einer Amazon Location API](#)
- [Verwenden eines API-Schlüssels zum Rendern einer Zuordnung](#)
- [Verwalten der Lebensdauer von API-Schlüsseln](#)

API-Schlüssel im Vergleich zu Amazon Cognito

API-Schlüssel und Amazon Cognito werden auf ähnliche Weise für ähnliche Szenarien verwendet. Warum würden Sie also einen gegenüber dem anderen verwenden? In der folgenden Liste werden einige der Unterschiede zwischen den beiden hervorgehoben.

- API-Schlüssel sind nur für Karten-, Orts- und Routenressourcen sowie nur für bestimmte Aktionen verfügbar. Amazon Cognito kann verwendet werden, um den Zugriff auf die meisten Amazon Location Service APIs zu authentifizieren.
- Die Leistung von Kartenanforderungen mit API-Schlüsseln ist in der Regel schneller als ähnliche Szenarien mit Amazon Cognito. Eine einfachere Authentifizierung bedeutet weniger Roundtrips zum Service und zwischengespeicherte Anforderungen, wenn innerhalb eines kurzen Zeitraums wieder dieselbe Kartenkachel abgerufen wird.
- Mit Amazon Cognito können Sie Ihren eigenen Authentifizierungsprozess verwenden oder mehrere Authentifizierungsmethoden kombinieren, indem Sie Amazon Cognito Federated Identities verwenden. Weitere Informationen finden Sie unter [Erste Schritte mit Verbundidentitäten](#) im Amazon Cognito-Entwicklerhandbuch.

Erstellen von API-Schlüsseln

Sie können einen API-Schlüssel erstellen und ihn einer oder mehreren Ressourcen in Ihrem zugeordneten AWS-Konto zuordnen.

Sie können einen API-Schlüssel mithilfe der Amazon Location Service-Konsole, der AWS CLI oder der Amazon Location APIs erstellen.

Console

So erstellen Sie einen API-Schlüssel mit der Amazon Location Service-Konsole

1. Wählen Sie in der [Amazon Location-Konsole](#) im linken Menü API-Schlüssel aus.
2. Wählen Sie auf der Seite API-Schlüssel die Option API-Schlüssel erstellen aus.
3. Geben Sie auf der Seite API-Schlüssel erstellen die folgenden Informationen ein:
 - Name – Ein Name für Ihren API-Schlüssel, z. B. MyWebAppKey.
 - Beschreibung – Eine optionale Beschreibung für Ihren API-Schlüssel.
 - Ressourcen – Wählen Sie die Amazon Location-Ressourcen aus, auf die Sie mit diesem API-Schlüssel Zugriff gewähren möchten, aus der Dropdown-Liste aus. Sie können mehr als eine Ressource hinzufügen, indem Sie Ressource hinzufügen auswählen.
 - Aktionen – Geben Sie die Aktionen an, die Sie mit diesem API-Schlüssel autorisieren möchten. Sie müssen mindestens eine Aktion auswählen, die jedem ausgewählten Ressourcentyp entspricht. Wenn Sie beispielsweise eine Ortsressource ausgewählt haben, müssen Sie mindestens eine der Optionen unter Ortsaktionen auswählen.
 - Ablaufzeit – Fügen Sie optional ein Ablaufdatum und eine Ablaufzeit für Ihren API-Schlüssel hinzu. Weitere Informationen finden Sie unter [Verwalten der Lebensdauer von API-Schlüsseln](#).
 - Referenten – Fügen Sie optional eine oder mehrere Domains hinzu, in denen Sie den API-Schlüssel verwenden können. Wenn der API-Schlüssel beispielsweise eine Anwendung zulassen soll, die auf der Website ausgeführt wird `example.com`, können Sie `*.example.com/` als zulässigen Referer verwenden.
 - Tags – Fügen Sie dem API-Schlüssel optional Tags hinzu.
4. Wählen Sie API-Schlüssel erstellen, um den API-Schlüssel zu erstellen.
5. Auf der Detailseite für den API-Schlüssel finden Sie Informationen zu dem von Ihnen erstellten API-Schlüssel. Wählen Sie API-Schlüssel anzeigen, um den Schlüsselwert anzuzeigen, den Sie beim Aufrufen von Amazon Location APIs verwenden. Der Schlüsselwert hat das Format `v1.public.a1b2c3d4...`. Weitere Informationen zur Verwendung des API-Schlüssels zum Rendern von Karten finden Sie unter [Verwenden eines API-Schlüssels zum Rendern einer Zuordnung](#).

API

So erstellen Sie einen API-Schlüssel mit den Amazon Location APIs

Verwenden Sie die [-CreateKey](#) Operation aus den Amazon Location APIs .

Das folgende Beispiel ist eine API-Anforderung zum Erstellen eines API-Schlüssels *ExampleKey* namens ohne Ablaufdatum und Zugriff auf eine einzelne Kartenressource.

```
POST /metadata/v0/keys HTTP/1.1
Content-type: application/json

{
  "KeyName": "ExampleKey"
  "Restrictions": {
    "AllowActions": [
      "geo:GetMap*"
    ],
    "AllowResources": [
      "arn:aws:geo:region:map/mapname"
    ]
  },
  "NoExpiry": true
}
```

Die Antwort enthält den API-Schlüsselwert, der beim Zugriff auf Ressourcen in Ihren Anwendungen verwendet werden soll. Der Schlüsselwert hat das Format `v1.public.a1b2c3d4...`. Weitere Informationen zur Verwendung des API-Schlüssels zum Rendern von Karten finden Sie unter [Verwenden eines API-Schlüssels zum Rendern einer Zuordnung](#).

Sie können auch die [DescribeKey](#)-API verwenden, um den Schlüsselwert für einen Schlüssel zu einem späteren Zeitpunkt zu finden.

AWS CLI

So erstellen Sie einen API-Schlüssel mit AWS CLI -Befehlen

Verwenden Sie den [create-key](#)-Befehl.

Im folgenden Beispiel wird ein API-Schlüssel namens *ExampleKey* ohne Ablaufdatum und Zugriff auf eine einzelne Kartenressource erstellt.

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

Die Antwort enthält den API-Schlüsselwert, der beim Zugriff auf Ressourcen in Ihren Anwendungen verwendet werden soll. Der Schlüsselwert hat das Format `v1.public.a1b2c3d4....`. Weitere Informationen zur Verwendung des API-Schlüssels zum Rendern von Karten finden Sie unter [Verwenden eines API-Schlüssels zum Rendern einer Zuordnung](#). Die Antwort auf `create-key` sieht wie folgt aus.

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

Sie können auch `describe-key` verwenden, um den Schlüsselwert zu einem späteren Zeitpunkt zu finden. Das folgende Beispiel zeigt, wie Sie `describe-key` für einen API-Schlüssel mit dem Namen `ExampleKey` aufrufen.

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

Verwenden eines API-Schlüssels zum Aufrufen einer Amazon Location API

Nachdem Sie einen API-Schlüssel erstellt haben, können Sie den Schlüsselwert verwenden, um Aufrufe an Amazon Location APIs in Ihrer Anwendung zu tätigen.

Die APIs, die API-Schlüssel unterstützen, verfügen über einen zusätzlichen Parameter, der den API-Schlüsselwert verwendet. Wenn Sie beispielsweise die `GetPlace`-API aufrufen, können Sie den [Schlüsselparameter](#) wie folgt ausfüllen:

```
GET /places/v0/indexes/IndexName/places/PlaceId?key=KeyValue
```

Wenn Sie diesen Wert eingeben, müssen Sie den API-Aufruf nicht wie gewohnt mit AWS Sig v4 authentifizieren.

Für JavaScript Entwickler können Sie Amazon Location verwenden, [Authentifizierungshelfer](#) um die Authentifizierung von API-Operationen mit API-Schlüsseln zu unterstützen.

Für Entwickler von Mobilgeräten können Sie die folgenden Amazon Location SDKs für die mobile Authentifizierung verwenden:

- [Amazon Location Service Mobile Authentication SDK für iOS](#)
- [Amazon Location Service Mobile Authentication SDK für Android](#)

Wenn Sie den `---key` Parameter verwenden, sollten Sie für AWS CLI Benutzer auch den `---no-sign-request` Parameter verwenden, um das Signieren mit Sig v4 zu vermeiden.

Note

Wenn Sie sowohl eine `key` als auch eine AWS Sig v4-Signatur in einen Aufruf an Amazon Location Service aufnehmen, wird nur der API-Schlüssel verwendet.

Verwenden eines API-Schlüssels zum Rendern einer Zuordnung

Sie können den API-Schlüsselwert verwenden, um eine Zuordnung in Ihrer Anwendung mithilfe von `MapLibre` zu rendern. Dies unterscheidet sich etwas von der Verwendung der API-Schlüssel in anderen Amazon Location APIs, die Sie direkt aufrufen, da diese Aufrufe für Sie `MapLibre` vornimmt.

Der folgende Beispielcode zeigt die Verwendung des `-API-Schlüssels` zum Rendern einer Karte auf einer einfachen Webseite mithilfe der `MapLibre microSD JS-Kartensteuerung`. Damit dieser Code ordnungsgemäß funktioniert, ersetzen Sie die *ExampleMap* Zeichenfolgen `v1.public.your-api-key-value`, `us-east-1` und durch Werte, die Ihrem entsprechen AWS-Konto.

```
<!-- index.html -->
<html>
  <head>
    <link href="https://unpkg.com/maplibre-gl@1.14.0/dist/maplibre-gl.css"
rel="stylesheet" />
    <style>
      body { margin: 0; }
      #map { height: 100vh; }
```

```
</style>
</head>
<body>
  <!-- Map container -->
  <div id="map" />
  <!-- JavaScript dependencies -->
  <script src="https://unpkg.com/maplibre-gl@1.14.0/dist/maplibre-gl.js"></script>
  <script>
    const apiKey = "v1.public.your-api-key-value"; // API key
    const region = "us-east-1"; // Region
    const mapName = "ExampleMap"; // Map name
    // URL for style descriptor
    const styleUrl = `https://maps.geo.${region}.amazonaws.com/maps/v0/maps/
  ${mapName}/style-descriptor?key=${apiKey}`;
    // Initialize the map
    const map = new maplibregl.Map({
      container: "map",
      style: styleUrl,
      center: [-123.1187, 49.2819],
      zoom: 11,
    });
    map.addControl(new maplibregl.NavigationControl(), "top-left");
  </script>
</body>
</html>
```

Verwalten der Lebensdauer von API-Schlüsseln

Sie können API-Schlüssel erstellen, die auf unbestimmte Zeit funktionieren. Wenn Sie jedoch einen temporären API-Schlüssel erstellen, API-Schlüssel regelmäßig rotieren oder einen vorhandenen API-Schlüssel widerrufen möchten, können Sie den Ablauf von API-Schlüsseln verwenden.

Wenn Sie einen neuen API-Schlüssel erstellen oder einen vorhandenen aktualisieren, können Sie die Ablaufzeit für diesen API-Schlüssel festlegen.

- Wenn ein API-Schlüssel seine Ablaufzeit erreicht hat, wird der Schlüssel automatisch deaktiviert. Inaktive Schlüssel können nicht mehr verwendet werden, um Kartenanforderungen zu stellen.
- Sie können einen API-Schlüssel 90 Tage nach der Deaktivierung löschen.
- Wenn Sie einen inaktiven Schlüssel haben, den Sie noch nicht gelöscht haben, können Sie ihn wiederherstellen, indem Sie die Ablaufzeit auf einen zukünftigen Zeitpunkt aktualisieren.
- Um einen permanenten Schlüssel zu erstellen, können Sie die Ablaufzeit entfernen.

- Wenn Sie versuchen, einen API-Schlüssel zu deaktivieren, der innerhalb der letzten 7 Tage verwendet wurde, werden Sie aufgefordert, zu bestätigen, dass Sie die Änderung vornehmen möchten. Wenn Sie die Amazon Location Service API oder die verwenden AWS CLI, erhalten Sie eine Fehlermeldung, es sei denn, Sie setzen den ForceUpdate Parameter auf „true“.

Erlauben des nicht authentifizierten Gastzugriffs auf Ihre Anwendung mit Amazon Cognito

Sie können die Amazon Cognito-Authentifizierung als Alternative zur direkten Verwendung von AWS Identity and Access Management (IAM) sowohl mit Frontend-SDKs als auch mit direkten HTTPS-Anfragen verwenden.

Sie können diese Form der Authentifizierung aus folgenden Gründen verwenden:

- Nicht authentifizierte Benutzer – Wenn Sie eine Website mit anonymen Benutzern haben, können Sie Amazon Cognito-Identitätspools verwenden. Weitere Informationen finden Sie im Abschnitt auf [the section called “Verwenden von Amazon Cognito”](#).
- Ihre eigene Authentifizierung – Wenn Sie Ihren eigenen Authentifizierungsprozess verwenden oder mehrere Authentifizierungsmethoden kombinieren möchten, können Sie Amazon Cognito Federated Identities verwenden. Weitere Informationen finden Sie unter [Erste Schritte mit Verbundidentitäten](#) im Amazon Cognito-Entwicklerhandbuch.

Amazon Cognito bietet Authentifizierung, Autorisierung und Benutzerverwaltung für Web- und mobile Apps. Sie können nicht authentifizierte Amazon Cognito-Identitätspools mit Amazon Location verwenden, um Anwendungen zu ermöglichen, temporäre, begrenzte AWS Anmeldeinformationen abzurufen.

Weitere Informationen finden Sie unter [Erste Schritte mit Benutzerpools](#) im Amazon Cognito-Entwicklerhandbuch.

Note

Für Entwickler von Mobilgeräten stellt Amazon Location SDKs für die mobile Authentifizierung sowohl für iOS als auch für Android bereit. Weitere Informationen finden Sie in den folgenden Github-Repositories:

- [Amazon Location Service Mobile Authentication SDK für iOS](#)

- [Amazon Location Service Mobile Authentication SDK für Android](#)

Amazon-Cognito-Identitätspool erstellen

Sie können Amazon Cognito-Identitätspools erstellen, um nicht authentifizierten Gastzugriff auf Ihre Anwendung über die Amazon Cognito-Konsole, die AWS CLI oder die Amazon Cognito-APIs zu ermöglichen.

Important

Der von Ihnen erstellte Pool muss sich in demselben AWS-Konto und derselben AWS Region befinden wie die von Ihnen verwendeten Amazon Location Service-Ressourcen.

Sie können IAM-Richtlinien, die nicht authentifizierten Identitätsrollen zugeordnet sind, mit den folgenden Aktionen verwenden:

- `geo:GetMap*`
- `geo:SearchPlaceIndex*`
- `geo:GetPlace`
- `geo:CalculateRoute*`
- `geo:GetGeofence`
- `geo:ListGeofences`
- `geo:PutGeofence`
- `geo:BatchDeleteGeofence`
- `geo:BatchPutGeofence`
- `geo:BatchEvaluateGeofences`
- `geo:GetDevicePosition*`
- `geo:ListDevicePositions`
- `geo:BatchDeleteDevicePositionHistory`
- `geo:BatchGetDevicePosition`
- `geo:BatchUpdateDevicePosition`

Das Einschließen anderer Amazon Location-Aktionen hat keine Auswirkungen und nicht authentifizierte Identitäten können sie nicht aufrufen.

Example

So erstellen Sie einen Identitäten-Pool mit der Amazon Cognito-Konsole

1. Melden Sie sich bei der [Amazon-Cognito-Konsole](#) an.
2. Klicken Sie auf Manage Identity Pools (Identitäten-Pools verwalten).
3. Wählen Sie Neuen Identitäten-Pool erstellen und geben Sie dann einen Namen für Ihren Identitäten-Pool ein.
4. Wählen Sie im Abschnitt Einklappbare nicht authentifizierte Identitäten die Option Zugriff auf nicht authentifizierte Identitäten aktivieren aus.
5. Wählen Sie Pool erstellen.
6. Wählen Sie aus, welche IAM-Rollen Sie mit Ihrem Identitäten-Pool verwenden möchten.
7. Erweitern Sie Details anzeigen.
8. Geben Sie unter Nicht authentifizierte Identitäten einen Rollennamen ein.
9. Erweitern Sie den Abschnitt Richtliniendokument anzeigen und wählen Sie dann Bearbeiten, um Ihre Richtlinie hinzuzufügen.
10. Fügen Sie Ihre Richtlinie hinzu, um Zugriff auf Ihre -Ressourcen zu gewähren.

Im Folgenden finden Sie Richtlinienbeispiele für Karten, Orte, Tracker und Routen. Um die Beispiele für Ihre eigene Richtlinie zu verwenden, ersetzen Sie die Platzhalter *region* und *accountID*:

Maps policy example

Die folgende Richtlinie gewährt schreibgeschützten Zugriff auf eine Kartenressource mit dem Namen *ExampleMap*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MapsReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:GetMapStyleDescriptor",
```

```

        "geo:GetMapGlyphs",
        "geo:GetMapSprites",
        "geo:GetMapTile"
    ],
    "Resource": "arn:aws:geo:region:accountID:map/ExampleMap"
}
]
}

```

Durch das Hinzufügen einer übereinstimmenden [IAM-Bedingung](#) `aws:referer` können Sie den Browserzugriff auf Ihre Ressourcen auf eine Liste von URLs oder URL-Präfixen beschränken. Das folgende Beispiel erlaubt den Zugriff auf eine Kartenressource namens nur `RasterEsriImagery` von der Website `ausexample.com`:

Warning

Obwohl den Zugriff einschränken `aws:referer` kann, handelt es sich nicht um einen sicheren Mechanismus. Ein öffentlich bekannter Referer-Header-Wert sollte möglichst nicht eingeschlossen werden. Nicht autorisierte Parteien können mit modifizierten oder benutzerdefinierten Browsern einen beliebigen `aws:referer`-Wert ihrer Wahl bereitstellen. Daher `aws:referer` sollte nicht verwendet werden, um Unbefugte daran zu hindern, direkte AWS Anfragen zu stellen. Die Funktion wird nur bereitgestellt, damit Kunden ihre digitalen, in Amazon S3 gespeicherten Inhalte vor der Referenzierung auf nicht autorisierte Drittanbieter-Websites schützen können. Weitere Informationen finden Sie unter [AWS: Referer](#).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:GetMap*",
      "Resource": "arn:aws:geo:us-west-2:111122223333:map/RasterEsriImagery",
      "Condition": {
        "StringLike": {
          "aws:referer": [
            "https://example.com/*",
            "https://www.example.com/*"
          ]
        }
      }
    }
  ]
}

```

```

    ]
  }
}
]
}

```

Wenn Sie [Tangram verwenden](#), um eine Karte anzuzeigen, verwendet es nicht die Stildeskriptoren, Glyphs oder -Sprites, die von der Maps-API zurückgegeben werden. Stattdessen wird sie konfiguriert, indem auf eine ZIP-Datei verwiesen wird, die Stilregeln und erforderliche Komponenten enthält. Die folgende Richtlinie gewährt schreibgeschützten Zugriff auf eine Kartenressource mit dem Namen *ExampleMap* für die `-GetMapTileOperation`.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MapsReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:GetMapTile"
      ],
      "Resource": "arn:aws:geo:region:accountID:map/ExampleMap"
    }
  ]
}

```

Places policy example

Die folgende Richtlinie gewährt schreibgeschützten Zugriff auf eine Ortsindexressource mit dem Namen *ExamplePlaceIndex* um nach Orten nach Text oder Positionen zu suchen.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PlacesReadOnly",
      "Effect": "Allow",
      "Action": [
        "geo:SearchPlaceIndex*"
      ],

```

```

        "geo:GetPlace"
      ],
      "Resource": "arn:aws:geo:region:accountID:place-index/ExamplePlaceIndex"
    }
  ]
}

```

Wenn Sie eine [IAM-Bedingung](#) hinzufügen, die übereinstimmt, `aws:referer` können Sie den Browserzugriff auf Ihre Ressourcen auf eine Liste von URLs oder URL-Präfixen beschränken. Im folgenden Beispiel wird der Zugriff auf eine Ortsindexressource mit dem Namen `ExamplePlaceIndex` von allen verweisenden Websites mit Ausnahme von `verweigertexample.com`.

Warning

Obwohl den Zugriff einschränken `aws:referer` kann, handelt es sich nicht um einen sicheren Mechanismus. Ein öffentlich bekannter Referer-Header-Wert sollte möglichst nicht eingeschlossen werden. Nicht autorisierte Parteien können mit modifizierten oder benutzerdefinierten Browsern einen beliebigen `aws:referer`-Wert ihrer Wahl bereitstellen. Daher `aws:referer` sollte nicht verwendet werden, um Unbefugte daran zu hindern, direkte AWS Anfragen zu stellen. Die Funktion wird nur bereitgestellt, damit Kunden ihre digitalen, in Amazon S3 gespeicherten Inhalte vor der Referenzierung auf nicht autorisierte Drittanbieter-Websites schützen können. Weitere Informationen finden Sie unter [AWS: Referer](#).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:*",
      "Resource": "arn:aws:geo:us-west-2:111122223333:place-
index/ExamplePlaceIndex",
      "Condition": {
        "StringLike": {
          "aws:referer": [
            "https://example.com/*",
            "https://www.example.com/*"
          ]
        }
      }
    }
  ]
}

```

```

    }
  }
]
}

```

Trackers policy example

Die folgende Richtlinie gewährt Zugriff auf eine Tracker-Ressource mit dem Namen , um Gerätepositionen *ExampleTracker* zu aktualisieren.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UpdateDevicePosition",
      "Effect": "Allow",
      "Action": [
        "geo:BatchUpdateDevicePosition"
      ],
      "Resource": "arn:aws:geo:region:accountID:tracker/ExampleTracker"
    }
  ]
}

```

Wenn Sie eine [IAM-Bedingung](#) hinzufügen, die übereinstimmt, `aws:referrer` können Sie den Browserzugriff auf Ihre Ressourcen auf eine Liste von URLs oder URL-Präfixen beschränken. Im folgenden Beispiel wird der Zugriff auf eine Tracker-Ressource mit dem Namen *ExampleTracker* von allen verweisenden Websites mit Ausnahme von `verweigertextample.com`.

Warning

Obwohl den Zugriff einschränken `aws:referrer` kann, handelt es sich nicht um einen sicheren Mechanismus. Ein öffentlich bekannter Referer-Header-Wert sollte möglichst nicht eingeschlossen werden. Nicht autorisierte Parteien können mit modifizierten oder benutzerdefinierten Browsern einen beliebigen `aws:referrer`-Wert ihrer Wahl bereitstellen. Daher `aws:referrer` sollte nicht verwendet werden, um Unbefugte daran zu hindern, direkte AWS Anfragen zu stellen. Die Funktion wird nur bereitgestellt, damit Kunden ihre digitalen, in Amazon S3 gespeicherten Inhalte

vor der Referenzierung auf nicht autorisierte Drittanbieter-Websites schützen können. Weitere Informationen finden Sie unter [AWS: Referer](#) .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:GetDevice*",
      "Resource": "arn:aws:geo:us-
west-2:111122223333:tracker/ExampleTracker",
      "Condition": {
        "StringLike": {
          "aws:referer": [
            "https://example.com/*",
            "https://www.example.com/*"
          ]
        }
      }
    }
  ]
}
```

Routes policy example

Die folgende Richtlinie gewährt Zugriff auf eine Routenrechner-Ressource namens , um eine Route *ExampleCalculator* zu berechnen.

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

```
}
```

Wenn Sie eine [IAM-Bedingung](#) hinzufügen, die übereinstimmt, `aws:referer` können Sie den Browserzugriff auf Ihre Ressourcen auf eine Liste von URLs oder URL-Präfixen beschränken. Das folgende Beispiel verweigert den Zugriff auf einen Routenrechner mit dem Namen *ExampleCalculator* von allen verweisenden Websites mit Ausnahme von `example.com`.

⚠ Warning

Obwohl den Zugriff einschränken `aws:referer` kann, handelt es sich nicht um einen sicheren Mechanismus. Ein öffentlich bekannter Referer-Header-Wert sollte möglichst nicht eingeschlossen werden. Nicht autorisierte Parteien können mit modifizierten oder benutzerdefinierten Browsern einen beliebigen `aws:referer`-Wert ihrer Wahl bereitstellen. Daher `aws:referer` sollte nicht verwendet werden, um Unbefugte daran zu hindern, direkte AWS Anfragen zu stellen. Die Funktion wird nur bereitgestellt, damit Kunden ihre digitalen, in Amazon S3 gespeicherten Inhalte vor der Referenzierung auf nicht autorisierte Drittanbieter-Websites schützen können. Weitere Informationen finden Sie unter [AWS: Referer](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "geo:*",
      "Resource": "arn:aws:geo:us-west-2:111122223333:route-
calculator/ExampleCalculator",
      "Condition": {
        "StringLike": {
          "aws:referer": [
            "https://example.com/*",
            "https://www.example.com/*"
          ]
        }
      }
    }
  ]
}
```

```
}
```

Note

Während nicht authentifizierte Identitätspools für die Offenlegung auf ungesicherten Websites vorgesehen sind, beachten Sie, dass sie gegen standardmäßige, zeitlich begrenzte AWS Anmeldeinformationen ausgetauscht werden.

Es ist wichtig, die IAM-Rollen, die nicht authentifizierten Identitätspools zugeordnet sind, entsprechend einzugrenzen.

11. Wählen Sie Zulassen, um Ihre Identitätspools zu erstellen.

Der resultierende Identitätspool folgt der Syntax `<region>:<GUID>`.

Beispielsweise:

```
us-east-1:1sample4-5678-90ef-aaaa-1234abcd56ef
```

Weitere Beispiele für Amazon Location finden Sie unter [the section called "Beispiele für identitätsbasierte Richtlinien"](#).

Verwenden der Amazon Cognito-Identitätspools in JavaScript

Im folgenden Beispiel wird der von Ihnen erstellte nicht authentifizierte Identitätspool gegen Anmeldeinformationen ausgetauscht, die dann zum Abrufen des Stildeskriptors für Ihre Kartenressource verwendet werden *ExampleMap*.

```
const AWS = require("aws-sdk");

const credentials = new AWS.CognitoIdentityCredentials({
  IdentityPoolId: "<identity pool ID>" // for example, us-east-1:1sample4-5678-90ef-
  aaaa-1234abcd56ef
});

const client = new AWS.Location({
  credentials,
  region: AWS.config.region || "<region>"
});
```

```
console.log(await client.getMapStyleDescriptor("ExampleMap").promise());
```

Note

Abrufen von Anmeldeinformationen von nicht authentifizierten Identitäten sind eine Stunde lang gültig.

Im Folgenden finden Sie ein Beispiel für eine Funktion, die Anmeldeinformationen automatisch erneuert, bevor sie ablaufen.

```
async function refreshCredentials() {
  await credentials.refreshPromise();
  // schedule the next credential refresh when they're about to expire
  setTimeout(refreshCredentials, credentials.expireTime - new Date());
}
```

Um diese Arbeit zu vereinfachen, können Sie den Amazon Location verwenden [Authentifizierungshelfer](#). Dies ersetzt sowohl das Abrufen der Anmeldeinformationen als auch deren Aktualisierung. In diesem Beispiel wird das AWS SDK für JavaScript v3 verwendet.

```
import { LocationClient, GetMapStyleDescriptorCommand } from "@aws-sdk/client-location";
import { withIdentityPoolId } from "@aws/amazon-location-utilities-auth-helper";

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

// Create an authentication helper instance using credentials from Cognito
const authHelper = await withIdentityPoolId(identityPoolId);

const client = new LocationClient({
  region: "<region>", // The region containing both the identity pool and tracker resource
  ...authHelper.getLocationClientConfig(), // Provides configuration required to make requests to Amazon Location
});

const input = {
  MapName: "ExampleMap",
};
```

```
const command = new GetMapStyleDescriptorCommand(input);  
  
console.log(await client.send(command));
```

Nächste Schritte

- Um Ihre Rollen zu ändern, wechseln Sie zur [IAM-Konsole](#) .
- Um Ihre Identitätspools zu verwalten, gehen Sie zur [Amazon Cognito-Konsole](#) .

Überwachen von Amazon Location Service

Wenn Sie Amazon Location Service verwenden, können Sie Ihre Nutzung und Ressourcen im Laufe der Zeit überwachen, indem Sie Folgendes verwenden:

- Amazon CloudWatch. Überwacht Ihre Amazon Location Service-Ressourcen und stellt Metriken mit Statistiken nahezu in Echtzeit bereit.
- AWS CloudTrail. Stellt die Ereignisnachverfolgung aller Aufrufe an Amazon Location Service APIs bereit.

Dieser Abschnitt enthält Informationen zur Verwendung dieser Services.

Themen

- [Überwachen von Amazon Location Service mit Amazon CloudWatch](#)
- [Protokollierung und Überwachung mit AWS CloudTrail](#)

Überwachen von Amazon Location Service mit Amazon CloudWatch

Amazon CloudWatch überwacht Ihre AWS Ressourcen und die Anwendungen, auf denen Sie ausgeführt werden, nahezu AWS in Echtzeit. Sie können Amazon Location-Ressourcen mit überwachen CloudWatch, das Rohdaten sammelt und Metriken nahezu in Echtzeit in aussagekräftige Statistiken verarbeitet. Sie können historische Informationen für bis zu 15 Monate anzeigen oder Metriken durchsuchen, die Sie in der Amazon- CloudWatch Konsole anzeigen können, um mehr über Ihre Amazon Location-Ressourcen zu erfahren. Sie können Alarme auch einstellen, indem Sie Schwellenwerte definieren, Benachrichtigungen senden oder Maßnahmen ergreifen, wenn diese Schwellenwerte erreicht werden.

Weitere Informationen finden Sie im [Amazon CloudWatch -Benutzerhandbuch](#).

Themen

- [Metriken von Amazon Location Service, die nach Amazon exportiert werden CloudWatch](#)
- [Anzeigen von Amazon Location Service-Metriken](#)
- [Erstellen von CloudWatch Alarmen für Amazon Location Service-Metriken](#)
- [Verwenden von CloudWatch zur Überwachung der Nutzung anhand von Kontingenten](#)
- [CloudWatch -Metrikbeispiele für Amazon Location Service](#)

Metriken von Amazon Location Service, die nach Amazon exportiert werden CloudWatch

Metriken sind zeitgeordnete Datenpunkte, die nach exportiert werden CloudWatch. Eine Dimension ist ein Name-Wert-Paar, das die Metrik identifiziert. Weitere Informationen finden Sie unter [Verwenden von CloudWatch Metriken](#) und [CloudWatch](#) Dimensionen im Amazon CloudWatch - Benutzerhandbuch.

Im Folgenden finden Sie Metriken, die Amazon Location Service CloudWatch in im AWS/Location Namespace exportiert.

Metrik	Beschreibung
CallCount	<p>Die Anzahl der Aufrufe an einen bestimmten API-Endpunkt.</p> <p>Gültige Dimensionen: Amazon Location Service API-Namen</p> <p>Gültige Statistiken: Summe</p> <p>Einheiten: Anzahl</p>
ErrorCount	<p>Die Anzahl der Fehlerantworten von Aufrufen an einen bestimmten API-Endpunkt.</p> <p>Gültige Dimensionen: Amazon Location Service API-Namen</p> <p>Gültige Statistiken: Summe</p> <p>Einheiten: Anzahl</p>

Metrik	Beschreibung
SuccessCount	Die Anzahl der erfolgreichen Aufrufe an einen bestimmten API-Endpunkt. Gültige Dimensionen: Amazon Location Service API-Namen Gültige Statistiken: Summe Einheiten: Anzahl
CallLatency	Die Zeit, die der Vorgang benötigt, um zu verarbeiten und eine Antwort zurückzugeben, wenn ein Aufruf an einen bestimmten API-Endpunkt erfolgt. Gültige Dimensionen: Amazon Location Service API-Namen Gültige Statistik: Durchschnitt Einheiten: Millisekunden

Anzeigen von Amazon Location Service-Metriken

Sie können Metriken für Amazon Location Service in der Amazon CloudWatch Konsole oder mithilfe der Amazon CloudWatch API anzeigen.

So zeigen Sie Metriken mit der CloudWatch Konsole an

Example

1. Öffnen Sie die - CloudWatch Konsole unter <https://console.aws.amazon.com/cloudwatch/>.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie auf der Registerkarte Alle Metriken den Amazon Location-Namespace aus.
4. Wählen Sie den Metriktyp aus, der angezeigt werden soll.
5. Wählen Sie die Metrik aus und fügen Sie sie dem Diagramm hinzu.

Weitere Informationen finden Sie unter [Anzeigen verfügbarer Metriken](#) im Amazon- CloudWatch Benutzerhandbuch.

Erstellen von CloudWatch Alarmen für Amazon Location Service-Metriken

Sie können verwenden CloudWatch , um Alarme für Ihre Amazon Location Service-Metriken festzulegen. Sie können beispielsweise einen Alarm in erstellen, CloudWatch um eine E-Mail zu senden, wenn ein Anstieg der Fehleranzahl auftritt.

Die folgenden Themen geben Ihnen eine grobe Übersicht darüber, wie Sie die Alarme mithilfe von CloudWatch einrichten können. Detaillierte Anweisungen finden Sie unter [Verwenden von Alarmen](#) im Amazon- CloudWatch Benutzerhandbuch.

So richten Sie Alarme mithilfe der CloudWatch Konsole ein

Example

1. Öffnen Sie die - CloudWatch Konsole unter <https://console.aws.amazon.com/cloudwatch/>.
2. Wählen Sie im Navigationsbereich Alarm aus.
3. Wählen Sie Alarm erstellen aus.
4. Wählen Sie Select metric (Metrik auswählen) aus.
5. Wählen Sie auf der Registerkarte Alle Metriken den Amazon Location-Namespace aus.
6. Wählen Sie eine Metrikkategorie aus.
7. Suchen Sie die Zeile mit der Metrik, für die Sie einen Alarm erstellen möchten, und aktivieren Sie dann das Kontrollkästchen neben dieser Zeile.
8. Wählen Sie Select metric (Metrik auswählen) aus.
9. Geben Sie unter Metrik die Werte ein.
10. Geben Sie die Alarmbedingungen an.
11. Wählen Sie Weiter aus.
12. Wenn Sie eine Benachrichtigung senden möchten, wenn die Alarmbedingungen erfüllt sind:
 - Wählen Sie unter Alarmstatusauslöser den Alarmstatus aus, um eine Benachrichtigung zum Senden anzufordern.
 - Wählen Sie unter SNS-Thema auswählen die Option Neues Thema erstellen aus, um ein neues Amazon Simple Notification Service (Amazon SNS)-Thema zu erstellen. Geben Sie den Themennamen und die E-Mail-Adresse ein, an die die Benachrichtigung gesendet werden soll.
 - Unter Benachrichtigung senden, um zusätzliche E-Mail-Adressen einzugeben, an die die Benachrichtigung gesendet werden soll.
 - Wählen Sie Add notification (Benachrichtigung hinzufügen) aus. Diese Liste wird gespeichert und erscheint für künftige Alarme in der Liste.

13. Klicken Sie auf Weiter, wenn Sie fertig sind.

14. Geben Sie einen Namen und eine Beschreibung für den Alarm ein und wählen Sie dann Weiter aus.

15. Bestätigen Sie die Alarmdetails und wählen Sie dann Weiter aus.

Note

Wenn Sie ein neues Amazon SNS-Thema erstellen, müssen Sie die E-Mail-Adresse verifizieren, bevor eine Benachrichtigung gesendet werden kann. Wenn die E-Mail nicht verifiziert wurde, wird die Benachrichtigung nicht empfangen, wenn ein Alarm durch eine Statusänderung ausgelöst wird.

Weitere Informationen zum Einrichten von Alarmen mithilfe der CloudWatch Konsole finden Sie unter [Erstellen eines Alarms, der E-Mails sendet](#) im Amazon CloudWatch-Benutzerhandbuch.

Verwenden von CloudWatch zur Überwachung der Nutzung anhand von Kontingenten

Sie können Amazon- CloudWatch Alarme erstellen, um Sie zu benachrichtigen, wenn Ihre Auslastung eines bestimmten Kontingents einen konfigurierbaren Schwellenwert überschreitet. Auf diese Weise können Sie erkennen, wann Sie sich in der Nähe Ihrer Kontingentlimits befinden, und entweder Ihre Auslastung anpassen, um Kostenüberschreitungen zu vermeiden, oder bei Bedarf eine Kontingenterhöhung beantragen. Informationen zur Verwendung von CloudWatch zur Überwachung von Kontingenten finden Sie unter [Visualisieren Ihrer Servicekontingente und Einstellen von Alarmen](#) im Amazon- CloudWatch Benutzerhandbuch.

CloudWatch -Metrikbeispiele für Amazon Location Service

Sie können die [GetMetricData](#)-API verwenden, um Metriken für Amazon Location abzurufen.

- Sie können beispielsweise einen Alarm für den Fall überwachen `CallCount` und einstellen, dass eine Verringerung der Zahl auftritt.

Die Überwachung der `CallCount` Metriken für `SendDeviceLocation` kann Ihnen helfen, einen Überblick über verfolgte Komponenten zu erhalten. Wenn die `CallCount` abnimmt, bedeutet dies, dass nachverfolgte Komponenten, wie z. B. eine Flotte von Lkw, ihre aktuellen Standorte nicht mehr gesendet haben. Wenn Sie dafür einen Alarm einrichten, können Sie sich benachrichtigen lassen, dass ein Problem aufgetreten ist.

- Ein weiteres Beispiel: Sie können einen Alarm überwachen `ErrorCount` und einstellen, wenn eine Zahlenspitze auftritt.

Tracker müssen mit Geofence-Sammlungen verknüpft sein, damit Gerätestandorte anhand von Geofences ausgewertet werden können. Wenn Sie über eine Geräteflotte verfügen, die kontinuierliche Standortaktualisierungen erfordert, zeigt die für `CallCount` `BatchEvaluateGeofence` oder `BatchPutDevicePosition` der Rückgang auf Null an, dass Updates nicht mehr ausgeführt werden.

Im Folgenden finden Sie eine Beispielausgabe für [GetMetricData](#) mit den Metriken für `CallCount` und `ErrorCount` zum Erstellen von Kartenressourcen.

```
{
  "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"
}
]
```

Protokollierung und Überwachung mit AWS CloudTrail

AWS CloudTrail ist ein Service, der die Aktionen eines Benutzers, einer Rolle oder eines - AWS-Services aufzeichnet. CloudTrail zeichnet alle API-Aufrufe als Ereignisse auf. Sie können Amazon Location Service mit verwenden, CloudTrail um Ihre API-Aufrufe zu überwachen, einschließlich Aufrufen von der Amazon Location Service-Konsole und AWS SDK-Aufrufen der Amazon Location Service API-Operationen.

Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen S3-Bucket aktivieren, einschließlich Ereignissen für Amazon Location Service. Wenn Sie keinen Trail konfigurieren, können Sie trotzdem die neuesten Ereignisse in der CloudTrail Konsole unter Ereignisverlauf anzeigen. Anhand der von CloudTrail gesammelten Informationen können Sie die an Amazon Location Service gestellte Anfrage, die IP-Adresse, von der die Anfrage gestellt wurde, den Initiator der Anfrage, den Zeitpunkt der Anfrage und zusätzliche Details bestimmen.

Weitere Informationen zu CloudTrail finden Sie im [AWS CloudTrail -Benutzerhandbuch](#).

Themen

- [Amazon Location Service-Informationen in CloudTrail](#)
- [Grundlegendes zu Amazon Location Service-Protokolldateieinträgen](#)

Amazon Location Service-Informationen in CloudTrail

CloudTrail wird beim Erstellen des AWS Kontos in Ihrem Konto aktiviert. Wenn eine Aktivität in Amazon Location Service auftritt, wird diese Aktivität in einem - CloudTrail Ereignis zusammen

mit anderen -AWS-Serviceereignissen im Ereignisverlauf aufgezeichnet. Sie können die neusten Ereignisse in Ihr AWS-Konto herunterladen und dort suchen und anzeigen. Weitere Informationen finden Sie unter [Anzeigen von Ereignissen mit dem CloudTrail Ereignisverlauf](#).

Erstellen Sie für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS Konto, einschließlich Ereignissen für Amazon Location Service, einen Trail. Ein Trail ermöglicht CloudTrail die Bereitstellung von Protokolldateien an einen S3-Bucket. Wenn Sie ein Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS-Partition und stellt die Protokolldateien für den von Ihnen angegebenen S3 Bucket bereit. Darüber hinaus können Sie andere -AWS-Services konfigurieren, um die in den CloudTrail Protokollen erfassten Ereignisdaten weiter zu analysieren und entsprechend zu agieren.

Weitere Informationen finden Sie hier:

- [Übersicht zum Erstellen eines Trails](#)
- [CloudTrail Unterstützte Services und Integrationen](#)
- [Konfigurieren von Amazon SNS-Benachrichtigungen für CloudTrail](#)
- [Empfangen von CloudTrail Protokolldateien aus mehreren Regionen](#) und [Empfangen von CloudTrail Protokolldateien aus mehreren Konten](#)

Alle Amazon Location Service-Aktionen werden von protokolliert CloudTrail und sind in den [API-Referenzen zu Amazon Location Service](#) dokumentiert. Aufrufe der DescribeTracker Aktionen CreateTracker, UpdateTracker und erzeugen beispielsweise Einträge in den CloudTrail Protokolldateien.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Anhand der Identitätsinformationen können Sie feststellen, ob eine Anforderung gestellt wurde:

- Mit Root- oder AWS Identity and Access Management (IAM)-Benutzeranmeldeinformationen.
- Mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen verbundenen Benutzer.
- Durch einen anderen AWS-Service.

Weitere Informationen finden Sie unter [CloudTrail userIdentity-Element](#).

Grundlegendes zu Amazon Location Service-Protokolldateieinträgen

Ein Trail ist eine Konfiguration, die die Bereitstellung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen S3-Bucket oder an Amazon CloudWatch Logs ermöglicht. Weitere Informationen finden Sie unter [Arbeiten mit CloudTrail Protokolldateien](#) im AWS CloudTrail - Benutzerhandbuch.

CloudTrail -Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, Datum und Uhrzeit der Operation, Anforderungsparameter usw.

Note

CloudTrail -Protokolldateien sind kein geordnetes Stacktrace der öffentlichen API-Aufrufe und erscheinen daher nicht in einer bestimmten Reihenfolge. Um die Reihenfolge der Operationen zu bestimmen, verwenden Sie [eventTime](#).

Das folgende Beispiel zeigt einen - CloudTrail Protokolleintrag, der die -CreateTrackerOperation demonstriert, die eine Tracker-Ressource erstellt.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "123456789012",
    "arn": "arn:aws:geo:us-east-1:123456789012:tracker/ExampleTracker",
    "accountId": "123456789012",
    "accessKeyId": "123456789012",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "123456789012",
        "arn": "arn:aws:geo:us-east-1:123456789012:tracker/ExampleTracker",
        "accountId": "123456789012",
        "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": "192.0.2.0/24-TEST-NET-1",
"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": "123456789012",
}

```

Im Folgenden wird ein Protokolleintrag für die `-DescribeTrackerOperation` gezeigt, der die Details einer Tracker-Ressource zurückgibt.

```

{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "123456789012",
    "arn": "arn:partition:service:region:account-id:resource-id",
    "accountId": "123456789012",
    "accessKeyId": "123456789012",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "123456789012",

```

```

        "arn": "arn:partition:service:region:account-id:resource-id",
        "accountId": "123456789012",
        "userName": "exampleUser",
    },
    "webIdFederationData": {},
    "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2020-10-22T16:36:07Z"
    }
}
},
"eventTime": "2020-10-22T17:43:33Z",
"eventSource": "geo.amazonaws.com",
"eventName": "DescribeTracker",
"awsRegion": "us-east-1",
"sourceIPAddress": "192.0.2.0/24-TEST-NET-1",
"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"
},
"responseElements": null,
"requestID": "997d5f93-cfef-429a-bbed-daab417ceab4",
"eventID": "d9e0eebe-173c-477d-b0c9-d1d8292da103",
"readOnly": true,
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012",
}

```

Erstellen von Amazon Location Service-Ressourcen mit AWS CloudFormation

Amazon Location Service ist in integriert AWS CloudFormation, einem Service, der Ihnen hilft, Ihre -AWS-Ressourcen zu modellieren und einzurichten, sodass Sie weniger Zeit mit der Erstellung und Verwaltung Ihrer Ressourcen und Infrastruktur verbringen können. Sie erstellen eine Vorlage, die alle gewünschten AWS Ressourcen beschreibt (z. B. Amazon Location-Ressourcen) und diese Ressourcen für Sie AWS CloudFormation bereitstellt und konfiguriert.

Wenn Sie verwenden AWS CloudFormation, können Sie Ihre Vorlage wiederverwenden, um Ihre Amazon Location-Ressourcen konsistent und wiederholt einzurichten. Sie beschreiben Ihre

Ressourcen dann einmal und können die gleichen Ressourcen dann in mehreren AWS-Konten und -Regionen immer wieder bereitstellen.

Amazon Location und AWS CloudFormation Vorlagen

Um Ressourcen für Amazon Location und zugehörige Services bereitzustellen und zu konfigurieren, müssen Sie [AWS CloudFormation die Vorlagen](#) verstehen. Vorlagen sind formatierte Textdateien in JSON oder YAML. Diese Vorlagen beschreiben die Ressourcen, die Sie in Ihren AWS CloudFormation-Stacks bereitstellen möchten. Wenn Sie noch keine Erfahrungen mit JSON oder YAML haben, können Sie AWS CloudFormation Designer verwenden, der den Einstieg in die Arbeit mit AWS CloudFormation-Vorlagen erleichtert. Weitere Informationen finden Sie unter [Was ist AWS CloudFormation-Designer?](#) im AWS CloudFormation-Benutzerhandbuch.

Amazon Location unterstützt das Erstellen der folgenden Ressourcentypen in AWS CloudFormation:

- [AWS::Location::Map](#)
- [AWS::Location::PlaceIndex](#)
- [AWS::Location::RouteCalculator](#)
- [AWS::Location::Tracker](#)
- [AWS::Location::TrackerConsumer](#)
- [AWS::Location::GeofenceCollection](#)

Weitere Informationen, einschließlich Beispiele für JSON- und YAML-Vorlagen für Amazon Location-Ressourcen, finden Sie in der [Ressourcentypreferenz für Amazon Location Service](#) im AWS CloudFormation -Benutzerhandbuch.

Weitere Informationen zu AWS CloudFormation

Weitere Informationen zu AWS CloudFormation finden Sie in den folgenden Ressourcen.

- [AWS CloudFormation](#)
- [AWS CloudFormation-Benutzerhandbuch](#)
- [AWS CloudFormation API Referenz](#)
- [AWS CloudFormation-Benutzerhandbuch für die Befehlszeilenschnittstelle](#)

Sicherheit in Amazon Location Service

Cloud-Sicherheit bei AWS hat höchste Priorität. Als - AWS Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die entwickelt wurden, um die Anforderungen der sicherheitssensibelsten Organisationen zu erfüllen.

Sicherheit ist eine geteilte Verantwortung zwischen AWS und Ihnen. Das [Modell der geteilten Verantwortung](#) beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- Sicherheit der Cloud – AWS ist für den Schutz der Infrastruktur verantwortlich, die AWS Services in der ausführt AWS Cloud. stellt Ihnen AWS außerdem Services bereit, die Sie sicher nutzen können. Externe Prüfer testen und überprüfen im Rahmen der [AWS Compliance-Programme](#) regelmäßig die Wirksamkeit unserer Sicherheit. Informationen zu den Compliance-Programmen, die für Amazon Location Service gelten, finden Sie unter [AWS Im Rahmen des Compliance-Programms zugelassene -ServicesIm](#).
- Sicherheit in der Cloud – Ihre Verantwortung wird durch den - AWS Service bestimmt, den Sie verwenden. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der geteilten Verantwortung bei der Verwendung von Amazon Location einsetzen können. Die folgenden Themen zeigen Ihnen, wie Sie Amazon Location konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere - AWS Services verwenden, die Sie bei der Überwachung und Sicherung Ihrer Amazon Location-Ressourcen unterstützen.

Themen

- [Datenschutz in Amazon Location Service](#)
- [Identity and Access Management für Amazon Location Service](#)
- [Reaktion auf Vorfälle in Amazon Location Service](#)
- [Compliance-Validierung für Amazon Location Service](#)
- [Ausfallsicherheit bei Amazon Location Service](#)
- [Infrastruktursicherheit in Amazon Location Service](#)
- [Konfigurations- und Schwachstellenanalyse in Amazon Location](#)
- [Serviceübergreifende Confused-Deputy-Prävention](#)
- [Bewährte Sicherheitsmethoden für Amazon Location Service](#)

- [Bewährte Methoden für Amazon Location Service](#)

Datenschutz in Amazon Location Service

Das AWS [Modell der geteilten Verantwortung](#) gilt für den Datenschutz in Amazon Location Service. Wie in diesem Modell beschrieben, AWS ist für den Schutz der globalen Infrastruktur verantwortlich, die alle ausführt AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich.

Weitere Informationen zum Datenschutz finden Sie unter [Häufig gestellte Fragen zum Datenschutz](#). Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag [AWS -Modell der geteilten Verantwortung und in der DSGVO](#) im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir Ihnen, -Anmeldeinformationen zu schützen AWS-Konto und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einzurichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor Authentifizierung (MFA).
- Verwenden Sie SSL/TLS für die Kommunikation mit - AWS Ressourcen. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API- und Benutzeraktivitätsprotokollierung mit ein AWS CloudTrail.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen Standardsicherheitskontrollen in AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff auf AWS über eine Befehlszeilenschnittstelle oder eine API FIPS-140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-2](#).

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit Amazon Location oder anderen AWS-Services über die Konsole, API AWS CLI oder AWS SDKs arbeiten. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen

externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Datenschutz

Mit Amazon Location Service behalten Sie die Kontrolle über die Daten Ihrer Organisation. Amazon Location anonymisiert alle an Datenanbieter gesendeten Abfragen, indem Kundenmetadaten und Kontoinformationen entfernt werden.

Amazon Location verwendet keine Datenanbieter für die Nachverfolgung und Geofencing. Das bedeutet, dass Ihre sensiblen Daten in Ihrem AWS Konto verbleiben. Auf diese Weise können vertrauliche Standortinformationen wie Einrichtung, Komponente und Personalstandort von Dritten bewältigt, der Datenschutz von Benutzern geschützt und das Sicherheitsrisiko Ihrer Anwendung reduziert werden.

Weitere Informationen finden Sie unter Häufig gestellte [AWS Fragen zum Datenschutz](#).

Aufbewahrung von Daten an Amazon Location

Die folgenden Merkmale beziehen sich darauf, wie Amazon Location Daten für den Service sammelt und speichert:

- Amazon Location Service Trackers – Wenn Sie die Trackers-APIs verwenden, um den Standort von Entitäten zu verfolgen, können deren Koordinaten gespeichert werden. Gerätestandorte werden 30 Tage lang gespeichert, bevor sie vom Service gelöscht werden.
- Amazon Location Service Geofences – Wenn Sie die Geofences-APIs verwenden, um Interessenbereiche zu definieren, speichert der Service die von Ihnen bereitgestellten Geometrien. Sie müssen explizit gelöscht werden.

Note

Durch das Löschen Ihres AWS Kontos werden alle darin enthaltenen Ressourcen gelöscht. Weitere Informationen finden Sie unter Häufig gestellte [AWS Fragen zum Datenschutz](#).

Datenverschlüsselung im Ruhezustand für Amazon Location Service

Amazon Location Service bietet standardmäßig Verschlüsselung, um sensible Kundendaten im Ruhezustand mit - AWS eigenen Verschlüsselungsschlüsseln zu schützen.

- **AWS -eigene Schlüssel** – Amazon Location verwendet diese Schlüssel standardmäßig, um persönlich identifizierbare Daten automatisch zu verschlüsseln. Sie können keine - AWS eigenen Schlüssel anzeigen, verwalten oder verwenden oder deren Verwendung überprüfen. Sie müssen jedoch keine Maßnahmen ergreifen oder Programme zum Schutz der Schlüssel ändern, die zur Verschlüsselung Ihrer Daten verwendet werden. Weitere Informationen finden Sie unter [AWS -eigene Schlüssel](#) im AWS Key Management Service -Entwicklerhandbuch.

Die standardmäßige Verschlüsselung von Daten im Ruhezustand trägt dazu bei, den betrieblichen Aufwand und die Komplexität zu reduzieren, die mit dem Schutz vertraulicher Daten verbunden sind. Gleichzeitig können Sie damit sichere Anwendungen erstellen, die strenge Verschlüsselungsvorschriften und gesetzliche Auflagen erfüllen.

Sie können diese Verschlüsselungsebene zwar nicht deaktivieren oder einen alternativen Verschlüsselungstyp auswählen, aber Sie können eine zweite Verschlüsselungsebene über den vorhandenen - AWS eigenen Verschlüsselungsschlüsseln hinzufügen, indem Sie beim Erstellen Ihrer Tracker- und Geofence-Sammlungsressourcen einen vom Kunden verwalteten Schlüssel auswählen:

- **Kundenverwaltete Schlüssel** – Amazon Location unterstützt die Verwendung eines symmetrischen kundenverwalteten Schlüssels, den Sie erstellen, besitzen und verwalten, um eine zweite Verschlüsselungsebene gegenüber der vorhandenen AWS -eigenen Verschlüsselung hinzuzufügen. Da Sie die volle Kontrolle über diese Verschlüsselungsebene haben, können Sie beispielsweise folgende Aufgaben ausführen:
 - Festlegung und Pflege wichtiger Richtlinien
 - Festlegung und Aufrechterhaltung von IAM-Richtlinien und -Erteilungen
 - Aktivieren und Deaktivieren von Schlüsselrichtlinien
 - Rotieren von wichtigen kryptografischen Materialien
 - Hinzufügen von Tags
 - Erstellen von Schlüsse-Aliases
 - Schlüssel für das Löschen von Schlüsseln planen

Weitere Informationen finden Sie unter [Kundenverwalteter Schlüssel](#) im AWS Key Management Service -Entwicklerhandbuch.

In der folgenden Tabelle wird zusammengefasst, wie Amazon Location persönlich identifizierbare Daten verschlüsselt.

Datentyp	AWS Verschlüsselung von - eigenen Schlüsseln	Vom Kunden verwaltete Schlüsselverschlüsselung (optional)
<p>Position</p> <p>Eine Punktgeometrie, die die Gerätepositionsdetails enthält.</p>	Aktiviert	Aktiviert
<p>PositionProperties</p> <p>Eine Reihe von Schlüssel-Wert-Paaren, die der Positionsaktualisierung zugeordnet sind.</p>	Aktiviert	Aktiviert
<p>GeofenceGeometry</p> <p>Eine Polygon-Geofence-Geometrie, die den geofenierten Bereich darstellt.</p>	Aktiviert	Aktiviert
<p>DeviceId</p> <p>Die Geräteerkennung, die beim Hochladen einer Gerätepositionsaktualisierung in eine Tracker-Ressource angegeben wird.</p>	Aktiviert	Nicht unterstützt
<p>GeofenceId</p> <p>Eine Kennung, die beim Speichern einer Geofence-Geometrie oder eines Stapels von Geofences in einer bestimmten Geofence-Sammlung angegeben wird.</p>	Aktiviert	Nicht unterstützt

Note

Amazon Location aktiviert automatisch die Verschlüsselung im Ruhezustand mit AWS - eigenen Schlüsseln, um persönlich identifizierbare Daten kostenlos zu schützen. Für die Verwendung eines vom Kunden verwalteten Schlüssels fallen jedoch AWS KMS Gebühren an. Weitere Informationen zu Preisen finden Sie unter [AWS Key Management Service – Preise](#).

Weitere Informationen zu AWS KMS finden Sie unter [Was ist AWS Key Management Service?](#)

So verwendet Amazon Location Service Erteilungen in AWS KMS

Amazon Location benötigt eine [Ertelung](#), um Ihren vom Kunden verwalteten Schlüssel zu verwenden.

Wenn Sie eine [Tracker-Ressource](#) oder [Geofence-Sammlung](#) erstellen, die mit einem vom Kunden verwalteten Schlüssel verschlüsselt ist, erstellt Amazon Location in Ihrem Namen eine Erteilung, indem es eine [CreateGrant](#) Anfrage an sendet AWS KMS. Erteilungen in AWS KMS werden verwendet, um Amazon Location Zugriff auf einen KMS-Schlüssel in einem Kundenkonto zu gewähren.

Amazon Location erfordert die Erteilung, um Ihren vom Kunden verwalteten Schlüssel für die folgenden internen Vorgänge zu verwenden:

- Senden Sie [DescribeKey](#) Anfragen an , AWS KMS um zu überprüfen, ob die symmetrische vom Kunden verwaltete KMS-Schlüssel-ID, die beim Erstellen einer Tracker- oder Geofence-Sammlung eingegeben wurde, gültig ist.
- Senden Sie [GenerateDataKeyWithoutPlaintext](#) Anfragen an AWS KMS , um Datenschlüssel zu generieren, die mit Ihrem vom Kunden verwalteten Schlüssel verschlüsselt wurden.
- Senden Sie [Decrypt](#)-Anforderungen an AWS KMS , um die verschlüsselten Datenschlüssel zu entschlüsseln, damit sie zur Verschlüsselung Ihrer Daten verwendet werden können.

Sie können den Zugriff auf die Genehmigung jederzeit widerrufen oder den Zugriff des Services auf den vom Kunden verwalteten Schlüssel entfernen. Wenn Sie dies tun, kann Amazon Location nicht auf die Daten zugreifen, die mit dem vom Kunden verwalteten Schlüssel verschlüsselt wurden, was sich auf Operationen auswirkt, die von diesen Daten abhängig sind. Wenn Sie beispielsweise

versuchen, [Gerätepositionen](#) von einem verschlüsselten Tracker abzurufen, auf den Amazon Location nicht zugreifen kann, gibt der Vorgang einen `AccessDeniedException` Fehler zurück.

Einen kundenverwalteten Schlüssel erstellen

Sie können einen symmetrischen, vom Kunden verwalteten Schlüssel erstellen AWS Management Console, indem Sie die oder die AWS KMS APIs verwenden.

Einen symmetrischen kundenverwalteten Schlüssel erstellen

Folgen Sie den Schritten zum [Erstellen eines symmetrischen kundenverwalteten Schlüssels](#) im Entwicklerhandbuch zum AWS Key Management Service .

Schlüsselrichtlinie

Schlüsselrichtlinien steuern den Zugriff auf den vom Kunden verwalteten Schlüssel. Jeder vom Kunden verwaltete Schlüssel muss über genau eine Schlüsselrichtlinie verfügen, die aussagt, wer den Schlüssel wie verwenden kann. Wenn Sie Ihren vom Kunden verwalteten Schlüssel erstellen, können Sie eine Schlüsselrichtlinie angeben. Weitere Informationen finden Sie unter [Verwalten des Zugriffs auf kundenverwaltete Schlüssel](#) im Entwicklerhandbuch zum AWS Key Management Service .

Um Ihren vom Kunden verwalteten Schlüssel mit Ihren Amazon Location-Ressourcen zu verwenden, müssen die folgenden API-Operationen in der Schlüsselrichtlinie zulässig sein:

- [kms:CreateGrant](#): Fügt einem kundenverwalteten Schlüssel eine Erteilung hinzu. Gewährt Kontrollzugriff auf einen angegebenen KMS-Schlüssel, der den Zugriff auf [Erteilungsvorgänge](#) ermöglicht, die Amazon Location benötigt. Weitere Informationen zur [Verwendung von Grants](#) finden Sie im AWS Key Management Service -Entwicklerhandbuch.

Auf diese Weise kann Amazon Location Folgendes tun:

- `GenerateDataKeyWithoutPlainText` aufrufen, um einen verschlüsselten Datenschlüssel zu generieren und zu speichern, da der Datenschlüssel nicht sofort zum Verschlüsseln verwendet wird.
- `Decrypt` aufrufen, um den gespeicherten verschlüsselten Datenschlüssel für den Zugriff auf verschlüsselte Daten zu verwenden.
- Richten Sie einen ausscheidenden Prinzipal ein, um dem Service zu erlauben `RetireGrant`.
- [kms:DescribeKey](#) – Stellt die vom Kunden verwalteten Schlüsseldetails bereit, damit Amazon Location den Schlüssel validieren kann.

Im Folgenden finden Sie Beispiele für Richtlinienanweisungen, die Sie für Amazon Location hinzufügen können:

```
"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" : "*"    
  }  
]
```

Weitere Informationen zum [Festlegen von Berechtigungen in einer Richtlinie](#) finden Sie im AWS Key Management Service -Entwicklerhandbuch.

Weitere Informationen zur [Fehlerbehebung beim Schlüsselzugriff](#) finden Sie im AWS Key Management Service -Entwicklerhandbuch.

Angeben eines vom Kunden verwalteten Schlüssels für Amazon Location

Sie können einen vom Kunden verwalteten Schlüssel als zweite Verschlüsselungsebene für die folgenden Ressourcen festlegen:

- [Tracker-Ressource](#)
- [Geofence-Sammlung](#)

Wenn Sie eine Ressource erstellen, können Sie den Datenschlüssel angeben, indem Sie eine KMS-ID eingeben, mit der Amazon Location die identifizierbaren personenbezogenen Daten verschlüsselt, die von der Ressource gespeichert werden.

- KMS-ID – Eine [Schlüsselkennung](#) für einen vom AWS KMS Kunden verwalteten Schlüssel. Geben Sie eine Schlüssel-ID, einen Schlüssel-ARN, einen Alias-Namen oder einen Alias-ARN ein.

Amazon Location Service-Verschlüsselungskontext

Ein [Verschlüsselungskontext](#) ist ein optionaler Satz von Schlüssel-Wert-Paaren, die zusätzliche kontextbezogene Informationen zu den Daten enthalten.

AWS KMS verwendet den Verschlüsselungskontext als [zusätzliche authentifizierte Daten](#), um die [authentifizierte Verschlüsselung](#) zu unterstützen. Wenn Sie einen Verschlüsselungskontext in eine Anforderung zur Verschlüsselung von Daten aufnehmen, AWS KMS bindet den Verschlüsselungskontext an die verschlüsselten Daten. Zur Entschlüsselung von Daten müssen Sie denselben Verschlüsselungskontext in der Anfrage übergeben.

Amazon Location Service-Verschlüsselungskontext

Amazon Location verwendet in allen AWS KMS kryptografischen Operationen denselben Verschlüsselungskontext, wobei der Schlüssel `aws:geo:arn` und der Wert der [Amazon-Ressourcename](#) (ARN) der Ressource ist.

Example

```
"encryptionContext": {
  "aws:geo:arn": "arn:aws:geo:us-west-2:111122223333:geofence-collection/SAMPLE-GeofenceCollection"
}
```

Verwenden des Verschlüsselungskontexts für die Überwachung

Wenn Sie einen symmetrischen kundenverwalteten Schlüssel verwenden, um Ihren Tracker oder Ihre Geofence-Sammlung zu verschlüsseln, können Sie den Verschlüsselungskontext auch in Prüfungsdatensätzen und Protokollen verwenden, um zu identifizieren, wie der vom Kunden verwaltete Schlüssel verwendet wird. Der Verschlüsselungskontext wird auch in [Protokollen angezeigt, die von AWS CloudTrail oder Amazon CloudWatch Logs generiert werden](#).

Verwendung des Verschlüsselungskontextes zur Steuerung des Zugriffs auf den vom Kunden verwalteten Schlüssel

Sie können den Verschlüsselungskontext in Schlüsselrichtlinien und IAM-Richtlinien als `conditions` verwenden, um den Zugriff auf Ihren symmetrischen, kundenverwalteten Schlüssel zu kontrollieren. Sie können Verschlüsselungskontext-Einschränkungen auch in einer Genehmigung verwenden.

Amazon Location verwendet eine Einschränkung des Verschlüsselungskontexts in Erteilungen, um den Zugriff auf den vom Kunden verwalteten Schlüssel in Ihrem Konto oder Ihrer Region zu steuern. Eine Genehmigungseinschränkung erfordert, dass durch die Genehmigung ermöglichte Vorgänge den angegebenen Verschlüsselungskontext verwenden.

Example

Im Folgenden finden Sie Beispiele für Schlüsselrichtlinienanweisungen zur Gewährung des Zugriffs auf einen vom Kunden verwalteten Schlüssel für einen bestimmten Verschlüsselungskontext. Die Bedingung in dieser Richtlinienanweisung setzt voraus, dass die Genehmigungen eine Einschränkung des Verschlüsselungskontextes haben, die den Verschlüsselungskontext spezifiziert.

```
{
  "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"
    }
  }
}
}

```

Überwachen Ihrer Verschlüsselungsschlüssel für Amazon Location Service

Wenn Sie einen vom AWS KMS Kunden verwalteten Schlüssel mit Ihren Amazon Location Service-Ressourcen verwenden, können Sie [AWS CloudTrail](#) oder [Amazon CloudWatch Logs](#) verwenden, um Anforderungen zu verfolgen, die Amazon Location an sendet AWS KMS.

Die folgenden Beispiele sind AWS CloudTrail Ereignisse für CreateGrant, GenerateDataKeyWithoutPlainText, und zur Überwachung von KMS-DecryptOperationen, DescribeKey die von Amazon Location aufgerufen werden, um auf Daten zuzugreifen, die mit Ihrem vom Kunden verwalteten Schlüssel verschlüsselt wurden:

CreateGrant

Wenn Sie einen vom AWS KMS Kunden verwalteten Schlüssel verwenden, um Ihre Tracker- oder Geofence-Erfassungsressourcen zu verschlüsseln, sendet Amazon Location in Ihrem Namen eine CreateGrant Anforderung für den Zugriff auf den KMS-Schlüssel in Ihrem AWS Konto. Die Erteilung, die Amazon Location erstellt, ist spezifisch für die Ressource, die dem vom AWS KMS Kunden verwalteten Schlüssel zugeordnet ist. Darüber hinaus verwendet Amazon Location die -RetireGrantOperation, um eine Erteilung zu entfernen, wenn Sie eine Ressource löschen.

Das folgende Beispiereignis zeichnet den Vorgang CreateGrant auf:

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE: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

Wenn Sie einen vom AWS KMS Kunden verwalteten Schlüssel für Ihre Tracker- oder Geofence-Erfassungsressource aktivieren, erstellt Amazon Location einen eindeutigen Tabellenschlüssel. Es sendet eine `GenerateDataKeyWithoutPlainText` Anforderung an AWS KMS, die den vom AWS KMS Kunden verwalteten Schlüssel für die Ressource angibt.

Das folgende Beispiereignis zeichnet den Vorgang `GenerateDataKeyWithoutPlainText` auf:

```

{
  "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": "57f5dbec-16da-413e-979f-2c4c6663475e"
}

```

Decrypt

Wenn Sie auf einen verschlüsselten Tracker oder eine Geofence-Sammlung zugreifen, ruft Amazon Location die Decrypt Operation auf, um den gespeicherten verschlüsselten Datenschlüssel für den Zugriff auf die verschlüsselten Daten zu verwenden.

Das folgende Beispiereignis zeichnet den Vorgang Decrypt auf:

```

{
  "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 verwendet die `-DescribeKeyOperation`, um zu überprüfen, ob der AWS KMS kundenverwaltete Schlüssel, der Ihrem Tracker oder Ihrer Geofence-Sammlung zugeordnet ist, im Konto und in der Region vorhanden ist.

Das folgende Beispiereignis zeichnet den Vorgang `DescribeKey` auf:

```

{
  "eventVersion": "1.08",
  "userIdentity": {

```

```
"type": "AssumedRole",
"principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
"arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
"accountId": "111122223333",
"accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
"sessionContext": {
  "sessionIssuer": {
    "type": "Role",
    "principalId": "AROAIQDTESTANDEXAMPLE: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"  
}
```

Weitere Informationen

Die folgenden Ressourcen enthalten weitere Informationen zur Datenverschlüsselung im Ruhezustand:

- Weitere Informationen zu grundlegenden [AWS Key Management Service -Konzepten](#) finden Sie im AWS Key Management Service -Entwicklerhandbuch.
- Weitere Informationen zu [bewährten Methoden für die Sicherheit AWS Key Management Service](#) für finden Sie im AWS Key Management Service -Entwicklerhandbuch.

Datenverschlüsselung während der Übertragung für Amazon Location Service

Amazon Location schützt Daten während der Übertragung, während sie zum und vom Service gelangen, indem es automatisch alle Daten zwischen Netzwerken mit dem Transport Layer Security (TLS) 1.2-Verschlüsselungsprotokoll verschlüsselt. Direkte HTTPS-Anfragen, die an die Amazon Location Service-APIs gesendet werden, werden mit dem [AWS Signature Version 4-Algorithmus signiert](#), um eine sichere Verbindung herzustellen.

Identity and Access Management für Amazon Location Service

AWS Identity and Access Management (IAM) ist ein AWS-Service , mit dem ein Administrator den Zugriff auf - AWS Ressourcen sicher steuern kann. IAM-Administratoren steuern, wer für die Nutzung von Amazon Location-Ressourcen authentifiziert (angemeldet) und autorisiert (im Besitz von Berechtigungen) werden kann. IAM ist ein AWS-Service , den Sie ohne zusätzliche Kosten verwenden können.

Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)

- [Funktionsweise von Amazon Location Service mit IAM](#)
- [Funktionsweise von Amazon Location Service mit nicht authentifizierten Benutzern](#)
- [Beispiele für identitätsbasierte Richtlinien für Amazon Location Service](#)
- [Fehlerbehebung für Amazon Location Service – Identität und Zugriff](#)

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, unterscheidet sich je nach Ihrer Arbeit an Amazon Location.

Service-Benutzer – Wenn Sie den Amazon-Location-Service zur Ausführung von Aufgaben verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen bereit, die Sie benötigen. Wenn Sie für Ihre Arbeit weitere Amazon Location-Funktionen ausführen, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Wenn Sie auf ein Feature in Amazon Location nicht zugreifen können, finden Sie weitere Informationen unter [Fehlerbehebung für Amazon Location Service – Identität und Zugriff](#).

Service-Administrator – Wenn Sie in Ihrem Unternehmen für die Amazon Location-Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollständigen Zugriff auf Amazon Location. Ihre Aufgabe besteht darin, zu bestimmen, auf welche Amazon-Location-Funktionen und -Ressourcen Ihre Service-Benutzer zugreifen sollen. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen dazu, wie Ihr Unternehmen IAM mit Amazon Location verwenden kann, finden Sie unter [Funktionsweise von Amazon Location Service mit IAM](#).

IAM-Administrator – Wenn Sie als IAM-Administrator fungieren, sollten Sie Einzelheiten dazu kennen, wie Sie Richtlinien zur Verwaltung des Zugriffs auf Amazon Location verfassen können. Beispiele für identitätsbasierte Amazon Location-Richtlinien, die Sie in IAM verwenden können, finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Location Service](#).

Authentifizierung mit Identitäten

Die Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsanmeldeinformationen bei anmelden. Sie müssen als Root-Benutzer des AWS-Kontos, als IAM-Benutzer oder durch Übernahme einer IAM-Rolle authentifiziert (bei angemeldet AWS) sein.

Sie können sich bei AWS als Verbundidentität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt werden. AWS IAM Identity Center (IAM Identity Center)-Benutzer, die Single-Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für Verbundidentitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie AWS über einen Verbund auf zugreifen, übernehmen Sie indirekt eine Rolle.

Je nachdem, um welchen Benutzertyp es sich handelt, können Sie sich bei der AWS Management Console oder im - AWS Zugriffsportal anmelden. Weitere Informationen zur Anmeldung bei AWS finden Sie unter [So melden Sie sich bei Ihrem an AWS-Konto](#) im AWS-Anmeldung - Benutzerhandbuch.

Wenn Sie AWS programmgesteuert auf zugreifen, AWS stellt ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (Command Line Interface, CLI) bereit, um Ihre Anforderungen mithilfe Ihrer Anmeldeinformationen kryptografisch zu signieren. Wenn Sie keine AWS Tools verwenden, müssen Sie Anforderungen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode zum eigenständigen Signieren von Anforderungen finden Sie unter [Signieren von AWS API-Anforderungen](#) im IAM-Benutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. empfiehlt beispielsweise, AWS Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung](#) im AWS IAM Identity Center - Benutzerhandbuch und [Verwenden der Multi-Faktor-Authentifizierung \(MFA\) in AWS](#) im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie ein erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services und Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet und Sie melden sich mit der E-Mail-Adresse und dem Passwort an, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter [Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

Verbundidentität

Fordern Sie als bewährte Methode menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, auf, den Verbund mit einem Identitätsanbieter zu verwenden, um AWS-Services mithilfe temporärer Anmeldeinformationen auf zuzugreifen.

Eine Verbundidentität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, ein Web-Identitätsanbieter, die AWS Directory Service, das Identity-Center-Verzeichnis oder jeder Benutzer, der mit AWS-Services Anmeldeinformationen auf zugreift, die über eine Identitätsquelle bereitgestellt werden. Wenn Verbundidentitäten auf zugreifen AWS-Konten, übernehmen sie Rollen und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen oder eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und synchronisieren, um sie für alle Ihre AWS-Konten und Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter [Was ist IAM Identity Center?](#) im AWS IAM Identity Center -Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein [IAM-Benutzer](#) ist eine Identität in Ihrem AWS-Konto mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter [Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

Eine [IAM-Gruppe](#) ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter [Erstellen eines IAM-Benutzers \(anstatt einer Rolle\)](#) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität in Ihrem AWS-Konto mit bestimmten Berechtigungen. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der übernehmen, AWS Management Console indem Sie die [Rollen wechseln](#). Sie können eine Rolle übernehmen, indem Sie eine AWS CLI - oder AWS -API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter [Verwenden von IAM-Rollen](#) im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- **Verbundbenutzerzugriff** – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter [Erstellen von Rollen für externe Identitätsanbieter](#) im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter [Berechtigungssätze](#) im AWS IAM Identity Center -Benutzerhandbuch.
- **Temporäre IAM-Benutzerberechtigungen** – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- **Kontoübergreifender Zugriff** – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können AWS-Services Sie jedoch eine Richtlinie direkt an eine Ressource anfügen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien](#) im IAM-Benutzerhandbuch.
- **Serviceübergreifender Zugriff** – Einige AWS-Services verwenden Funktionen in anderen AWS-Services. Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon EC2 aus oder speichert Objekte in Amazon S3. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
 - **Forward Access Sessions (FAS)** – Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen in auszuführen AWS, gelten Sie als Prinzipal. Bei einigen Services

könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service auslösen. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anforderung AWS-Service , Anfragen an nachgelagerte Services zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Service eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder -Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).

- **Servicerolle:** Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.
- **Serviceverknüpfte Rolle** – Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einem verknüpft ist AWS-Service. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem angezeigt AWS-Konto und gehören dem Service. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.
- **Anwendungen, die auf Amazon EC2 ausgeführt werden** – Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und - AWS CLI oder AWS -API-Anforderungen stellen. Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Um einer EC2-Instance eine - AWS Rolle zuzuweisen und sie für alle ihre Anwendungen verfügbar zu machen, erstellen Sie ein Instance-Profil, das der Instance zugeordnet ist. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter [Verwenden einer IAM-Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden](#) im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter [Erstellen einer IAM-Rolle \(anstatt eines Benutzers\)](#) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie steuern den Zugriff in , AWS indem Sie Richtlinien erstellen und sie an AWS Identitäten oder Ressourcen anfügen. Eine Richtlinie ist ein Objekt in , AWS das, wenn es einer Identität oder Ressource zugeordnet wird, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anforderung stellt. Berechtigungen

in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden in AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

Administratoren können AWS JSON-Richtlinien verwenden, um anzugeben, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen über die AWS Management Console, die AWS CLI oder die AWS -API abrufen.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem anfügen können AWS-Konto. Verwaltete Richtlinien umfassen - AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter [Auswahl zwischen verwalteten und eingebundenen Richtlinien](#) im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und

Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder umfassen AWS-Services.

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Services AWS WAF, die ACLs unterstützen. Weitere Informationen zu ACLs finden Sie unter [Zugriffssteuerungsliste \(ACL\) – Übersicht](#) (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger häufig verwendete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- **Berechtigungsgrenzen** – Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im IAM-Benutzerhandbuch.
- **Service-Kontrollrichtlinien (SCPs)** – SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in angeben AWS Organizations. AWS Organizations ist ein Service zum Gruppieren und zentralen Verwalten mehrerer AWS-Konten, die Ihrem Unternehmen gehören. Wenn Sie innerhalb einer Organisation

alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. Die SCP beschränkt Berechtigungen für Entitäten in Mitgliedskonten, einschließlich jeder Root-Benutzer des AWS-Kontos. Weitere Informationen zu Organizations und SCPs finden Sie unter [Funktionsweise von SCPs](#) im AWS Organizations -Benutzerhandbuch.

- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Wie AWS bestimmt, ob eine Anforderung zugelassen werden soll, wenn mehrere Richtlinientypen beteiligt sind, erfahren Sie unter [Logik der Richtlinienbewertung](#) im IAM-Benutzerhandbuch.

Funktionsweise von Amazon Location Service mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf Amazon Location zu verwalten, erfahren Sie, welche IAM-Funktionen Sie mit Amazon Location verwenden können.

IAM-Funktionen, die Sie mit Amazon Location Service verwenden können

IAM-Feature	Amazon Location-Unterstützung
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Richtlinienbedingungsschlüssel (servicespezifisch)	Ja

IAM-Feature	Amazon Location-Unterstützung
ACLs	Nein
ABAC (Tags in Richtlinien)	Ja
Temporäre Anmeldeinformationen	Ja
Hauptberechtigungen	Nein
Servicerollen	Nein
Serviceverknüpfte Rollen	Nein

Einen Überblick über das Zusammenwirken von Amazon Location und anderen - AWS Services mit den meisten IAM-Funktionen finden Sie unter [-AWS Services, die mit IAM funktionieren](#) im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien für Amazon Location

Unterstützt Richtlinien auf Identitätsbasis.	Ja
--	----

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für Amazon Location

Beispiele für identitätsbasierte Amazon Location-Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Location Service](#).

Ressourcenbasierte Richtlinien in Amazon Location

Unterstützt ressourcenbasierte Richtlinien	Nein
--	------

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Prinzipale können Konten, Benutzer, Rollen, Verbundbenutzer oder umfassen AWS-Services.

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource in unterschiedlichen befinden AWS-Konten, muss ein IAM-Administrator im vertrauenswürdigen Konto der Prinzipal-Entität (Benutzer oder Rolle) auch die Berechtigung für den Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter [Wie sich IAM-Rollen von ressourcenbasierten Richtlinien unterscheiden](#) im IAM-Benutzerhandbuch.

Richtlinienaktionen für Amazon Location

Unterstützt Richtlinienaktionen	Ja
---------------------------------	----

Administratoren können AWS JSON-Richtlinien verwenden, um anzugeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie die zugehörige AWS API-Operation. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der Amazon Location-Aktionen finden Sie unter [Von Amazon Location Service definierte Aktionen](#) in der Service Authorization Reference.

Richtlinienaktionen in Amazon Location verwenden das folgende Präfix vor der Aktion:

```
geo
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

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

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Get` beginnen, einschließlich der folgenden Aktion:

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

Beispiele für identitätsbasierte Amazon Location-Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Location Service](#).

Richtlinienressourcen für Amazon Location

Unterstützt Richtlinienressourcen	Ja
-----------------------------------	----

Administratoren können AWS JSON-Richtlinien verwenden, um anzugeben, wer Zugriff auf was hat. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein `Resource` oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen \(ARN\)](#) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"

```

Eine Liste der Amazon Location-Ressourcentypen und ihrer ARNs finden Sie unter [Von Amazon Location Service definierte Ressourcen](#) in der Service Authorization Reference. Informationen zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter [Von Amazon Location Service definierte Aktionen](#).

Beispiele für identitätsbasierte Amazon Location-Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Location Service](#).

Richtlinienbedingungsschlüssel für Amazon Location

Unterstützt servicespezifische Richtlinienbedingungsschlüssel	Ja
---	----

Administratoren können AWS JSON-Richtlinien verwenden, um anzugeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition block`) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#) verwenden, z. B. `ist gleich oder kleiner als`, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen AND-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter [IAM-Richtlinienelemente: Variablen und Tags](#) im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und servicespezifische Bedingungsschlüssel. Informationen zum Anzeigen aller AWS globalen Bedingungsschlüssel finden Sie unter [AWS Globale Bedingungskontextschlüssel](#) im IAM-Benutzerhandbuch.

Eine Liste der Bedingungsschlüssel für Amazon Location finden Sie unter [Bedingungsschlüssel für Amazon Location Service](#) in der Service-Autorisierungs-Referenz. Informationen dazu, mit welchen Aktionen und Ressourcen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Von Amazon Location Service definierte Aktionen](#).

Amazon Location unterstützt Bedingungsschlüssel, mit denen Sie den Zugriff auf bestimmte Geofences oder Geräte in Ihren Richtlinienanweisungen zulassen oder verweigern können. Die folgenden Bedingungsschlüssel sind verfügbar:

- `geo:GeofenceIds` zur Verwendung mit Geofence-Aktionen. Der Typ ist `ArrayOfString`.
- `geo:DeviceIds` zur Verwendung mit Tracker-Aktionen. Der Typ ist `ArrayOfString`.

Die folgenden Aktionen können mit `geo:GeofenceIds` in Ihrer IAM-Richtlinie verwendet werden:

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

Die folgenden Aktionen können mit `geo:DeviceIds` in Ihrer IAM-Richtlinie verwendet werden:

- `BatchDeleteDevicePositionHistory`

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

Note

Sie können diese Bedingungsschlüssel nicht mit den `ListDevicePosition` Aktionen `BatchEvaluateGeofencesListGeofences`, oder verwenden.

Beispiele für identitätsbasierte Amazon Location-Richtlinien finden Sie unter [Beispiele für identitätsbasierte Richtlinien für Amazon Location Service](#).

ACLs in Amazon Location

Unterstützt ACLs	Nein
------------------	------

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit Amazon Location

Unterstützt ABAC (Tags in Richtlinien)	Ja
--	----

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In werden AWSdiese Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anfügen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungelement einer [Richtlinie Tag-Informationen](#) an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter [Was ist ABAC?](#) im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe [Attributbasierte Zugriffskontrolle \(ABAC\)](#) verwenden im IAM-Benutzerhandbuch.

Weitere Informationen zum Markieren von Amazon Location-Ressourcen finden Sie unter [Markieren Ihrer Amazon Location Service-Ressourcen](#).

Ein Beispiel für eine identitätsbasierte Richtlinie zur Einschränkung des Zugriffs auf eine Ressource auf der Grundlage der Markierungen dieser Ressource finden Sie unter [Steuern des Ressourcenzugriffs basierend auf Tags](#).

Verwenden temporärer Anmeldeinformationen mit Amazon Location

Unterstützt temporäre Anmeldeinformationen	Ja
--	----

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, darunter welche mit temporären Anmeldeinformationen AWS-Services funktionieren, finden Sie unter [AWS-Services , die mit IAM funktionieren](#) im IAM-Benutzerhandbuch.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich AWS Management Console mit einer anderen Methode als einem Benutzernamen und einem Passwort bei der anmelden. Wenn Sie beispielsweise AWS über den SSO-Link (Single Sign-On) Ihres Unternehmens auf zugreifen, erstellt dieser Prozess automatisch temporäre Anmeldeinformationen. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter [Wechseln zu einer Rolle \(Konsole\)](#) im IAM-Benutzerhandbuch.

Sie können temporäre Anmeldeinformationen manuell mit der AWS CLI oder der AWS API erstellen. Sie können diese temporären Anmeldeinformationen dann verwenden, um auf zuzugreifen AWS. AWS empfohlen, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt

langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Sicherheitsanmeldeinformationen in IAM](#).

Serviceübergreifende Prinzipalberechtigungen für Amazon Location

Unterstützt Forward Access Sessions (FAS)	Nein
---	------

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen in auszuführen AWS, gelten Sie als Prinzipal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service auslösen. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anforderung AWS-Service , Anfragen an nachgelagerte Services zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Service eine Anfrage erhält, die Interaktionen mit anderen AWS-Services oder -Ressourcen erfordert. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).

Servicerollen für Amazon Location

Unterstützt Servicerollen	Nein
---------------------------	------

Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.

Warning

Das Ändern der Berechtigungen für eine Servicerolle könnte die Funktionalität von Amazon Location beeinträchtigen. Bearbeiten Sie Servicerollen nur, wenn Amazon Location dazu Anleitungen gibt.

Serviceverknüpfte Rollen für Amazon Location

Unterstützt serviceverknüpfte Rollen	Nein
--------------------------------------	------

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einem verknüpft ist AWS-Service. Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Serviceverknüpfte Rollen werden in Ihrem angezeigt AWS-Konto und gehören dem Service. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter [AWS -Services, die mit IAM funktionieren](#). Suchen Sie in der Tabelle nach einem Service mit einem Yes in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Funktionsweise von Amazon Location Service mit nicht authentifizierten Benutzern

Viele Szenarien für die Verwendung von Amazon Location Service, einschließlich der Anzeige von Karten im Web oder in einer mobilen Anwendung, erfordern, dass Benutzern, die sich nicht mit IAM angemeldet haben, Zugriff gewährt wird. Für diese nicht authentifizierten Szenarien haben Sie zwei Möglichkeiten.

- API-Schlüssel verwenden – Um nicht authentifizierten Benutzern Zugriff zu gewähren, können Sie API-Schlüssel erstellen, die schreibgeschützten Zugriff auf Ihre Amazon Location Service-Ressourcen gewähren. Dies ist in Fällen nützlich, in denen Sie nicht jeden Benutzer authentifizieren möchten. Zum Beispiel eine Webanwendung. Weitere Informationen zu API-Schlüsseln finden Sie unter [Erlauben des nicht authentifizierten Gastzugriffs auf Ihre Anwendung mithilfe von API-Schlüsseln](#).
- Amazon Cognito verwenden – Eine Alternative zu API-Schlüsseln ist die Verwendung von Amazon Cognito, um anonymen Zugriff zu gewähren. Mit Amazon Cognito können Sie eine umfassendere Autorisierung mit IAM-Richtlinie erstellen, um zu definieren, was von nicht authentifizierten Benutzern getan werden kann. Weitere Informationen zur Verwendung von Amazon Cognito finden Sie unter [Erlauben des nicht authentifizierten Gastzugriffs auf Ihre Anwendung mit Amazon Cognito](#).

Eine Übersicht über die Bereitstellung des Zugriffs für nicht authentifizierte Benutzer finden Sie unter [Gewähren des Zugriffs auf Amazon Location Service](#).

Beispiele für identitätsbasierte Richtlinien für Amazon Location Service

Benutzer und Rollen besitzen standardmäßig keine Berechtigungen zum Erstellen oder Ändern von Amazon Location-Ressourcen. Sie können auch keine Aufgaben mithilfe der AWS Management

Console, AWS Command Line Interface (AWS CLI) oder AWS API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Einzelheiten zu Aktionen und Ressourcentypen, die von Amazon Location definiert werden, einschließlich des Formats der ARNs für die einzelnen Ressourcentypen, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für Amazon Location Service](#) in der Service-Autorisierungs-Referenz.

Themen

- [Bewährte Methoden für Richtlinien](#)
- [Verwenden der Amazon Location-Konsole](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)
- [Verwenden von Amazon Location Service-Ressourcen in der Richtlinie](#)
- [Berechtigungen zum Aktualisieren von Gerätepositionen](#)
- [Schreibgeschützte Richtlinie für Tracker-Ressourcen](#)
- [Richtlinie zum Erstellen von Geofences](#)
- [Schreibgeschützte Richtlinie für Geofences](#)
- [Berechtigungen zum Rendern einer Kartenressource](#)
- [Berechtigungen zum Zulassen von Suchvorgängen](#)
- [Schreibgeschützte Richtlinie für Routenrechner](#)
- [Steuern des Ressourcenzugriffs basierend auf Bedingungsschlüsseln](#)
- [Steuern des Ressourcenzugriffs basierend auf Tags](#)

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand Amazon Location-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Erste Schritte mit AWS -verwalteten Richtlinien und Umstellung auf Berechtigungen mit den geringsten Berechtigungen – Um Ihren Benutzern und Workloads Berechtigungen zu erteilen, verwenden Sie die -AWS verwalteten Richtlinien, die Berechtigungen für viele häufige Anwendungsfälle gewähren. Sie sind in Ihrem verfügbar AWS-Konto. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die für Ihre Anwendungsfälle spezifisch sind. Weitere Informationen finden Sie unter [AWS -verwaltete Richtlinien](#) oder [AWS -verwaltete Richtlinien für Auftrags-Funktionen](#) im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten: Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter [Richtlinien und Berechtigungen in IAM](#) im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Service-Aktionen zu gewähren, wenn sie über eine bestimmte verwendet werden AWS-Service, z. B. AWS CloudFormation. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter [Richtlinienvvalidierung zum IAM Access Analyzer](#) im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich – Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter [Konfigurieren eines MFA-geschützten API-Zugriffs](#) im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter [Bewährte Methoden für die Sicherheit in IAM](#) im IAM-Benutzerhandbuch.

Verwenden der Amazon Location-Konsole

Um auf die Amazon Location Service-Konsole zugreifen zu können, müssen Sie über einen Mindestsatz von Berechtigungen verfügen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den Amazon Location-Ressourcen in Ihrem aufzulisten und anzuzeigen AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Für Benutzer, die nur Aufrufe an die AWS CLI oder die AWS API durchführen, müssen Sie keine Mindestberechtigungen für die Konsole erteilen. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen die Amazon Location-Konsole verwenden können, fügen Sie den Entitäten die folgende Richtlinie hinzu. Weitere Informationen finden Sie unter [Hinzufügen von Berechtigungen zu einem Benutzer](#) im IAM-Benutzerhandbuch.

Die folgende Richtlinie gewährt Zugriff auf die Amazon Location Service-Konsole, um Details zu Amazon Location-Ressourcen in Ihrem AWS Konto erstellen, löschen, auflisten und anzeigen zu können.

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

Alternativ können Sie schreibgeschützte Berechtigungen erteilen, um den schreibgeschützten Zugriff zu erleichtern. Bei schreibgeschützten Berechtigungen wird eine Fehlermeldung angezeigt, wenn

der Benutzer versucht, Aktionen wie das Erstellen oder Löschen von Ressourcen zu schreiben. Ein Beispiel finden Sie unter [. the section called "Schreibgeschützte Richtlinie für Tracker"](#)

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie enthält Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der AWS CLI oder AWS API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupForUser",
        "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": "*"
    }
  ]
}
```

Verwenden von Amazon Location Service-Ressourcen in der Richtlinie

Amazon Location Service verwendet die folgenden Präfixe für Ressourcen:

Ressourcenpräfix für Amazon Location

Ressource	Ressourcenpräfix
Zuordnen von Ressourcen	map
Platzieren von Ressourcen	place-index
Routing-Ressourcen	route-calculator
Nachverfolgen von Ressourcen	tracker
Geofence-Sammlungsressourcen	geofence-collection

Verwenden Sie die folgende ARN-Syntax:

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

Weitere Informationen zum Format von ARNs finden Sie unter [Amazon-Ressourcennamen \(ARNs\) und AWS Service-Namespaces](#).

Beispiele

- Verwenden Sie den folgenden ARN, um den Zugriff auf eine angegebene Kartenressource zu erlauben.

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

- Um den Zugriff auf alle map Ressourcen anzugeben, die zu einem bestimmten Konto gehören, verwenden Sie den Platzhalter (*):

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

- Einige Amazon Location-Aktionen, z. B. das Erstellen von Ressourcen, können nicht für eine bestimmte Ressource ausgeführt werden. In diesen Fällen müssen Sie den Platzhalter (*) verwenden.

```
"Resource": "*"

```

Eine Liste der Amazon Location-Ressourcentypen und ihrer ARNs finden Sie unter [Von Amazon Location Service definierte Ressourcen](#) in der Service Authorization Reference . Informationen zu den Aktionen, mit denen Sie den ARN einzelner Ressourcen angeben können, finden Sie unter [Von Amazon Location Service definierte Aktionen](#).

Berechtigungen zum Aktualisieren von Gerätepositionen

Um die Gerätepositionen für mehrere Tracker zu aktualisieren, möchten Sie einem Benutzer Zugriff auf eine oder mehrere Ihrer Tracker-Ressourcen gewähren. Sie möchten dem Benutzer auch erlauben, einen Stapel von Gerätepositionen zu aktualisieren.

In diesem Beispiel gewährt die folgende Richtlinie nicht nur Zugriff auf die *Tracker1*- und *Tracker2*-Ressourcen, sondern auch die Berechtigung, die `geo:BatchUpdateDevicePosition` Aktion für die *Tracker1*- und *Tracker2*-Ressourcen zu verwenden.

```
{
  "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"
      ]
    }
  ]
}
```

Wenn Sie den Benutzer so einschränken möchten, dass er nur Gerätepositionen für ein bestimmtes Gerät aktualisieren kann, können Sie einen Bedingungschlüssel für diese Geräte-ID hinzufügen.

```
{
  "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"
        ]
      }
    }
  }
]
}

```

Schreibgeschützte Richtlinie für Tracker-Ressourcen

Um eine schreibgeschützte Richtlinie für alle Tracker-Ressourcen in Ihrem AWS Konto zu erstellen, müssen Sie Zugriff auf alle Tracker-Ressourcen gewähren. Sie möchten einem Benutzer auch Zugriff auf Aktionen gewähren, mit denen er die Geräteposition für mehrere Geräte abrufen, die Geräteposition von einem einzigen Gerät abrufen und den Positionsverlauf abrufen kann.

In diesem Beispiel gewährt die folgende Richtlinie die Berechtigung für die folgenden Aktionen:

- `geo:BatchGetDevicePosition` , um die Position mehrerer Geräte abzurufen.
- `geo:GetDevicePosition` , um die Position eines einzelnen Geräts abzurufen.
- `geo:GetDevicePositionHistory` , um den Positionsverlauf eines Geräts abzurufen.

```

{
  "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/*"
  }
]
}

```

Richtlinie zum Erstellen von Geofences

Um eine Richtlinie zu erstellen, die es einem Benutzer ermöglicht, Geofences zu erstellen, müssen Sie Zugriff auf bestimmte Aktionen gewähren, die es Benutzern ermöglichen, einen oder mehrere Geofences für eine Geofence-Sammlung zu erstellen.

Die folgende Richtlinie gewährt die Berechtigung für die folgenden Aktionen für *Sammlung* :

- `geo:BatchPutGeofence` , um mehrere Geofences zu erstellen.
- `geo:PutGeofence` , um einen einzelnen Geofence zu erstellen.

```

{
  "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"
    }
  ]
}

```

Schreibgeschützte Richtlinie für Geofences

Um eine schreibgeschützte Richtlinie für Geofences zu erstellen, die in einer Geofence-Sammlung in Ihrem AWS Konto gespeichert sind, müssen Sie Zugriff auf Aktionen gewähren, die aus der Geofence-Sammlung lesen, in der die Geofences gespeichert sind.

Die folgende Richtlinie gewährt die Berechtigung für die folgenden Aktionen für *Sammlung* :

- `geo:ListGeofences` um Geofences in der angegebenen Geofence-Sammlung aufzulisten.
- `geo:GetGeofence` , um einen Geofence aus der Geofence-Sammlung abzurufen.

```
{
  "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"
    }
  ]
}
```

Berechtigungen zum Rendern einer Kartenressource

Um ausreichend Berechtigungen zum Rendern von Karten zu erteilen, müssen Sie Zugriff auf Kartenkacheln, Schreibvorgänge, Glyphs und den Stildeskriptor gewähren:

- `geo:GetMapTile` ruft Kartenkacheln ab, die verwendet werden, um Features auf einer Karte selektiv zu rendern.
- `geo:GetMapSprites` ruft das PNG-Sprite-Blatt und das entsprechende JSON-Dokument ab, das die darin enthaltenen Offsets beschreibt.
- `geo:GetMapGlyphs` ruft die Glyphs ab, die für die Anzeige von Text verwendet werden.
- `geo:GetMapStyleDescriptor` ruft den Stildeskriptor der Karte ab, der Rendering-Regeln enthält.

```
{
  "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"
  }
]
}

```

Berechtigungen zum Zulassen von Suchvorgängen

Um eine Richtlinie zu erstellen, die Suchvorgänge zulässt, müssen Sie zunächst Zugriff auf die Ortsindex-Ressource in Ihrem AWS Konto gewähren. Sie möchten auch Zugriff auf Aktionen gewähren, mit denen der Benutzer anhand von Text durch Geokodierung suchen kann, und mithilfe einer Position durch Reverse-Geokodierung suchen kann.

In diesem Beispiel gewährt die folgende Richtlinie nicht nur Zugriff auf *PlaceIndex*, sondern auch die Berechtigung für die folgenden Aktionen:

- `geo:SearchPlaceIndexForPosition` ermöglicht Ihnen die Suche nach Orten oder Interessenpunkten in der Nähe einer bestimmten Position.
- `geo:SearchPlaceIndexForText` ermöglicht Ihnen die Suche nach einer Adresse, einem Namen, einer Stadt oder einer Region mithilfe von Freiformtext.

```

{
  "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"
    }
  ]
}

```

```
}
```

Schreibgeschützte Richtlinie für Routenrechner

Sie können eine schreibgeschützte Richtlinie erstellen, um einem Benutzer den Zugriff auf eine Routenrechner-Ressource zum Berechnen einer Route zu ermöglichen.

In diesem Beispiel gewährt die folgende Richtlinie nicht nur Zugriff auf *ExampleCalculator*, sondern auch die Berechtigung für die folgende Operation:

- `geo:CalculateRoute` berechnet eine Route anhand einer Ausgangsposition, einer Zielposition und einer Liste von Wegpunktpositionen.

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

Steuern des Ressourcenzugriffs basierend auf Bedingungsschlüsseln

Wenn Sie eine IAM-Richtlinie erstellen, um Zugriff auf die Verwendung von Geofences oder Gerätepositionen zu gewähren, können Sie [Bedingungsoperatoren](#) verwenden, um genauer zu steuern, auf welche Geofences oder Geräte ein Benutzer zugreifen kann. Sie können dies tun, indem Sie die Geofence-ID oder die Geräte-ID in das `-ConditionElement` Ihrer Richtlinie aufnehmen.

Die folgende Beispielrichtlinie zeigt, wie Sie eine Richtlinie erstellen können, die es einem Benutzer ermöglicht, Gerätepositionen für ein bestimmtes Gerät zu aktualisieren.

```
{
  "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"
            ]
        }
    }
}

```

Steuern des Ressourcenzugriffs basierend auf Tags

Wenn Sie eine IAM-Richtlinie erstellen, um Zugriff auf Ihre Amazon [Location-Ressourcen zu gewähren, können Sie die attributbasierte Zugriffskontrolle](#) verwenden, um besser kontrollieren zu können, welche Ressourcen ein Benutzer ändern, verwenden oder löschen kann. Sie können dies tun, indem Sie Tag-Informationen in das `-ConditionElement` Ihrer Richtlinie aufnehmen, um den Zugriff basierend auf Ihren Ressourcen-[Tags](#) zu steuern.

Die folgende Beispielrichtlinie zeigt, wie Sie eine Richtlinie erstellen können, die es einem Benutzer ermöglicht, Geofences zu erstellen. Dadurch wird die Berechtigung für die folgenden Aktionen erteilt, einen oder mehrere Geofences für eine Geofence-Sammlung namens *Sammlung zu* erstellen:

- `geo:BatchPutGeofence` , um mehrere Geofences zu erstellen.
- `geo:PutGeofence` , um einen einzelnen Geofence zu erstellen.

Diese Richtlinie verwendet jedoch das `-ConditionElement`, um die Berechtigung nur zu erteilen, wenn das *Sammlungs*-Tag den Wert des Benutzernamens dieses Benutzers `Owner` hat.

- Wenn ein Benutzer mit dem Namen beispielsweise `richard-roe` versucht, eine Amazon Location *Collection* anzuzeigen, muss die *Sammlung* mit dem Tag `Owner=richard-roe` oder `seowner=richard-roe` versehen sein. Andernfalls wird dem Benutzer der Zugriff verweigert.

Note

Der Tag-Schlüssel `Owner` der Bedingung stimmt sowohl mit `Owner` als auch mit `owner` überein, da die Namen von Bedingungsschlüsseln nicht zwischen Groß- und Kleinschreibung unterscheiden. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.

```
{
  "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}"}
      }
    }
  ]
}
```

Ein Tutorial [zum Definieren von Berechtigungen für den Zugriff auf AWS-Ressourcen basierend auf Tags](#) finden Sie im AWS Identity and Access Management -Benutzerhandbuch.

Fehlerbehebung für Amazon Location Service – Identität und Zugriff

Verwenden Sie die folgenden Informationen, um häufige Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit Amazon Location und IAM auftreten können.

Themen

- [Ich bin nicht autorisiert, eine Aktion in Amazon Location auszuführen](#)
- [Ich bin nicht autorisiert, iam durchzuführen:PassRole](#)
- [Ich möchte Personen außerhalb meines AWS-Konto Zugriff auf meine Amazon Location-Ressourcen gewähren](#)

Ich bin nicht autorisiert, eine Aktion in Amazon Location auszuführen

Wenn Sie eine Fehlermeldung erhalten, dass Sie nicht zur Durchführung einer Aktion berechtigt sind, müssen Ihre Richtlinien aktualisiert werden, damit Sie die Aktion durchführen können.

Der folgende Beispielfehler tritt auf, wenn der IAM-Benutzer `mateojackson` versucht, über die Konsole Details zu einer fiktiven `my-example-widget`-Ressource anzuzeigen, jedoch nicht über `geo: GetWidget`-Berechtigungen verfügt.

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

In diesem Fall muss die Richtlinie für den Benutzer `mateojackson` aktualisiert werden, damit er mit der `geo: GetWidget`-Aktion auf die `my-example-widget`-Ressource zugreifen kann.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich bin nicht autorisiert, iam durchzuführen:PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zum Ausführen der `iam:PassRole` Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an Amazon Location übergeben zu können.

Einige AWS-Services ermöglichen es Ihnen, eine vorhandene Rolle an diesen Service zu übergeben, anstatt eine neue Servicerolle oder serviceverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in Amazon Location auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

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

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb meines AWS-Konto Zugriff auf meine Amazon Location-Ressourcen gewähren

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Services, die ressourcenbasierte Richtlinien oder Zugriffssteuerungslisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen dazu, ob Amazon Location diese Funktionen unterstützt, finden Sie unter [Funktionsweise von Amazon Location Service mit IAM](#).
- Informationen zum Gewähren des Zugriffs auf Ihre AWS-Konten -Ressourcen in Ihrem Besitz finden Sie unter [Gewähren des Zugriffs für einen IAM-Benutzer in einem anderen AWS-Konto, das Sie besitzen](#) im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre -Ressourcen gewähren AWS-Konten, finden Sie unter [Gewähren von Zugriff auf im AWS-Konten Besitz von Dritten](#) im IAM-Benutzerhandbuch.
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter [Gewähren von Zugriff für extern authentifizierte Benutzer \(Identitätsverbund\)](#) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien](#) im IAM-Benutzerhandbuch.

Reaktion auf Vorfälle in Amazon Location Service

Sicherheit hat bei höchste Priorität AWS. Im Rahmen des AWS Cloud-[Modells der übergreifenden Verantwortlichkeit](#) AWS verwaltet ein Rechenzentrum und eine Netzwerkarchitektur, die die Anforderungen der sicherheitssensibelsten Organisationen erfüllt. Als - AWS Kunde haben Sie eine gemeinsame Verantwortung für die Aufrechterhaltung der Sicherheit in der Cloud. Das bedeutet, dass Sie die Sicherheit steuern können, die Sie über die AWS Tools und Funktionen implementieren möchten, auf die Sie Zugriff haben.

Indem Sie eine Sicherheitsgrundlage einrichten, die die Ziele für Ihre Anwendungen erfüllt, die in der Cloud ausgeführt werden, können Sie Abweichungen erkennen, auf die Sie reagieren können. Da die Reaktion auf Sicherheitsvorfälle ein komplexes Thema sein kann, empfehlen wir Ihnen, die folgenden

Ressourcen zu lesen, damit Sie die Auswirkungen der Reaktion auf Vorfälle (IR) und Ihrer Auswahl auf Ihre Unternehmensziele besser verstehen können: [AWS Security Incident Response Guide](#), [AWS Security Best Practices](#) Whitepaper und [AWS Cloud Adoption Framework \(AWS CAF\)](#).

Protokollierung und Überwachung in Amazon Location Service

Protokollierung und Überwachung sind ein wichtiger Bestandteil der Reaktion auf Vorfälle. Auf diese Weise können Sie eine Sicherheitsbasis einrichten, um Abweichungen zu erkennen, die Sie untersuchen und darauf reagieren können. Durch die Implementierung von Protokollierung und Überwachung für Amazon Location Service können Sie die Zuverlässigkeit, Verfügbarkeit und Leistung Ihrer Projekte und Ressourcen aufrechterhalten.

AWS bietet mehrere Tools, mit denen Sie Daten zur Reaktion auf Vorfälle protokollieren und sammeln können:

AWS CloudTrail

Amazon Location Service ist in integriert AWS CloudTrail. Dabei handelt es sich um einen Service, der die Aktionen eines Benutzers, einer Rolle oder eines AWS Services aufzeichnet. Dazu gehören Aktionen von der Amazon Location Service-Konsole und programmgesteuerte Aufrufe von Amazon Location API-Operationen. Diese Aktionsdatensätze werden als Ereignisse bezeichnet. Weitere Informationen finden Sie unter [Protokollieren und Überwachen von Amazon Location Service mit AWS CloudTrail](#).

Amazon CloudWatch

Sie können Amazon verwenden CloudWatch , um Metriken zu Ihrem Amazon Location Service-Konto zu erfassen und zu analysieren. Sie können CloudWatch Alarme aktivieren, um Sie zu benachrichtigen, wenn eine Metrik bestimmte Bedingungen erfüllt und einen bestimmten Schwellenwert erreicht hat. Wenn Sie einen Alarm erstellen, CloudWatch sendet eine Benachrichtigung an einen von Ihnen definierten Amazon Simple Notification Service. Weitere Informationen finden Sie unter [Überwachen von Amazon Location Service mit Amazon CloudWatch](#).

AWS Health Dashboards

Mit [AWS Health Dashboards](#) können Sie den Status des Amazon Location Service-Services überprüfen. Sie können auch historische Daten zu Ereignissen oder Problemen überwachen und anzeigen, die sich auf Ihre AWS Umgebung auswirken könnten. Weitere Informationen finden Sie im [AWS Health -Benutzerhandbuch](#).

Compliance-Validierung für Amazon Location Service

Informationen darüber, ob ein in den Geltungsbereich bestimmter Compliance-Programme AWS-Service fällt, finden Sie [AWS-Services unter im Geltungsbereich nach Compliance-Programm](#) und wählen Sie das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter [AWS Compliance-Programme](#)

Sie können Auditberichte von Drittanbietern mit herunterladen AWS Artifact. Weitere Informationen finden Sie unter [Herunterladen von Berichten unter AWS Artifact](#) .

Ihre Compliance-Verantwortung bei der Verwendung von AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Unterstützung der Compliance bereit:

- [Schnellstartanleitungen für Sicherheit und Compliance](#) – In diesen Bereitstellungsleitfäden werden Überlegungen zur Architektur erörtert und Schritte für die Bereitstellung von Basisumgebungen in bereitgestellt AWS , die sich auf Sicherheit und Compliance konzentrieren.
- [Architekturerstellung für HIPAA-Sicherheit und -Compliance in Amazon Web Services](#) – In diesem Whitepaper wird beschrieben, wie Unternehmen mithilfe AWS von HIPAA-berechtigte Anwendungen erstellen können.

Note

Nicht alle AWS-Services sind HIPAA-fähig. Weitere Informationen finden Sie in der [Referenz für HIPAA-berechtigte Services](#).

- [AWS Compliance-Ressourcen](#) – Diese Sammlung von Arbeitsmappen und Leitfäden kann für Ihre Branche und Ihren Standort gelten.
- [AWS Kunden-Compliance-Leitfäden](#) – Verstehen Sie das Modell der geteilten Verantwortung anhand der Compliance. Die Leitfäden fassen die bewährten Methoden zur Sicherung zusammen AWS-Services und ordnen die Leitlinien den Sicherheitskontrollen in mehreren Frameworks zu (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Officer (PCI) und der International Organization for Standardization (ISO)).
- [Bewertung von Ressourcen mit Regeln](#) im -AWS Config Entwicklerhandbuch – Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.

- [AWS Security Hub](#) – Dies AWS-Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus in AWS. Security Hub verwendet Sicherheitskontrollen, um Ihre AWS - Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der [Security-Hub-Steuerungsreferenz](#).
- [AWS Audit Manager](#) – Auf diese AWS-Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um den Umgang mit Risiken und die Einhaltung von Branchenstandards zu vereinfachen.

Ausfallsicherheit bei Amazon Location Service

Die AWS globale -Infrastruktur ist um AWS-Regionen und Availability Zones herum aufgebaut. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die mit einem Netzwerk mit niedriger Latenz, hohem Durchsatz und hoher Redundanz verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu AWS-Regionen und Availability Zones finden Sie unter [AWS Globale Infrastruktur](#).

Neben der AWS globalen -Infrastruktur stellt Amazon Location verschiedene Funktionen bereit, um Ihren Anforderungen an Ausfallsicherheit und Datensicherung gerecht zu werden.

Infrastruktursicherheit in Amazon Location Service

Als verwalteter Service ist Amazon Location Service durch die AWS globale Netzwerksicherheit von geschützt. Informationen zu AWS Sicherheitsservices und wie die Infrastruktur AWS schützt, finden Sie unter [AWS Cloud-Sicherheit](#). Informationen zum Entwerfen Ihrer AWS Umgebung mit den bewährten Methoden für die Infrastruktursicherheit finden Sie unter [Infrastrukturschutz](#) in Security Pillar AWS Well-Architected Framework.

Sie verwenden durch AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf Amazon Location zuzugreifen. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.

- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit [AWS Security Token Service](#) (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Konfigurations- und Schwachstellenanalyse in Amazon Location

Konfigurations- und IT-Kontrollen sind eine übergreifende Verantwortlichkeit zwischen AWS und Ihnen, unserem Kunden. Weitere Informationen finden Sie im AWS [Modell der geteilten Verantwortung](#).

Serviceübergreifende Confused-Deputy-Prävention

Das Problem des verwirrten Stellvertreters ist ein Sicherheitsproblem, bei dem eine Entität, die keine Berechtigung zur Durchführung einer Aktion hat, eine privilegiertere Entität zur Durchführung der Aktion zwingen kann. In kann ein AWSserviceübergreifender Identitätswechsel zu dem Problem des verwirrten Stellvertreters führen. Ein dienstübergreifender Identitätswechsel kann auftreten, wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der aufrufende Service kann manipuliert werden, um seine Berechtigungen zu verwenden, um Aktionen auf die Ressourcen eines anderen Kunden auszuführen, für die er sonst keine Zugriffsberechtigung haben sollte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Amazon Location Service fungiert nicht als aufrufender Service in Ihrem Namen für andere - AWS Services, daher müssen Sie diese Schutzmaßnahmen in diesem Fall nicht hinzufügen. Weitere Informationen zum verwirrten Stellvertreter finden Sie unter [Das Problem des verwirrten Stellvertreters](#) im AWS Identity and Access Management -Benutzerhandbuch.

Bewährte Sicherheitsmethoden für Amazon Location Service

Amazon Location Service bietet eine Reihe von Sicherheitsfunktionen, die Sie bei der Entwicklung und Implementierung Ihrer eigenen Sicherheitsrichtlinien berücksichtigen sollten. Die folgenden

bewährten Methoden sind allgemeine Richtlinien und keine vollständige Sicherheitslösung. Da diese bewährten Methoden für Ihre Umgebung möglicherweise nicht angemessen oder ausreichend sind, sollten Sie sie als hilfreiche Überlegungen und nicht als bindend ansehen.

Bewährte Methoden für die Sicherheit in Detective für Amazon Location Service

Die folgenden bewährten Methoden für Amazon Location Service können dabei helfen, Sicherheitsvorfälle zu erkennen:

AWS-Überwachungstools implementieren

Die Überwachung ist für die Reaktion auf Vorfälle von entscheidender Bedeutung und gewährleistet die Zuverlässigkeit und Sicherheit der Ressourcen von Amazon Location Service und Ihrer -Lösungen. Sie können Überwachungstools aus den verschiedenen Tools und Services implementieren, die über AWS verfügbar sind, um Ihre Ressourcen und Ihre anderen AWS-Services zu überwachen.

Mit Amazon CloudWatch können Sie beispielsweise Metriken für Amazon Location Service überwachen und Alarme einrichten, um Sie zu benachrichtigen, wenn eine Metrik bestimmte Bedingungen erfüllt, die Sie festgelegt haben, und einen von Ihnen definierten Schwellenwert erreicht hat. Wenn Sie einen Alarm erstellen, können Sie mit Amazon Simple Notification Service festlegen CloudWatch , dass eine Benachrichtigung gesendet wird. Weitere Informationen finden Sie unter [the section called “Protokollieren und Überwachen”](#).

Aktivieren von AWS-Protokollierungstools

Die Protokollierung bietet eine Aufzeichnung der von einem Benutzer, einer Rolle oder einem AWS-Service in Amazon Location Service durchgeführten Aktionen. Sie können Protokollierungstools wie implementieren AWS CloudTrail , um Daten zu Aktionen zu sammeln, um ungewöhnliche API-Aktivitäten zu erkennen.

Wenn Sie einen Trail erstellen, können Sie so konfigurieren, CloudTrail dass Ereignisse protokolliert werden. Ereignisse sind Datensätze von Ressourcenoperationen, die an oder innerhalb einer Ressource ausgeführt werden, z. B. die an Amazon Location gestellte Anfrage, die IP-Adresse, von der die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann die Anfrage gestellt wurde, sowie zusätzliche Daten. Weitere Informationen finden Sie unter [Protokollieren von Datenereignissen für Trails](#) im AWS CloudTrail -Benutzerhandbuch.

Bewährte Methoden für präventive Sicherheit für Amazon Location Service

Die folgenden bewährten Methoden für Amazon Location Service können dazu beitragen, Sicherheitsvorfälle zu verhindern:

Verwenden sicherer Verbindungen

Verwenden Sie immer verschlüsselte Verbindungen, z. B. solche, die mit `https://` beginnen, um vertrauliche Informationen während der Übertragung zu schützen.

Implementieren des Zugriffs mit den geringsten Berechtigungen auf -Ressourcen

Wenn Sie benutzerdefinierte Richtlinien für Amazon Location-Ressourcen erstellen, gewähren Sie nur die Berechtigungen, die zum Ausführen einer Aufgabe erforderlich sind. Es wird empfohlen, mit einem Mindestsatz von Berechtigungen zu beginnen und bei Bedarf zusätzliche Berechtigungen zu erteilen. Die Implementierung des Zugriffs mit den geringsten Berechtigungen ist unerlässlich, um das Risiko und die Auswirkungen zu reduzieren, die sich aus Fehlern oder böswilligen Angriffen ergeben könnten. Weitere Informationen finden Sie unter [the section called "Identitäts- und Zugriffsverwaltung"](#).

Verwenden global eindeutiger IDs als Geräte-IDs

Verwenden Sie die folgenden Konventionen für Geräte-IDs.

- Geräte-IDs müssen eindeutig sein.
- Geräte-IDs sollten nicht geheim sein, da sie als Fremdschlüssel für andere Systeme verwendet werden können.
- Geräte-IDs sollten keine persönlich identifizierbaren Informationen (PII) wie Telefongeräte-IDs oder E-Mail-Adressen enthalten.
- Geräte-IDs sollten nicht vorhersehbar sein. Opaque-Kennungen wie UUIDs werden empfohlen.

Nehmen Sie PII nicht in Gerätepositionseigenschaften auf

Geben Sie beim Senden von Geräteaktualisierungen (z. B. mit [DevicePositionUpdate](#)) keine persönlich identifizierbaren Informationen (PII) wie Telefonnummer oder E-Mail-Adresse in die `anPositionProperties`.

Bewährte Methoden für Amazon Location Service

Dieses Thema enthält bewährte Methoden, die Sie bei der Verwendung von Amazon Location Service unterstützen. Diese bewährten Methoden können Ihnen zwar helfen, den Amazon

Location Service voll auszunutzen, stellen jedoch keine vollständige Lösung dar. Sie sollten nur die Empfehlungen befolgen, die für Ihre Umgebung gelten.

Themen

- [Sicherheit](#)
- [Ressourcenmanagement](#)
- [Fakturierung und Kostenmanagement](#)
- [Kontingente und Nutzung](#)

Sicherheit

Um Sicherheitsrisiken zu bewältigen oder sogar zu vermeiden, sollten Sie die folgenden bewährten Methoden berücksichtigen:

- Verwenden Sie Identitätsverbund und IAM-Rollen, um den Zugriff auf Ihre Amazon Location-Ressourcen zu verwalten, zu steuern oder einzuschränken. Weitere Informationen finden Sie unter [Bewährte Methoden für IAM](#) im IAM-Benutzerhandbuch.
- Folgen Sie dem Prinzip der geringsten Berechtigung, um nur den mindestens erforderlichen Zugriff auf Ihre Amazon Location Service-Ressourcen zu gewähren. Weitere Informationen finden Sie unter [the section called “Verwalten des Zugriffs mit Richtlinien”](#).
- Beschränken Sie für Amazon Location Service-Ressourcen, die in Webanwendungen verwendet werden, den Zugriff mithilfe einer `aws:referer` IAM-Bedingung und beschränken Sie die Verwendung durch andere Websites als die, die in der Zulassungsliste enthalten sind.
- Verwenden Sie Überwachungs- und Protokollierungstools, um den Ressourcenzugriff und die Ressourcennutzung zu verfolgen. Weitere Informationen finden Sie unter [the section called “Protokollieren und Überwachen”](#) und [Protokollieren von Datenereignissen für Trails](#) im AWS CloudTrail -Benutzerhandbuch.
- Verwenden Sie sichere Verbindungen, z. B. solche, die mit `beginnen, https://` um Sicherheit zu erhöhen und Benutzer vor Angriffen zu schützen, während Daten zwischen dem Server und dem Browser übertragen werden.

Weitere Informationen zu bewährten Methoden für detektivische und präventive Sicherheit finden Sie im Thema auf [the section called “Bewährte Methoden für die Gewährleistung der Sicherheit”](#).

Ressourcenmanagement

Um Ihre Standortressourcen effektiv in Amazon Location Service zu verwalten, sollten Sie die folgenden bewährten Methoden berücksichtigen:

- Verwenden Sie regionale Endpunkte, die für Ihre erwartete Benutzerbasis von zentraler Bedeutung sind, um ihre Erfahrung zu verbessern. Weitere Informationen zu Regionsendpunkten finden Sie unter [Regionen und Endpunkte von Amazon Location](#).
- Stellen Sie bei Ressourcen, die Datenanbieter verwenden, wie Kartenressourcen und Ortsindexressourcen sicher, dass Sie die Nutzungsbedingungen des jeweiligen Datenanbieters befolgen. Weitere Informationen finden Sie unter [Datenanbieter](#).
- Minimieren Sie die Erstellung von Ressourcen, indem Sie über eine Ressource für jede Konfiguration von Karte, Ortsindex oder Routen verfügen. Innerhalb einer Region benötigen Sie in der Regel nur eine Ressource pro Datenanbieter oder Kartenstil. Die meisten Anwendungen verwenden vorhandene Ressourcen und erstellen zur Laufzeit keine Ressourcen.
- Wenn Sie verschiedene Ressourcen in einer einzigen Anwendung verwenden, z. B. eine Kartenressource und einen Routenrechner, verwenden Sie in jeder Ressource denselben Datenanbieter, um sicherzustellen, dass die Daten übereinstimmen. Wenn beispielsweise eine Routengeometrie, die Sie mit Ihrem Routenrechner erstellen, mit den Straßen auf der Karte übereinstimmt, die mit der Kartenressource grafisch dargestellt wird.

Fakturierung und Kostenmanagement

Um Ihre Kosten und Fakturierung zu verwalten, sollten Sie die folgende bewährte Methode in Betracht ziehen:

- Verwenden Sie Überwachungstools wie Amazon CloudWatch, um Ihre Ressourcennutzung zu verfolgen. Sie können Warnmeldungen einrichten, die Sie benachrichtigen, wenn die Nutzung Ihre angegebenen Grenzwerte überschreitet. Weitere Informationen finden Sie unter [Erstellen eines Fakturierungsalarms zur Überwachung Ihrer geschätzten AWS-Gebühren](#) im Amazon-CloudWatch Benutzerhandbuch.

Kontingente und Nutzung

Sie AWS-Konto schließen Kontingente ein, die ein Standardlimit für Ihren Nutzungsbetrag festlegen. Sie können Alarmer einrichten, um Sie zu warnen, wenn Ihre Nutzung Ihr Limit erreicht, und Sie

können bei Bedarf eine Erhöhung eines Kontingents anfordern. Weitere Informationen zum Arbeiten mit Kontingenten finden Sie in den folgenden Themen.

- [Amazon Location Service-Kontingente](#)
- [Verwenden von CloudWatch zur Überwachung der Nutzung anhand von Kontingenten](#)
- [Visualisieren Ihrer Service Quotas und Einstellen von Alarmen](#) im Amazon- CloudWatch Benutzerhandbuch.

Sie können Alarme erstellen, um Sie im Voraus zu warnen, wenn Sie Ihre Grenzwerte fast überschreiten. Wir empfehlen, Alarme für jedes Kontingent in jedem festzulegen AWS-Region , in dem Sie Amazon Location verwenden. Sie können beispielsweise Ihre Verwendung des `-SearchPlaceIndexForText` Vorgangs überwachen und einen Alarm erstellen, wenn Sie 80 Prozent Ihres aktuellen Kontingents überschreiten.

Wenn Sie eine Alarmwarnung über Ihr Kontingent erhalten, müssen Sie entscheiden, was Sie tun möchten. Möglicherweise verwenden Sie zusätzliche Ressourcen, da Ihr Kundenstamm wächst. In diesem Fall können Sie eine Erhöhung Ihres Kontingents anfordern, z. B. eine Erhöhung des Kontingents um 50 Prozent für einen API-Aufruf in dieser Region. Oder vielleicht gibt es einen Fehler in Ihrem Service, der dazu führt, dass Sie zusätzliche unnötige Aufrufe an Amazon Location tätigen. In diesem Fall möchten Sie das Problem in Ihrem Service lösen.

Dokumentverlauf

In der folgenden Tabelle wird die Dokumentation für Amazon Location Service beschrieben. Um Benachrichtigungen über Aktualisierungen zu erhalten, können Sie einen RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
Amazon Location Service veröffentlicht ein neues SDK für JavaScript	Um die Entwicklung von Amazon Location-Anwendungen mit in Web-Frontends zu vereinfachen, fügt Amazon Location ein neues Open-Source-SDK hinzu, das AWS SDK for JavaScript v3 unterstützt, wodurch die Authentifizierung und die Verwendung von GeoJSON vereinfacht werden. Weitere Informationen finden Sie unter Amazon Location SDK .	6. Juli 2023
Amazon Location Service veröffentlicht API-Schlüssel für allgemeine Verfügbarkeit	Amazon Location bietet Unterstützung für Ort und Route und kündigt die allgemeine Verfügbarkeit der API-Schlüsselfunktion an. Weitere Informationen finden Sie unter Verwenden von API-Schlüsseln .	6. Juli 2023
Amazon Location Service fügt Amazon- EventBridge Ereignisse für Positionsaktualisierungen hinzu	Amazon Location fügt Unterstützung für das Senden von Tracker-Positionsaktualisierungsereignissen an hinzu EventBridge. Weitere Informationen, einschließlich der Aktivierung der Ereigniss	6. Juli 2023

	e für einen Tracker, finden Sie unter Reagieren auf Ereignisse mit EventBridge .	
Amazon Location fügt Geofences Metadaten hinzu	Mit der Amazon Location API können Sie jetzt Metadaten eigenschaften zu Ihren Geofences hinzufügen. Diese werden mit Ihrem Geofence gespeichert und in Ereignissen im Zusammenhang mit dem Geofence in Amazon aufgenommen EventBridge. Weitere Informationen finden Sie unter Zeichnen von Geofences und Erstellen von Ereignisregeln .	15. Juni 2023
Amazon Location fügt Kategorien für Orte hinzu	Amazon Location fügt Kategorien zu Suchergebnissen hinzu und filtert Ergebnisse nach Kategorien. Weitere Informationen finden Sie unter Kategorien und Filtern von .	15. Juni 2023
Amazon Location führt Polnische Ansichten ein	Amazon Location fügt bestimmten Kartenstilen öffentliche Ansichten hinzu. Weitere Informationen finden Sie unter Politische Ansichten .	23. Mai 2023

[Amazon Location führt neue Demo- und Beispielwebsite ein](#)

Amazon Location kündigt eine neue Website an, die Ihnen Zugriff auf Amazon Location-Demos und -Beispiele bietet. Weitere Informationen finden Sie auf der [Demo-Website von Amazon Location](#).

3. Mai 2023

[Amazon Location führt längere Routen in ein Calculate RouteMatrix](#)

Amazon Location ermöglicht jetzt Routen mit unbegrenzter Länge für Routenmatrixrouten, die mit dem HERE-Date nanbieter erstellt wurden. Weitere Informationen finden Sie unter [Längere Routenplanung](#).

24. April 2023

[Die Amazon Location-Dokumentation fügt Funktionsunterschiede nach Datenanbieter hinzu](#)

Die Dokumentation zu Amazon Location wurde mit Informationen zu den Unterschieden zwischen den einzelnen Datenanbietern in Karten, Ortssuche und Routing aktualisiert. Weitere Informationen finden Sie unter [Funktionen nach Datenanbieter](#).

30. März 2023

[Amazon Location Open Data ordnet die allgemeine Verfügbarkeit zu](#)

Allgemeine Verfügbarkeit des Datenanbieters und -OpenStreetMapstils von Amazon Location Service, basierend auf den Sommerkarten von . Weitere Informationen finden Sie unter [Open Data](#).

7. März 2023

[Amazon Location fügt neue Autorisierungsmethode in der Vorschau hinzu](#)

Amazon Location Service fügt API-Schlüssel als neue Autorisierungsmethode für anonyme Benutzer im Vorschaumodus hinzu. Weitere Informationen finden Sie unter [Zulassen des nicht authentifizierten Gastzugriffs auf Ihre Anwendung mithilfe von API-Schlüsseln](#).

23. Februar 2023

[Die Dokumentation zu Amazon Location wurde mit den neuesten bewährten IAM-Methoden aktualisiert](#)

Die Dokumentation zu Amazon Location Service wurde aktualisiert, um den neuesten AWS Identity and Access Management bewährten Methoden zu entsprechen. Weitere Informationen finden Sie unter [Sicherheit in Amazon Location Service](#).

26. Januar 2023

[Amazon Location Service fügt GrabMaps als Datenanbieter in Südostasien hinzu](#)

Amazon Location führt GrabMaps als Datenanbieter in Südostasien ein. Weitere Informationen finden Sie unter [GrabMaps](#).

10. Januar 2023

[Open-Data-Übersichten von Amazon Location Service in der Vorschau](#)

Neuer Amazon Location-Datenanbieter und -Stil wurden in der öffentlichen Vorschau hinzugefügt, basierend auf OpenStreetMapden Daylight-Übersichten von . Weitere Informationen finden Sie unter [Open Data \(Vorschau\)](#).

15. Dezember 2022

Neue HERE-Satellitenbilder	Für Karten, die HERE als Datenanbieter verwenden, wurden zwei neue Kartenstile hinzugefügt: HERE microSDe Imagery und HERE Hybrid Map styles. Weitere Informationen finden Sie unter HERE-Kartenstile .	25. Oktober 2022
Einheiten in Adressen	Amazon Location Service unterstützt jetzt Einheiten in Adressen, z. B. „123 Main St, Apartment 3B , Beliebig, USA“.	20. September 2022
Abrufen von Orten nach ID	Amazon Location Service unterstützt jetzt die Suche nach dem genauen Speicherort, den die SearchPlaceIndexForSuggestions Operation mithilfe der GetPlace Operation vorschlägt. Weitere Informationen finden Sie unter Verwenden der automatischen Vervollständigung von .	20. September 2022
Zusätzliche Bedingungschlüssel für die IAM-Richtlinie	Amazon Location Service unterstützt jetzt zusätzliche Bedingungschlüssel, mit denen Sie den Zugriff für bestimmte Geofences oder Geräte in der IAM-Richtlinie festlegen können. Siehe Bedingungschlüssel .	23. August 2022

Lokale Geofences	Amazon Location Service unterstützt jetzt Geofences , die als Kreis mit einem Mittelpunkt und einem Radius definiert sind, um Ereignisse abzurufen, wenn sich Geräte innerhalb einer bestimmten Entfernung von einem Standort befinden. Weitere Informationen finden Sie unter Hinzufügen von Zirkelgeofences .	11. August 2022
Kombinierte API-Referenz	Amazon Location Service verfügt jetzt über ein einziges API-Referenzhandbuch und nicht über separate Leitfäden für jeden Subservice. Weitere Informationen zu den APIs finden Sie unter Amazon Location APIs .	7. Juli 2022
Integration von Service Quotas	Amazon Location ist jetzt in Service Quotas integriert , sodass Sie Ihre Kontingente über die AWS Management Console oder mithilfe der anzeigen und verwalten könnenAWS CLI.	6. Juli 2022
Kapitel zur Dokumentation aktualisierter Konzepte	Das Kapitel Amazon-Standortkonzepte wurde mit weiteren Informationen für Benutzer von Amazon Location aktualisiert.	22. April 2022

Neues Android-Schnellstart-Tutorial	Ein neues Schnellstart-Tutorial für die Android-Entwicklung mit Kotlin wurde hinzugefügt, um Entwicklern einen schnellen Einstieg zu ermöglichen.	15. April 2022
Neue HERE-Map-Stile	Für Karten, die HERE als Datenanbieter verwenden, wurden zwei neue Kartenstile hinzugefügt. Weitere Informationen finden Sie unter HERE-Kartenstile .	15. März 2022
Dokumentation mit zusätzlichen Codebeispielen und Tutorials umstrukturieren	Dieses Entwicklerhandbuch wurde umstrukturiert, um Themen einfacher zu finden, einschließlich der neuen Kapitel Schnellstart und Codebeispiele .	25. Februar 2022
Genauigkeitsbasierte Positionsfiltrierung für Tracker	Sie können jetzt genauigkeitsbasierte Filter verwenden, wenn Sie eine Tracker-Ressource erstellen .	7. Dezember 2021
Automatische Vervollständigung für Ortsindizes	Sie können jetzt die automatische Vervollständigung verwenden, wenn Sie Ortsindizes durchsuchen.	6. Dezember 2021

[Neues Amplify-Tutorial für die Verwendung von Karten](#)

Es ist ein neues Tutorial verfügbar, das zeigt, wie Sie verwenden, AWS Amplify um Karten in einer Webanwendung anzuzeigen. Das Tutorial finden Sie unter [Verwenden der Amplify-Bibliothek mit Amazon Location Service](#).

24. November 2021

[Abfrageerweiterungen platzieren](#)

Amazon Location Service unterstützt jetzt das Festlegen einer bevorzugten Sprache für Ergebnisse bei Geokodierung oder Reverse-Geokodierung und fügt die Zeitzone und andere Informationen zu den Ergebnissen hinzu. Weitere Informationen zur Geokodierung und umgekehrten Geokodierung finden Sie unter [Geokodierung, umgekehrte Geokodierung und Suche nach](#) .

16. November 2021

[Tracker-Positionsfilterung](#)

Amazon Location Service fügt den Trackern eine neue Positionsfilterfunktion hinzu, mit der Sie die Kosten kontrollieren können. Diese Funktion filtert einige Positionsaktualisierungen auf Geräten heraus, bevor die Updates gespeichert oder anhand von Geofences ausgewertet werden. Weitere Informationen zur Positionsfilterung finden Sie unter [Tracker](#).

5. Oktober 2021

[Aktualisierungsvorgänge](#)

Die folgenden Operationen wurden den API-Referenzen für Amazon Location Service hinzugefügt: [UpdateMap](#), [UpdatePlaceIndexUpdateRouteCalculator](#) [UpdateGeofenceCollection](#), und [UpdateTracker](#).

19. Juli 2021

[Tutorial-Update: Amazon Aurora PostgreSQL benutzerdefinierte Funktionen](#)

Ein neues Tutorial zur Verwendung [Amazon Aurora PostgreSQL benutzerdefinierte Funktionen mit Amazon Location](#) zum Validieren, Bereinigen und Anreichern von Geodaten wurde hinzugefügt.

19. Juli 2021

[AWS- CloudFormation Ressourcen](#)

Amazon Location unterstützt jetzt [das Erstellen der folgenden Ressourcentypen in AWS CloudFormation-Ressourcen](#): `AWS::Location::Map` , `AWS::Location::PlaceIndex` , `AWS::Location::RouteCalculator` , `AWS::Location::Tracker` `AWS::Location::TrackerConsumer` , und `AWS::Location::GeofenceCollection` .

7. Juni 2021

[Markieren von Ressourcen](#)

Sie können [Ihren Amazon Location-Ressourcen jetzt Tags hinzufügen](#), um Ihre Ressourcen zu verwalten, zu identifizieren, zu organisieren, zu suchen und zu filtern.

1. Juni 2021

[Allgemeine Verfügbarkeit](#)

Allgemeine Verfügbarkeit der Entwicklerdokumentation zu Amazon Location Service: [Region und Endpunkte](#) und [Service Quotas](#) aktualisiert.

1. Juni 2021

[Esri-Imagery](#)

Amazon Location unterstützt jetzt die Verwendung des Esri-Kartenstils: [Esri Imagery](#) . Weitere Informationen finden Sie unter [Esri World Imagery](#) auf der Esri-Website.

1. Juni 2021

[Berechnen von Routen](#)

Sie können jetzt [Routenrec hner von Amazon Location verwenden](#), um Routen zu berechnen und die Reisezeit auf der Grundlage von up-to-date Straßennetz- und Live-Datenverkehrsinformationen des von Ihnen ausgewählten Datenanbieters zu schätzen.

1. Juni 2021

[AWS KMS Kundenverwaltete Schlüsselverschlüsselung für Daten im Ruhezustand](#)

Amazon Location unterstützt jetzt die Verwendung eines symmetrischen kundenverwalteten Schlüssels, den Sie erstellen, besitzen und verwalten, um [eine zweite Verschlüsselungsebene über die vorhandene -AWSeigene Verschlüsselung hinzuzufügen](#).

1. Juni 2021

[Veröffentlichung der öffentlichen Vorschau](#)

Erstveröffentlichung der öffentlichen Vorschau dokumentation.

16. Dezember 2020

[Tutorial-Update: Anzeigen von Karten](#)

Tutorials zum Anzeigen von Karten mit MapLibre für Android und iOS wurden aktualisiert, um das MapLibre native SDK zu verwenden.

17. März 2020

AWS-Glossar

Die neueste AWS-Terminologie finden Sie im [AWS-Glossar](#) in der AWS-Glossar-Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.