aws

開發人員指南

Amazon Pinpoint

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Pinpoint: 開發人員指南

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

..... x
Amazon Pinpoint 是什麼？ 1
使用 Amazon Pinpoint 傳送訊息給對象客群，並分析資料 1
定義對象客群 1
排程簡訊行銷活動 1
傳送交易訊息 2
使用分析和指標報告 2
使用端點來定義您的對象 3
新增端點 4
範例 4
相關資訊 10
將使用者與端點建立關聯 10
範例 11
相關資訊 15
新增一批端點 15
範例 15
相關資訊 23
匯入端點 23
開始之前 23
範例 24
相關資訊 35
從 Amazon Pinpoint 匯出端點至 Amazon S3 儲存貯體 35
開始之前 36
範例 36
相關資訊 47
在 Amazon Pinpoint 專案中查詢端點 47
範例 47
相關資訊 53
列出端點 IDs 53
管理端點上限 55
刪除端點 56
範例 56
建立或匯入客群 60
建置客群 60

使用 建置客群 適用於 Java 的 AWS SDK 60
匯入客群 64
使用 匯入客群 適用於 Java 的 AWS SDK 64
自訂客群 66
事件資料 68
建立 Lambda 函式 68
指派 Lambda 函數政策 70
將 Lambda 函數指派給行銷活動 73
以程式設計方式建立 Amazon Pinpoint 行銷活動 74
使用適用於 Java 的 SDK 建立行銷活動 74
建立 A/B 測試行銷活動 77
管理標籤 80
在 IAM 政策中使用標籤 81
將標籤新增到資源 82
使用 API 新增標籤 82
使用 新增標籤 AWS CLI 83
顯示 資源的標籤 84
使用 API 顯示標籤 84
使用 顯示標籤 AWS CLI 85
更新或覆寫標籤 86
移除 資源的標籤 86
使用 API 移除標籤 86
使用 移除標籤 AWS CLI 87
與您的應用程式整合 88
使用 AWS SDKs 88
使用 Amplify 連接您的前端應用程式 89
下一步驟 90
在應用程式中註冊端點 90
開始之前 90
AWS 行動SDKs 90
AWS Amplify 91
後續步驟 91
報告應用程式中的事件 91
開始之前 92
AWS 行動SDKs 92
Web 與 React Native 91

Amazon Pinpoint 事件 API ... 93
後續步驟 ... 93
從應用程式傳送交易訊息 ... 94
傳送交易電子郵件 ... 94
選擇要傳送電子郵件的方法 ... 94
在 Amazon Pinpoint 和 Amazon SES 之間進行選擇 ... 95
使用 API 傳送電子郵件 ... 95
新增電子郵件取消訂閱標頭 ... 109
傳送簡訊 ... 111
傳送語音訊息 ... 124
使用 SMS 和語音 API ... 132
產生一次性密碼 ... 133
SendOtpMessage 回應 ... 135
驗證 OTP 訊息 ... 136
VerifyOtpMessage 回應 ... 136
Amazon Pinpoint 中的 OTP 程式碼範例 ... 137
產生參考 ID ... 137
傳送 OTP 代碼 ... 137
驗證 OTP 代碼 ... 139
自訂應用程式內訊息 ... 140
擷取端點的應用程式內訊息 ... 140
GetInAppMessages API 回應 JSON 範例 ... 142
InAppMessageCampaigns 物件 ... 144
InAppMessage 物件 ... 145
HeaderConfig 物件 ... 146
BodyConfig 物件 ... 146
InAppMessageContent 物件 ... 147
Schedule 物件 ... 147
InAppMessageButton 物件 ... 148
DefaultButtonConfig 物件 ... 149
OverrideButtonConfig 物件 ... 150
電話號碼驗證 ... 151
Amazon Pinpoint 電話號碼驗證使用案例 ... 151
使用 驗證電話號碼 AWS CLI ... 152
電話號碼驗證回應 ... 152
建立自訂頻道 ... 156

使用 Webhook 156
使用 Lambda 函數 156
將 Lambda 函數或 Webhook 指派給個別行銷活動 158
為 Amazon Pinpoint 行銷活動建立和設定 Lambda 函數 159
Lambda 函數範例 159
Amazon Pinpoint 的 Lambda 函數回應格式 163
授予叫用 函數的許可 164
串流應用程式事件資料 167
設定事件資料串流 168
先決條件 168
AWS CLI 168
適用於 Java 的 AWS SDK 168
從 Amazon Pinpoint 的應用程式事件資料串流 169
應用程式事件範例 170
應用程式事件屬性 171
來自 Amazon Pinpoint 的行銷活動事件資料串流 175
行銷活動事件範例 175
行銷活動事件屬性 176
來自 Amazon Pinpoint 的旅程事件資料 183
旅程事件範例 183
旅程事件屬性 184
從 Amazon Pinpoint 傳送電子郵件事件資料串流 188
電子郵件事件範例 189
電子郵件事件屬性 195
來自 Amazon Pinpoint 的簡訊事件資料串流 201
SMS 事件範例 201
簡訊事件屬性 202
刪除事件串流 209
AWS CLI 209
適用於 Java 的 AWS SDK 209
查詢分析資料 210
查詢 Amazon Pinpoint 中的指標 210
IAM 政策 211
專案、行銷活動和旅程的標準指標 215
行銷活動的應用程式指標 216
電子郵件的應用程式指標 220

SMS 的應用程式指標 226
行銷活動指標 231
行程參與指標 237
行程執行指標 242
行程活動執行指標 243
旅程和行銷活動執行指標 246
查詢行銷活動資料 247
先決條件 248
查詢一個行銷活動的資料 249
查詢多個行銷活動的資料 253
查詢交易簡訊資料 258
先決條件 259
查詢交易電子郵件的資料 259
查詢交易 SMS 的資料 264
使用 JSON 查詢結果 269
JSON 結構 269
JSON 物件和欄位 274
使用 CloudTrail 記錄 API 呼叫 276
CloudTrail 中的 Amazon Pinpoint 資訊 276
CloudTrail 日誌檔案中的 API 動作 277
CloudTrail 日誌檔案中的電子郵件 API 動作 281
CloudTrail 日誌檔案中支援的 SMS 和語音 API v1 動作 282
CloudTrail 日誌項目範例 283
使用推薦者模型 288
將建議新增至訊息 288
叫用建議者模型的 Lambda 函數 290
輸入事件資料 290
回應資料和需求 292
指派政策來處理建議資料 296
授權 Amazon Pinpoint 叫用 Lambda 函數 298
設定推薦者模型 299
刪除 Amazon Pinpoint 專案資料 300
刪除所有 Amazon Pinpoint 專案資料 300
程式碼範例 301
Amazon Pinpoint 302
基本概念 303

案例 393
Amazon Pinpoint SMS 和語音 API 405
基本概念 406
安全 426
資料保護 426
資料加密 428
網際網路流量隱私權 429
為 Amazon Pinpoint 建立介面 VPC 端點 429
身分與存取管理 431
目標對象 431
使用身分驗證 432
使用政策管理存取權 433
Amazon Pinpoint 如何搭配 IAM 運作 434
Amazon Pinpoint 政策動作 440
身分型政策範例 469
一般任務的 IAM 角色 482
疑難排解 499
日誌記錄和監控 500
法規遵循驗證 501
恢復能力 502
基礎設施安全性 502
組態與漏洞分析 503
安全最佳實務 503
配額 505
專案配額 505
API 請求配額 506
行銷活動配額 508
電子郵件配額 509
電子郵件訊息配額 509
電子郵件寄件者和收件人配額 509
電子郵件傳送份額 511
端點配額 512
端點匯入配額 513
事件擷取配額 513
行程配額 514
Lambda 配額 515

Machine Learning 配額 516
訊息範本配額 517
推播通知配額 518
應用程式內訊息配額 518
區段配額 519
簡訊配額 519
10DLC 配額 519
語音配額 520
請求提高配額 520
文件歷史紀錄 522
舊版更新 528

支援結束通知：2026 年 10 月 30 日， AWS 將結束對 Amazon Pinpoint 的支援。2026 年 10 月 30 日之後，您將無法再存取 Amazon Pinpoint 主控台或 Amazon Pinpoint 資源 (端點、區段、行銷活動、旅程和分析)。如需詳細資訊，請參閱 Amazon Pinpoint 終止支援。注意：與 SMS、語音、行動推播、OTP 和電話號碼驗證相關的 APIs 不受此變更影響，並受 AWS 最終使用者傳訊支援。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。

Amazon Pinpoint 是什麼？

Amazon Pinpoint 是一項 AWS 服務，您可以用來跨多個簡訊管道與客戶互動。您可以使用 Amazon Pinpoint 來傳送推播通知、電子郵件、SMS 文字訊息或語音訊息。

本開發人員指南中的資訊適用於應用程式開發人員。本指南包含透過程式化方式，使用 Amazon Pinpoint 功能的相關資訊。還包含行動應用程式開發人員特別有興趣的資訊，例如整合分析和簡訊功能與您的應用程式的程序。

Amazon Pinpoint 適用於北美洲、歐洲、亞洲和大洋洲的數個 AWS 區域。如需 的詳細資訊 AWS 區域，請參閱《》中的管理 AWS 區域Amazon Web Services 一般參考。如需目前可使用 Amazon Pinpoint 的所有區域，請參閱 Amazon Web Services 一般參考 中的 Amazon Pinpoint 端點和配額及 AWS 服務端點查看清單。如需進一步了解各區域之可用區域數量的資訊，請參閱 AWS 全球基礎設施。

如需 Amazon Pinpoint 的詳細資訊，請參閱下列指南：

- Amazon Pinpoint API 參考
- Amazon Pinpoint SMS 和語音 API
- Amazon Pinpoint 使用者指南

使用 Amazon Pinpoint 傳送訊息給對象客群，並分析資料

您可以使用 Amazon Pinpoint 來定義受眾客群、傳送訊息行銷活動和交易訊息，以及使用指標來分析使用者行為。

定義對象客群

定義對象客群，讓您的訊息觸達正確的對象。客群指定哪些使用者會接收行銷活動送來的訊息。您可以根據您的應用程式所報告的資料，例如作業系統或行動裝置類型，以定義動態客群。您也可以匯入您使用另一個服務或應用程式定義的靜態區段。

排程簡訊行銷活動

建立簡訊行銷活動以吸引您的對象。行銷活動會按照您定義的排程傳送量身訂做的訊息。您可以建立行銷活動來傳送行動推送、電子郵件或簡訊。

若要試驗替代行銷活動策略，請將您的行銷活動設為 A/B 測試，並透過 Amazon Pinpoint 分析來分析結果。

傳送交易訊息

直接向特定使用者傳送交易行動推播和 SMS 訊息 (例如新帳戶啟用訊息、訂單確認和密碼重設通知)，讓客戶隨時了解情況。您可以使用 Amazon Pinpoint REST API 傳送交易訊息。

使用分析和指標報告

Amazon Pinpoint 提供的分析，有助您深入了解自己的受眾，以及您行銷活動的成效。您可以檢視有關使用者參與程度、購買活動和人口統計等的趨勢。您也可以藉由檢視指標 (例如針對行銷活動或應用程式所傳送或開啟的訊息總數) 來監控訊息流量。有了 Amazon Pinpoint API，您的應用程式可以回報自訂資料，讓 Amazon Pinpoint 將這些資料用於分析，您可以就某些標準指標，查詢分析資料。

若要在 Amazon Pinpoint 以外的地方分析或存放分析資料，可以設定 Amazon Pinpoint 將資料串流到 Amazon Kinesis。

使用端點在 Amazon Pinpoint 中代表您的對象

在 Amazon Pinpoint 中，每位受眾成員都由一個或多個端點代表。當您使用 Amazon Pinpoint 傳送訊息時，您可以將該訊息導向至代表目標受眾成員的端點。每個端點定義都包含一個訊息目的地，例如裝置權杖、電子郵件地址或電話號碼。它還包含關於您的使用者及其裝置的資料。您必須先將端點新增至 Amazon Pinpoint 專案，才能分析、分割或吸引受眾。

隨著您的對象成長和變更，您的端點資料也會成長和變更。若要檢視 Amazon Pinpoint 擁有的對象最新資訊，您可以個別查詢端點，也可以從 Amazon Pinpoint 專案匯出所有端點。透過查看端點資料，您可以查看下列有關使用者的資訊：

- 他們的裝置和平台。
- 他們的時區。
- 安裝在其裝置上的應用程式版本。
- 他們的城市和國家/地區位置。
- 您記錄的其他自訂屬性和指標。

Amazon Pinpoint 主控台也會提供端點擷取的人口統計和自訂屬性的分析。

下列主題說明如何在 Amazon Pinpoint 中使用端點。如需有關使用 Android、iOS 或 JavaScript 用戶端自動新增端點的資訊，請參閱 在應用程式中註冊 Amazon Pinpoint 端點。

主題

- 將端點新增至 Amazon Pinpoint
- 將使用者與 Amazon Pinpoint 端點建立關聯
- 將一批端點新增至 Amazon Pinpoint
- 將端點匯入 Amazon Pinpoint
- 從 Amazon Pinpoint 匯出端點至 Amazon S3 儲存貯體
- 在 Amazon Pinpoint 專案中查詢端點
- 使用 Amazon Pinpoint 列出端點 IDs
- 管理 Amazon Pinpoint 中的端點數量上限
- 以程式設計方式從 Amazon Pinpoint 刪除端點

將端點新增至 Amazon Pinpoint

端點代表訊息的目的地，例如行動裝置、電話號碼或電子郵件地址。在您可以傳送訊息給您的對象的成員之前，您必須為該個人定義一或多個端點。

將端點加到 Amazon Pinpoint，端點會成長為受眾資料庫。這些資料包含：

- 使用 Amazon Pinpoint API 加入或更新的端點。
- 隨著使用者來到您的應用程式，您的用戶端程式碼新增或更新的端點。

定義端點時，您會指定管道和地址。管道是用來傳送訊息給端點的平台類型。管道的範例包含推送通知服務、簡訊或電子郵件。指定要傳送訊息給端點的位置，例如裝置字符、電話號碼或電子郵件地址。

若要新增關於您的對象的更多詳細資訊，您可以以自訂和標準屬性來豐富您的端點。這些屬性包括有關您的使用者、其偏好設定、裝置、其使用的用戶端版本或其位置的資料。新增此類型的資料到您的端點時，您可以：

- 在 Amazon Pinpoint 主控台中檢視受眾相關圖表。
- 根據端點屬性區隔您的對象，使得您可以將訊息傳送到適合的鎖定目標對象。
- 透過整合將取代為使用者屬性值的訊息變數來個人化您的訊息。

如果您使用 Mobile SDKs 或 AWS Amplify JavaScript 程式庫整合 Amazon Pinpoint， AWS 則行動或 JavaScript 用戶端應用程式會自動註冊端點。用戶端會為每個新使用者註冊端點，並且為傳回的使用者更新端點。若要從行動裝置或 JavaScript 用戶端註冊端點，請參閱在應用程式中註冊 Amazon Pinpoint 端點。

範例

以下範例說明如何將端點加入 Amazon Pinpoint 專案。此端點代表位在西雅圖並使用 iPhone 的對象成員。您可以透過 Apple 推送通知服務 (APN) 傳送訊息給此人員。端點的地址為 APN 提供的裝置字符。

AWS CLI

透過 AWS CLI執行命令，可以使用 Amazon Pinpoint。

Example Update Endpoint 命令

若要新增或更新端點，請使用 update-endpoint 命令：

```
$ aws pinpoint update-endpoint \
> --application-id application-id \
> --endpoint-id endpoint-id \
> --endpoint-request file://endpoint-request-file.json
```

其中：

- application-id 是您要加入或更新端點的 Amazon Pinpoint 專案的 ID。
- example-endpoint 是您要指派給新端點的 ID，或是您要更新的現有端點的 ID。
- endpoint-request-file.json 是包含 `--endpoint-request` 參數輸入的本機 JSON 檔案的檔案路徑。

Example端點請求檔案

範例 `update-endpoint` 命令會使用 JSON 檔案做為 `--endpoint-request` 參數的引數。此檔案包含的端點定義類似以下：

```
{
  "ChannelType": "APNS",
  "Address": "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
  "Attributes": {
    "Interests": [
      "Technology",
      "Music",
      "Travel"
    ]
  },
  "Metrics": {
    "technology_interest_level": 9.0,
    "music_interest_level": 6.0,
    "travel_interest_level": 4.0
  },
  "Demographic": {
    "AppVersion": "1.0",
    "Make": "apple",
    "Model": "iPhone",
    "ModelVersion": "8",
    "Platform": "ios",
    "PlatformVersion": "11.3.1",
    "Timezone": "America/Los_Angeles"
  },
```

```
  "Location": {
    "Country": "US",
    "City": "Seattle",
    "PostalCode": "98121",
    "Latitude": 47.61,
    "Longitude": -122.33
  }
}
```

如需可用於定義端點的屬性，請參閱 Amazon Pinpoint API 參考中的 EndpointRequest 結構描述。

適用於 Java 的 AWS SDK

使用 適用於 Java 的 AWS SDK提供的用戶端，可以在 Java 應用程式中使用 Amazon Pinpoint API。

Example代碼

若要加入端點，請初始化 EndpointRequest 物件，然後傳遞給 AmazonPinpoint 用戶端的 updateEndpoint 方法：

```
import com.amazonaws.regions.Regions;
import com.amazonaws.services.pinpoint.AmazonPinpoint;
import com.amazonaws.services.pinpoint.AmazonPinpointClientBuilder;
import com.amazonaws.services.pinpoint.model.*;
import java.util.Arrays;

public class AddExampleEndpoint {

 public static void main(String[] args) {

  final String USAGE = "\n" +
    "AddExampleEndpoint - Adds an example endpoint to an Amazon Pinpoint
 application." +
    "Usage: AddExampleEndpoint <applicationId>" +
    "Where:\n" +
    "  applicationId - The ID of the Amazon Pinpoint application to add the example
 " +
    "endpoint to.";

  if (args.length < 1) {
   System.out.println(USAGE);
   System.exit(1);
```

```
  }

  String applicationId = args[0];

  // The device token assigned to the user's device by Apple Push Notification
  // service (APNs).
  String deviceToken =
"1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f";

  // Initializes an endpoint definition with channel type and address.
  EndpointRequest wangXiulansIphoneEndpoint = new EndpointRequest()
    .withChannelType(ChannelType.APNS)
    .withAddress(deviceToken);

  // Adds custom attributes to the endpoint.
  wangXiulansIphoneEndpoint.addAttributesEntry("interests", Arrays.asList(
    "technology",
    "music",
    "travel"));

  // Adds custom metrics to the endpoint.
  wangXiulansIphoneEndpoint.addMetricsEntry("technology_interest_level", 9.0);
  wangXiulansIphoneEndpoint.addMetricsEntry("music_interest_level", 6.0);
  wangXiulansIphoneEndpoint.addMetricsEntry("travel_interest_level", 4.0);

  // Adds standard demographic attributes.
  wangXiulansIphoneEndpoint.setDemographic(new EndpointDemographic()
    .withAppVersion("1.0")
    .withMake("apple")
    .withModel("iPhone")
    .withModelVersion("8")
    .withPlatform("ios")
    .withPlatformVersion("11.3.1")
    .withTimezone("America/Los_Angeles"));

  // Adds standard location attributes.
  wangXiulansIphoneEndpoint.setLocation(new EndpointLocation()
    .withCountry("US")
    .withCity("Seattle")
    .withPostalCode("98121")
    .withLatitude(47.61)
    .withLongitude(-122.33));

  // Initializes the Amazon Pinpoint client.
```

```
  AmazonPinpoint pinpointClient = AmazonPinpointClientBuilder.standard()
    .withRegion(Regions.US_EAST_1).build();

  // Updates or creates the endpoint with Amazon Pinpoint.
  UpdateEndpointResult result = pinpointClient.updateEndpoint(new
 UpdateEndpointRequest()
    .withApplicationId(applicationId)
    .withEndpointId("example_endpoint")
    .withEndpointRequest(wangXiulansIphoneEndpoint));

  System.out.format("Update endpoint result: %s\n",
 result.getMessageBody().getMessage());

 }
}
```

HTTP

對 REST API 直接提出 HTTP 請求，可以使用 Amazon Pinpoint。

Example PUT Endpoint 請求

若要新增端點，請將 PUT 請求發出至位在以下 URI 的 Endpoint 資源：

`/v1/apps/application-id/endpoints/endpoint-id`

其中：

- application-id 是您要加入或更新端點的 Amazon Pinpoint 專案的 ID。
- endpoint-id 是您要指派給新端點的 ID，或是您要更新的現有端點的 ID。

在您的請求中，包括所需的標頭，並提供 EndpointRequest JSON 做為內文：

```
PUT /v1/apps/application_id/endpoints/example_endpoint HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180415T182538Z
Content-Type: application/json
Accept: application/json
X-Amz-Date: 20180428T004705Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180428/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;content-length;content-
```

```
type;host;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{
  "ChannelType": "APNS",
  "Address": "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
  "Attributes": {
    "Interests": [
      "Technology",
      "Music",
      "Travel"
    ]
  },
  "Metrics": {
    "technology_interest_level": 9.0,
    "music_interest_level": 6.0,
    "travel_interest_level": 4.0
  },
  "Demographic": {
    "AppVersion": "1.0",
    "Make": "apple",
    "Model": "iPhone",
    "ModelVersion": "8",
    "Platform": "ios",
    "PlatformVersion": "11.3.1",
    "Timezone": "America/Los_Angeles"
  },
  "Location": {
    "Country": "US",
    "City": "Seattle",
    "PostalCode": "98121",
    "Latitude": 47.61,
    "Longitude": -122.33
  }
}
```

如果您的請求成功，您會收到類似以下的回應：

```
{
    "RequestID": "67e572ed-41d5-11e8-9dc5-db288f3cbb72",
    "Message": "Accepted"
}
```

相關資訊

如需 Amazon Pinpoint API 中端點資源的詳細資訊 (包括支援的 HTTP 方法和請求參數)，請參閱 Amazon Pinpoint API 參考中的端點。

如需使用變數將訊息個人化的詳細資訊，請參閱 Amazon Pinpoint 使用者指南中的訊息變數。

如需端點適用之配額的相關資訊，例如您可指派的屬性數目，請參閱 the section called “端點配額”。

將使用者與 Amazon Pinpoint 端點建立關聯

端點可以包含定義使用者的屬性，其代表您的對象中的人員。例如，使用者可能代表安裝您的行動應用程式的人員，或具備您的網站上帳戶的人員。

您可以透過指定唯一的使用者 ID 和選擇性的自訂使用者屬性來定義使用者。如果某個人員在多個裝置上使用您的應用程式，或如果可以透過在多個地址傳送訊息給該人員，則您可以將相同的使用者 ID 指派給多個端點。在這種情況下，Amazon Pinpoint 會同步所有端點的使用者屬性。因此如果您將使用者屬性加入一個端點，Amazon Pinpoint 會將該屬性加入含同一個使用者 ID 的所有端點。

您可以新增使用者屬性以追蹤套用至個人的資料，並且不會因該人員所使用的裝置而不同。例如，您可以新增人員的名稱、年齡或帳戶狀態的屬性。

> **Tip**
>
> 如果您的應用程式使用 Amazon Cognito 使用者集區處理使用者驗證，Amazon Cognito 會將使用者 ID 和屬性自動加入您的端點。若是使用者 ID 值端點，Amazon Cognito 會指派已指派給使用者集區中使用者的 sub 值。若要了解如何使用 Amazon Cognito 加入使用者，請參閱 Amazon Cognito 開發人員指南中的透過 Amazon Cognito 使用者集區使用 amazon pinpoint 分析。

新增使用者定義至端點之後，對於如何區隔您的對象，您會有更多選項。您可以根據使用者屬性定義客群，或者您可以透過匯入使用者 ID 的清單來定義客群。當您傳送訊息給根據使用者的客群，可能的目的地會包含該客群中每個使用者相關聯的每個端點。

對於如何傳送訊息給對象，也會有更多的選項。您可以使用行銷活動來傳送訊息給某個客群的使用者，或者您可以直接傳送訊息給使用者 ID 的清單。若要個人化您的訊息，您可以包含將取代為使用者屬性值的訊息變數。

範例

以下範例說明如何新增使用者定義至端點。

AWS CLI

透過 AWS CLI執行命令，可以使用 Amazon Pinpoint。

Example Update Endpoint 命令

若要新增使用者至端點，請使用 update-endpoint 命令。對於 `--endpoint-request` 參數，您可以定義一個新的端點，其包括使用者。或者，若要更新現有的端點，您可以只提供您要變更的屬性。以下範例會透過只提供使用者屬性，將使用者新增至現有的端點：

```
$ aws pinpoint update-endpoint \
> --application-id application-id \
> --endpoint-id endpoint-id \
> --endpoint-request file://endpoint-request-file.json
```

其中：

- *application-id* 是您要加入或更新端點的 Amazon Pinpoint 專案的 ID。
- *endpoint-id* 是您要指派給新端點的 ID，或是您要更新的現有端點的 ID。
- *endpoint-request-file.json* 是包含 `--endpoint-request` 參數輸入的本機 JSON 檔案的檔案路徑。

Example端點請求檔案

範例 `update-endpoint` 命令會使用 JSON 檔案做為 `--endpoint-request` 參數的引數。此檔案包含的使用者定義類似以下：

```
{
    "User":{
        "UserId":"example_user",
        "UserAttributes":{
            "FirstName":["Wang"],
            "LastName":["Xiulan"],
            "Gender":["Female"],
            "Age":["39"]
```

```
        }
    }
}
```

如需可用於定義使用者的屬性，請參閱 Amazon Pinpoint API 參考中 EndpointRequest 結構描述的 `User` 物件。

適用於 Java 的 AWS SDK

使用 適用於 Java 的 AWS SDK提供的用戶端，可以在 Java 應用程式中使用 Amazon Pinpoint API。

Example代碼

若要將使用者新增至端點，請初始化 EndpointRequest 物件，並將其傳遞至`AmazonPinpoint`用戶端的 `updateEndpoint`方法。您可以使用此物件來定義一個新的端點，其包括使用者。或者，若要更新現有的端點，您可以只更新您要變更的屬性。以下範例會透過將 EndpointUser 物件新增到 EndpointRequest 物件，將使用者新增到現有的端點：

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
   public static void updatePinpointEndpoint(PinpointClient pinpoint, String
 applicationId, String endPointId) {
        try {
            List<String> wangXiList = new ArrayList<>();
            wangXiList.add("cooking");
            wangXiList.add("running");
            wangXiList.add("swimming");

            Map myMapWang = new HashMap<>();
```

```
            myMapWang.put("interests", wangXiList);

            List<String> myNameWang = new ArrayList<>();
            myNameWang.add("Wang ");
            myNameWang.add("Xiulan");

            Map wangName = new HashMap<>();
            wangName.put("name", myNameWang);

            EndpointUser wangMajor = EndpointUser.builder()
                    .userId("example_user_10")
                    .userAttributes(wangName)
                    .build();

            // Create an EndpointBatchItem object for Mary Major.
            EndpointRequest wangXiulanEndpoint = EndpointRequest.builder()
                    .channelType(ChannelType.EMAIL)
                    .address("wang_xiulan@example.com")
                    .attributes(myMapWang)
                    .user(wangMajor)
                    .build();

            // Adds multiple endpoint definitions to a single request object.
            UpdateEndpointRequest endpointList = UpdateEndpointRequest.builder()
                    .applicationId(applicationId)
                    .endpointRequest(wangXiulanEndpoint)
                    .endpointId(endPointId)
                    .build();

            UpdateEndpointResponse result = pinpoint.updateEndpoint(endpointList);
            System.out.format("Update endpoint result: %s\n",
result.messageBody().message());

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

如需完整的 SDK 範例，請參閱 GitHub 上的 AddExampleUser.java。

HTTP

對 REST API 直接提出 HTTP 請求，可以使用 Amazon Pinpoint。

Example使用使用者定義放置端點請求

若要將使用者新增至端點，請將 PUT 請求發出至位在以下 URI 的 Endpoint 資源：

`/v1/apps/application-id/endpoints/endpoint-id`

其中：

- *application-id* 是您要加入或更新端點的 Amazon Pinpoint 專案的 ID。
- *endpoint-id* 是您要指派給新端點的 ID，或是您要更新的現有端點的 ID。

在您的請求中，包括所需的標頭，並提供 EndpointRequest JSON 做為內文。請求內文可以定義一個新的端點，其包括使用者。或者，若要更新現有的端點，您可以只提供您要變更的屬性。以下範例會透過只提供使用者屬性，將使用者新增至現有的端點：

```
PUT /v1/apps/application_id/endpoints/example_endpoint HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180415T182538Z
Content-Type: application/json
Accept: application/json
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180501/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;content-length;content-
type;host;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{
    "User":{
        "UserId":"example_user",
        "UserAttributes":{
            "FirstName":"Wang",
            "LastName":"Xiulan",
            "Gender":"Female",
            "Age":"39"
        }
    }
}
```

如果請求成功，您會收到類似如下的回應：

```
{
```

```
    "RequestID": "67e572ed-41d5-11e8-9dc5-db288f3cbb72",
    "Message": "Accepted"
}
```

相關資訊

如需 Amazon Pinpoint API 中端點資源的詳細資訊 (包括支援的 HTTP 方法和請求參數)，請參閱 Amazon Pinpoint API 參考中的端點。

如需使用變數將訊息個人化的詳細資訊，請參閱 Amazon Pinpoint 使用者指南中的訊息變數。

如需了解如何透過匯入使用者 ID 清單來定義客群，請參閱 Amazon Pinpoint 使用者指南中的匯入客群。

如需將私訊傳送給多達 100 個使用者 ID 的相關資訊，請參閱 Amazon Pinpoint API 參考中的使用者訊息。

如需端點適用之配額的相關資訊，例如您可指派的使用者屬性數量，請參閱 the section called “端點配額”。

將一批端點新增至 Amazon Pinpoint

您可以透過以批次提供端點，在單一操作中新增或更新提供多個端點。每個批次請求可以包含高達 100 個端點定義。

如果您要在單一操作中新增或更新超過 100 個端點，請改為參閱將端點匯入 Amazon Pinpoint。

範例

以下範例說明如何透過在批次請求中併入端點來一次新增兩個端點。

AWS CLI

透過 AWS CLI執行命令，可以使用 Amazon Pinpoint。

Example Update Endpoints Batch 命令

若要提交端點批次請求，請使用 `update-endpoints-batch` 命令：

```
$ aws pinpoint update-endpoints-batch \
```

```
> --application-id application-id \
> --endpoint-batch-request file://endpoint_batch_request_file.json
```

其中：

- *application-id* 是您要加入或更新端點的 Amazon Pinpoint 專案的 ID。
- *endpoint_batch_request_file.json* 是包含 --endpoint-batch-request 參數輸入的本機 JSON 檔案的檔案路徑。

Example端點批次請求檔案

範例 update-endpoints-batch 命令會使用 JSON 檔案做為 --endpoint-request 參數的引數。此檔案包含的端點批次定義類似以下：

```
{
    "Item": [
        {
            "ChannelType": "EMAIL",
            "Address": "richard_roe@example.com",
            "Attributes": {
                "Interests": [
                    "Music",
                    "Books"
                ]
            },
            "Metrics": {
                "music_interest_level": 3.0,
                "books_interest_level": 7.0
            },
            "Id": "example_endpoint_1",
            "User":{
                "UserId": "example_user_1",
                "UserAttributes": {
                    "FirstName": "Richard",
                    "LastName": "Roe"
                }
            }
        },
        {
            "ChannelType": "SMS",
            "Address": "+16145550100",
            "Attributes": {
```

```
                "Interests": [
                    "Cooking",
                    "Politics",
                    "Finance"
                ]
            },
            "Metrics": {
                "cooking_interest_level": 5.0,
                "politics_interest_level": 8.0,
                "finance_interest_level": 4.0
            },
            "Id": "example_endpoint_2",
            "User": {
                "UserId": "example_user_2",
                "UserAttributes": {
                    "FirstName": "Mary",
                    "LastName": "Major"
                }
            }
        }
    ]
}
```

如需可用於定義大量端點的屬性，請參閱 Amazon Pinpoint API 參考中的 EndpointBatchRequest 結構描述。

適用於 Java 的 AWS SDK

使用 適用於 Java 的 AWS SDK提供的用戶端，可以在 Java 應用程式中使用 Amazon Pinpoint API。

Example Code

若要提交端點批次請求，請初始化 `EndpointBatchRequest` 物件，然後傳遞給 `AmazonPinpoint` 用戶端的 `updateEndpointsBatch` 方法。以下範例會為 `EndpointBatchRequest` 物件填入兩個 `EndpointBatchItem` 物件：

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointsBatchResponse;
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
import software.amazon.awssdk.services.pinpoint.model.EndpointBatchItem;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
```

```
import software.amazon.awssdk.services.pinpoint.model.EndpointBatchRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointsBatchRequest;
import java.util.Map;
import java.util.List;
import java.util.ArrayList;
import java.util.HashMap;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointsBatchResponse;
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
import software.amazon.awssdk.services.pinpoint.model.EndpointBatchItem;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.EndpointBatchRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointsBatchRequest;
import java.util.Map;
import java.util.List;
import java.util.ArrayList;
import java.util.HashMap;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class AddExampleEndpoints {

        public static void main(String[] args) {
                final String usage = """

                                Usage:    <appId>

                                Where:
                                   appId - The ID of the application.

                                """;

                if (args.length != 1) {
```

```
                        System.out.println(usage);
                        System.exit(1);
                }

                String applicationId = args[0];
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();

                updateEndpointsViaBatch(pinpoint, applicationId);
                pinpoint.close();
        }

        public static void updateEndpointsViaBatch(PinpointClient pinpoint, String
applicationId) {
                try {
                        List<String> myList = new ArrayList<>();
                        myList.add("music");
                        myList.add("books");

                        Map myMap = new HashMap<String, List>();
                        myMap.put("attributes", myList);

                        List<String> myNames = new ArrayList<String>();
                        myList.add("Richard");
                        myList.add("Roe");

                        Map myMap2 = new HashMap<String, List>();
                        myMap2.put("name", myNames);

                        EndpointUser richardRoe = EndpointUser.builder()
                                        .userId("example_user_1")
                                        .userAttributes(myMap2)
                                        .build();

                        // Create an EndpointBatchItem object for Richard Roe.
                        EndpointBatchItem richardRoesEmailEndpoint =
EndpointBatchItem.builder()
                                        .channelType(ChannelType.EMAIL)
                                        .address("richard_roe@example.com")
                                        .id("example_endpoint_1")
                                        .attributes(myMap)
                                        .user(richardRoe)
                                        .build();
```

```
                        List<String> myListMary = new ArrayList<String>();
                        myListMary.add("cooking");
                        myListMary.add("politics");
                        myListMary.add("finance");

                        Map myMapMary = new HashMap<String, List>();
                        myMapMary.put("interests", myListMary);

                        List<String> myNameMary = new ArrayList<String>();
                        myNameMary.add("Mary ");
                        myNameMary.add("Major");

                        Map maryName = new HashMap<String, List>();
                        myMapMary.put("name", myNameMary);

                        EndpointUser maryMajor = EndpointUser.builder()
                                        .userId("example_user_2")
                                        .userAttributes(maryName)
                                        .build();

                        // Create an EndpointBatchItem object for Mary Major.
                        EndpointBatchItem maryMajorsSmsEndpoint =
EndpointBatchItem.builder()
                                        .channelType(ChannelType.SMS)
                                        .address("+16145550100")
                                        .id("example_endpoint_2")
                                        .attributes(myMapMary)
                                        .user(maryMajor)
                                        .build();

                        // Adds multiple endpoint definitions to a single request
object.
                        EndpointBatchRequest endpointList =
EndpointBatchRequest.builder()
                                        .item(richardRoesEmailEndpoint)
                                        .item(maryMajorsSmsEndpoint)
                                        .build();

                        // Create the UpdateEndpointsBatchRequest.
                        UpdateEndpointsBatchRequest batchRequest =
UpdateEndpointsBatchRequest.builder()
                                        .applicationId(applicationId)
                                        .endpointBatchRequest(endpointList)
```

```
                                        .build();

                        // Updates the endpoints with Amazon Pinpoint.
                        UpdateEndpointsBatchResponse result =
 pinpoint.updateEndpointsBatch(batchRequest);
                        System.out.format("Update endpoints batch result: %s\n",
 result.messageBody().message());

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
        }
}
```

如需完整的 SDK 範例，請參閱 GitHub 上的 AddExampleEndpoints.java。

HTTP

對 REST API 直接提出 HTTP 請求，可以使用 Amazon Pinpoint。

Example Put Endpoints 請求

若要提交端點批次請求，請發出 PUT 請求至位在以下 URI 的 Endpoints 資源：

/v1/apps/*application-id*/endpoints

application-id 是您要加入或更新端點的 Amazon Pinpoint 專案的 ID。

在您的請求中，包括所需的標頭，並提供 EndpointBatchRequest JSON 做為內文：

```
PUT /v1/apps/application_id/endpoints HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
X-Amz-Date: 20180501T184948Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180501/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;content-length;content-
type;host;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{
    "Item": [
        {
```

```
            "ChannelType": "EMAIL",
            "Address": "richard_roe@example.com",
            "Attributes": {
                "Interests": [
                    "Music",
                    "Books"
                ]
            },
            "Metrics": {
                "music_interest_level": 3.0,
                "books_interest_level": 7.0
            },
            "Id": "example_endpoint_1",
            "User":{
                "UserId": "example_user_1",
                "UserAttributes": {
                    "FirstName": "Richard",
                    "LastName": "Roe"
                }
            }
        },
        {
            "ChannelType": "SMS",
            "Address": "+16145550100",
            "Attributes": {
                "Interests": [
                    "Cooking",
                    "Politics",
                    "Finance"
                ]
            },
            "Metrics": {
                "cooking_interest_level": 5.0,
                "politics_interest_level": 8.0,
                "finance_interest_level": 4.0
            },
            "Id": "example_endpoint_2",
            "User": {
                "UserId": "example_user_2",
                "UserAttributes": {
                    "FirstName": "Mary",
                    "LastName": "Major"
                }
            }
```

```
        }
    ]
}
```

如果您的請求成功，您會收到類似以下的回應：

```
{
    "RequestID": "67e572ed-41d5-11e8-9dc5-db288f3cbb72",
    "Message": "Accepted"
}
```

相關資訊

如需 Amazon Pinpoint API 中端點資源的詳細資訊 (包括支援的 HTTP 方法和請求參數)，請參閱 Amazon Pinpoint API 參考中的端點。

將端點匯入 Amazon Pinpoint

將端點從 Amazon S3 儲存貯體匯入，可以新增或更新大量端點。若在 Amazon Pinpoint 以外的地方有受眾記錄，而且想將此資訊加入 Amazon Pinpoint 專案，匯入端點是有用的方法。在此情況下，您可以：

1. 建立根據您自己的對象資料的端點定義。
2. 請將這些端點定義儲存在一個或多個檔案中，並將檔案上傳至 Amazon S3 儲存貯體。
3. 將端點從儲存貯體匯入，可以將端點加入 Amazon Pinpoint 專案。

每個匯入任務可傳輸最多 1 GB 的資料。在一般任務中，其中的每個端點為 4 KB 或更少，您可以匯入大約 250,000 個端點。每個 AWS 帳戶最多可以執行兩個並行匯入任務。如果您的匯入任務需要更多頻寬，您可以向 提交服務配額增加請求 支援。如需詳細資訊，請參閱請求提高配額。

開始之前

在您可以匯入端點之前，您需要您的 AWS 帳戶中的下列資源：

- Amazon S3 儲存貯體。若要建立儲存貯體，請參閱 Amazon Simple Storage Service 使用者指南中的建立儲存貯體。
- 為 Amazon S3 儲存貯體授予 Amazon Pinpoint 讀取許可的 AWS Identity and Access Management (IAM) 角色。 Amazon S3 若要建立角色，請參閱用於匯入端點或客群的 IAM 角色。

範例

以下範例說明如何將端點定義加入 Amazon S3 儲存貯體，然後將這些端點匯入 Amazon Pinpoint 專案。

含有端點定義的檔案

您加入 Amazon S3 儲存貯體的檔案，可以包含 CSV 格式或換行分隔 JSON 格式的端點定義。如需可用於定義端點的屬性，請參閱 Amazon Pinpoint API 參考中的 EndpointRequest JSON 結構描述。

CSV

您可以匯入在 CSV 檔案中定義的端點，如以下範例所示：

```
ChannelType,Address,Location.Country,Demographic.Platform,Demographic.Make,User.UserId
SMS,12065550182,CN,Android,LG,example-user-id-1
APNS,1a2b3c4d5e6f7g8h9i0j1a2b3c4d5e6f,US,iOS,Apple,example-user-id-2
EMAIL,john.stiles@example.com,US,iOS,Apple,example-user-id-2
```

第一行是標頭，其中包含端點屬性。使用點表示法來指定巢狀屬性，如 `Location.Country`。

隨後的幾行提供標頭中每個屬性的值，以定義端點。

若要在值中包含逗號或雙引號，請將值括在雙引號中，如 `"aaa,bbb"` 所示。

CSV 中的值不支援分行符號。

JSON

您可以匯入以換行分隔的 JSON 檔案中定義的端點，如以下範例所示：

```
{"ChannelType":"SMS","Address":"12065550182","Location":
{"Country":"CN"},"Demographic":{"Platform":"Android","Make":"LG"},"User":
{"UserId":"example-user-id-1"}}
{"ChannelType":"APNS","Address":"1a2b3c4d5e6f7g8h9i0j1a2b3c4d5e6f","Location":
{"Country":"US"},"Demographic":{"Platform":"iOS","Make":"Apple"},"User":
{"UserId":"example-user-id-2"}}
{"ChannelType":"EMAIL","Address":"john.stiles@example.com","Location":
{"Country":"US"},"Demographic":{"Platform":"iOS","Make":"Apple"},"User":
{"UserId":"example-user-id-2"}}
```

在這個格式中，每一列是一個完整的 JSON 物件，其中包含個別端點定義。

匯入任務請求

以下範例說明如何透過將本機檔案上傳到儲存貯體，將端點定義加入 Amazon S3。接著將端點定義匯入 Amazon Pinpoint 專案。

AWS CLI

透過 AWS CLI執行命令，可以使用 Amazon Pinpoint。

Example S3 CP 命令

若要將本機檔案上傳到 Amazon S3 儲存貯體，請使用 Amazon S3 cp 命令：

```
$ aws s3 cp ./endpoints-file s3://bucket-name/prefix/
```

其中：

- ./endpoints-file 是本機檔案的檔案路徑，其中包含端點定義。
- bucket-name/prefix/ 是 Amazon S3 儲存貯體的名稱，也可以當作前置詞，幫助您將儲存貯體中的物件依階層結構整理。例如，有用的字首可能是 `pinpoint/imports/endpoints/`。

Example Create Import Job 命令

若要從 Amazon S3 儲存貯體匯入端點定義，請使用 `create-import-job` 命令：

```
$ aws pinpoint create-import-job \
> --application-id application-id \
> --import-job-request \
> S3Url=s3://bucket-name/prefix/key,\
> RoleArn=iam-import-role-arn,\
> Format=format,\
> RegisterEndpoints=true
```

其中：

- application-id 是您要匯入端點的 Amazon Pinpoint 專案的 ID。
- bucket-name/prefix/key 是 Amazon S3 中的一個位置，包含一個或多個要匯入的物件。位置的結尾可以是個別物件的金鑰，或者結尾可以是符合多個物件的字首。
- iam-import-role-arn 是 IAM 角色的 Amazon Resource Name (ARN)，可向儲存貯體授予 Amazon Pinpoint 讀取存取權限。

- format 可以是 JSON 或 CSV，取決於您用來定義端點的格式。如果 Amazon S3 位置包含各種格式的各種物件，Amazon Pinpoint 只會匯入符合指定格式的物件。
- RegisterEndpoints 可以是 `true` 或 `false`。設為 true 時，匯入任務會在匯入端點定義時在 Amazon Pinpoint 註冊端點。

  RegisterEndpoints 和 DefineSegments 組合

RegisterEndpoints	DefineSegments	Description
true	true	Amazon Pinpoint 將匯入端點，並建立包含端點的客群。
true	false	Amazon Pinpoint 將匯入端點，但不會建立客群。
false	true	Amazon Pinpoint 將匯入端點，並建立包含端點的客群。不會儲存端點，也不會覆寫現有的端點。
false	false	Amazon Pinpoint 將拒絕此請求。

回應包括關於匯入任務的詳細資訊：

```
{
    "ImportJobResponse": {
        "CreationDate": "2018-05-24T21:26:33.995Z",
        "Definition": {
            "DefineSegment": false,
            "ExternalId": "463709046829",
            "Format": "JSON",
            "RegisterEndpoints": true,
            "RoleArn": "iam-import-role-arn",
            "S3Url": "s3://bucket-name/prefix/key"
        },
        "Id": "d5ecad8e417d498389e1d5b9454d4e0c",
        "JobStatus": "CREATED",
        "Type": "IMPORT"
```

```
    }
}
```

回應使用 Id 屬性提供任務 ID。您可以使用此 ID 來檢查匯入任務的目前狀態。

Example Get Import Job 命令

若要檢查匯入任務的目前狀態，請使用 get-import-job 命令：

```
$ aws pinpoint get-import-job \
> --application-id application-id \
> --job-id job-id
```

其中：

- application-id 是匯入任務接受初始化的 Amazon Pinpoint 專案的 ID。
- job-id 是您要檢查的匯入任務 ID。

此命令的回應提供匯入任務的目前狀態：

```
{
    "ImportJobResponse": {
        "ApplicationId": "application-id",
        "CompletedPieces": 1,
        "CompletionDate": "2018-05-24T21:26:45.308Z",
        "CreationDate": "2018-05-24T21:26:33.995Z",
        "Definition": {
            "DefineSegment": false,
            "ExternalId": "463709046829",
            "Format": "JSON",
            "RegisterEndpoints": true,
            "RoleArn": "iam-import-role-arn",
            "S3Url": "s3://s3-bucket-name/prefix/endpoint-definitions.json"
        },
        "FailedPieces": 0,
        "Id": "job-id",
        "JobStatus": "COMPLETED",
        "TotalFailures": 0,
        "TotalPieces": 1,
        "TotalProcessed": 3,
        "Type": "IMPORT"
    }
```

```
}
```

回應使用 JobStatus 屬性提供任務狀態。

適用於 Java 的 AWS SDK

使用 適用於 Java 的 AWS SDK提供的用戶端，可以在 Java 應用程式中使用 Amazon Pinpoint API。

Example Code

若要將有端點定義的檔案上傳到 Amazon S3，請使用 AmazonS3 用戶端的 putObject 方法。

若要將端點匯入 Amazon Pinpoint 專案，請初始化 CreateImportJobRequest 物件。接著將此物件傳遞至 AmazonPinpoint 用戶端的 createImportJob 方法。

```
package com.amazonaws.examples.pinpoint;

import com.amazonaws.AmazonServiceException;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.pinpoint.AmazonPinpoint;
import com.amazonaws.services.pinpoint.AmazonPinpointClientBuilder;
import com.amazonaws.services.pinpoint.model.CreateImportJobRequest;
import com.amazonaws.services.pinpoint.model.CreateImportJobResult;
import com.amazonaws.services.pinpoint.model.Format;
import com.amazonaws.services.pinpoint.model.GetImportJobRequest;
import com.amazonaws.services.pinpoint.model.GetImportJobResult;
import com.amazonaws.services.pinpoint.model.ImportJobRequest;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.AmazonS3Exception;
import java.io.File;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.List;
import java.util.concurrent.TimeUnit;

public class ImportEndpoints {

    public static void main(String[] args) {

        final String USAGE = "\n" +
```

```
                "ImportEndpoints - Adds endpoints to an Amazon Pinpoint application
by: \n" +
                "1.) Uploading the endpoint definitions to an Amazon S3 bucket. \n"
+
                "2.) Importing the endpoint definitions from the bucket to an Amazon
Pinpoint " +
                "application.\n\n" +
                "Usage: ImportEndpoints <endpointsFileLocation> <s3BucketName>
<iamImportRoleArn> " +
                "<applicationId>\n\n" +
                "Where:\n" +
                "  endpointsFileLocation - The relative location of the JSON file
that contains the " +
                "endpoint definitions.\n" +
                "  s3BucketName - The name of the Amazon S3 bucket to upload the
JSON file to. If the " +
                "bucket doesn't exist, a new bucket is created.\n" +
                "  iamImportRoleArn - The ARN of an IAM role that grants Amazon
Pinpoint read " +
                "permissions to the S3 bucket.\n" +
                "  applicationId - The ID of the Amazon Pinpoint application to add
the endpoints to.";

        if (args.length < 1) {
            System.out.println(USAGE);
            System.exit(1);
        }

        String endpointsFileLocation = args[0];
        String s3BucketName = args[1];
        String iamImportRoleArn = args[2];
        String applicationId = args[3];

        Path endpointsFilePath = Paths.get(endpointsFileLocation);
        File endpointsFile = new
File(endpointsFilePath.toAbsolutePath().toString());
        uploadToS3(endpointsFile, s3BucketName);

        importToPinpoint(endpointsFile.getName(), s3BucketName, iamImportRoleArn,
applicationId);

    }

    private static void uploadToS3(File endpointsFile, String s3BucketName) {
```

```
        // Initializes Amazon S3 client.
        final AmazonS3 s3 = AmazonS3ClientBuilder.defaultClient();

        // Checks whether the specified bucket exists. If not, attempts to create
one.
        if (!s3.doesBucketExistV2(s3BucketName)) {
            try {
                s3.createBucket(s3BucketName);
                System.out.format("Created S3 bucket %s.\n", s3BucketName);
            } catch (AmazonS3Exception e) {
                System.err.println(e.getErrorMessage());
                System.exit(1);
            }
        }

        // Uploads the endpoints file to the bucket.
        String endpointsFileName = endpointsFile.getName();
        System.out.format("Uploading %s to S3 bucket %s . . .\n", endpointsFileName,
s3BucketName);
        try {
            s3.putObject(s3BucketName, "imports/" + endpointsFileName,
endpointsFile);
            System.out.println("Finished uploading to S3.");
        } catch (AmazonServiceException e) {
            System.err.println(e.getErrorMessage());
            System.exit(1);
        }
    }

    private static void importToPinpoint(String endpointsFileName, String
s3BucketName,
            String iamImportRoleArn, String applicationId) {

        // The S3 URL that Amazon Pinpoint requires to find the endpoints file.
        String s3Url = "s3://" + s3BucketName + "/imports/" + endpointsFileName;

        // Defines the import job that Amazon Pinpoint runs.
        ImportJobRequest importJobRequest = new ImportJobRequest()
                .withS3Url(s3Url)
                .withRegisterEndpoints(true)
                .withRoleArn(iamImportRoleArn)
                .withFormat(Format.JSON);
        CreateImportJobRequest createImportJobRequest = new CreateImportJobRequest()
```

```
                .withApplicationId(applicationId)
                .withImportJobRequest(importJobRequest);

        // Initializes the Amazon Pinpoint client.
        AmazonPinpoint pinpointClient = AmazonPinpointClientBuilder.standard()
                .withRegion(Regions.US_EAST_1).build();

        System.out.format("Importing endpoints in %s to Amazon Pinpoint application
%s . . .\n",
                endpointsFileName, applicationId);

        try {

            // Runs the import job with Amazon Pinpoint.
            CreateImportJobResult importResult =
pinpointClient.createImportJob(createImportJobRequest);

            String jobId = importResult.getImportJobResponse().getId();
            GetImportJobResult getImportJobResult = null;
            String jobStatus = null;

            // Checks the job status until the job completes or fails.
            do {
                getImportJobResult = pinpointClient.getImportJob(new
GetImportJobRequest()
                        .withJobId(jobId)
                        .withApplicationId(applicationId));
                jobStatus =
getImportJobResult.getImportJobResponse().getJobStatus();
                System.out.format("Import job %s . . .\n", jobStatus.toLowerCase());
                TimeUnit.SECONDS.sleep(3);
            } while (!jobStatus.equals("COMPLETED") && !jobStatus.equals("FAILED"));

            if (jobStatus.equals("COMPLETED")) {
                System.out.println("Finished importing endpoints.");
            } else {
                System.err.println("Failed to import endpoints.");
                System.exit(1);
            }

            // Checks for entries that failed to import.
            // getFailures provides up to 100 of the first failed entries for the
job, if
            // any exist.
```

```
            List<String> failedEndpoints =
 getImportJobResult.getImportJobResponse().getFailures();
            if (failedEndpoints != null) {
                System.out.println("Failed to import the following entries:");
                for (String failedEndpoint : failedEndpoints) {
                    System.out.println(failedEndpoint);
                }
            }

        } catch (AmazonServiceException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }

    }

}
```

HTTP

對 REST API 直接提出 HTTP 請求，可以使用 Amazon Pinpoint。

Example S3 PUT Object 請求

若要將端點定義加入儲存貯體，請使用 Amazon S3 PUT 物件操作，並提供端點定義當作內文：

```
PUT /prefix/key HTTP/1.1
Content-Type: text/plain
Accept: application/json
Host: bucket-name.s3.amazonaws.com
X-Amz-Content-Sha256:
 c430dc094b0cec2905bc88d96314914d058534b14e2bc6107faa9daa12fdff2d
X-Amz-Date: 20180605T184132Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180605/
us-east-1/s3/aws4_request, SignedHeaders=accept;cache-control;content-
length;content-type;host;postman-token;x-amz-content-sha256;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{"ChannelType":"SMS","Address":"2065550182","Location":
{"Country":"CAN"},"Demographic":{"Platform":"Android","Make":"LG"},"User":
{"UserId":"example-user-id-1"}}
```

```
{"ChannelType":"APNS","Address":"1a2b3c4d5e6f7g8h9i0j1a2b3c4d5e6f","Location":
{"Country":"USA"},"Demographic":{"Platform":"iOS","Make":"Apple"},"User":
{"UserId":"example-user-id-2"}}
{"ChannelType":"EMAIL","Address":"john.stiles@example.com","Location":
{"Country":"USA"},"Demographic":{"Platform":"iOS","Make":"Apple"},"User":
{"UserId":"example-user-id-2"}}
```

其中：

- /prefix/key 是在上傳之後將包含端點定義之物件的字首和金鑰名稱。您可以使用字首來組織您的物件階層。例如，有用的字首可能是 `pinpoint/imports/endpoints/`。
- bucket-name 是您要加入端點定義的 Amazon S3 儲存貯體的名稱。

Example POST Import Job 請求

若要從 Amazon S3 儲存貯體匯入端點定義，請向匯入任務資源提出 POST 請求。在您的請求中，包括所需的標頭，並提供 ImportJobRequest JSON 做為內文：

```
POST /v1/apps/application_id/jobs/import HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180605T214912Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180605/
us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-
control;content-length;content-type;host;postman-token;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{
  "S3Url": "s3://bucket-name/prefix/key",
  "RoleArn": "iam-import-role-arn",
  "Format": "format",
  "RegisterEndpoints": true
}
```

其中：

- application-id 是您要匯入端點的 Amazon Pinpoint 專案的 ID。
- bucket-name/prefix/key 是 Amazon S3 中的一個位置，包含一個或多個要匯入的物件。位置的結尾可以是個別物件的金鑰，或者結尾可以是符合多個物件的字首。

- iam-import-role-arn 是 IAM 角色的 Amazon Resource Name (ARN)，可向儲存貯體授予 Amazon Pinpoint 讀取存取權限。
- format 可以是 JSON 或 CSV，取決於您用來定義端點的格式。如果 Amazon S3 位置包含各種格式的各種檔案，Amazon Pinpoint 只會匯入符合指定格式的檔案。

如果您的請求成功，您會收到類似以下的回應：

```
{
    "Id": "a995ce5d70fa44adb563b7d0e3f6c6f5",
    "JobStatus": "CREATED",
    "CreationDate": "2018-06-05T21:49:15.288Z",
    "Type": "IMPORT",
    "Definition": {
        "S3Url": "s3://bucket-name/prefix/key",
        "RoleArn": "iam-import-role-arn",
        "ExternalId": "external-id",
        "Format": "JSON",
        "RegisterEndpoints": true,
        "DefineSegment": false
    }
}
```

回應使用 Id 屬性提供任務 ID。您可以使用此 ID 來檢查匯入任務的目前狀態。

Example GET Import Job 請求

若要檢查匯入任務目前的狀態，請向匯入任務資源提出 GET 請求：

```
GET /v1/apps/application_id/jobs/import/job_id HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180605T220744Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180605/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-control;content-
type;host;postman-token;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache
```

其中：

- application_id 是匯入任務接受初始化的 Amazon Pinpoint 專案的 ID。
- job_id 是您要檢查的匯入任務 ID。

如果您的請求成功，您會收到類似以下的回應：

```
{
    "ApplicationId": "application_id",
    "Id": "70a51b2cf442447492d2c8e50336a9e8",
    "JobStatus": "COMPLETED",
    "CompletedPieces": 1,
    "FailedPieces": 0,
    "TotalPieces": 1,
    "CreationDate": "2018-06-05T22:04:49.213Z",
    "CompletionDate": "2018-06-05T22:04:58.034Z",
    "Type": "IMPORT",
    "TotalFailures": 0,
    "TotalProcessed": 3,
    "Definition": {
        "S3Url": "s3://bucket-name/prefix/key.json",
        "RoleArn": "iam-import-role-arn",
        "ExternalId": "external-id",
        "Format": "JSON",
        "RegisterEndpoints": true,
        "DefineSegment": false
    }
}
```

回應使用 `JobStatus` 屬性提供任務狀態。

相關資訊

如需 Amazon Pinpoint API 中匯入任務資源的詳細資訊 (包括支援的 HTTP 方法和請求參數)，請參閱 Amazon Pinpoint API 參考中的匯入任務。

從 Amazon Pinpoint 匯出端點至 Amazon S3 儲存貯體

若要取得 Amazon Pinpoint 所有的受眾相關資訊，可以匯出屬於專案的端點定義。匯出時，Amazon Pinpoint 會將端點定義置於您指定的 Amazon S3 儲存貯體中。當您想要執行下列動作時，匯出端點很有用：

- 檢視您的用戶端應用程式向 Amazon Pinpoint 註冊的新端點和現有端點的最新資料。
- 將 Amazon Pinpoint 中的端點資料與您自己的客戶關係管理 (CRM) 系統同步。
- 建立相關報告或分析您的客戶資料。

> **Note**
>
> 傳遞至 Amazon S3 儲存貯體的內容可能包含客戶內容。如果您需要刪除匯出至 Amazon S3 儲存貯體的端點資料，您必須在 Amazon S3 中執行此操作。如需移除敏感資料的詳細資訊，請參閱如何清空 S3 儲存貯體？或如何刪除 S3 儲存貯體？。

開始之前

在您可以匯出端點之前，您需要您的 AWS 帳戶中的下列資源：

- Amazon S3 儲存貯體。若要建立儲存貯體，請參閱 Amazon Simple Storage Service 使用者指南中的建立儲存貯體。
- 授予 Amazon Pinpoint 寫入許可的 AWS Identity and Access Management (IAM) 角色Amazon S3。若要建立角色，請參閱用於匯出端點或客群的 IAM 角色。

範例

以下範例說明如何從 Amazon Pinpoint 匯出端點，然後從 Amazon S3 儲存貯體下載這些端點。

AWS CLI

透過 AWS CLI執行命令，可以使用 Amazon Pinpoint。

Example Create Export Job 命令

若要匯出 Amazon Pinpoint 專案中的端點，請使用 `create-export-job` 命令：

```
$ aws pinpoint create-export-job \
> --application-id application-id \
> --export-job-request \
> S3UrlPrefix=s3://bucket-name/prefix/,\
> RoleArn=iam-export-role-arn
```

其中：

- *application-id* 是包含端點的 Amazon Pinpoint 專案的 ID。
- *bucket-name/prefix/* 是 Amazon S3 儲存貯體的名稱，也可以當作前置詞，幫助您將儲存貯體中的物件依階層結構整理。例如，有用的字首可能是 `pinpoint/exports/endpoints/`。
- *iam-export-role-arn* 是 IAM 角色的 Amazon Resource Name (ARN)，可以為 Amazon Pinpoint 授予儲存貯體的寫入存取權限。

此命令的回應提供有關匯出任務的詳細資訊：

```
{
    "ExportJobResponse": {
        "CreationDate": "2018-06-04T22:04:20.585Z",
        "Definition": {
            "RoleArn": "iam-export-role-arn",
            "S3UrlPrefix": "s3://s3-bucket-name/prefix/"
        },
        "Id": "7390e0de8e0b462380603c5a4df90bc4",
        "JobStatus": "CREATED",
        "Type": "EXPORT"
    }
}
```

回應使用 `Id` 屬性提供任務 ID。您可以使用此 ID 來檢查匯出任務的目前狀態。

Example Get Export Job 命令

若要檢查匯出任務的目前狀態，請使用 get-export-job 命令：

```
$ aws pinpoint get-export-job \
> --application-id application-id \
> --job-id job-id
```

其中：

- *application-id* 是您匯出端點的來源 Amazon Pinpoint 專案的 ID。
- *job-id* 是您要檢查的任務 ID。

此命令的回應提供匯出任務的目前狀態：

```
{
    "ExportJobResponse": {
        "ApplicationId": "application-id",
        "CompletedPieces": 1,
        "CompletionDate": "2018-05-08T22:16:48.228Z",
        "CreationDate": "2018-05-08T22:16:44.812Z",
        "Definition": {},
        "FailedPieces": 0,
        "Id": "6c99c463f14f49caa87fa27a5798bef9",
        "JobStatus": "COMPLETED",
        "TotalFailures": 0,
        "TotalPieces": 1,
        "TotalProcessed": 215,
        "Type": "EXPORT"
    }
}
```

回應使用 JobStatus 屬性提供任務狀態。若任務狀態值是 COMPLETED，您可以從 Amazon S3 儲存貯體取得匯出的端點。

Example S3 CP 命令

若要下載您匯出的端點，請使用 Amazon S3 cp 命令：

```
$ aws s3 cp s3://bucket-name/prefix/key.gz /local/directory/
```

其中：

- *bucket-name/prefix/key* 是您匯出端點時，Amazon Pinpoint 在您的儲存貯體加入 .gz 檔案的位置。此檔案包含匯出的端點定義。例如，在 URL https://PINPOINT-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/Exports/example.csv 中，PINPOINT-EXAMPLE-BUCKET 是儲存貯體的名稱，而 Exports/example.csv 是金鑰。如需金鑰的詳細資訊，請參閱《Amazon S3 使用者指南》中的金鑰。
- */local/directory/* 是您要下載端點的本機目錄檔案路徑。

適用於 Java 的 AWS SDK

使用 適用於 Java 的 AWS SDK提供的用戶端，可以在 Java 應用程式中使用 Amazon Pinpoint API。

Example代碼

若要從 Amazon Pinpoint 專案匯出端點，請初始化 `CreateExportJobRequest` 物件。接著將此物件傳遞至 `AmazonPinpoint` 用戶端的 `createExportJob` 方法。

若要從 Amazon Pinpoint 下載匯出的端點，請使用 `AmazonS3` 用戶端的 `getObject` 方法。

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.ExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobRequest;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Request;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Response;
import software.amazon.awssdk.services.s3.model.S3Object;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.concurrent.TimeUnit;
import java.util.stream.Collectors;
```

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.ExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobRequest;
```

```
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Request;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Response;
import software.amazon.awssdk.services.s3.model.S3Object;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.concurrent.TimeUnit;
import java.util.stream.Collectors;

/**
 * To run this code example, you need to create an AWS Identity and Access
 * Management (IAM) role with the correct policy as described in this
 * documentation:
 * https://docs.aws.amazon.com/pinpoint/latest/developerguide/audience-data-
export.html
 *
 * Also, set up your development environment, including your credentials.
 *
 * For information, see this documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */

public class ExportEndpoints {
    public static void main(String[] args) {
        final String usage = """

                This program performs the following steps:

                1. Exports the endpoints to an Amazon S3 bucket.
                2. Downloads the exported endpoints files from Amazon S3.
                3. Parses the endpoints files to obtain the endpoint IDs and prints
 them.
                Usage: ExportEndpoints <applicationId> <s3BucketName>
 <iamExportRoleArn> <path>
```

```
                Where:
                  applicationId - The ID of the Amazon Pinpoint application that has
the endpoint.
                  s3BucketName - The name of the Amazon S3 bucket to export the JSON
file to.\s
                  iamExportRoleArn - The ARN of an IAM role that grants Amazon
Pinpoint write permissions to the S3 bucket.  path - The path where the files
downloaded from the Amazon S3 bucket are written (for example, C:/AWS/).
                """;

       if (args.length != 4) {
           System.out.println(usage);
           System.exit(1);
       }

       String applicationId = args[0];
       String s3BucketName = args[1];
       String iamExportRoleArn = args[2];
       String path = args[3];
       System.out.println("Deleting an application with ID: " + applicationId);

       Region region = Region.US_EAST_1;
       PinpointClient pinpoint = PinpointClient.builder()
               .region(region)
               .build();

       S3Client s3Client = S3Client.builder()
               .region(region)
               .build();

       exportAllEndpoints(pinpoint, s3Client, applicationId, s3BucketName, path,
iamExportRoleArn);
       pinpoint.close();
       s3Client.close();
   }

   public static void exportAllEndpoints(PinpointClient pinpoint,
           S3Client s3Client,
           String applicationId,
           String s3BucketName,
           String path,
           String iamExportRoleArn) {
```

```
        try {
            List<String> objectKeys = exportEndpointsToS3(pinpoint, s3Client,
 s3BucketName, iamExportRoleArn,
                    applicationId);
            List<String> endpointFileKeys = objectKeys.stream().filter(o ->
 o.endsWith(".gz"))
                    .collect(Collectors.toList());
            downloadFromS3(s3Client, path, s3BucketName, endpointFileKeys);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static List<String> exportEndpointsToS3(PinpointClient pinpoint, S3Client
 s3Client, String s3BucketName,
            String iamExportRoleArn, String applicationId) {

        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd-
HH_mm:ss.SSS_z");
        String endpointsKeyPrefix = "exports/" + applicationId + "_" +
 dateFormat.format(new Date());
        String s3UrlPrefix = "s3://" + s3BucketName + "/" + endpointsKeyPrefix +
 "/";
        List<String> objectKeys = new ArrayList<>();
        String key;

        try {
            // Defines the export job that Amazon Pinpoint runs.
            ExportJobRequest jobRequest = ExportJobRequest.builder()
                    .roleArn(iamExportRoleArn)
                    .s3UrlPrefix(s3UrlPrefix)
                    .build();

            CreateExportJobRequest exportJobRequest =
 CreateExportJobRequest.builder()
                    .applicationId(applicationId)
                    .exportJobRequest(jobRequest)
                    .build();

            System.out.format("Exporting endpoints from Amazon Pinpoint application
 %s to Amazon S3 " +
                    "bucket %s . . .\n", applicationId, s3BucketName);
```

```
            CreateExportJobResponse exportResult =
pinpoint.createExportJob(exportJobRequest);
            String jobId = exportResult.exportJobResponse().id();
            System.out.println(jobId);
            printExportJobStatus(pinpoint, applicationId, jobId);

            ListObjectsV2Request v2Request = ListObjectsV2Request.builder()
                    .bucket(s3BucketName)
                    .prefix(endpointsKeyPrefix)
                    .build();

            // Create a list of object keys.
            ListObjectsV2Response v2Response = s3Client.listObjectsV2(v2Request);
            List<S3Object> objects = v2Response.contents();
            for (S3Object object : objects) {
                key = object.key();
                objectKeys.add(key);
            }

            return objectKeys;

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    private static void printExportJobStatus(PinpointClient pinpointClient,
            String applicationId,
            String jobId) {

        GetExportJobResponse getExportJobResult;
        String status;

        try {
            // Checks the job status until the job completes or fails.
            GetExportJobRequest exportJobRequest = GetExportJobRequest.builder()
                    .jobId(jobId)
                    .applicationId(applicationId)
                    .build();

            do {
```

```
                getExportJobResult = pinpointClient.getExportJob(exportJobRequest);
                status =
getExportJobResult.exportJobResponse().jobStatus().toString().toUpperCase();
                System.out.format("Export job %s . . .\n", status);
                TimeUnit.SECONDS.sleep(3);

            } while (!status.equals("COMPLETED") && !status.equals("FAILED"));

            if (status.equals("COMPLETED")) {
                System.out.println("Finished exporting endpoints.");
            } else {
                System.err.println("Failed to export endpoints.");
                System.exit(1);
            }

        } catch (PinpointException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    // Download files from an Amazon S3 bucket and write them to the path location.
    public static void downloadFromS3(S3Client s3Client, String path, String
s3BucketName, List<String> objectKeys) {

        String newPath;
        try {
            for (String key : objectKeys) {
                GetObjectRequest objectRequest = GetObjectRequest.builder()
                        .bucket(s3BucketName)
                        .key(key)
                        .build();

                ResponseBytes<GetObjectResponse> objectBytes =
s3Client.getObjectAsBytes(objectRequest);
                byte[] data = objectBytes.asByteArray();

                // Write the data to a local file.
                String fileSuffix = new
SimpleDateFormat("yyyyMMddHHmmss").format(new Date());
                newPath = path + fileSuffix + ".gz";
                File myFile = new File(newPath);
                OutputStream os = new FileOutputStream(myFile);
                os.write(data);
```

```
            }
            System.out.println("Download finished.");

        } catch (S3Exception | NullPointerException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

如需完整的 SDK 範例，請參閱 GitHub 上的 ExportEndpoints.java。

HTTP

對 REST API 直接提出 HTTP 請求，可以使用 Amazon Pinpoint。

Example POST Export Job 請求

若要匯出 Amazon Pinpoint 專案中的端點，請向匯出任務資源提出 POST 請求：

```
POST /v1/apps/application_id/jobs/export HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180606T001238Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180606/
us-east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-
control;content-length;content-type;host;postman-token;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache

{
  "S3UrlPrefix": "s3://bucket-name/prefix",
  "RoleArn": "iam-export-role-arn"
}
```

其中：

- *application-id* 是包含端點的 Amazon Pinpoint 專案的 ID。
- *bucket-name/prefix* 是 Amazon S3 儲存貯體的名稱，也可以當作前置詞，幫助您將儲存貯體中的物件依階層結構整理。例如，有用的字首可能是 `pinpoint/exports/endpoints/`。
- *iam-export-role-arn* 是 IAM 角色的 Amazon Resource Name (ARN)，可以為 Amazon Pinpoint 授予儲存貯體的寫入存取權限。

此請求的回應提供有關匯出任務的詳細資訊：

```
{
    "Id": "611bdc54c75244bfa51fe7001ddb2e36",
    "JobStatus": "CREATED",
    "CreationDate": "2018-06-06T00:12:43.271Z",
    "Type": "EXPORT",
    "Definition": {
        "S3UrlPrefix": "s3://bucket-name/prefix",
        "RoleArn": "iam-export-role-arn"
    }
}
```

回應使用 Id 屬性提供任務 ID。您可以使用此 ID 來檢查匯出任務的目前狀態。

Example GET Export Jobs 請求

若要檢查匯出任務目前的狀態，請向匯出任務資源提出 GET 請求：

```
GET /v1/apps/application_id/jobs/export/job_id HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: pinpoint.us-east-1.amazonaws.com
X-Amz-Date: 20180606T002443Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20180606/us-
east-1/mobiletargeting/aws4_request, SignedHeaders=accept;cache-control;content-
type;host;postman-token;x-amz-date,
 Signature=c25cbd6bf61bd3b3667c571ae764b9bf2d8af61b875cacced95d1e68d91b4170
Cache-Control: no-cache
```

其中：

- *application-id* 是您匯出端點的來源 Amazon Pinpoint 專案的 ID。
- *job-id* 是您要檢查的任務 ID。

此請求的回應提供匯出任務的目前狀態：

```
{
    "ApplicationId": "application_id",
    "Id": "job_id",
    "JobStatus": "COMPLETED",
```

```
    "CompletedPieces": 1,
    "FailedPieces": 0,
    "TotalPieces": 1,
    "CreationDate": "2018-06-06T00:12:43.271Z",
    "CompletionDate": "2018-06-06T00:13:01.141Z",
    "Type": "EXPORT",
    "TotalFailures": 0,
    "TotalProcessed": 217,
    "Definition": {}
}
```

回應使用 `JobStatus` 屬性提供任務狀態。若任務狀態值是 `COMPLETED`，您可以從 Amazon S3 儲存貯體取得匯出的端點。

相關資訊

若要尋找特定端點的端點 ID，您必須判斷端點所屬的客群，然後從 Amazon Pinpoint 匯出客群。匯出的資料將包含每個端點的端點 ID。您可以使用 Amazon Pinpoint 主控台將客群匯出至檔案。如需匯出客群的詳細資訊，請參閱《Amazon Pinpoint 使用者指南》中的匯出客群。

如需 Amazon Pinpoint API 中匯出任務資源的詳細資訊 (包括支援的 HTTP 方法和請求參數)，請參閱 Amazon Pinpoint API 參考中的匯出任務。

在 Amazon Pinpoint 專案中查詢端點

您可以查閱加入 Amazon Pinpoint 專案的任何單一端點的詳細資訊。這些詳細資訊可以包含您的訊息的目的地地址、訊息管道、有關使用者裝置的資料、有關使用者位置的資料，以及在您的端點中記錄的任何自訂屬性。

若要查詢端點，您需要端點 ID。如果您不知道 ID，您可以改為透過匯出來取得端點資料。若要匯出端點，請參閱the section called “從 Amazon Pinpoint 匯出端點至 Amazon S3 儲存貯體”。

範例

以下範例說明如何透過指定 ID 來查詢個別端點。

AWS CLI

透過 AWS CLI執行命令，可以使用 Amazon Pinpoint。

Example Get Endpoint 命令

若要查詢端點，請使用 get-endpoint 命令：

```
$ aws pinpoint get-endpoint \
> --application-id application-id \
> --endpoint-id endpoint-id
```

其中：

- *application-id* 是包含端點的 Amazon Pinpoint 專案的 ID。
- *endpoint-id* 是您要查詢的端點 ID。

此命令的回應是您刪除的端點的 JSON 定義，如以下範例所示：

```
{
    "EndpointResponse": {
        "Address":
 "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
        "ApplicationId": "application-id",
        "Attributes": {
            "Interests": [
                "Technology",
                "Music",
                "Travel"
            ]
        },
        "ChannelType": "APNS",
        "CohortId": "63",
        "CreationDate": "2018-05-01T17:31:01.046Z",
        "Demographic": {
            "AppVersion": "1.0",
            "Make": "apple",
            "Model": "iPhone",
            "ModelVersion": "8",
            "Platform": "ios",
            "PlatformVersion": "11.3.1",
            "Timezone": "America/Los_Angeles"
        },
        "EffectiveDate": "2018-05-07T19:03:29.963Z",
        "EndpointStatus": "ACTIVE",
        "Id": "example_endpoint",
```

```
        "Location": {
            "City": "Seattle",
            "Country": "US",
            "Latitude": 47.6,
            "Longitude": -122.3,
            "PostalCode": "98121"
        },
        "Metrics": {
            "music_interest_level": 6.0,
            "travel_interest_level": 4.0,
            "technology_interest_level": 9.0
        },
        "OptOut": "ALL",
        "RequestId": "7f546cac-6858-11e8-adcd-2b5a07aab338",
        "User": {
            "UserAttributes": {
                "Gender": "Female",
                "FirstName": "Wang",
                "LastName": "Xiulan",
                "Age": "39"
            },
            "UserId": "example_user"
        }
    }
}
```

適用於 Java 的 AWS SDK

使用 適用於 Java 的 AWS SDK提供的用戶端，可以在 Java 應用程式中使用 Amazon Pinpoint API。

Example代碼

若要查詢端點，請初始化 `GetEndpointRequest` 物件。接著將此物件傳遞至 `AmazonPinpoint` 用戶端的 `getEndpoint` 方法：

```
import com.google.gson.FieldNamingPolicy;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
```

```
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;
```

```
import com.google.gson.FieldNamingPolicy;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class LookUpEndpoint {
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appId> <endpoint>

                Where:
                  appId - The ID of the application to delete.
                  endpoint - The ID of the endpoint.\s
                  """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String endpoint = args[1];
        System.out.println("Looking up an endpoint point with ID: " + endpoint);
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();
```

```
        lookupPinpointEndpoint(pinpoint, appId, endpoint);
        pinpoint.close();
    }

    public static void lookupPinpointEndpoint(PinpointClient pinpoint, String appId,
 String endpoint) {
        try {
            GetEndpointRequest appRequest = GetEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpoint)
                    .build();

            GetEndpointResponse result = pinpoint.getEndpoint(appRequest);
            EndpointResponse endResponse = result.endpointResponse();

            // Uses the Google Gson library to pretty print the endpoint JSON.
            Gson gson = new GsonBuilder()
                    .setFieldNamingPolicy(FieldNamingPolicy.UPPER_CAMEL_CASE)
                    .setPrettyPrinting()
                    .create();

            String endpointJson = gson.toJson(endResponse);
            System.out.println(endpointJson);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
    }
}
```

若要以可讀取格式列印端點資料，此範例會使用 Google GSON 程式庫將 `EndpointResponse` 物件轉換為 JSON 字串。

HTTP

對 REST API 直接提出 HTTP 請求，可以使用 Amazon Pinpoint。

Example GET Endpoint 請求

若要查閱端點，請向端點資源提出 GET 請求：

```
GET /v1/apps/application_id/endpoints/endpoint_id HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

其中：

- *application-id* 是包含端點的 Amazon Pinpoint 專案的 ID。
- *endpoint-id* 是您要查詢的端點 ID。

此請求的回應是端點的 JSON 定義，如以下範例所示：

```
{
    "ChannelType": "APNS",
    "Address": "1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7q8r9s0t1u2v3w4x5y6z7a8b9c0d1e2f",
    "EndpointStatus": "ACTIVE",
    "OptOut": "NONE",
    "RequestId": "b720cfa8-6924-11e8-aeda-0b22e0b0fa59",
    "Location": {
        "Latitude": 47.6,
        "Longitude": -122.3,
        "PostalCode": "98121",
        "City": "Seattle",
        "Country": "US"
    },
    "Demographic": {
        "Make": "apple",
        "Model": "iPhone",
        "ModelVersion": "8",
        "Timezone": "America/Los_Angeles",
        "AppVersion": "1.0",
        "Platform": "ios",
        "PlatformVersion": "11.3.1"
    },
    "EffectiveDate": "2018-06-06T00:58:19.865Z",
    "Attributes": {
        "Interests": [
            "Technology",
            "Music",
            "Travel"
        ]
```

```
    },
    "Metrics": {
        "music_interest_level": 6,
        "travel_interest_level": 4,
        "technology_interest_level": 9
    },
    "User": {},
    "ApplicationId": "application_id",
    "Id": "example_endpoint",
    "CohortId": "39",
    "CreationDate": "2018-06-06T00:58:19.865Z"
}
```

相關資訊

如需 Amazon Pinpoint API 中端點資源的詳細資訊，請參閱 Amazon Pinpoint API 參考中的端點。

使用 Amazon Pinpoint 列出端點 IDs

若要更新或刪除端點，您需要端點 ID。因此若要在 Amazon Pinpoint 專案的所有端點上執行這些操作，第一個步驟是列出屬於該專案的所有端點 ID。然後，您可以重複這些 ID 以便，例如，全域新增屬性或刪除您的專案中的所有端點。

下列範例使用 適用於 Java 的 AWS SDK 並執行下列動作：

1. 從 Amazon Pinpoint 匯出端點中的範例程式碼呼叫範例`exportEndpointsToS3`方法。此方法會從 Amazon Pinpoint 專案匯出端點定義。端點定義會以 gzip 檔案形式加入 Amazon S3 儲存貯體。
2. 下載匯出的 gzip 檔案。
3. 讀取 gzip 檔案並從每個端點的 JSON 定義取得端點 ID。
4. 列印端點 ID 至主控台。
5. 刪除 Amazon Pinpoint 加到 Amazon S3 的檔案即可清理。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import java.util.List;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class ListEndpointIds {
    public static void main(String[] args) {
        final String usage = """

                Usage:    <applicationId> <userId>

                Where:
                   applicationId - The ID of the Amazon Pinpoint application that has
 the endpoint.
                   userId - The user id applicable to the endpoints""";

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String applicationId = args[0];
        String userId = args[1];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listAllEndpoints(pinpoint, applicationId, userId);
        pinpoint.close();
    }
```

```
    public static void listAllEndpoints(PinpointClient pinpoint,
            String applicationId,
            String userId) {

        try {
            GetUserEndpointsRequest endpointsRequest =
 GetUserEndpointsRequest.builder()
                    .userId(userId)
                    .applicationId(applicationId)
                    .build();

            GetUserEndpointsResponse response =
 pinpoint.getUserEndpoints(endpointsRequest);
            List<EndpointResponse> endpoints = response.endpointsResponse().item();

            // Display the results.
            for (EndpointResponse endpoint : endpoints) {
                System.out.println("The channel type is: " + endpoint.channelType());
                System.out.println("The address is  " + endpoint.address());
            }

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

如需完整的 SDK 範例，請參閱 GitHub 上的 ListEndpointIs.java。

管理 Amazon Pinpoint 中的端點數量上限

每位受眾成員最多可以擁有 15 個與其 UserId 相關聯的端點，詳情請參閱 端點配額。若嘗試加入第 16 個端點，可能會收到 BadRequestException，或者系統會移除 EffectiveDate 最舊的端點，視 ChannelType 而定。

新增第 16 個端點

- 如果端點的新管道類型是 SMS、PUSH、VOICE、EMAIL、CUSTOM 或 IN_APP，將傳回 BadRequestException，因為受眾成員的端點數目已達上限。您需要移除與受眾成員相關聯的端點，然後再試一次，詳情請參閱 以程式設計方式從 Amazon Pinpoint 刪除端點。

- 如果端點的新管道類型是 ADM、GCM、APNS、APNS_VOIP、APNS_VOIP_SANDBOX 或 BAIDU：
  - 檢查目前與受眾成員相關聯的端點，是否至少有一個具有 ADM、GCM、APNS、APNS_VOICE、APNS_VOIP_SANDBOX 或 BAIDU 的 ChannelType。如果沒有，則傳回 BadRequestException，且需要移除端點，然後再試一次，請參閱以程式設計方式從 Amazon Pinpoint 刪除端點。
  - 否則，EffectiveDate 最舊的端點會設為 `INACTIVE`，其 ChannelType 是 ADM、GCM、APNS、APNS_VOIP、APNS_VOIP_SANDBOX 或 BAIDU。
    - 從舊端點移除 UserId。
    - 新端點與受眾成員相關聯，且端點數目已達上限。

將狀態設為 `ACTIVE`，並將 UserId 加回端點，即可重新啟用端點。

以程式設計方式從 Amazon Pinpoint 刪除端點

端點代表聯絡您的其中一個客戶的單一方法。每個端點可以是指客戶的電子郵件地址、行動裝置識別符、電話號碼，或是可以接收所傳送訊息的其他目的地類型。在許多轄區，這類資訊可能視為個人所有。若不想再傳訊給特定目的地 (例如無法連接目的地，或客戶關閉帳戶)，可以刪除端點。

範例

以下範例說明如何刪除端點。

AWS CLI

透過 AWS CLI執行命令，可以使用 Amazon Pinpoint。

Example Delete Endpoint 命令

若要刪除端點，請使用 `delete-endpoint` 命令：

```
$ aws pinpoint delete-endpoint \
> --application-id application-id \
> --endpoint-id endpoint-id
```

其中：

- application-id 是包含端點的 Amazon Pinpoint 專案的 ID。

- endpoint-id 是您要刪除的端點 ID。

此命令的回應是您刪除的端點的 JSON 定義。

適用於 Java 的 AWS SDK

使用 適用於 Java 的 AWS SDK提供的用戶端，可以在 Java 應用程式中使用 Amazon Pinpoint API。

Example代碼

若要刪除端點，請使用 AmazonPinpoint 用戶端的 deleteEndpoint 方法。提供 DeleteEndpointRequest 物件做為方法引數：

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class DeleteEndpoint {
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appName> <endpointId >

                Where:
                  appId - The id of the application to delete.
```

```
                  endpointId - The id of the endpoint to delete.
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String endpointId = args[1];
        System.out.println("Deleting an endpoint with id: " + endpointId);
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        deletePinEncpoint(pinpoint, appId, endpointId);
        pinpoint.close();
    }

    public static void deletePinEncpoint(PinpointClient pinpoint, String appId,
 String endpointId) {
        try {
            DeleteEndpointRequest appRequest = DeleteEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpointId)
                    .build();

            DeleteEndpointResponse result = pinpoint.deleteEndpoint(appRequest);
            String id = result.endpointResponse().id();
            System.out.println("The deleted endpoint id  " + id);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
    }
}
```

如需完整的 SDK 範例，請參閱 GitHub 上的 DeleteEndpoint.java。

HTTP

對 REST API 直接提出 HTTP 請求，可以使用 Amazon Pinpoint。

Example DELETE Endpoint 請求

若要刪除端點，請向端點資源提出 DELETE 請求：

```
DELETE /v1/apps/application-id/endpoints/endpoint-id HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

其中：

- application-id 是包含端點的 Amazon Pinpoint 專案的 ID。
- endpoint-id 是您要刪除的端點 ID。

此請求的回應是您刪除的端點的 JSON 定義。

在 Amazon Pinpoint 中建立或匯入客群

使用者客群會根據共用特性來代表使用者子集，例如使用者最近使用您的應用程式的時間，或是他們使用的裝置平台。客群指定哪些使用者會接收行銷活動所傳送的訊息。當您想邀請使用者回到您的應用程式、提供特別優惠，或促進使用者參與度和購買時，請定義客群以吸引適當的對象。

客群建立之後可用在一或多個行銷活動中。行銷活動會將量身訂做的訊息傳遞給客群中的使用者。

如需詳細資訊，請參閱客群。

主題

- 在 Amazon Pinpoint 中建置客群
- 在 Amazon Pinpoint 中匯入客群
- 使用 AWS Lambda 函數自訂 Amazon Pinpoint 客群

在 Amazon Pinpoint 中建置客群

若要聯繫行銷活動的目標對象，請根據您的應用程式所報告的資料來建置客群。例如，若要聯繫最近沒有使用您的應用程式的使用者，您可以為過去 30 天沒有使用您的應用程式的使用者，定義一個客群。

如需更多程式碼範例，請參閱程式碼範例。

使用 建置客群 適用於 Java 的 AWS SDK

以下範例示範如何使用 適用於 Java 的 AWS SDK建置客群。此範例會建立團隊是 Lakers且在過去 30 天內處於作用中狀態的使用者客群。建立客群後，您就可以將其用作行銷活動或旅程的一部分。如需搭配行銷活動使用客群的範例，請參閱 以程式設計方式建立 Amazon Pinpoint 行銷活動。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AttributeDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.AttributeType;
import software.amazon.awssdk.services.pinpoint.model.RecencyDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentBehaviors;
import software.amazon.awssdk.services.pinpoint.model.SegmentDemographics;
import software.amazon.awssdk.services.pinpoint.model.SegmentLocation;
import software.amazon.awssdk.services.pinpoint.model.SegmentDimensions;
```

```
import software.amazon.awssdk.services.pinpoint.model.WriteSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AttributeDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.AttributeType;
import software.amazon.awssdk.services.pinpoint.model.RecencyDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentBehaviors;
import software.amazon.awssdk.services.pinpoint.model.SegmentDemographics;
import software.amazon.awssdk.services.pinpoint.model.SegmentLocation;
import software.amazon.awssdk.services.pinpoint.model.SegmentDimensions;
import software.amazon.awssdk.services.pinpoint.model.WriteSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class CreateSegment {
        public static void main(String[] args) {
                final String usage = """

                                Usage:   <appId>

                                Where:
                                  appId - The application ID to create a segment for.

                                """;
```

```
                if (args.length != 1) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String appId = args[0];
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();

                SegmentResponse result = createSegment(pinpoint, appId);
                System.out.println("Segment " + result.name() + " created.");
                System.out.println(result.segmentType());
                pinpoint.close();
        }

        public static SegmentResponse createSegment(PinpointClient client, String
appId) {
                try {
                        Map<String, AttributeDimension> segmentAttributes = new
HashMap<>();
                        segmentAttributes.put("Team", AttributeDimension.builder()
                                        .attributeType(AttributeType.INCLUSIVE)
                                        .values("Lakers")
                                        .build());

                        RecencyDimension recencyDimension = RecencyDimension.builder()
                                        .duration("DAY_30")
                                        .recencyType("ACTIVE")
                                        .build();

                        SegmentBehaviors segmentBehaviors = SegmentBehaviors.builder()
                                        .recency(recencyDimension)
                                        .build();

                        SegmentDemographics segmentDemographics = SegmentDemographics
                                        .builder()
                                        .build();

                        SegmentLocation segmentLocation = SegmentLocation
                                        .builder()
                                        .build();

                        SegmentDimensions dimensions = SegmentDimensions
```

```
                                        .builder()
                                        .attributes(segmentAttributes)
                                        .behavior(segmentBehaviors)
                                        .demographic(segmentDemographics)
                                        .location(segmentLocation)
                                        .build();

                        WriteSegmentRequest writeSegmentRequest =
 WriteSegmentRequest.builder()
                                        .name("MySegment")
                                        .dimensions(dimensions)
                                        .build();

                        CreateSegmentRequest createSegmentRequest =
 CreateSegmentRequest.builder()
                                        .applicationId(appId)
                                        .writeSegmentRequest(writeSegmentRequest)
                                        .build();

                        CreateSegmentResponse createSegmentResult =
 client.createSegment(createSegmentRequest);
                        System.out.println("Segment ID: " +
 createSegmentResult.segmentResponse().id());
                        System.out.println("Done");
                        return createSegmentResult.segmentResponse();

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
                return null;
        }
}
```

執行此範例時，下列內容會列印至 IDE 的主控台視窗：

```
Segment ID: 09cb2967a82b4a2fbab38fead8d1f4c4
```

如需完整的 SDK 範例，請參閱 GitHub 上的 CreateSegment.java。

在 Amazon Pinpoint 中匯入客群

您可以透過 Amazon Pinpoint，匯入客群所屬端點的相關資訊，以定義使用者客群。端點是單一簡訊目的地，例如行動推送裝置字符、行動電話號碼或電子郵件地址。

如果您已經在 Amazon Pinpoint 以外的地方建立了客群，但您希望透過 Amazon Pinpoint 行銷活動吸引用戶，那麼匯入客群很有用。

您匯入客群時，Amazon Pinpoint 會從 Amazon Simple Storage Service (Amazon S3) 取得客群的端點。匯入之前，您需要將端點加入 Amazon S3，並建立一個允許 Amazon Pinpoint 存取 Amazon S3 的 IAM 角色。接著為 Amazon Pinpoint 提供儲存端點的 Amazon S3 位置，Amazon Pinpoint 會將每個端點加入客群。

若要建立 IAM 角色，請參閱 用於匯入端點或客群的 IAM 角色。如需使用 Amazon Pinpoint 主控台匯入客群的相關資訊，請參閱 Amazon Pinpoint 使用者指南中的匯入客群。

如需更多程式碼範例，請參閱程式碼範例。

使用 匯入客群 適用於 Java 的 AWS SDK

以下範例示範如何使用 適用於 Java 的 AWS SDK匯入客群。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.ImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.ImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.Format;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.ImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.ImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.Format;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
```

```
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class ImportSegment {
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appId> <bucket> <key> <roleArn>\s

                Where:
                  appId - The application ID to create a segment for.
                  bucket - The name of the Amazon S3 bucket that contains the segment
 definitons.
                  key - The key of the S3 object.
                  roleArn - ARN of the role that allows Amazon Pinpoint to access S3.
 You need to set trust management for this to work. See https://docs.aws.amazon.com/
IAM/latest/UserGuide/reference_policies_elements_principal.html
                  """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String bucket = args[1];
        String key = args[2];
        String roleArn = args[3];

        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        ImportJobResponse response = createImportSegment(pinpoint, appId, bucket, key,
 roleArn);
        System.out.println("Import job for " + bucket + " submitted.");
        System.out.println("See application " + response.applicationId() + " for import
 job status.");
        System.out.println("See application " + response.jobStatus() + " for import job
 status.");
        pinpoint.close();
```

```
    }

    public static ImportJobResponse createImportSegment(PinpointClient client,
            String appId,
            String bucket,
            String key,
            String roleArn) {

        try {
            ImportJobRequest importRequest = ImportJobRequest.builder()
                    .defineSegment(true)
                    .registerEndpoints(true)
                    .roleArn(roleArn)
                    .format(Format.JSON)
                    .s3Url("s3://" + bucket + "/" + key)
                    .build();

            CreateImportJobRequest jobRequest = CreateImportJobRequest.builder()
                    .importJobRequest(importRequest)
                    .applicationId(appId)
                    .build();

            CreateImportJobResponse jobResponse = client.createImportJob(jobRequest);
            return jobResponse.importJobResponse();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }
}
```

如需完整的 SDK 範例，請參閱 GitHub 上的 ImportingSegments.java。

使用 AWS Lambda 函數自訂 Amazon Pinpoint 客群

這是公共測試版中某個功能的預先發布文件。內容可能變動。

您可以使用 AWS Lambda 來量身打造 Amazon Pinpoint 行銷活動如何吸引目標受眾。使用 AWS Lambda，您可以在 Amazon Pinpoint 傳送行銷活動訊息的那一刻修改行銷活動的客群。

AWS Lambda 是一種運算服務，您可以用來執行程式碼，而無需佈建或管理伺服器。您可以封裝程式碼，並以 Lambda 函數的形式上傳至 Lambda。Lambda 在調用函數時會執行函數，這可由您手動完成，也可以根據事件自動完成。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》https://docs.aws.amazon.com/lambda/latest/dg/welcome.html。

若要將 Lambda 函數指派給行銷活動，可以使用 Amazon Pinpoint API 中的行銷活動資源，定義行銷活動的 `CampaignHook` 設定。這些設定包括 Lambda 函數名稱，以及 `CampaignHook` 模式，該模式指定 Amazon Pinpoint 是否接收函數的回傳值。

您指派給行銷活動的 Lambda 函數稱為 Amazon Pinpoint 擴充功能。

定義 `CampaignHook` 設定後，Amazon Pinpoint 會在執行行銷活動時，自動調用 Lambda 函數，然後才傳送行銷活動訊息。Amazon Pinpoint 調用該函數時，會提供訊息交付的相關事件資料。此資料包括行銷活動的客群，也就是 Amazon Pinpoint 傳送訊息的目的地端點的清單。

如果將 `CampaignHook` 模式設為 `FILTER`，Amazon Pinpoint 在傳送訊息之前，允許該函數修改和傳回客群。例如該函數可能使用包含 Amazon Pinpoint 外部來源的資料的屬性，更新端點定義。或者，該函數可能根據您的函數程式碼中的條件來移除特定端點，以篩選客群。Amazon Pinpoint 收到您函數修改過的客群後，會使用行銷活動的交付管道，將訊息傳送到客群的每個端點。

透過使用 處理您的客群 AWS Lambda，您可以更好地控制傳送訊息的對象以及這些訊息包含的內容。您可以在傳送行銷活動訊息時，即時修改您的行銷活動。篩選客群可讓您吸引更嚴格定義的一部分客群。新增或更新端點屬性也能讓您提供用於訊息變數的新資料。

> **Note**
>
> 您也可以使用 `CampaignHook` 設定，指派要處理訊息交付的 Lambda 函數。若要透過 Amazon Pinpoint 不支援的自訂頻道 (例如社群媒體平台) 傳遞訊息，這類函數很有用。如需詳細資訊，請參閱使用 Webhook 或 Lambda 函數在 Amazon Pinpoint 中建立自訂頻道。
> 使用 Amazon Pinpoint 調用 Lambda 勾點時，Lambda 函數也必須位於與 Amazon Pinpoint 專案相同的區域。

若要使用 修改行銷活動客群 AWS Lambda，請先建立 函數來處理 Amazon Pinpoint 傳送的事件資料，並傳回修改過的客群。接著指派 Lambda 函數政策，以授權 Amazon Pinpoint 調用函數。最後，透過定義 `CampaignHook` 設定來將函數指派給一或多個行銷活動。

如需更多程式碼範例，請參閱程式碼範例。

事件資料

Amazon Pinpoint 調用 Lambda 函數時，會提供以下承載當作事件資料：

```
{
  "MessageConfiguration": {Message configuration}
  "ApplicationId": ApplicationId,
  "CampaignId": CampaignId,
  "TreatmentId": TreatmentId,
  "ActivityId": ActivityId,
  "ScheduledTime": Scheduled Time,
  "Endpoints": {
    EndpointId: {Endpoint definition}
    . . .
  }
}
```

AWS Lambda 會將事件資料傳遞至您的函數程式碼。事件資料可提供以下屬性：

- `MessageConfiguration` – 其結構與 Amazon Pinpoint API 中訊息資源的 `DirectMessageConfiguration` 物件相同。
- `ApplicationId` – 行銷活動所屬的 Amazon Pinpoint 專案的 ID。
- `CampaignId` – 調用函數的 Amazon Pinpoint 行銷活動的 ID。
- `TreatmentId` – 用於 A/B 測試的行銷活動版本的 ID。
- `ActivityId` – 行銷活動正在執行的活動的 ID。
- `ScheduledTime` – 交付行銷活動訊息的日期和時間 (ISO 8601 格式)。
- `Endpoints` – 將端點 ID 與端點定義建立關聯的對應。每個事件資料承載包含最多 50 個端點。如果行銷活動客群包含超過 50 個端點，Amazon Pinpoint 會重複調用函數，一次最多呼叫 50 個端點，直到所有端點處理完畢為止。

建立 Lambda 函式

若要了解如何建立 Lambda 函數，請參閱 AWS Lambda 開發人員指南中的入門。建立函數時，請注意以下會造成訊息交付失敗的條件：

- Lambda 函數超過 15 秒才傳回修改過的客群。
- Amazon Pinpoint 不能對函數的回傳值解碼。

- 此函數需要 Amazon Pinpoint 嘗試 3 次以上，才能成功調用。

Amazon Pinpoint 只接受函數傳回值中的端點定義。函數無法修改事件資料中的其他元素。

Lambda 函數範例

您的 Lambda 函數處理 Amazon Pinpoint 傳送的事件資料，並傳回修改後的端點，如以下在 Node.js 中編寫的範例處理程序所示：

```
'use strict';

exports.handler = (event, context, callback) => {
    for (var key in event.Endpoints) {
        if (event.Endpoints.hasOwnProperty(key)) {
            var endpoint = event.Endpoints[key];
            var attr = endpoint.Attributes;
            if (!attr) {
                attr = {};
                endpoint.Attributes = attr;
            }
            attr["CreditScore"] = [ Math.floor(Math.random() * 200) + 650];
        }
    }
    console.log("Received event:", JSON.stringify(event, null, 2));
    callback(null, event.Endpoints);
};
```

Lambda 將事件資料當作為 `event` 參數，傳遞給處理常式。

在這個範例中，處理常式會逐一查看 `event.Endpoints` 物件中的每個端點，並將新的屬性 `CreditScore` 新增到端點。`CreditScore` 屬性的值只是隨機數字。

`console.log()` 陳述式會將事件記錄在 CloudWatch Logs 中。

`callback()` 陳述式會將修改過的端點傳回 Amazon Pinpoint。一般而言，`callback` 參數在 Node.js Lambda 函數中並非必要，但在此情境中則是必要，因為 Lambda 函數必須將更新的端點傳回 Amazon Pinpoint。

您的函數必須以事件資料提供的相同格式來傳回端點，這是將端點 ID 與端點定義建立關聯的映射，如下列範例所示：

```
{
```

```
    "eqmj8wpxszeqy/b3vch04sn41yw": {
        "ChannelType": "GCM",
        "Address": "4d5e6f1a2b3c4d5e6f7g8h9i0j1a2b3c",
        "EndpointStatus": "ACTIVE",
        "OptOut": "NONE",
        "Demographic": {
            "Make": "android"
        },
        "EffectiveDate": "2017-11-02T21:26:48.598Z",
        "User": {}
    },
    "idrexqqtn8sbwfex0ouscod0yto": {
        "ChannelType": "APNS",
        "Address": "1a2b3c4d5e6f7g8h9i0j1a2b3c4d5e6f",
        "EndpointStatus": "ACTIVE",
        "OptOut": "NONE",
        "Demographic": {
            "Make": "apple"
        },
        "EffectiveDate": "2017-11-02T21:26:48.598Z",
        "User": {}
    }
}
```

範例函數會修改並傳回事件資料中收到的 `event.Endpoints` 物件。

在您傳回的端點定義中，您可以選擇性包含 `TitleOverride` 和 `BodyOverride` 屬性。

> **Note**
>
> 您使用此解決方案傳送訊息時，只有在端點的 `ChannelType` 屬性值是 `ADM`、`APNS`、`APNS_SANDBOX`、`APNS_VOIP`、`APNS_VOIP_SANDBOX`、`BAIDU`、`GCM` 或 `SMS` 時，Amazon Pinpoint 才會採用 `TitleOverride` 和 `BodyOverride` 屬性。若端點的 `ChannelType` 屬性值是 `EMAIL`，Amazon Pinpoint 不會採用這些屬性。

指派 Lambda 函數政策

使用 Lambda 函數處理端點之前，必須先授權 Amazon Pinpoint 調用 Lambda 函數。若要授予調用許可，請將 Lambda 函數政策指派給 Lambda 函數。Lambda 函數政策是資源型許可政策，指定哪些實體可以使用函數，以及這些實體可以執行哪些動作。

詳情請參閱 AWS Lambda 開發人員指南中的為 AWS Lambda使用資源型政策。

範例函數政策

以下政策授予 Amazon Pinpoint 服務主體對特定行銷活動 (*campaign-id*) 使用 `lambda:InvokeFunction` 動作的許可：

```
{
  "Sid": "sid",
  "Effect": "Allow",
  "Principal": {
    "Service": "pinpoint.us-east-1.amazonaws.com"
  },
  "Action": "lambda:InvokeFunction",
  "Resource": "{arn:aws:lambda:us-east-1:account-id:function:function-name}",
  "Condition": {
    "StringEquals": {
      "AWS:SourceAccount": "111122223333"
    },
    "ArnLike": {
      "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:account-id:apps/application-
id/campaigns/campaign-id"
    }
  }
}
```

您的函數政策需要 `Condition` 區塊，其包含一個 `AWS:SourceArn` 金鑰。此程式碼指明允許哪個 Amazon Pinpoint 行銷活動調用函數。在此範例中，政策會授予僅單一行銷活動的許可。`Condition` 區塊也必須包含 `AWS:SourceAccount`金鑰，以控制哪些 AWS 帳戶可以叫用 動作。

若要編寫較一般的政策，請使用多字元比對萬用字元 (*)。例如您可以使用以下 `Condition` 區塊，允許特定 Amazon Pinpoint 專案 (*application-id*) 中的任何行銷活動調用該函數：

```
...
"Condition": {
  "StringEquals": {
    "AWS:SourceAccount": "111122223333"
  },
  "ArnLike": {
    "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:account-id:apps/application-id/
campaigns/*"
  }
```

```
}
...
```

如果您要將 Lambda 函數設成專案中所有行銷活動的預設函數，建議您以上述方式設定政策的 `Condition` 區塊。如需將 Lambda 函數設為專案中所有行銷活動的預設函數的相關資訊，請參閱 將 Lambda 函數指派給行銷活動。

授予 Amazon Pinpoint 調用許可

您可以使用 AWS Command Line Interface (AWS CLI) 將許可新增至指派給 Lambda 函數的 Lambda 函數政策。若要允許 Amazon Pinpoint 調用特定行銷活動的函數，請使用 Lambda `add-permission` 命令，如以下範例所示：

```
$ aws lambda add-permission \
> --function-name function-name \
> --statement-id sid \
> --action lambda:InvokeFunction \
> --principal pinpoint.us-east-1.amazonaws.com \
> --source-account 111122223333
> --source-arn arn:aws:mobiletargeting:us-east-1:account-id:apps/application-id/
campaigns/campaign-id
```

您可以使用 AWS CLI中的 get-campaigns 命令，尋找您的行銷活動 ID。您也可以使用 get-apps 命令，尋找您的應用程式 ID。

當您執行 Lambda `add-permission` 命令時，Lambda 會傳回以下輸出：

```
{
  "Statement": "{\"Sid\":\"sid\",
    \"Effect\":\"Allow\",
    \"Principal\":{\"Service\":\"pinpoint.us-east-1.amazonaws.com\"},
    \"Action\":\"lambda:InvokeFunction\",
    \"Resource\":\"arn:aws:lambda:us-east-1:111122223333:function:function-name\",
    \"Condition\":
      {\"ArnLike\":
        {\"AWS:SourceArn\":
         \"arn:aws:mobiletargeting:us-east-1:111122223333:apps/application-id/
campaigns/campaign-id\"}}
      {\"StringEquals\":
        {\"AWS:SourceAccount\":
          \"111122223333\"}}}
```

```
}
```

Statement 值是陳述式的 JSON 字串版本，且已新增至 Lambda 函數政策。

將 Lambda 函數指派給行銷活動

您可以將 Lambda 函數指派給個別 Amazon Pinpoint 行銷活動。或者，您可以將 Lambda 函數設為專案中所有行銷活動的預設值 (個別指派函數的行銷活動除外)。

若要將 Lambda 函數指派給個別行銷活動，請使用 Amazon Pinpoint API 建立或更新 Campaign 物件，以及定義其 CampaignHook 屬性。若要將 Lambda 函數設為專案中所有行銷活動的預設值，請建立或更新該專案的 Settings 資源，並定義其 CampaignHook 物件。

在這兩種情況下，設定下列 CampaignHook 屬性：

- LambdaFunctionName – Amazon Pinpoint 傳送行銷活動訊息之前，調用的 Lambda 函數的名稱或 ARN。
- Mode – 設為 FILTER。在此模式下，Amazon Pinpoint 調用函數並等待函數傳回修改過的端點。等到收到端點，Amazon Pinpoint 再傳送訊息。Amazon Pinpoint 等待最多 15 秒，才確認訊息交付失敗。

Amazon Pinpoint 在傳送行銷活動訊息之前，會使用為行銷活動定義的 CampaignHook 設定，調用指定的 Lambda 函數。Amazon Pinpoint 等待從函數接收修改過的端點。如果 Amazon Pinpoint 收到已更新的端點，會使用已更新的端點資料，繼續訊息交付。

以程式設計方式建立 Amazon Pinpoint 行銷活動

若要提升應用程式與其使用者的互動，請使用 Amazon Pinpoint 建立並管理可觸及特定使用者客群的推播通知行銷活動。

例如，您的行銷活動可能會邀請最近未執行應用程式的使用者回到您的應用程式，或提供特殊促銷活動給最近沒有購買的使用者。

行銷活動會傳送量身訂做的訊息到您指定的使用者客群。行銷活動可以將訊息傳送給客群中的所有使用者，或者您可以分配 holdout 值，這是未收到訊息的使用者百分比。

您可以設定行銷活動的排程來傳送訊息一次或以重複的頻率傳送訊息 (例如，每週一次)。若要避免使用者在不方便的時候收到您的訊息，排程可以包含不會傳送訊息的安靜時段。

若要使用替代的行銷活動策略進行測試，請將您的行銷活動設定為 A/B 測試。A/B 測試包含訊息或排程兩個或多個的處理方法。處理方法是您的訊息或排程的變化。隨著您的使用者回應行銷活動，您可以檢視行銷活動分析以比較每個處理的有效性。

如需詳細資訊，請參閱《Amazon Pinpoint REST API 指南》中的行銷活動或《Amazon Pinpoint 使用者指南》中的行銷活動。

建立標準 Amazon Pinpoint 行銷活動

標準行銷活動會根據您定義的排程，傳送自訂推送通知到指定的客群。以下範例示範如何使用 適用於 Java 的 AWS SDK建立行銷活動。如需建立要傳入的客群範例，請參閱 在 Amazon Pinpoint 中建置客群。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.Message;
import software.amazon.awssdk.services.pinpoint.model.Schedule;
import software.amazon.awssdk.services.pinpoint.model.Action;
import software.amazon.awssdk.services.pinpoint.model.MessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.WriteCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import software.amazon.awssdk.regions.Region;
```

```
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.Message;
import software.amazon.awssdk.services.pinpoint.model.Schedule;
import software.amazon.awssdk.services.pinpoint.model.Action;
import software.amazon.awssdk.services.pinpoint.model.MessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.WriteCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class CreateCampaign {
    public static void main(String[] args) {

        final String usage = """

                Usage:   <appId> <segmentId>

                Where:
                  appId - The ID of the application to create the campaign in.
                  segmentId - The ID of the segment to create the campaign from.
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String segmentId = args[1];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        createPinCampaign(pinpoint, appId, segmentId);
        pinpoint.close();
```

```
    }

    public static void createPinCampaign(PinpointClient pinpoint, String appId, String
segmentId) {
        CampaignResponse result = createCampaign(pinpoint, appId, segmentId);
        System.out.println("Campaign " + result.name() + " created.");
        System.out.println(result.description());
    }

    public static CampaignResponse createCampaign(PinpointClient client, String appID,
String segmentID) {

        try {
            Schedule schedule = Schedule.builder()
                    .startTime("IMMEDIATE")
                    .build();

            Message defaultMessage = Message.builder()
                    .action(Action.OPEN_APP)
                    .body("My message body.")
                    .title("My message title.")
                    .build();

            MessageConfiguration messageConfiguration = MessageConfiguration.builder()
                    .defaultMessage(defaultMessage)
                    .build();

            WriteCampaignRequest request = WriteCampaignRequest.builder()
                    .description("My description")
                    .schedule(schedule)
                    .name("MyCampaign")
                    .segmentId(segmentID)
                    .messageConfiguration(messageConfiguration)
                    .build();

            CreateCampaignResponse result =
client.createCampaign(CreateCampaignRequest.builder()
                    .applicationId(appID)
                    .writeCampaignRequest(request).build());

            System.out.println("Campaign ID: " + result.campaignResponse().id());
            return result.campaignResponse();

        } catch (PinpointException e) {
```

```
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }

        return null;
    }
}
```

執行此範例時，下列內容會列印至 IDE 的主控台視窗：

```
Campaign ID: b1c3de717aea4408a75bb3287a906b46
```

如需完整的 SDK 範例，請參閱 GitHub 上的 CreateCampaign.java。

使用 建立 A/B 測試 Amazon Pinpoint 行銷活動 適用於 Java 的 AWS SDK

A/B 測試行銷活動的行為類似於標準行銷活動，但可讓您為行銷活動訊息或排程定義不同的處理方式。A/B 測試包含訊息或排程兩個或多個的處理方法。處理方法是您的訊息或排程的變化。隨著您的使用者回應行銷活動，您可以檢視行銷活動分析以比較每個處理的有效性。

以下範例示範如何使用 適用於 Java 的 AWS SDK建立 A/B 測試行銷活動。

```
import com.amazonaws.services.pinpoint.AmazonPinpointClient;
import com.amazonaws.services.pinpoint.model.Action;
import com.amazonaws.services.pinpoint.model.CampaignResponse;
import com.amazonaws.services.pinpoint.model.CreateCampaignRequest;
import com.amazonaws.services.pinpoint.model.CreateCampaignResult;
import com.amazonaws.services.pinpoint.model.Message;
import com.amazonaws.services.pinpoint.model.MessageConfiguration;
import com.amazonaws.services.pinpoint.model.Schedule;
import com.amazonaws.services.pinpoint.model.WriteCampaignRequest;
import com.amazonaws.services.pinpoint.model.WriteTreatmentResource;

import java.util.ArrayList;
import java.util.List;

public class PinpointCampaignSample {

    public CampaignResponse createAbCampaign(AmazonPinpointClient client, String appId,
 String segmentId) {
```

```
        Schedule schedule = new Schedule()
                .withStartTime("IMMEDIATE");

        // Default treatment.
        Message defaultMessage = new Message()
                .withAction(Action.OPEN_APP)
                .withBody("My message body.")
                .withTitle("My message title.");

        MessageConfiguration messageConfiguration = new MessageConfiguration()
                .withDefaultMessage(defaultMessage);

        // Additional treatments
        WriteTreatmentResource treatmentResource = new WriteTreatmentResource()
                .withMessageConfiguration(messageConfiguration)
                .withSchedule(schedule)
                .withSizePercent(40)
                .withTreatmentDescription("My treatment description.")
                .withTreatmentName("MyTreatment");

        List<WriteTreatmentResource> additionalTreatments = new
ArrayList<WriteTreatmentResource>();
        additionalTreatments.add(treatmentResource);

        WriteCampaignRequest request = new WriteCampaignRequest()
                .withDescription("My description.")
                .withSchedule(schedule)
                .withSegmentId(segmentId)
                .withName("MyCampaign")
                .withMessageConfiguration(messageConfiguration)
                .withAdditionalTreatments(additionalTreatments)
                .withHoldoutPercent(10); // Hold out of A/B test

        CreateCampaignRequest createCampaignRequest = new CreateCampaignRequest()
                .withApplicationId(appId).withWriteCampaignRequest(request);

        CreateCampaignResult result = client.createCampaign(createCampaignRequest);

        System.out.println("Campaign ID: " + result.getCampaignResponse().getId());

        return result.getCampaignResponse();
    }
```

```
}
```

執行此範例時，下列內容會列印至 IDE 的主控台視窗：

```
Campaign ID: b1c3de717aea4408a75bb3287a906b46
```

管理 Amazon Pinpoint 資源標籤

標籤是您選擇性定義並與 AWS 資源建立關聯的標籤，包括特定類型的 Amazon Pinpoint 資源。標籤可協助您以不同方式分類和管理資源，例如透過用途、擁有者、環境或其他條件。例如，您可以使用標籤來套用政策或自動化，或識別受制於特定合規要求的資源。您可以將標籤加入以下類型的 Amazon Pinpoint 資源：

- 行銷活動
- 訊息範本
- 專案 (應用程式)
- 客群

資源最多可以擁有 50 個標籤。每個標籤皆包含由您定義的必要「標籤金鑰」與選用「標籤值」。標籤金鑰是一般標籤，做為更特定標籤值的類別。標籤值是標籤金鑰的描述項。

標籤金鑰最多可包含 128 個字元。標籤值最多可包含 256 個字元。這些字元可以是 Unicode 字母、數字、空格或以下其中一個符號：_、:、/、=、+、-。以下額外限制適用於標籤：

- 標籤鍵與值皆區分大小寫。
- 對於每個相關聯的資源，每個標籤金鑰必須是唯一的，而且只能有一個值。
- 字`aws:`首保留供 使用 AWS；您無法在定義的任何標籤索引鍵或值中使用。此外，您無法編輯或移除使用此字首的標籤金鑰或值。使用此字首的標籤不會計入每個資源 50 個標籤的配額。
- 您無法更新或刪除僅以其標籤為基礎的資源。您也須指定 Amazon Resource Name (ARN) 或資源 ID，取決於您使用的操作。
- 您可以將標籤與公有或共用資源建立關聯。不過，這些標籤僅適用於您的帳戶 AWS ，不適用於共用資源的任何其他帳戶。此外，這些標籤僅適用於位於您 AWS 帳戶指定 AWS 區域中的資源。

若要從 Amazon Pinpoint 資源新增、顯示、更新和移除標籤金鑰和值，您可以使用 AWS Command Line Interface (AWS CLI)、Amazon Pinpoint API、 AWS Resource Groups 標記 API 或 AWS SDK。若要管理您 AWS 帳戶特定 AWS 區域中所有 AWS 資源 （ 包括 Amazon Pinpoint 資源 ） 的標籤索引鍵和值，請使用AWS Resource Groups 標記 API。

如需可用於管理 Amazon Pinpoint 資源的 CLI 命令的詳細資訊，請參閱 AWS CLI 命令參考的 Amazon Pinpoint 部分。

如需深入了解 Amazon Pinpoint API 中的資源 (包括支援的 HTTP(S) 方法、參數和結構描述) 的詳細資訊，請參閱 Amazon Pinpoint API 參考。

在 IAM 政策和 API 操作中使用 Amazon Pinpoint 標籤

開始實作標籤後，您可以將標籤型資源層級許可套用至 AWS Identity and Access Management (IAM) 政策和 API 操作。這包括支援在建立資源時將標籤新增至資源的作業。透過以這種方式使用標籤，您可以對 AWS 帳戶中的哪些群組和使用者具有建立和標記資源的許可，以及哪些群組和使用者具有更廣泛建立、更新和移除標籤的許可，實作精細的控制。

例如您可以建立一個政策，讓使用者針對所有 Amazon Pinpoint 資源擁有完全存取權，其名稱是資源 `Owner` 標籤中的值：

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Sid": "ModifyResourceIfOwner",
         "Effect": "Allow",
         "Action": "mobiletargeting:*",
         "Resource": "*",
         "Condition": {
            "StringEqualsIgnoreCase": {
               "aws:ResourceTag/Owner": "${aws:username}"
            }
         }
      }
   ]
}
```

如果您定義標籤型、資源層級許可，則許可會立即生效。這表示您的資源一旦建立就會更安全，而且您可以快速開始強制使用新資源的標籤。您也可以使用資源層級許可，以控制哪些標籤金鑰和值可以與新的和現有的資源相關聯。詳情請參閱 AWS IAM 使用者指南中的使用標籤控制存取權。

以程式設計方式將標籤新增至 Amazon Pinpoint 資源

以下範例說明如何使用 AWS CLI 和 Amazon Pinpoint REST API，將標籤加入 Amazon Pinpoint 資源。您也可以使用任何支援的 AWS SDK 將標籤新增至資源。

若要在單一操作中將標籤新增至多個 Amazon Pinpoint 資源，請使用 AWS CLI 或 AWS Resource Groups 標記 API 的資源群組標記操作。

使用 API 新增標籤

若要使用 Amazon Pinpoint REST API 建立新資源並將標籤加入其中，請將 POST 請求傳送到適當的資源 URI。在請求內文中包含 `tags` 參數和值。以下範例說明如何在建立新專案時指定標籤。

```
POST /v1/apps HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/x-www-form-urlencoded
Accept: application/json
Cache-Control: no-cache

{
   "Name":"MyProject",
   "tags":{
      "key1":"value1"
   }
}
```

若要將標籤新增到現有資源，請將 POST 請求傳送到標籤 URI。在 URI 中包含資源的 Amazon Resource Name (ARN)。ARN 應為 URL 編碼。在請求的內文中，附上 `tags` 參數和值，如以下範例所示。

```
POST /v1/tags/resource-arn HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache

{
   "tags":{
      "key1":"value1"
   }
}
```

使用 新增標籤 AWS CLI

若要使用 建立新的資源並新增標籤 AWS CLI，請使用適當的資源`create`命令。包括 `tags` 參數和值。以下範例說明如何在建立新專案時指定標籤。

Linux, macOS, or Unix

```
$ aws pinpoint create-app \
  --create-application-request '{
    "Name":"MyProject",
    "tags": {
      "key1":"value1",
      "key2":"value2"
    }
  }'
```

Windows Command prompt

```
C:\> aws pinpoint create-app ^
     --create-application-request Name=MyProject,tags={key1=value1,key2=value2}
```

在上述範例中，執行下列動作：

- 將 *MyProject* 替換成您要為專案指定的名稱。
- 將 *key1* 和 *key2* 替換成您要加入資源的標籤的索引鍵。
- 將 *value1* 和 *value2* 替換成您要為各索引鍵加入的標籤的值。

如需可建立 Amazon Pinpoint 資源的命令的相關資訊，請參閱 AWS CLI 命令參考。

若要新增標籤至現有的資源，請使用 `tag-resource` 命令，然後針對所需參數指定正確的值：

Linux, macOS, or Unix

```
$ aws pinpoint tag-resource \
  --resource-arn resource-arn \
  --tags-model '{
    "tags": {
      "key1":"value1",
      "key2":"value2"
```

```
    }
  }'
```

Windows Command Prompt

```
C:\> aws pinpoint tag-resource ^
     --resource-arn resource-arn ^
     --tags-model tags={key1=value1,key2=value2}
```

在上述範例中，執行下列動作：

- 將 *resource-arn* 替換成您要新增標籤的資源的 Amazon Resource Name (ARN)。
- 將 *key1* 和 *key2* 替換成您要加入資源的標籤的索引鍵。
- 將 *value1* 和 *value2* 替換成您要為各索引鍵加入的標籤的值。

以程式設計方式顯示 Amazon Pinpoint 資源的標籤

以下範例顯示如何使用 AWS CLI 及 Amazon Pinpoint REST API，藉以顯示所有與 Amazon Pinpoint 資源相關聯的所有標籤 (金鑰和值) 的清單。您也可以使用任何支援的 AWS SDK 來顯示與資源相關聯的標籤。

使用 API 顯示標籤

若要使用 Amazon Pinpoint REST API 顯示與特定資源相關聯的所有標籤，請將 GET 請求傳送到標籤 URI，並在 URI 中加入資源的 Amazon Resource Name (ARN)。ARN 應為 URL 編碼。例如，以下請求會擷取所有與指定行銷活動相關聯的標籤 (*resource-arn*)：

```
GET /v1/tags/resource-arn HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

JSON 對於請求的回應包含一個 `tags` 物件。`tags` 物件會列出與行銷活動相關聯的所有標籤金鑰和值。

若要顯示與相同類型的多個資源相關聯的所有標籤，請針對該資源類型，將 GET 請求傳送至適當的 URI。例如，以下請求會擷取指定專案 (*application-id*) 中所有行銷活動的相關資訊：

```
GET /v1/apps/application-id/campaigns HTTP/1.1
Host: pinpoint.us-east-1.amazonaws.com
Content-Type: application/json
Accept: application/json
Cache-Control: no-cache
```

JSON 對於請求的回應會列出所有在專案中的行銷活動。每個行銷活動的 tags 物件會列出與行銷活動相關聯的所有標籤金鑰和值。

使用 顯示標籤 AWS CLI

若要使用 AWS CLI 顯示與特定資源相關聯的標籤清單，請執行 list-tags-for-resource命令，並為 resource-arn 參數指定資源的 Amazon Resource Name (ARN)，如下列範例所示。

Linux, macOS, or Unix

```
$ aws pinpoint list-tags-for-resource \
  --resource-arn resource-arn
```

Windows Command Prompt

```
C:\> aws pinpoint list-tags-for-resource ^
     --resource-arn resource-arn
```

若要顯示具有標籤的所有 Amazon Pinpoint 資源清單，以及與每個資源相關聯的所有標籤，請使用 AWS Resource Groups 標記 API 的 get-resources 命令。將 resource-type-filters 參數設為 mobiletargeting，如以下範例所示。

Linux, macOS, or Unix

```
$ aws resourcegroupstaggingapi get-resources \
     --resource-type-filters "mobiletargeting"
```

Windows Command Prompt

```
C:\> aws resourcegroupstaggingapi get-resources ^
     --resource-type-filters "mobiletargeting"
```

命令的輸出是一份 ARN 清單，列出具有標籤的所有 Amazon Pinpoint 資源。此清單包含與每個資源相關聯的所有標籤金鑰和值。

以程式設計方式更新或覆寫 Amazon Pinpoint 資源的標籤

有多種方法可以更新 (覆寫) Amazon Pinpoint 資源的標籤。更新標籤的最佳方式取決於：

- 您要更新其標籤的資源類型。
- 您想要更新一個或多個資源的標籤。
- 您想要更新標籤金鑰、標籤值，還是兩者。

若要同時更新 Amazon Pinpoint 專案或多個資源的標籤，請使用 AWS CLI 或 標記 AWS Resource Groups API 的資源群組標記操作。Amazon Pinpoint API 目前無法為這兩個任務，提供直接支援。

若要更新一個資源的標籤，可以使用 Amazon Pinpoint API 移除目前的標籤和新增標籤。

以程式設計方式從 Amazon Pinpoint 資源移除標籤

以下範例說明如何使用 AWS CLI 和 Amazon Pinpoint REST API，從 Amazon Pinpoint 資源移除標籤 (索引鍵和值)。您也可以使用任何支援的 AWS SDK 從資源中移除標籤。

若要在單一操作中移除多個 Amazon Pinpoint 資源的標籤，請使用 AWS CLI 或 AWS Resource Groups 標記 API 的資源群組標記操作。若要從資源中只移除特定的標籤值 (而非標籤金鑰)，您可以更新資源的標籤。

使用 API 移除標籤

若要使用 Amazon Pinpoint REST API 從資源移除標籤，請傳送 DELETE 請求到標籤 URI。在 URI 中，加入要從中移除標籤之資源的 Amazon Resource Name (ARN)，然後是要移除的 `tagKeys` 參數和標籤。例如：

```
https://endpoint/v1/tags/resource-arn?tagKeys=key
```

其中：

- *##*是託管資源的 AWS 區域的 Amazon Pinpoint 端點。
- *resource-arn* 是您要從中移除標籤之資源的 ARN。

- *key* 是您要從資源中移除的標籤。

所有參數都必須針對 URL 進行編碼。

若要從資源中移除多個標籤金鑰及其關聯值，請對每個要移除的額外標籤附加 tagKeys 參數和引數，並以 & 符號區隔。例如：

```
https://endpoint/v1/tags/resource-arn?tagKeys=key1&tagKeys=key2
```

所有參數都必須針對 URL 進行編碼。

使用 移除標籤 AWS CLI

若要使用 從資源移除標籤 AWS CLI，請執行 untag-resource命令。加入 tag-keys 參數和引數，如以下範例所示。

Linux, macOS, or Unix

```
$ aws pinpoint untag-resource \
  --resource-arn resource-arn \
  --tag-keys key1 key2
```

Windows Command Prompt

```
C:\> aws pinpoint untag-resource ^
     --resource-arn resource-arn ^
     --tag-keys key1 key2
```

在上述範例中，進行下列變更：

- 將 *resource-arn* 替換成您要移除標籤的資源的 ARN。
- 將 *key1* 和 *key2* 替換成您要移除資源的標籤的索引鍵。

將 Amazon Pinpoint 與您的應用程式整合

將 Amazon Pinpoint 與您的用戶端程式碼整合，以了解並吸引使用者。

在您整合 且您的使用者啟動應用程式後，它會連線到 Amazon Pinpoint 服務來新增或更新端點。端點代表您可以傳訊的目的地—例如使用者裝置、電子郵件地址或電話號碼。

然後，您的應用程式可以提供用量資料或事件。在 Amazon Pinpoint 主控台檢視事件資料，了解您有多少使用者、他們多常使用您的應用程式、何時使用等資訊。

在您的應用程式提供端點和事件之後，您可以使用此資訊來為特定對象或客群量身打造簡訊行銷活動。(您也可以直接傳送訊息給一份簡單的收件人清單，而不需要建立行銷活動。)

使用本節的主題，將 Amazon Pinpoint 與行動或 Web 應用程式整合。這些主題包含了與 JavaScript、Android、Swift 或 Flutter 應用程式整合的程式碼範例和程序。若要開始整合您的應用程式，請參閱the section called “使用 Amplify 連接您的前端應用程式”。

您可以在用戶端以外的地方，使用支援的 AWS SDK 或 Amazon Pinpoint API，匯入端點、匯出事件資料、定義客戶客群、建立和執行行銷活動等。

主題

- 搭配 AWS SDK 使用 Amazon Pinpoint
- 使用 AWS Amplify 將您的前端應用程式連線至 Amazon Pinpoint
- 在應用程式中註冊 Amazon Pinpoint 端點
- 在您的應用程式中報告 Amazon Pinpoint 事件

搭配 AWS SDK 使用 Amazon Pinpoint

AWS 軟體開發套件 (SDKs) 適用於許多熱門的程式設計語言。每個 SDK 都提供 API、程式碼範例和說明文件，讓開發人員能夠更輕鬆地以偏好的語言建置應用程式。

SDK 文件	代碼範例
適用於 C++ 的 AWS SDK	適用於 C++ 的 AWS SDK 程式碼範例
AWS CLI	AWS CLI 程式碼範例

SDK 文件	代碼範例
適用於 Go 的 AWS SDK	適用於 Go 的 AWS SDK 程式碼範例
適用於 Java 的 AWS SDK	適用於 Java 的 AWS SDK 程式碼範例
適用於 JavaScript 的 AWS SDK	適用於 JavaScript 的 AWS SDK 程式碼範例
適用於 Kotlin 的 AWS SDK	適用於 Kotlin 的 AWS SDK 程式碼範例
適用於 .NET 的 AWS SDK	適用於 .NET 的 AWS SDK 程式碼範例
適用於 PHP 的 AWS SDK	適用於 PHP 的 AWS SDK 程式碼範例
AWS Tools for PowerShell	AWS Tools for PowerShell 程式碼範例
適用於 Python (Boto3) 的 AWS SDK	適用於 Python (Boto3) 的 AWS SDK 程式碼範例
適用於 Ruby 的 AWS SDK	適用於 Ruby 的 AWS SDK 程式碼範例
適用於 Rust 的 AWS SDK	適用於 Rust 的 AWS SDK 程式碼範例
適用於 SAP ABAP 的 AWS SDK	適用於 SAP ABAP 的 AWS SDK 程式碼範例
適用於 Swift 的 AWS SDK	適用於 Swift 的 AWS SDK 程式碼範例

如需 Amazon Pinpoint 專用範例，請參閱 使用 AWS SDKs Amazon Pinpoint 程式碼範例。

> **可用性範例**
>
> 找不到所需的內容嗎？ 請使用本頁面底部的提供意見回饋連結申請程式碼範例。

使用 AWS Amplify 將您的前端應用程式連線至 Amazon Pinpoint

使用 AWS Amplify 將您的應用程式與 整合 AWS。如需 Swift 應用程式的詳細資訊，請參閱 Amplify for Swift 文件中的開始使用。如果是 Android 應用程式，請參閱 Amplify for Android SDK 文件中的開始使用。如需 React Native 應用程式的詳細資訊，請參閱 Amplify JavaScript 文件中的開始使用。如需 Flutter 應用程序的詳細資訊，請參閱 Flutter SDK 文件中的開始使用。這些主題可協助您：

- 設定您的後端資源。
- 使用 Amplify 程式庫將您的應用程式連接到後端資源。

若要進一步了解如何將前端應用程式連線至 Amazon Pinpoint 進行分析、應用程式內訊息和推播通知，請參閱 AWS Amplify。

下一步驟

將 AWS Amplify 與您的應用程式整合後，請更新您的程式碼，將使用者的裝置註冊為端點。如需詳細資訊，請參閱在應用程式中註冊 Amazon Pinpoint 端點。

在應用程式中註冊 Amazon Pinpoint 端點

使用者啟動工作階段時 (例如透過啟動行動應用程式)，您的行動或 Web 應用程式可以使用 Amazon Pinpoint 自動註冊 (或更新) 端點。端點代表使用者啟動工作階段所針對的裝置。它包含屬性來描述裝置，也可以包含您定義的自訂屬性。端點也可以代表與客戶通訊的其他方法，例如電子郵件地址或行動電話號碼。

當您的應用程式註冊端點之後，您可以根據端點的屬性來區隔對象。然後，您可以透過量身訂做的簡訊行銷活動，與這些客群交流。您也可以使用 Amazon Pinpoint 主控台中的分析頁面，檢視有關端點註冊和活動的圖表，例如新端點和每日作用中端點。

您可以將單一使用者 ID 指派給多個端點。使用者 ID 代表單一使用者，而每個獲指派使用者 ID 的端點代表使用者的其中一個裝置。您將使用者 ID 指派給端點之後，可以在主控台檢視使用者活動相關圖表，例如每日作用中使用者和每月作用中使用者。

開始之前

如果您尚未這麼做，請整合適用於 Android 或 iOS 的 AWS Mobile SDK，或整合 AWS Amplify JavaScript 程式庫與您的應用程式。如需詳細資訊，請參閱使用 AWS Amplify 將您的前端應用程式連線至 Amazon Pinpoint。

使用適用於 Android 或 iOS AWS 的行動 SDKs註冊端點

您可以使用適用於 Android 或 iOS 的 AWS Mobile SDK 來註冊和自訂端點。 SDKs 如需詳細資訊及檢視程式碼範例，請參閱下列文件：

- Android 開發套件文件中的在您的應用程式中註冊端點。
- iOS 開發套件文件中的在您的應用程式中註冊端點。

向 AWS Amplify JavaScript 程式庫註冊端點

您可以使用 AWS Amplify JavaScript 程式庫在應用程式中註冊和更新端點。如需詳細資訊，以及檢視程式碼範例，請參閱 AWS Amplify JavaScript 文件中的更新端點。

後續步驟

在您更新應用程式以註冊端點後，當使用者啟動您的應用程式時，裝置資訊和自訂屬性會提供給 Amazon Pinpoint。您可以使用此資訊來定義對象客群。您也可以使用 主控台來查看端點指標和獲指派使用者 IDs的使用者。您也可以完成 中的步驟在您的應用程式中報告 Amazon Pinpoint 事件，以更新應用程式來報告用量資料。

在您的應用程式中報告 Amazon Pinpoint 事件

在行動或 Web 應用程式中，您可以使用 AWS Mobile SDKs或 Amazon Pinpoint 事件 API 向 Amazon Pinpoint 報告用量資料或事件。您可以報告事件來擷取資訊，例如，工作階段時間、使用者的購買行為、登入嘗試、或您需要的任何自訂事件類型。

您的應用程式回報事件之後，您可以在 Amazon Pinpoint 主控台檢視分析。Analytics (分析) 頁面上的圖表提供許多使用者行為層面的指標。詳情請參閱 Amazon Pinpoint 使用者指南中的 Amazon Pinpoint 分析圖表參考。

若要在 Amazon Pinpoint 以外的地方分析或儲存資料，可以設定 Amazon Pinpoint 將資料串流到 Amazon Kinesis。如需詳細資訊，請參閱使用 Amazon Pinpoint 透過 Kinesis 和 Firehose 串流應用程式事件資料。

透過使用 AWS Mobile SDKs和 AWS Amplify JavaScript 程式庫，您可以呼叫 Amazon Pinpoint API 來報告下列類型的事件：

工作階段事件

指出使用者開啟和關閉您的應用程式的時間和頻率。

應用程式報告工作階段事件後，使用 Amazon Pinpoint 主控台的分析頁面，檢視工作階段、每日作用中端點、7 天保留率等圖表。

自訂事件

您指派自訂事件類型所定義的非標準事件。您可以將自訂屬性和指標新增到自訂事件。

在主控台的 Analytics (分析) 頁面，Events (事件) 標籤會顯示應用程式報告的所有自訂事件的指標。

獲利事件

報告您的應用程式產生的營收，以及使用者購買的項目數。

在 Analytics (分析) 頁面，Revenue (營收) 標籤會顯示 Revenue (營收)、Paying users (付費使用者)、Units sold (售出品項數量) 等圖表。

身分驗證事件

指出使用者向您的應用程式進行驗證的頻率。

在 Analytics (分析) 頁面，Users (使用者) 標籤會顯示 Sign-ins (登入次數)、Sign-ups (註冊次數) 和 Authentication failures (身分驗證失敗) 的圖表。

開始之前

如果您尚未執行以下操作，請現在執行：

- 將您的應用程式與 AWS Amplify 整合。請參閱 使用 AWS Amplify 將您的前端應用程式連線至 Amazon Pinpoint。
- 更新您的應用程式來註冊端點。請參閱 在應用程式中註冊 Amazon Pinpoint 端點。

使用適用於 Android 或 iOS AWS 的行動 SDKs報告事件

您可以使用適用於 iOS 和 Android AWS 的 Mobile SDKs，讓行動應用程式向 Amazon Pinpoint 報告事件。

如需更新應用程式以記錄事件並將其提交至 Amazon Pinpoint 的詳細資訊，請參閱 AWS Amplify 文件中的下列頁面：

- iOS 開發套件文件中的分析
- Android 開發套件文件中的分析

使用 AWS Amplify JavaScript 程式庫報告事件

您可以使用 AWS Amplify JavaScript 程式庫，讓 JavaScript 和 React Native 應用程式向 Amazon Pinpoint 報告應用程式用量事件。如需更新應用程式以將事件提交到 Amazon Pinpoint 的詳細資訊，請參閱 AWS Amplify JavaScript 文件中的分析。

使用 Amazon Pinpoint API 報告事件

您可以使用 Amazon Pinpoint API 或 AWS SDK 大量提交事件至 Amazon Pinpoint。詳情請參閱 Amazon Pinpoint API 參考中的事件。

後續步驟

在您更新應用程式以報告事件後，它會將用量資料傳送至 Amazon Pinpoint。您可以在 主控台中檢視此資料，並將其串流至 Amazon Kinesis。您也可以更新您的應用程式，以處理您使用 Amazon Pinpoint 傳送的推送通知。如需詳細資訊，請參閱AWS 《 最終使用者傳訊推送使用者指南》中的下列主題。

- 設定推送通知
- 設定 Swift 推播通知
- 設定 Android 推播通知
- 設定流暢推播通知
- 設定 React Native 推播通知
- 建立專案
- 處理推送通知

使用 Amazon Pinpoint 從應用程式傳送交易訊息

您可以使用 Amazon Pinpoint API 和 AWS SDKs，直接從應用程式傳送交易訊息。相對於您傳送至客群的訊息，交易訊息是您傳送至特定收件人的訊息。有幾個原因，您可能想要傳送交易訊息，而不是行銷活動型訊息。例如，您可以在客戶下訂單時，透過電子郵件傳送訂單確認。您也可以透過簡訊或語音傳送一次性密碼，客戶可以使用此簡訊或語音，完成為您的服務建立帳戶的程序。

本節包括採用數種程式設計語言的範例程式碼，您可以使用這些範例程式碼，開始傳送交易電子郵件、簡訊和語音訊息。

如需端點、區段和頻道的更多程式碼範例，請參閱程式碼範例。

本節主題：

- 使用 Amazon Pinpoint 傳送交易電子郵件
- 使用 Amazon Pinpoint 傳送交易簡訊
- 使用 Amazon Pinpoint 傳送語音訊息

使用 Amazon Pinpoint 傳送交易電子郵件

本節提供完整的程式碼範例，您可以使用這些程式碼範例，透過 Amazon Pinpoint 傳送交易電子郵件訊息：

- 使用 Amazon Pinpoint API 中的 SendMessages 操作：您可以使用 Amazon Pinpoint API 中的 `SendMessages` 操作，在 Amazon Pinpoint 支援的所有管道中傳送訊息，包括推播通知、SMS、語音和電子郵件管道。

  使用此操作的優點是，在所有管道之中，傳送訊息的請求語法非常類似。這可讓您更輕鬆地重新規劃現有的程式碼。`SendMessages` 操作也可讓您替換電子郵件訊息中的內容，並讓您將電子郵件傳送到 Amazon Pinpoint 端點 ID，而不是特定的電子郵件地址。

本節包括採用數種程式設計語言的範例程式碼，您可以使用這些範例程式碼，開始傳送交易電子郵件。

如需端點、區段和頻道的更多程式碼範例，請參閱程式碼範例。

選擇要傳送電子郵件的方法

用於傳送交易電子郵件的最佳方法取決於您的使用案例。例如，如果您需要使用第三方應用程式傳送電子郵件，或者您的程式設計語言沒有可用的 AWS 軟體開發套件，您可能必須使用 SMTP 界面。如果

要在 Amazon Pinpoint 支援的其他管道中傳送訊息，而且使用一致的程式碼提出這些請求，應該使用 Amazon Pinpoint API 中的 `SendMessages` 操作。

在 Amazon Pinpoint 和 Amazon SES 之間進行選擇

如果您會傳送大量交易電子郵件 (例如購買確認或密碼重設訊息)，請考慮使用 Amazon SES。Amazon SES 具有 API 和 SMTP 界面，這兩種界面都非常適合從您的應用程式或服務傳送電子郵件。它還提供額外的電子郵件功能，包括電子郵件接收功能、組態集和傳送授權功能。

Amazon SES 還包含一個 SMTP 界面，您可將其與現有的第三方應用程式整合，包括客戶關係管理 (CRM) 服務 (例如 Salesforce)。如需有關使用 Amazon SES 傳送電子郵件的詳細資訊，請參閱 Amazon Simple Email Service 開發人員指南以取得詳細資訊。

使用 Amazon Pinpoint API 傳送電子郵件

本節包含完整的程式碼範例，您可以使用 SDK 透過 Amazon Pinpoint API 傳送電子郵件 AWS 。您需要先驗證電子郵件地址或網域，才能傳送訊息。

C#

使用這個範例，可藉由使用 適用於 .NET 的 AWS SDK 來傳送電子郵件。此範例假設您已安裝並設定 適用於 .NET 的 SDK。詳情請參閱 適用於 .NET 的 AWS SDK 開發人員指南中的 適用於 .NET 的 AWS SDK入門。

此範例假設您是使用共用的憑證檔案，來指定現有使用者的存取金鑰和私密存取金鑰。如需詳細資訊，請參閱《 適用於 .NET 的 AWS SDK 開發人員指南》中的設定 AWS 登入資料。

此程式碼範例已使用 適用於 .NET 的 AWS SDK 版本 3.3.29.13 和 NET Core 執行時間版本 2.1.2 進行測試。

```
using Amazon;
using Amazon.Pinpoint;
using Amazon.Pinpoint.Model;
using Microsoft.Extensions.Configuration;

namespace SendEmailMessage;

public class SendEmailMainClass
{
    public static async Task Main(string[] args)
    {
```

```
        var configuration = new ConfigurationBuilder()
        .SetBasePath(Directory.GetCurrentDirectory())
        .AddJsonFile("settings.json") // Load test settings from .json file.
        .AddJsonFile("settings.local.json",
            true) // Optionally load local settings.
        .Build();

        // The AWS Region that you want to use to send the email. For a list of
        // AWS Regions where the Amazon Pinpoint API is available, see
        // https://docs.aws.amazon.com/pinpoint/latest/apireference/
        string region = "us-east-1";

        // The "From" address. This address has to be verified in Amazon Pinpoint
        // in the region you're using to send email.
        string senderAddress = configuration["SenderAddress"]!;

        // The address on the "To" line. If your Amazon Pinpoint account is in
        // the sandbox, this address also has to be verified.
        string toAddress = configuration["ToAddress"]!;

        // The Amazon Pinpoint project/application ID to use when you send this
message.
        // Make sure that the SMS channel is enabled for the project or application
        // that you choose.
        string appId = configuration["AppId"]!;

        try
        {
            await SendEmailMessage(region, appId, toAddress, senderAddress);
        }
        catch (Exception ex)
        {
            Console.WriteLine("The message wasn't sent. Error message: " +
ex.Message);
        }
    }

    public static async Task<MessageResponse> SendEmailMessage(
        string region, string appId, string toAddress, string senderAddress)
    {
        var client = new
AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));

        // The subject line of the email.
```

```
        string subject = "Amazon Pinpoint Email test";

        // The body of the email for recipients whose email clients don't
        // support HTML content.
        string textBody = @"Amazon Pinpoint Email Test (.NET)"
                          + "\n---------------------------------"
                          + "\nThis email was sent using the Amazon Pinpoint API
 using the AWS SDK for .NET.";

        // The body of the email for recipients whose email clients support
        // HTML content.
        string htmlBody = @"<html>"
                          + "\n<head></head>"
                          + "\n<body>"
                          + "\n  <h1>Amazon Pinpoint Email Test (AWS SDK for .NET)</
h1>"
                          + "\n  <p>This email was sent using the "
                          + "\n    <a href='https://aws.amazon.com/pinpoint/'>Amazon
 Pinpoint</a> API "
                          + "\n    using the <a href='https://aws.amazon.com/sdk-
for-net/'>AWS SDK for .NET</a>"
                          + "\n  </p>"
                          + "\n</body>"
                          + "\n</html>";

        // The character encoding the you want to use for the subject line and
        // message body of the email.
        string charset = "UTF-8";

        var sendRequest = new SendMessagesRequest
        {
            ApplicationId = appId,
            MessageRequest = new MessageRequest
            {
                Addresses = new Dictionary<string, AddressConfiguration>
                {
                    {
                        toAddress,
                        new AddressConfiguration
                        {
                            ChannelType = ChannelType.EMAIL
                        }
                    }
                },
```

```
                MessageConfiguration = new DirectMessageConfiguration
                {
                    EmailMessage = new EmailMessage
                    {
                        FromAddress = senderAddress,
                        SimpleEmail = new SimpleEmail
                        {
                            HtmlPart = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = htmlBody
                            },
                            TextPart = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = textBody
                            },
                            Subject = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = subject
                            }
                        }
                    }
                }
            };
            Console.WriteLine("Sending message...");
            SendMessagesResponse response = await client.SendMessagesAsync(sendRequest);
            Console.WriteLine("Message sent!");
            return response.MessageResponse;
        }
    }
```

Java

使用這個範例，可藉由使用 適用於 Java 的 AWS SDK 來傳送電子郵件。此範例假設您已安裝並設定 AWS SDK for Java 2.x。詳情請參閱 AWS SDK for Java 2.x 開發人員指南中的開始使用。

此範例假設您是使用共用的憑證檔案，來指定現有使用者的存取金鑰和私密存取金鑰。詳情請參閱適用於 Java 的 AWS SDK 開發人員指南中的設定預設憑證和區域。

此程式碼範例使用 2.3.1 適用於 Java 的 AWS SDK 版和 OpenJDK 11.0.1 版進行測試。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmailPart;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmail;
import software.amazon.awssdk.services.pinpoint.model.EmailMessage;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpointemail.PinpointEmailClient;
import software.amazon.awssdk.services.pinpointemail.model.Body;
import software.amazon.awssdk.services.pinpointemail.model.Content;
import software.amazon.awssdk.services.pinpointemail.model.Destination;
import software.amazon.awssdk.services.pinpointemail.model.EmailContent;
import software.amazon.awssdk.services.pinpointemail.model.Message;
import software.amazon.awssdk.services.pinpointemail.model.SendEmailRequest;

import java.util.HashMap;
import java.util.Map;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmailPart;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmail;
import software.amazon.awssdk.services.pinpoint.model.EmailMessage;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpointemail.PinpointEmailClient;
import software.amazon.awssdk.services.pinpointemail.model.Body;
import software.amazon.awssdk.services.pinpointemail.model.Content;
import software.amazon.awssdk.services.pinpointemail.model.Destination;
import software.amazon.awssdk.services.pinpointemail.model.EmailContent;
import software.amazon.awssdk.services.pinpointemail.model.Message;
import software.amazon.awssdk.services.pinpointemail.model.SendEmailRequest;

import java.util.HashMap;
import java.util.Map;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SendEmailMessage {

        // The character encoding the you want to use for the subject line and
        // message body of the email.
        public static String charset = "UTF-8";

    // The body of the email for recipients whose email clients support HTML
 content.
    static final String body = """
        Amazon Pinpoint test (AWS SDK for Java 2.x)

        This email was sent through the Amazon Pinpoint Email API using the AWS SDK
 for Java 2.x

        """;

        public static void main(String[] args) {
                final String usage = """

                                Usage:    <subject> <appId> <senderAddress>
 <toAddress>

            Where:
               subject - The email subject to use.
               senderAddress - The from address. This address has to be verified in
 Amazon Pinpoint in the region you're using to send email\s
               toAddress - The to address. This address has to be verified in Amazon
 Pinpoint in the region you're using to send email\s
            """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }
```

```
        String subject = args[0];
        String senderAddress = args[1];
        String toAddress = args[2];
        System.out.println("Sending a message");
        PinpointEmailClient pinpoint = PinpointEmailClient.builder()
            .region(Region.US_EAST_1)
            .build();

        sendEmail(pinpoint, subject, senderAddress, toAddress);
        System.out.println("Email was sent");
        pinpoint.close();
    }

    public static void sendEmail(PinpointEmailClient pinpointEmailClient, String
subject, String senderAddress, String toAddress) {
        try {
            Content content = Content.builder()
                .data(body)
                .build();

            Body messageBody = Body.builder()
                .text(content)
                .build();

            Message message = Message.builder()
                .body(messageBody)
                .subject(Content.builder().data(subject).build())
                .build();

            Destination destination = Destination.builder()
                .toAddresses(toAddress)
                .build();

            EmailContent emailContent = EmailContent.builder()
                .simple(message)
                .build();

            SendEmailRequest sendEmailRequest = SendEmailRequest.builder()
                .fromEmailAddress(senderAddress)
                .destination(destination)
                .content(emailContent)
                .build();

            pinpointEmailClient.sendEmail(sendEmailRequest);
```

```
            System.out.println("Message Sent");

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

如需完整的 SDK 範例，請參閱 GitHub 上的 SendEmailMessage.java。

JavaScript (Node.js)

使用此範例，藉由 AWS SDK for JavaScript in Node.js 傳送電子郵件。此範例假設您已安裝並設定了 SDK for JavaScript in Node.js。詳情請參閱 AWS SDK for JavaScript in Node.js 開發人員指南中的開始使用。

此範例假設您是使用共用的憑證檔案，來指定現有使用者的存取金鑰和私密存取金鑰。詳情請參閱 AWS SDK for JavaScript in Node.js 開發人員指南中的設定憑證。

此程式碼範例使用 Node.js 中 JavaScript 軟體開發套件 2.388.0 版和 11.7.0 版進行測試。

```
"use strict";

const AWS = require("aws-sdk");

// The AWS Region that you want to use to send the email. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/
const aws_region = "us-west-2";

// The "From" address. This address has to be verified in Amazon Pinpoint
// in the region that you use to send email.
const senderAddress = "sender@example.com";

// The address on the "To" line. If your Amazon Pinpoint account is in
// the sandbox, this address also has to be verified.
var toAddress = "recipient@example.com";

// The Amazon Pinpoint project/application ID to use when you send this message.
// Make sure that the SMS channel is enabled for the project or application
// that you choose.
const appId = "ce796be37f32f178af652b26eexample";
```

```
// The subject line of the email.
var subject = "Amazon Pinpoint (AWS SDK for JavaScript in Node.js)";

// The email body for recipients with non-HTML email clients.
var body_text = `Amazon Pinpoint Test (SDK for JavaScript in Node.js)
----------------------------------------------------
This email was sent with Amazon Pinpoint using the AWS SDK for JavaScript in
 Node.js.
For more information, see https:\/\/aws.amazon.com/sdk-for-node-js/`;

// The body of the email for recipients whose email clients support HTML content.
var body_html = `<html>
<head></head>
<body>
  <h1>Amazon Pinpoint Test (SDK for JavaScript in Node.js)</h1>
  <p>This email was sent with
    <a href='https://aws.amazon.com/pinpoint/'>the Amazon Pinpoint API</a> using the
    <a href='https://aws.amazon.com/sdk-for-node-js/'>
      AWS SDK for JavaScript in Node.js</a>.</p>
</body>
</html>`;

// The character encoding the you want to use for the subject line and
// message body of the email.
var charset = "UTF-8";

// Specify that you're using a shared credentials file.
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpoint = new AWS.Pinpoint();

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: appId,
  MessageRequest: {
    Addresses: {
      [toAddress]: {
        ChannelType: "EMAIL",
```

```
      },
    },
    MessageConfiguration: {
      EmailMessage: {
        FromAddress: senderAddress,
        SimpleEmail: {
          Subject: {
            Charset: charset,
            Data: subject,
          },
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
          TextPart: {
            Charset: charset,
            Data: body_text,
          },
        },
      },
    },
  },
};

//Try to send the email.
pinpoint.sendMessages(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
  } else {
    console.log(
      "Email sent! Message ID: ",
      data["MessageResponse"]["Result"][toAddress]["MessageId"]
    );
  }
});
```

Python

使用這個範例，可藉由使用 適用於 Python (Boto3) 的 AWS SDK 來傳送電子郵件。此範例假設您已安裝並設定了 SDK for Python (Boto3)。詳情請參閱 AWS SDK for Python (Boto3) API 參考中的 Quickstart。

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


def send_email_message(
    pinpoint_client,
    app_id,
    sender,
    to_addresses,
    char_set,
    subject,
    html_message,
    text_message,
):
    """
    Sends an email message with HTML and plain text versions.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param app_id: The Amazon Pinpoint project ID to use when you send this message.
    :param sender: The "From" address. This address must be verified in
                   Amazon Pinpoint in the AWS Region you're using to send email.
    :param to_addresses: The addresses on the "To" line. If your Amazon Pinpoint
 account
                         is in the sandbox, these addresses must be verified.
    :param char_set: The character encoding to use for the subject line and message
                     body of the email.
    :param subject: The subject line of the email.
    :param html_message: The body of the email for recipients whose email clients
 can
                         display HTML content.
    :param text_message: The body of the email for recipients whose email clients
                         don't support HTML content.
    :return: A dict of to_addresses and their message IDs.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=app_id,
            MessageRequest={
                "Addresses": {
```

```
                    to_address: {"ChannelType": "EMAIL"} for to_address in
 to_addresses
                },
                "MessageConfiguration": {
                    "EmailMessage": {
                        "FromAddress": sender,
                        "SimpleEmail": {
                            "Subject": {"Charset": char_set, "Data": subject},
                            "HtmlPart": {"Charset": char_set, "Data": html_message},
                            "TextPart": {"Charset": char_set, "Data": text_message},
                        },
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send email.")
        raise
    else:
        return {
            to_address: message["MessageId"]
            for to_address, message in response["MessageResponse"]["Result"].items()
        }


def main():
    app_id = "ce796be37f32f178af652b26eexample"
    sender = "sender@example.com"
    to_address = "recipient@example.com"
    char_set = "UTF-8"
    subject = "Amazon Pinpoint Test (SDK for Python (Boto3))"
    text_message = """Amazon Pinpoint Test (SDK for Python)
    -------------------------------------
    This email was sent with Amazon Pinpoint using the AWS SDK for Python (Boto3).
    For more information, see https://aws.amazon.com/sdk-for-python/
                """
    html_message = """<html>
    <head></head>
    <body>
      <h1>Amazon Pinpoint Test (SDK for Python (Boto3)</h1>
      <p>This email was sent with
        <a href='https://aws.amazon.com/pinpoint/'>Amazon Pinpoint</a> using the
        <a href='https://aws.amazon.com/sdk-for-python/'>
          AWS SDK for Python (Boto3)</a>.</p>
```

```
    </body>
    </html>
                """

    print("Sending email.")
    message_ids = send_email_message(
        boto3.client("pinpoint"),
        app_id,
        sender,
        [to_address],
        char_set,
        subject,
        html_message,
        text_message,
    )
    print(f"Message sent! Message IDs: {message_ids}")


if __name__ == "__main__":
    main()
```

您也可以使用訊息範本傳送電子郵件，如以下範例所示：

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


def send_templated_email_message(
    pinpoint_client, project_id, sender, to_addresses, template_name,
 template_version
):
    """
    Sends an email message with HTML and plain text versions.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: The Amazon Pinpoint project ID to use when you send this
 message.
    :param sender: The "From" address. This address must be verified in
                   Amazon Pinpoint in the AWS Region you're using to send email.
    :param to_addresses: The addresses on the "To" line. If your Amazon Pinpoint
```

```
                        account is in the sandbox, these addresses must be
 verified.
    :param template_name: The name of the email template to use when sending the
 message.
    :param template_version: The version number of the message template.

    :return: A dict of to_addresses and their message IDs.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=project_id,
            MessageRequest={
                "Addresses": {
                    to_address: {"ChannelType": "EMAIL"} for to_address in
 to_addresses
                },
                "MessageConfiguration": {"EmailMessage": {"FromAddress": sender}},
                "TemplateConfiguration": {
                    "EmailTemplate": {
                        "Name": template_name,
                        "Version": template_version,
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send email.")
        raise
    else:
        return {
            to_address: message["MessageId"]
            for to_address, message in response["MessageResponse"]["Result"].items()
        }


def main():
    project_id = "296b04b342374fceb661bf494example"
    sender = "sender@example.com"
    to_addresses = ["recipient@example.com"]
    template_name = "My_Email_Template"
    template_version = "1"

    print("Sending email.")
    message_ids = send_templated_email_message(
```

```
        boto3.client("pinpoint"),
        project_id,
        sender,
        to_addresses,
        template_name,
        template_version,
    )
    print(f"Message sent! Message IDs: {message_ids}")


if __name__ == "__main__":
    main()
```

這些範例假設您是使用共用的憑證檔案，來指定現有使用者的存取金鑰和私密存取金鑰。詳情請參閱 AWS SDK for Python (Boto3) API 參考中的憑證。

使用 Amazon Pinpoint 將取消訂閱標頭新增至電子郵件

> **Note**
>
> 在使用電子郵件標頭之前，如果您要從行銷活動或旅程傳送電子郵件，則必須設定電子郵件協同運作傳送角色。對於直接傳送電子郵件，您必須擁有 `ses:SendEmail`和 的許可`ses:SendRawEmail`。如需詳細資訊，請參閱《Amazon Pinpoint 使用者指南》中的建立電子郵件協同運作傳送角色。

在電子郵件中加入取消訂閱連結是最佳做法，在某些國家/地區也是法律規定。若要新增一鍵式取消訂閱連結，請新增下列標頭：

1. 將標頭名稱設定為 `List-Unsubscribe`，並將值設定為取消訂閱連結。連結必須支援 HTTP POST 請求，才能處理收件人取消訂閱請求。
2. 將標頭名稱設定為 `List-Unsubscribe-Post`，並將值設定為 `List-Unsubscribe=One-Click`。

您最多可以將 15 個標頭新增至電子郵件訊息。如需支援的標頭清單，請參閱《Amazon SES Simple Email Service 開發人員指南》中的 Amazon SES 標頭欄位。 https://docs.aws.amazon.com/ses/latest/DeveloperGuide/

下列範例示範如何使用 傳送具有取消訂閱標頭的電子郵件訊息 AWS Command Line Interface。如需設定 的詳細資訊 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》中的設定 AWS CLI。

在下列命令中，執行下列動作：

- 將 *AppId* 取代為您的應用程式 ID。
- 將 *richard_roe@example.com* 取代為收件人的電子郵件地址。
- 將 *https://example.com/unsub* 取代為您的取消訂閱連結。
- 將 *example123456* 取代為收件人的唯一識別符。

```
aws pinpoint send-messages --application-id AppId --message-request '{
 "Addresses": {
   "richard_roe@example.com": {
     "ChannelType": "EMAIL"
   }
 },
 "MessageConfiguration": {
   "EmailMessage": {
     "Substitutions": {
       "url": [
         "https://example.com/unsub"
       ],
        "id1": [
          "/example123456"
        ]
     },
     "SimpleEmail": {
       "TextPart": {
         "Data": "Sample email message with an subscribe header",
         "Charset": "UTF-8"
       },
       "Subject": {
         "Data": "Hello",
         "Charset": "UTF-8"
       },
       "Headers": [
         {
           "Name": "List-Unsubscribe",
           "Value": "{{url}}{{id1}}"
         },
```

```
        {
          "Name": "List-Unsubscribe-Post",
          "Value": "List-Unsubscribe=One-Click"
        }
      ]
    }
  }
 }
}'
```

使用 Amazon Pinpoint 傳送交易簡訊

您可以使用 Amazon Pinpoint API 將 SMS 訊息 (文字訊息) 傳送到特定電話號碼或端點 ID。本節包含完整的程式碼範例，您可以使用 SDK 透過 Amazon Pinpoint API 傳送簡訊 AWS 。您的帳戶必須處於生產狀態，而且您具有可傳送簡訊的作用中起始身分。

如需端點、區段和頻道的更多程式碼範例，請參閱程式碼範例。

C#

使用這個範例，可藉由使用 適用於 .NET 的 AWS SDK 來傳送簡訊。此範例假設您已安裝並設定適用於 .NET 的 SDK。詳情請參閱 適用於 .NET 的 AWS SDK 開發人員指南中的開始使用。

此範例假設您是使用共用的登入資料檔案，來指定現有 IAM 使用者的存取金鑰和私密存取金鑰。詳情請參閱 適用於 .NET 的 AWS SDK 開發人員指南中的設定 AWS 憑證。

```
using Amazon;
using Amazon.Pinpoint;
using Amazon.Pinpoint.Model;
using Microsoft.Extensions.Configuration;

namespace SendSmsMessage;

public class SendSmsMessageMainClass
{
    public static async Task Main(string[] args)
    {
        var configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load test settings from .json file.
            .AddJsonFile("settings.local.json",
```

```
                true) // Optionally load local settings.
            .Build();

        // The AWS Region that you want to use to send the message. For a list of
        // AWS Regions where the Amazon Pinpoint API is available, see
        // https://docs.aws.amazon.com/pinpoint/latest/apireference/
        string region = "us-east-1";

        // The phone number or short code to send the message from. The phone number
        // or short code that you specify has to be associated with your Amazon
 Pinpoint
        // account. For best results, specify long codes in E.164 format.
        string originationNumber = configuration["OriginationNumber"]!;

        // The recipient's phone number.  For best results, you should specify the
        // phone number in E.164 format.
        string destinationNumber = configuration["DestinationNumber"]!;

        // The Pinpoint project/ application ID to use when you send this message.
        // Make sure that the SMS channel is enabled for the project or application
        // that you choose.
        string appId = configuration["AppId"]!;

        // The type of SMS message that you want to send. If you plan to send
        // time-sensitive content, specify TRANSACTIONAL. If you plan to send
        // marketing-related content, specify PROMOTIONAL.
        MessageType messageType = MessageType.TRANSACTIONAL;

        // The registered keyword associated with the originating short code.
        string? registeredKeyword = configuration["RegisteredKeyword"];

        // The sender ID to use when sending the message. Support for sender ID
        // varies by country or region. For more information, see
        // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
        string? senderId = configuration["SenderId"];

        try
        {
            var response = await SendSmsMessage(region, appId, destinationNumber,
                originationNumber, registeredKeyword, senderId, messageType);
            Console.WriteLine($"Message sent to
 {response.MessageResponse.Result.Count} recipient(s).");
            foreach (var messageResultValue in
```

```
                    response.MessageResponse.Result.Select(r => r.Value))
            {
                Console.WriteLine($"{messageResultValue.MessageId} Status:
{messageResultValue.DeliveryStatus}");
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine("The message wasn't sent. Error message: " +
ex.Message);
        }
    }

    public static async Task<SendMessagesResponse> SendSmsMessage(
        string region, string appId, string destinationNumber, string
originationNumber,
        string? keyword, string? senderId, MessageType messageType)
    {

        // The content of the SMS message.
        string message = "This message was sent through Amazon Pinpoint using" +
                         " the AWS SDK for .NET. Reply STOP to opt out.";


        var client = new
AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));

        SendMessagesRequest sendRequest = new SendMessagesRequest
        {
            ApplicationId = appId,
            MessageRequest = new MessageRequest
            {
                Addresses =
                    new Dictionary<string, AddressConfiguration>
                    {
                        {
                            destinationNumber,
                            new AddressConfiguration { ChannelType =
ChannelType.SMS }
                        }
                    },
                MessageConfiguration = new DirectMessageConfiguration
                {
                    SMSMessage = new SMSMessage
```

```
                    {
                        Body = message,
                        MessageType = MessageType.TRANSACTIONAL,
                        OriginationNumber = originationNumber,
                        SenderId = senderId,
                        Keyword = keyword
                    }
                }
            }
        };
        SendMessagesResponse response = await client.SendMessagesAsync(sendRequest);
        return response;
    }
}
```

Java

使用這個範例，可藉由使用 適用於 Java 的 AWS SDK 來傳送簡訊。此範例假設您已安裝並設定了 SDK for Java。詳情請參閱 適用於 Java 的 AWS SDK 開發人員指南中的開始使用。

此範例假設您是使用共用的登入資料檔案，來指定現有 IAM 使用者的存取金鑰和私密存取金鑰。詳情請參閱 適用於 Java 的 AWS SDK 開發人員指南中的設定預設憑證和區域。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;
```

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
```

```
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SendMessage {

        // The type of SMS message that you want to send. If you plan to send
        // time-sensitive content, specify TRANSACTIONAL. If you plan to send
        // marketing-related content, specify PROMOTIONAL.
        public static String messageType = "TRANSACTIONAL";

        // The registered keyword associated with the originating short code.
        public static String registeredKeyword = "myKeyword";

        // The sender ID to use when sending the message. Support for sender ID
        // varies by country or region. For more information, see
        // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
        public static String senderId = "MySenderID";

        public static void main(String[] args) {
                final String usage = """

                                Usage:   <message> <appId> <originationNumber>
 <destinationNumber>\s

                                Where:
                                  message - The body of the message to send.
                                  appId - The Amazon Pinpoint project/application ID
 to use when you send this message.
```

```
                                originationNumber - The phone number or short code
that you specify has to be associated with your Amazon Pinpoint account. For best
results, specify long codes in E.164 format (for example, +1-555-555-5654).
                                destinationNumber - The recipient's phone number.
For best results, you should specify the phone number in E.164 format (for example,
+1-555-555-5654).\s
                                """;

                if (args.length != 4) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String message = args[0];
                String appId = args[1];
                String originationNumber = args[2];
                String destinationNumber = args[3];
                System.out.println("Sending a message");
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();

                sendSMSMessage(pinpoint, message, appId, originationNumber,
destinationNumber);
                pinpoint.close();
        }

        public static void sendSMSMessage(PinpointClient pinpoint, String message,
String appId,
                        String originationNumber,
                        String destinationNumber) {
                try {
                        Map<String, AddressConfiguration> addressMap = new
HashMap<String, AddressConfiguration>();
                        AddressConfiguration addConfig =
AddressConfiguration.builder()
                                        .channelType(ChannelType.SMS)
                                        .build();

                        addressMap.put(destinationNumber, addConfig);
                        SMSMessage smsMessage = SMSMessage.builder()
                                        .body(message)
                                        .messageType(messageType)
                                        .originationNumber(originationNumber)
```

```
                                        .senderId(senderId)
                                        .keyword(registeredKeyword)
                                        .build();

                        // Create a DirectMessageConfiguration object.
                        DirectMessageConfiguration direct =
 DirectMessageConfiguration.builder()
                                        .smsMessage(smsMessage)
                                        .build();

                        MessageRequest msgReq = MessageRequest.builder()
                                        .addresses(addressMap)
                                        .messageConfiguration(direct)
                                        .build();

                        // create a SendMessagesRequest object
                        SendMessagesRequest request = SendMessagesRequest.builder()
                                        .applicationId(appId)
                                        .messageRequest(msgReq)
                                        .build();

                        SendMessagesResponse response =
 pinpoint.sendMessages(request);
                        MessageResponse msg1 = response.messageResponse();
                        Map map1 = msg1.result();

                        // Write out the result of sendMessage.
                        map1.forEach((k, v) -> System.out.println((k + ":" + v)));

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
        }
}
```

如需完整的 SDK 範例，請參閱 GitHub 上的 SendMessage.java。

JavaScript (Node.js)

使用此範例，藉由 AWS SDK for JavaScript in Node.js 傳送 SMS 訊息。此範例假設您已安裝並設定了 SDK for JavaScript in Node.js。詳情請參閱 AWS SDK for JavaScript in Node.js 開發人員指南中的開始使用。

此範例假設您是使用共用的登入資料檔案，來指定現有 IAM 使用者的存取金鑰和私密存取金鑰。詳情請參閱 AWS SDK for JavaScript in Node.js 開發人員指南中的設定憑證。

```
"use strict";

var AWS = require("aws-sdk");

// The AWS Region that you want to use to send the message. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/.
var aws_region = "us-east-1";

// The phone number or short code to send the message from. The phone number
// or short code that you specify has to be associated with your Amazon Pinpoint
// account. For best results, specify long codes in E.164 format.
var originationNumber = "+12065550199";

// The recipient's phone number.  For best results, you should specify the
// phone number in E.164 format.
var destinationNumber = "+14255550142";

// The content of the SMS message.
var message =
  "This message was sent through Amazon Pinpoint " +
  "using the AWS SDK for JavaScript in Node.js. Reply STOP to " +
  "opt out.";

// The Amazon Pinpoint project/application ID to use when you send this message.
// Make sure that the SMS channel is enabled for the project or application
// that you choose.
var applicationId = "ce796be37f32f178af652b26eexample";

// The type of SMS message that you want to send. If you plan to send
// time-sensitive content, specify TRANSACTIONAL. If you plan to send
// marketing-related content, specify PROMOTIONAL.
var messageType = "TRANSACTIONAL";

// The registered keyword associated with the originating short code.
var registeredKeyword = "myKeyword";

// The sender ID to use when sending the message. Support for sender ID
// varies by country or region. For more information, see
```

```
// https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-countries.html
var senderId = "MySenderID";

// Specify that you're using a shared credentials file, and optionally specify
// the profile that you want to use.
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpoint = new AWS.Pinpoint();

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: applicationId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
        SenderId: senderId,
      },
    },
  },
};

//Try to send the message.
pinpoint.sendMessages(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
    // Otherwise, show the unique ID for the message.
  } else {
    console.log(
      "Message sent! " +
```

```
        data["MessageResponse"]["Result"][destinationNumber]["StatusMessage"]
    );
  }
});
```

Python

使用這個範例，可藉由使用 適用於 Python (Boto3) 的 AWS SDK 來傳送簡訊。此範例假設您已安裝並設定了 SDK for Python。如需詳細資訊，請參閱《 AWS 適用於 Python 的 SDK (Boto3) 入門》中的 Quickstart。

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


def send_sms_message(
    pinpoint_client,
    app_id,
    origination_number,
    destination_number,
    message,
    message_type,
):
    """
    Sends an SMS message with Amazon Pinpoint.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param app_id: The Amazon Pinpoint project/application ID to use when you send
                   this message. The SMS channel must be enabled for the project or
                   application.
    :param destination_number: The recipient's phone number in E.164 format.
    :param origination_number: The phone number to send the message from. This phone
                               number must be associated with your Amazon Pinpoint
                               account and be in E.164 format.
    :param message: The content of the SMS message.
    :param message_type: The type of SMS message that you want to send. If you send
                         time-sensitive content, specify TRANSACTIONAL. If you send
                         marketing-related content, specify PROMOTIONAL.
```

```
    :return: The ID of the message.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=app_id,
            MessageRequest={
                "Addresses": {destination_number: {"ChannelType": "SMS"}},
                "MessageConfiguration": {
                    "SMSMessage": {
                        "Body": message,
                        "MessageType": message_type,
                        "OriginationNumber": origination_number,
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send message.")
        raise
    else:
        return response["MessageResponse"]["Result"][destination_number]
["MessageId"]


def main():
    app_id = "ce796be37f32f178af652b26eexample"
    origination_number = "+12065550199"
    destination_number = "+14255550142"
    message = (
        "This is a sample message sent from Amazon Pinpoint by using the AWS SDK for
 "
        "Python (Boto 3)."
    )
    message_type = "TRANSACTIONAL"

    print("Sending SMS message.")
    message_id = send_sms_message(
        boto3.client("pinpoint"),
        app_id,
        origination_number,
        destination_number,
        message,
        message_type,
    )
```

```
    print(f"Message sent! Message ID: {message_id}.")


if __name__ == "__main__":
    main()
```

您也可以使用訊息範本傳送 SMS 訊息，如以下範例所示：

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


def send_templated_sms_message(
    pinpoint_client,
    project_id,
    destination_number,
    message_type,
    origination_number,
    template_name,
    template_version,
):
    """
    Sends an SMS message to a specific phone number using a pre-defined template.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: An Amazon Pinpoint project (application) ID.
    :param destination_number: The phone number to send the message to.
    :param message_type: The type of SMS message (promotional or transactional).
    :param origination_number: The phone number that the message is sent from.
    :param template_name: The name of the SMS template to use when sending the
 message.
    :param template_version: The version number of the message template.

    :return The ID of the message.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=project_id,
            MessageRequest={
                "Addresses": {destination_number: {"ChannelType": "SMS"}},
```

```
                "MessageConfiguration": {
                    "SMSMessage": {
                        "MessageType": message_type,
                        "OriginationNumber": origination_number,
                    }
                },
                "TemplateConfiguration": {
                    "SMSTemplate": {"Name": template_name, "Version":
 template_version}
                },
            },
        )

    except ClientError:
        logger.exception("Couldn't send message.")
        raise
    else:
        return response["MessageResponse"]["Result"][destination_number]
["MessageId"]


def main():
    region = "us-east-1"
    origination_number = "+18555550001"
    destination_number = "+14255550142"
    project_id = "7353f53e6885409fa32d07cedexample"
    message_type = "TRANSACTIONAL"
    template_name = "My_SMS_Template"
    template_version = "1"
    message_id = send_templated_sms_message(
        boto3.client("pinpoint", region_name=region),
        project_id,
        destination_number,
        message_type,
        origination_number,
        template_name,
        template_version,
    )
    print(f"Message sent! Message ID: {message_id}.")


if __name__ == "__main__":
    main()
```

這些範例假設您是使用共用的憑證檔案，來指定現有 IAM 使用者的存取金鑰和私密存取金鑰。詳情請參閱 AWS SDK for Python (Boto3) API 參考中的憑證。

使用 Amazon Pinpoint 傳送語音訊息

您可以使用 Amazon Pinpoint API，將語音訊息傳送到特定電話號碼。本節包含完整的程式碼範例，您可以使用 SDK 透過 Amazon Pinpoint SMS 和語音 API 傳送語音訊息 AWS 。您的帳戶必須處於生產狀態，而且您具有可傳送語音訊息的作用中起始身分。

如需端點、區段和頻道的更多程式碼範例，請參閱程式碼範例。

Java

使用這個範例，可藉由使用 適用於 Java 的 AWS SDK 來傳送語音訊息。此範例假設您已安裝並設定了 SDK for Java。詳情請參閱 適用於 Java 的 AWS SDK 開發人員指南中的開始使用。

此範例假設您是使用共用的憑證檔案，來指定現有使用者的存取金鑰和私密存取金鑰。如需詳細資訊，請參閱《 適用於 Java 的 AWS SDK 開發人員指南》中的設定 AWS 憑證和開發區域。

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
```

```
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 * <p>
 * For more information, see the following documentation topic:
 * <p>
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class SendVoiceMessage {

    // The Amazon Polly voice that you want to use to send the message. For a list
    // of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
    static final String voiceName = "Matthew";

    // The language to use when sending the message. For a list of supported
    // languages, see
    // https://docs.aws.amazon.com/polly/latest/dg/SupportedLanguage.html
    static final String languageCode = "en-US";

    // The content of the message. This example uses SSML to customize and control
    // certain aspects of the message, such as by adding pauses and changing
    // phonation. The message can't contain any line breaks.
    static final String ssmlMessage = "<speak>This is a test message sent from "
            + "<emphasis>Amazon Pinpoint</emphasis> "
            + "using the <break strength='weak'/>AWS "
            + "SDK for Java. "
            + "<amazon:effect phonation='soft'>Thank "
            + "you for listening.</amazon:effect></speak>";

    public static void main(String[] args) {

        final String usage = """
                Usage:   <originationNumber> <destinationNumber>\s
```

```
                Where:
                  originationNumber - The phone number or short code that you
specify has to be associated with your Amazon Pinpoint account. For best results,
specify long codes in E.164 format (for example, +1-555-555-5654).
                  destinationNumber - The recipient's phone number.  For best
results, you should specify the phone number in E.164 format (for example,
+1-555-555-5654).\s
                """;

       if (args.length != 2) {
           System.out.println(usage);
           System.exit(1);
       }
       String originationNumber = args[0];
       String destinationNumber = args[1];
       System.out.println("Sending a voice message");

       // Set the content type to application/json.
       List<String> listVal = new ArrayList<>();
       listVal.add("application/json");
       Map<String, List<String>> values = new HashMap<>();
       values.put("Content-Type", listVal);

       ClientOverrideConfiguration config2 = ClientOverrideConfiguration.builder()
               .headers(values)
               .build();

       PinpointSmsVoiceClient client = PinpointSmsVoiceClient.builder()
               .overrideConfiguration(config2)
               .region(Region.US_EAST_1)
               .build();

       sendVoiceMsg(client, originationNumber, destinationNumber);
       client.close();
   }

   public static void sendVoiceMsg(PinpointSmsVoiceClient client, String
originationNumber,
                                   String destinationNumber) {
       try {
           SSMLMessageType ssmlMessageType = SSMLMessageType.builder()
                   .languageCode(languageCode)
                   .text(ssmlMessage)
                   .voiceId(voiceName)
```

```
                    .build();

            VoiceMessageContent content = VoiceMessageContent.builder()
                    .ssmlMessage(ssmlMessageType)
                    .build();

            SendVoiceMessageRequest voiceMessageRequest =
 SendVoiceMessageRequest.builder()
                    .destinationPhoneNumber(destinationNumber)
                    .originationPhoneNumber(originationNumber)
                    .content(content)
                    .build();

            client.sendVoiceMessage(voiceMessageRequest);
            System.out.println("The message was sent successfully.");

        } catch (PinpointSmsVoiceException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

如需完整的 SDK 範例，請參閱 GitHub 上的 SendVoiceMessage.java。

JavaScript (Node.js)

使用此範例，透過使用 Node.js 中適用於 JavaScript 的 AWS SDK 來傳送語音訊息。此範例假設您已安裝並設定了 SDK for JavaScript in Node.js。

此範例假設您是使用共用的憑證檔案，來指定現有使用者的存取金鑰和私密存取金鑰。詳情請參閱 AWS SDK for JavaScript in Node.js 開發人員指南中的設定憑證。

```
"use strict";

var AWS = require("aws-sdk");

// The AWS Region that you want to use to send the voice message. For a list of
// AWS Regions where the Amazon Pinpoint SMS and Voice API is available, see
// https://docs.aws.amazon.com/pinpoint-sms-voice/latest/APIReference/
var aws_region = "us-east-1";

// The phone number that the message is sent from. The phone number that you
```

```
// specify has to be associated with your Amazon Pinpoint account. For best results,
 you
// should specify the phone number in E.164 format.
var originationNumber = "+12065550110";

// The recipient's phone number. For best results, you should specify the phone
// number in E.164 format.
var destinationNumber = "+12065550142";

// The language to use when sending the message. For a list of supported
// languages, see https://docs.aws.amazon.com/polly/latest/dg/SupportedLanguage.html
var languageCode = "en-US";

// The Amazon Polly voice that you want to use to send the message. For a list
// of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
var voiceId = "Matthew";

// The content of the message. This example uses SSML to customize and control
// certain aspects of the message, such as the volume or the speech rate.
// The message can't contain any line breaks.
var ssmlMessage =
  "<speak>" +
  "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> " +
  "using the <break strength='weak'/>AWS SDK for JavaScript in Node.js. " +
  "<amazon:effect phonation='soft'>Thank you for listening." +
  "</amazon:effect>" +
  "</speak>";

// The phone number that you want to appear on the recipient's device. The phone
// number that you specify has to be associated with your Amazon Pinpoint account.
var callerId = "+12065550199";

// The configuration set that you want to use to send the message.
var configurationSet = "ConfigSet";

// Specify that you're using a shared credentials file, and optionally specify
// the profile that you want to use.
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
```

```
var pinpointsmsvoice = new AWS.PinpointSMSVoice();

var params = {
  CallerId: callerId,
  ConfigurationSetName: configurationSet,
  Content: {
    SSMLMessage: {
      LanguageCode: languageCode,
      Text: ssmlMessage,
      VoiceId: voiceId,
    },
  },
  DestinationPhoneNumber: destinationNumber,
  OriginationPhoneNumber: originationNumber,
};

//Try to send the message.
pinpointsmsvoice.sendVoiceMessage(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
    // Otherwise, show the unique ID for the message.
  } else {
    console.log("Message sent! Message ID: " + data["MessageId"]);
  }
});
```

Python

使用這個範例，可藉由使用 適用於 Python (Boto3) 的 AWS SDK來傳送語音訊息。此範例假設您已安裝並設定了 SDK for Python (Boto3)。

此範例假設您是使用共用的憑證檔案，來指定現有使用者的存取金鑰和私密存取金鑰。詳情請參閱 AWS SDK for Python (Boto3) API 參考中的憑證。

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)
```

```
def send_voice_message(
    sms_voice_client,
    origination_number,
    caller_id,
    destination_number,
    language_code,
    voice_id,
    ssml_message,
):
    """
    Sends a voice message using speech synthesis provided by Amazon Polly.

    :param sms_voice_client: A Boto3 PinpointSMSVoice client.
    :param origination_number: The phone number that the message is sent from.
                               The phone number must be associated with your Amazon
                               Pinpoint account and be in E.164 format.
    :param caller_id: The phone number that you want to appear on the recipient's
                      device. The phone number must be associated with your Amazon
                      Pinpoint account and be in E.164 format.
    :param destination_number: The recipient's phone number. Specify the phone
                               number in E.164 format.
    :param language_code: The language to use when sending the message.
    :param voice_id: The Amazon Polly voice that you want to use to send the
 message.
    :param ssml_message: The content of the message. This example uses SSML to
 control
                         certain aspects of the message, such as the volume and the
                         speech rate. The message must not contain line breaks.
    :return: The ID of the message.
    """
    try:
        response = sms_voice_client.send_voice_message(
            DestinationPhoneNumber=destination_number,
            OriginationPhoneNumber=origination_number,
            CallerId=caller_id,
            Content={
                "SSMLMessage": {
                    "LanguageCode": language_code,
                    "VoiceId": voice_id,
                    "Text": ssml_message,
                }
            },
        )
```

```
    except ClientError:
        logger.exception(
            "Couldn't send message from %s to %s.",
            origination_number,
            destination_number,
        )
        raise
    else:
        return response["MessageId"]


def main():
    origination_number = "+12065550110"
    caller_id = "+12065550199"
    destination_number = "+12065550142"
    language_code = "en-US"
    voice_id = "Matthew"
    ssml_message = (
        "<speak>"
        "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> "
        "using the <break strength='weak'/>AWS SDK for Python (Boto3). "
        "<amazon:effect phonation='soft'>Thank you for listening."
        "</amazon:effect>"
        "</speak>"
    )
    print(f"Sending voice message from {origination_number} to
 {destination_number}.")
    message_id = send_voice_message(
        boto3.client("pinpoint-sms-voice"),
        origination_number,
        caller_id,
        destination_number,
        language_code,
        voice_id,
        ssml_message,
    )
    print(f"Message sent!\nMessage ID: {message_id}")


if __name__ == "__main__":
    main()
```

使用 AWS 最終使用者傳訊簡訊和語音 API，第 2 版

Amazon Pinpoint 包含 API，稱為 SMS 和語音 API，第 2 版，專為傳送 SMS 和語音訊息而設計。Amazon Pinpoint API 主要是透過排程和事件型行銷活動和旅程傳送訊息，SMS 和語音 API 則提供了新特色和功能，可以直接向各收件人傳送 SMS 和語音訊息。您可以獨立於 Amazon Pinpoint 行銷活動和旅程功能使用 SMS 和語音 API，也可以同時使用兩者來容納不同的使用案例。如果您已使用 Amazon Pinpoint 傳送 SMS 或語音訊息，表示您的帳戶已設為使用此 API。

對於擁有多租戶架構 (例如獨立軟體開發廠商 (ISV) 的使用者而言)，這個 API 是個不錯的解決方案。此 API 可以更容易確保不同租戶的事件資料、起始電話號碼和選擇不接收清單都是分開的。

當您使用 SMS 和語音 API 時，建議您設定組態集和事件目的地。SMS 和語音 API 不會自動為您傳送的訊息發出事件資料。設定事件目的地可確保您擷取重要的事件資料，例如訊息交付和失敗事件。

此 API 的第 2 版早於第 1 版。如果您目前使用此 API 的第 1 版，它將繼續可用，而且您可以繼續使用它。如果您遷移到第 2 版，將會獲得其他功能，例如能夠建立電話號碼集區、以程式設計方式請求新的電話號碼，以及啟用或停用電話號碼的某些功能。

> **Note**
>
> 某些任務只能使用 Amazon Pinpoint 主控台完成。例如，驗證帳戶在 SMS 沙盒中時要使用的電話號碼，並註冊使用 10DLC。

如需 Amazon Pinpoint SMS 和語音第 2 版 API 的詳細資訊，請參閱 SMS 和語音第 2 版 API 參考。如需有關如何建立、設定和管理 AWS 最終使用者簡訊簡訊和語音資源的資訊，請參閱AWS 最終使用者簡訊簡訊使用者指南

使用 Amazon Pinpoint 產生一次性密碼 (OTPs)

Amazon Pinpoint 包含一次性密碼 (OTP) 管理功能，可用來產生新的一次性密碼，並以簡訊的形式傳送給收件人。

> **Important**
>
> 若要使用此功能，您的帳戶必須具有生產存取權和作用中的原始身分。如需詳細資訊，請參閱《 AWS 使用者傳訊簡訊使用者指南》中的關於 SMS/MMS 和語音沙盒和請求電話號碼。

在某些國家和地區，您必須先取得專用的電話號碼或起始 ID，才能傳送簡訊。例如傳送訊息給美國收件人時，您必須擁有專用的免付費電話號碼、10DLC 號碼或短碼。傳送訊息給印度收件人時，您必須要有已註冊寄件者 ID，寄件者 ID 包含主體實體 ID (PEID) 和範本 ID。使用 OTP 功能時，這些要求仍然適用。

若要使用此功能，您需要傳送和驗證 OTP 訊息的許可，詳情請參閱 一次性密碼。如果您需要判斷許可的協助，請參閱 疑難排解 Amazon Pinpoint 身分和存取管理。

您可以使用 Amazon Pinpoint API 中的 `SendOtpMessages` 操作，傳送 OTP 密碼給應用程式的使用者。您使用此 API 時，Amazon Pinpoint 會產生一組隨機代碼，並透過 SMS 訊息傳送給您的使用者。您的請求可以包含下列參數：

- `Channel` – 傳送 OTP 密碼所通過的通訊管道。目前只支援 SMS 訊息，因此唯一可接受的值是 SMS。
- `BrandName` – 與 OTP 密碼相關聯的品牌、公司或產品的名稱。此名稱最多可包含 20 個字元。

> **Note**
>
> Amazon Pinpoint 傳送 OTP 訊息時，品牌名稱會自動插入以下訊息範本：
>
> ```
> This is your One Time Password: {{otp}} from {{brand}}
> ```
>
> 因此如果您將 ExampleCorp 指定為品牌名稱，而 Amazon Pinpoint 產生的一次性密碼為 123456，您的使用者會收到以下訊息：
>
> ```
> This is your One Time Password: 123456 from ExampleCorp
> ```

- CodeLength – 傳送給收件人的 OTP 密碼的位數。OTP 密碼可以包含 5 到 8 位數字。
- ValidityPeriod – OTP 密碼的有效時間 (以分鐘為單位)。有效期限是 5 至 60 分鐘。
- AllowedAttempts – 收件人嘗試驗證 OTP 失敗的次數。如果嘗試次數超過此值，OTP 將自動失效。允許的嘗試次數上限是 5 次。
- Language – 傳送訊息時使用的語言，以 IETF BCP-47 格式表示。可接受的值為：
  - de-DE – 德文
  - en-GB – 英文 (英國)
  - en-US – 英文 (美國)
  - es-419 – 西班牙文 (拉丁美洲)
  - es-ES – 西班牙文
  - fr-CA – 法文 (加拿大)
  - fr-FR – 法文
  - it-IT – 義大利文
  - ja-JP – 日文
  - ko-KR – 韓文
  - pt-BR – 葡萄牙文 (巴西)
  - zh-CN – 簡體中文
  - zh-TW – 繁體中文
- OriginationIdentity – 用於傳送 OTP 密碼的起始身分 (例如長碼、短碼或寄件者 ID)。如果您使用長碼或免付費電話號碼傳送 OTP，電話號碼必須以 E.164 格式顯示。
- DestinationIdentity – 接收 OTP 密碼的電話號碼 (E.164 格式)。
- ReferenceId – 請求的唯一參考 ID。參考 ID 與您驗證 OTP 時提供的參考 ID 完全相符。參考 ID 可以包含 1 到 48 個字元。
- EntityId – 已在監管機構註冊的實體 ID。此參數目前只適用於傳送訊息給印度收件人。如果沒有要傳訊給印度收件人，可以省略此參數。
- TemplateId – 已在監管機構註冊的範本 ID。此參數目前只適用於傳送訊息給印度收件人。如果沒有要傳訊給印度收件人，可以省略此參數。

> **Note**
>
> 如需有關傳送訊息給印度收件人要求的詳細資訊，請參閱《Amazon Pinpoint 使用者指南》中的印度寄件者 ID 註冊程序。

為了確保您的 Amazon Pinpoint 帳戶已正確設定為傳送 OTP 訊息，您可以使用 AWS CLI 來傳送測試訊息。如需 的詳細資訊 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》。

若要使用 傳送測試 OTP 訊息 AWS CLI，請在終端機中執行 send-otp-message 命令：

```
aws pinpoint send-otp-message --application-id 7353f53e6885409fa32d07cedexample --send-
otp-message-request-parameters
 Channel=SMS,BrandName=ExampleCorp,CodeLength=5,ValidityPeriod=20,AllowedAttempts=5,Origination
```

針對上述命令執行以下事項：

- 將 *7353f53e6885409fa32d07cedexample* 取代為您的應用程式 ID。
- 以您的公司名稱取代 *ExampleCorp*。
- CodeLegth 以傳送給收件人的 OTP 代碼中的位數取代 *5*。
- ValidityPeriod 將 中的 *20* 取代為 OTP 程式碼將有效的時間，以分鐘為單位。
- 將 中的 *5* 取代AllowedAttempts為收件人無法成功嘗試驗證 OTP 的次數。
- 將 中的 *+18555550142* 取代OriginationIdentity為用於傳送 OTP 程式碼的原始身分。
- 將 中的 *+12065550007* DestinationIdentity 取代為要傳送 OTP 代碼的電話號碼。
- 以請求ReferenceId的唯一參考 ID 取代 中的 *SampleReferenceId*。

SendOtpMessage 回應

若傳送 OTP 訊息成功，會收到類似以下範例的回應：

```
{
    "MessageResponse": {
        "ApplicationId": "7353f53e6885409fa32d07cedexample",
        "RequestId": "255d15ea-75fe-4040-b919-096f2example",
        "Result": {
            "+12065550007": {
                "DeliveryStatus": "SUCCESSFUL",
                "MessageId": "nvrmgq9kq4en96qgp0tlqli3og1at6aexample",
                "StatusCode": 200,
                "StatusMessage": "MessageId: nvrmgq9kq4en96qgp0tlqli3og1at6aexample"
            }
        }
    }
}
```

在 Amazon Pinpoint 中驗證 OTP 訊息

傳送one-time-password後，您的應用程式可以呼叫 Amazon Pinpoint API 進行驗證。若要驗證 OTP 密碼，請呼叫 `VerifyOtpMessages` API。您的請求必須包含以下參數：

- `DestinationIdentity` – 接收 OTP 密碼的電話號碼 (E.164 格式)。
- `ReferenceId` – 將 OTP 密碼傳送給收件人時使用的參考 ID。參考 ID 必須完全相符。
- `Otp` – 您正在驗證的 OTP 密碼。

您可以使用 AWS CLI 來測試驗證程序。如需安裝和設定 的詳細資訊 AWS CLI，請參閱 AWS Command Line Interface 使用者指南。

若要使用 驗證 OTP AWS CLI，請在終端機中執行 verify-otp-message 命令：

```
aws pinpoint verify-otp-message --application-id 7353f53e6885409fa32d07cedexample --
verify-otp-message-request-parameters
 DestinationIdentity=+12065550007,ReferenceId=SampleReferenceId,Otp=01234
```

針對上述命令執行以下事項：

- 將 *7353f53e6885409fa32d07cedexample* 取代為您的應用程式 ID。
- 將 中的 *+12065550007* 取代`DestinationIdentity`為傳送 OTP 代碼的電話號碼。
- 以請求`ReferenceId`的唯一參考 ID 取代 中的 *SampleReferenceId*。此值必須符合用來傳送請求`ReferenceID`的 。
- `Otp` 將 中的 *01234* 取代為傳送至 的 Otp`DestinationIdentity`。

`VerifyOtpMessage` 回應

若向 `VerifyOTPMessage` API 傳送請求，它會傳回一個包含單一屬性 `Valid` 的 `VerificationResponse` 物件。如果參考 ID、電話號碼和 OTP 都與 Amazon Pinpoint 預期的值相符，且 OTP 尚未過期，則 `Valid` 的值將是 `true`；否則就是 `false`。以下是 OTP 驗證成功的回應範例：

```
{
    "VerificationResponse": {
        "Valid": true
    }
```

```
}
```

在 Amazon Pinpoint 中使用 SDK for Python (Boto3) 的 OTP 程式碼範例

本節包含的密碼範例，說明如何使用 SDK for Python (Boto3) 傳送和驗證 OTP 密碼。

產生參考 ID

以下功能會根據收件人的電話號碼、收件人接收 OTP 的產品或品牌，以及請求來源 (可能是網站或應用程式中的頁面名稱)，為每位收件人產生一組唯一的參考 ID。驗證 OTP 密碼時，您必須傳遞相同的參考 ID，才能驗證成功。傳送密碼和驗證密碼範例，都使用此公用程式函數。

此函數非必要，但如果要將 OTP 傳送和驗證程序範圍限定為特定交易，以在驗證期間輕鬆重新提交，此函數就很有用。您可以使用任何您要的參考 ID—這只是一個基本範例。但本節中的其他代碼範例仰賴此函數。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier:  Apache-2.0

import hashlib

def generate_ref_id(destinationNumber,brandName,source):
    refId = brandName + source + destinationNumber
    return hashlib.md5(refId.encode()).hexdigest()
```

傳送 OTP 代碼

以下程式碼範例說明如何使用 SDK for Python (Boto3) 傳送 OTP 密碼。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier:  Apache-2.0

import boto3
from botocore.exceptions import ClientError
from generate_ref_id import generate_ref_id

### Some variables that are unlikely to change from request to request. ###

# The AWS Region that you want to use to send the message.
```

```
region = "us-east-1"

# The phone number or short code to send the message from.
originationNumber = "+18555550142"

# The project/application ID to use when you send the message.
appId = "7353f53e6885409fa32d07cedexample"

# The number of times the user can unsuccessfully enter the OTP code before it becomes
 invalid.
allowedAttempts = 3

# Function that sends the OTP as an SMS message.
def send_otp(destinationNumber,codeLength,validityPeriod,brandName,source,language):
    client = boto3.client('pinpoint',region_name=region)
    try:
        response = client.send_otp_message(
            ApplicationId=appId,
            SendOTPMessageRequestParameters={
                'Channel': 'SMS',
                'BrandName': brandName,
                'CodeLength': codeLength,
                'ValidityPeriod': validityPeriod,
                'AllowedAttempts': allowedAttempts,
                'Language': language,
                'OriginationIdentity': originationNumber,
                'DestinationIdentity': destinationNumber,
                'ReferenceId': generate_ref_id(destinationNumber,brandName,source)
            }
        )

    except ClientError as e:
        print(e.response)
    else:
        print(response)

# Send a message to +14255550142 that contains a 6-digit OTP that is valid for 15
 minutes. The
# message will include the brand name "ExampleCorp", and the request originated from a
 part of your
# site or application called "CreateAccount". The US English message template should be
 used to
# send the message.
```

```
send_otp("+14255550142",6,15,"ExampleCorp","CreateAccount","en-US")
```

驗證 OTP 代碼

以下程式碼範例說明如何使用 SDK for Python (Boto3) 驗證已傳送的 OTP 密碼。為了使驗證步驟成功，您的請求包含的參考 ID，必須與傳送訊息的參考 ID 完全相符。

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier:  Apache-2.0

import boto3
from botocore.exceptions import ClientError
from generate_ref_id import generate_ref_id

# The AWS Region that you want to use to send the message.
region = "us-east-1"

# The project/application ID to use when you send the message.
appId = "7353f53e6885409fa32d07cedexample"

# Function that verifies the OTP code.
def verify_otp(destinationNumber,otp,brandName,source):
    client = boto3.client('pinpoint',region_name=region)
    try:
        response = client.verify_otp_message(
            ApplicationId=appId,
            VerifyOTPMessageRequestParameters={
                'DestinationIdentity': destinationNumber,
                'ReferenceId': generate_ref_id(destinationNumber,brandName,source),
                'Otp': otp
            }
        )

    except ClientError as e:
        print(e.response)
    else:
        print(response)

# Verify the OTP 012345, which was sent to +14255550142. The brand name ("ExampleCorp")
 and the
# source name ("CreateAccount") are used to generate the correct reference ID.
verify_otp("+14255550142","012345","ExampleCorp","CreateAccount")
```

使用 Amazon Pinpoint 和 Amplify 自訂應用程式內訊息

您可以使用應用程式內訊息，將鎖定受眾的訊息傳送給應用程式的使用者。應用程式內訊息可自由自訂。應用程式內訊息可以包含開啟網站的按鈕，或是將使用者帶到應用程式特定部分的按鈕。您可以設定背景和文字顏色、決定文字的位置，並在通知中加上按鈕和影像。您可以傳送單一訊息，或建立最多含五則不重複訊息的浮動切換。如需應用程式內訊息的概觀 (包括建立應用程式內訊息範本的說明)，請參閱 Amazon Pinpoint 使用者指南中的建立應用程式內範本。

您可以使用 AWS Amplify 將 Amazon Pinpoint 的應用程式內傳訊功能無縫整合至您的應用程式。Amplify 能夠自動處理以下程序：擷取訊息、轉譯訊息，並將分析資料傳送到 Amazon Pinpoint。React Native 應用程式目前支援此整合。詳情請參閱 Amplify Framework 文件中的應用程式內傳訊。

使用 Amazon Pinpoint 以程式設計方式擷取端點的應用程式內訊息

您的應用程式可以呼叫 GetInAppMessages API，以擷取指定端點有權存取的所有應用程式內訊息。呼叫 `GetInAppMessages` API 時，會提供以下參數：

- `ApplicationId` – 與應用程式內訊息行銷活動相關聯的 Amazon Pinpoint 專案的唯一 ID。
- `EndpointId` – 您擷取訊息的端點的唯一 ID。

您使用這些值呼叫 API 時，會傳回訊息清單。如需深入了解此操作產生的回應，請參閱 GetInAppMessages Amazon Pinpoint API 回應 JSON 範例。

您可以使用 AWS SDKs來呼叫 `GetInAppMessages`操作。以下程式碼範例，包含擷取應用程式內訊息的函數。

JavaScript

在單獨的模組中建立用戶端並將其匯出：

```
import { PinpointClient } from "@aws-sdk/client-pinpoint";
const REGION = "us-east-1";
const pinClient = new PinpointClient({ region: REGION });
export { pinClient };
```

擷取端點的應用程式內訊息：

```
// Import required AWS SDK clients and commands for Node.js
import { PinpointClient, GetInAppMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "./lib/pinClient.js";

("use strict");

//The Amazon Pinpoint application ID.
const projectId = "4c545b28d21a490cb51b0b364example";

//The ID of the endpoint to retrieve messages for.
const endpointId = "c5ac671ef67ee3ad164cf7706example";

const params = {
  ApplicationId: projectId,
  EndpointId: endpointId
};

const run = async () => {
  try {
    const data = await pinClient.send(new GetInAppMessagesCommand(params));
    console.log(JSON.stringify(data, null, 4));
    return data;
  } catch (err) {
    console.log("Error", err);
  }
};
run();
```

Python

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

def retrieve_inapp_messages(
            pinpoint_client, project_id, endpoint_id):
    """
    Retrieves the in-app messages that a given endpoint is entitled to.

    :param pinpoint_client: A Boto3 Pinpoint client.
```

```
    :param project_id: An Amazon Pinpoint project ID.
    :param endpoint_id: The ID of the endpoint to retrieve messages for.
    :return: A JSON object that contains information about the in-app message.
    """

    try:
        response = pinpoint_client.get_in_app_messages(
            ApplicationId=project_id,
            EndpointId=endpoint_id)
    except ClientError:
        logger.exception("Couldn't retrieve messages.")
        raise
    else:
        return response

def main():
    project_id = "4c545b28d21a490cb51b0b364example"
    endpoint_id = "c5ac671ef67ee3ad164cf7706example"
    inapp_response = retrieve_inapp_messages(
        boto3.client('pinpoint'), project_id, endpoint_id)
    print(inapp_response)

if __name__ == '__main__':
    main()
```

GetInAppMessages Amazon Pinpoint API 回應 JSON 範例

您呼叫 GetInAppMessages API 操作時，它會傳回指定端點有權存取的訊息的清單。接著您的應用程式可以根據回應中的值轉譯訊息。

呼叫 GetInAppMessages API 傳回的 JSON 物件範例如下：

```
{
  "InAppMessagesResponse":{
    "InAppMessageCampaigns":[
      {
        "CampaignId":"inAppTestCampaign-4c545b28d21a490cb51b0b364example",
        "DailyCap":0,
        "InAppMessage":{
          "Content":[
            {
              "BackgroundColor":"#f8e71c",
```

```
            "BodyConfig":{
              "Alignment":"CENTER",
              "Body":"This is a sample in-app message sent using Amazon Pinpoint.",
              "TextColor":"#d0021b"
            },
            "HeaderConfig":{
              "Alignment":"CENTER",
              "Header":"Sample In-App Message",
              "TextColor":"#d0021b"
            },
            "ImageUrl":"https://example.com/images/thumbnail.png",
            "PrimaryBtn":{
              "DefaultConfig":{
                "BackgroundColor":"#d0021b",
                "BorderRadius":50,
                "ButtonAction":"CLOSE",
                "Text":"Dismiss",
                "TextColor":"#f8e71c"
              }
            }
          }
        ],
        "Layout":"MIDDLE_BANNER"
      },
      "Priority":3,
      "Schedule":{
        "EndDate":"2021-11-06T00:08:05Z",
        "EventFilter":{
          "Dimensions":{
            "Attributes":{

            },
            "EventType":{
              "DimensionType":"INCLUSIVE",
              "Values":[
                "_session.start"
              ]
            },
            "Metrics":{

            }
          }
        }
      },
```

```
        "SessionCap":0,
        "TotalCap":0,
        "TreatmentId":"0"
      }
    ]
  }
}
```

下列各節提供有關此回應的元件及其屬性的資訊。

InAppMessageCampaigns 物件

InAppMessageCampaigns 物件包含以下屬性：

屬性	Description	設定是
CampaignId	這個字串包含訊息來源的 Amazon Pinpoint 行銷活動的名稱和唯一行銷活動 ID。名稱會在行銷活動 ID 之前。這兩個值會以連字號 (-) 分隔。	您建立行銷活動時，由 Amazon Pinpoint 自動建立。
TreatmentId	一個整數，代表此訊息的行銷活動處理方法 ID。如果行銷活動只有一種處理方法，則值是 0。	
Priority	應用程式內訊息的優先順序，以 1 到 5 之間的整數表示，1 表示優先順序最高，5 表示優先順序最低。	行銷活動建立程序的第 1 步。
InAppMessage	InAppMessage 物件 包含如何轉譯訊息的相關資訊。	根據您為行銷活動指定的應用程式內訊息範本的內容。
Schedule	Schedule 物件包含訊息傳送時間的相關資訊。	行銷活動建立程序的第 4 步 (如果行銷活動是在主控台建立) 或 Schedule 物件 (如果行

屬性	Description	設定是
		銷活動是使用 API 或 SDK 建立)。
DailyCap	24 小時內，可向使用者顯示應用程式內訊息的次數 (以整數顯示)。	繼承自專案層級設定。如果行銷活動包含覆寫專案設定的設定，則會改用這些設定。
SessionCap	應用程式工作階段期間，可向使用者顯示應用程式內訊息的次數 (以整數表示)。	
TotalCap	每個行銷活動可向端點顯示任何應用程式內訊息的總次數 (以整數表示)。	

InAppMessage 物件

InAppMessage 物件包含以下屬性：

屬性	Description	設定是
Content	這個陣列包含 InAppMessageContent 物件，該物件說明了訊息的內容。	根據您為行銷活動指定的應用程式內訊息範本的內容。
Layout	這個字串說明應用程式內訊息顯示在收件人裝置上的外觀。可能值為： • BOTTOM_BANNER – 在頁面底部顯示為橫幅的訊息。 • TOP_BANNER – 在頁面頂端顯示為橫幅的訊息。 • OVERLAYS – 覆蓋整個畫面的訊息。	

屬性	Description	設定是
	• MOBILE_FEED – 在頁面前方視窗中顯示的訊息。 • MIDDLE_BANNER – 在頁面中間顯示為橫幅的訊息。 • CAROUSEL – 最多五條唯一訊息的可捲動配置。	

HeaderConfig 物件

HeaderConfig 物件包含以下屬性：

屬性	Description	設定是
Alignment	指定標題文字對齊方式的字串。可能值為 LEFT、CENTER 及 RIGHT。	根據您為行銷活動指定的應用程式內訊息範本的內容。
Header	郵件標頭文字。	
TextColor	標頭文字的顏色，以十六進位顏色代碼表示 (例如 #000000 代表黑色)。	

BodyConfig 物件

BodyConfig 物件包含以下屬性：

屬性	Description	設定是
Alignment	指定訊息內文對齊方式的字串。可能值為 LEFT、CENTER 及 RIGHT。	根據您為行銷活動指定的應用程式內訊息範本的內容。
Body	訊息的主要本文。	

屬性	Description	設定是
TextColor	內文的顏色，以十六進位顏色代碼表示 (例如 #000000 代表黑色)。	

InAppMessageContent 物件

InAppMessageContent 物件包含以下屬性：

屬性	Description	設定是
BackgroundColor	應用程式內訊息的背景顏色，以十六進位顏色代碼表示 (例如 #000000 代表黑色)。	根據您為行銷活動指定的應用程式內訊息範本的內容。
BodyConfig	BodyConfig 物件，包含訊息內文的相關資訊。	
HeaderConfig	HeaderConfig 物件，包含訊息標頭或標題的相關資訊。	
ImageUrl	訊息內影像的 URL。	
PrimaryBtn	InAppMessageButton 物件，包含訊息內主要按鈕的相關資訊。	
SecondaryBtn	InAppMessageButton 物件，包含訊息內次要按鈕的相關資訊。如果應用程式內訊息範本沒有指定次要按鈕，就不存在。	

Schedule 物件

Schedule 物件包含以下屬性：

屬性	Description	設定是
EndDate	為行銷活動排定的結束時間 (依 ISO 8601 格式顯示)。	行銷活動建立程序的第 4 步 (如果行銷活動是在主控台建立) 或 Schedule 物件 (如果行銷活動是使用 API 或 SDK 建立)。
EventFilter	有關觸發應用程式內訊息的事件的相關資訊。您產生與 Amazon Pinpoint 應用程式內活動相符的事件時，將顯示該訊息。	

InAppMessageButton 物件

InAppMessageButton 物件包含以下屬性：

屬性	Description	設定是
DefaultConfig	DefaultButtonConfig 物件，包含應用程式內訊息中按鈕的預設設定的相關資訊。	根據您為行銷活動指定的應用程式內訊息範本的內容。
Android	OverrideButtonConfig 物件，指定按鈕在 Android 裝置上的行為。這會覆寫 DefaultConfig 物件中詳細說明的預設按鈕組態。	
IOS	OverrideButtonConfig 物件，指定按鈕在 iOS 裝置上的行為。這會覆寫 DefaultConfig 物件中詳細說明的預設按鈕組態。	
Web	OverrideButtonConfig 物件，指定按鈕在 Web 應用程式上的行為。這會覆寫 DefaultCo	

屬性	Description	設定是
	nfig 物件中詳細說明的預設按鈕組態。	

DefaultButtonConfig 物件

DefaultButtonConfig 物件包含以下屬性：

屬性	Description	設定是
BackgroundColor	按鈕的背景顏色，以十六進位顏色代碼表示 (例如 #000000 代表黑色)。	根據您為行銷活動指定的應用程式內訊息範本的內容。
BorderRadius	按鈕邊框的半徑 (以像素為單位) 以整數表示。數字越大，圓角的程度越大。	
ButtonAction	這個字串說明收件人選擇應用程式內訊息中的按鈕時發生的動作。可能值為： • LINK – Web 目的地的連結。 • DEEP_LINK – 應用程式中特定頁面的連結。 • CLOSE – 關閉訊息。	
Link	按鈕的目的地 URL。ButtonAction 是 CLOSE 的按鈕，沒有這個項目。	
Text	按鈕上顯示的文字。	

屬性	Description	設定是
TextColor	按鈕上文字的顏色，以十六進位顏色代碼表示 (例如 #000000 代表黑色)。	

OverrideButtonConfig 物件

只有應用程式內訊息範本使用覆寫按鈕時，才會出現 OverrideButtonConfig 物件。覆寫按鈕具有特定組態，適用於特定裝置類型 (例如 iOS 裝置、Android 裝置或網頁瀏覽器)。

OverrideButtonConfig 物件包含以下屬性：

屬性	Description	設定是
ButtonAction	收件人選擇應用程式內訊息中的按鈕時發生的動作。可能值為： • LINK – Web 目的地的連結。 • DEEP_LINK – 應用程式中特定頁面的連結。 • CLOSE – 關閉訊息。	根據您為行銷活動指定的應用程式內訊息範本的內容。
Link	按鈕的目的地 URL。ButtonAction 是 CLOSE 的按鈕，沒有這個項目。	
Text	按鈕上顯示的文字。	
TextColor	按鈕上文字的顏色，以十六進位顏色代碼表示 (例如 #000000 代表黑色)。	

使用 Amazon Pinpoint 電話號碼驗證服務

Amazon Pinpoint 包含電話號碼驗證服務，您可以利用該服務，判斷電話號碼是否有效，並取得有關電話號碼本身的其他資訊。例如，使用電話號碼驗證服務時，它會傳回以下資訊：

- 採用 E.164 的電話號碼格式。
- 電話號碼類型 (例如行動電話、市內電話或 VoIP)。
- 電話號碼所在的城市和國家。
- 與電話號碼相關聯的服務提供者。

使用電話號碼驗證服務需要支付額外的費用。詳情請參閱 Amazon Pinpoint 定價。

> **Important**
>
> 若使用美國和加拿大的電話號碼，電話號碼驗證 API 將不再傳回 `City`、`County`、`Timezone` 和 `ZipCode` 的資料。

Amazon Pinpoint 電話號碼驗證使用案例

您可以使用電話號碼驗證服務，來啟用多種使用案例，包括下列：

- 驗證網頁型表單上的電話號碼 – 如果您使用網頁型表單收集客戶的聯絡資訊，您可以在提交表單之前，驗證客戶提供的電話號碼。在網站的後端利用 Amazon Pinpoint API 來驗證號碼。API 回應指出該號碼是否無效，例如電話號碼格式是否不正確。如果您判斷客戶提供的電話號碼無效，您的 Web 表單可以提示客戶提供不同的號碼。
- 清理您現有的聯絡人資料庫 – 如果您有客戶電話號碼資料庫，您可以驗證每個電話號碼，然後根據調查結果更新資料庫。例如，如果您發現端點的電話號碼沒有接收 SMS 訊息的功能，您可以將端點的 `ChannelType` 屬性從 `SMS` 變更為 `VOICE`。您可以先驗證電話號碼，然後為新的或現有端點更新 `ChannelType` 屬性。如為單一端點請遵照 將端點新增至 Amazon Pinpoint 的指示，多個端點則請參照 將一批端點新增至 Amazon Pinpoint。
- 傳送訊息之前選擇正確的管道 – 如果您想傳送 SMS 訊息，但發現目的地號碼無效，您可以透過不同的管道將訊息傳送給收件人。例如，如果端點無法接收 SMS 訊息，您可以改為傳送語音訊息。

使用 驗證電話號碼 AWS CLI

下列範例示範如何使用 驗證電話號碼 AWS CLI。如需詳細資訊，請參閱《 AWS CLI 命令參考》中的 phone-number-validate。如需驗證回應的範例，請參閱 電話號碼驗證回應。如需設定 的詳細資訊 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》中的設定 AWS CLI。

使用 來使用電話號碼驗證服務 AWS CLI

- 在命令列中輸入以下命令：

```
aws pinpoint phone-number-validate --number-validate-request
 PhoneNumber=+442079460881,IsoCountryCode=GB
```

在上述命令中，將 *+442079460881* 取代為您要驗證的電話號碼，並將 *GB* 取代為兩位數 ISO 國家或地區代碼。

> **Note**
>
> 當您向電話號碼驗證服務提供的電話號碼時，應該一律包含國家代碼。如果您未包括國家代碼，服務可能會傳回不同國家電話號碼的資訊。您可以在電話號碼中使用破折號，例如 *+44-207-946-0881*。

電話號碼驗證回應

電話號碼驗證服務提供的資訊，會因您提供的電話號碼可用的資料稍有不同。本節包含電話號碼驗證服務傳回的回應範例。

> **Note**
>
> 電話號碼驗證服務提供的資料，是基於電信服務提供者和全球的其他實體提供的資訊。某些國家的提供者更新此資訊的頻率，可能較其他國家中的提供者低。例如，如果您發出請求來驗證行動電話號碼，而您提供的號碼是從不同行動電信業者移植，來自電話號碼驗證服務的回應可能會包含原始電信業者名稱，而不是目前的電信業者。

有效行動電話號碼

傳送請求到電話號碼驗證服務，而且電話號碼是有效的行動電話號碼時，它會傳回類似以下範例的資訊：

```
{
    "NumberValidateResponse": {
        "Carrier": "ExampleCorp Mobile",
        "City": "Seattle",
        "CleansedPhoneNumberE164": "+12065550142",
        "CleansedPhoneNumberNational": "2065550142",
        "Country": "United States",
        "CountryCodeIso2": "US",
        "CountryCodeNumeric": "1",
        "OriginalPhoneNumber": "+12065550142",
        "PhoneType": "MOBILE",
        "PhoneTypeCode": 0,
        "Timezone": "America/Los_Angeles",
        "ZipCode": "98101"
    }
}
```

有效的市內電話號碼

如果您的請求包含有效的市內電話號碼，電話號碼驗證服務會傳回類似以下範例的資訊：

```
{
    "CountryCodeIso2": "US",
    "CountryCodeNumeric": "1",
    "Country": "United States",
    "City": "Santa Clara",
    "ZipCode": "95037",
    "Timezone": "America/Los_Angeles",
    "CleansedPhoneNumberNational": "4085550101",
    "CleansedPhoneNumberE164": "14085550101",
    "Carrier": "AnyCompany",
    "PhoneTypeCode": 1,
    "PhoneType": "LANDLINE",
    "OriginalPhoneNumber": "+14085550101"
}
```

有效 VoIP 電話號碼

如果您的請求包含有效的網路電話 (VoIP) 電話號碼，電話號碼驗證服務會傳回類似以下範例的資訊：

```
{
    "NumberValidateResponse": {
        "Carrier": "ExampleCorp",
        "City": "Countrywide",
        "CleansedPhoneNumberE164": "+441514960001",
        "CleansedPhoneNumberNational": "1514960001",
        "Country": "United Kingdom",
        "CountryCodeIso2": "GB",
        "CountryCodeNumeric": "44",
        "OriginalPhoneNumber": "+441514960001",
        "PhoneType": "VOIP",
        "PhoneTypeCode": 2
    }
}
```

電話號碼無效

如果您的請求包含無效的電話號碼，電話號碼驗證服務會傳回類似以下範例的資訊：

```
{
    "NumberValidateResponse": {
        "CleansedPhoneNumberE164": "+44163296076",
        "CleansedPhoneNumberNational": "163296076",
        "Country": "United Kingdom",
        "CountryCodeIso2": "GB",
        "CountryCodeNumeric": "44",
        "OriginalPhoneNumber": "+440163296076",
        "PhoneType": "INVALID",
        "PhoneTypeCode": 3
    }
}
```

請注意，此回應中的 PhoneType 屬性指出此電話號碼為 INVALID，且不包含有關電信業者或與電話號碼相關聯位置的資訊。您應該避免傳送簡訊或語音訊息到 PhoneType 為 INVALID 的電話號碼，因為這些號碼很可能屬於實際的收件人。

其他電話號碼

有時候，來自電話號碼驗證服務的回應會包含一個 OTHER 的 PhoneType 值。服務可能會在下列情況中傳回這類回應：

- 電話號碼是免付費號碼。

- 電話號碼保留在電視節目和影片中使用，例如，以 555 為開頭的北美電話號碼。
- 電話號碼包含目前未使用的區域碼，例如，999 區域碼在北美。
- 電話號碼保留供某些其他用途。

以下範例顯示當您的請求包含虛構的北美電話號碼時，電話號碼驗證服務提供的回應：

```
{
    "NumberValidateResponse": {
        "Carrier": "Multiple OCN Listing",
        "CleansedPhoneNumberE164": "+14255550199",
        "CleansedPhoneNumberNational": "4255550199",
        "Country": "United States",
        "CountryCodeIso2": "US",
        "CountryCodeNumeric": "1",
        "OriginalPhoneNumber": "+14255550199",
        "PhoneType": "OTHER",
        "PhoneTypeCode": 4,
        "Timezone": "America/Los_Angeles"
    }
}
```

預付電話號碼

如果您的請求包含有效的預付電話號碼，電話號碼驗證服務將傳回類似以下範例的資訊：

```
{
    "NumberValidateResponse": {
        "Carrier": "ExampleCorp",
        "City": "Countrywide",
        "CleansedPhoneNumberE164": "+14255550199",
        "CleansedPhoneNumberNational": "4255550199",
        "Country": "United States",
        "CountryCodeIso2": "US",
        "CountryCodeNumeric": "1",
        "OriginalPhoneNumber": "+14255550199",
        "PhoneType": "PREPAID",
        "PhoneTypeCode": 5
    }
}
```

如需深入了解這些回應中包含的資訊，請參閱《Amazon Pinpoint API 參考》中的電話號碼驗證。

使用 Webhook 或 Lambda 函數在 Amazon Pinpoint 中建立自訂頻道

Amazon Pinpoint 包含內建支援，幫助您透過推播通知、電子郵件、SMS 和語音管道傳送訊息。建立自訂頻道，也可以設定 Amazon Pinpoint 透過其他管道傳送訊息。Amazon Pinpoint 中的自訂頻道可讓您透過任何具有 API 的服務 (包括第三方服務) 傳送訊息。您可以使用 Webhook 或呼叫AWS Lambda 函數與 APIs 互動。

您傳送自訂頻道行銷活動的區段可以包含所有類型的端點 (意即，`ChannelType` 屬性值為 EMAIL (電子郵件)、VOICE (語音)、SMS、CUSTOM (自訂) 或其中一種推播通知端點類型)。

使用 Webhook

如果您使用 Webhook 傳送自訂頻道訊息，Webhook 的 URL 必須以「https：//」開頭。Webhook URL 只能包含英數字元，加上下列符號：連字號 (-)、句號 (.)、底線 (_)、波狀符號 (~)、問號 (?)、斜線或實線 (/)、井號或雜湊符號 (#)，以及分號 (:)。URL 必須符合 RFC3986 的規定。

您建立指定 webhook URL 的行銷活動時，Amazon Pinpoint 會向該 URL 發出 HTTP `HEAD`。對 `HEAD` 請求的回應必須包含一個名為 `X-Amz-Pinpoint-AccountId` 標頭。此標頭的值必須等於AWS您的帳戶 ID。

使用 Lambda 函數

如果您選擇透過建立 Lambda 函數來傳送自訂頻道訊息，最好先熟悉 Amazon Pinpoint 發出的資料。Amazon Pinpoint 行銷活動透過自訂管道傳送訊息時，會將承載傳送到與以下範例類似的目標 Lambda 函數：

```
{
  "Message":{},
  "Data":"The payload that's provided in the CustomMessage object in
 MessageConfiguration",
  "ApplicationId":"3a9b1f4e6c764ba7b031e7183example",
  "CampaignId":"13978104ce5d6017c72552257example",
  "TreatmentId":"0",
  "ActivityId":"575cb1929d5ba43e87e2478eeexample",
  "ScheduledTime":"2020-04-08T19:00:16.843Z",
```

```
  "Endpoints":{
    "1dbcd396df28ac6cf8c1c2b7fexample":{
      "ChannelType":"EMAIL",
      "Address":"mary.major@example.com",
      "EndpointStatus":"ACTIVE",
      "OptOut":"NONE",
      "Location":{
        "City":"Seattle",
        "Country":"USA"
      },
      "Demographic":{
        "Make":"OnePlus",
        "Platform":"android"
      },
      "EffectiveDate":"2020-04-01T01:05:17.267Z",
      "Attributes":{
        "CohortId":[
          "42"
        ]
      },
      "CreationDate":"2020-04-01T01:05:17.267Z"
    }
  }
}
```

事件資料可提供以下屬性：

- `ApplicationId` – 行銷活動所屬的 Amazon Pinpoint 專案的 ID。
- `CampaignId` – 調用 Lambda 函數的 Amazon Pinpoint 行銷活動的 ID。
- `TreatmentId` – 行銷活動變體的 ID。如果您建立了標準行銷活動，此值永遠為 0。如果您建立 A/B 測試行銷活動，此值是介於 0 到 4 之間的整數。
- `ActivityId` – 行銷活動正在執行的活動的 ID。
- `ScheduledTime` – Amazon Pinpoint 執行活動的時間，以 ISO 8601 格式顯示。
- `Endpoints` – 行銷活動鎖定的端點的清單。每個承載最多可包含 50 個端點。如果行銷活動傳送的客群包含超過 50 個端點，Amazon Pinpoint 會重複調用函數，一次最多呼叫 50 個端點，直到所有端點處理完畢為止。

您可以在建立和測試自訂管道 Lambda 函數時使用此範例資料。

使用 Amazon Pinpoint API 將 Lambda 函數或 Webhook 指派給個別行銷活動

若要將 Lambda 函數或 Webhook 指派給個別行銷活動，請使用 Amazon Pinpoint API 建立或更新行銷活動物件。

行銷活動中的 MessageConfiguration 物件也必須包含 CustomMessage 物件。此物件有一個成員：Data。Data 的值是 JSON 字串，其中包含您要傳送至自訂頻道的訊息承載。

該活動必須包含一個 CustomDeliveryConfiguration 物件。在 CustomDeliveryConfiguration 物件內，請指定下列項目：

- EndpointTypes – 一個陣列，包含了應收到自訂管道行銷活動的所有端點類型。它可以包含下列任何或所有頻道類型：
  - ADM
  - APNS
  - APNS_SANDBOX
  - APNS_VOIP
  - APNS_VOIP_SANDBOX
  - BAIDU
  - CUSTOM
  - EMAIL
  - GCM
  - SMS
  - VOICE
- DeliveryUri – 收到端點的目的地。您可以指定下列其中一個選項：
  - 行銷活動進行時，您想要傳送端點資料的 webhook URL。
  - 您要在行銷活動執行時，執行的 Lambda 函數的 Amazon Resource Name (ARN)。

> **Note**
>
> 該 Campaign 物件也可以包含一個 Hook 物件。此物件只在執行行銷活動時，建立由 Lambda 函數自訂的客群。如需詳細資訊，請參閱使用 AWS Lambda 函數自訂 Amazon Pinpoint 客群。

為 Amazon Pinpoint 行銷活動建立和設定 Lambda 函數

本節提供建立透過自訂頻道傳送訊息之 Lambda 函數的步驟概觀。首先，您建立函數。然後，您將執行政策新增至函數。此政策允許 Amazon Pinpoint 在行銷活動執行時，執行該政策。

如需建立 Lambda 函數的簡介，請參閱AWS Lambda開發人員指南中的建置 Lambda 函數。

Lambda 函數範例

以下程式碼範例會處理承載，並在 CloudWatch 中記錄每個端點類型的端點數。

```
import boto3
import random
import pprint
import json
import time

cloudwatch = boto3.client('cloudwatch')

def lambda_handler(event, context):
    customEndpoints = 0
    smsEndpoints = 0
    pushEndpoints = 0
    emailEndpoints = 0
    voiceEndpoints = 0
    numEndpoints = len(event['Endpoints'])

    print("Payload:\n", event)
    print("Endpoints in payload: " + str(numEndpoints))

    for key in event['Endpoints'].keys():
        if event['Endpoints'][key]['ChannelType'] == "CUSTOM":
            customEndpoints += 1
        elif event['Endpoints'][key]['ChannelType'] == "SMS":
            smsEndpoints += 1
```

```
        elif event['Endpoints'][key]['ChannelType'] == "EMAIL":
            emailEndpoints += 1
        elif event['Endpoints'][key]['ChannelType'] == "VOICE":
            voiceEndpoints += 1
        else:
            pushEndpoints += 1

    response = cloudwatch.put_metric_data(
        MetricData = [
            {
                'MetricName': 'EndpointCount',
                'Dimensions': [
                    {
                        'Name': 'CampaignId',
                        'Value': event['CampaignId']
                    },
                    {
                        'Name': 'ApplicationId',
                        'Value': event['ApplicationId']
                    }
                ],
                'Unit': 'None',
                'Value': len(event['Endpoints'])
            },
            {
                'MetricName': 'CustomCount',
                'Dimensions': [
                    {
                        'Name': 'CampaignId',
                        'Value': event['CampaignId']
                    },
                    {
                        'Name': 'ApplicationId',
                        'Value': event['ApplicationId']
                    }
                ],
                'Unit': 'None',
                'Value': customEndpoints
            },
            {
                'MetricName': 'SMSCount',
                'Dimensions': [
                    {
                        'Name': 'CampaignId',
```

```
                    'Value': event['CampaignId']
                },
                {
                    'Name': 'ApplicationId',
                    'Value': event['ApplicationId']
                }
            ],
            'Unit': 'None',
            'Value': smsEndpoints
        },
        {
            'MetricName': 'EmailCount',
            'Dimensions': [
                {
                    'Name': 'CampaignId',
                    'Value': event['CampaignId']
                },
                {
                    'Name': 'ApplicationId',
                    'Value': event['ApplicationId']
                }
            ],
            'Unit': 'None',
            'Value': emailEndpoints
        },
        {
            'MetricName': 'VoiceCount',
            'Dimensions': [
                {
                    'Name': 'CampaignId',
                    'Value': event['CampaignId']
                },
                {
                    'Name': 'ApplicationId',
                    'Value': event['ApplicationId']
                }
            ],
            'Unit': 'None',
            'Value': voiceEndpoints
        },
        {
            'MetricName': 'PushCount',
            'Dimensions': [
                {
```

```
                    'Name': 'CampaignId',
                    'Value': event['CampaignId']
                },
                {
                    'Name': 'ApplicationId',
                    'Value': event['ApplicationId']
                }
            ],
            'Unit': 'None',
            'Value': pushEndpoints
        },
        {
            'MetricName': 'EndpointCount',
            'Dimensions': [
            ],
            'Unit': 'None',
            'Value': len(event['Endpoints'])
        },
        {
            'MetricName': 'CustomCount',
            'Dimensions': [
            ],
            'Unit': 'None',
            'Value': customEndpoints
        },
        {
            'MetricName': 'SMSCount',
            'Dimensions': [
            ],
            'Unit': 'None',
            'Value': smsEndpoints
        },
        {
            'MetricName': 'EmailCount',
            'Dimensions': [
            ],
            'Unit': 'None',
            'Value': emailEndpoints
        },
        {
            'MetricName': 'VoiceCount',
            'Dimensions': [
            ],
            'Unit': 'None',
```

```
                'Value': voiceEndpoints
            },
            {
                'MetricName': 'PushCount',
                'Dimensions': [
                ],
                'Unit': 'None',
                'Value': pushEndpoints
            }
        ],
        Namespace = 'PinpointCustomChannelExecution'
    )
    print("cloudwatchResponse:\n",response)
```

Amazon Pinpoint 行銷活動執行此 Lambda 函數時，Amazon Pinpoint 會向 Lambda 函數傳送客群成員清單。該函數會計算每個 `ChannelType` 端點的數目。然後將該資料傳送到 Amazon CloudWatch。您可以在 CloudWatch 主控台的指標部分中，檢視這些指標。這些指標可以在 PinpointCustomChannelExecution 命名空間中使用。

您可以修改此程式碼範例，讓它也連線至外部服務的 API，以便透過該服務傳送訊息。

Amazon Pinpoint 的 Lambda 函數回應格式

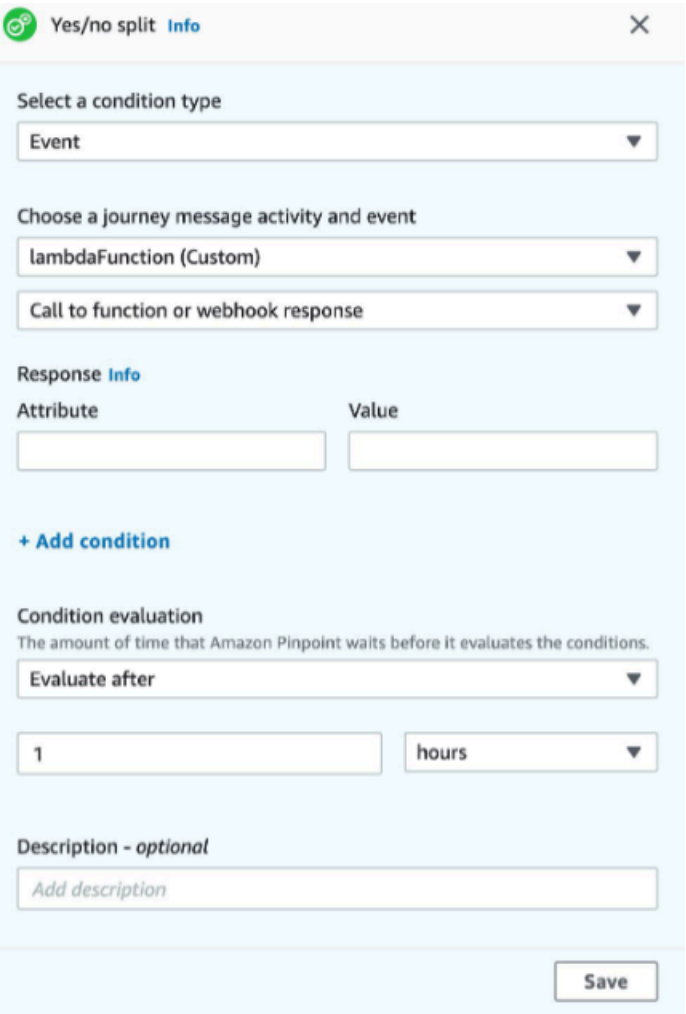
如果您想要在自訂頻道活動之後使用旅程多變量或是/否分割來判斷端點路徑，則必須將您的 Lambda 函數回應建構為 Amazon Pinpoint 可以理解的格式，然後沿著正確的路徑傳送端點。

回應的結構應依以下格式顯示：

```
{
    <Endpoint ID 1>:{
        EventAttributes: {
            <Key1>: <Value1>,
            <Key2>: <Value2>,
            ...
        }
    },
    <Endpoint ID 2>:{
        EventAttributes: {
            <Key1>: <Value1>,
            <Key2>: <Value2>,
            ...
        }
```

```
    },
...
}
```

這可讓您選擇一個索引鍵和值來決定端點路徑。



授予 Amazon Pinpoint 叫用 Lambda 函數的許可

您可以使用AWS Command Line Interface(AWS CLI) 將許可新增至指派給 Lambda 函數的 Lambda 函數政策。若要允許 Amazon Pinpoint 調用函數，請使用 Lambda add-permission 命令，如以下範例所示：

```
aws lambda add-permission \
--function-name myFunction \
--statement-id sid0 \
--action lambda:InvokeFunction \
--principal pinpoint.us-east-1.amazonaws.com \
```

```
--source-arn arn:aws:mobiletargeting:us-east-1:111122223333:apps/*
--source-account 111122223333
```

針對上述命令執行以下事項：

- 將 *myFunction* 替換成 Lambda 函數的名稱。
- 將 *us-east-1* 取代為您使用 Amazon Pinpoint AWS的區域。
- 將 *111122223333* 取代為您的AWS帳戶 ID。

執行 `add-permission` 命令時，Lambda 會傳回以下輸出：

```
{
  "Statement": "{\"Sid\":\"sid\",
    \"Effect\":\"Allow\",
    \"Principal\":{\"Service\":\"pinpoint.us-east-1.amazonaws.com\"},
    \"Action\":\"lambda:InvokeFunction\",
    \"Resource\":\"arn:aws:lambda:us-east-1:111122223333:function:myFunction\",
    \"Condition\":
      {\"ArnLike\":
        {\"AWS:SourceArn\":
         \"arn:aws:mobiletargeting:us-east-1:111122223333:apps/*\"}},
      {\"StringEquals\":
        {\"AWS:SourceAccount\":
          \"111122223333\"}}}
}
```

`Statement` 值是加入 Lambda 函數政策的陳述式的 JSON 字串版本。

進一步限制執行政策

將執行政策限定在特定 Amazon Pinpoint 專案，可以修改該執行策略。若要執行這項操作，請以專案的唯一 ID 取代上述範例 *。您可以將政策限制在特定行銷活動中，以進一步限制政策。例如，若要限制政策為只在有專案 ID `dbaf6ec2226f0a9a8615e3ea5example` 中的專案內允許有行銷活動 ID `95fee4cd1d7f5cd67987c1436example` 的行銷活動，請針對 `source-arn` 屬性使用下列值：

```
arn:aws:mobiletargeting:us-east-1:111122223333:apps/dbaf6ec2226f0a9a8615e3ea5example/
campaigns/95fee4cd1d7f5cd67987c1436example
```

> **Note**
>
> 如果您將 Lambda 函數的執行，限制為特定行銷活動，您必須先建立政策限制較少的函數。接著您必須在 Amazon Pinpoint 中建立行銷活動並選擇該函數。最後，您必須更新執行策略來引用指定的行銷活動。

使用 Amazon Pinpoint 透過 Kinesis 和 Firehose 串流應用程式事件資料

在 Amazon Pinpoint 中，事件是指以下動作：使用者與您其中一個應用程式互動、您從行銷活動或旅程傳送訊息，或您傳送交易 SMS 或電子郵件訊息。例如，如果您傳送電子郵件訊息，會發生數個事件：

- 當您傳送訊息時，就會發生「傳送」事件。
- 當訊息寄送到收件者的收件匣時，就會發生「傳遞」的事件。
- 當收件者開啟訊息時，會發生「開啟」的事件。

您可以設定 Amazon Pinpoint 傳送事件相關資訊到 Amazon Kinesis。Kinesis 平台提供可用來即時收集、處理和分析 AWS 服務資料的服務。Amazon Pinpoint 可以將事件資料傳送至 Firehose，而 Firehose 會將這些資料串流至 Amazon S3 或 Amazon Redshift 等 AWS 資料存放區。Amazon Pinpoint 也可以將資料串流到 Kinesis Data Streams，該串流會擷取和存放多個資料串流，供分析應用程式處理。

Amazon Pinpoint 事件串流，包含使用者與連接到 Amazon Pinpoint 的應用程式 (app) 互動的相關資訊。這也包含所有與您從行銷活動、透過任何管道和從旅程傳送的訊息相關資訊。這也可以包括您定義的任何自訂事件。最後，它包含您傳送的所有交易電子郵件和簡訊資訊。

> **Note**
>
> Amazon Pinpoint 不會串流交易推播通知或語音訊息相關資訊。

本章教您設定 Amazon Pinpoint，將事件資料串流到 Kinesis。亦包含 Amazon Pinpoint 串流的事件資料範例。

主題

- 設定 Amazon Pinpoint 透過 Amazon Kinesis 或 Amazon Data Firehose 串流應用程式事件資料
- 從 Amazon Pinpoint 的應用程式事件資料串流
- 來自 Amazon Pinpoint 的行銷活動事件資料串流
- 來自 Amazon Pinpoint 的旅程事件資料
- 從 Amazon Pinpoint 傳送電子郵件事件資料串流

- 來自 Amazon Pinpoint 的簡訊事件資料串流
- 從 Amazon Pinpoint 刪除事件串流

設定 Amazon Pinpoint 透過 Amazon Kinesis 或 Amazon Data Firehose 串流應用程式事件資料

您可以設定 Amazon Pinpoint 將事件資料傳送至 Amazon Kinesis 串流或 Amazon Data Firehose 交付串流。Amazon Pinpoint 可以傳送行銷活動、旅程、交易電子郵件和 SMS 訊息的事件資料。

本節包含透過編寫程式的方式設定事件串流的資訊。您也可以使用 Amazon Pinpoint 主控台設定事件串流。如需使用 Amazon Pinpoint 主控台設定事件串流的相關資訊，請參閱 Amazon Pinpoint 使用者指南中的事件串流設定。

先決條件

本節中的範例需要以下輸入：

- 與 Amazon Pinpoint 和報告事件整合的應用程式 ID。如需有關如何整合的資訊，請參閱將 Amazon Pinpoint 與您的應用程式整合。
- 您 AWS 帳戶中 Kinesis 串流或 Firehose 交付串流的 Amazon Resource Name (ARN)。如需有關建立這些資源的資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的建立和管理串流，或《Amazon Data Firehose 開發人員指南》中的建立 Amazon Data Firehose 交付串流。
- 授權 Amazon Pinpoint 將資料傳送至串流的 AWS Identity and Access Management (IAM) 角色 ARN。如需有關建立角色的資訊，請參閱將事件串流到 Kinesis 的 IAM 角色。

AWS CLI

下列 AWS CLI 範例使用 put-event-stream 命令。此命令設定 Amazon Pinpoint 將傳送事件到 Kinesis 串流：

```
aws pinpoint put-event-stream \
--application-id projectId \
--write-event-stream DestinationStreamArn=streamArn,RoleArn=roleArn
```

適用於 Java 的 AWS SDK

以下 Java 範例設定 Amazon Pinpoint 將事件傳送到 Kinesis 串流：

```
public PutEventStreamResult createEventStream(AmazonPinpoint pinClient,
        String appId, String streamArn, String roleArn) {

    WriteEventStream stream = new WriteEventStream()
            .withDestinationStreamArn(streamArn)
            .withRoleArn(roleArn);

    PutEventStreamRequest request = new PutEventStreamRequest()
            .withApplicationId(appId)
            .withWriteEventStream(stream);

    return pinClient.putEventStream(request);
}
```

此範例建構了 `WriteEventStream` 物件，以存放 Kinesis 串流和 IAM 角色的 ARN。將 `WriteEventStream` 物件傳遞到 `PutEventStreamRequest` 物件，以設定 Amazon Pinpoint 為特定應用程式串流事件。將 `PutEventStreamRequest` 物件傳遞至 Amazon Pinpoint 用戶端的 `putEventStream` 方法。

您可以將 Kinesis 串流指派給多個應用程式。如果您執行此操作，Amazon Pinpoint 會將每個應用程式中，以 base64 編碼的事件資料傳送到串流，讓您將資料當成集合分析。以下範例方法接受應用程式 (app) ID 的清單，並使用先前的範例方法 `createEventStream` 將串流指派給每個應用程式：

```
public List<PutEventStreamResult> createEventStreamFromAppList(
        AmazonPinpoint pinClient, List<String> appIDs,
        String streamArn, String roleArn) {

    return appIDs.stream()
            .map(appId -> createEventStream(pinClient, appId, streamArn,
                    roleArn))
            .collect(Collectors.toList());
}
```

雖然您可以將一個串流指派給多個應用程式，但您無法將多個串流指派給一個應用程式。

從 Amazon Pinpoint 的應用程式事件資料串流

將應用程式（應用程式）與 Amazon Pinpoint 整合並設定事件串流後，Amazon Pinpoint 會從您在設定期間指定的目的地擷取應用程式的使用者活動、自訂事件和訊息傳遞資料，供您檢視。如需如何

設定事件串流以便檢視事件資料的詳細資訊，請參閱 設定 Amazon Pinpoint 透過 Amazon Kinesis 或 Amazon Data Firehose 串流應用程式事件資料。

應用程式事件範例

應用程式事件的 JSON 物件包含以下範例所示的資料。

```
{
  "event_type": "_session.stop",
  "event_timestamp": 1487973802507,
  "arrival_timestamp": 1487973803515,
  "event_version": "3.0",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "cognito_identity_pool_id": "us-east-1:a1b2c3d4-e5f6-g7h8-i9j0-k1l2m3n4o5p6",
    "package_name": "main.page",
    "sdk": {
      "name": "aws-sdk-mobile-analytics-js",
      "version": "0.9.1:2.4.8"
    },
    "title": "title",
    "version_name": "1.0",
    "version_code": "1"
  },
  "client": {
    "client_id": "m3n4o5p6-a1b2-c3d4-e5f6-g7h8i9j0k1l2",
    "cognito_id": "us-east-1:i9j0k1l2-m3n4-o5p6-a1b2-c3d4e5f6g7h8"
  },
  "device": {
    "locale": {
      "code": "en_US",
      "country": "US",
      "language": "en"
    },
    "make": "generic web browser",
    "model": "Unknown",
    "platform": {
      "name": "android",
      "version": "10.10"
    }
  },
  "session": {
    "session_id": "f549dea9-1090-945d-c3d1-e4967example",
```

```
    "start_timestamp": 1487973202531,
    "stop_timestamp": 1487973802507
  },
  "attributes": {},
  "metrics": {}
}
```

應用程式事件屬性

本節定義應用程式事件串流先前範例中包含的屬性。

屬性	描述
event_type	事件的類型。可能值為： • _session.start – 端點已開始新的工作階段。 • _session.stop – 端點已結束工作階段。 • _userauth.sign_in – 端點已登入您的應用程式。 • _userauth.sign_up – 新端點已在應用程式中完成註冊程序。 • _userauth.auth_fail – 端點曾嘗試登入您的應用程式，但無法完成程序。 • _monetization.purchase – 端點在應用程式中購買了商品。 • _session.pause – 端點已暫停一個工作階段。暫停的工作階段可以繼續，所以您可繼續收集指標，不用啟動全新的工作階段。 • _session.resume – 端點已恢復一個工作階段。
event_timestamp	報告事件的時間，以 Unix 時間顯示 (毫秒)。
arrival_timestamp	Amazon Pinpoint 收到事件的時間，顯示為 Unix 時間 (以毫秒為單位)。
event_version	事件 JSON 結構描述的版本。

屬性	描述
	Tip 在您的事件處理應用程式中檢查此版本，讓您知道何時更新應用程式以回應結構描述更新。
`application`	與事件相關聯的 Amazon Pinpoint 專案的資訊。如需詳細資訊，請參閱應用程式表。
`client`	回報事件的端點相關資訊。如需詳細資訊，請參閱用戶端表。
`device`	回報事件的裝置資訊。如需詳細資訊，請參閱裝置表。
`session`	產生事件的工作階段資訊。如需詳細資訊，請參閱工作階段表。
`attributes`	與事件相關聯的屬性。針對您應用程式報告的事件，此物件包含您定義的自訂屬性。
`metrics`	與事件有關的指標。(選用) 您可以設定應用程式將自訂指標傳送到 Amazon Pinpoint。

應用程式

包含與事件相關聯的 Amazon Pinpoint 專案資訊。

屬性	描述
`app_id`	回報事件的 Amazon Pinpoint 專案的唯一 ID。
`cognito_identity_pool_id`	與端點相關聯的 Amazon Cognito 身分池的 ID。

屬性	描述
package_name	應用程式套件的名稱，例如 com.example.my_app 。
sdk	用於報告事件的開發套件資訊。如需詳細資訊，請參閱軟體開發套件表。
title	應用程式的名稱。
version_name	應用程式的版本名稱，例如 V2.5。
version_code	應用程式的版本號碼，例如 3。

SDK

包括用於報告事件的軟體開發套件資訊。

屬性	描述
name	用來報告事件的開發套件名稱。
version	開發套件的版本。

用戶端

包含產生事件的端點相關資訊。

屬性	描述
client_id	端點的 ID。
cognito_id	與端點相關聯的 Amazon Cognito ID 權杖。

裝置

包含產生事件之端點的裝置相關資訊。

屬性	描述
locale	端點裝置的語言和區域設定相關資訊。如需詳細資訊，請參閱地區設定表。
make	端點裝置的製造商。
model	端點裝置的型號識別碼。
platform	端點裝置上作業系統的相關資訊。如需詳細資訊，請參閱平台表。

Locale

包括端點裝置的語言和區域設定相關資訊。

屬性	描述
code	與裝置關聯的地區設定識別符。
country	與裝置地區設定相關聯的國家或區域。
language	與裝置地區設定相關聯的語言。

平台

包括端點裝置上作業系統的相關資訊。

屬性	描述
name	裝置上的作業系統名稱。
version	裝置上的作業系統版本。

Session (工作階段)

包含產生事件的工作階段資訊。

屬性	描述
session_id	識別工作階段的唯一 ID。
start_timestamp	工作階段開始的日期和時間，以 Unix 時間顯示 (毫秒)。
stop_timestamp	工作階段結束的日期和時間，以 Unix 時間顯示 (毫秒)。

來自 Amazon Pinpoint 的行銷活動事件資料串流

如果您使用 Amazon Pinpoint 透過管道傳送行銷活動，Amazon Pinpoint 可以串流這些行銷活動的事件資料。設定事件串流後，Amazon Pinpoint 會從您在設定期間指定的目的地擷取您從行銷活動傳送的電子郵件或簡訊的事件資料，供您檢視。如需 Amazon Pinpoint 串流用於電子郵件和簡訊資料的詳細資訊，請參閱 the section called “從 Amazon Pinpoint 傳送電子郵件事件資料串流”和 the section called “來自 Amazon Pinpoint 的簡訊事件資料串流”。如需如何設定事件串流的資訊，請參閱 設定 Amazon Pinpoint 透過 Amazon Kinesis 或 Amazon Data Firehose 串流應用程式事件資料。

行銷活動事件範例

行銷活動事件的 JSON 物件包含下列範例所示的資料。

```
{
  "event_type": "_campaign.send",
  "event_timestamp": 1562109497426,
  "arrival_timestamp": 1562109497494,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "d8dcf7c5-e81a-48ae-8313-f540cexample"
  },
  "device": {
    "platform": {}
  },
  "session": {},
```

```
  "attributes": {
    "treatment_id": "0",
    "campaign_activity_id": "5473285727f04865bc673e527example",
    "delivery_type": "GCM",
    "campaign_id": "4f8d6097c2e8400fa3081d875example",
    "campaign_send_status": "SUCCESS"
  },
  "client_context": {
    "custom": {
      "endpoint": "{\"ChannelType\":\"GCM\",\"EndpointStatus\":\"ACTIVE\",
          #\"OptOut\":\"NONE\",\"RequestId\":\"ec229696-9d1e-11e9-8bf1-85d0aexample\",
          #\"EffectiveDate\":\"2019-07-02T23:12:54.836Z\",\"User\":{}}"
    }
  },
  "awsAccountId": "123456789012"
}
```

行銷活動事件屬性

本區段定義包含在行銷活動事件串流中的屬性。

屬性	Description
event_type	事件的類型。可能值為： • _campaign.send – Amazon Pinpoint 執行了該行銷活動。 • _campaign.opened_notification – 適用於推播通知行銷活動，此事件類型表示收件人已點選並開啟通知。 • _campaign.received_foreground – 適用於推播通知行銷活動，此事件類型表示收件人已收到前景通知訊息。 • _campaign.received_background – 適用於推播通知行銷活動，此事件類型表示收件人已收到背景通知訊息。

屬性	Description
	Note 只有當您使用 AWS Amplify時，才會傳回 _campaign.opened_notification、_campaign.received_foreground 和 _campaign.received_background。如需將應用程式與 整合的詳細資訊 AWS Amplify。請參閱 使用 AWS Amplify 將您的前端應用程式連線至 Amazon Pinpoint。
`event_timestamp`	報告事件的時間，以 Unix 時間顯示 (毫秒)。
`arrival_timestamp`	Amazon Pinpoint 收到事件的時間，顯示為 Unix 時間 (以毫秒為單位)。
`event_version`	事件 JSON 結構描述的版本。 **Tip** 在您的事件處理應用程式中檢查此版本，讓您知道何時更新應用程式以回應結構描述更新。
`application`	與事件相關聯的 Amazon Pinpoint 專案的資訊。如需詳細資訊，請參閱應用程式表。
`client`	事件相關聯的端點資訊。如需詳細資訊，請參閱用戶端表。
`device`	回報事件的裝置資訊。若為行銷活動和交易訊息，此物件為空。
`session`	產生事件的工作階段資訊。若為行銷活動，此物件為空。

屬性	Description
attributes	與事件相關聯的屬性。若為您應用程式之一所報告的事件，此物件可包含由應用程式定義的自訂屬性。若為您傳送行銷活動時所建立的事件，此物件可包含與行銷活動相關聯的屬性。若為您傳送交易訊息時所產生的事件，此物件可包含訊息本身的相關資訊。 如需詳細資訊，請參閱屬性表。
client_context	包含一個 custom 物件，其中包含一個 endpoint 屬性。endpoint 屬性包含行銷活動傳送目標端點的端點紀錄內容。
awsAccountId	用來傳送訊息 AWS 的帳戶 ID。

應用程式

包含與事件相關聯的 Amazon Pinpoint 專案資訊。

屬性	Description
app_id	回報事件的 Amazon Pinpoint 專案的唯一 ID。
sdk	過去用來報告事件的開發套件。

屬性

包含產生事件的行銷活動資訊。

屬性	Description
treatment_id	如果訊息曾使用 A/B 測試行銷活動傳送，則此值代表訊息的處理方式號碼。若為標準行銷活動，此值為 0。

屬性	Description
`campaign_activity_id`	事件發生時，Amazon Pinpoint 產生的唯一 ID。

屬性	Description
delivery_type	行銷活動的交付方法。請勿將此屬性與 client_context 的 endpoint 屬性下指定的 ChannelType 欄位搞混。ChannelType 欄位通常基於訊息傳送的目的地端點而定。 對於只支援一種端點類型的管道，delivery_type 和 ChannelType 欄位的值相同。以電子郵件管道為例，delivery_type 和 ChannelType 欄位的值與 EMAIL 相同。 但對於支援不同端點類型的管道 (例如自訂頻道) 而言，情況並不總是如此。您可以針對不同的端點使用自訂管道，例如 EMAIL、SMS、CUSTOM 等。就本案例而言，delivery_type 識別自訂交付事件 CUSTOM，且 ChannelType 指定傳送行銷活動的目的地端點類型，例如 EMAIL、SMS、CUSTOM 等。如需建立自訂頻道的詳細資訊，請參閱 建立自訂頻道。 可能值為： • EMAIL • SMS • ADM • APNS • APNS_SANDBOX • APNS_VOIP • APNS_VOIP_SANDBOX • VOICE • GCM • BAIDU • PUSH

屬性	Description
	• CUSTOM
campaign_id	訊息傳送來源的行銷活動唯一 ID。

屬性	Description
campaign_send_status	指出目標端點的行銷活動狀態。可能的值包括： • SUCCESS – 行銷活動已成功傳送到端點。 • FAILURE – 行銷活動未傳送到端點。 • DAILY_CAP – 行銷活動未傳送到端點，因為已向端點傳送了每日訊息數上限。 • EXPIRED – 行銷活動未傳送到端點，因為傳送該行銷活動，將超出該行銷活動的持續時間上限或傳送速率設定。 • QUIET_TIME – 因靜止時間限制，未將行銷活動傳送到端點。 • HOLDOUT – 行銷活動未傳送到該端點，因為該端點是保留群組的成員。 • DUPLICATE_ADDRESS – 客群中有重複的端點地址。行銷活動已傳送到端點地址一次。 • QUIET_TIME – 因靜止時間限制，未將行銷活動傳送到端點。 • CAMPAIGN_CAP – 行銷活動未傳送到該端點，因為從這個行銷活動傳送到端點的訊息數目已達到上限。 • FAILURE_PERMANENT – 傳送到端點時發生永久失敗。 • TRANSIENT_FAILURE – 傳送到端點時發生暫時失敗。 • THROTTLED – 傳送已被限流。 • UNKNOWN – 未知的失敗。 • HOOK_FAILURE – 行銷活動勾點失敗。 • CUSTOM_DELIVERY_FAILURE – 自訂傳送失敗。 • RECOMMENDATION_FAILURE – 推薦者失敗。

屬性	Description
	• UNSUPPORTED_CHANNEL – 不支援管道。

用戶端

包含行銷活動鎖定的目標端點資訊。

屬性	Description
`client_id`	行銷活動傳送端點的 ID。

來自 Amazon Pinpoint 的旅程事件資料

當您發佈旅程時，Amazon Pinpoint 可以串流您從旅程傳送的電子郵件、簡訊、推送和自訂訊息的事件資料。設定事件串流之後，Amazon Pinpoint 會從您在設定期間指定的目的地擷取資料，供您檢視。如需 Amazon Pinpoint 串流用於電子郵件和簡訊資料的詳細資訊，請參閱 the section called “從 Amazon Pinpoint 傳送電子郵件事件資料串流”和 the section called “來自 Amazon Pinpoint 的簡訊事件資料串流”。如需如何設定事件串流的資訊，請參閱 設定 Amazon Pinpoint 透過 Amazon Kinesis 或 Amazon Data Firehose 串流應用程式事件資料。

旅程事件範例

旅程事件的 JSON 物件包含以下範例中顯示的資料。

```
{
   "event_type":"_journey.send",
   "event_timestamp":1572989078843,
   "arrival_timestamp":1572989078843,
   "event_version":"3.1",
   "application":{
      "app_id":"a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
      "sdk":{

      }
   },
   "client":{
      "client_id":"d8dcf7c5-e81a-48ae-8313-f540cexample"
```

```
   },
   "device":{
      "platform":{

      }
   },
   "session":{

   },
   "attributes":{
      "journey_run_id":"edc9a0b577164d1daf72ebd15example",
      "journey_send_status":"SUCCESS",
      "journey_id":"546401670c5547b08811ac6a9example",
      "journey_activity_id":"0yKexample",
      "journey_activity_type": "EMAIL",
      "journey_send_status_message": "200",
      "journey_send_status_code": "200"
   },
   "client_context":{
      "custom":{
         "endpoint":"{\"ChannelType\":\"EMAIL\",\"EndpointStatus\":\"ACTIVE\",\"OptOut
\":\"NONE\",\"Demographic\":{\"Timezone\":\"America/Los_Angeles\"}}"
      }
   },
   "awsAccountId":"123456789012"
}
```

旅程事件屬性

本節針對 Amazon Pinpoint 為旅程產生的事件串流資料所包含的屬性給予定義。

屬性	說明
event_type	事件的類型。若是旅程事件，此屬性的值一律為 _journey.send ，表示 Amazon Pinpoint 已執行旅程。
event_timestamp	報告事件的時間，以 Unix 時間顯示 (毫秒)。
arrival_timestamp	Amazon Pinpoint 收到事件的時間，顯示為 Unix 時間 (以毫秒為單位)。

屬性	說明
event_version	事件 JSON 結構描述的版本。 **Tip** 在您的事件處理應用程式中檢查此版本，讓您知道何時更新應用程式以回應結構描述更新。
application	與事件相關聯的 Amazon Pinpoint 專案的資訊。如需詳細資訊，請參閱應用程式表。
client	與事件相關聯的端點相關資訊。如需詳細資訊，請參閱用戶端表。
device	回報事件的裝置資訊。如為旅程，此物件為空。
session	產生事件的工作階段資訊。如為旅程，此物件為空。
attributes	與產生事件的旅程和旅程活動相關聯的屬性。如需詳細資訊，請參閱屬性表。
client_context	包含一個 custom 物件，其中包含一個 endpoint 屬性。endpoint 屬性包含與事件相關聯的端點記錄內容。
awsAccountId	用來執行旅程 AWS 的帳戶 ID。

應用程式

包含與事件相關聯的 Amazon Pinpoint 專案的資訊。

屬性	說明
app_id	回報事件的 Amazon Pinpoint 專案的唯一 ID。

屬性	說明
sdk	過去用來報告事件的開發套件。

用戶端

包含事件相關聯的端點相關資訊。

屬性	說明
client_id	端點的 ID。

屬性

包含產生事件的旅程相關資訊。

屬性	說明
journey_run_id	產生事件的旅程執行的唯一 ID。Amazon Pinpoint 會自動產生這個 ID，並指派給每個新的旅程執行。
journey_send_status	表示與事件相關聯的訊息交付狀態。可能的值包括： • SUCCESS – 訊息已成功傳送到端點。 • FAILURE – 訊息因為發生錯誤而未傳送到端點。 • CUSTOM_DELIVERY_FAILURE – 自訂傳送失敗。 • FAILURE_PERMANENT – 傳送到端點時發生永久失敗。

屬性	說明
	Tip 您可以篩選具有 FAILURE_PERMANENT 狀態的事件，並將 `journey_send_status_code` 設定為 403，以判斷是否存在存取政策和角色違規。對於有語音的對外行銷活動，這些例外狀況通常適用於連線行銷活動執行角色將 Amazon Pinpoint 旅程繫結至 Connect Customer 行銷活動的執行個體，因為傳輸中旅程執行不小心遭到刪除。 • THROTTLED – 傳送已被限流。 • UNSUPPORTED_CHANNEL – 不支援管道。 • DAILY_CAP – 該訊息未傳送到端點，因為傳送訊息將超出旅程或專案在 24 小時內可以傳送到單一端點的訊息數目上限。 • QUIET_TIME – 訊息因旅程或專案的靜止時間限制而未傳送。 • QUIET_TIME_MISSING_TIMEZONE – 未傳送訊息，因為時區估算無法估算端點的時區，且啟用了靜止時間。
`journey_id`	產生事件的旅程唯一 ID。
`journey_activity_id`	產生事件的旅程活動唯一 ID。

屬性	說明
journey_activity_type	事件的旅程活動類型。這可以是 EMAIL、SMS、PUSH、CONTACT_CENTER 或 CUSTOM。 **Note** VOICE 不是支援的旅程活動類型。當 journey_send_status 設定為 QUIET_TIME_WAIT_FINISHED 時，journey_activity_type 欄位不存在。
journey_send_status_message	傳送事件的狀態描述。
journey_send_status_code	請求的 HTTP 狀態碼。

從 Amazon Pinpoint 傳送電子郵件事件資料串流

如果您使用 Amazon Pinpoint 傳送電子郵件，Amazon Pinpoint 可以串流這些電子郵件的事件資料。設定事件串流之後，Amazon Pinpoint 會從您在設定期間指定的目的地擷取事件資料，供您檢視。如需如何設定事件串流的資訊，請參閱 設定 Amazon Pinpoint 透過 Amazon Kinesis 或 Amazon Data Firehose 串流應用程式事件資料。Amazon Pinpoint 會串流電子郵件訊息的下列事件類型相關資料：

- 傳送
- 交付
- 退信
- 投訴
- 開啟數
- 點按數
- 拒絕
- 取消訂閱
- 轉譯失敗

這些事件類型會在 中詳細說明電子郵件事件屬性。

根據您用來傳送電子郵件訊息的 API 和設定，您可能會看到其他事件類型或不同的資料。假設您使用將事件資料發佈到 Amazon Kinesis 的組態集傳送訊息 (例如 Amazon Simple Email Service (Amazon SES) 提供的組態集)，則資料也可以包含範本轉譯失敗的事件。如需該資料的相關資訊，請參閱 Amazon Simple Email Service 開發人員指南中的使用 Amazon SES 事件發佈進行監控。您必須先設定事件串流，才能檢視事件，請參閱 設定 Amazon Pinpoint 透過 Amazon Kinesis 或 Amazon Data Firehose 串流應用程式事件資料。設定事件串流時，您可以指定要儲存事件資料的目的地，然後使用目的地來擷取事件資料以供檢視。

電子郵件事件範例

電子郵件傳送

「電子郵件傳送」事件的 JSON 物件包含以下範例所示的資料。

```
{
  "event_type": "_email.send",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618622025,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "9a311b17-6f8e-4093-be61-4d0bbexample"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "received"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
```

```
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "send": {}
      }
    }
  }
}
```

電子郵件已交付

「電子郵件已交付」事件的 JSON 物件包含以下範例所示的資料。

```
{
  "event_type": "_email.delivered",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618622690,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
```

```
    "client_id": "e9a3000d-daa2-40dc-ac47-1cd34example"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "delivered"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "delivery": {
          "smtp_response": "250 ok:  Message 82080542 accepted",
          "reporting_mta": "a8-53.smtp-out.amazonses.com",
```

```
          "recipients": ["recipient@example.com"],
          "processing_time_millis": 1310
        }
      }
    }
  }
}
```

電子郵件點按

「電子郵件點按」事件的 JSON 物件包含以下範例所示的資料。

```
{
  "event_type": "_email.click",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618713751,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "49c1413e-a69c-46dc-b1c4-6470eexample"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "https://aws.amazon.com/pinpoint/"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
```

```
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }, {
            "name": "Message-ID",
            "value": "null"
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
            "subject": "Amazon Pinpoint Test"
          }
        },
        "click": {
          "ip_address": "72.21.198.67",
          "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6)
 AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.1.2 Safari/605.1.15",
          "link": "https://aws.amazon.com/pinpoint/"
        }
      }
    }
  }
}
```

電子郵件開啟

「電子郵件開啟」事件的 JSON 物件包含以下範例所示的資料。

```
{
  "event_type": "_email.open",
  "event_timestamp": 1564618621380,
  "arrival_timestamp": 1564618712316,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
```

```
    "sdk": {}
  },
  "client": {
    "client_id": "8dc1f651-b3ec-46fc-9b67-2a050example"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "feedback": "opened"
  },
  "awsAccountId": "123456789012",
  "facets": {
    "email_channel": {
      "mail_event": {
        "mail": {
          "message_id": "0200000073rnbmd1-mbvdg3uo-q8ia-m3ku-ibd3-ms77kexample-000000",
          "message_send_timestamp": 1564618621380,
          "from_address": "sender@example.com",
          "destination": ["recipient@example.com"],
          "headers_truncated": false,
          "headers": [{
            "name": "From",
            "value": "sender@example.com"
          }, {
            "name": "To",
            "value": "recipient@example.com"
          }, {
            "name": "Subject",
            "value": "Amazon Pinpoint Test"
          }, {
            "name": "MIME-Version",
            "value": "1.0"
          }, {
            "name": "Content-Type",
            "value": "multipart/alternative;  boundary=\"----=_Part_314159_271828\""
          }, {
            "name": "Message-ID",
            "value": "null"
          }],
          "common_headers": {
            "from": "sender@example.com",
            "to": ["recipient@example.com"],
```

```
            "subject": "Amazon Pinpoint Test"
          }
        },
        "open": {
          "ip_address": "72.21.198.67",
          "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6)
 AppleWebKit/605.1.15 (KHTML, like Gecko)"
        }
      }
    }
  }
}
```

電子郵件事件屬性

本節定義了 Amazon Pinpoint 在您傳送電子郵件訊息時產生的事件串流資料先前範例中包含的屬性。

屬性	描述
event_type	事件的類型。可能值為： • _email.send – Amazon Pinpoint 已收到訊息並嘗試交付給收件人。 • _email.delivered – 訊息已交付給收件人。 • _email.rejected – Amazon Pinpoint 判定訊息包含惡意軟體，所以未嘗試傳送。 • _email.hardbounce – 導致 Amazon Pinpoint 無法交付訊息的永久性問題。Amazon Pinpoint 不會再次嘗試交付訊息。 • _email.softbounce – 導致 Amazon Pinpoint 無法交付訊息的暫時性問題。Amazon Pinpoint 將在一段時間後，嘗試再次交付訊息。如果仍然無法交付訊息，不會再嘗試重試。電子郵件的最終狀態將是 SOFTBOUNCE。 • _email.complaint – 收件人收到訊息後，向電子郵件供應商回報該訊息為垃圾郵件 (例如使

屬性	描述
	用其電子郵件用戶端的「回報為垃圾郵件」功能)。 • _email.open – 收件人收到並開啟訊息。 • _email.click – 收件人收到訊息並點選其中的連結。 • _email.unsubscribe – 收件人收到訊息後，點選其中的取消訂閱連結。 • _email.rendering_failure – 因轉譯失敗而未傳送電子郵件。範本資料遺失或範本參數與資料不符時，可能出現這種情況。
`event_timestamp`	傳送訊息的時間，以 Unix 時間顯示 (毫秒)。一般而言，此值在所有針對訊息產生的事件中都是相同的。
`arrival_timestamp`	Amazon Pinpoint 收到事件的時間，顯示為 Unix 時間 (以毫秒為單位)。
`event_version`	事件 JSON 結構描述的版本。 **Tip** 在您的事件處理應用程式中檢查此版本，讓您知道何時更新應用程式以回應結構描述更新。
`application`	與事件相關聯的 Amazon Pinpoint 專案的資訊。如需詳細資訊，請參閱「應用程式」表。
`client`	回報事件之裝置上安裝的應用程式用戶端資訊。如需詳細資訊，請參閱用戶端表。

屬性	描述
device	回報事件的裝置資訊。如需詳細資訊，請參閱裝置表。 若為電子郵件事件，此物件為空。
session	若為電子郵件事件，此物件為空。
attributes	與事件相關聯的屬性。如需詳細資訊，請參閱屬性表。 若為您應用程式之一所報告的事件，此物件可包含由應用程式定義的自訂屬性。若為您傳送來自行銷活動或旅程的訊息時所建立的事件，此物件可包含與行銷活動或旅程相關聯的屬性。若為您傳送交易訊息時所產生的事件，此物件可包含訊息本身的相關資訊。
client_context	如為電子郵件事件，此物件包含 custom 物件，且此物件包含 legacy_identifier 屬性。legacy_identifier 屬性的值是訊息傳送來源的專案 ID。
facets	關於訊息的其他資訊，例如電子郵件標題。如需詳細資訊，請參閱「面向」表。
awsAccountId	用來傳送訊息 AWS 的帳戶 ID。

應用程式

包含與事件相關聯的 Amazon Pinpoint 專案資訊。

屬性	描述
app_id	回報事件的 Amazon Pinpoint 專案的唯一 ID。
sdk	過去用來報告事件的開發套件。如果您透過直接呼叫 Amazon Pinpoint API 或使用 Amazon

屬性	描述
	Pinpoint 主控台傳送交易電子郵件，此物件會是空的。

Attributes

包含產生事件的行銷活動或旅程相關資訊。

Campaign

包含產生事件的行銷活動資訊。

屬性	描述
`feedback`	若為 `_email.click` 事件，此屬性的值為收件人在訊息中點按以產生事件的連結 URL。若為其他事件，此值代表事件類型 (例如 `received`、`opened` 或 `clicked`)。
`treatment_id`	如果訊息曾使用 A/B 測試行銷活動傳送，則此值代表訊息的處理方式號碼。若為標準行銷活動和交易電子郵件訊息，此值為 `0`。
`campaign_activity_id`	事件發生時，Amazon Pinpoint 產生的唯一 ID。
`campaign_id`	傳送訊息的行銷活動唯一 ID。

旅程

包含產生事件的旅程相關資訊。

屬性	描述
`journey_run_id`	傳送訊息的旅程的唯一 ID。Amazon Pinpoint 會自動產生這個 ID，並指派給每個新的旅程執行。

屬性	描述
feedback	若為 _email.click 事件，此屬性的值為收件人在訊息中點按以產生事件的連結 URL。若為其他事件，此值代表事件類型 (例如 received、delivered 或 opened)。
journey_id	傳送訊息的旅程唯一 ID。
journey_activity_id	傳送郵件的旅程活動唯一 ID。

用戶端

行銷活動或旅程鎖定的用戶端的唯一識別符。

屬性	描述
client_id	用戶端的 ID。該值是行銷活動和旅程的端點 ID，對於交易傳送則是 UUID。

面向

包含訊息和事件類型的相關資訊。

屬性	描述
email_channel	包含 mail_event 物件，其包含兩個物件：mail 以及與事件類型對應的物件。

Mail

包含電子郵件訊息內容的相關資訊，以及有關訊息的中繼資料。

屬性	描述
message_id	訊息的唯一 ID。Amazon Pinpoint 接受訊息時，會自動產生這個 ID。
message_send_timestamp	訊息傳送的日期和時間，以 RFC 822 中指定的格式。
from_address	訊息傳送來源的電子郵件地址。
destination	陣列，訊息曾傳送至其包含的電子郵件地址。
headers_truncated	指出電子郵件標題是否遭截斷的布林值。
headers	物件，其包含的數個名稱/值對會對應到訊息中的標題。此物件通常包含下列標頭的資訊： • From – 寄件者的電子郵件地址。 • To – 收件人的電子郵件地址。 • Subject – 電子郵件的主旨行。 **Tip** campaign _email.send 事件不包含主旨標題。 • MIME-Version – 表示訊息依 MIME 格式顯示。如果此標頭存在，則值一律為 1.0。 • Content-Type – 訊息內容的 MIME 媒體類型。
common_headers	包含電子郵件訊息的多個常見標題相關資訊。這些資訊可能包括訊息送出的日期、收件人、寄件者及訊息的主旨列。

來自 Amazon Pinpoint 的簡訊事件資料串流

如果為專案啟用了 SMS 管道，Amazon Pinpoint 可以針對為該專案傳送的 SMS 訊息，串流事件資料。設定事件串流之後，Amazon Pinpoint 會從您在設定期間指定的目的地擷取事件資料，供您檢視。如需如何設定事件串流的資訊，請參閱 設定 Amazon Pinpoint 透過 Amazon Kinesis 或 Amazon Data Firehose 串流應用程式事件資料。

> **Note**
>
> 電信業者產生的簡訊事件最多可能需要 72 小時才會收到，不應用於判斷傳出訊息傳遞是否有延遲。72 小時後，如果 Amazon Pinpoint 尚未收到電信業者的最終事件，服務會自動傳回 UNKNOWN record_status，因為 Amazon Pinpoint 不知道該訊息發生了什麼情況。

SMS 事件範例

簡訊事件的 JSON 物件包含以下範例所示的資料。

```
{
  "event_type": "_SMS.SUCCESS",
  "event_timestamp": 1553104954322,
  "arrival_timestamp": 1553104954064,
  "event_version": "3.1",
  "application": {
    "app_id": "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6",
    "sdk": {}
  },
  "client": {
    "client_id": "123456789012"
  },
  "device": {
    "platform": {}
  },
  "session": {},
  "attributes": {
    "sender_request_id": "565d4425-4b3a-11e9-b0a5-example",
    "campaign_activity_id": "cbcfc3c5e3bd48a8ae2b9cb41example",
    "origination_phone_number": "+12065550142",
    "destination_phone_number": "+14255550199",
    "record_status": "DELIVERED",
    "iso_country_code": "US",
```

```
    "treatment_id": "0",
    "number_of_message_parts": "1",
    "message_id": "1111-2222-3333",
    "message_type": "Transactional",
    "campaign_id": "52dc44b35c4742c98c5935269example"
    "customer_context": "{\"userId\":\"user-id-4\"}"
  },
  "metrics": {
    "price_in_millicents_usd": 645.0
  },
  "awsAccountId": "123456789012"
}
```

簡訊事件屬性

本節定義 Amazon Pinpoint 在您傳送簡訊時產生事件串流資料先前範例中包含的屬性。

事件

屬性	描述
event_type	事件的類型。可能值為： • _SMS.BUFFERED – 訊息仍在交付給收件人的過程中。 • _SMS.SUCCESS – 訊息已成功由承運人接受/交付給收件人。 • _SMS.FAILURE – Amazon Pinpoint 無法將訊息交付給收件人。若要進一步了解無法交付訊息的錯誤，請參閱attributes.record_status 。 • _SMS.OPTOUT – 客戶收到訊息，並傳送選擇不接收關鍵字 (通常是 "STOP") 回應。
event_timestamp	報告事件的時間，以 Unix 時間顯示 (毫秒)。
arrival_timestamp	Amazon Pinpoint 收到事件的時間，顯示為 Unix 時間 (以毫秒為單位)。
event_version	事件 JSON 結構描述的版本。

屬性	描述
	Tip 在您的事件處理應用程式中檢查此版本，讓您知道何時更新應用程式以回應結構描述更新。
`application`	與事件相關聯的 Amazon Pinpoint 專案的資訊。如需詳細資訊，請參閱應用程式表。
`client`	回報事件的裝置中安裝的應用程式用戶端相關資訊。如需詳細資訊，請參閱用戶端表。
`device`	回報事件的裝置資訊。如需詳細資訊，請參閱裝置表。 若為簡訊事件，此物件為空。
`session`	若為簡訊事件，此物件為空。
`attributes`	與事件相關聯的屬性。若為您應用程式之一所報告的事件，此物件可包含由應用程式定義的自訂屬性。若為您傳送行銷活動時所建立的事件，此物件可包含與行銷活動相關聯的屬性。若為您傳送交易訊息時所產生的事件，此物件可包含訊息本身的相關資訊。 如需詳細資訊，請參閱屬性表。
`metrics`	與事件相關聯的其他指標。如需詳細資訊，請參閱指標表。
`awsAccountId`	用來傳送訊息 AWS 的帳戶 ID。

應用程式

包含與事件相關聯的 Amazon Pinpoint 專案的資訊，以及用於回報事件的 SDK (如果有的話)。

屬性	描述
app_id	回報事件的 Amazon Pinpoint 專案的唯一 ID。
sdk	過去用來報告事件的開發套件。如果您透過直接呼叫 Amazon Pinpoint API 或使用 Amazon Pinpoint 主控台傳送交易 SMS 訊息，此物件會是空的。

Attributes

包含與事件相關聯的屬性資訊。

屬性	描述
sender_request_id	與傳送簡訊請求相關聯的唯一 ID。
campaign_activity_id	行銷活動內的活動唯一 ID。
origination_phone_number	訊息傳送來源的電話號碼。
destination_phone_number	您嘗試傳送訊息的電話號碼。
record_status	訊息狀態的其他資訊。可能的值包括： • 成功/已交付給 – 訊息已成功交付。 • PENDING – 尚未將訊息交付至收件人裝置。 • INVALID – 目的地電話號碼無效。 • UNREACHABLE – 收件人裝置目前無法連線或無法使用。例如，裝置可能關機，或可能與網路中斷連線。您可以稍後再嘗試傳送訊息。 • UNKNOWN – 發生錯誤，以致無法交付訊息。這通常是暫時性錯誤，稍後可以再嘗試傳送訊息。 • BLOCKED – 收件人裝置封鎖了起始號碼的 SMS 訊息。

屬性	描述
	• CARRIER_UNREACHABLE – 收件人的行動網路發生問題，以致無法交付訊息。這通常是暫時性錯誤，稍後可以再嘗試傳送訊息。 • SPAM – 收件人的行動電信業者判定訊息內容是垃圾郵件，並將訊息封鎖，不予交付。 • INVALID_MESSAGE – SMS 訊息內文無效且無法傳送。 • CARRIER_BLOCKED – 收件人的電信業者已封鎖訊息，不予交付。這通常發生在電信業者將訊息內容識別為未經要求或惡意時。 • TTL_EXPIRED – 無法在特定時間內傳送 SMS 訊息。這通常是暫時性錯誤，稍後可以再嘗試傳送訊息。 • MAX_PRICE_EXCEEDED – 傳送訊息可能導致費用超過帳戶的每月 SMS 支出配額。完成 Amazon Pinpoint 使用者指南中請求調高每月 SMS 支出配額的程序，可請求提高此配額。 • OPTED_OUT – 未傳送 SMS 訊息，因為收件人選擇不接收您的訊息。 • NO_QUOTA_LEFT_ON_ACCOUNT – 您的帳戶沒有足夠的支出配額可以傳送訊息。您可以透過完成《 AWS 最終使用者簡訊簡訊使用者指南》中的請求增加每月簡訊花費配額中的程序，來請求提高此配額。 • NO_ORIGINATION_IDENTITY_AVAILABLE_TO_SEND – 您的帳戶不包含可將訊息傳送到目的地的電話號碼。 • DESTINATION_COUNTRY_NOT_SUPPORTED – 已封鎖目的地國家/地區。對於所有支援的國家/地區，請參閱《 AWS 最終使用者傳訊簡訊使用者指南》中的支援的國家/地區和區域 (SMS 管道)。

屬性	描述
	• ACCOUNT_IN_SANDBOX – 您的帳戶位於沙盒中，只能傳送到通過驗證的目的地號碼。您可以在 Amazon Pinpoint 主控台中驗證目的地號碼，或開始將帳戶移出沙盒的程序，請參閱AWS 《最終使用者傳訊簡訊使用者指南》中的關於 SMS/MMS 和語音沙盒。 • RATE_EXCEEDED – 您嘗試傳送訊息的速度太快而被限流。您需要放慢呼叫速度。如需限制的詳細資訊，請參閱《 AWS 最終使用者傳訊簡訊使用者指南》中的每秒訊息部分 (MPS) 限制。 • INVALID_ORIGINATION_IDENTITY – 提供的起始身分無效。 • ORIGINATION_IDENTITY_DOES_NOT_EXIST – 提供的起始身分不存在。 • INVALID_DLT_PARAMETERS – 提供了無效的 DLT 參數 (印度目的地必填)。 • INVALID_PARAMETERS – 提供了無效的參數。 • ACCESS_DENIED – 您的帳戶已被封鎖，不能傳送訊息。請聯絡客戶支援找出原因並解決問題。 • INVALID_KEYWORD – 提供的關鍵字無效。關鍵字可能格式不正確，或未在您的帳戶中設定。 • INVALID_SENDER_ID – 提供的寄件者 ID 無效。寄件者 ID 的格式或長度可能不正確。 • INVALID_POOL_ID – 提供的集池 ID 無效。集池 ID 可能格式不正確，或不屬於您的帳戶。 • SENDER_ID_NOT_SUPPORTED_FOR_DESTINATION – 目的地國家/地區不支援寄

屬性	描述
	件者 ID。您必須使用電話號碼或其他起始身分傳送。 • INVALID_PHONE_NUMBER – 提供的起始電話號碼無效。電話號碼的格式或長度可能不正確。
iso_country_code	與收件人電話號碼相關聯的國家，以 ISO 3166-1 alpha-2 格式顯示。
treatment_id	訊息處理方式的 ID (如果已使用 A/B 行銷活動傳送訊息)。
treatment_id	如果訊息曾使用 A/B 測試行銷活動傳送，則此值代表訊息的處理方式號碼。若為交易簡訊，此值為 0。
number_of_message_parts	Amazon Pinpoint 為傳送訊息而建立的訊息部分數。 一般而言，簡訊只能包含 160 個 GSM-7 字元或 67 個非 GSM 字元，但這些限制會因國家而異。如果您傳送的訊息超過這些限制，Amazon Pinpoint 會自動將訊息分割成較小的部分。收費依傳送的訊息片段數量為準。
message_id	Amazon Pinpoint 接受訊息時，產生的唯一 ID。
message_type	訊息的類型。可能的值為 Promotional (促銷) 和 Transactional (交易)。您可以在建立行銷活動時，或使用 Amazon Pinpoint API 中的 SendMessages 操作時指定此值。
campaign_id	傳送訊息的 Amazon Pinpoint 行銷活動的唯一 ID。
customer_context	來自 Amazon Pinpoint SendMessages 操作中傳送之Context映射內容的 JSON 字串。

用戶端

包含回報事件之裝置上安裝的應用程式用戶端資訊。

屬性	描述
client_id	若為應用程式產生的事件，則此值會是安裝在裝置上的應用程式用戶端唯一 ID。此 ID 由 AWS Mobile SDK for iOS 和 自動產生 適用於 Android 的 AWS Mobile SDK。 若為傳送行銷活動和交易訊息時產生的事件，此值等於您訊息傳送目標的端點 ID。
cognito_id	這個唯一 ID 已指派給您應用程式使用的 Amazon Cognito 身分池中的應用程式用戶端。

裝置

包含回報事件的裝置資訊。

屬性	描述
locale	裝置地區設定。
make	裝置品牌，例如 Apple 或 Samsung。
model	裝置型號，例如 iPhone。
platform	裝置平台，例如 ios 或 android。

指標

包含與事件相關聯的指標資訊。

屬性	描述
price_in_millicents_usd	我們向您收取的傳訊費金額。此價格以千分之多少美國分顯示。假設此屬性的值是 645，則我們收取的傳訊費用是 0.645¢ (645/1000 = 0.645¢ = $0.00645)。 **Note** 此屬性不會針對 event_type 為 _SMS.BUFFERED 的訊息顯示。

從 Amazon Pinpoint 刪除事件串流

如果您將 Kinesis 串流指派給某個應用程式，您可以停用該應用程式的事件串流。Amazon Pinpoint 會停止將事件串流到 Kinesis，但您使用 Amazon Pinpoint 主控台可以檢視事件分析。

AWS CLI

使用 delete-event-stream 命令：

```
aws pinpoint delete-event-stream --application-id application-id
```

適用於 Java 的 AWS SDK

使用 Amazon Pinpoint 用戶端的 deleteEventStream方法：

```
pinClient.deleteEventStream(new DeleteEventStreamRequest().withApplicationId(appId));
```

查詢 Amazon Pinpoint 分析資料

除了使用 Amazon Pinpoint 主控台的分析頁面，也可以使用 Amazon Pinpoint Analytics API 查詢標準指標子集的分析資料，深入了解與使用者互動、行銷活動推廣等相關的趨勢。這些指標也稱為「關鍵績效指標 (KPI)」，是可測量的值，可協助您監控與評估專案、行銷活動和行程的績效。

若使用 API 查詢分析資料，可以使用自選的報告工具分析資料，不須登入 Amazon Pinpoint 主控台或分析 Amazon Kinesis 串流等來源的原始事件資料。例如，您可以建置自訂儀表板，顯示每週的行銷活動結果，或是提供您行銷活動交付率的深入分析。

您可以使用 Amazon Pinpoint REST API、 AWS Command Line Interface (AWS CLI) 或 SDK 來 AWS 查詢資料。若要查詢資料，可以將請求傳送到 Amazon Pinpoint API，並使用支援的參數，指定所需的資料，以及任何您要套用的篩選條件。您提交查詢後，Amazon Pinpoint 會以 JSON 回應傳回查詢結果。然後，您可以將結果傳送到另一個服務或應用程式進行更深入的分析、儲存或報告。

Amazon Pinpoint 會針對您所有專案、行銷活動和旅程的所有支援指標，自動收集並彙整相關資料。此外，此資料會持續更新，產生上限約兩個小時的資料延遲時間範圍。但請注意，某些指標可能會有額外的資料延遲。這是因為某些指標的資料是以我們從收件者電子郵件提供者收到的資訊為基礎。有些提供者會立即傳送這些資訊給我們，有些提供者的傳送次數則較不頻繁。

Amazon Pinpoint 會存放資料 90 天。若要存放資料超過 90 天或即時存取原始分析資料，您可以設定 Amazon Pinpoint 專案，將事件資料串流至 Amazon Kinesis Data Streams 或 Amazon Data Firehose。如需設定事件串流的相關資訊，請參閱使用 Amazon Pinpoint 透過 Kinesis 和 Firehose 串流應用程式事件資料。

查詢 Amazon Pinpoint 中指標的元件和參數

若要查詢指標的資料，請將 `get` 請求傳送至適當的 Amazon Pinpoint API 指標資源。在您的請求中，您可以針對下列查詢元件，使用支援的參數來定義您的查詢：

- 專案 – 提供專案 ID 當作 `application-id` 參數值來指定專案。此參數是所有指標的必要參數。
- 行銷活動 – 提供行銷活動 ID 當作 `campaign-id` 參數值來指定行銷活動。只有行銷活動指標才需要此參數。
- 旅程 – 提供旅程 ID 當作 `journey-id` 參數值來指定旅程。僅行程參與和執行指標以及行程活動執行指標需要此參數。
- 旅程活動 – 提供旅程活動 ID 當作 `journey-activity-id` 參數值，藉此指定旅程活動。僅行程活動執行指標需要此參數。

- 日期範圍 – (選用) 若要依日期範圍篩選資料，請使用支援的開始和結束時間參數，提供日期範圍的第一個和最後一個日期和時間。這些值應依延伸的 ISO 8601 格式顯示，並使用國際標準時間 (UTC)，例如 2019 年 7 月 19 日 8:00 PM UTC 即以 `2019-07-19T20:00:00Z` 顯示。

  日期範圍包含在內，且必須不超過 31 個日曆日。此外，第一個日期和時間必須小於從當天起算 90 天。若沒有指定日期範圍，Amazon Pinpoint 會傳回前 31 個日曆天的資料。除行程執行指標和行程活動執行指標外，所有指標都支援日期範圍參數。
- 指標 – 提供指標名稱當作 `kpi-name` 參數值可指定指標。此值會說明相關聯的指標，並由兩個以上的術語組成；這些術語都是由小寫的英數字元組成，並以連字號分隔。範例包括 `email-open-rate` 和 `successful-delivery-rate`。除行程執行指標和行程活動執行指標外，所有指標都需要此參數。如需支援指標的完整清單及要針對每個項目使用的 `kpi-name` 值，請參閱專案、行銷活動和旅程的標準指標。

您傳送查詢後，Amazon Pinpoint 會以 JSON 回應傳回查詢結果。在回應中，結果的結構會因您查詢的指標而不同。

有些指標只提供一個值，例如行銷活動已交付的訊息數。其他指標提供多個值，而且通常會依相關欄位將這些值分組，例如每個行銷活動執行時交付的訊息數 (依行銷活動分組)。若指標提供並分組多個值，則 JSON 回應即會包含一個欄位，指出分組資料所用的欄位。若要進一步了解查詢結果的結構，請參閱 使用 JSON 查詢結果。

查詢 Amazon Pinpoint 分析資料的 IAM 政策

使用 Amazon Pinpoint API 可以查詢標準指標子集的分析資料，亦稱為關鍵績效指標 (KPI)，適用於 Amazon Pinpoint 專案、行銷活動和旅程。這些指標可協助您監控與評估專案、行銷活動和行程的績效。

若要管理對此資料的存取，您可以建立 AWS Identity and Access Management (IAM) 政策，為有權存取資料的 IAM 角色或使用者定義許可。為了支援此資料存取權限的細微控制，Amazon Pinpoint 提供了多種讓您在 IAM 政策中指定的操作。有項操作是專門在 Amazon Pinpoint 主控台 (`mobiletargeting:GetReports`) 檢視分析資料，還有其他使用 Amazon Pinpoint API，以程式化方式存取分析資料的操作。

若要建立管理分析資料存取權的 IAM 政策，您可以使用 AWS 管理主控台 AWS CLI、 或 IAM API。請注意， AWS 管理主控台 的視覺化編輯器索引標籤目前未包含檢視或查詢 Amazon Pinpoint 分析資料的操作。但是，您可以使用主控台上的 JSON 標籤，手動將必要的動作新增到 IAM 政策。

例如，下列政策允許以程式設計方式存取所有 AWS 區域中所有專案、行銷活動和旅程的所有分析資料：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "QueryAllAnalytics",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:GetApplicationDateRangeKpi",
                "mobiletargeting:GetCampaignDateRangeKpi",
                "mobiletargeting:GetJourneyDateRangeKpi",
                "mobiletargeting:GetJourneyExecutionMetrics",
                "mobiletargeting:GetJourneyExecutionActivityMetrics"
            ],
            "Resource": [
                "arn:aws:mobiletargeting:*:111122223333:apps/*/kpis/*",
                "arn:aws:mobiletargeting:*:111122223333:apps/*/campaigns/*/kpis/
*",
                "arn:aws:mobiletargeting:*:111122223333:apps/*/journeys/*/kpis/
*",
                "arn:aws:mobiletargeting:*:111122223333:apps/*/journeys/*/
execution-metrics",
                "arn:aws:mobiletargeting:*:111122223333:apps/*/journeys/*/
activities/*/execution-metrics"
            ]
        }
    ]
}
```

其中 *accountId* 是 AWS 您的帳戶 ID。

但是，根據最佳實務，您應該定義遵循「最低權限」原則的政策。換句話說，您應建立其中只包含執行特定任務所需許可的政策。若要支援此實務並實作更精細的控制，您可以限制對特定 AWS 區域中特定專案分析資料的程式設計存取，例如：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "QueryProjectAnalytics",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:GetApplicationDateRangeKpi",
                "mobiletargeting:GetCampaignDateRangeKpi",
                "mobiletargeting:GetJourneyDateRangeKpi",
                "mobiletargeting:GetJourneyExecutionMetrics",
                "mobiletargeting:GetJourneyExecutionActivityMetrics"
            ],
            "Resource": [
            "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/kpis/
*",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/campaigns/*/
kpis/*",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/journeys/*/
kpis/*",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/journeys/*/
execution-metrics",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/journeys/*/
activities/*/execution-metrics"
            ]
        }
    ]
}
```

其中：

- *region* 是託管專案的 AWS 區域名稱。
- *accountId* 是 AWS 您的帳戶 ID。
- *projectId* 是您希望提供存取的專案識別符。

同樣地，以下範例政策只會允許以程式設計的方式存取特定行銷活動的分析資料：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "QueryCampaignAnalytics",
            "Effect": "Allow",
            "Action": "mobiletargeting:GetCampaignDateRangeKpi",
            "Resource": "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/projectId/campaigns/campaignId/kpis/*"
        }
    ]
}
```

其中：

- *region* 是託管專案的 AWS 區域名稱。
- *accountId* 是您的 AWS 帳戶 ID。
- *projectId* 是與行銷活動相關聯的專案識別符。
- *campaignId* 是您希望提供存取的行銷活動識別符。

下列範例政策可讓您以程式設計方式存取特定行程及構成此行程之活動的所有分析資料，包括參與和執行資料：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "QueryJourneyAnalytics",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:GetJourneyDateRangeKpi",
                "mobiletargeting:GetJourneyExecutionMetrics",
                "mobiletargeting:GetJourneyExecutionActivityMetrics"
            ],
```

```
            "Resource": [
                "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/
journeys/journeyId/kpis/*",
                "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/
journeys/journeyId/execution-metrics",
                "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/
journeys/journeyId/activities/*/execution-metrics"
            ]
        }
    ]
}
```

其中：

- *region* 是託管專案的 AWS 區域名稱。
- *accountId* 是 AWS 您的帳戶 ID。
- *projectId* 是與行程相關聯的專案識別符。
- *journeyId* 是您希望可供存取的行程識別符。

如需可在 IAM 政策中使用的 Amazon Pinpoint API 操作的完整列表，請參閱 IAM 政策的 Amazon Pinpoint 動作。如需建立和管理 IAM 政策的詳細資訊，請參閱 IAM 使用者指南。

適用於 Amazon Pinpoint 專案、行銷活動和旅程的標準指標

您可以使用 Amazon Pinpoint Analytics API 查詢分析資料，以取得適用於 Amazon Pinpoint 專案、行銷活動和旅程的標準指標子集。這些指標也稱為「關鍵績效指標 (KPI)」，是可測量的值，可協助您監控與評估專案、行銷活動和行程的績效。

Amazon Pinpoint 針對多種標準指標，提供分析資料的程式設計存取權限：

- 應用程式指標 – 這些指標讓您深入了解與專案相關聯的所有行銷活動和交易訊息的趨勢，亦稱為應用程式。例如，您可以使用應用程式指標來取得由與專案相關聯每個行銷活動的收件人所開啟的訊息數量明細。
- 行銷活動指標 – 這些指標讓您深入了解各行銷活動的表現。例如，您可以使用行銷活動指標來判斷行銷訊息傳送到了多少端點，或是那些訊息中有多少交付到了端點。
- 旅程參與指標 – 這些指標讓您深入了解各旅程的表現。例如，您可以使用行程參與指標取得參與者在行程每個活動中開啟的郵件數目明細。

- 旅程執行指標 – 這些指標讓您深入了解各旅程的參與趨勢。例如，您可以使用行程執行指標判斷開始行程的參與者人數。
- 旅程活動執行指標 – 這些指標讓您深入了解旅程中各項活動的參與趨勢。例如，您可以使用行程活動執行指標判斷開始活動的參與者人數，以及完成活動中每條路線的參與者人數。

本節中的主題會列出和說明您可以針對每種指標類型查詢的個別指標。

主題

- 行銷活動的 Amazon Pinpoint 應用程式指標
- 交易電子郵件訊息的 Amazon Pinpoint 應用程式指標
- 交易簡訊的 Amazon Pinpoint 應用程式指標
- Amazon Pinpoint 行銷活動指標
- Amazon Pinpoint 旅程參與指標
- Amazon Pinpoint 旅程執行指標
- Amazon Pinpoint 旅程活動執行指標
- Amazon Pinpoint 旅程和行銷活動執行指標

行銷活動的 Amazon Pinpoint 應用程式指標

下表列出並描述了標準應用程式指標，您可以針對與 Amazon Pinpoint 專案相關聯的所有行銷活動，查詢這些指標並評估表現。若要查詢這些指標的資料，請使用 Amazon Pinpoint API 的應用程式指標資源。表格中的 kpi-name 欄會指出在查詢中針對 `kpi-name` 參數要使用的值。

指標	kpi-name	說明
交付率	`successful-delivery-rate`	對於與專案相關聯的所有行銷活動，交付給收件人的訊息百分比。 此指標的計算方式為專案的所有行銷活動傳送並交付給收件人的訊息數量，除以所有行銷活動傳送的訊息數量。

指標	kpi-name	說明
Delivery rate, grouped by date (交付率，依日期分組)	`successful-delivery-rate-grouped-by-date`	對於與專案相關聯的所有行銷活動，在指定的日期範圍中的每天交付給收件人的訊息百分比。 此指標的計算方式為針對指定日期範圍中的每天，專案的所有行銷活動傳送並交付給收件人的訊息數量，除以所有行銷活動傳送的訊息數量。 此指標的查詢結果會依日曆天分組，採用延伸 ISO 8601 格式。
電子郵件開啟率	`email-open-rate`	對於與專案相關聯的所有行銷活動，由收件人開啟的電子郵件訊息百分比。 此指標的計算方式為專案的所有行銷活動傳送並由收件人開啟的電子郵件訊息數量，除以所有行銷活動傳送並交付給收件人的電子郵件訊息數量。

指標	kpi-name	說明
電子郵件開啟率，依行銷活動分組	email-open-rate-grouped-by-campaign	對於與專案相關聯的每個行銷活動，由收件人開啟的電子郵件訊息百分比。 此指標的計算方式為行銷活動傳送並由收件人開啟的電子郵件訊息數量，除以行銷活動傳送並交付給收件人的電子郵件訊息數量。 此指標的查詢結果會依行銷活動 ID (CampaignId) 分組，這是可唯一識別行銷活動的字串。
端點交付	unique-deliveries	針對與專案相關聯的所有行銷活動，交付訊息的唯一端點數量。
端點交付，依行銷活動分組	unique-deliveries-grouped-by-campaign	針對與專案相關聯的每個行銷活動，交付訊息的唯一端點數量。 此指標的查詢結果會依行銷活動 ID (CampaignId) 分組，這是可唯一識別行銷活動的字串。
端點交付，依日期分組	unique-deliveries-grouped-by-date	針對與專案相關聯的所有行銷活動，在指定日期範圍內，每天交付訊息的唯一端點數量。 此指標的查詢結果會依日曆天分組，採用延伸 ISO 8601 格式。

指標	kpi-name	說明
訊息交付，依行銷活動分組	`successful-deliveries-grouped-by-campaign`	對於與專案相關聯的每個行銷活動，交付給收件人的訊息數量。 此指標的計算方式為行銷活動傳送的訊息數量，減去行銷活動執行傳送且由於硬退信而無法傳送到收件人的訊息數量。 此指標的查詢結果會依行銷活動 ID (`CampaignId`) 分組，這是可唯一識別行銷活動的字串。
推送開啟率	`push-open-rate`	對於與專案相關聯的所有行銷活動，由收件人開啟的推送通知百分比。 此指標的計算方式為：由專案所有行銷活動傳送並經收件人開啟的推送通知數量，除以所有行銷活動傳送並交付予收件人的推送通知數量。

指標	kpi-name	說明
推送開啟率，依行銷活動分組	`push-open-rate-grouped-by-campaign`	對於與專案相關聯的每個行銷活動，由收件人開啟的推送通知百分比。 此指標的計算方式為：由行銷活動傳送並經收件人開啟的推送通知數量，除以行銷活動傳送並交付予收件人的推送通知數量。 此指標的查詢結果會依行銷活動 ID (`CampaignId`) 分組，這是可唯一識別行銷活動的字串。

交易電子郵件訊息的 Amazon Pinpoint 應用程式指標

下表列出並描述了標準應用程式指標，您可以針對與 Amazon Pinpoint 專案相關聯的所有交易電子郵件，查詢這些指標並監控趨勢。若要查詢這些指標的資料，請使用 Amazon Pinpoint API 的應用程式指標資源。表格中的 kpi-name 欄會指出在查詢中針對 `kpi-name` 參數要使用的值。

請注意，這些指標不會提供行銷活動所傳送的電子郵件訊息相關資料。它們僅提供交易電子郵件訊息的相關資料。若要查詢一或多個行銷活動傳送的訊息資料，請使用行銷活動指標或行銷活動的應用程式指標。

指標	kpi-name	說明
點按數	`txn-emails-clicked`	收件人點按訊息中連結的次數。如果單一收件人點按訊息中的多個連結，或點按相同連結多次，則會在計數中包含每個點按。
Clicks, grouped by date (點按數，依日期分組)	`txn-emails-clicked-grouped-by-date`	收件人於指定日期範圍內每天點選訊息中連結的次數。如果

指標	kpi-name	說明
		單一收件人點按訊息中的多個連結，或點按相同連結多次，則會在計數中包含每個點按。 此指標的查詢結果會依日曆天分組，採用延伸 ISO 8601 格式。
投訴率	`txn-emails-complaint-rate`	收件人回報為未經要求或不想收到的電子郵件的訊息百分比。 此指標的計算方式為：收件人回報為未經要求或不想收到的電子郵件的訊息數量，除以傳送的訊息數量。
Complaint rate, grouped by date (投訴率，依日期分組)	`txn-emails-complaint-rate-grouped-by-date`	在指定日期範圍內的每一天，收件人回報為未經要求或不想收到的電子郵件的訊息百分比。 此指標的計算方式為：在指定日期範圍內的每一天，收件人回報為未經要求或不想收到的電子郵件的訊息數量，除以傳送的訊息數量。 此指標的查詢結果會依日曆天分組，採用延伸 ISO 8601 格式。
投訴	`txn-emails-with-complaints`	收件人回報為未經要求或不想收到的電子郵件的訊息數量。

指標	kpi-name	說明
Complaints, grouped by date (投訴數，依日期分組)	`txn-emails-with-complaints-grouped-by-date`	在指定日期範圍內的每一天，收件人回報為未經要求或不想收到的電子郵件的訊息數量。 此指標的查詢結果會依日曆天分組，採用延伸 ISO 8601 格式。
交付	`txn-emails-delivered`	交付給收件人的訊息數量 此指標的計算方式為：傳送的訊息數量，減去由於軟退信或硬退信或因拒絕而無法交付的訊息數量。如果 Amazon Pinpoint 判定訊息含惡意程式碼，將拒絕訊息。Amazon Pinpoint 不會嘗試傳送被拒絕的訊息。
Deliveries, grouped by date (交付數，依日期分組)	`txn-emails-delivered-grouped-by-date`	在指定日期範圍內每天交付給收件人的訊息數量。 此指標的計算方式為：在指定日期範圍內的每一天，傳送的訊息數量，減去由於軟退信或硬退信或因拒絕而無法交付的訊息數量。如果 Amazon Pinpoint 判定訊息含惡意程式碼，將拒絕訊息。Amazon Pinpoint 不會嘗試傳送被拒絕的訊息。 此指標的查詢結果會依日曆天分組，採用延伸 ISO 8601 格式。

指標	kpi-name	說明
交付率	`txn-emails-delivery-rate`	交付給收件人的訊息百分比。 此指標的計算方式為：傳送並交付給收件人的訊息數量，除以傳送的訊息數量。
Delivery rate, grouped by date (交付率，依日期分組)	`txn-emails-delivery-rate-grouped-by-date`	在指定日期範圍內每天交付給收件人的訊息百分比。 此指標的計算方式為：在指定日期範圍內的每一天，傳送並交付給收件人的訊息數量，除以傳送的訊息數量。 此指標的查詢結果會依日曆天分組，採用延伸 ISO 8601 格式。
硬退信	`txn-emails-hard-bounced`	由於硬退信而無法交付給收件人的訊息數量。如果存在永久性問題導致訊息無法交付 (例如收件人的電子郵件地址不存在)，將會硬退信。
Hard bounces, grouped by date (硬退信，依日期分組)	`txn-emails-hard-bounced-grouped-by-date`	在指定日期範圍內的每一天，由於硬退信而無法交付給收件人的訊息數量。如果存在永久性問題導致訊息無法交付 (例如收件人的電子郵件地址不存在)，將會硬退信。 此指標的查詢結果會依日曆天分組，採用延伸 ISO 8601 格式。
開啟數	`txn-emails-opened`	收件人開啟的訊息數量。

指標	kpi-name	說明
Opens, grouped by date (開啟數，依日期分組)	`txn-emails-opened-grouped-by-date`	收件人於指定日期範圍內每天開啟的訊息數量。 此指標的查詢結果會依日曆天分組，採用延伸 ISO 8601 格式。
傳送	`txn-emails-sent`	傳送的訊息數量。
Sends, grouped by date (傳送數，依日期分組)	`txn-emails-sent-grouped-by-date`	在指定日期範圍內每天傳送的訊息數量。 此指標的查詢結果會依日曆天分組，採用延伸 ISO 8601 格式。
Soft bounces (軟退信)	`txn-emails-soft-bounced`	由於軟退信而無法交付給收件人的訊息數量。如果存在暫時性問題導致訊息無法交付 (例如收件人的收件匣已滿，或接收伺服器暫時無法使用)，則會軟退信。
Soft bounces, grouped by date (軟退信，依日期分組)	`txn-emails-soft-bounced-grouped-by-date`	在指定日期範圍內的每一天，由於軟退信而無法交付給收件人的訊息數量。如果存在暫時性問題導致訊息無法交付 (例如收件人的收件匣已滿，或接收伺服器暫時無法使用)，則會軟退信。 此指標的查詢結果會依日曆天分組，採用延伸 ISO 8601 格式。

指標	kpi-name	說明
Unique user click events (唯一使用者點按事件數)	`txn-emails-unique-clicks`	點按訊息中連結的唯一收件人 (端點) 數量。 與 Clicks (點按數) 指標不同，此指標回報的是點選連結的唯一收件人數量，而非發生的點按事件數量。例如，如果單一收件人點按相同訊息中的多個連結，或多次點按相同的連結，則此指標只會針對該收件人回報一個點按事件。
Unique user click events, grouped by date (唯一使用者點按事件數，依日期分組)	`txn-emails-unique-clicks-grouped-by-date`	在指定日期範圍內每天點按訊息中連結的唯一收件人 (端點) 數量。 與 Clicks, grouped by date (點按數，依日期分組) 不同，此指標回報的是點按連結的唯一收件人數量，而非發生的點按事件數量。例如，如果單一收件人點按相同訊息中的多個連結，或多次點按相同的連結，則此指標只會針對該收件人回報一個點按事件。 此指標的查詢結果會依日曆天分組，採用延伸 ISO 8601 格式。

指標	kpi-name	說明
Unique user open events (唯一使用者開啟事件數)	`txn-emails-unique-opens`	開啟訊息的唯一收件人 (端點) 數量。 與 Opens (開啟數) 指標不同，此指標回報的是開啟訊息的唯一收件人數量，而非發生的開啟事件數量。例如，如果單一收件人多次開啟相同訊息，則此指標只會針對該收件人回報一個開啟事件。
Unique user open events, grouped by date (唯一使用者開啟事件數，依日期分組)	`txn-emails-unique-opens-grouped-by-date`	在指定日期範圍內每天開啟訊息的唯一收件人 (端點) 數量。 與 Opens, grouped by date (開啟數，依日期分組) 指標不同，此指標回報的是開啟訊息的唯一收件人數量。而非發生的開啟事件數量。例如，如果單一收件人多次開啟相同訊息，則此指標只會針對該收件人回報一個開啟事件。 此指標的查詢結果會依日曆天分組，採用延伸 ISO 8601 格式。

交易簡訊的 Amazon Pinpoint 應用程式指標

下表列出並描述了標準應用程式指標，您可以針對與 Amazon Pinpoint 專案相關聯的所有交易 SMS 訊息，查詢這些指標並監控趨勢。若要查詢這些指標的資料，請使用 Amazon Pinpoint API 的應用程式指標資源。表格中的 kpi-name 欄會指出在查詢中針對 `kpi-name` 參數要使用的值。

請注意，這些指標不會提供行銷活動所傳送之簡訊的相關資料。它們僅提供交易簡訊的相關資料。若要查詢一或多個行銷活動傳送的訊息資料，請使用行銷活動指標或行銷活動的應用程式指標。

指標	kpi-name	說明
每則訊息的平均價格 (依國家/地區分組)	`txn-sms-average-price-grouped-by-country`	傳訊到各國家/地區或區域的單則訊息平均成本。價格以美分的千分之幾顯示。假設此屬性的值是 645，則我們收取的傳訊費用是 0.645¢ (645/1000 = 0.645¢ = $0.00645)。 此指標的計算方式為傳送到每個國家或地區收件人所有訊息的總成本，除以每個國家和地區中傳送到收件人的訊息數。 此指標的查詢結果會依國家或區域分組，採用 ISO 3166-1 alpha-2 格式。
每則訊息部分的平均價格 (依國家/地區分組)	`txn-sms-average-price-by-parts-grouped-by-country`	傳訊到各國家/地區或區域的單則訊息部分平均成本。訊息部分是 SMS 訊息的一部分。價格以美分的千分之幾顯示。假設此屬性的值是 645，則我們收取的傳訊費用是 0.645¢ (645/1000 = 0.645¢ = $0.00645)。 此指標的計算方法是，傳訊給各國家/地區或區域的收件人所產生的所有訊息部分總成本，除以傳送的訊息部分數。 此指標的查詢結果會依國家或區域分組，採用 ISO 3166-1 alpha-2 格式。
交付	`txn-sms-delivered`	交付給收件人的訊息數量

指標	kpi-name	說明
Deliveries, grouped by country (交付數，依國家或區域分組)	`txn-sms-delivered-grouped-by-country`	針對訊息所傳送至的國家或區域，交付給收件人的訊息數量。 此指標的查詢結果會依國家或區域分組，採用 ISO 3166-1 alpha-2 格式。
Deliveries, grouped by date (交付數，依日期分組)	`txn-sms-delivered-grouped-by-date`	在指定日期範圍內每天交付給收件人的訊息數量。 此指標的查詢結果會依日曆天分組，採用延伸 ISO 8601 格式。
Delivery errors (交付錯誤數)	`txn-sms-error-distribution`	針對發生的每種錯誤類型，嘗試交付訊息時發生錯誤的次數。 針對發生的每種錯誤類型，此指標的查詢結果會依錯誤碼分組。
交付率	`txn-sms-delivery-rate`	交付給收件人的訊息百分比。 此指標的計算方式為：傳送並交付給收件人的訊息數量，除以傳送的訊息數量。

指標	kpi-name	說明
Delivery rate, grouped by date (交付率，依日期分組)	`txn-sms-delivery-rate-grouped-by-date`	在指定日期範圍內每天交付給收件人的訊息百分比。 此指標的計算方式為：在指定日期範圍內的每一天，傳送並交付給收件人的訊息數量，除以傳送的訊息數量。 此指標的查詢結果會依日曆天分組，採用延伸 ISO 8601 格式。
已交付的訊息部分	`txn-sms-delivered-by-parts`	已交付的訊息部分數。訊息部分是 SMS 訊息的一部分。如果 SMS 訊息包含的字元數超過 SMS 通訊協定允許的字元數，Amazon Pinpoint 會視需要將訊息分割成多則訊息，以將訊息傳送給收件人。
已交付的訊息部分，依國家/地區分組	`txn-sms-delivered-by-parts-grouped-by-country`	已交付給各國家/地區或區域的訊息部分數。訊息部分是 SMS 訊息的一部分。 此指標的查詢結果會依國家或區域分組，採用 ISO 3166-1 alpha-2 格式。
已傳送的訊息部分	`txn-sms-sent-by-parts`	已傳送的訊息部分的數目。訊息部分是 SMS 訊息的一部分。如果 SMS 訊息包含的字元數超過 SMS 通訊協定允許的字元數，Amazon Pinpoint 會視需要將訊息分割成多則訊息，以將訊息傳送給收件人。

指標	kpi-name	說明
已傳送的訊息部分，依國家/地區分組	txn-sms-sent-by-parts-grouped-by-country	已傳送到各國家/地區或區域的訊息部分數。訊息部分是 SMS 訊息的一部分。 此指標的查詢結果會依國家或區域分組，採用 ISO 3166-1 alpha-2 格式。
已傳送的訊息	txn-sms-sent	傳送的訊息數量。
已傳送的訊息，依國家/地區分組	txn-sms-sent-grouped-by-country	針對訊息所傳送至的國家或區域，傳送的訊息數量。 此指標的查詢結果會依國家或區域分組，採用 ISO 3166-1 alpha-2 格式。
已傳送的訊息，依日期分組	txn-sms-sent-grouped-by-date	在指定日期範圍內每天傳送的訊息數量。 此指標的查詢結果會依日曆天分組，採用延伸 ISO 8601 格式。
Total price, grouped by country (總價，依國家或區域分組)	txn-sms-total-price-grouped-by-country	傳訊到各國家/地區或區域的總成本。價格以美分的千分之幾顯示。假設此屬性的值是 645，則我們收取的傳訊費用是 0.645¢ (645/1000 = 0.645¢ = $0.00645)。 此指標的查詢結果會依國家或區域分組，採用 ISO 3166-1 alpha-2 格式。

Amazon Pinpoint 行銷活動指標

下表列出及描述您可以查詢的標準行銷活動指標，以評估個別行銷活動的績效。若要查詢這些指標的資料，請使用 Amazon Pinpoint API 的行銷活動指標資源。表格中的 kpi-name 欄指出在查詢中針對 `kpi-name` 參數要使用的值。

指標	kpi-name	說明
退信率	`hard-bounce-rate`	對於所有行銷活動執行，無法交付其收件人的電子郵件訊息百分比。此指標只測量硬退信，也就是收件人的電子郵件地址存在永久性問題，導致無法交付訊息的情況。 此指標的計算方式為專案的所有行銷活動執行傳送的退信的電子郵件訊息數量，除以所有行銷活動執行傳送的電子郵件訊息數量。
退信率，依行銷活動執行分組	`hard-bounce-rate-grouped-by-campaign-activity`	對於每個行銷活動執行，無法交付給收件人的電子郵件訊息百分比。此指標只測量硬退信，也就是收件人的電子郵件地址存在永久性問題，導致無法交付訊息的情況。 此指標的計算方式為專案的行銷活動執行傳送的退信的電子郵件訊息數量，除以行銷活動執行傳送的電子郵件訊息數量。 此指標的查詢結果會依行銷活動的活動 ID (`CampaignActivityId`) 分組，這是

指標	kpi-name	說明
		可唯一識別行銷活動執行的字串。
交付率	`successful-delivery-rate`	對於所有行銷活動執行，交付給收件人的訊息百分比。 此指標的計算方式為所有行銷活動執行傳送並交付給收件人的訊息數量，除以所有行銷活動執行傳送的訊息數量。
交付率，依行銷活動執行分組	`successful-delivery-rate-grouped-by-campaign-activity`	對於每個行銷活動執行，交付給收件人的訊息百分比。 此指標的計算方式為行銷活動執行傳送並交付給收件人的訊息數量，除以行銷活動執行傳送的訊息數量。 此指標的查詢結果會依行銷活動的活動 ID (`CampaignActivityId`) 分組，這是可唯一識別行銷活動執行的字串。

指標	kpi-name	說明
Delivery rate, grouped by date (交付率，依日期分組)	`successful-delivery-rate-grouped-by-date`	對於所有行銷活動執行，在指定的日期範圍中的每天交付給收件人的訊息百分比。 此指標的計算方式為針對指定日期範圍中的每天，所有行銷活動執行傳送並交付給收件人的訊息數量，除以所有行銷活動執行傳送的訊息數量。 此指標的查詢結果會依日曆天分組，採用延伸 ISO 8601 格式。
電子郵件開啟率	`email-open-rate`	對於所有行銷活動執行，由收件人開啟的電子郵件訊息百分比。 此指標的計算方式為所有行銷活動執行傳送並由收件人開啟的電子郵件訊息數量，除以所有行銷活動執行傳送並交付給收件人的電子郵件訊息數量。

指標	kpi-name	說明
電子郵件開啟率，依行銷活動執行分組	email-open-rate-grouped-by-campaign-activity	對於每個行銷活動執行，由收件人開啟的電子郵件訊息百分比。 此指標的計算方式為行銷活動執行傳送並由收件人開啟的電子郵件訊息數量，除以行銷活動執行傳送並交付給收件人的電子郵件訊息數量。 此指標的查詢結果會依行銷活動的活動 ID (CampaignActivityId) 分組，這是可唯一識別行銷活動執行的字串。
開啟的電子郵件，依行銷活動執行分組	direct-email-opens-grouped-by-campaign-activity	對於每個行銷活動執行，由收件人開啟的電子郵件訊息數量。 此指標的查詢結果會依行銷活動的活動 ID (CampaignActivityId) 分組，這是可唯一識別行銷活動執行的字串。
端點交付	unique-deliveries	針對所有的行銷活動回合，交付訊息的唯一端點數量。

指標	kpi-name	說明
端點交付，依行銷活動執行分組	`unique-deliveries-grouped-by-campaign-activity`	針對每回合的行銷活動，交付訊息的唯一端點數量。 此指標的查詢結果會依行銷活動的活動 ID (`CampaignActivityId`) 分組，這是可唯一識別行銷活動執行的字串。
端點交付，依日期分組	`unique-deliveries-grouped-by-date`	針對所有的行銷活動回合，在指定日期範圍內，每天交付訊息的唯一端點數量。 此指標的查詢結果會依日曆天分組，採用延伸 ISO 8601 格式。
點按連結，依行銷活動執行分組	`clicks-grouped-by-campaign-activity`	對於所有行銷活動執行，收件人點按電子郵件訊息中連結的次數。如果單一收件人點按訊息中的多個連結，或點按相同連結多次，則會在計數中包含每個點按。 此指標的查詢結果會依行銷活動的活動 ID (`CampaignActivityId`) 分組，這是可唯一識別行銷活動執行的字串。

指標	kpi-name	說明
訊息交付，依行銷活動執行分組	`successful-deliveries-grouped-by-campaign-activity`	對於每個行銷活動執行，交付給收件人的訊息數量。 此指標的計算方式為行銷活動執行傳送的訊息數量，減去由於硬退信而無法傳送到執行的收件人的訊息數量。 此指標的查詢結果會依行銷活動的活動 ID (`CampaignActivityId`) 分組，這是可唯一識別行銷活動執行的字串。
訊息傳送，依行銷活動執行分組	`attempted-deliveries-grouped-by-campaign-activity`	對於每個行銷活動執行，傳送的訊息數量。 此指標的查詢結果會依行銷活動的活動 ID (`CampaignActivityId`) 分組，這是可唯一識別行銷活動執行的字串。
推送開啟率	`push-open-rate`	對於所有行銷活動執行，由收件人開啟的推送通知百分比。 此指標的計算方式為：由所有行銷活動回合傳送並經收件人開啟的推送通知數量，除以所有行銷活動回合傳送並交付予收件人的推送通知數量。

指標	kpi-name	說明
推送開啟率，依行銷活動執行分組	`push-open-rate-grouped-by-campaign-activity`	對於每個行銷活動執行，由收件人開啟的推送通知百分比。 此指標的計算方式為：由行銷活動回合傳送並經收件人開啟的推送通知數量，除以行銷活動回合傳送並交付予收件人的推送通知數量。 此指標的查詢結果會依行銷活動的活動 ID (`CampaignActivityId`) 分組，這是可唯一識別行銷活動執行的字串。
開啟的推送總數，依行銷活動執行分組	`direct-push-opens-grouped-by-campaign-activity`	對於每個行銷活動執行，由收件人開啟的推送通知數量。 此指標的查詢結果會依行銷活動的活動 ID (`CampaignActivityId`) 分組，這是可唯一識別行銷活動執行的字串。
Total SMS spend (簡訊費用總計)	`sms-spend`	所有行銷活動傳送 SMS 所支出的總金額 (以 milicents 為單位)。

Amazon Pinpoint 旅程參與指標

下表列出並描述了標準旅程參與指標，您可以針對 Amazon Pinpoint 旅程傳送的所有電子郵件訊息，查詢這些指標並監控趨勢。若要查詢這些指標的資料，請使用 Amazon Pinpoint API 的旅程參與指標資源。表格中的 kpi-name 欄會指出在查詢中針對 `kpi-name` 參數要使用的值。

指標	kpi-name	說明
點按數	journey-emails-clicked	收件人點按訊息中連結的次數。如果單一收件人點按訊息中的多個連結，或多次點按相同連結，每次點按都會計入計數。
按一下，依活動分組	emails-clicked-grouped-by-journey-activity	行程中每項活動之參與者按下郵件中連結的次數。如果單一收件人點按訊息中的多個連結，或多次點按相同連結，每次點按都會計入計數。 此指標的查詢結果會依活動 ID (JourneyActivityId) 分組，這是唯一能夠識別活動的字串。
投訴	journey-emails-complained	參與者回報為未經要求或不想收到的電子郵件訊息數量。
投訴，依活動分組	emails-complained-grouped-by-journey-activity	行程中每項活動之參與者回報為未經要求或不想收到的電子郵件訊息數量。 此指標的查詢結果會依活動 ID (JourneyActivityId) 分組，這是唯一能夠識別活動的字串。
交付	journey-emails-delivered	已交付予參與者的訊息數量。 此指標的計算方式為：傳送的訊息數量，減去由於軟退信或硬退信或因拒絕而無法交付的訊息數量。

指標	kpi-name	說明
交付數，依活動分組	`emails-delivered-grouped-by-journey-activity`	行程中每項活動之已交付予參與者的郵件數目。 此指標的計算方式為：行程中每項活動已傳送的訊息數量，減去由於軟退信或硬退信或因拒絕而無法交付的訊息數量。 此指標的查詢結果會依活動 ID (`JourneyActivityId`) 分組，這是唯一能夠識別活動的字串。
硬退信	`journey-emails-hardbounced`	由於硬退信而無法交付予參與者的訊息數量。如果存在永久性問題導致訊息無法交付 (例如參與者的電子郵件地址不存在)，將會硬退信。
硬退信，依活動分組	`emails-hardbounced-grouped-by-journey-activity`	行程中每項活動之因為硬退信而無法交付予參與者的郵件數量。如果存在永久性問題導致訊息無法交付 (例如參與者的電子郵件地址不存在)，將會硬退信。 此指標的查詢結果會依活動 ID (`JourneyActivityId`) 分組，這是唯一能夠識別活動的字串。
開啟數	`journey-emails-opened`	參與者開啟的訊息數量。

指標	kpi-name	說明
開啟，依活動分組	`emails-opened-grouped-by-journey-activity`	行程中每項活動之參與者開啟的郵件數量。 此指標的查詢結果會依活動 ID (`JourneyActivityId`) 分組，這是唯一能夠識別活動的字串。
拒絕	`journey-emails-rejected`	因為被拒絕而未傳送給參與者的郵件數目。如果 Amazon Pinpoint 判定訊息含惡意程式碼，將拒絕訊息。Amazon Pinpoint 不會嘗試傳送被拒絕的訊息。
拒絕，依活動分組	`emails-rejected-grouped-by-journey-activity`	行程中每項活動之因被拒絕而未傳送給參與者的郵件數目。如果 Amazon Pinpoint 判定訊息含惡意程式碼，將拒絕訊息。Amazon Pinpoint 不會嘗試傳送被拒絕的訊息。 此指標的查詢結果會依活動 ID (`JourneyActivityId`) 分組，這是唯一能夠識別活動的字串。
傳送	`journey-emails-sent`	傳送的訊息數量。
傳送，依活動分組	`emails-sent-grouped-by-journey-activity`	行程中每項活動之已傳送的郵件數目。 此指標的查詢結果會依活動 ID (`JourneyActivityId`) 分組，這是唯一能夠識別活動的字串。

指標	kpi-name	說明
Soft bounces (軟退信)	`journey-emails-softbounced`	由於軟退信而無法交付了參與者的訊息數量。如果存在暫時性問題導致訊息無法交付 (例如參與者的收件匣已滿，或接收伺服器暫時無法使用)，則會軟退信。
軟退信，依活動分組	`emails-softbounced-grouped-by-journey-activity`	行程中每項活動之因為軟退信而無法交付予參與者的郵件數量。如果存在暫時性問題導致訊息無法交付 (例如參與者的收件匣已滿，或接收伺服器暫時無法使用)，則會軟退信。 此指標的查詢結果會依活動 ID (`JourneyActivityId`) 分組，這是唯一能夠識別活動的字串。
取消訂閱	`journey-emails-unsubscribed`	參與者按下郵件中取消訂閱連結的次數。如果單一參與者多次按下同一個取消訂閱連結，則每次按下都會計入計數。
取消訂閱，依活動分組	`emails-unsubscribed-grouped-by-journey-activity`	行程中每項活動之參與者按下郵件中取消訂閱連結的次數。如果單一參與者多次按下同一個取消訂閱連結，則每次按下都會計入計數。 此指標的查詢結果會依活動 ID (`JourneyActivityId`) 分組，這是唯一能夠識別活動的字串。

Amazon Pinpoint 旅程執行指標

下表列出並描述了標準執行指標，您可以查詢這些指標，以評估 Amazon Pinpoint 旅程中參與者的狀態。若要查詢這些指標的資料，請使用 Amazon Pinpoint API 的旅程執行指標資源。表格中的 Field (欄位) 欄位可識別查詢結果所顯示之每項指標的欄位名稱。

指標	欄位	說明
活動參與者	ENDPOINT_ACTIVE	積極繼續行程活動的參與者人數。 此指標的計算方式為：已開始行程的參與者人數，減去退出行程的參與者人數和從行程中移除的參與者人數。
參與者取消	CANCELLED	因旅程取消而未完成旅程的參與者人數。
參與者離開	ENDPOINT_LEFT	離開行程的參與者人數。
參與者加入	ENDPOINT_ENTERED	開始行程的參與者人數。
參與者例外狀況，重複加入限制	REENTRY_CAP_EXCEEDED	因為超過單一參與者可以重複加入行程的次數上限，而未完成行程的參與者人數。
參與者例外狀況，拒絕	ACTIVE_ENDPOINT_REJECTED	因為已是行程中的活動參與者而無法開始行程的參與者人數。 如果參與者開始行程，且您隨後以會影響區段 (根據區段準則) 或行程 (根據活動條件) 中的參與者更新參與者的端點定義，則會拒絕參與者。

Amazon Pinpoint 旅程活動執行指標

下表列出並描述了標準執行指標，您可以查詢這些指標，以評估參與者在 Amazon Pinpoint 旅程中各類單項活動的狀態。若要查詢這些指標的資料，請使用 Amazon Pinpoint API 的旅程活動執行指標資源。表格中的 Metrics (指標) 欄位會列出查詢結果所顯示之各活動類型的欄位。並提供每個欄位的簡短描述。

活動類型	指標
是/否分割 (CONDITIONAL_SPLIT)	這些指標為： • Branch_FALSE – 不符合活動條件，且依照「否」路徑繼續參與活動的參與者人數。 • Branch_TRUE – 符合活動條件，且依照「是」路徑繼續參與活動的參與者人數。 活動中每個路徑可使用的其他指標。如需這些指標的相關資訊，請參閱本表中該活動類型的資料列。
暫停 (HOLDOUT)	這些指標為： • HOLDOUT – 旅程中被移除的參與者人數，屬於活動的保留百分比。 • PASSED – 在旅程中繼續參與下一個活動的參與者人數。
電子郵件 (MESSAGE)	這些指標為： • DAILY_CAP_EXCEEDED – 因已超過單一參與者可在 24 小時內收到的訊息數目上限，而未傳送的訊息數目。 • FAILURE_PERMANENT – 因永久性問題而未傳送的訊息數。 • QUIET_TIME – 本應在參與者所在時區的靜止時間傳送而未交付的訊息數。

活動類型	指標
	• SERVICE_FAILURE – 因 Amazon Pinpoint 發生問題而未傳送的訊息數。 • SUCCESS – 訊息已成功交付給參與者的數目。 • THROTTLED – 因傳送訊息將超出您 Amazon Pinpoint 帳戶的傳送配額而未傳送的訊息數。 • TRANSIENT_FAILURE – 因臨時發生的問題而未傳送的訊息數。 • UNKNOWN – 因不明問題而未傳送的訊息數。
多變量分割 (MULTI_CONDITIONAL_SPLIT)	適用於活動的每個路徑，即前往路徑上活動的參與者人數。 此指標的查詢結果依路徑分組；Branch_## 是路徑的數字識別碼，例如 Branch_1 是活動的第一個路徑。 活動中每個路徑可使用的其他指標。如需這些指標的相關資訊，請參閱本表中該活動類型的資料列。
隨機分割 (RANDOM_SPLIT)	適用於活動的每個路徑，即前往路徑上活動的參與者人數。 此指標的查詢結果依路徑分組；Branch_## 是路徑的數字識別碼，例如 Branch_1 是活動的第一個路徑。 活動中每個路徑可使用的其他指標。如需這些指標的相關資訊，請參閱本表中該活動類型的資料列。

活動類型	指標
等候 (WAIT)	這些指標為： • WAIT_FINISHED – 等到指定時間長度結束的參與者人數。 • WAIT_SKIPPED – 未等到指定時間長度結束的參與者人數，通常是因為他們在活動排定的結束時間之後，才開始參與活動或旅程。 • WAIT_STARTED – 已開始等待，且未跳過或甚至完成等待的參與者人數。
聯絡中心 (CONTACT_CENTER)	這些指標為： • CALL_QUEUED – 已撥入 並排入 Connect Customer 佇列的參與者人數。包括重撥嘗試。 • CONTINUE_WAITING – 繼續等待要嘗試撥號的參與者人數。 • DIAL_FAILURE – 撥號嘗試失敗的參與者人數。 • DROPPED – 傳送時，已不符合先前旅程活動定義的條件的參與者人數。 • TIMEOUT – 在多次撥號嘗試後未收到 Connect Customer 處置代碼的參與者人數。 • WAIT_FINISHED – 等到指定時間長度結束的參與者人數。 • WAIT_FOR_QUIET_HOURS – 為了傳送到管道，而等待靜止時間結束的參與者人數。 • WAIT_STARTED – 已開始等待，且未跳過或甚至完成等待的參與者人數。

Amazon Pinpoint 旅程和行銷活動執行指標

您可以查詢標準執行指標，以評估參與者在 Amazon Pinpoint 旅程或行銷活動中各類單項活動的狀態。若要查詢這些指標的資料，請使用 Amazon Pinpoint API 的旅程執行活動執行指標或行銷活動指標資源。下表列出各類活動查詢結果中顯示的欄位。

指標名稱	適用於旅程、行銷活動或兩者	說明
ENDPOINT_PRODUCED	兩者	篩選之前，最初從客群或事件產生的端點數。
ENDPOINTS_FROM_USER	兩者	如果客戶只有使用者 ID 客群，將新增這些使用者的所有端點。此測量結果會測量以這種方式加入的端點的數目。
ENDPOINT_OPT_OUT	兩者	該端點已選擇退出，且未參與行銷活動或旅程。
ENDPOINT_INACTIVE	兩者	端點處於非作用中狀態，並未參與行銷活動或旅程。
FILTERED_OUT_BY_SEGMENT	兩者	端點不符合客群篩選條件，且未參與行銷活動或旅程。
ENDPOINT_MISSING_ADDRESS	兩者	端點缺少地址，且未參與行銷活動或旅程。
ENDPOINT_MISSING_CHANNEL	兩者	端點缺少一個管道，且未參與行銷活動或旅程。
ENDPOINT_MISSING_TIMEZONE	兩者	端點缺少時區的值，且已被篩選掉。只有在需要時區值時才會發生這種情況。
ENDPOINT_TIMEZONE_MISMATCH	兩者	端點處於當時未在執行範圍內的時區。

指標名稱	適用於旅程、行銷活動或兩者	說明
ENDPOINT_CHANNEL_MISMATCH	行銷活動	行銷活動未針對此端點的管道類型設定訊息。
DUPLICATE_ENDPOINT	兩者	找到重複的端點並刪除。
DUPLICATE_USER	兩者	發現重複的使用者，已從只有使用者 ID 的客群中刪除。如果他們的使用者 ID 相同，會發出 1 指標。
PAUSED	旅程	已從執行移除，因為旅程已暫停。
ENDED	旅程	已從執行移除，因為旅程已結束。
TREATMENT_HOLDOUT	行銷活動	這是在 A/B 活動中，針對隊列與目前處理方法不符的端點而發出。例如在 50/50 A/B 拆分測試中，50% 的端點將針對各種處理方法發出此指標
ENDPOINT_ESTIMATED_TIMEZONE	旅程	時區判斷能夠判斷出端點的時區。

查詢行銷活動的 Amazon Pinpoint 分析資料

除了使用 Amazon Pinpoint 主控台的分析頁面外，也可以使用 Amazon Pinpoint Analytics API 查詢標準指標子集的分析資料，深入了解行銷活動的成果和互動趨勢。

這些指標每一個都是可測量的值，也稱為「關鍵績效指標 (KPI)」，可協助您監控與評估一或多個行銷活動的績效。例如，您可以使用指標了解行銷活動訊息送往的端點數量，或是這些訊息交付給預期端點的數量。

Amazon Pinpoint 會自動收集並彙總所有行銷活動的資料。資料可存放 90 天。如果您使用 Mobile AWS SDK 將行動應用程式與 Amazon Pinpoint 整合，Amazon Pinpoint 會將此支援擴展為包含其他

指標，例如收件人開啟的推送通知百分比。如需整合行動裝置應用程式的資訊，請參閱 將 Amazon Pinpoint 與您的應用程式整合。

若使用 Amazon Pinpoint Analytics API 查詢資料，可以選擇定義查詢範圍、資料、群組和篩選條件的各種選項。您可以使用參數來執行此作業；這些參數會指定您希望查詢的專案、行銷活動和指標，以及任何您希望套用的日期類型篩選條件。

本主題會提供範例，說明如何選擇這些選項，以及如何查詢一或多個行銷活動的資料。

先決條件

在您查詢一或多個行銷活動的分析資料前，收集以下用於定義查詢的資訊，對您將會有所助益：

- 專案 ID – 與行銷活動相關聯的專案的唯一識別符。在 Amazon Pinpoint API 中，此值存放在 `application-id` 屬性內。在 Amazon Pinpoint 主控台上，此值是所有專案頁面上的專案 ID。
- 行銷活動 ID – 行銷活動的唯一識別符 (若您只希望查詢個別行銷活動的資料的話)。在 Amazon Pinpoint API 中，此值存放在 `campaign-id` 屬性內。這個值不會顯示在主控台中。
- 日期範圍 – (選用) 查詢資料日期範圍的第一個和最後一個日期和時間。日期範圍包含在內，且必須不超過 31 個日曆日。此外，也必須從當天起算少於 90 天。若沒有指定日期範圍，Amazon Pinpoint 會自動查詢前 31 個日曆天的資料。
- 指標類型 – 要查詢的指標類型。有兩種類型：「應用程式指標」和「行銷活動指標」。「應用程式指標」可提供與專案相關聯所有行銷活動的資料，也稱為「應用程式」。「行銷活動指標」則只能提供單一行銷活動的資料。
- 指標 – 要查詢的指標名稱，明確地說就是指標的 `kpi-name` 值。如需支援指標的完整清單及每個指標的 `kpi-name` 值，請參閱 專案、行銷活動和旅程的標準指標。

這也有助於您判斷是否要根據相關欄位對資料進行分組。若您執行此作業，您可以選擇旨在為您自動分組資料的指標，來簡化您的分析和報告。例如 Amazon Pinpoint 提供多種標準指標，用於回報訊息已交付給行銷活動收件人的百分比。其中一項指標會依日期 (`successful-delivery-rate-grouped-by-date`) 自動分組資料。另一項指標會依行銷活動執行 (`successful-delivery-rate-grouped-by-campaign-activity`) 自動分組資料。第三個指標只傳回一個值 - 訊息已交付給所有行銷活動收件人的百分比 (`successful-delivery-rate`)。

如果沒有標準指標能按您想要的方式分組資料，您可以開發一系列的查詢，傳回您想要的資料。然後，手動拆分或合併查詢結果，使其成為您設計的自訂群組。

最後，請務必確認您已獲得授權存取您希望查詢的資料。如需詳細資訊，請參閱查詢 Amazon Pinpoint 分析資料的 IAM 政策。

查詢一個行銷活動的 Amazon Pinpoint 資料

若要查詢單一行銷活動的資料，您可以使用行銷活動指標 API，並指定下列必要參數的值：

- application-id – 專案 ID，也就是與行銷活動相關聯的專案的唯一識別符。在 Amazon Pinpoint 中，「專案」和「應用程式」這兩個術語意義相同。
- campaign-id – 行銷活動的唯一識別符。
- kpi-name – 要查詢的指標名稱。此值會說明相關聯的指標，並由兩個以上的術語組成；這些術語都是由小寫的英數字元組成，並以連字號分隔。如需支援指標的完整清單及每個指標的 `kpi-name` 值，請參閱 專案、行銷活動和旅程的標準指標。

您也可以套用篩選條件，查詢特定日期範圍的資料。若沒有指定日期範圍，Amazon Pinpoint 會傳回前 31 個日曆天的資料。若要根據不同的日期篩選資料，請使用支援的日期範圍參數來指定日期範圍第一個和最後一個日期及時間。這些值應依延伸的 ISO 8601 格式顯示，並使用國際標準時間 (UTC)，例如 2019 年 7 月 19 日 8:00 PM UTC 即以 `2019-07-19T20:00:00Z` 顯示。日期範圍包含在內，且必須不超過 31 個日曆日。此外，第一個日期和時間必須小於從當天起算 90 天。

下列範例示範如何使用 Amazon Pinpoint REST API AWS CLI、 和 查詢行銷活動的分析資料 適用於 Java 的 AWS SDK。您可以使用任何支援的 AWS SDK 來查詢行銷活動的分析資料。這些 AWS CLI 範例已針對 Microsoft Windows 格式化。針對 Unix、Linux 和 macOS，請將插入點 (^) 行接續字元替換成反斜線 (\)。

REST API

若要使用 Amazon Pinpoint REST API 查詢行銷活動的分析資料，請傳送 HTTP(S) GET 請求到行銷活動指標 URI。在 URI 中，指定必要路徑參數的適當值：

```
https://endpoint/v1/apps/application-id/campaigns/campaign-id/kpis/daterange/kpi-name
```

其中：

- *##*是託管與行銷活動相關聯專案之 AWS 區域的 Amazon Pinpoint 端點。
- *application-id* 是與行銷活動相關聯專案的唯一識別符。
- *campaign-id* 是行銷活動的唯一識別符。
- *kpi-name* 是要查詢指標的 `kpi-name` 值。

所有參數都必須針對 URL 進行編碼。

若要套用篩選條件，查詢特定日期範圍的資料，請將 start-time 和 end-time 查詢參數及值附加到 URI。藉由使用這些參數，您便可以指定要擷取資料內含日期範圍的第一個及最後一個日期和時間 (格式為已擴充的 ISO 8601 格式)。請使用 & 符號來分隔參數。

例如，以下請求會擷取 2019 年 7 月 19 日至 2019 年 7 月 26 日間，所有已執行行銷活動交付訊息的唯一端點數。

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456example/
campaigns/80b8efd84042ff8d9c96ce2f8example/kpis/daterange/unique-deliveries?start-
time=2019-07-19T00:00:00Z&end-time=2019-07-26T23:59:59Z
```

其中：

- pinpoint.us-east-1.amazonaws.com 是託管專案的 AWS 區域的 Amazon Pinpoint 端點。
- 12345678901234567890123456example 是與行銷活動相關聯專案的唯一識別符。
- 80b8efd84042ff8d9c96ce2f8example 是行銷活動的唯一識別符。
- unique-deliveries 是端點交付行銷活動指標的 kpi-name 值，該指標回報有多少個不重複端點，已收到行銷活動傳送的所有訊息。
- 2019-07-19T00:00:00Z 是要擷取資料的第一個日期和時間，它是內含日期範圍的一部分。
- 2019-07-26T23:59:59Z 則是要擷取資料的最後一個日期和時間，它也是內含日期範圍的一部分。

AWS CLI

若要使用 查詢行銷活動的分析資料 AWS CLI，請使用 get-campaign-date-range-kpi命令並指定所需參數的適當值：

```
C:\> aws pinpoint get-campaign-date-range-kpi ^
    --application-id application-id ^
    --campaign-id campaign-id ^
    --kpi-name kpi-name
```

其中：

- *application-id* 是與行銷活動相關聯專案的唯一識別符。
- *campaign-id* 是行銷活動的唯一識別符。

- *kpi-name* 是要查詢指標的 kpi-name 值。

若要套用篩選條件，查詢特定日期範圍的資料，請將 start-time 和 end-time 參數及值新增到您的查詢。藉由使用這些參數，您便可以指定要擷取資料內含日期範圍的第一個及最後一個日期和時間 (格式為已擴充的 ISO 8601 格式)。例如，以下請求會擷取 2019 年 7 月 19 日至 2019 年 7 月 26 日間，所有已執行行銷活動交付訊息的唯一端點數。

```
C:\> aws pinpoint get-campaign-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --campaign-id 80b8efd84042ff8d9c96ce2f8example ^
    --kpi-name unique-deliveries ^
    --start-time 2019-07-19T00:00:00Z ^
    --end-time 2019-07-26T23:59:59Z
```

其中：

- 12345678901234567890123456example 是與行銷活動相關聯專案的唯一識別符。
- 80b8efd84042ff8d9c96ce2f8example 是行銷活動的唯一識別符。
- unique-deliveries 是端點交付行銷活動指標的 kpi-name 值，該指標回報有多少個不重複端點，已收到行銷活動傳送的所有訊息。
- 2019-07-19T00:00:00Z 是要擷取資料的第一個日期和時間，它是內含日期範圍的一部分。
- 2019-07-26T23:59:59Z 則是要擷取資料的最後一個日期和時間，它也是內含日期範圍的一部分。

SDK for Java

若要使用 查詢行銷活動的分析資料 適用於 Java 的 AWS SDK，請使用行銷活動指標 API 的 GetCampaignDateRangeKpiRequest方法。為所需參數指定適當的值：

```
GetCampaignDateRangeKpiRequest request = new GetCampaignDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withCampaignId("campaignId")
        .withKpiName("kpiName")
```

其中：

- *applicationId* 是與行銷活動相關聯專案的唯一識別符。

- *campaignId* 是行銷活動的唯一識別符。
- *kpiName* 是要查詢指標的 kpi-name 值。

若要套用篩選條件，查詢特定日期範圍的資料，請在您的查詢中包含 startTime 和 endTime 參數及值。藉由使用這些參數，您便可以指定要擷取資料內含日期範圍的第一個及最後一個日期和時間 (格式為已擴充的 ISO 8601 格式)。例如，以下請求會擷取 2019 年 7 月 19 日至 2019 年 7 月 26 日間，所有已執行行銷活動交付訊息的唯一端點數。

```
GetCampaignDateRangeKpiRequest request = new GetCampaignDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123456example")
        .withCampaignId("80b8efd84042ff8d9c96ce2f8example")
        .withKpiName("unique-deliveries")
        .withStartTime(Date.from(Instant.parse("2019-07-19T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-07-26T23:59:59Z")));
```

其中：

- 12345678901234567890123456example 是與行銷活動相關聯專案的唯一識別符。
- 80b8efd84042ff8d9c96ce2f8example 是行銷活動的唯一識別符。
- unique-deliveries 是端點交付行銷活動指標的 kpi-name 值，該指標回報有多少個不重複端點，已收到行銷活動傳送的所有訊息。
- 2019-07-19T00:00:00Z 是要擷取資料的第一個日期和時間，它是內含日期範圍的一部分。
- 2019-07-26T23:59:59Z 則是要擷取資料的最後一個日期和時間，它也是內含日期範圍的一部分。

您傳送查詢後，Amazon Pinpoint 會以 JSON 回應傳回查詢結果。結果的結構會因您查詢的指標而不同。有些指標只會傳回一個值。例如上述範例使用的端點交付 (unique-deliveries) 行銷活動指標傳回一個值 - 有多少個不重複端點，已收到行銷活動傳送的所有訊息。本案例的 JSON 回應如下：

```
{
    "CampaignDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "CampaignId":"80b8efd84042ff8d9c96ce2f8example",
        "EndTime":"2019-07-26T23:59:59Z",
        "KpiName":"unique-deliveries",
        "KpiResult":{
            "Rows":[
```

```
                {
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"123.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-07-19T00:00:00Z"
    }
}
```

其他指標則會傳回多個值，並會根據相關欄位為值進行分組。若指標傳回多個值，則 JSON 回應會包含欄位，指出使用了哪些欄位來為資料進行分組。

若要進一步了解查詢結果的結構，請參閱 使用 JSON 查詢結果。

查詢多個行銷活動的 Amazon Pinpoint 資料

有兩種方式可查詢多個行銷活動的資料。最佳方式取決於您是否全要查詢與相同專案相關聯的行銷活動資料。若如此做，也取決於您要查詢這些行銷活動的全部資料，或僅查詢其部分資料。

若要查詢與不同專案相關聯的行銷活動資料，或是與相同專案相關聯的一部分行銷活動資料，最佳的方法便是建立及執行一系列的個別查詢，其中每個查詢都適用於您希望查詢其資料的每個行銷活動。上一節說明了如何查詢單一行銷活動的資料。

若要查詢與相同專案相關聯的全部行銷活動資料，您可以使用應用程式指標 API。提定下列必要參數的值：

- application-id – 專案 ID，即專案的唯一識別符。在 Amazon Pinpoint 中，「專案」和「應用程式」這兩個術語意義相同。
- kpi-name – 要查詢的指標名稱。此值會說明相關聯的指標，並由兩個以上的術語組成；這些術語都是由小寫的英數字元組成，並以連字號分隔。如需支援指標的完整清單及每個指標的 `kpi-name` 值，請參閱 專案、行銷活動和旅程的標準指標。

您也可以透過日期範圍篩選資料。若沒有指定日期範圍，Amazon Pinpoint 會傳回前 31 個日曆天的資料。若要根據不同的日期篩選資料，請使用支援的日期範圍參數來指定日期範圍第一個和最後一個日期

及時間。這些值應依延伸的 ISO 8601 格式顯示，並使用國際標準時間 (UTC)，例如 2019 年 7 月 19 日 8:00 PM UTC 即以 2019-07-19T20:00:00Z 顯示。日期範圍包含在內，且必須不超過 31 個日曆日。此外，第一個日期和時間必須小於從當天起算 90 天。

下列範例示範如何使用 Amazon Pinpoint REST API AWS CLI、 和 查詢行銷活動的分析資料 適用於 Java 的 AWS SDK。您可以使用任何支援的 AWS SDK 來查詢行銷活動的分析資料。這些 AWS CLI 範例已針對 Microsoft Windows 格式化。針對 Unix、Linux 和 macOS，請將插入點 (^) 行接續字元替換成反斜線 (\)。

REST API

若要使用 Amazon Pinpoint REST API 查詢多個行銷活動的分析資料，請傳送 HTTP(S) GET 請求到應用程式指標 URI。在 URI 中，指定必要路徑參數的適當值：

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

其中：

- *##*是託管與行銷活動相關聯專案之 AWS 區域的 Amazon Pinpoint 端點。
- *application-id* 是與行銷活動相關聯專案的唯一識別符。
- *kpi-name* 是要查詢指標的 kpi-name 值。

所有參數都必須針對 URL 進行編碼。

若要套用篩選條件，擷取特定日期範圍的資料，請將 start-time 和 end-time 查詢參數及值附加到 URI。藉由使用這些參數，您便可以指定要擷取資料內含日期範圍的第一個及最後一個日期和時間 (格式為已擴充的 ISO 8601 格式)。請使用 & 符號來分隔參數。

例如，以下請求會擷取 2019 年 7 月 19 日至 2019 年 7 月 26 日間，專案中每個行銷活動交付訊息到的唯一端點數。

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456example/
kpis/daterange/unique-deliveries-grouped-by-campaign?start-
time=2019-07-19T00:00:00Z&end-time=2019-07-26T23:59:59Z
```

其中：

- pinpoint.us-east-1.amazonaws.com 是託管專案的 AWS 區域的 Amazon Pinpoint 端點。

- 12345678901234567890123456example 是與行銷活動相關聯專案的唯一識別符。
- unique-deliveries-grouped-by-campaign 是端點交付行銷活動指標的 kpi-name 值，該指標傳回有多少個不重複端點，已收到每個行銷活動的訊息。
- 2019-07-19T00:00:00Z 是要擷取資料的第一個日期和時間，它是內含日期範圍的一部分。
- 2019-07-26T23:59:59Z 則是要擷取資料的最後一個日期和時間，它也是內含日期範圍的一部分。

AWS CLI

若要使用 查詢多個行銷活動的分析資料 AWS CLI，請使用 get-application-date-range-kpi命令並指定所需參數的適當值：

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id application-id ^
    --kpi-name kpi-name
```

其中：

- *application-id* 是與行銷活動相關聯專案的唯一識別符。
- *kpi-name* 是要查詢指標的 kpi-name 值。

若要套用篩選條件，擷取特定日期範圍的資料，請在您的查詢中包含 start-time 和 end-time 參數及值。藉由使用這些參數，您便可以指定要擷取資料內含日期範圍的第一個及最後一個日期和時間 (格式為已擴充的 ISO 8601 格式)。例如，以下請求會擷取 2019 年 7 月 19 日至 2019 年 7 月 26 日間，專案中每個行銷活動交付訊息到的唯一端點數。

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --kpi-name unique-deliveries-grouped-by-campaign ^
    --start-time 2019-07-19T00:00:00Z ^
    --end-time 2019-07-26T23:59:59Z
```

其中：

- 12345678901234567890123456example 是與行銷活動相關聯專案的唯一識別符。
- unique-deliveries-grouped-by-campaign 是端點交付行銷活動指標的 kpi-name 值，該指標傳回有多少個不重複端點，已收到每個行銷活動的訊息。

- 2019-07-19T00:00:00Z 是要擷取資料的第一個日期和時間，它是內含日期範圍的一部分。
- 2019-07-26T23:59:59Z 則是要擷取資料的最後一個日期和時間，它也是內含日期範圍的一部分。

SDK for Java

若要使用 查詢多個行銷活動的分析資料 適用於 Java 的 AWS SDK，請使用 Application Metrics API 的 GetApplicationDateRangeKpiRequest方法。為所需參數指定適當的值：

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

其中：

- *applicationId* 是與行銷活動相關聯專案的唯一識別符。
- *kpiName* 是要查詢指標的 kpi-name 值。

若要套用篩選條件，擷取特定日期範圍的資料，請在您的查詢中包含 startTime 和 endTime 參數及值。藉由使用這些參數，您便可以指定要擷取資料內含日期範圍的第一個及最後一個日期和時間 (格式為已擴充的 ISO 8601 格式)。例如，以下請求會擷取 2019 年 7 月 19 日至 2019 年 7 月 26 日間，專案中每個行銷活動交付訊息到的唯一端點數。

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123example")
        .withKpiName("unique-deliveries-grouped-by-campaign")
        .withStartTime(Date.from(Instant.parse("2019-07-19T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-07-26T23:59:59Z")));
```

其中：

- 12345678901234567890123example 是與行銷活動相關聯專案的唯一識別符。
- unique-deliveries-grouped-by-campaign 是端點交付行銷活動指標的 kpi-name 值，該指標傳回有多少個不重複端點，已收到每個行銷活動的訊息。
- 2019-07-19T00:00:00Z 是要擷取資料的第一個日期和時間，它是內含日期範圍的一部分。
- 2019-07-26T23:59:59Z 則是要擷取資料的最後一個日期和時間，它也是內含日期範圍的一部分。

您傳送查詢後，Amazon Pinpoint 會以 JSON 回應傳回查詢結果。結果的結構會因您查詢的指標而不同。有些指標只會傳回一個值。其他指標則會傳回多個值，並會根據相關欄位為那些值進行分組。若指標傳回多個值，則 JSON 回應會包含欄位，指出使用了哪些欄位來為資料進行分組。

例如上述範例使用的端點交付 (依行銷活動分組) (unique-deliveries-grouped-by-campaign) 應用程式指標會傳回多個值 - 有多少個不重複端點，已收到每個與專案相關聯的行銷活動的訊息。本案例的 JSON 回應如下：

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-07-26T23:59:59Z",
        "KpiName":"unique-deliveries-grouped-by-campaign",
        "KpiResult":{
            "Rows":[
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"80b8efd84042ff8d9c96ce2f8example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"123.0"
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"810c7aab86d42fb2b56c8c966example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
```

```
                        "Value":"456.0"
                    }
                ]
            },
            {
                "GroupedBys":[
                    {
                        "Key":"CampaignId",
                        "Type":"String",
                        "Value":"42d8c7eb0990a57ba1d5476a3example"
                    }
                ],
                "Values":[
                    {
                        "Key":"UniqueDeliveries",
                        "Type":"Double",
                        "Value":"789.0"
                    }
                ]
            }
        ]
    },
    "StartTime":"2019-07-19T00:00:00Z"
  }
}
```

在此情況下，GroupedBys 欄位會指出值是以行銷活動 ID (CampaignId) 進行分組。

若要進一步了解查詢結果的結構，請參閱 使用 JSON 查詢結果。

查詢交易訊息的 Amazon Pinpoint 分析資料

除了使用 Amazon Pinpoint 主控台的分析頁面外，也可以使用 Amazon Pinpoint Analytics API 查詢標準指標子集的分析資料，深入了解為專案傳送的交易訊息的成果和互動趨勢。

這些指標每一個都是可測量的值，也稱為「關鍵績效指標 (KPI)」，其可協助您監控與評估交易訊息的績效。例如您可以使用指標了解自己傳送了多少交易電子郵件或 SMS 訊息，或者有多少訊息已交付給收件人。Amazon Pinpoint 會針對您為專案傳送的所有交易電子郵件和 SMS 訊息，自動收集並彙總相關資料。資料可存放 90 天。

若使用 Amazon Pinpoint Analytics API 查詢資料，可以選擇定義查詢範圍、資料、群組和篩選條件的各種選項。您可以使用參數來執行此作業；這些參數會指定您希望查詢的專案和指標，以及任何您希望套用的日期類型篩選條件。

本主題會提供範例，說明如何選擇這些選項以及查詢專案的交易訊息資料。

先決條件

在您查詢交易訊息的分析資料前，收集以下可用於定義查詢的資訊，將會對您有所幫助：

- 專案 ID – 傳送訊息的專案的唯一識別符。在 Amazon Pinpoint API 中，此值存放在 `application-id` 屬性內。在 Amazon Pinpoint 主控台上，此值是所有專案頁面上的專案 ID。
- 日期範圍 – (選用) 查詢資料日期範圍的第一個和最後一個日期和時間。日期範圍包含在內，且必須不超過 31 個日曆日。此外，也必須從當天起算少於 90 天。若沒有指定日期範圍，Amazon Pinpoint 會自動查詢前 31 個日曆天的資料。
- 指標 – 要查詢的指標名稱，明確地說就是指標的 `kpi-name` 值。如需支援指標的完整清單及每個指標的 `kpi-name` 值，請參閱 專案、行銷活動和旅程的標準指標。

這也有助於您判斷是否要根據相關欄位對資料進行分組。若您執行此作業，您可以選擇旨在為您自動分組資料的指標，來簡化您的分析和報告。例如 Amazon Pinpoint 提供了多個標準指標，用於回報已交付給收件人的交易 SMS 訊息數。其中一項指標會依日期 (`txn-sms-delivered-grouped-by-date`) 自動分組資料。另一項指標會依國家或地區 (`txn-sms-delivered-grouped-by-country`) 自動分組資料。第三個指標只傳回一個值 - 訊息已交付給收件人的數目 (`txn-sms-delivered`)。如果沒有標準指標能按您想要的方式分組資料，您可以開發一系列的查詢，傳回您想要的資料。然後，手動拆分或合併查詢結果，使其成為您設計的自訂群組。

最後，請務必確認您已獲得授權存取您希望查詢的資料。如需詳細資訊，請參閱查詢 Amazon Pinpoint 分析資料的 IAM 政策。

查詢交易電子郵件訊息的 Amazon Pinpoint 資料

若要查詢為專案傳送的交易電子郵件訊息資料，您可以使用應用程式指標 API 並指定下列必要參數的值：

- application-id – 專案 ID，即專案的唯一識別符。在 Amazon Pinpoint 中，「專案」和「應用程式」這兩個術語意義相同。

- kpi-name – 要查詢的指標名稱。此值會說明相關聯的指標，並由兩個以上的術語組成；這些術語都是由小寫的英數字元組成，並以連字號分隔。如需支援指標的完整清單及每個指標的 kpi-name 值，請參閱 專案、行銷活動和旅程的標準指標。

您也可以套用篩選條件，查詢特定日期範圍的資料。若沒有指定日期範圍，Amazon Pinpoint 會傳回前 31 個日曆天的資料。若要根據不同的日期篩選資料，請使用支援的日期範圍參數來指定日期範圍第一個和最後一個日期及時間。這些值應依延伸的 ISO 8601 格式顯示，並使用國際標準時間 (UTC)，例如 2019 年 9 月 6 日 8:00 PM UTC 即以 2019-09-06T20:00:00Z 顯示。日期範圍包含在內，且必須不超過 31 個日曆日。此外，第一個日期和時間必須小於從當天起算 90 天。

下列範例示範如何使用 Amazon Pinpoint REST API AWS CLI、 和 查詢交易電子郵件訊息的分析資料適用於 Java 的 AWS SDK。您可以使用任何支援的 AWS 開發套件來查詢交易訊息的分析資料。這些 AWS CLI 範例已針對 Microsoft Windows 格式化。針對 Unix、Linux 和 macOS，請將插入點 (^) 行接續字元替換成反斜線 (\)。

REST API

若要使用 Amazon Pinpoint REST API 查詢交易電子郵件訊息的分析資料，請傳送 HTTP(S) GET 請求到應用程式指標 URI。在 URI 中，指定必要路徑參數的適當值：

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

其中：

- *##*是託管專案之 AWS 區域的 Amazon Pinpoint 端點。
- *application-id* 是專案的唯一識別符。
- *kpi-name* 是要查詢指標的 kpi-name 值。

所有參數都必須針對 URL 進行編碼。

若要套用篩選條件，查詢特定日期範圍的資料，請將 start-time 和 end-time 查詢參數及值附加到 URI。藉由使用這些參數，您便可以指定要擷取資料內含日期範圍的第一個及最後一個日期和時間 (格式為已擴充的 ISO 8601 格式)。請使用 & 符號來分隔參數。

例如，以下請求會擷取在 2019 年 9 月 6 日至 2019 年 9 月 13 日間，為專案傳送交易電子郵件訊息的數量：

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456789012345example/
kpis/daterange/txn-emails-sent?start-time=2019-09-06T00:00:00Z&end-
time=2019-09-13T23:59:59Z
```

其中：

- pinpoint.us-east-1.amazonaws.com 是託管專案的 AWS 區域的 Amazon Pinpoint 端點。
- 12345678901234567890123456789012345example 是專案的唯一識別符。
- txn-emails-sent 是傳送應用程式指標的 kpi-name 值，該指標回報針對專案傳送了多少交易電子郵件訊息。
- 2019-09-06T00:00:00Z 是要擷取資料的第一個日期和時間，它是內含日期範圍的一部分。
- 2019-09-13T23:59:59Z 則是要擷取資料的最後一個日期和時間，它也是內含日期範圍的一部分。

AWS CLI

若要使用 查詢交易電子郵件訊息的分析資料 AWS CLI，請使用 get-application-date-range-kpi命令，並指定所需參數的適當值：

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id application-id ^
    --kpi-name kpi-name
```

其中：

- *application-id* 是專案的唯一識別符。
- *kpi-name* 是要查詢指標的 kpi-name 值。

若要套用篩選條件，查詢特定日期範圍的資料，請將 start-time 和 end-time 參數及值新增到您的查詢。藉由使用這些參數，您便可以指定要擷取資料內含日期範圍的第一個及最後一個日期和時間 (格式為已擴充的 ISO 8601 格式)。例如，以下請求會擷取在 2019 年 9 月 6 日至 2019 年 9 月 13 日間，為專案傳送交易電子郵件訊息的數量：

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id 12345678901234567890123456789012345example ^
    --kpi-name txn-emails-sent ^
    --start-time 2019-09-06T00:00:00Z ^
```

```
    --end-time 2019-09-13T23:59:59Z
```

其中：

- 12345678901234567890123456example 是專案的唯一識別符。
- txn-emails-sent 是傳送應用程式指標的 kpi-name 值，該指標回報針對專案傳送了多少交易電子郵件訊息。
- 2019-09-06T00:00:00Z 是要擷取資料的第一個日期和時間，它是內含日期範圍的一部分。
- 2019-09-13T23:59:59Z 則是要擷取資料的最後一個日期和時間，它也是內含日期範圍的一部分。

SDK for Java

若要使用 查詢交易電子郵件訊息的分析資料 適用於 Java 的 AWS SDK，請使用 Application Metrics API 的 GetApplicationDateRangeKpiRequest方法。為所需參數指定適當的值：

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

其中：

- *applicationId* 是專案的唯一識別符。
- *kpiName* 是要查詢指標的 kpi-name 值。

若要套用篩選條件，查詢特定日期範圍的資料，請在您的查詢中包含 startTime 和 endTime 參數及值。藉由使用這些參數，您便可以指定要擷取資料內含日期範圍的第一個及最後一個日期和時間 (格式為已擴充的 ISO 8601 格式)。例如，以下請求會擷取在 2019 年 9 月 6 日至 2019 年 9 月 13 日間，為專案傳送交易電子郵件訊息的數量：

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("12345678901234567890123456example")
        .withKpiName("txn-emails-sent")
        .withStartTime(Date.from(Instant.parse("2019-09-06T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-09-13T23:59:59Z")));
```

其中：

- 12345678901234567890123456example 是專案的唯一識別符。
- txn-emails-sent 是傳送應用程式指標的 kpi-name 值，該指標回報針對專案傳送了多少交易電子郵件訊息。
- 2019-09-06T00:00:00Z 是要擷取資料的第一個日期和時間，它是內含日期範圍的一部分。
- 2019-09-13T23:59:59Z 則是要擷取資料的最後一個日期和時間，它也是內含日期範圍的一部分。

您傳送查詢後，Amazon Pinpoint 會以 JSON 回應傳回查詢結果。結果的結構會因您查詢的指標而不同。有些指標只會傳回一個值。例如上述範例使用的傳送 (txn-emails-sent) 應用程式指標會傳回一個值 - 從專案傳送了多少交易電子郵件訊息。本案例的 JSON 回應如下：

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-09-13T23:59:59Z",
        "KpiName":"txn-emails-sent",
        "KpiResult":{
            "Rows":[
                {
                    "Values":[
                        {
                            "Key":"TxnEmailsSent",
                            "Type":"Double",
                            "Value":"62.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-09-06T00:00:00Z"
    }
}
```

其他指標則會傳回多個值，並依相關欄位分組這些值。若指標傳回多個值，則 JSON 回應會包含欄位，指出使用了哪些欄位來為資料進行分組。

若要進一步了解查詢結果的結構，請參閱 使用 JSON 查詢結果。

查詢交易簡訊的 Amazon Pinpoint 資料

若要查詢為專案傳送的交易簡訊訊息資料，您可以使用應用程式指標 API 並指定下列必要參數的值：

- application-id – 專案 ID，即專案的唯一識別符。在 Amazon Pinpoint 中，「專案」和「應用程式」這兩個術語意義相同。
- kpi-name – 要查詢的指標名稱。此值會說明相關聯的指標，並由兩個以上的術語組成；這些術語都是由小寫的英數字元組成，並以連字號分隔。如需支援指標的完整清單及每個指標的 `kpi-name` 值，請參閱 專案、行銷活動和旅程的標準指標。

您也可以套用篩選條件，查詢特定日期範圍的資料。若沒有指定日期範圍，Amazon Pinpoint 會傳回前 31 個日曆天的資料。若要按不同的日期篩選資料，請使用支援的日期範圍參數指定日期範圍的第一個和最後一個日期及時間。這些值應依延伸的 ISO 8601 格式顯示，並使用國際標準時間 (UTC)，例如 2019 年 9 月 6 日 8:00 PM UTC 即以 `2019-09-06T20:00:00Z` 顯示。日期範圍包含在內，且必須不超過 31 個日曆日。此外，第一個日期和時間必須小於從當天起算 90 天。

下列範例示範如何使用 Amazon Pinpoint REST API AWS CLI、 和 查詢交易簡訊的分析資料 適用於 Java 的 AWS SDK。您可以使用任何支援的 AWS SDK 來查詢交易訊息的分析資料。這些 AWS CLI 範例已針對 Microsoft Windows 格式化。針對 Unix、Linux 和 macOS，請將插入點 (^) 行接續字元替換成反斜線 (\)。

REST API

若要透過 Amazon Pinpoint REST API 查詢交易 SMS 訊息的分析資料，請傳送 HTTP(S) GET 請求到應用程式指標 URI。在 URI 中，指定必要路徑參數的適當值：

```
https://endpoint/v1/apps/application-id/kpis/daterange/kpi-name
```

其中：

- *##*是託管專案之 AWS 區域的 Amazon Pinpoint 端點。
- *application-id* 是專案的唯一識別符。
- *kpi-name* 是要查詢指標的 `kpi-name` 值。

所有參數都必須針對 URL 進行編碼。

若要套用篩選條件，擷取特定日期範圍的資料，請將 start-time 和 end-time 查詢參數及值附加到 URI。藉由使用這些參數，您便可以指定要擷取資料內含日期範圍的第一個及最後一個日期和時間 (格式為已擴充的 ISO 8601 格式)。請使用 & 符號來分隔參數。

例如，以下請求會擷取在 2019 年 9 月 6 日至 2019 年 9 月 8 日間，每天為專案傳送交易簡訊訊息的數量：

```
https://pinpoint.us-east-1.amazonaws.com/v1/apps/12345678901234567890123456example/
kpis/daterange/txn-sms-sent-grouped-by-date?start-time=2019-09-06T00:00:00Z&end-
time=2019-09-08T23:59:59Z
```

其中：

- pinpoint.us-east-1.amazonaws.com 是託管專案的 AWS 區域的 Amazon Pinpoint 端點。
- 12345678901234567890123456example 是專案的唯一識別符。
- txn-sms-sent-grouped-by-date 是傳送 (依日期分組) 應用程式指標的 kpi-name 值，該指標傳回日期範圍內每天傳送了多少交易 SMS 訊息。
- 2019-09-06T00:00:00Z 是要擷取資料的第一個日期和時間，它是內含日期範圍的一部分。
- 2019-09-08T23:59:59Z 則是要擷取資料的最後一個日期和時間，它也是內含日期範圍的一部分。

AWS CLI

若要使用 查詢交易簡訊的分析資料 AWS CLI，請使用 get-application-date-range-kpi命令，並指定所需參數的適當值：

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id application-id ^
    --kpi-name kpi-name
```

其中：

- *application-id* 是專案的唯一識別符。
- *kpi-name* 是要查詢指標的 kpi-name 值。

若要套用篩選條件，擷取特定日期範圍的資料，請在您的查詢中包含 start-time 和 end-time 參數及值。藉由使用這些參數，您便可以指定要擷取資料內含日期範圍的第一個及最後一個日期和

時間 (格式為已擴充的 ISO 8601 格式)。例如，以下請求會擷取在 2019 年 9 月 6 日至 2019 年 9 月 8 日間，每天為專案傳送交易簡訊訊息的數量：

```
C:\> aws pinpoint get-application-date-range-kpi ^
    --application-id 12345678901234567890123456example ^
    --kpi-name txn-sms-sent-grouped-by-date ^
    --start-time 2019-09-06T00:00:00Z ^
    --end-time 2019-09-08T23:59:59Z
```

其中：

- `12345678901234567890123456example` 是專案的唯一識別符。
- `txn-sms-sent-grouped-by-date` 是傳送 (依日期分組) 應用程式指標的 `kpi-name` 值，該指標傳回日期範圍內每天傳送了多少交易 SMS 訊息。
- `2019-09-06T00:00:00Z` 是要擷取資料的第一個日期和時間，它是內含日期範圍的一部分。
- `2019-09-08T23:59:59Z` 則是要擷取資料的最後一個日期和時間，它也是內含日期範圍的一部分。

SDK for Java

若要使用 查詢交易簡訊的分析資料 適用於 Java 的 AWS SDK，請使用 Application Metrics API 的 GetApplicationDateRangeKpiRequest方法，並指定所需參數的適當值：

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
        .withApplicationId("applicationId")
        .withKpiName("kpiName")
```

其中：

- *`applicationId`* 是專案的唯一識別符。
- *`kpiName`* 是要查詢指標的 `kpi-name` 值。

若要套用篩選條件，擷取特定日期範圍的資料，請在您的查詢中包含 `startTime` 和 `endTime` 參數及值。藉由使用這些參數，您便可以指定要擷取資料內含日期範圍的第一個及最後一個日期和時間 (格式為已擴充的 ISO 8601 格式)。例如，以下請求會擷取在 2019 年 9 月 6 日至 2019 年 9 月 8 日間，每天為專案傳送交易簡訊訊息的數量：

```
GetApplicationDateRangeKpiRequest request = new GetApplicationDateRangeKpiRequest()
```

```
        .withApplicationId("12345678901234567890123456example")
        .withKpiName("txn-sms-sent-grouped-by-date")
        .withStartTime(Date.from(Instant.parse("2019-09-06T00:00:00Z")))
        .withEndTime(Date.from(Instant.parse("2019-09-08T23:59:59Z")));
```

其中：

- 12345678901234567890123456example 是專案的唯一識別符。
- txn-sms-sent-grouped-by-date 是傳送 (依日期分組) 應用程式指標的 kpi-name 值，該指標傳回日期範圍內每天傳送了多少交易 SMS 訊息。
- 2019-09-06T00:00:00Z 是要擷取資料的第一個日期和時間，它是內含日期範圍的一部分。
- 2019-09-08T23:59:59Z 則是要擷取資料的最後一個日期和時間，它也是內含日期範圍的一部分。

您傳送查詢後，Amazon Pinpoint 會以 JSON 回應傳回查詢結果。結果的結構會因您查詢的指標而不同。有些指標只會傳回一個值。其他指標則會傳回多個值，並依相關欄位分組這些值。若指標傳回多個值，則 JSON 回應會包含欄位，指出使用了哪些欄位來為資料進行分組。

例如上述範例使用的傳送 (依日期分組) (txn-sms-sent-grouped-by-date) 應用程式指標會傳回多個值 - 在指定的日期範圍內，每天傳送了多少交易 SMS 訊息。本案例的 JSON 回應如下：

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-09-08T23:59:59Z",
        "KpiName":"txn-sms-sent-grouped-by-date",
        "KpiResult":{
            "Rows":[
                {
                    "GroupedBys":[
                        {
                            "Key":"Date",
                            "Type":"String",
                            "Value":"2019-09-06"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"TxnSmsSent",
                            "Type":"Double",
```

```
                        "Value":"29.0"
                    }
                ]
            },
            {
                "GroupedBys":[
                    {
                        "Key":"Date",
                        "Type":"String",
                        "Value":"2019-09-07"
                    }
                ],
                "Values":[
                    {
                        "Key":"TxnSmsSent",
                        "Type":"Double",
                        "Value":"35.0"
                    }
                ]
            },
            {
                "GroupedBys":[
                    {
                        "Key":"Date",
                        "Type":"String",
                        "Value":"2019-09-08"
                    }
                ],
                "Values":[
                    {
                        "Key":"TxnSmsSent",
                        "Type":"Double",
                        "Value":"10.0"
                    }
                ]
            }
        ]
    },
    "StartTime":"2019-09-06T00:00:00Z"
  }
}
```

在此情況下，GroupedBys 欄位會指出值是以日曆中的天 (Date) 進行分組。這表示：

- 2019 年 9 月 6 日傳送了 29 則訊息。
- 2019 年 9 月 7 日傳送了 35 則訊息。
- 2019 年 9 月 8 日傳送了 10 則訊息。

若要進一步了解查詢結果的結構，請參閱 使用 JSON 查詢結果。

使用 Amazon Pinpoint 分析 JSON 查詢結果

使用 Amazon Pinpoint Analytics API 查詢分析資料時，Amazon Pinpoint 會在 JSON 回應中傳回結果。應用程式指標、行銷活動指標和旅程參與指標的回應資料，符合用於回報 Amazon Pinpoint 分析資料的標準 JSON 結構描述。

這表示您可以使用所選程式設計語言或工具實作自訂的解決方案，查詢一或多個這些指標的資料、擷取每個查詢的結果，然後將結果寫入資料表、物件或其他位置。您接著便可以在該位置使用其他服務或應用程式來使用查詢結果。

例如，您可以：

- 建置自訂儀表板，定期查詢一組指標，並使用您慣用的資料視覺化框架來顯示結果。
- 建立報告，查詢適當的指標來追蹤參與率，並在圖表或其他您設計的報告類型中顯示結果。
- 剖析並將分析資料寫入特定儲存格式，然後將結果移植到長期儲存體解決方案。

請注意，Amazon Pinpoint Analytics API 並非用於建立或儲存任何永久性物件，讓您隨後可以在 Amazon Pinpoint 專案或 Amazon Pinpoint 帳戶中讀取或使用。相反地，API 旨在協助您擷取分析資料，並將該資料傳輸到其他服務和應用程式進行進一步的分析、儲存或報告。部分使用相同的 JSON 回應結構和結構描述處理所有分析資料以執行此作業，分析資料可以以程式設計方式查詢其應用程式指標、行銷活動指標和行程參與指標。

本主題說明應用程式指標、行銷活動指標或行程參與指標查詢之 JSON 回應的結構、物件和欄位。如需行程執行指標或行程活動執行指標查詢之 JSON 回應的欄位相關資訊，請參閱適用於 Amazon Pinpoint 專案、行銷活動和旅程的標準指標。

JSON 結構

為了協助您剖析和使用查詢結果，Amazon Pinpoint Analytics API 對所有 Amazon Pinpoint 分析資料使用相同的 JSON 回應結構，您可以透過程式化方式查詢應用程式指標、行銷活動指標和旅程參與指

標。每個 JSON 回應都會指定定義查詢的值，例如專案 ID (ApplicationId)。回應還包括一個 (且只有一個) KpiResult 物件。KpiResult 物件包含查詢的整體結果集。

每個 KpiResult 物件都包含一個 Rows 物件。此為物件陣列，包含查詢結果及這些結果值的相關中繼資料。Rows 物件的結構和內容具有下列一般特性：

- 查詢結果的每一列都是個別的 JSON 物件，名為 Values，並位於 Rows 物件中。例如，如果查詢傳回三個值，則 Rows 物件會包含三個 Values 物件。每個 Values 物件都包含查詢的個別結果。
- 查詢結果的每一行都是其適用 Values 物件的屬性。行的名稱會存放在 Values 物件的 Key 欄位中。
- 針對分組的查詢結果，每個 Values 物件都會有相關聯的 GroupedBys 物件。GroupedBys 物件會指出分組結果所用的欄位。也會提供相關聯 Values 物件的分組值。
- 若指標的查詢結果為 Null，則 Rows 物件將為空白。

除了這些一般特性外，Rows 物件的結構和內容會因指標而不同。這是因為 Amazon Pinpoint 支援兩種指標類型：單一值指標和多值指標。

「單一值指標」僅提供一個累積值。例如，所有行銷活動執行交付予收件者的郵件百分比。「多值指標」提供一個以上的值，並依相關欄位分組這些值。例如，依行銷活動執行分組，行銷活動每次執行交付予收件者的郵件百分比。

您可以藉由查看指標的名稱，來快速判斷指標是單一值指標還是多個值指標。若名稱不包含 grouped-by，則為單一值指標。若包含該項目，則為多個值指標。如需您可以透過編寫程式方式查詢的指標完整清單，請參閱適用於 Amazon Pinpoint 專案、行銷活動和旅程的標準指標。

單一值指標

針對單一值指標，Rows 物件會包含一個 Values 物件，該物件：

- 指定查詢指標的易記名稱。
- 提供查詢指標的值。
- 識別傳回值的資料類型。

例如，下列 JSON 回應包含單一值指標的查詢結果。此指標會報告自 2019 年 8 月 1 日至 2019 年 8 月 31 日期間，與專案相關聯之所有行銷活動交付郵件的唯一端點數目：

```
{
    "ApplicationDateRangeKpiResponse":{
```

```
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-08-31T23:59:59Z",
        "KpiName":"unique-deliveries",
        "KpiResult":{
            "Rows":[
                {
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"1368.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-08-01T00:00:00Z"
    }
}
```

在此範例中，回應指出所有專案的行銷活動於 2019 年 8 月 1 日到 2019 年 8 月 31 日間，一共將訊息交付到了 1,368 個唯一端點，其中：

- Key 是其值在 Value 欄位中指定的指標易記名稱 (UniqueDeliveries)。
- Type 是在 Value 欄位中指定值的資料類型 (Double)。
- Value 是經查詢指標的實際值，包括套用的任何篩選條件 (1368.0)。

如果單一值測量結果的查詢結果為 Null (不大於或等於零)，Rows 物件會是空的。如果沒有任何要為指標傳回的資料，Amazon Pinpoint 會傳回該指標的空值。例如：

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"23456789012345678901234567890example",
        "EndTime":"2019-08-31T23:59:59Z",
        "KpiName":"unique-deliveries",
        "KpiResult":{
            "Rows":[

            ]
        },
        "StartTime":"2019-08-01T00:00:00Z"
```

```
    }
}
```

多個值指標

多個值指標的 Rows 物件結構和內容大致上與單一值指標相同。多值指標的 Rows 物件也包含一個 Values 物件。Values 物件會指定所查詢的指標易記名稱、提供該指標的值，以及識別該值的資料類型。

但是，多個值指標的 Rows 物件也包含一或多個 GroupedBy 物件。查詢結果中每個 Values 物件都有一個 GroupedBy 物件。GroupedBy 物件會指出分組結果資料所用的欄位，以及該欄位的資料類型。也會指出該欄位的分組值 (針對相關聯的 Values 物件)。

例如，以下 JSON 回應包含多個值指標的查詢結果，報告從 2019 年 8 月 1 日到 2019 年 8 月 31 日間，與專案相關聯每個行銷活動交付訊息到的唯一端點數。

```
{
    "ApplicationDateRangeKpiResponse":{
        "ApplicationId":"12345678901234567890123456example",
        "EndTime":"2019-08-31T23:59:59Z",
        "KpiName":"unique-deliveries-grouped-by-campaign",
        "KpiResult":{
            "Rows":[
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"80b8efd84042ff8d9c96ce2f8example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"123.0"
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
```

```
                            "Type":"String",
                            "Value":"810c7aab86d42fb2b56c8c966example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"456.0"
                        }
                    ]
                },
                {
                    "GroupedBys":[
                        {
                            "Key":"CampaignId",
                            "Type":"String",
                            "Value":"42d8c7eb0990a57ba1d5476a3example"
                        }
                    ],
                    "Values":[
                        {
                            "Key":"UniqueDeliveries",
                            "Type":"Double",
                            "Value":"789.0"
                        }
                    ]
                }
            ]
        },
        "StartTime":"2019-08-01T00:00:00Z"
    }
}
```

在此範例中，回應指出有三個專案的行銷活動在 2019 年 8 月 1 日到 2019 年 8 月 31 日間將訊息交付到了唯一端點。針對這些行銷活動，交付計數的明細如下：

- 行銷活動 80b8efd84042ff8d9c96ce2f8example 將訊息交付到了 123 個唯一端點。
- 行銷活動 810c7aab86d42fb2b56c8c966example 將訊息交付到了 456 個唯一端點。
- 行銷活動 42d8c7eb0990a57ba1d5476a3example 將訊息交付到了 789 個唯一端點。

其中物件和欄位的一般結構是：

- GroupedBys.Key – 屬性或欄位的名稱，在 GroupedBys.Value 欄位中存放了指定的分組值 (CampaignId)。
- GroupedBys.Type – GroupedBys.Value 欄位中指定值的資料類型 (String)。
- GroupedBys.Value – 用於對資料分組的欄位實際值，如 GroupedBys.Key 欄位 (行銷活動 ID) 所示。
- Values.Key – Values.Value 欄位中指定值的指標易記名稱 (UniqueDeliveries)。
- Values.Type – Values.Value 欄位中指定值的資料類型 (Double)。
- Values.Value – 所查詢指標的實際值，包括套用的任何篩選器。

若特定專案、行銷活動或其他資源的多值指標查詢結果為 Null (不大於或等於零)，Amazon Pinpoint 不會傳回該資源的任何物件或欄位。若所有資源的多值指標查詢結果是 Null，Amazon Pinpoint 會傳回空白的 Rows 物件。

JSON 物件和欄位

除了指定定義查詢的值，例如專案 ID (ApplicationId)，應用程式指標、行銷活動指標或行程參與指標查詢的每個 JSON 回應都包含一個 KpiResult 物件。此物件包含查詢的整體結果集，您可以剖析此結果集來傳送分析資料到另一個服務或應用程式。每個 KpiResult 物件都包含一些或所有下列標準物件和欄位，取決於指標。

物件或欄位	Description
Rows	包含查詢結果集的物件陣列。
Rows.GroupedBys	針對多個值指標，此為定義欄位，以及用來為查詢結果中資料進行分組值的欄位陣列。
Rows.GroupedBys.Key	針對多個值指標，此為存放 GroupedBy s.Value 欄位中指定值的屬性或欄位名稱。
Rows.GroupedBys.Type	針對多個值指標，此為 GroupedBys.Value 欄位中指定值的資料類型。
Rows.GroupedBys.Value	針對多個值指標，此為用來為查詢結果中資料進行分組欄位的實際值。此值與關聯的 Values 物件相互關聯。

物件或欄位	Description
`Rows.Values`	包含查詢結果的欄位陣列。
`Rows.Values.Key`	查詢的指標易記名稱。指標的值會在 `Values.Value` 欄位中指定。
`Rows.Values.Type`	`Values.Value` 欄位中指定值的資料類型。
`Rows.Values.Value`	經查詢指標的實際值，包含任何套用的篩選條件。

如需行程執行指標或行程活動執行指標查詢之 JSON 回應的欄位相關資訊，請參閱適用於 Amazon Pinpoint 專案、行銷活動和旅程的標準指標。

使用 記錄 Amazon Pinpoint API 呼叫 AWS CloudTrail

Amazon Pinpoint 已與 整合 AWS CloudTrail，此服務提供 Amazon Pinpoint 中使用者、角色 AWS 或服務所採取動作的記錄。CloudTrail 會擷取 Amazon Pinpoint API 呼叫當作事件。擷取的呼叫包括 Amazon Pinpoint 主控台的呼叫，以及對 Amazon Pinpoint API 操作進行的程式碼呼叫。

如果建立追蹤，就可以將 CloudTrail 事件持續交付給 Amazon Simple Storage Service (Amazon S3) 儲存貯體，包括 Amazon Pinpoint 事件。即使您未設定追蹤，依然可以透過 CloudTrail 主控台中的事件歷史記錄檢視最新事件。您可以利用 CloudTrail 收集的資訊，判斷向 Amazon Pinpoint 提出的請求、提出請求的 IP 地址、提出請求的人員、提出請求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，包括如何設定及啟用，請參閱《AWS CloudTrail 使用者指南》。

CloudTrail 中的 Amazon Pinpoint 資訊

當您建立 AWS 帳戶時，會在您的帳戶上啟用 CloudTrail。當 Amazon Pinpoint 中發生支援的事件活動時，該活動會與事件歷史記錄中的其他 AWS 服務事件一起記錄在 CloudTrail 事件中。您可以在 AWS 帳戶中檢視、搜尋和下載最近的事件。如需詳細資訊，請參閱使用 CloudTrail 事件歷史記錄檢視事件。

若要持續記錄您 AWS 帳戶中的事件，包括 Amazon Pinpoint 的事件，請建立追蹤。線索能讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。根據預設，當您在主控台中建立線索時，線索會套用至所有 AWS 區域。線索會記錄 AWS 分割區中所有區域的事件，並將日誌檔案傳送到您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析和處理 CloudTrail 日誌中收集的事件資料。如需詳細資訊，請參閱下列內容：

- 建立追蹤的概觀
- CloudTrail 支援的服務和整合
- 設定 CloudTrail 的 Amazon SNS 通知
- 接收多個區域的 CloudTrail 日誌檔案和接收多個帳戶的 CloudTrail 日誌檔案

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷：

- 是否使用根登入資料或 AWS Identity and Access Management 使用者登入資料提出請求。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 請求是否由其他 AWS 服務提出。

如需更多詳細資訊，請參閱 CloudTrail userIdentity 元素。

您可以建立追蹤記錄，並將日誌檔案存放在 Amazon S3 儲存貯體，存放時間無限制。您也可以定義 Amazon S3 生命週期規則，自動封存或刪除日誌檔案。您的日誌檔案預設使用 Amazon S3 伺服器端加密 (SSE) 加密。

若要收到日誌檔案交付的通知，請設定 CloudTrail 在新日誌檔案交付時發布 SNS 通知。如需詳細資訊，請參閱為 CloudTrail 設定 Amazon SNS 通知。

您也可以將多個 AWS 區域和多個 AWS 帳戶的 Amazon Pinpoint 日誌檔案彙總到單一 Amazon S3 儲存貯體。詳情請參閱從多個區域接收 CloudTrail 日誌檔案，以及從多個帳戶接收 CloudTrail 日誌檔案。

您可以使用 CloudTrail，記錄以下 Amazon Pinpoint API 的動作：

- Amazon Pinpoint API
- Amazon Pinpoint SMS 和語音 API

CloudTrail 日誌檔案中支援的 Amazon Pinpoint API 動作

Amazon Pinpoint API 支援將下列動作，記錄為 CloudTrail 日誌檔案中的事件：

- CreateApp
- CreateCampaign
- CreateEmailTemplate
- CreateExportJob
- CreateImportJob
- CreateJourney
- CreatePushTemplate
- CreateRecommenderConfiguration
- CreateSegment
- CreateSmsTemplate
- CreateVoiceTemplate
- DeleteAdmChannel
- DeleteApnsChannel
- DeleteApnsSandboxChannel

- DeleteApnsVoipChannel
- DeleteApnsVoipSandboxChannel
- DeleteApp
- DeleteBaiduChannel
- DeleteCampaign
- DeleteEmailChannel
- DeleteEmailTemplate
- DeleteEndpoint
- DeleteEventStream
- DeleteGcmChannel
- DeleteJourney
- DeletePushTemplate
- DeleteRecommenderConfiguration
- DeleteSegment
- DeleteSmsChannel
- DeleteSmsTemplate
- DeleteUserEndpoints
- DeleteVoiceChannel
- DeleteVoiceTemplate
- GetAdmChannel
- GetApnsChannel
- GetApnsSandboxChannel
- GetApnsVoipChannel
- GetApnsVoipSandboxChannel
- GetApp
- GetApplicationDateRangeKpi
- GetApplicationSettings
- GetApps
- GetBaiduChannel
- GetCampaign

- GetCampaignActivities
- GetCampaignDateRangeKpi
- GetCampaignVersion
- GetCampaignVersions
- GetCampaigns
- GetChannels
- GetEmailChannel
- GetEmailTemplate
- GetEndpoint
- GetEventStream
- GetExportJob
- GetExportJobs
- GetGcmChannel
- GetImportJob
- GetImportJobs
- GetJourney
- GetJourneyDateRangeKpi
- GetJourneyExecutionActivityMetrics
- GetJourneyExecutionMetrics
- GetPushTemplate
- GetRecommenderConfiguration
- GetRecommenderConfigurations
- GetSegment
- GetSegmentExportJobs
- GetSegmentImportJobs
- GetSegmentVersion
- GetSegmentVersions
- GetSegments
- GetSmsChannel
- GetSmsTemplate

- GetUserEndpoints
- GetVoiceChannel
- GetVoiceTemplate
- ListJourneys
- ListTagsForResource
- ListTemplates
- ListTemplateVersions
- PhoneNumberValidate
- PutEvents
- PutEventStream
- RemoveAttributes
- SendMessages
- SendOTPMessage
- SendUsersMessages
- TagResource
- UntagResource
- UpdateAdmChannel
- UpdateApnsChannel
- UpdateApnsSandboxChannel
- UpdateApnsVoipChannel
- UpdateApnsVoipSandboxChannel
- UpdateApplicationSettings
- UpdateBaiduChannel
- UpdateCampaign
- UpdateEmailChannel
- UpdateEmailTemplate
- UpdateEndpoint
- UpdateEndpointsBatch
- UpdateGcmChannel
- UpdateJourney

- UpdateJourneyState
- UpdatePushTemplate
- UpdateRecommenderConfiguration
- UpdateSegment
- UpdateSmsChannel
- UpdateSmsTemplate
- UpdateTemplateActiveVersion
- UpdateVoiceChannel
- UpdateVoiceTemplate

CloudTrail 日誌檔案中支援的 Amazon Pinpoint 電子郵件 API 動作

Amazon Pinpoint 電子郵件 API 支援將下列動作，記錄為 CloudTrail 日誌檔案中的事件：

- CreateConfigurationSet
- CreateConfigurationSetEventDestination
- CreateDedicatedIpPool
- CreateEmailIdentity
- DeleteConfigurationSet
- DeleteConfigurationSetEventDestination
- DeleteDedicatedIpPool
- DeleteEmailIdentity
- GetAccount
- GetConfigurationSet
- GetConfigurationSetEventDestinations
- GetDedicatedIp
- GetDedicatedIps
- GetEmailIdentity
- ListConfigurationSets
- ListDedicatedIpPools
- ListEmailIdentities

- PutAccountDedicatedIpWarmupAttributes
- PutAccountSendingAttributes
- PutConfigurationSetDeliveryOptions
- PutConfigurationSetReputationOptions
- PutConfigurationSetSendingOptions
- PutConfigurationSetTrackingOptions
- PutDedicatedIpInPool
- PutDedicatedIpWarmupAttributes
- PutEmailIdentityDkimAttributes
- PutEmailIdentityFeedbackAttributes
- PutEmailIdentityMailFromAttributes
- UpdateConfigurationSetEventDestination

以下 Amazon Pinpoint 電子郵件 API 動作，不會記錄在 CloudTrail 中：

- SendEmail

CloudTrail 日誌檔案中支援的 Amazon Pinpoint SMS 和語音 API 第 1 版動作

Amazon Pinpoint SMS 和語音 API 第 1 版支援將下列動作，記錄為 CloudTrail 日誌檔案中的事件：

- CreateConfigurationSet
- CreateConfigurationSetEventDestination
- DeleteConfigurationSet
- DeleteConfigurationSetEventDestination
- GetConfigurationSetEventDestinations
- UpdateConfigurationSetEventDestination

以下 Amazon Pinpoint SMS 和語音第 1 版 API 動作，未記錄在 CloudTrail 中：

- SendVoiceMessage

顯示 Amazon Pinpoint API 動作的 CloudTrail 日誌項目範例

追蹤是一種組態，能讓事件以日誌檔案的形式交付到您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一個或多個日誌項目。事件代表來自任何來源的單一請求。其中包含了請求的動作、動作的日期和時間、請求參數等相關資訊。CloudTrail 日誌檔並非依公有 API 呼叫的堆疊追蹤排序，因此不會以任何特定順序出現。

以下範例顯示的 CloudTrail 日誌項目，是 Amazon Pinpoint API 的 `GetCampaigns` 和 `CreateCampaign` 動作。


```
{
  "Records": [
    {
      "awsRegion": "us-east-1",
      "eventID": "example0-09a3-47d6-a810-c5f9fd2534fe",
      "eventName": "GetCampaigns",
      "eventSource": "pinpoint.amazonaws.com",
      "eventTime": "2018-02-03T00:56:48Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.05",
      "readOnly": true,
      "recipientAccountId": "123456789012",
      "requestID": "example1-b9bb-50fa-abdb-80f274981d60",
      "requestParameters": {
        "application-id": "example71dfa4c1aab66332a5839798f",
        "page-size": "1000"
      },
      "responseElements": null,
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "Jersey/${project.version} (HttpUrlConnection 1.8.0_144)",
      "userIdentity": {
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "accountId": "123456789012",
        "arn": "arn:aws:iam::123456789012:root",
        "principalId": "123456789012",
        "sessionContext": {
          "attributes": {
            "creationDate": "2018-02-02T16:55:29Z",
            "mfaAuthenticated": "false"
          }
        },
        "type": "Root"
```

```
      }
    },
    {
      "awsRegion": "us-east-1",
      "eventID": "example0-09a3-47d6-a810-c5f9fd2534fe",
      "eventName": "CreateCampaign",
      "eventSource": "pinpoint.amazonaws.com",
      "eventTime": "2018-02-03T01:05:16Z",
      "eventType": "AwsApiCall",
      "eventVersion": "1.05",
      "readOnly": false,
      "recipientAccountId": "123456789012",
      "requestID": "example1-b9bb-50fa-abdb-80f274981d60",
      "requestParameters": {
        "Description": "***",
        "HoldoutPercent": 0,
        "IsPaused": false,
        "MessageConfiguration": "***",
        "Name": "***",
        "Schedule": {
          "Frequency": "ONCE",
          "IsLocalTime": true,
          "StartTime": "2018-02-03T00:00:00-08:00",
          "Timezone": "utc-08"
        },
        "SegmentId": "exampleda204adf991a80281aa0e591",
        "SegmentVersion": 1,
        "application-id": "example71dfa4c1aab66332a5839798f"
      },
      "responseElements": {
        "ApplicationId": "example71dfa4c1aab66332a5839798f",
        "CreationDate": "2018-02-03T01:05:16.425Z",
        "Description": "***",
        "HoldoutPercent": 0,
        "Id": "example54a654f80948680cbba240ede",
        "IsPaused": false,
        "LastModifiedDate": "2018-02-03T01:05:16.425Z",
        "MessageConfiguration": "***",
        "Name": "***",
        "Schedule": {
          "Frequency": "ONCE",
          "IsLocalTime": true,
          "StartTime": "2018-02-03T00:00:00-08:00",
          "Timezone": "utc-08"
```

```
        },
        "SegmentId": "example4da204adf991a80281example",
        "SegmentVersion": 1,
        "State": {
          "CampaignStatus": "SCHEDULED"
        },
        "Version": 1
      },
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "aws-cli/1.14.9 Python/3.4.3 Linux/3.4.0+ botocore/1.8.34",
      "userIdentity": {
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "accountId": "123456789012",
        "arn": "arn:aws:iam::123456789012:user/userName",
        "principalId": "AIDAIHTHRCDA62EXAMPLE",
        "type": "IAMUser",
        "userName": "userName"
      }
    }
  ]
}
```

以下範例顯示的 CloudTrail 日誌項目，說明了 Amazon Pinpoint SMS 和語音 API 中的 `CreateConfigurationSet` 和 `CreateConfigurationSetEventDestination` 動作。

```
{
  "Records": [
    {
      "eventVersion":"1.05",
      "userIdentity":{
        "type":"IAMUser",
        "principalId":"AIDAIHTHRCDA62EXAMPLE",
        "arn":"arn:aws:iam::111122223333:user/SampleUser",
        "accountId":"111122223333",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "userName":"SampleUser"
      },
      "eventTime":"2018-11-06T21:45:55Z",
      "eventSource":"sms-voice.amazonaws.com",
      "eventName":"CreateConfigurationSet",
      "awsRegion":"us-east-1",
      "sourceIPAddress":"192.0.0.1",
      "userAgent":"PostmanRuntime/7.3.0",
```

```
      "requestParameters":{
        "ConfigurationSetName":"MyConfigurationSet"
      },
      "responseElements":null,
      "requestID":"56dcc091-e20d-11e8-87d2-9994aexample",
      "eventID":"725843fc-8846-41f4-871a-7c52dexample",
      "readOnly":false,
      "eventType":"AwsApiCall",
      "recipientAccountId":"123456789012"
    },
    {
      "eventVersion":"1.05",
      "userIdentity":{
        "type":"IAMUser",
        "principalId":"AIDAIHTHRCDA62EXAMPLE",
        "arn":"arn:aws:iam::111122223333:user/SampleUser",
        "accountId":"111122223333",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "userName":"SampleUser"
      },
      "eventTime":"2018-11-06T21:47:08Z",
      "eventSource":"sms-voice.amazonaws.com",
      "eventName":"CreateConfigurationSetEventDestination",
      "awsRegion":"us-east-1",
      "sourceIPAddress":"192.0.0.1",
      "userAgent":"PostmanRuntime/7.3.0",
      "requestParameters":{
        "EventDestinationName":"CloudWatchEventDestination",
        "ConfigurationSetName":"MyConfigurationSet",
        "EventDestination":{
          "Enabled":true,
          "MatchingEventTypes":[
            "INITIATED_CALL",
            "INITIATED_CALL"
          ],
          "CloudWatchLogsDestination":{
            "IamRoleArn":"arn:aws:iam::111122223333:role/iamrole-01",
            "LogGroupArn":"arn:aws:logs:us-east-1:111122223333:log-
group:clientloggroup-01"
          }
        }
      },
      "responseElements":null,
      "requestID":"81de1e73-e20d-11e8-b158-d5536example",
```

```
      "eventID":"fcafc21f-7c93-4a3f-9e72-fca2dexample",
      "readOnly":false,
      "eventType":"AwsApiCall",
      "recipientAccountId":"111122223333"
    }
  ]
}
```

在 Amazon Pinpoint 中使用推薦者模型搭配 AWS Lambda

您可以在 Amazon Pinpoint 中，從推薦者模型擷取個人化建議，並加入您要透過行銷活動和旅程傳送的訊息中。推薦人模型是一種機器學習 (ML) 模型，可在資料中找出模式，並根據它找到的模式產生預測和建議。它會預測特定使用者偏好的一組特定產品或項目，並以一組建議的形式將該資訊提供給使用者。

透過 Amazon Pinpoint 使用推薦者模型，就能根據每個收件人的屬性和行為，將個人化建議傳送給訊息收件人。透過 AWS Lambda，您也可以自訂和增強這些建議。例如，您可以動態將建議從單一文字值 (例如產品名稱或 ID) 轉換為更複雜的內容 (例如產品名稱、描述和影像)。Amazon Pinpoint 傳送訊息時，您也可以即時執行此操作。

此功能適用於下列 AWS 區域：美國東部（維吉尼亞北部）、美國西部（奧勒岡）、亞太區域（孟買）、亞太區域（雪梨）和歐洲（愛爾蘭）。

將推薦者模型建議新增至 Amazon Pinpoint 中的訊息

若要將推薦者模型搭配 Amazon Pinpoint 使用，您可以先從建立 Amazon Personalize 解決方案開始，然後將該解決方案部署為 Amazon Personalize 行銷活動。然後您可以在 Amazon Pinpoint 中建立推薦者模型的組態。您可以在組態中指定設定，決定從 Amazon Personalize 行銷活動，擷取和處理建議資料的方式。這包括是否叫用 AWS Lambda 函數來執行所擷取資料的額外處理。

Amazon Personalize 是一項 AWS 服務，旨在協助您建立 ML 模型，為使用應用程式的客戶提供即時、個人化的建議。Amazon Personalize 會引導您完成建立和訓練 ML 模型的程序，然後將模型準備和部署為 Amazon Personalize 行銷活動。然後，您可以從行銷活動中擷取即時、個人化建議。若要進一步了解 Amazon Personalize，請參閱 Amazon Personalize 開發人員指南。

AWS Lambda 是一種運算服務，您可以用來執行程式碼，而無需佈建或管理伺服器。您可以封裝程式碼並將其上傳到 AWS Lambda 做為 Lambda 函數。 AWS Lambda 然後， 會在叫用函數時執行函數。您可以手動調用函數，也可以根據事件，或根據應用程式或服務 (包括 Amazon Pinpoint) 提出的請求，自動調用函數。如需建立並調用 Lambda 函數的相關資訊，請參閱AWS Lambda 開發人員指南。

建立推薦者模型的 Amazon Pinpoint 組態後，可將模型中的建議，加入從行銷活動和旅程傳送的訊息。您將使用包含建議屬性訊息變數之訊息範本來執行這項操作。建議的屬性是用來存放建議資料的動態端點或使用者屬性。您可以在建立推薦者模型的組態時定義這些屬性。

您可以針對下列訊息範本類型中的建議屬性使用變數：

- 電子郵件範本，適用於您從行銷活動或旅程傳送的電子郵件訊息。
- 推播通知範本，用於您從行銷活動傳送的推播通知。
- SMS 範本，適用於您從行銷活動傳送的 SMS 文字訊息。

如需使用推薦者模型搭配 Amazon Pinpoint 的詳細資訊，請參閱 Amazon Pinpoint 使用者指南中的機器學習模型。

如果將 Amazon Pinpoint 設為調用 Lambda 函數以處理建議資料，Amazon Pinpoint 每次在行銷活動或旅程的訊息中傳送個人化建議時，都會執行以下一般任務：

1. 評估和處理訊息和訊息範本的組態設定和內容。
2. 決定訊息範本連線到推薦人模型。
3. 評估連接和使用模型的組態設定。這些是由模型的推薦者模型資源所定義。
4. 為模型組態設定定義的推薦屬性偵測一或多個訊息變數。
5. 從您在模型組態設定中指定的 Amazon Personalize 行銷活動，擷取建議資料。它使用 Amazon Personalize Runtime API 的 GetRecommendations 操作執行這項任務。
6. 將適當的建議資料新增至每個訊息收件人的動態建議屬性 (`RecommendationItems`)。
7. 調用 Lambda 函數，並將每個收件人的建議資料傳送到 Lambda 函數進行處理。

   該資料會以 JSON 物件的形式傳送，其中包含每個收件人的端點定義。每個端點定義都包含一個 `RecommendationItems` 欄位，內含 1-5 個值的已排序陣列。陣列中值的數量依模型的組態設定而定。
8. 等待 Lambda 函數處理資料並傳回結果。

   結果是其中包含每個收件人更新的端點定義 JSON 物件。每個更新的端點定義都包含一個新 `Recommendations` 物件。此物件包含 1-10 個欄位，每個欄位對應您在模型組態設定中定義的自訂推薦屬性。這些欄位都會存放端點的增強建議資料。
9. 使用每個收件人更新的端點定義，以該收件人的適當值取代每個訊息變數。
10. 傳送訊息版本，其中包含每個訊息收件人的個人化建議。

若要以此方式自訂和增強建議，請先建立一個 Lambda 函數來處理 Amazon Pinpoint 傳送的端點定義，然後傳回更新的端點定義。接著為 Lambda 函數指派 Lambda 函數政策，並授權 Amazon Pinpoint 調用 Lambda 函數。接著在 Amazon Pinpoint 中設定推薦者模型。在您設定模型時，請指定要叫用的函數，並定義要使用的建議屬性。

為 Amazon Pinpoint 建立 Lambda 函數以叫用推薦者模型

若要了解如何建立 Lambda 函數，請參閱 AWS Lambda 開發人員指南中的入門。在您設計和開發函數時，請牢記以下要求和準則。

輸入事件資料

Amazon Pinpoint 為推薦者模型調用 Lambda 函數時，會傳送一個承載，內含傳送訊息的行銷活動或旅程的組態和其他設定。承載包括一個 Endpoints 物件，它是一種對應，可將端點 ID 與訊息收件人的端點定義產生關聯。

端點定義使用的結構，是由 Amazon Pinpoint API 的端點資源定義。不過，它們也包含一個名為 RecommendationItems 的動態建議屬性欄位。此 RecommendationItems 欄位包含一或多個該端點的推薦項目，由 Amazon Personalize 行銷活動傳回。此欄位的值，是一個包含 1-5 個推薦項目 (以字串表示) 的已排序陣組。陣列中的項目數，視您為每個端點或使用者設定 Amazon Pinpoint 要擷取的推薦項目數而定。

例如：

```
"Endpoints": {
    "endpointIDexample-1":{
        "ChannelType":"EMAIL",
        "Address":"sofiam@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.875Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"SofiaMartínez",
            "UserAttributes":{
                "LastName":[
                    "Martínez"
                ],
                "FirstName":[
                    "Sofia"
                ],
                "Neighborhood":[
                    "East Bay"
```

```
                ]
            }
        },
        "RecommendationItems":[
            "1815",
            "2009",
            "1527"
        ],
        "CreationDate":"2020-02-26T18:56:24.875Z"
    },
    "endpointIDexample-2":{
        "ChannelType":"EMAIL",
        "Address":"alejandror@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.897Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"AlejandroRosalez",
            "UserAttributes":{
                "LastName ":[
                    "Rosalez"
                ],
                "FirstName":[
                    "Alejandro"
                ],
                "Neighborhood":[
                    "West Bay"
                ]
            }
        },
        "RecommendationItems":[
            "1210",
            "6542",
            "4582"
        ],
        "CreationDate":"2020-02-26T18:56:24.897Z"
    }
}
```

在上述的範例中，相關的 Amazon Pinpoint 設定如下：

- 推薦者模型是設定為擷取每個端點或使用者的三個建議項目。(`RecommendationsPerMessage` 屬性的值設為 3) Amazon Pinpoint 使用此設定，為每個端點或使用者只擷取並新增第一個、第二個和第三個推薦項目。
- 該專案已設定為使用自訂使用者屬性，可存放每個使用者的名字、姓氏和他們所在的社區。(`UserAttributes` 物件包含這些屬性的值)。
- 該專案已設定為使用自訂端點屬性 (`AddressType`)，指出端點是否為使用者偏好的地址 (通道)，以接收來自專案的訊息。(`Attributes` 物件包含此屬性的值)。

Amazon Pinpoint 調用 Lambda 函數，並將此承載當作事件資料傳送時， AWS Lambda 將資料傳遞給 Lambda 函數進行處理。

每個承載最多可包含 50 個端點的資料。如果客群包含超過 50 個端點，Amazon Pinpoint 會重複調用函數，一次最多呼叫 50 個端點，直到該函數處理完所有資料為止。

回應資料和需求

設計和開發 Lambda 函數時，請記住機器學習模型的配額。如果函數不符合這些配額定義的條件，Amazon Pinpoint 將無法處理並傳送訊息。

同時請記住下列需求：

- 函數必須以輸入事件資料所提供的相同格式傳回更新的端點定義。
- 每個更新的端點定義可以包含 1–10 個端點或使用者的自訂建議屬性。這些屬性的名稱，必須符合您在 Amazon Pinpoint 中設定推薦者模型時所指定的屬性名稱。
- 針對每一個端點或使用者，所有自訂建議屬性都必須以單一的 `Recommendations` 物件回傳。此需求有助於確保不會發生命名衝突。您可以將 `Recommendations` 物件新增至端點定義中的任何位置。
- 每個自訂建議屬性的值必須是字串 (單一值) 或字串陣列 (多個值)。如果值是字串陣列，建議您維持 Amazon Personalize 傳回的推薦項目順序，如 `RecommendationItems` 欄位所示。否則，您的內容可能無法反映模型對端點或使用者的預測。
- 函數不應該修改事件資料中的其他元素，包括端點或使用者的其他屬性值。它應該只新增並傳回自訂建議屬性的值。Amazon Pinpoint 不會接受函數回應中任何其他值的更新。
- 函數必須託管在與叫用函數的 Amazon Pinpoint 專案相同的 AWS 區域中。如果函數和專案不在同一個區域，Amazon Pinpoint 無法將事件資料傳送到該函數。

如果不符合上述任一要求，Amazon Pinpoint 將無法處理訊息，並將訊息傳送至一個或多個端點。這可能會導致行銷活動或旅程活動失敗。

最後，我們建議您為函數保留 256 個並行執行。

整體而言，您的 Lambda 函數應處理 Amazon Pinpoint 傳送的事件資料，並傳回修改後的端點定義。透過在 `Endpoints` 物件中的逐一查看每個端點，並針對每個端點建立和設定您要使用的自訂建議屬性值來完成此操作。下列範例處理常式 (使用 Python 編寫並繼續上述的輸入事件資料範例) 顯示了這一點：

```
import json
import string

def lambda_handler(event, context):
    print("Received event: " + json.dumps(event))
    print("Received context: " +  str(context))
    segment_endpoints = event["Endpoints"]
    new_segment = dict()
    for endpoint_id in segment_endpoints.keys():
        endpoint = segment_endpoints[endpoint_id]
        if supported_endpoint(endpoint):
            new_segment[endpoint_id] = add_recommendation(endpoint)

    print("Returning endpoints: " + json.dumps(new_segment))
    return new_segment

def supported_endpoint(endpoint):
    return True

def add_recommendation(endpoint):
    endpoint["Recommendations"] = dict()

    customTitleList = list()
    customGenreList = list()
    for i,item in enumerate(endpoint["RecommendationItems"]):
        item = int(item)
        if item == 1210:
            customTitleList.insert(i, "Hanna")
            customGenreList.insert(i, "Action")
        elif item == 1527:
            customTitleList.insert(i, "Catastrophe")
            customGenreList.insert(i, "Comedy")
        elif item == 1815:
```

```
            customTitleList.insert(i, "Fleabag")
            customGenreList.insert(i, "Comedy")
        elif item == 2009:
            customTitleList.insert(i, "Late Night")
            customGenreList.insert(i, "Drama")
        elif item == 4582:
            customTitleList.insert(i, "Agatha Christie\'s The ABC Murders")
            customGenreList.insert(i, "Crime")
        elif item == 6542:
            customTitleList.insert(i, "Hunters")
            customGenreList.insert(i, "Drama")

    endpoint["Recommendations"]["Title"] = customTitleList
    endpoint["Recommendations"]["Genre"] = customGenreList

    return endpoint
```

在上述範例中，會將事件資料 AWS Lambda 傳遞給處理常式做為 `event` 參數。處理常式會逐一查看 `Endpoints` 物件中的每個端點，並為名為 `Recommendations.Title` 和 `Recommendations.Genre` 的自訂建議屬性設定值。`return` 陳述式會將每個更新的端點定義傳回 Amazon Pinpoint。

繼續先前的輸入事件資料範例，更新的端點定義為：

```
"Endpoints":{
    "endpointIDexample-1":{
        "ChannelType":"EMAIL",
        "Address":"sofiam@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.875Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"SofiaMartínez",
            "UserAttributes":{
                "LastName":[
                    "Martínez"
                ],
                "FirstName":[
```

```
                    "Sofia"
                ],
                "Neighborhood":[
                    "East Bay"
                ]
            }
        },
        "RecommendationItems":[
            "1815",
            "2009",
            "1527"
        ],
        "CreationDate":"2020-02-26T18:56:24.875Z",
        "Recommendations":{
            "Title":[
                "Fleabag",
                "Late Night",
                "Catastrophe"
            ],
            "Genre":[
                "Comedy",
                "Comedy",
                "Comedy"
            ]
        }
    },
    "endpointIDexample-2":{
        "ChannelType":"EMAIL",
        "Address":"alejandror@example.com",
        "EndpointStatus":"ACTIVE",
        "OptOut":"NONE",
        "EffectiveDate":"2020-02-26T18:56:24.897Z",
        "Attributes":{
            "AddressType":[
                "primary"
            ]
        },
        "User":{
            "UserId":"AlejandroRosalez",
            "UserAttributes":{
                "LastName ":[
                    "Rosalez"
                ],
                "FirstName":[
```

```
                    "Alejandro"
                ],
                "Neighborhood":[
                    "West Bay"
                ]
            }
        },
        "RecommendationItems":[
            "1210",
            "6542",
            "4582"
        ],
        "CreationDate":"2020-02-26T18:56:24.897Z",
        "Recommendations":{
            "Title":[
                "Hanna",
                "Hunters",
                "Agatha Christie\'s The ABC Murders"
            ],
            "Genre":[
                "Action",
                "Drama",
                "Crime"
            ]
        }
    }
}
```

在上述範例中，函數會修改接收並傳回結果的 Endpoints 物件。每個端點的 Endpoint 物件現在包含一個新的 Recommendations 物件，其中包含 Title 和 Genre 欄位。每個欄位都存放了三個值 (作為字串) 的排序陣列，其中每個值都會為 RecommendationItems 欄位中對應的建議項目提供增強內容。

指派 Lambda 函數政策以授權 Amazon Pinpoint 處理建議資料

使用 Lambda 函數處理建議資料之前，必須先授權 Amazon Pinpoint 調用 Lambda 函數。若要授予調用許可，請將 Lambda 函數政策指派給 Lambda 函數。Lambda 函數政策是資源型許可政策，指定哪些實體可以使用函數，以及這些實體可以執行哪些動作。詳情請參閱 AWS Lambda 開發人員指南中的為 AWS Lambda使用資源型政策。

######## Amazon Pinpoint ######## Amazon Pinpoint ## (projectId) 中的特定 Amazon Pinpoint 行銷活動 (*campaignId*) 使用 lambda:InvokeFunction 動作：

```
{
  "Sid": "sid",
  "Effect": "Allow",
  "Principal": {
    "Service": "pinpoint.us-east-1.amazonaws.com"
  },
  "Action": "lambda:InvokeFunction",
  "Resource": "{arn:aws:lambda:us-east-1:accountId:function:function-name}",
  "Condition": {
    "ArnLike": {
      "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
    }
  }
}
```

您的函數政策需要包含 AWS:SourceArn 金鑰的 Condition 區塊。此金鑰指定了允許呼叫函數的資源。在前面的範例中，該政策允許一個特定的行銷活動叫用該函數。

您也可以撰寫政策，讓 Amazon Pinpoint 服務主體針對特定 Amazon Pinpoint 專案 (*projectId*) 中的所有行銷活動和旅程使用 lambda:InvokeFunction 動作。此如以下範例政策所示：

```
{
  "Sid": "sid",
  "Effect": "Allow",
  "Principal": {
    "Service": "pinpoint.us-east-1.amazonaws.com"
  },
  "Action": "lambda:InvokeFunction",
  "Resource": "{arn:aws:lambda:us-east-1:accountId:function:function-name}",
  "Condition": {
    "ArnLike": {
      "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
    }
  }
}
```

與第一個範例不同，這個例子 Condition 區塊中的 AWS:SourceArn 金鑰允許一個特定的專案叫用該函數。此許可適用於專案中的所有行銷活動和旅程。

若要編寫較一般的政策，您可以使用多字元比對萬用字元 (*)。例如您可以使用以下 Condition 區塊，允許任何 Amazon Pinpoint 專案調用函數：

```
"Condition": {
  "ArnLike": {
    "AWS:SourceArn": "arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*"
  }
}
```

如果想將 Lambda 函數用於 Amazon Pinpoint 帳戶的所有專案，建議您依照上述方式設定政策的 `Condition` 區塊。不過，最佳實務是您應該建立僅包含在特定資源執行特定動作所需許可的政策。

授權 Amazon Pinpoint 使用 AWS CLI 和 Lambda add-permission 命令叫用 Lambda 函數

將 Lambda 函數政策指派給函數之後，您可以新增許可，允許 Amazon Pinpoint 叫用特定專案、行銷活動或旅程的函數。您可以使用 AWS Command Line Interface (AWS CLI) 和 Lambda [add-permission](#)命令來執行此操作。以下範例顯示了如何為特定專案 (*projectId*) 執行此操作：

```
$ aws lambda add-permission \
--function-name function-name \
--statement-id sid \
--action lambda:InvokeFunction \
--principal pinpoint.us-east-1.amazonaws.com \
--source-arn arn:aws:mobiletargeting:us-east-1:accountId:recommenders/*
```

上述範例格式適用於 Unix、Linux 和 macOS。若使用 Microsoft Windows，請使用插入符號(^) 取代反斜線 (\) 行接續字元。

如果此命令成功執行，您會看到類似如下的輸出：

```
{
  "Statement": "{\"Sid\":\"sid\",
    \"Effect\":\"Allow\",
    \"Principal\":{\"Service\":\"pinpoint.us-east-1.amazonaws.com\"},
    \"Action\":\"lambda:InvokeFunction\",
    \"Resource\":\"arn:aws:lambda:us-east-1:111122223333:function:function-name\",
    \"Condition\":
      {\"ArnLike\":
        {\"AWS:SourceArn\":
         \"arn:aws:mobiletargeting:us-east-1:111122223333:recommenders/*\"}}}"
}
```

`Statement` 值是陳述式的 JSON 字串版本，且已新增至 Lambda 函數政策。

設定 Amazon Pinpoint 以叫用建議者模型的 Lambda 函數

若要將 Amazon Pinpoint 設為可調用推薦者統模型的 Lambda 函數，請為該模型指定以下 Lambda 專用的組態設定：

- `RecommendationTransformerUri` – 此屬性指定 Lambda 函數的名稱或 Amazon Resource Name (ARN)。
- `Attributes` – 此物件是一個對應，定義了函數新增到每個端點定義的自訂建議屬性。這些屬性中的每一個都可以做為訊息範本中的訊息變數使用。

使用 Amazon Pinpoint API 的推薦者統模型資源 (若已為模型建立組態)，或 Amazon Pinpoint API 的推薦者統模型資源 (若已更新模型的組態)，可指定這些設定。您也可以使用 Amazon Pinpoint 主控台定義這些設定。

如需使用推薦者模型搭配 Amazon Pinpoint 的詳細資訊，請參閱 Amazon Pinpoint 使用者指南中的機器學習模型。

刪除您的 Amazon Pinpoint 專案並移除敏感個人資料

Amazon Pinpoint 可能儲存某些可能被視為個人資料的內容，視您的使用方式而定。舉例而言，Amazon Pinpoint 中的端點包含最終使用者的聯絡資訊，像是電子郵件地址或行動電話號碼。

您可以使用主控台或 Amazon Pinpoint API 永久刪除個人資料。本主題包含刪除各種可視為個人資料的程序。

您也可以完全關閉 AWS 您的帳戶。如需詳細資訊，請參閱《 AWS 帳戶管理 參考指南》中的關閉 AWS 帳戶。

刪除所有 Amazon Pinpoint 專案資料

這有可能永久刪除您為 Amazon Pinpoint 專案儲存的所有資料。只要刪除該專案，即可完成此操作。

> **Warning**
>
> 如果您刪除專案，Amazon Pinpoint 會將該專案所有的專案專用設定及資料一併刪除。資訊無法復原。

您刪除項目時，Amazon Pinpoint 會針對推播通知和雙向簡訊管道，刪除所有與專案相關的設定，以及儲存在 Amazon Pinpoint 中的所有客群、行銷活動、旅程和專案相關分析資料，例如：

- 客群 – 所有客群設定和資料。如果是動態客群，這包括您定義的客群群組和篩選條件。如果是匯入的客群，這包括端點、使用者 ID 和您匯入的其他資料，以及您套用的任何篩選條件。
- 行銷活動 – 所有訊息、訊息處理和變數、分析資料、排程及其他設定。
- 旅程 – 所有活動、分析資料、排程及其他設定。
- 分析 – 所有參與指標的資料，例如為行銷活動和旅程傳送和交付的訊息數，以及所有旅程執行指標。對於行動和 Web 應用程式，所有未串流到其他服務的事件資料， AWS 例如 Amazon Kinesis、所有漏斗，以及應用程式用量、收入和人口統計指標的資料。刪除專案之前，我們建議您將此筆資料會出到另一個位置。

您可以使用 Amazon Pinpoint 主控台刪除專案。如需詳細資訊，請參閱 Amazon Pinpoint 使用者指南中的刪除專案。您也可以使用 Amazon Pinpoint API 的應用程式資源，以程式化方式刪除專案。

使用 AWS SDKs Amazon Pinpoint 程式碼範例

下列程式碼範例示範如何搭配 AWS 軟體開發套件 (SDK) 使用 Amazon Pinpoint。

如需 AWS SDK 開發人員指南的完整清單和程式碼範例，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

程式碼範例

- 使用AWS SDKs Amazon Pinpoint 程式碼範例
  - 使用AWS SDKs Amazon Pinpoint 基本範例
    - 使用AWS SDKs 的 Amazon Pinpoint 動作
      - CreateApp 搭配AWS SDK 或 CLI 使用
      - CreateCampaign 搭配AWS SDK 使用
      - CreateExportJob 搭配AWS SDK 使用
      - CreateImportJob 搭配AWS SDK 使用
      - CreateSegment 搭配AWS SDK 使用
      - DeleteApp 搭配AWS SDK 或 CLI 使用
      - DeleteEndpoint 搭配AWS SDK 使用
      - GetEndpoint 搭配AWS SDK 或 CLI 使用
      - GetSegments 搭配AWS SDK 使用
      - GetSmsChannel 搭配AWS SDK 或 CLI 使用
      - GetUserEndpoints 搭配AWS SDK 使用
      - SendMessages 搭配AWS SDK 或 CLI 使用
      - UpdateEndpoint 搭配AWS SDK 使用
  - 使用AWS SDKs Amazon Pinpoint 案例
    - 推送通知入門
- 使用 SDK 的 Amazon Pinpoint SMS 和語音 API 程式碼範例 AWS SDKs
  - 使用 AWS SDKs 的 Amazon Pinpoint SMS 和語音 API 基本範例
    - 使用 AWS SDKs 的 Amazon Pinpoint SMS 和語音 API 動作
      - CreateConfigurationSet 搭配 AWS SDK 使用
      - CreateConfigurationSetEventDestination 搭配 AWS SDK 使用
      - DeleteConfigurationSet 搭配 AWS SDK 使用

- DeleteConfigurationSetEventDestination 搭配 AWS SDK 使用
- GetConfigurationSetEventDestinations 搭配 AWS SDK 使用
- ListConfigurationSets 搭配 AWS SDK 使用
- SendVoiceMessage 搭配 AWS SDK 使用
- UpdateConfigurationSetEventDestination 搭配 AWS SDK 使用

使用AWS SDKs Amazon Pinpoint 程式碼範例

下列程式碼範例示範如何使用 Amazon Pinpoint 搭配AWS軟體開發套件 (SDK)。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

案例是向您展示如何呼叫服務中的多個函數或與其他AWS 服務組合來完成特定任務的程式碼範例。

如需AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

Amazon Pinpoint

- 使用AWS SDKs Amazon Pinpoint 基本範例
  - 使用AWS SDKs 的 Amazon Pinpoint 動作
    - CreateApp 搭配AWS SDK 或 CLI 使用
    - CreateCampaign 搭配AWS SDK 使用
    - CreateExportJob 搭配AWS SDK 使用
    - CreateImportJob 搭配AWS SDK 使用
    - CreateSegment 搭配AWS SDK 使用
    - DeleteApp 搭配AWS SDK 或 CLI 使用
    - DeleteEndpoint 搭配AWS SDK 使用
    - GetEndpoint 搭配AWS SDK 或 CLI 使用
    - GetSegments 搭配AWS SDK 使用
    - GetSmsChannel 搭配AWS SDK 或 CLI 使用
    - GetUserEndpoints 搭配AWS SDK 使用
    - SendMessages 搭配AWS SDK 或 CLI 使用
    - UpdateEndpoint 搭配AWS SDK 使用

- 使用AWS SDKs Amazon Pinpoint 案例
  - 推送通知入門

使用AWS SDKs Amazon Pinpoint 基本範例

下列程式碼範例示範如何搭配AWS SDK 使用 Amazon Pinpoint 的基本功能。

範例

- 使用AWS SDKs 的 Amazon Pinpoint 動作
  - CreateApp 搭配AWS SDK 或 CLI 使用
  - CreateCampaign 搭配AWS SDK 使用
  - CreateExportJob 搭配AWS SDK 使用
  - CreateImportJob 搭配AWS SDK 使用
  - CreateSegment 搭配AWS SDK 使用
  - DeleteApp 搭配AWS SDK 或 CLI 使用
  - DeleteEndpoint 搭配AWS SDK 使用
  - GetEndpoint 搭配AWS SDK 或 CLI 使用
  - GetSegments 搭配AWS SDK 使用
  - GetSmsChannel 搭配AWS SDK 或 CLI 使用
  - GetUserEndpoints 搭配AWS SDK 使用
  - SendMessages 搭配AWS SDK 或 CLI 使用
  - UpdateEndpoint 搭配AWS SDK 使用

使用AWS SDKs 的 Amazon Pinpoint 動作

下列程式碼範例示範如何使用AWS SDKs 執行個別 Amazon Pinpoint 動作。每個範例均包含 GitHub 的連結，您可以在連結中找到設定和執行程式碼的相關說明。

這些片段會呼叫 Amazon Pinpoint API，而且是必須在內容中執行的大型程式的程式碼片段。您可以在使用AWS SDKs Amazon Pinpoint 案例 中查看內容中的動作。

下列範例僅包含最常使用的動作。如需完整清單，請參閱 Amazon Pinpoint API 參考。

範例

- CreateApp 搭配AWS SDK 或 CLI 使用

- CreateCampaign 搭配AWS SDK 使用
- CreateExportJob 搭配AWS SDK 使用
- CreateImportJob 搭配AWS SDK 使用
- CreateSegment 搭配AWS SDK 使用
- DeleteApp 搭配AWS SDK 或 CLI 使用
- DeleteEndpoint 搭配AWS SDK 使用
- GetEndpoint 搭配AWS SDK 或 CLI 使用
- GetSegments 搭配AWS SDK 使用
- GetSmsChannel 搭配AWS SDK 或 CLI 使用
- GetUserEndpoints 搭配AWS SDK 使用
- SendMessages 搭配AWS SDK 或 CLI 使用
- UpdateEndpoint 搭配AWS SDK 使用

CreateApp 搭配AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 `CreateApp`。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- 推送通知入門

CLI

AWS CLI

範例 1：建立應用程式

以下 `create-app` 範例會建立新應用程式 (專案)。

```
aws pinpoint create-app \
    --create-application-request Name=ExampleCorp
```

輸出：

```
{
    "ApplicationResponse": {
```

```
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {}
    }
}
```

範例 2：建立已標記的應用程式

下列 create-app 範例會建立新的應用程式 (專案)，並將標籤 (金鑰和值) 與應用程式產生關聯。

```
aws pinpoint create-app \
    --create-application-request Name=ExampleCorp,tags={"Stack"="Test"}
```

輸出：

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {
            "Stack": "Test"
        }
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI命令參考》中的 CreateApp。

Java

SDK for Java 2.x

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CreateAppRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateAppResponse;
import software.amazon.awssdk.services.pinpoint.model.CreateApplicationRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateApp {
    public static void main(String[] args) {
        final String usage = """

                Usage:  <appName>

                Where:
                 appName - The name of the application to create.

               """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }
        String appName = args[0];
        System.out.println("Creating an application with name: " + appName);

        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        String appID = createApplication(pinpoint, appName);
        System.out.println("App ID is: " + appID);
        pinpoint.close();
    }
```

```
    public static String createApplication(PinpointClient pinpoint, String
 appName) {
        try {
            CreateApplicationRequest appRequest =
 CreateApplicationRequest.builder()
                    .name(appName)
                    .build();

            CreateAppRequest request = CreateAppRequest.builder()
                    .createApplicationRequest(appRequest)
                    .build();

            CreateAppResponse result = pinpoint.createApp(request);
            return result.applicationResponse().id();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考中的 CreateApp。

Kotlin

適用於 Kotlin 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

```
suspend fun createApplication(applicationName: String?): String? {
    val createApplicationRequestOb =
        CreateApplicationRequest {
            name = applicationName
        }
```

```
    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val result =
            pinpoint.createApp(
                CreateAppRequest {
                    createApplicationRequest = createApplicationRequestOb
                },
            )
        return result.applicationResponse?.id
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Kotlin API 參考中的 CreateApp。

如需AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

CreateCampaign 搭配AWS SDK 使用

下列程式碼範例示範如何使用 `CreateCampaign`。

Java

適用於 Java 2.x 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

建立行銷活動。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CampaignResponse;
import software.amazon.awssdk.services.pinpoint.model.Message;
import software.amazon.awssdk.services.pinpoint.model.Schedule;
import software.amazon.awssdk.services.pinpoint.model.Action;
import software.amazon.awssdk.services.pinpoint.model.MessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.WriteCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignResponse;
```

```
import software.amazon.awssdk.services.pinpoint.model.CreateCampaignRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateCampaign {
    public static void main(String[] args) {

        final String usage = """

                Usage:   <appId> <segmentId>

                Where:
                  appId - The ID of the application to create the campaign in.
                  segmentId - The ID of the segment to create the campaign from.
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String segmentId = args[1];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        createPinCampaign(pinpoint, appId, segmentId);
        pinpoint.close();
    }

    public static void createPinCampaign(PinpointClient pinpoint, String appId,
 String segmentId) {
        CampaignResponse result = createCampaign(pinpoint, appId, segmentId);
        System.out.println("Campaign " + result.name() + " created.");
        System.out.println(result.description());
```

```
   }

   public static CampaignResponse createCampaign(PinpointClient client, String
appID, String segmentID) {

       try {
           Schedule schedule = Schedule.builder()
                   .startTime("IMMEDIATE")
                   .build();

           Message defaultMessage = Message.builder()
                   .action(Action.OPEN_APP)
                   .body("My message body.")
                   .title("My message title.")
                   .build();

           MessageConfiguration messageConfiguration =
MessageConfiguration.builder()
                   .defaultMessage(defaultMessage)
                   .build();

           WriteCampaignRequest request = WriteCampaignRequest.builder()
                   .description("My description")
                   .schedule(schedule)
                   .name("MyCampaign")
                   .segmentId(segmentID)
                   .messageConfiguration(messageConfiguration)
                   .build();

           CreateCampaignResponse result =
client.createCampaign(CreateCampaignRequest.builder()
                   .applicationId(appID)
                   .writeCampaignRequest(request).build());

           System.out.println("Campaign ID: " + result.campaignResponse().id());
           return result.campaignResponse();

       } catch (PinpointException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }

       return null;
   }
```

```
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考中的 CreateCampaign。

Kotlin

適用於 Kotlin 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

```
suspend fun createPinCampaign(
    appId: String,
    segmentIdVal: String,
) {
    val scheduleOb =
        Schedule {
            startTime = "IMMEDIATE"
        }

    val defaultMessageOb =
        Message {
            action = Action.OpenApp
            body = "My message body"
            title = "My message title"
        }

    val messageConfigurationOb =
        MessageConfiguration {
            defaultMessage = defaultMessageOb
        }

    val writeCampaign =
        WriteCampaignRequest {
            description = "My description"
            schedule = scheduleOb
            name = "MyCampaign"
            segmentId = segmentIdVal
```

```
            messageConfiguration = messageConfigurationOb
        }

    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val result: CreateCampaignResponse =
            pinpoint.createCampaign(
                CreateCampaignRequest {
                    applicationId = appId
                    writeCampaignRequest = writeCampaign
                },
            )
        println("Campaign ID is ${result.campaignResponse?.id}")
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Kotlin API 參考中的 CreateCampaign。

如需AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

`CreateExportJob` 搭配AWS SDK 使用

以下程式碼範例顯示如何使用 `CreateExportJob`。

Java

適用於 Java 2.x 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

匯出端點。

```
import software.amazon.awssdk.core.ResponseBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.ExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.GetExportJobRequest;
import software.amazon.awssdk.services.s3.S3Client;
import software.amazon.awssdk.services.s3.model.GetObjectRequest;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Request;
import software.amazon.awssdk.services.s3.model.ListObjectsV2Response;
import software.amazon.awssdk.services.s3.model.S3Object;
import software.amazon.awssdk.services.s3.model.GetObjectResponse;
import software.amazon.awssdk.services.s3.model.S3Exception;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.concurrent.TimeUnit;
import java.util.stream.Collectors;

/**
 * To run this code example, you need to create an AWS Identity and Access
 * Management (IAM) role with the correct policy as described in this
 * documentation:
 * https://docs.aws.amazon.com/pinpoint/latest/developerguide/audience-data-
export.html
 *
 * Also, set up your development environment, including your credentials.
 *
 * For information, see this documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class ExportEndpoints {
    public static void main(String[] args) {
        final String usage = """

                This program performs the following steps:

                1. Exports the endpoints to an Amazon S3 bucket.
```

```
                2. Downloads the exported endpoints files from Amazon S3.
                3. Parses the endpoints files to obtain the endpoint IDs and
prints them.
                Usage: ExportEndpoints <applicationId> <s3BucketName>
<iamExportRoleArn> <path>

                Where:
                  applicationId - The ID of the Amazon Pinpoint application that
has the endpoint.
                  s3BucketName - The name of the Amazon S3 bucket to export the
JSON file to.\s
                  iamExportRoleArn - The ARN of an IAM role that grants Amazon
Pinpoint write permissions to the S3 bucket.  path - The path where the files
downloaded from the Amazon S3 bucket are written (for example, C:/AWS/).
                """;

       if (args.length != 4) {
           System.out.println(usage);
           System.exit(1);
       }

       String applicationId = args[0];
       String s3BucketName = args[1];
       String iamExportRoleArn = args[2];
       String path = args[3];
       System.out.println("Deleting an application with ID: " + applicationId);

       Region region = Region.US_EAST_1;
       PinpointClient pinpoint = PinpointClient.builder()
               .region(region)
               .build();

       S3Client s3Client = S3Client.builder()
               .region(region)
               .build();

       exportAllEndpoints(pinpoint, s3Client, applicationId, s3BucketName, path,
iamExportRoleArn);
       pinpoint.close();
       s3Client.close();
   }

   public static void exportAllEndpoints(PinpointClient pinpoint,
           S3Client s3Client,
```

```
            String applicationId,
            String s3BucketName,
            String path,
            String iamExportRoleArn) {

        try {
            List<String> objectKeys = exportEndpointsToS3(pinpoint, s3Client,
 s3BucketName, iamExportRoleArn,
                    applicationId);
            List<String> endpointFileKeys = objectKeys.stream().filter(o ->
 o.endsWith(".gz"))
                    .collect(Collectors.toList());
            downloadFromS3(s3Client, path, s3BucketName, endpointFileKeys);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static List<String> exportEndpointsToS3(PinpointClient pinpoint,
 S3Client s3Client, String s3BucketName,
            String iamExportRoleArn, String applicationId) {

        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd-
HH_mm:ss.SSS_z");
        String endpointsKeyPrefix = "exports/" + applicationId + "_" +
 dateFormat.format(new Date());
        String s3UrlPrefix = "s3://" + s3BucketName + "/" + endpointsKeyPrefix +
 "/";
        List<String> objectKeys = new ArrayList<>();
        String key;

        try {
            // Defines the export job that Amazon Pinpoint runs.
            ExportJobRequest jobRequest = ExportJobRequest.builder()
                    .roleArn(iamExportRoleArn)
                    .s3UrlPrefix(s3UrlPrefix)
                    .build();

            CreateExportJobRequest exportJobRequest =
 CreateExportJobRequest.builder()
                    .applicationId(applicationId)
                    .exportJobRequest(jobRequest)
```

```
                    .build();

            System.out.format("Exporting endpoints from Amazon Pinpoint
application %s to Amazon S3 " +
                    "bucket %s . . .\n", applicationId, s3BucketName);

            CreateExportJobResponse exportResult =
pinpoint.createExportJob(exportJobRequest);
            String jobId = exportResult.exportJobResponse().id();
            System.out.println(jobId);
            printExportJobStatus(pinpoint, applicationId, jobId);

            ListObjectsV2Request v2Request = ListObjectsV2Request.builder()
                    .bucket(s3BucketName)
                    .prefix(endpointsKeyPrefix)
                    .build();

            // Create a list of object keys.
            ListObjectsV2Response v2Response = s3Client.listObjectsV2(v2Request);
            List<S3Object> objects = v2Response.contents();
            for (S3Object object : objects) {
                key = object.key();
                objectKeys.add(key);
            }

            return objectKeys;

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    private static void printExportJobStatus(PinpointClient pinpointClient,
            String applicationId,
            String jobId) {

        GetExportJobResponse getExportJobResult;
        String status;

        try {
            // Checks the job status until the job completes or fails.
            GetExportJobRequest exportJobRequest = GetExportJobRequest.builder()
```

```
                    .jobId(jobId)
                    .applicationId(applicationId)
                    .build();

            do {
                getExportJobResult =
pinpointClient.getExportJob(exportJobRequest);
                status =
getExportJobResult.exportJobResponse().jobStatus().toString().toUpperCase();
                System.out.format("Export job %s . . .\n", status);
                TimeUnit.SECONDS.sleep(3);

            } while (!status.equals("COMPLETED") && !status.equals("FAILED"));

            if (status.equals("COMPLETED")) {
                System.out.println("Finished exporting endpoints.");
            } else {
                System.err.println("Failed to export endpoints.");
                System.exit(1);
            }

        } catch (PinpointException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    // Download files from an Amazon S3 bucket and write them to the path
location.
    public static void downloadFromS3(S3Client s3Client, String path, String
s3BucketName, List<String> objectKeys) {

        String newPath;
        try {
            for (String key : objectKeys) {
                GetObjectRequest objectRequest = GetObjectRequest.builder()
                        .bucket(s3BucketName)
                        .key(key)
                        .build();

                ResponseBytes<GetObjectResponse> objectBytes =
s3Client.getObjectAsBytes(objectRequest);
                byte[] data = objectBytes.asByteArray();
```

```
                // Write the data to a local file.
                String fileSuffix = new
 SimpleDateFormat("yyyyMMddHHmmss").format(new Date());
                newPath = path + fileSuffix + ".gz";
                File myFile = new File(newPath);
                OutputStream os = new FileOutputStream(myFile);
                os.write(data);
            }
            System.out.println("Download finished.");

        } catch (S3Exception | NullPointerException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考中的 CreateExportJob。

如需AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

`CreateImportJob` 搭配AWS SDK 使用

以下程式碼範例顯示如何使用 `CreateImportJob`。

Java

適用於 Java 2.x 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

匯入客群。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobRequest;
```

```
import software.amazon.awssdk.services.pinpoint.model.ImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.ImportJobRequest;
import software.amazon.awssdk.services.pinpoint.model.Format;
import software.amazon.awssdk.services.pinpoint.model.CreateImportJobResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ImportSegment {
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appId> <bucket> <key> <roleArn>\s

                Where:
                  appId - The application ID to create a segment for.
                  bucket - The name of the Amazon S3 bucket that contains the
 segment definitons.
                  key - The key of the S3 object.
                  roleArn - ARN of the role that allows Amazon
 Pinpoint to access S3. You need to set trust management for this
 to work. See https://docs.aws.amazon.com/IAM/latest/UserGuide/
reference_policies_elements_principal.html
                  """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String bucket = args[1];
        String key = args[2];
        String roleArn = args[3];

        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
```

```
                .build();

        ImportJobResponse response = createImportSegment(pinpoint, appId, bucket,
 key, roleArn);
        System.out.println("Import job for " + bucket + " submitted.");
        System.out.println("See application " + response.applicationId() + " for
 import job status.");
        System.out.println("See application " + response.jobStatus() + " for
 import job status.");
        pinpoint.close();
    }

    public static ImportJobResponse createImportSegment(PinpointClient client,
            String appId,
            String bucket,
            String key,
            String roleArn) {

        try {
            ImportJobRequest importRequest = ImportJobRequest.builder()
                    .defineSegment(true)
                    .registerEndpoints(true)
                    .roleArn(roleArn)
                    .format(Format.JSON)
                    .s3Url("s3://" + bucket + "/" + key)
                    .build();

            CreateImportJobRequest jobRequest = CreateImportJobRequest.builder()
                    .importJobRequest(importRequest)
                    .applicationId(appId)
                    .build();

            CreateImportJobResponse jobResponse =
 client.createImportJob(jobRequest);
            return jobResponse.importJobResponse();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考中的 CreateImportJob。

如需AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

`CreateSegment` 搭配AWS SDK 使用

下列程式碼範例示範如何使用 `CreateSegment`。

Java

適用於 Java 2.x 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AttributeDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.AttributeType;
import software.amazon.awssdk.services.pinpoint.model.RecencyDimension;
import software.amazon.awssdk.services.pinpoint.model.SegmentBehaviors;
import software.amazon.awssdk.services.pinpoint.model.SegmentDemographics;
import software.amazon.awssdk.services.pinpoint.model.SegmentLocation;
import software.amazon.awssdk.services.pinpoint.model.SegmentDimensions;
import software.amazon.awssdk.services.pinpoint.model.WriteSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentRequest;
import software.amazon.awssdk.services.pinpoint.model.CreateSegmentResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
```

```
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class CreateSegment {
        public static void main(String[] args) {
                final String usage = """

                                Usage:   <appId>

                                Where:
                                  appId - The application ID to create a segment
 for.

                                """;

                if (args.length != 1) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String appId = args[0];
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();

                SegmentResponse result = createSegment(pinpoint, appId);
                System.out.println("Segment " + result.name() + " created.");
                System.out.println(result.segmentType());
                pinpoint.close();
        }

        public static SegmentResponse createSegment(PinpointClient client, String
 appId) {
                try {
                        Map<String, AttributeDimension> segmentAttributes = new
 HashMap<>();
                        segmentAttributes.put("Team",
 AttributeDimension.builder()
                                        .attributeType(AttributeType.INCLUSIVE)
                                        .values("Lakers")
                                        .build());
```

```
                        RecencyDimension recencyDimension =
RecencyDimension.builder()
                                        .duration("DAY_30")
                                        .recencyType("ACTIVE")
                                        .build();

                        SegmentBehaviors segmentBehaviors =
SegmentBehaviors.builder()
                                        .recency(recencyDimension)
                                        .build();

                        SegmentDemographics segmentDemographics =
SegmentDemographics
                                        .builder()
                                        .build();

                        SegmentLocation segmentLocation = SegmentLocation
                                        .builder()
                                        .build();

                        SegmentDimensions dimensions = SegmentDimensions
                                        .builder()
                                        .attributes(segmentAttributes)
                                        .behavior(segmentBehaviors)
                                        .demographic(segmentDemographics)
                                        .location(segmentLocation)
                                        .build();

                        WriteSegmentRequest writeSegmentRequest =
WriteSegmentRequest.builder()
                                        .name("MySegment")
                                        .dimensions(dimensions)
                                        .build();

                        CreateSegmentRequest createSegmentRequest =
CreateSegmentRequest.builder()
                                        .applicationId(appId)
                                        .writeSegmentRequest(writeSegmentRequest)
                                        .build();

                        CreateSegmentResponse createSegmentResult =
client.createSegment(createSegmentRequest);
                        System.out.println("Segment ID: " +
createSegmentResult.segmentResponse().id());
```

```
                        System.out.println("Done");
                        return createSegmentResult.segmentResponse();

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
                return null;
        }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考中的 CreateSegment。

Kotlin

適用於 Kotlin 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

```
suspend fun createPinpointSegment(applicationIdVal: String?): String? {
    val segmentAttributes = mutableMapOf<String, AttributeDimension>()
    val myList = mutableListOf<String>()
    myList.add("Lakers")

    val atts =
        AttributeDimension {
            attributeType = AttributeType.Inclusive
            values = myList
        }

    segmentAttributes["Team"] = atts
    val recencyDimension =
        RecencyDimension {
            duration = Duration.fromValue("DAY_30")
            recencyType = RecencyType.fromValue("ACTIVE")
        }
```

```
    val segmentBehaviors =
        SegmentBehaviors {
            recency = recencyDimension
        }

    val segmentLocation = SegmentLocation {}
    val dimensionsOb =
        SegmentDimensions {
            attributes = segmentAttributes
            behavior = segmentBehaviors
            demographic = SegmentDemographics {}
            location = segmentLocation
        }

    val writeSegmentRequestOb =
        WriteSegmentRequest {
            name = "MySegment101"
            dimensions = dimensionsOb
        }

    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val createSegmentResult: CreateSegmentResponse =
            pinpoint.createSegment(
                CreateSegmentRequest {
                    applicationId = applicationIdVal
                    writeSegmentRequest = writeSegmentRequestOb
                },
            )
        println("Segment ID is ${createSegmentResult.segmentResponse?.id}")
        return createSegmentResult.segmentResponse?.id
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Kotlin API 參考中的 CreateSegment。

如需AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

`DeleteApp` 搭配AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 `DeleteApp`。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- 推送通知入門

CLI

AWS CLI

如欲刪除應用程式

以下 delete-app 範例會刪除應用程式 (專案)。

```
aws pinpoint delete-app \
    --application-id 810c7aab86d42fb2b56c8c966example
```

輸出：

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {}
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI命令參考》中的 DeleteApp。

Java

SDK for Java 2.x

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

刪除應用程式。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DeleteAppRequest;
import software.amazon.awssdk.services.pinpoint.model.DeleteAppResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteApp {
    public static void main(String[] args) {
        final String usage = """

                Usage:  <appId>

                Where:
                 appId - The ID of the application to delete.

                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        System.out.println("Deleting an application with ID: " + appId);
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        deletePinApp(pinpoint, appId);
        System.out.println("Done");
        pinpoint.close();
    }
```

```
    public static void deletePinApp(PinpointClient pinpoint, String appId) {
        try {
            DeleteAppRequest appRequest = DeleteAppRequest.builder()
                    .applicationId(appId)
                    .build();

            DeleteAppResponse result = pinpoint.deleteApp(appRequest);
            String appName = result.applicationResponse().name();
            System.out.println("Application " + appName + " has been deleted.");

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考中的 DeleteApp。

Kotlin

適用於 Kotlin 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

```
suspend fun deletePinApp(appId: String?) {
    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val result =
            pinpoint.deleteApp(
                DeleteAppRequest {
                    applicationId = appId
                },
            )
        val appName = result.applicationResponse?.name
        println("Application $appName has been deleted.")
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Kotlin API 參考中的 DeleteApp。

如需AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

`DeleteEndpoint` 搭配AWS SDK 使用

下列程式碼範例示範如何使用 `DeleteEndpoint`。

Java

適用於 Java 2.x 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

刪除端點。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.DeleteEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteEndpoint {
    public static void main(String[] args) {
        final String usage = """
```

```
                Usage:   <appName> <endpointId >

                Where:
                  appId - The id of the application to delete.
                  endpointId - The id of the endpoint to delete.
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String endpointId = args[1];
        System.out.println("Deleting an endpoint with id: " + endpointId);
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        deletePinEncpoint(pinpoint, appId, endpointId);
        pinpoint.close();
    }

    public static void deletePinEncpoint(PinpointClient pinpoint, String appId,
 String endpointId) {
        try {
            DeleteEndpointRequest appRequest = DeleteEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpointId)
                    .build();

            DeleteEndpointResponse result = pinpoint.deleteEndpoint(appRequest);
            String id = result.endpointResponse().id();
            System.out.println("The deleted endpoint id  " + id);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考中的 DeleteEndpoint。

Kotlin

適用於 Kotlin 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

```
suspend fun deletePinEncpoint(
    appIdVal: String?,
    endpointIdVal: String?,
) {
    val deleteEndpointRequest =
        DeleteEndpointRequest {
            applicationId = appIdVal
            endpointId = endpointIdVal
        }

    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val result = pinpoint.deleteEndpoint(deleteEndpointRequest)
        val id = result.endpointResponse?.id
        println("The deleted endpoint is  $id")
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Kotlin API 參考中的 DeleteEndpoint。

如需AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

`GetEndpoint` 搭配AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 `GetEndpoint`。

CLI

AWS CLI

如要擷取應用程式之特定端點的設定和屬性相關資訊

下列 get-endpoint 範例會擷取應用程式之特定端點的設定和屬性相關資訊。

```
aws pinpoint get-endpoint \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --endpoint-id testendpoint \
    --region us-east-1
```

輸出：

```
{
    "EndpointResponse": {
        "Address": "+11234567890",
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "Attributes": {},
        "ChannelType": "SMS",
        "CohortId": "63",
        "CreationDate": "2019-01-28T23:55:11.534Z",
        "EffectiveDate": "2021-08-06T00:04:51.763Z",
        "EndpointStatus": "ACTIVE",
        "Id": "testendpoint",
        "Location": {
            "Country": "USA"
        },
        "Metrics": {
            "SmsDelivered": 1.0
        },
        "OptOut": "ALL",
        "RequestId": "a204b1f2-7e26-48a7-9c80-b49a2143489d",
        "User": {
            "UserAttributes": {
                "Age": [
                    "24"
                ]
            },
        "UserId": "testuser"
        }
    }
```

```
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI命令參考》中的 GetEndpoint。

Java

SDK for Java 2.x

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

```
import com.google.gson.FieldNamingPolicy;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class LookUpEndpoint {
    public static void main(String[] args) {
        final String usage = """

                Usage:   <appId> <endpoint>

                Where:
                  appId - The ID of the application to delete.
                  endpoint - The ID of the endpoint.\s
```

```
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        String endpoint = args[1];
        System.out.println("Looking up an endpoint point with ID: " + endpoint);
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        lookupPinpointEndpoint(pinpoint, appId, endpoint);
        pinpoint.close();
    }

    public static void lookupPinpointEndpoint(PinpointClient pinpoint, String
appId, String endpoint) {
        try {
            GetEndpointRequest appRequest = GetEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpoint)
                    .build();

            GetEndpointResponse result = pinpoint.getEndpoint(appRequest);
            EndpointResponse endResponse = result.endpointResponse();

            // Uses the Google Gson library to pretty print the endpoint JSON.
            Gson gson = new GsonBuilder()
                    .setFieldNamingPolicy(FieldNamingPolicy.UPPER_CAMEL_CASE)
                    .setPrettyPrinting()
                    .create();

            String endpointJson = gson.toJson(endResponse);
            System.out.println(endpointJson);

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        System.out.println("Done");
    }
```

```
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考中的 GetEndpoint。

Kotlin

適用於 Kotlin 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

```
suspend fun lookupPinpointEndpoint(
    appId: String?,
    endpoint: String?,
) {
    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val result =
            pinpoint.getEndpoint(
                GetEndpointRequest {
                    applicationId = appId
                    endpointId = endpoint
                },
            )
        val endResponse = result.endpointResponse

        // Uses the Google Gson library to pretty print the endpoint JSON.
        val gson: com.google.gson.Gson =
            GsonBuilder()
                .setFieldNamingPolicy(FieldNamingPolicy.UPPER_CAMEL_CASE)
                .setPrettyPrinting()
                .create()

        val endpointJson: String = gson.toJson(endResponse)
        println(endpointJson)
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Kotlin API 參考中的 GetEndpoint。

如需AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

`GetSegments` 搭配AWS SDK 使用

下列程式碼範例示範如何使用 `GetSegments`。

Java

適用於 Java 2.x 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

列出客群。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.GetSegmentsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetSegmentsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.SegmentResponse;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListSegments {
    public static void main(String[] args) {
        final String usage = """
```

```
                Usage:   <appId>

                Where:
                  appId - The ID of the application that contains a segment.

                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listSegs(pinpoint, appId);
        pinpoint.close();
    }

    public static void listSegs(PinpointClient pinpoint, String appId) {
        try {
            GetSegmentsRequest request = GetSegmentsRequest.builder()
                    .applicationId(appId)
                    .build();

            GetSegmentsResponse response = pinpoint.getSegments(request);
            List<SegmentResponse> segments = response.segmentsResponse().item();
            for (SegmentResponse segment : segments) {
                System.out
                        .println("Segement " + segment.id() + " " +
 segment.name() + " " + segment.lastModifiedDate());
            }

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考中的 GetSegments。

Kotlin

適用於 Kotlin 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

```
suspend fun listSegs(appId: String?) {
    PinpointClient.fromEnvironment { region = "us-west-2" }.use { pinpoint ->
        val response =
            pinpoint.getSegments(
                GetSegmentsRequest {
                    applicationId = appId
                },
            )
        response.segmentsResponse?.item?.forEach { segment ->
            println("Segement id is ${segment.id}")
        }
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Kotlin API 參考中的 GetSegments。

如需AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

`GetSmsChannel` 搭配AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 `GetSmsChannel`。

CLI

AWS CLI

如要擷取應用程式 SMS 管道的狀態和設定相關資訊

下列 `get-sms-channel` 範例會擷取應用程式 SMS 管道的狀態和設定。

```
aws pinpoint get-sms-channel \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

輸出：

```
{
    "SMSChannelResponse": {
        "ApplicationId": "6e0b7591a90841d2b5d93fa11143e5a7",
        "CreationDate": "2019-10-08T18:39:18.511Z",
        "Enabled": true,
        "Id": "sms",
        "IsArchived": false,
        "LastModifiedDate": "2019-10-08T18:39:18.511Z",
        "Platform": "SMS",
        "PromotionalMessagesPerSecond": 20,
        "TransactionalMessagesPerSecond": 20,
        "Version": 1
    }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI命令參考》中的 GetSmsChannel。

Java

SDK for Java 2.x

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.SMSChannelResponse;
import software.amazon.awssdk.services.pinpoint.model.GetSmsChannelRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.SMSChannelRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateSmsChannelRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateSmsChannelResponse;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class UpdateChannel {
    public static void main(String[] args) {
        final String usage = """

                Usage: CreateChannel <appId>

                Where:
                  appId - The name of the application whose channel is updated.

                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        SMSChannelResponse getResponse = getSMSChannel(pinpoint, appId);
        toggleSmsChannel(pinpoint, appId, getResponse);
        pinpoint.close();
    }

    private static SMSChannelResponse getSMSChannel(PinpointClient client, String
 appId) {
        try {
            GetSmsChannelRequest request = GetSmsChannelRequest.builder()
                    .applicationId(appId)
                    .build();
```

```
            SMSChannelResponse response =
 client.getSmsChannel(request).smsChannelResponse();
            System.out.println("Channel state is " + response.enabled());
            return response;

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    private static void toggleSmsChannel(PinpointClient client, String appId,
 SMSChannelResponse getResponse) {
        boolean enabled = !getResponse.enabled();
        try {
            SMSChannelRequest request = SMSChannelRequest.builder()
                    .enabled(enabled)
                    .build();

            UpdateSmsChannelRequest updateRequest =
 UpdateSmsChannelRequest.builder()
                    .smsChannelRequest(request)
                    .applicationId(appId)
                    .build();

            UpdateSmsChannelResponse result =
 client.updateSmsChannel(updateRequest);
            System.out.println("Channel state: " +
 result.smsChannelResponse().enabled());

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考中的 GetSmsChannel。

如需AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

`GetUserEndpoints` 搭配AWS SDK 使用

以下程式碼範例顯示如何使用 `GetUserEndpoints`。

Java

適用於 Java 2.x 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsRequest;
import software.amazon.awssdk.services.pinpoint.model.GetUserEndpointsResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListEndpointIds {
    public static void main(String[] args) {
        final String usage = """

                Usage:    <applicationId> <userId>

                Where:
                   applicationId - The ID of the Amazon Pinpoint application that
 has the endpoint.
                   userId - The user id applicable to the endpoints""";

        if (args.length != 2) {
```

```
            System.out.println(usage);
            System.exit(1);
        }

        String applicationId = args[0];
        String userId = args[1];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listAllEndpoints(pinpoint, applicationId, userId);
        pinpoint.close();
    }

    public static void listAllEndpoints(PinpointClient pinpoint,
            String applicationId,
            String userId) {

        try {
            GetUserEndpointsRequest endpointsRequest =
 GetUserEndpointsRequest.builder()
                    .userId(userId)
                    .applicationId(applicationId)
                    .build();

            GetUserEndpointsResponse response =
 pinpoint.getUserEndpoints(endpointsRequest);
            List<EndpointResponse> endpoints =
 response.endpointsResponse().item();

            // Display the results.
            for (EndpointResponse endpoint : endpoints) {
                System.out.println("The channel type is: " +
 endpoint.channelType());
                System.out.println("The address is  " + endpoint.address());
            }

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考中的 GetUserEndpoints。

如需AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

SendMessages 搭配AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 SendMessages。

動作範例是大型程式的程式碼摘錄，必須在內容中執行。您可以在下列程式碼範例的內容中看到此動作：

- 推送通知入門

.NET

適用於 .NET 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

傳送電子郵件訊息。

```
using Amazon;
using Amazon.Pinpoint;
using Amazon.Pinpoint.Model;
using Microsoft.Extensions.Configuration;

namespace SendEmailMessage;

public class SendEmailMainClass
{
    public static async Task Main(string[] args)
    {
        var configuration = new ConfigurationBuilder()
```

```
        .SetBasePath(Directory.GetCurrentDirectory())
        .AddJsonFile("settings.json") // Load test settings from .json file.
        .AddJsonFile("settings.local.json",
            true) // Optionally load local settings.
        .Build();

        // The AWS Region that you want to use to send the email. For a list of
        // AWS Regions where the Amazon Pinpoint API is available, see
        // https://docs.aws.amazon.com/pinpoint/latest/apireference/
        string region = "us-east-1";

        // The "From" address. This address has to be verified in Amazon
Pinpoint
        // in the region you're using to send email.
        string senderAddress = configuration["SenderAddress"]!;

        // The address on the "To" line. If your Amazon Pinpoint account is in
        // the sandbox, this address also has to be verified.
        string toAddress = configuration["ToAddress"]!;

        // The Amazon Pinpoint project/application ID to use when you send this
message.
        // Make sure that the SMS channel is enabled for the project or
application
        // that you choose.
        string appId = configuration["AppId"]!;

        try
        {
            await SendEmailMessage(region, appId, toAddress, senderAddress);
        }
        catch (Exception ex)
        {
            Console.WriteLine("The message wasn't sent. Error message: " +
ex.Message);
        }
    }

    public static async Task<MessageResponse> SendEmailMessage(
        string region, string appId, string toAddress, string senderAddress)
    {
        var client = new
AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));
```

```
        // The subject line of the email.
        string subject = "Amazon Pinpoint Email test";

        // The body of the email for recipients whose email clients don't
        // support HTML content.
        string textBody = @"Amazon Pinpoint Email Test (.NET)"
                          + "\n---------------------------------"
                          + "\nThis email was sent using the Amazon Pinpoint API
 using the AWS SDK for .NET.";

        // The body of the email for recipients whose email clients support
        // HTML content.
        string htmlBody = @"<html>"
                          + "\n<head></head>"
                          + "\n<body>"
                          + "\n  <h1>Amazon Pinpoint Email Test (AWS SDK
 for .NET)</h1>"
                          + "\n  <p>This email was sent using the "
                          + "\n    <a href='https://aws.amazon.com/
pinpoint/'>Amazon Pinpoint</a> API "
                          + "\n    using the <a href='https://aws.amazon.com/sdk-
for-net/'>AWS SDK for .NET</a>"
                          + "\n  </p>"
                          + "\n</body>"
                          + "\n</html>";

        // The character encoding the you want to use for the subject line and
        // message body of the email.
        string charset = "UTF-8";

        var sendRequest = new SendMessagesRequest
        {
            ApplicationId = appId,
            MessageRequest = new MessageRequest
            {
                Addresses = new Dictionary<string, AddressConfiguration>
                {
                    {
                        toAddress,
                        new AddressConfiguration
                        {
                            ChannelType = ChannelType.EMAIL
                        }
                    }
```

```
                },
                MessageConfiguration = new DirectMessageConfiguration
                {
                    EmailMessage = new EmailMessage
                    {
                        FromAddress = senderAddress,
                        SimpleEmail = new SimpleEmail
                        {
                            HtmlPart = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = htmlBody
                            },
                            TextPart = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = textBody
                            },
                            Subject = new SimpleEmailPart
                            {
                                Charset = charset,
                                Data = subject
                            }
                        }
                    }
                }
            }
        };
        Console.WriteLine("Sending message...");
        SendMessagesResponse response = await
 client.SendMessagesAsync(sendRequest);
        Console.WriteLine("Message sent!");
        return response.MessageResponse;
    }
}
```

傳送一則 SMS 訊息。

```
using Amazon;
using Amazon.Pinpoint;
```

```
using Amazon.Pinpoint.Model;
using Microsoft.Extensions.Configuration;

namespace SendSmsMessage;

public class SendSmsMessageMainClass
{
    public static async Task Main(string[] args)
    {
        var configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load test settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally load local settings.
            .Build();

        // The AWS Region that you want to use to send the message. For a list of
        // AWS Regions where the Amazon Pinpoint API is available, see
        // https://docs.aws.amazon.com/pinpoint/latest/apireference/
        string region = "us-east-1";

        // The phone number or short code to send the message from. The phone
 number
        // or short code that you specify has to be associated with your Amazon
 Pinpoint
        // account. For best results, specify long codes in E.164 format.
        string originationNumber = configuration["OriginationNumber"]!;

        // The recipient's phone number.  For best results, you should specify
 the
        // phone number in E.164 format.
        string destinationNumber = configuration["DestinationNumber"]!;

        // The Pinpoint project/ application ID to use when you send this
 message.
        // Make sure that the SMS channel is enabled for the project or
 application
        // that you choose.
        string appId = configuration["AppId"]!;

        // The type of SMS message that you want to send. If you plan to send
        // time-sensitive content, specify TRANSACTIONAL. If you plan to send
        // marketing-related content, specify PROMOTIONAL.
        MessageType messageType = MessageType.TRANSACTIONAL;
```

```
        // The registered keyword associated with the originating short code.
        string? registeredKeyword = configuration["RegisteredKeyword"];

        // The sender ID to use when sending the message. Support for sender ID
        // varies by country or region. For more information, see
        // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
        string? senderId = configuration["SenderId"];

        try
        {
            var response = await SendSmsMessage(region, appId, destinationNumber,
                originationNumber, registeredKeyword, senderId, messageType);
            Console.WriteLine($"Message sent to
 {response.MessageResponse.Result.Count} recipient(s).");
            foreach (var messageResultValue in
                     response.MessageResponse.Result.Select(r => r.Value))
            {
                Console.WriteLine($"{messageResultValue.MessageId} Status:
 {messageResultValue.DeliveryStatus}");
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine("The message wasn't sent. Error message: " +
 ex.Message);
        }
    }

    public static async Task<SendMessagesResponse> SendSmsMessage(
        string region, string appId, string destinationNumber, string
 originationNumber,
        string? keyword, string? senderId, MessageType messageType)
    {

        // The content of the SMS message.
        string message = "This message was sent through Amazon Pinpoint using" +
                         " the AWS SDK for .NET. Reply STOP to opt out.";


        var client = new
 AmazonPinpointClient(RegionEndpoint.GetBySystemName(region));
```

```
        SendMessagesRequest sendRequest = new SendMessagesRequest
        {
            ApplicationId = appId,
            MessageRequest = new MessageRequest
            {
                Addresses =
                    new Dictionary<string, AddressConfiguration>
                    {
                        {
                            destinationNumber,
                            new AddressConfiguration { ChannelType =
 ChannelType.SMS }
                        }
                    },
                MessageConfiguration = new DirectMessageConfiguration
                {
                    SMSMessage = new SMSMessage
                    {
                        Body = message,
                        MessageType = MessageType.TRANSACTIONAL,
                        OriginationNumber = originationNumber,
                        SenderId = senderId,
                        Keyword = keyword
                    }
                }
            }
        };
        SendMessagesResponse response = await
 client.SendMessagesAsync(sendRequest);
        return response;
    }
}
```

- 如需 API 詳細資訊，請參閱 適用於 .NET 的 AWS SDK API 參考中的 SendMessages。

CLI

AWS CLI

如要使用應用程式的端點傳送 SMS 訊息

下列 send-messages 範例會針對具有端點的應用程式傳送直接訊息。

```
aws pinpoint send-messages \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --message-request file://myfile.json \
    --region us-west-2
```

myfile.json 的內容：

```
{
    "MessageConfiguration": {
        "SMSMessage": {
            "Body": "hello, how are you?"
        }
    },
    "Endpoints": {
        "testendpoint": {}
    }
}
```

輸出：

```
{
    "MessageResponse": {
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "EndpointResult": {
            "testendpoint": {
                "Address": "+12345678900",
                "DeliveryStatus": "SUCCESSFUL",
                "MessageId": "itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0",
                "StatusCode": 200,
                "StatusMessage": "MessageId:
 itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0"
            }
        },
        "RequestId": "c7e23264-04b2-4a46-b800-d24923f74753"
    }
}
```

詳情請參閱《Amazon Pinpoint SMS 使用者指南》中的 Amazon Pinpoint SMS 管道。

- 如需 API 詳細資訊，請參閱《AWS CLI命令參考》中的 SendMessages。

Java

SDK for Java 2.x

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

傳送電子郵件訊息。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmailPart;
import software.amazon.awssdk.services.pinpoint.model.SimpleEmail;
import software.amazon.awssdk.services.pinpoint.model.EmailMessage;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpointemail.PinpointEmailClient;
import software.amazon.awssdk.services.pinpointemail.model.Body;
import software.amazon.awssdk.services.pinpointemail.model.Content;
import software.amazon.awssdk.services.pinpointemail.model.Destination;
import software.amazon.awssdk.services.pinpointemail.model.EmailContent;
import software.amazon.awssdk.services.pinpointemail.model.Message;
import software.amazon.awssdk.services.pinpointemail.model.SendEmailRequest;

import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
```

```
public class SendEmailMessage {

        // The character encoding the you want to use for the subject line and
        // message body of the email.
        public static String charset = "UTF-8";

    // The body of the email for recipients whose email clients support HTML
 content.
    static final String body = """
        Amazon Pinpoint test (AWS SDK for Java 2.x)

        This email was sent through the Amazon Pinpoint Email API using the AWS
 SDK for Java 2.x

        """;

        public static void main(String[] args) {
                final String usage = """

                                Usage:    <subject> <appId> <senderAddress>
 <toAddress>

            Where:
               subject - The email subject to use.
               senderAddress - The from address. This address has to be verified
 in Amazon Pinpoint in the region you're using to send email\s
               toAddress - The to address. This address has to be verified in
 Amazon Pinpoint in the region you're using to send email\s
            """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String subject = args[0];
        String senderAddress = args[1];
        String toAddress = args[2];
        System.out.println("Sending a message");
        PinpointEmailClient pinpoint = PinpointEmailClient.builder()
            .region(Region.US_EAST_1)
            .build();

        sendEmail(pinpoint, subject, senderAddress, toAddress);
```

```
        System.out.println("Email was sent");
        pinpoint.close();
    }

    public static void sendEmail(PinpointEmailClient pinpointEmailClient, String
 subject, String senderAddress, String toAddress) {
        try {
            Content content = Content.builder()
                .data(body)
                .build();

            Body messageBody = Body.builder()
                .text(content)
                .build();

            Message message = Message.builder()
                .body(messageBody)
                .subject(Content.builder().data(subject).build())
                .build();

            Destination destination = Destination.builder()
                .toAddresses(toAddress)
                .build();

            EmailContent emailContent = EmailContent.builder()
                .simple(message)
                .build();

            SendEmailRequest sendEmailRequest = SendEmailRequest.builder()
                .fromEmailAddress(senderAddress)
                .destination(destination)
                .content(emailContent)
                .build();

            pinpointEmailClient.sendEmail(sendEmailRequest);
            System.out.println("Message Sent");

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

傳送帶 CC 值的電子郵件。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import  software.amazon.awssdk.services.pinpointemail.PinpointEmailClient;
import software.amazon.awssdk.services.pinpointemail.model.Body;
import software.amazon.awssdk.services.pinpointemail.model.Content;
import software.amazon.awssdk.services.pinpointemail.model.Destination;
import software.amazon.awssdk.services.pinpointemail.model.EmailContent;
import software.amazon.awssdk.services.pinpointemail.model.Message;
import software.amazon.awssdk.services.pinpointemail.model.SendEmailRequest;
import java.util.ArrayList;

/**
 * Before running this Java V2 code example, set up your development environment,
 including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendEmailMessageCC {

    // The body of the email.
    static final String body = """
        Amazon Pinpoint test (AWS SDK for Java 2.x)

        This email was sent through the Amazon Pinpoint Email API using the AWS
 SDK for Java 2.x

        """;
    public static void main(String[] args) {
        final String usage = """

            Usage:    <subject> <senderAddress> <toAddress> <ccAddress>

            Where:
               subject - The email subject to use.
               senderAddress - The from address. This address has to be verified
 in Amazon Pinpoint in the region you're using to send email\s
```

```
                toAddress - The to address. This address has to be verified in
Amazon Pinpoint in the region you're using to send email\s
                ccAddress - The CC address.
            """;

        if (args.length != 4) {
            System.out.println(usage);
            System.exit(1);
        }

        String subject = args[0];
        String senderAddress = args[1];
        String toAddress = args[2];
        String ccAddress = args[3];

        System.out.println("Sending a message");
        PinpointEmailClient pinpoint = PinpointEmailClient.builder()
            .region(Region.US_EAST_1)
            .build();

        ArrayList<String> ccList = new ArrayList<>();
        ccList.add(ccAddress);
        sendEmail(pinpoint, subject, senderAddress, toAddress, ccList);
        pinpoint.close();
    }

    public static void sendEmail(PinpointEmailClient pinpointEmailClient, String
subject, String senderAddress, String toAddress, ArrayList<String> ccAddresses)
{
        try {
            Content content = Content.builder()
                .data(body)
                .build();

            Body messageBody = Body.builder()
                .text(content)
                .build();

            Message message = Message.builder()
                .body(messageBody)
                .subject(Content.builder().data(subject).build())
                .build();

            Destination destination = Destination.builder()
```

```
                .toAddresses(toAddress)
                .ccAddresses(ccAddresses)
                .build();

            EmailContent emailContent = EmailContent.builder()
                .simple(message)
                .build();

            SendEmailRequest sendEmailRequest = SendEmailRequest.builder()
                .fromEmailAddress(senderAddress)
                .destination(destination)
                .content(emailContent)
                .build();

            pinpointEmailClient.sendEmail(sendEmailRequest);
            System.out.println("Message Sent");

        } catch (PinpointException e) {
            // Handle exception
            e.printStackTrace();
        }
    }
}
```

傳送一則 SMS 訊息。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
```

```
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendMessage {

        // The type of SMS message that you want to send. If you plan to send
        // time-sensitive content, specify TRANSACTIONAL. If you plan to send
        // marketing-related content, specify PROMOTIONAL.
        public static String messageType = "TRANSACTIONAL";

        // The registered keyword associated with the originating short code.
        public static String registeredKeyword = "myKeyword";

        // The sender ID to use when sending the message. Support for sender ID
        // varies by country or region. For more information, see
        // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
        public static String senderId = "MySenderID";

        public static void main(String[] args) {
                final String usage = """

                                Usage:   <message> <appId> <originationNumber>
 <destinationNumber>\s

                                Where:
                                  message - The body of the message to send.
                                  appId - The Amazon Pinpoint project/application
 ID to use when you send this message.
                                  originationNumber - The phone number or
 short code that you specify has to be associated with your Amazon Pinpoint
 account. For best results, specify long codes in E.164 format (for example,
 +1-555-555-5654).
                                  destinationNumber - The recipient's phone
 number.  For best results, you should specify the phone number in E.164 format
 (for example, +1-555-555-5654).\s
                                  """;

                if (args.length != 4) {
                        System.out.println(usage);
```

```
                        System.exit(1);
                }

                String message = args[0];
                String appId = args[1];
                String originationNumber = args[2];
                String destinationNumber = args[3];
                System.out.println("Sending a message");
                PinpointClient pinpoint = PinpointClient.builder()
                                .region(Region.US_EAST_1)
                                .build();

                sendSMSMessage(pinpoint, message, appId, originationNumber,
destinationNumber);
                pinpoint.close();
        }

        public static void sendSMSMessage(PinpointClient pinpoint, String
message, String appId,
                        String originationNumber,
                        String destinationNumber) {
                try {
                        Map<String, AddressConfiguration> addressMap = new
HashMap<String, AddressConfiguration>();
                        AddressConfiguration addConfig =
AddressConfiguration.builder()
                                        .channelType(ChannelType.SMS)
                                        .build();

                        addressMap.put(destinationNumber, addConfig);
                        SMSMessage smsMessage = SMSMessage.builder()
                                        .body(message)
                                        .messageType(messageType)
                                        .originationNumber(originationNumber)
                                        .senderId(senderId)
                                        .keyword(registeredKeyword)
                                        .build();

                        // Create a DirectMessageConfiguration object.
                        DirectMessageConfiguration direct =
DirectMessageConfiguration.builder()
                                        .smsMessage(smsMessage)
                                        .build();
```

```
                        MessageRequest msgReq = MessageRequest.builder()
                                        .addresses(addressMap)
                                        .messageConfiguration(direct)
                                        .build();

                        // create a SendMessagesRequest object
                        SendMessagesRequest request =
 SendMessagesRequest.builder()
                                        .applicationId(appId)
                                        .messageRequest(msgReq)
                                        .build();

                        SendMessagesResponse response =
 pinpoint.sendMessages(request);
                        MessageResponse msg1 = response.messageResponse();
                        Map map1 = msg1.result();

                        // Write out the result of sendMessage.
                        map1.forEach((k, v) -> System.out.println((k + ":" +
 v)));

                } catch (PinpointException e) {
                        System.err.println(e.awsErrorDetails().errorMessage());
                        System.exit(1);
                }
        }
}
```

傳送批次 SMS 訊息。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.DirectMessageConfiguration;
import software.amazon.awssdk.services.pinpoint.model.SMSMessage;
import software.amazon.awssdk.services.pinpoint.model.AddressConfiguration;
import software.amazon.awssdk.services.pinpoint.model.ChannelType;
import software.amazon.awssdk.services.pinpoint.model.MessageRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesRequest;
import software.amazon.awssdk.services.pinpoint.model.SendMessagesResponse;
import software.amazon.awssdk.services.pinpoint.model.MessageResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
```

```
import java.util.HashMap;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 * <p>
 * For more information, see the following documentation topic:
 * <p>
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendMessageBatch {

    // The type of SMS message that you want to send. If you plan to send
    // time-sensitive content, specify TRANSACTIONAL. If you plan to send
    // marketing-related content, specify PROMOTIONAL.
    public static String messageType = "TRANSACTIONAL";

    // The registered keyword associated with the originating short code.
    public static String registeredKeyword = "myKeyword";

    // The sender ID to use when sending the message. Support for sender ID
    // varies by country or region. For more information, see
    // https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
    public static String senderId = "MySenderID";

    public static void main(String[] args) {
        final String usage = """

                Usage:   <message> <appId> <originationNumber>
 <destinationNumber> <destinationNumber1>\s

                Where:
                  message - The body of the message to send.
                  appId - The Amazon Pinpoint project/application ID to use when
 you send this message.
                  originationNumber - The phone number or short code that
 you specify has to be associated with your Amazon Pinpoint account. For best
 results, specify long codes in E.164 format (for example, +1-555-555-5654).
```

```
                destinationNumber - The recipient's phone number.  For best
results, you should specify the phone number in E.164 format (for example,
+1-555-555-5654).
                destinationNumber1 - The second recipient's phone number.  For
best results, you should specify the phone number in E.164 format (for example,
+1-555-555-5654).\s
               """;

       if (args.length != 5) {
           System.out.println(usage);
           System.exit(1);
       }

       String message = args[0];
       String appId = args[1];
       String originationNumber = args[2];
       String destinationNumber = args[3];
       String destinationNumber1 = args[4];
       System.out.println("Sending a message");
       PinpointClient pinpoint = PinpointClient.builder()
               .region(Region.US_EAST_1)
               .build();

       sendSMSMessage(pinpoint, message, appId, originationNumber,
destinationNumber, destinationNumber1);
       pinpoint.close();
   }

   public static void sendSMSMessage(PinpointClient pinpoint, String message,
String appId,
                                     String originationNumber,
                                     String destinationNumber, String
destinationNumber1) {
       try {
           Map<String, AddressConfiguration> addressMap = new HashMap<String,
AddressConfiguration>();
           AddressConfiguration addConfig = AddressConfiguration.builder()
                   .channelType(ChannelType.SMS)
                   .build();

           // Add an entry to the Map object for each number to whom you want to
send a
           // message.
           addressMap.put(destinationNumber, addConfig);
```

```
            addressMap.put(destinationNumber1, addConfig);
            SMSMessage smsMessage = SMSMessage.builder()
                    .body(message)
                    .messageType(messageType)
                    .originationNumber(originationNumber)
                    .senderId(senderId)
                    .keyword(registeredKeyword)
                    .build();

            // Create a DirectMessageConfiguration object.
            DirectMessageConfiguration direct =
 DirectMessageConfiguration.builder()
                    .smsMessage(smsMessage)
                    .build();

            MessageRequest msgReq = MessageRequest.builder()
                    .addresses(addressMap)
                    .messageConfiguration(direct)
                    .build();

            // Create a SendMessagesRequest object.
            SendMessagesRequest request = SendMessagesRequest.builder()
                    .applicationId(appId)
                    .messageRequest(msgReq)
                    .build();

            SendMessagesResponse response = pinpoint.sendMessages(request);
            MessageResponse msg1 = response.messageResponse();
            Map map1 = msg1.result();

            // Write out the result of sendMessage.
            map1.forEach((k, v) -> System.out.println((k + ":" + v)));

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考中的 SendMessages。

JavaScript

適用於 JavaScript (v3) 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

在單獨的模組中建立用戶端並將其匯出。

```
import { PinpointClient } from "@aws-sdk/client-pinpoint";
// Set the AWS Region.
const REGION = "us-east-1";
export const pinClient = new PinpointClient({ region: REGION });
```

傳送電子郵件訊息。

```
// Import required AWS SDK clients and commands for Node.js
import { SendMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "./libs/pinClient.js";

// The FromAddress must be verified in SES.
const fromAddress = "FROM_ADDRESS";
const toAddress = "TO_ADDRESS";
const projectId = "PINPOINT_PROJECT_ID";

// The subject line of the email.
const subject = "Amazon Pinpoint Test (AWS SDK for JavaScript in Node.js)";

// The email body for recipients with non-HTML email clients.
const body_text = `Amazon Pinpoint Test (SDK for JavaScript in Node.js)
----------------------------------------------------
This email was sent with Amazon Pinpoint using the AWS SDK for JavaScript in
 Node.js.
For more information, see https://aws.amazon.com/sdk-for-node-js/`;

// The body of the email for recipients whose email clients support HTML content.
const body_html = `<html>
<head></head>
```

```
<body>
  <h1>Amazon Pinpoint Test (SDK for JavaScript in Node.js)</h1>
  <p>This email was sent with
    <a href='https://aws.amazon.com/pinpoint/'>the Amazon Pinpoint Email API</a>
 using the
    <a href='https://aws.amazon.com/sdk-for-node-js/'>
      AWS SDK for JavaScript in Node.js</a>.</p>
</body>
</html>`;

// The character encoding for the subject line and message body of the email.
const charset = "UTF-8";

const params = {
  ApplicationId: projectId,
  MessageRequest: {
    Addresses: {
      [toAddress]: {
        ChannelType: "EMAIL",
      },
    },
    MessageConfiguration: {
      EmailMessage: {
        FromAddress: fromAddress,
        SimpleEmail: {
          Subject: {
            Charset: charset,
            Data: subject,
          },
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
          TextPart: {
            Charset: charset,
            Data: body_text,
          },
        },
      },
    },
  },
};

const run = async () => {
```

```
  try {
    const { MessageResponse } = await pinClient.send(
      new SendMessagesCommand(params),
    );

    if (!MessageResponse) {
      throw new Error("No message response.");
    }

    if (!MessageResponse.Result) {
      throw new Error("No message result.");
    }

    const recipientResult = MessageResponse.Result[toAddress];

    if (recipientResult.StatusCode !== 200) {
      throw new Error(recipientResult.StatusMessage);
    }
    console.log(recipientResult.MessageId);
  } catch (err) {
    console.log(err.message);
  }
};

run();
```

傳送一則 SMS 訊息。

```
// Import required AWS SDK clients and commands for Node.js
import { SendMessagesCommand } from "@aws-sdk/client-pinpoint";
import { pinClient } from "./libs/pinClient.js";

/* The phone number or short code to send the message from. The phone number
 or short code that you specify has to be associated with your Amazon Pinpoint
account. For best results, specify long codes in E.164 format. */
const originationNumber = "SENDER_NUMBER"; //e.g., +1XXXXXXXXXX

// The recipient's phone number.  For best results, you should specify the phone
 number in E.164 format.
const destinationNumber = "RECEIVER_NUMBER"; //e.g., +1XXXXXXXXXX
```

```
// The content of the SMS message.
const message =
  "This message was sent through Amazon Pinpoint " +
  "using the AWS SDK for JavaScript in Node.js. Reply STOP to " +
  "opt out.";

/*The Amazon Pinpoint project/application ID to use when you send this message.
Make sure that the SMS channel is enabled for the project or application
that you choose.*/
const projectId = "PINPOINT_PROJECT_ID"; //e.g., XXXXXXXX66e4e9986478cXXXXXXXXX

/* The type of SMS message that you want to send. If you plan to send
time-sensitive content, specify TRANSACTIONAL. If you plan to send
marketing-related content, specify PROMOTIONAL.*/
const messageType = "TRANSACTIONAL";

// The registered keyword associated with the originating short code.
const registeredKeyword = "myKeyword";

/* The sender ID to use when sending the message. Support for sender ID
// varies by country or region. For more information, see
https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html.*/

const senderId = "MySenderID";

// Specify the parameters to pass to the API.
const params = {
  ApplicationId: projectId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
        SenderId: senderId,
      },
    },
```

```
  },
};

const run = async () => {
  try {
    const data = await pinClient.send(new SendMessagesCommand(params));
    console.log(
      `Message sent!
 ${data.MessageResponse.Result[destinationNumber].StatusMessage}`,
    );
  } catch (err) {
    console.log(err);
  }
};
run();
```

- 如需 API 詳細資訊，請參閱 適用於 JavaScript 的 AWS SDK API 參考中的 SendMessages。

適用於 JavaScript (v2) 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

傳送電子郵件訊息。

```
"use strict";

const AWS = require("aws-sdk");

// The AWS Region that you want to use to send the email. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/
const aws_region = "us-west-2";

// The "From" address. This address has to be verified in Amazon Pinpoint
// in the region that you use to send email.
const senderAddress = "sender@example.com";
```

```
// The address on the "To" line. If your Amazon Pinpoint account is in
// the sandbox, this address also has to be verified.
var toAddress = "recipient@example.com";

// The Amazon Pinpoint project/application ID to use when you send this message.
// Make sure that the SMS channel is enabled for the project or application
// that you choose.
const appId = "ce796be37f32f178af652b26eexample";

// The subject line of the email.
var subject = "Amazon Pinpoint (AWS SDK for JavaScript in Node.js)";

// The email body for recipients with non-HTML email clients.
var body_text = `Amazon Pinpoint Test (SDK for JavaScript in Node.js)
-----------------------------------------------------
This email was sent with Amazon Pinpoint using the AWS SDK for JavaScript in
 Node.js.
For more information, see https:\/\/aws.amazon.com/sdk-for-node-js/`;

// The body of the email for recipients whose email clients support HTML content.
var body_html = `<html>
<head></head>
<body>
  <h1>Amazon Pinpoint Test (SDK for JavaScript in Node.js)</h1>
  <p>This email was sent with
    <a href='https://aws.amazon.com/pinpoint/'>the Amazon Pinpoint API</a> using
 the
    <a href='https://aws.amazon.com/sdk-for-node-js/'>
      AWS SDK for JavaScript in Node.js</a>.</p>
</body>
</html>`;

// The character encoding the you want to use for the subject line and
// message body of the email.
var charset = "UTF-8";

// Specify that you're using a shared credentials file.
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
```

```
var pinpoint = new AWS.Pinpoint();

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: appId,
  MessageRequest: {
    Addresses: {
      [toAddress]: {
        ChannelType: "EMAIL",
      },
    },
    MessageConfiguration: {
      EmailMessage: {
        FromAddress: senderAddress,
        SimpleEmail: {
          Subject: {
            Charset: charset,
            Data: subject,
          },
          HtmlPart: {
            Charset: charset,
            Data: body_html,
          },
          TextPart: {
            Charset: charset,
            Data: body_text,
          },
        },
      },
    },
  },
};

//Try to send the email.
pinpoint.sendMessages(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
  } else {
    console.log(
      "Email sent! Message ID: ",
      data["MessageResponse"]["Result"][toAddress]["MessageId"]
    );
  }
```

```
});
```

傳送一則 SMS 訊息。

```
"use strict";

var AWS = require("aws-sdk");

// The AWS Region that you want to use to send the message. For a list of
// AWS Regions where the Amazon Pinpoint API is available, see
// https://docs.aws.amazon.com/pinpoint/latest/apireference/.
var aws_region = "us-east-1";

// The phone number or short code to send the message from. The phone number
// or short code that you specify has to be associated with your Amazon Pinpoint
// account. For best results, specify long codes in E.164 format.
var originationNumber = "+12065550199";

// The recipient's phone number.  For best results, you should specify the
// phone number in E.164 format.
var destinationNumber = "+14255550142";

// The content of the SMS message.
var message =
  "This message was sent through Amazon Pinpoint " +
  "using the AWS SDK for JavaScript in Node.js. Reply STOP to " +
  "opt out.";

// The Amazon Pinpoint project/application ID to use when you send this message.
// Make sure that the SMS channel is enabled for the project or application
// that you choose.
var applicationId = "ce796be37f32f178af652b26eexample";

// The type of SMS message that you want to send. If you plan to send
// time-sensitive content, specify TRANSACTIONAL. If you plan to send
// marketing-related content, specify PROMOTIONAL.
var messageType = "TRANSACTIONAL";

// The registered keyword associated with the originating short code.
var registeredKeyword = "myKeyword";
```

```
// The sender ID to use when sending the message. Support for sender ID
// varies by country or region. For more information, see
// https://docs.aws.amazon.com/pinpoint/latest/userguide/channels-sms-
countries.html
var senderId = "MySenderID";

// Specify that you're using a shared credentials file, and optionally specify
// the profile that you want to use.
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpoint = new AWS.Pinpoint();

// Specify the parameters to pass to the API.
var params = {
  ApplicationId: applicationId,
  MessageRequest: {
    Addresses: {
      [destinationNumber]: {
        ChannelType: "SMS",
      },
    },
    MessageConfiguration: {
      SMSMessage: {
        Body: message,
        Keyword: registeredKeyword,
        MessageType: messageType,
        OriginationNumber: originationNumber,
        SenderId: senderId,
      },
    },
  },
};

//Try to send the message.
pinpoint.sendMessages(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
```

```
    // Otherwise, show the unique ID for the message.
  } else {
    console.log(
      "Message sent! " +
        data["MessageResponse"]["Result"][destinationNumber]["StatusMessage"]
    );
  }
});
```

- 如需 API 詳細資訊，請參閱 適用於 JavaScript 的 AWS SDK API 參考中的 SendMessages。

Kotlin

適用於 Kotlin 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

```
/**
Before running this Kotlin code example, set up your development environment,
including your credentials.

For more information, see the following documentation topic:
https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html
 */

val body: String =
    """
    Amazon Pinpoint test (AWS SDK for Kotlin)

    This email was sent through the Amazon Pinpoint Email API using the AWS SDK
 for Kotlin.

    """.trimIndent()

suspend fun main(args: Array<String>) {
```

```
    val usage = """
    Usage:
        <subject> <appId> <senderAddress> <toAddress>

    Where:
        subject - The email subject to use.
        senderAddress - The from address. This address has to be verified in
 Amazon Pinpoint in the region you're using to send email
        toAddress - The to address. This address has to be verified in Amazon
 Pinpoint in the region you're using to send email
    """

    if (args.size != 3) {
        println(usage)
        exitProcess(0)
    }

    val subject = args[0]
    val senderAddress = args[1]
    val toAddress = args[2]
    sendEmail(subject, senderAddress, toAddress)
}

suspend fun sendEmail(
    subjectVal: String?,
    senderAddress: String,
    toAddressVal: String,
) {
    var content =
        Content {
            data = body
        }

    val messageBody =
        Body {
            text = content
        }

    val subContent =
        Content {
            data = subjectVal
        }

    val message =
```

```
        Message {
            body = messageBody
            subject = subContent
        }

    val destinationOb =
        Destination {
            toAddresses = listOf(toAddressVal)
        }

    val emailContent =
        EmailContent {
            simple = message
        }

    val sendEmailRequest =
        SendEmailRequest {
            fromEmailAddress = senderAddress
            destination = destinationOb
            this.content = emailContent
        }

    PinpointEmailClient.fromEnvironment { region = "us-east-1" }.use
 { pinpointemail ->
        pinpointemail.sendEmail(sendEmailRequest)
        println("Message Sent")
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Kotlin API 參考中的 SendMessages。

Python

適用於 Python 的 SDK (Boto3)

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

傳送電子郵件訊息。

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


def send_email_message(
    pinpoint_client,
    app_id,
    sender,
    to_addresses,
    char_set,
    subject,
    html_message,
    text_message,
):
    """
    Sends an email message with HTML and plain text versions.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param app_id: The Amazon Pinpoint project ID to use when you send this
 message.
    :param sender: The "From" address. This address must be verified in
                   Amazon Pinpoint in the AWS Region you're using to send email.
    :param to_addresses: The addresses on the "To" line. If your Amazon Pinpoint
 account
                         is in the sandbox, these addresses must be verified.
    :param char_set: The character encoding to use for the subject line and
 message
                     body of the email.
    :param subject: The subject line of the email.
    :param html_message: The body of the email for recipients whose email clients
 can
                         display HTML content.
    :param text_message: The body of the email for recipients whose email clients
                         don't support HTML content.
    :return: A dict of to_addresses and their message IDs.
    """
    try:
        response = pinpoint_client.send_messages(
```

```
            ApplicationId=app_id,
            MessageRequest={
                "Addresses": {
                    to_address: {"ChannelType": "EMAIL"} for to_address in
 to_addresses
                },
                "MessageConfiguration": {
                    "EmailMessage": {
                        "FromAddress": sender,
                        "SimpleEmail": {
                            "Subject": {"Charset": char_set, "Data": subject},
                            "HtmlPart": {"Charset": char_set, "Data":
 html_message},
                            "TextPart": {"Charset": char_set, "Data":
 text_message},
                        },
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send email.")
        raise
    else:
        return {
            to_address: message["MessageId"]
            for to_address, message in response["MessageResponse"]
["Result"].items()
        }


def main():
    app_id = "ce796be37f32f178af652b26eexample"
    sender = "sender@example.com"
    to_address = "recipient@example.com"
    char_set = "UTF-8"
    subject = "Amazon Pinpoint Test (SDK for Python (Boto3))"
    text_message = """Amazon Pinpoint Test (SDK for Python)
    -----------------------------------
    This email was sent with Amazon Pinpoint using the AWS SDK for Python
 (Boto3).
    For more information, see https://aws.amazon.com/sdk-for-python/
                """
    html_message = """<html>
```

```
    <head></head>
    <body>
      <h1>Amazon Pinpoint Test (SDK for Python (Boto3)</h1>
      <p>This email was sent with
        <a href='https://aws.amazon.com/pinpoint/'>Amazon Pinpoint</a> using the
        <a href='https://aws.amazon.com/sdk-for-python/'>
          AWS SDK for Python (Boto3)</a>.</p>
    </body>
    </html>
                """

    print("Sending email.")
    message_ids = send_email_message(
        boto3.client("pinpoint"),
        app_id,
        sender,
        [to_address],
        char_set,
        subject,
        html_message,
        text_message,
    )
    print(f"Message sent! Message IDs: {message_ids}")


if __name__ == "__main__":
    main()
```

傳送一則 SMS 訊息。

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


def send_sms_message(
    pinpoint_client,
    app_id,
    origination_number,
```

```
    destination_number,
    message,
    message_type,
):
    """
    Sends an SMS message with Amazon Pinpoint.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param app_id: The Amazon Pinpoint project/application ID to use when you
 send
                   this message. The SMS channel must be enabled for the project
 or
                   application.
    :param destination_number: The recipient's phone number in E.164 format.
    :param origination_number: The phone number to send the message from. This
 phone
                               number must be associated with your Amazon
 Pinpoint
                               account and be in E.164 format.
    :param message: The content of the SMS message.
    :param message_type: The type of SMS message that you want to send. If you
 send
                         time-sensitive content, specify TRANSACTIONAL. If you
 send
                         marketing-related content, specify PROMOTIONAL.
    :return: The ID of the message.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=app_id,
            MessageRequest={
                "Addresses": {destination_number: {"ChannelType": "SMS"}},
                "MessageConfiguration": {
                    "SMSMessage": {
                        "Body": message,
                        "MessageType": message_type,
                        "OriginationNumber": origination_number,
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send message.")
        raise
```

```
    else:
        return response["MessageResponse"]["Result"][destination_number]
["MessageId"]


def main():
    app_id = "ce796be37f32f178af652b26eexample"
    origination_number = "+12065550199"
    destination_number = "+14255550142"
    message = (
        "This is a sample message sent from Amazon Pinpoint by using the AWS SDK
 for "
        "Python (Boto 3)."
    )
    message_type = "TRANSACTIONAL"

    print("Sending SMS message.")
    message_id = send_sms_message(
        boto3.client("pinpoint"),
        app_id,
        origination_number,
        destination_number,
        message,
        message_type,
    )
    print(f"Message sent! Message ID: {message_id}.")


if __name__ == "__main__":
    main()
```

使用現有的電子郵件範本，傳送電子郵件訊息。

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


def send_templated_email_message(
```

```
    pinpoint_client, project_id, sender, to_addresses, template_name,
 template_version
):
    """
    Sends an email message with HTML and plain text versions.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: The Amazon Pinpoint project ID to use when you send this
 message.
    :param sender: The "From" address. This address must be verified in
                   Amazon Pinpoint in the AWS Region you're using to send email.
    :param to_addresses: The addresses on the "To" line. If your Amazon Pinpoint
                         account is in the sandbox, these addresses must be
 verified.
    :param template_name: The name of the email template to use when sending the
 message.
    :param template_version: The version number of the message template.

    :return: A dict of to_addresses and their message IDs.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=project_id,
            MessageRequest={
                "Addresses": {
                    to_address: {"ChannelType": "EMAIL"} for to_address in
 to_addresses
                },
                "MessageConfiguration": {"EmailMessage": {"FromAddress":
 sender}},
                "TemplateConfiguration": {
                    "EmailTemplate": {
                        "Name": template_name,
                        "Version": template_version,
                    }
                },
            },
        )
    except ClientError:
        logger.exception("Couldn't send email.")
        raise
    else:
        return {
            to_address: message["MessageId"]
```

```
            for to_address, message in response["MessageResponse"]
["Result"].items()
        }


def main():
    project_id = "296b04b342374fceb661bf494example"
    sender = "sender@example.com"
    to_addresses = ["recipient@example.com"]
    template_name = "My_Email_Template"
    template_version = "1"

    print("Sending email.")
    message_ids = send_templated_email_message(
        boto3.client("pinpoint"),
        project_id,
        sender,
        to_addresses,
        template_name,
        template_version,
    )
    print(f"Message sent! Message IDs: {message_ids}")


if __name__ == "__main__":
    main()
```

使用現有的 SMS 範本傳送文字訊息。

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


def send_templated_sms_message(
    pinpoint_client,
    project_id,
    destination_number,
    message_type,
    origination_number,
```

```
    template_name,
    template_version,
):
    """
    Sends an SMS message to a specific phone number using a pre-defined template.

    :param pinpoint_client: A Boto3 Pinpoint client.
    :param project_id: An Amazon Pinpoint project (application) ID.
    :param destination_number: The phone number to send the message to.
    :param message_type: The type of SMS message (promotional or transactional).
    :param origination_number: The phone number that the message is sent from.
    :param template_name: The name of the SMS template to use when sending the
 message.
    :param template_version: The version number of the message template.

    :return The ID of the message.
    """
    try:
        response = pinpoint_client.send_messages(
            ApplicationId=project_id,
            MessageRequest={
                "Addresses": {destination_number: {"ChannelType": "SMS"}},
                "MessageConfiguration": {
                    "SMSMessage": {
                        "MessageType": message_type,
                        "OriginationNumber": origination_number,
                    }
                },
                "TemplateConfiguration": {
                    "SMSTemplate": {"Name": template_name, "Version":
 template_version}
                },
            },
        )

    except ClientError:
        logger.exception("Couldn't send message.")
        raise
    else:
        return response["MessageResponse"]["Result"][destination_number]
["MessageId"]


def main():
```

```
    region = "us-east-1"
    origination_number = "+18555550001"
    destination_number = "+14255550142"
    project_id = "7353f53e6885409fa32d07cedexample"
    message_type = "TRANSACTIONAL"
    template_name = "My_SMS_Template"
    template_version = "1"
    message_id = send_templated_sms_message(
        boto3.client("pinpoint", region_name=region),
        project_id,
        destination_number,
        message_type,
        origination_number,
        template_name,
        template_version,
    )
    print(f"Message sent! Message ID: {message_id}.")


if __name__ == "__main__":
    main()
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Python (Boto3) API 參考中的 SendMessages。

SAP ABAP

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

傳送電子郵件訊息。

```
    " Build the addresses map from the list of to_addresses
    DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
    LOOP AT it_to_addresses INTO DATA(lo_address).
      INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
        key = lo_address->get_value( )
```

```
      value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'EMAIL' )
    ) INTO TABLE lt_addresses.
  ENDLOOP.

  " Send the email message
  DATA(lo_result) = lo_ppt->sendmessages(
    iv_applicationid = iv_app_id
    io_messagerequest = NEW /aws1/cl_pptmessagerequest(
      it_addresses = lt_addresses
      io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
        io_emailmessage = NEW /aws1/cl_pptemailmessage(
          iv_fromaddress = iv_sender
          io_simpleemail = NEW /aws1/cl_pptsimpleemail(
            io_subject = NEW /aws1/cl_pptsimpleemailpart(
              iv_charset = iv_char_set
              iv_data = iv_subject
            )
            io_htmlpart = NEW /aws1/cl_pptsimpleemailpart(
              iv_charset = iv_char_set
              iv_data = iv_html_message
            )
            io_textpart = NEW /aws1/cl_pptsimpleemailpart(
              iv_charset = iv_char_set
              iv_data = iv_text_message
            )
          )
        )
      )
    )
  ).

  " Extract message IDs from response
  DATA(lo_message_response) = lo_result->get_messageresponse( ).
  ot_message_ids = lo_message_response->get_result( ).

  MESSAGE 'Email message sent successfully.' TYPE 'I'.
```

傳送一則 SMS 訊息。

```
  " Build the addresses map for the destination number
  DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
  INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
```

```
      key = iv_destination_number
      value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'SMS' )
    ) INTO TABLE lt_addresses.

    " Send the SMS message
    DATA(lo_result) = lo_ppt->sendmessages(
      iv_applicationid = iv_app_id
      io_messagerequest = NEW /aws1/cl_pptmessagerequest(
        it_addresses = lt_addresses
        io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
          io_smsmessage = NEW /aws1/cl_pptsmsmessage(
            iv_body = iv_message
            iv_messagetype = iv_message_type
            iv_originationnumber = iv_origination_number
          )
        )
      )
    ).

    " Extract message ID from response
    DATA(lo_message_response) = lo_result->get_messageresponse( ).
    DATA(lt_results) = lo_message_response->get_result( ).
    LOOP AT lt_results INTO DATA(ls_result).
      IF ls_result-key = iv_destination_number.
        ov_message_id = ls_result-value->get_messageid( ).
        EXIT.
      ENDIF.
    ENDLOOP.

    MESSAGE 'SMS message sent successfully.' TYPE 'I'.
```

使用現有的電子郵件範本，傳送電子郵件訊息。

```
    " Build the addresses map from the list of to_addresses
    DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
    LOOP AT it_to_addresses INTO DATA(lo_address).
      INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
        key = lo_address->get_value( )
        value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'EMAIL' )
      ) INTO TABLE lt_addresses.
    ENDLOOP.
```

```
" Send the email message using a template
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_emailmessage = NEW /aws1/cl_pptemailmessage(
        iv_fromaddress = iv_sender
      )
    )
    io_templateconfiguration = NEW /aws1/cl_ppttemplateconf(
      io_emailtemplate = NEW /aws1/cl_ppttemplate(
        iv_name = iv_template_name
        iv_version = iv_template_version
      )
    )
  )
).

" Extract message IDs from response
DATA(lo_message_response) = lo_result->get_messageresponse( ).
ot_message_ids = lo_message_response->get_result( ).

MESSAGE 'Templated email message sent successfully.' TYPE 'I'.
```

使用現有的 SMS 範本傳送文字訊息。

```
" Build the addresses map for the destination number
DATA lt_addresses TYPE /aws1/cl_pptaddressconf=>tt_mapofaddressconfiguration.
INSERT VALUE /aws1/cl_pptaddressconf=>ts_mapofaddressconf_maprow(
  key = iv_destination_number
  value = NEW /aws1/cl_pptaddressconf( iv_channeltype = 'SMS' )
) INTO TABLE lt_addresses.

" Send the SMS message using a template
DATA(lo_result) = lo_ppt->sendmessages(
  iv_applicationid = iv_app_id
  io_messagerequest = NEW /aws1/cl_pptmessagerequest(
    it_addresses = lt_addresses
    io_messageconfiguration = NEW /aws1/cl_pptdirectmessageconf(
      io_smsmessage = NEW /aws1/cl_pptsmsmessage(
        iv_messagetype = iv_message_type
```

```
            iv_originationnumber = iv_origination_number
          )
        )
        io_templateconfiguration = NEW /aws1/cl_ppttemplateconf(
          io_smstemplate = NEW /aws1/cl_ppttemplate(
            iv_name = iv_template_name
            iv_version = iv_template_version
          )
        )
      )
    ).

    " Extract message ID from response
    DATA(lo_message_response) = lo_result->get_messageresponse( ).
    DATA(lt_results) = lo_message_response->get_result( ).
    LOOP AT lt_results INTO DATA(ls_result).
      IF ls_result-key = iv_destination_number.
        ov_message_id = ls_result-value->get_messageid( ).
        EXIT.
      ENDIF.
    ENDLOOP.

    MESSAGE 'Templated SMS message sent successfully.' TYPE 'I'.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 SendMessages。

如需AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

`UpdateEndpoint` 搭配AWS SDK 使用

以下程式碼範例顯示如何使用 `UpdateEndpoint`。

Java

適用於 Java 2.x 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS程式碼範例儲存庫中設定和執行。

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpoint.PinpointClient;
import software.amazon.awssdk.services.pinpoint.model.EndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.EndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.UpdateEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointRequest;
import software.amazon.awssdk.services.pinpoint.model.GetEndpointResponse;
import software.amazon.awssdk.services.pinpoint.model.PinpointException;
import software.amazon.awssdk.services.pinpoint.model.EndpointDemographic;
import software.amazon.awssdk.services.pinpoint.model.EndpointLocation;
import software.amazon.awssdk.services.pinpoint.model.EndpointUser;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.List;
import java.util.UUID;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
import java.util.Date;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class UpdateEndpoint {
    public static void main(String[] args) {
```

```
        final String usage = """

                Usage:  <appId>

                Where:
                  appId - The ID of the application to create an endpoint for.

                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String appId = args[0];
        PinpointClient pinpoint = PinpointClient.builder()
                .region(Region.US_EAST_1)
                .build();

        EndpointResponse response = createEndpoint(pinpoint, appId);
        System.out.println("Got Endpoint: " + response.id());
        pinpoint.close();
    }

    public static EndpointResponse createEndpoint(PinpointClient client, String
appId) {
        String endpointId = UUID.randomUUID().toString();
        System.out.println("Endpoint ID: " + endpointId);

        try {
            EndpointRequest endpointRequest = createEndpointRequestData();
            UpdateEndpointRequest updateEndpointRequest =
UpdateEndpointRequest.builder()
                    .applicationId(appId)
                    .endpointId(endpointId)
                    .endpointRequest(endpointRequest)
                    .build();

            UpdateEndpointResponse updateEndpointResponse =
client.updateEndpoint(updateEndpointRequest);
            System.out.println("Update Endpoint Response: " +
updateEndpointResponse.messageBody());

            GetEndpointRequest getEndpointRequest = GetEndpointRequest.builder()
```

```
                    .applicationId(appId)
                    .endpointId(endpointId)
                    .build();

            GetEndpointResponse getEndpointResponse =
client.getEndpoint(getEndpointRequest);
            System.out.println(getEndpointResponse.endpointResponse().address());

System.out.println(getEndpointResponse.endpointResponse().channelType());

System.out.println(getEndpointResponse.endpointResponse().applicationId());

System.out.println(getEndpointResponse.endpointResponse().endpointStatus());

System.out.println(getEndpointResponse.endpointResponse().requestId());
            System.out.println(getEndpointResponse.endpointResponse().user());

            return getEndpointResponse.endpointResponse();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    private static EndpointRequest createEndpointRequestData() {
        try {
            List<String> favoriteTeams = new ArrayList<>();
            favoriteTeams.add("Lakers");
            favoriteTeams.add("Warriors");
            HashMap<String, List<String>> customAttributes = new HashMap<>();
            customAttributes.put("team", favoriteTeams);

            EndpointDemographic demographic = EndpointDemographic.builder()
                    .appVersion("1.0")
                    .make("apple")
                    .model("iPhone")
                    .modelVersion("7")
                    .platform("ios")
                    .platformVersion("10.1.1")
                    .timezone("America/Los_Angeles")
                    .build();
```

```
            EndpointLocation location = EndpointLocation.builder()
                    .city("Los Angeles")
                    .country("US")
                    .latitude(34.0)
                    .longitude(-118.2)
                    .postalCode("90068")
                    .region("CA")
                    .build();

            Map<String, Double> metrics = new HashMap<>();
            metrics.put("health", 100.00);
            metrics.put("luck", 75.00);

            EndpointUser user = EndpointUser.builder()
                    .userId(UUID.randomUUID().toString())
                    .build();

            DateFormat df = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm'Z'"); //
 Quoted "Z" to indicate UTC, no timezone
                                                                           //
 offset
            String nowAsISO = df.format(new Date());

            return EndpointRequest.builder()
                    .address(UUID.randomUUID().toString())
                    .attributes(customAttributes)
                    .channelType("APNS")
                    .demographic(demographic)
                    .effectiveDate(nowAsISO)
                    .location(location)
                    .metrics(metrics)
                    .optOut("NONE")
                    .requestId(UUID.randomUUID().toString())
                    .user(user)
                    .build();

        } catch (PinpointException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考中的 UpdateEndpoint。

如需AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

使用AWS SDKs Amazon Pinpoint 案例

下列程式碼範例示範如何在 Amazon Pinpoint AWS SDKs 中實作常見案例。這些案例說明如何透過在 Amazon Pinpoint 中呼叫多個函數或與其他函數結合，來完成特定任務AWS 服務。每個案例均包含完整原始碼的連結，您可在連結中找到如何設定和執行程式碼的相關指示。

案例的目標是獲得中等水平的經驗，協助您了解內容中的服務動作。

範例

- 推送通知入門

推送通知入門

以下程式碼範例顯示做法：

- 建立應用程式
- 啟用推送通知管道
- 傳送推播通知
- 清除資源

Bash

AWS CLI使用 Bash 指令碼

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在範例開發人員教學課程儲存庫中設定和執行。

```
#!/bin/bash
```

```
# AWS End User Messaging Push Getting Started Script
# This script creates an AWS End User Messaging Push application and demonstrates
# how to enable push notification channels and send a test message.
#
# Prerequisites:
# - AWS CLI installed and configured
# - Appropriate IAM permissions for Pinpoint operations
#
# Usage: ./aws-end-user-messaging-gs.sh [--auto-cleanup]

set -uo pipefail

# Security: Set secure umask for created files
umask 0077

# Set up logging
LOG_DIR="./aws-eump-logs"
mkdir -p "$LOG_DIR"
LOG_FILE="$LOG_DIR/aws-end-user-messaging-push-script-$(date +%Y%m%d-%H%M%S).log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting AWS End User Messaging Push setup script..."
echo "Logging to $LOG_FILE"
echo "Timestamp: $(date)"

# Security: Track created resources for cleanup
declare -a TEMP_FILES=()
declare -a AWS_RESOURCES=()

# Cleanup function with improved security
cleanup() {
    local exit_code=$?
    echo "Cleaning up temporary resources..."

    # Remove temporary files securely
    for temp_file in "${TEMP_FILES[@]+"${TEMP_FILES[@]}"}"; do
        if [ -f "$temp_file" ]; then
            shred -vfz -n 3 "$temp_file" 2>/dev/null || rm -f "$temp_file"
        fi
    done

    # Optionally delete AWS resources
    if [ "${DELETE_AWS_RESOURCES:-false}" = "true" ]; then
```

```
        for resource in "${AWS_RESOURCES[@]+"${AWS_RESOURCES[@]}"}"; do
            echo "Deleting AWS resource: $resource"
            aws pinpoint delete-app --application-id "$resource" 2>/dev/null || \
                echo "Warning: Failed to delete application $resource"
        done
    fi

    exit "$exit_code"
}

trap cleanup EXIT INT TERM

# Function to check for errors in command output
check_error() {
    local output=$1
    local cmd=$2
    local ignore_error=${3:-false}

    if echo "$output" | grep -qi "error\|exception\|fail"; then
        echo "ERROR: Command failed: $cmd" >&2
        echo "Error details: $output" >&2

        if [ "$ignore_error" = "true" ]; then
            echo "Ignoring error and continuing..." >&2
            return 1
        else
            return 2
        fi
    fi

    return 0
}

# Function to validate AWS CLI is configured
validate_aws_cli() {
    echo "Validating AWS CLI configuration..."

    # Check if AWS CLI is installed
    if ! command -v aws &> /dev/null; then
        echo "ERROR: AWS CLI is not installed. Please install it first." >&2
        echo "Visit: https://docs.aws.amazon.com/cli/latest/userguide/getting-
started-install.html" >&2
        return 1
    fi
```

```
    # Check AWS CLI version
    AWS_VERSION=$(aws --version 2>&1 | head -n1)
    echo "AWS CLI version: $AWS_VERSION"

    # Verify credentials are set (check for credential env vars or config file)
    if ! aws sts get-caller-identity &> /dev/null; then
        echo "ERROR: AWS CLI credentials are not configured or invalid." >&2
        echo "Please configure credentials via environment variables, credential
 file, or 'aws configure'" >&2
        return 1
    fi

    # Get current AWS identity and region
    CALLER_IDENTITY=$(aws sts get-caller-identity)
    CURRENT_REGION=$(aws configure get region 2>/dev/null || echo "us-east-1")
    echo "AWS CLI configured for:"
    echo "$CALLER_IDENTITY"
    echo "Current region: $CURRENT_REGION"
    echo ""

    return 0
}

# Function to check if jq is available for JSON parsing
check_json_tools() {
    if command -v jq &> /dev/null; then
        USE_JQ=true
        echo "jq is available for JSON parsing"
    else
        USE_JQ=false
        echo "jq is not available, using grep for JSON parsing"
        echo "Consider installing jq for better JSON handling: https://
stedolan.github.io/jq/"
    fi
}

# Function to extract JSON values safely
extract_json_value() {
    local json=$1
    local key=$2

    if [ "$USE_JQ" = "true" ]; then
```

```
        echo "$json" | jq -r ".ApplicationResponse.$key // empty" 2>/dev/null ||
 echo ""
    else
        # Fallback to grep method with better validation
        echo "$json" | grep -o "\"$key\": \"[^\"]*" | cut -d'"' -f4 | head -n1 ||
 echo ""
    fi
}

# Function to validate required IAM permissions
validate_permissions() {
    echo "Validating IAM permissions..."

    # Test basic Pinpoint permissions
    if ! aws pinpoint get-apps > /dev/null 2>&1; then
        echo "WARNING: Unable to list Pinpoint applications." >&2
        echo "Please ensure you have appropriate IAM permissions for Pinpoint
 operations." >&2
        echo "Required permissions:" >&2
        echo "  - mobiletargeting:GetApps" >&2
        echo "  - mobiletargeting:CreateApp" >&2
        echo "  - mobiletargeting:DeleteApp" >&2
        echo "  - mobiletargeting:UpdateGcmChannel" >&2
        echo "  - mobiletargeting:UpdateApnsChannel" >&2
        echo "  - mobiletargeting:SendMessages" >&2
    else
        echo "Basic Pinpoint permissions validated."
    fi
}

# Function to validate input parameters
validate_input() {
    local app_name=$1

    # Validate app name length and characters
    if [ ${#app_name} -gt 64 ]; then
        echo "ERROR: Application name exceeds maximum length of 64 characters"
 >&2
        return 1
    fi

    if ! [[ "$app_name" =~ ^[a-zA-Z0-9_-]+$ ]]; then
        echo "ERROR: Application name contains invalid characters" >&2
        return 1
```

```
    fi

    return 0
}

# Function to create secure temporary files
create_temp_file() {
    local temp_file
    temp_file=$(mktemp) || {
        echo "ERROR: Failed to create temporary file" >&2
        return 1
    }
    chmod 600 "$temp_file"
    TEMP_FILES+=("$temp_file")
    echo "$temp_file"
}

# Validate prerequisites
if ! validate_aws_cli; then
    exit 1
fi

check_json_tools
validate_permissions

# Generate a random suffix for resource names to avoid conflicts
RANDOM_SUFFIX=$(LC_ALL=C tr -dc 'a-z0-9' < /dev/urandom | fold -w 8 | head -n1)
APP_NAME="PushNotificationApp-${RANDOM_SUFFIX}"

# Validate input
if ! validate_input "$APP_NAME"; then
    exit 1
fi

echo "Creating application with name: $APP_NAME"

# Step 1: Create an application
echo "Executing: aws pinpoint create-app --create-application-request Name=
${APP_NAME}"
CREATE_APP_OUTPUT=$(aws pinpoint create-app --create-application-request "Name=
${APP_NAME},tags={project=doc-smith,tutorial=aws-end-user-messaging-gs}" 2>&1)

if ! check_error "$CREATE_APP_OUTPUT" "create-app"; then
    exit 1
```

```
fi

echo "Application created successfully:"
echo "$CREATE_APP_OUTPUT"

# Extract the application ID from the output
APP_ID=$(extract_json_value "$CREATE_APP_OUTPUT" "Id")

if [ -z "$APP_ID" ] || [ "$APP_ID" = "null" ]; then
    echo "ERROR: Failed to extract application ID from output" >&2
    echo "Output was: $CREATE_APP_OUTPUT" >&2
    exit 1
fi

echo "Application ID: $APP_ID"
AWS_RESOURCES+=("$APP_ID")

# Step 2: Enable FCM (GCM) channel with a sample API key
echo ""
echo "========================================="
echo "ENABLING FCM (GCM) CHANNEL"
echo "========================================="
echo "Note: This is using a placeholder API key for demonstration purposes only."
echo "In a production environment, you should use your actual FCM API key from
 Firebase Console."
echo ""
echo "IMPORTANT: The following command will likely fail because we're using a
 placeholder API key."
echo "This is expected behavior for this demonstration script."

echo "Executing: aws pinpoint update-gcm-channel --application-id $APP_ID --gcm-
channel-request ..."
UPDATE_GCM_OUTPUT=$(aws pinpoint update-gcm-channel \
    --application-id "$APP_ID" \
    --gcm-channel-request '{"Enabled": true, "ApiKey": "sample-fcm-api-key-for-
demo-only"}' 2>&1)

# We'll ignore this specific error since we're using a placeholder API key
if check_error "$UPDATE_GCM_OUTPUT" "update-gcm-channel" "true"; then
    echo "FCM channel enabled successfully:"
    echo "$UPDATE_GCM_OUTPUT"
else
    echo "As expected, FCM channel update failed with the placeholder API key."
    echo "Error details: $UPDATE_GCM_OUTPUT"
```

```
    echo ""
    echo "To enable FCM in production:"
    echo "1. Go to Firebase Console (https://console.firebase.google.com/)"
    echo "2. Create or select your project"
    echo "3. Go to Project Settings > Cloud Messaging"
    echo "4. Copy the Server Key"
    echo "5. Replace 'sample-fcm-api-key-for-demo-only' with your actual Server
 Key"
fi

# Step 3: Try to enable APNS channel (this will also fail without real
 certificates)
echo ""
echo "========================================="
echo "ENABLING APNS CHANNEL (OPTIONAL)"
echo "========================================="
echo "Attempting to enable APNS channel with placeholder certificate..."
echo "This will also fail without real APNS certificates, which is expected."

# Create a placeholder APNS configuration
echo "Executing: aws pinpoint update-apns-channel --application-id $APP_ID --
apns-channel-request ..."
UPDATE_APNS_OUTPUT=$(aws pinpoint update-apns-channel \
    --application-id "$APP_ID" \
    --apns-channel-request '{"Enabled": true, "Certificate": "placeholder-
certificate", "PrivateKey": "placeholder-private-key"}' 2>&1)

if check_error "$UPDATE_APNS_OUTPUT" "update-apns-channel" "true"; then
    echo "APNS channel enabled successfully:"
    echo "$UPDATE_APNS_OUTPUT"
else
    echo "As expected, APNS channel update failed with placeholder certificates."
    echo "Error details: $UPDATE_APNS_OUTPUT"
    echo ""
    echo "To enable APNS in production:"
    echo "1. Generate APNS certificates from Apple Developer Console"
    echo "2. Convert certificates to PEM format"
    echo "3. Use the actual certificate and private key in the update-apns-
channel command"
fi

# Step 4: Create message files for different platforms
echo ""
echo "========================================="
```

```
echo "CREATING MESSAGE FILES"
echo "========================================="

# Create FCM message file securely
GCM_MESSAGE_FILE=$(create_temp_file)
echo "Creating FCM message file..."
cat > "$GCM_MESSAGE_FILE" << 'EOF'
{
  "Addresses": {
    "SAMPLE-DEVICE-TOKEN-FCM": {
      "ChannelType": "GCM"
    }
  },
  "MessageConfiguration": {
    "GCMMessage": {
      "Action": "OPEN_APP",
      "Body": "Hello from AWS End User Messaging Push! This is an FCM
 notification.",
      "Priority": "normal",
      "SilentPush": false,
      "Title": "My First FCM Push Notification",
      "TimeToLive": 30,
      "Data": {
        "key1": "value1",
        "key2": "value2"
      }
    }
  }
}
EOF

# Create APNS message file securely
APNS_MESSAGE_FILE=$(create_temp_file)
echo "Creating APNS message file..."
cat > "$APNS_MESSAGE_FILE" << 'EOF'
{
  "Addresses": {
    "SAMPLE-DEVICE-TOKEN-APNS": {
      "ChannelType": "APNS"
    }
  },
  "MessageConfiguration": {
    "APNSMessage": {
      "Action": "OPEN_APP",
```

```
      "Body": "Hello from AWS End User Messaging Push! This is an APNS
 notification.",
      "Priority": "normal",
      "SilentPush": false,
      "Title": "My First APNS Push Notification",
      "TimeToLive": 30,
      "Badge": 1,
      "Sound": "default"
    }
  }
}
EOF

echo "Message files created:"
echo "- FCM message file (for FCM/Android)"
echo "- APNS message file (for APNS/iOS)"
echo ""
echo "Note: These messages use placeholder device tokens and will not actually be
 delivered."
echo "To send real messages, you would need to replace the sample device tokens
 with actual ones."

# Step 5: Demonstrate how to send messages (this will fail with placeholder
 tokens)
echo ""
echo "=========================================="
echo "DEMONSTRATING MESSAGE SENDING"
echo "=========================================="
echo "Attempting to send FCM message (will fail with placeholder token)..."

echo "Executing: aws pinpoint send-messages --application-id $APP_ID --message-
request file://<gcm-message>"
SEND_FCM_OUTPUT=$(aws pinpoint send-messages \
    --application-id "$APP_ID" \
    --message-request "file://$GCM_MESSAGE_FILE" 2>&1)

if check_error "$SEND_FCM_OUTPUT" "send-messages (FCM)" "true"; then
    echo "FCM message sent successfully:"
    echo "$SEND_FCM_OUTPUT"
else
    echo "As expected, FCM message sending failed with placeholder token."
    echo "Error details: $SEND_FCM_OUTPUT"
fi
```

```
echo ""
echo "Attempting to send APNS message (will fail with placeholder token)..."

echo "Executing: aws pinpoint send-messages --application-id $APP_ID --message-
request file://<apns-message>"
SEND_APNS_OUTPUT=$(aws pinpoint send-messages \
    --application-id "$APP_ID" \
    --message-request "file://$APNS_MESSAGE_FILE" 2>&1)

if check_error "$SEND_APNS_OUTPUT" "send-messages (APNS)" "true"; then
    echo "APNS message sent successfully:"
    echo "$SEND_APNS_OUTPUT"
else
    echo "As expected, APNS message sending failed with placeholder token."
    echo "Error details: $SEND_APNS_OUTPUT"
fi

# Step 6: Show application details
echo ""
echo "=========================================="
echo "APPLICATION DETAILS"
echo "=========================================="
echo "Retrieving application details..."

echo "Executing: aws pinpoint get-app --application-id $APP_ID"
GET_APP_OUTPUT=$(aws pinpoint get-app --application-id "$APP_ID" 2>&1)
if check_error "$GET_APP_OUTPUT" "get-app"; then
    echo "Application details:"
    echo "$GET_APP_OUTPUT"
fi

# Display summary of created resources
echo ""
echo "=========================================="
echo "RESOURCES CREATED"
echo "=========================================="
echo "AWS Resources:"
for resource in "${AWS_RESOURCES[@]+"${AWS_RESOURCES[@]}"}"; do
    echo "- Application: $resource"
done

echo ""
echo "Files created:"
echo "- $LOG_FILE (script log)"
```

```
# Auto-cleanup information
echo ""
echo "============================================="
echo "CLEANUP INFORMATION"
echo "============================================="
echo "This script created AWS resources that may incur charges."
echo "AWS resources will be automatically cleaned up on script exit."
echo ""
echo "To manually delete resources later, use:"
echo "  aws pinpoint delete-app --application-id $APP_ID"

# Set flag to delete AWS resources on cleanup
DELETE_AWS_RESOURCES=true

echo ""
echo "============================================="
echo "SCRIPT COMPLETED SUCCESSFULLY"
echo "============================================="
echo "This script demonstrated:"
echo "1. Creating an AWS End User Messaging Push application"
echo "2. Attempting to enable FCM and APNS channels (with placeholder
 credentials)"
echo "3. Creating message templates for different platforms"
echo "4. Demonstrating message sending commands (with placeholder tokens)"
echo "5. Retrieving application details"
echo "6. Proper cleanup of resources"
echo ""
echo "Security best practices implemented:"
echo "- Secure temporary file handling with restricted permissions"
echo "- Input validation for application names"
echo "- Error handling with proper exit codes"
echo "- Credential validation before AWS operations"
echo "- Automatic cleanup on script exit"
echo "- Secure file destruction for sensitive data"
echo ""
echo "For production use:"
echo "- Replace placeholder API keys with real FCM server keys"
echo "- Replace placeholder certificates with real APNS certificates"
echo "- Replace placeholder device tokens with real device tokens"
echo "- Use AWS IAM roles instead of long-term credentials"
echo "- Implement comprehensive error handling"
echo "- Store sensitive credentials in AWS Secrets Manager or Parameter Store"
echo "- Enable CloudTrail for audit logging"
```

```
echo "- Use VPC endpoints for private AWS API access"
echo ""
echo "Log file: $LOG_FILE"
echo "Script completed at: $(date)"
```

- 如需 API 詳細資訊，請參閱《AWS CLI命令參考》中的下列主題。
  - CreateApp
  - DeleteApp
  - GetApp
  - GetApps
  - GetCallerIdentity
  - SendMessages
  - UpdateApnsChannel
  - UpdateGcmChannel

如需AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

使用 SDK 的 Amazon Pinpoint SMS 和語音 API 程式碼範例 AWS SDKs

下列程式碼範例示範如何搭配 AWS 軟體開發套件 (SDK) 使用 Amazon Pinpoint SMS 和語音 API。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

Amazon Pinpoint SMS 和語音 API

- 使用 AWS SDKs 的 Amazon Pinpoint SMS 和語音 API 基本範例
  - 使用 AWS SDKs 的 Amazon Pinpoint SMS 和語音 API 動作
    - CreateConfigurationSet 搭配 AWS SDK 使用
    - CreateConfigurationSetEventDestination 搭配 AWS SDK 使用
    - DeleteConfigurationSet 搭配 AWS SDK 使用

- DeleteConfigurationSetEventDestination 搭配 AWS SDK 使用
- GetConfigurationSetEventDestinations 搭配 AWS SDK 使用
- ListConfigurationSets 搭配 AWS SDK 使用
- SendVoiceMessage 搭配 AWS SDK 使用
- UpdateConfigurationSetEventDestination 搭配 AWS SDK 使用

使用 AWS SDKs 的 Amazon Pinpoint SMS 和語音 API 基本範例

以下程式碼範例示範如何搭配 AWS SDK 使用 Amazon Pinpoint SMS 和語音 API 的基本功能。

範例

- 使用 AWS SDKs 的 Amazon Pinpoint SMS 和語音 API 動作
  - CreateConfigurationSet 搭配 AWS SDK 使用
  - CreateConfigurationSetEventDestination 搭配 AWS SDK 使用
  - DeleteConfigurationSet 搭配 AWS SDK 使用
  - DeleteConfigurationSetEventDestination 搭配 AWS SDK 使用
  - GetConfigurationSetEventDestinations 搭配 AWS SDK 使用
  - ListConfigurationSets 搭配 AWS SDK 使用
  - SendVoiceMessage 搭配 AWS SDK 使用
  - UpdateConfigurationSetEventDestination 搭配 AWS SDK 使用

使用 AWS SDKs 的 Amazon Pinpoint SMS 和語音 API 動作

下列程式碼範例示範如何使用 AWS SDKs 執行個別 Amazon Pinpoint SMS 和語音 API 動作。每個範例均包含 GitHub 的連結，您可以在連結中找到設定和執行程式碼的相關說明。

下列範例僅包含最常使用的動作。如需完整清單，請參閱 Amazon Pinpoint SMS 和語音 API API 參考。

範例

- CreateConfigurationSet 搭配 AWS SDK 使用
- CreateConfigurationSetEventDestination 搭配 AWS SDK 使用
- DeleteConfigurationSet 搭配 AWS SDK 使用
- DeleteConfigurationSetEventDestination 搭配 AWS SDK 使用

- GetConfigurationSetEventDestinations 搭配 AWS SDK 使用
- ListConfigurationSets 搭配 AWS SDK 使用
- SendVoiceMessage 搭配 AWS SDK 使用
- UpdateConfigurationSetEventDestination 搭配 AWS SDK 使用

CreateConfigurationSet 搭配 AWS SDK 使用

以下程式碼範例顯示如何使用 CreateConfigurationSet。

SAP ABAP

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Create a new configuration set
        lo_pps->createconfigurationset(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
        ).

        MESSAGE 'Configuration set created successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsalreadyexistsex INTO DATA(lo_already_exists_ex).
        MESSAGE lo_already_exists_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_already_exists_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppslimitexceededex INTO DATA(lo_limit_exceeded_ex).
        MESSAGE lo_limit_exceeded_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_limit_exceeded_ex.
```

```
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateConfigurationSet。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

`CreateConfigurationSetEventDestination` 搭配 AWS SDK 使用

以下程式碼範例顯示如何使用 `CreateConfigurationSetEventDestination`。

SAP ABAP

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Create event destination for the configuration set
        lo_pps->createconfseteventdst(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
          iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-
event-dest'
          io_eventdestination = io_event_destination
        ).

        MESSAGE 'Event destination created successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsalreadyexistsex INTO DATA(lo_already_exists_ex).
        MESSAGE lo_already_exists_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_already_exists_ex.
```

```
    CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
      MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_not_found_ex.
    CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
      MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_bad_request_ex.
    CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
      MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_internal_error_ex.
    CATCH /aws1/cx_ppslimitexceededex INTO DATA(lo_limit_exceeded_ex).
      MESSAGE lo_limit_exceeded_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_limit_exceeded_ex.
    CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
      MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_too_many_requests_ex.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 CreateConfigurationSetEventDestination。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

`DeleteConfigurationSet` 搭配 AWS SDK 使用

以下程式碼範例顯示如何使用 `DeleteConfigurationSet`。

SAP ABAP

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
  TRY.
      " Delete the configuration set
      lo_pps->deleteconfigurationset(
```

```
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
        ).

        MESSAGE 'Configuration set deleted successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteConfigurationSet。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

DeleteConfigurationSetEventDestination 搭配 AWS SDK 使用

以下程式碼範例顯示如何使用 `DeleteConfigurationSetEventDestination`。

SAP ABAP

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Delete the event destination
        lo_pps->deleteconfseteventdst(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
          iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-
event-dest'
        ).

        MESSAGE 'Event destination deleted successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 DeleteConfigurationSetEventDestination。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

GetConfigurationSetEventDestinations 搭配 AWS SDK 使用

以下程式碼範例顯示如何使用 GetConfigurationSetEventDestinations。

SAP ABAP

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Get event destinations for the configuration set
        oo_result = lo_pps->getconfseteventdestinations(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
        ).

        " Process the event destinations
        LOOP AT oo_result->get_eventdestinations( ) INTO DATA(lo_event_dest).
          DATA(lv_dest_name) = lo_event_dest->get_name( ).
          DATA(lv_enabled) = lo_event_dest->get_enabled( ).

          MESSAGE |Event destination: { lv_dest_name }, Enabled: { lv_enabled }|
 TYPE 'I'.

          " Check for CloudWatch Logs destination
          DATA(lo_cloudwatch_dest) = lo_event_dest->get_cloudwatchlogsdst( ).
          IF lo_cloudwatch_dest IS NOT INITIAL.
            DATA(lv_log_group_arn) = lo_cloudwatch_dest->get_loggrouparn( ).
            MESSAGE |  CloudWatch Logs destination: { lv_log_group_arn }| TYPE
 'I'.
          ENDIF.

          " Check for Kinesis Firehose destination
          DATA(lo_firehose_dest) = lo_event_dest->get_kinesisfirehosedst( ).
          IF lo_firehose_dest IS NOT INITIAL.
            DATA(lv_delivery_stream) = lo_firehose_dest-
>get_deliverystreamarn( ).
            MESSAGE |  Kinesis Firehose destination: { lv_delivery_stream }| TYPE
 'I'.
          ENDIF.
```

```
          " Check for SNS destination
          DATA(lo_sns_dest) = lo_event_dest->get_snsdestination( ).
          IF lo_sns_dest IS NOT INITIAL.
            DATA(lv_topic_arn) = lo_sns_dest->get_topicarn( ).
            MESSAGE |  SNS destination: { lv_topic_arn }| TYPE 'I'.
          ENDIF.
        ENDLOOP.

      CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
        MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_not_found_ex.
      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 GetConfigurationSetEventDestinations。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

`ListConfigurationSets` 搭配 AWS SDK 使用

以下程式碼範例顯示如何使用 `ListConfigurationSets`。

SAP ABAP

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " List all configuration sets
        oo_result = lo_pps->listconfigurationsets(
          iv_nexttoken = iv_next_token    " Optional: Token for pagination
          iv_pagesize = iv_page_size      " Optional: Number of results per page,
e.g., '10'
        ).

        " Process the configuration sets
        LOOP AT oo_result->get_configurationsets( ) INTO DATA(lo_config_set).
          DATA(lv_config_set_name) = lo_config_set->get_value( ).
          MESSAGE |Configuration set: { lv_config_set_name }| TYPE 'I'.
        ENDLOOP.

        " Check if there are more results
        DATA(lv_next_token) = oo_result->get_nexttoken( ).
        IF lv_next_token IS NOT INITIAL.
          MESSAGE |More results available. Next token: { lv_next_token }| TYPE
'I'.
        ENDIF.

      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 ListConfigurationSets。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

SendVoiceMessage 搭配 AWS SDK 使用

下列程式碼範例示範如何使用 SendVoiceMessage。

Java

適用於 Java 2.x 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
import software.amazon.awssdk.core.client.config.ClientOverrideConfiguration;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.pinpointsmsvoice.PinpointSmsVoiceClient;
import software.amazon.awssdk.services.pinpointsmsvoice.model.SSMLMessageType;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.VoiceMessageContent;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.SendVoiceMessageRequest;
import
 software.amazon.awssdk.services.pinpointsmsvoice.model.PinpointSmsVoiceException;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 * <p>
 * For more information, see the following documentation topic:
 * <p>
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class SendVoiceMessage {

    // The Amazon Polly voice that you want to use to send the message. For a
 list
    // of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
    static final String voiceName = "Matthew";
```

```
   // The language to use when sending the message. For a list of supported
   // languages, see
   // https://docs.aws.amazon.com/polly/latest/dg/SupportedLanguage.html
   static final String languageCode = "en-US";

   // The content of the message. This example uses SSML to customize and
control
   // certain aspects of the message, such as by adding pauses and changing
   // phonation. The message can't contain any line breaks.
   static final String ssmlMessage = "<speak>This is a test message sent from "
           + "<emphasis>Amazon Pinpoint</emphasis> "
           + "using the <break strength='weak'/>AWS "
           + "SDK for Java. "
           + "<amazon:effect phonation='soft'>Thank "
           + "you for listening.</amazon:effect></speak>";

   public static void main(String[] args) {

       final String usage = """
               Usage:   <originationNumber> <destinationNumber>\s

               Where:
                 originationNumber - The phone number or short code that
you specify has to be associated with your Amazon Pinpoint account. For best
results, specify long codes in E.164 format (for example, +1-555-555-5654).
                 destinationNumber - The recipient's phone number.  For best
results, you should specify the phone number in E.164 format (for example,
+1-555-555-5654).\s
               """;

       if (args.length != 2) {
           System.out.println(usage);
           System.exit(1);
       }
       String originationNumber = args[0];
       String destinationNumber = args[1];
       System.out.println("Sending a voice message");

       // Set the content type to application/json.
       List<String> listVal = new ArrayList<>();
       listVal.add("application/json");
       Map<String, List<String>> values = new HashMap<>();
       values.put("Content-Type", listVal);
```

```
        ClientOverrideConfiguration config2 =
ClientOverrideConfiguration.builder()
                .headers(values)
                .build();

        PinpointSmsVoiceClient client = PinpointSmsVoiceClient.builder()
                .overrideConfiguration(config2)
                .region(Region.US_EAST_1)
                .build();

        sendVoiceMsg(client, originationNumber, destinationNumber);
        client.close();
    }

    public static void sendVoiceMsg(PinpointSmsVoiceClient client, String
originationNumber,
                                    String destinationNumber) {
        try {
            SSMLMessageType ssmlMessageType = SSMLMessageType.builder()
                    .languageCode(languageCode)
                    .text(ssmlMessage)
                    .voiceId(voiceName)
                    .build();

            VoiceMessageContent content = VoiceMessageContent.builder()
                    .ssmlMessage(ssmlMessageType)
                    .build();

            SendVoiceMessageRequest voiceMessageRequest =
SendVoiceMessageRequest.builder()
                    .destinationPhoneNumber(destinationNumber)
                    .originationPhoneNumber(originationNumber)
                    .content(content)
                    .build();

            client.sendVoiceMessage(voiceMessageRequest);
            System.out.println("The message was sent successfully.");

        } catch (PinpointSmsVoiceException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

```
}
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Java 2.x API 參考中的 SendVoiceMessage。

JavaScript

適用於 JavaScript (v2) 的 SDK

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
"use strict";

var AWS = require("aws-sdk");

// The AWS Region that you want to use to send the voice message. For a list of
// AWS Regions where the Amazon Pinpoint SMS and Voice API is available, see
// https://docs.aws.amazon.com/pinpoint-sms-voice/latest/APIReference/
var aws_region = "us-east-1";

// The phone number that the message is sent from. The phone number that you
// specify has to be associated with your Amazon Pinpoint account. For best
 results, you
// should specify the phone number in E.164 format.
var originationNumber = "+12065550110";

// The recipient's phone number. For best results, you should specify the phone
// number in E.164 format.
var destinationNumber = "+12065550142";

// The language to use when sending the message. For a list of supported
// languages, see https://docs.aws.amazon.com/polly/latest/dg/
SupportedLanguage.html
var languageCode = "en-US";

// The Amazon Polly voice that you want to use to send the message. For a list
// of voices, see https://docs.aws.amazon.com/polly/latest/dg/voicelist.html
```

```
var voiceId = "Matthew";

// The content of the message. This example uses SSML to customize and control
// certain aspects of the message, such as the volume or the speech rate.
// The message can't contain any line breaks.
var ssmlMessage =
  "<speak>" +
  "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> " +
  "using the <break strength='weak'/>AWS SDK for JavaScript in Node.js. " +
  "<amazon:effect phonation='soft'>Thank you for listening." +
  "</amazon:effect>" +
  "</speak>";

// The phone number that you want to appear on the recipient's device. The phone
// number that you specify has to be associated with your Amazon Pinpoint
 account.
var callerId = "+12065550199";

// The configuration set that you want to use to send the message.
var configurationSet = "ConfigSet";

// Specify that you're using a shared credentials file, and optionally specify
// the profile that you want to use.
var credentials = new AWS.SharedIniFileCredentials({ profile: "default" });
AWS.config.credentials = credentials;

// Specify the region.
AWS.config.update({ region: aws_region });

//Create a new Pinpoint object.
var pinpointsmsvoice = new AWS.PinpointSMSVoice();

var params = {
  CallerId: callerId,
  ConfigurationSetName: configurationSet,
  Content: {
    SSMLMessage: {
      LanguageCode: languageCode,
      Text: ssmlMessage,
      VoiceId: voiceId,
    },
  },
  DestinationPhoneNumber: destinationNumber,
  OriginationPhoneNumber: originationNumber,
```

```
};

//Try to send the message.
pinpointsmsvoice.sendVoiceMessage(params, function (err, data) {
  // If something goes wrong, print an error message.
  if (err) {
    console.log(err.message);
    // Otherwise, show the unique ID for the message.
  } else {
    console.log("Message sent! Message ID: " + data["MessageId"]);
  }
});
```

- 如需 API 詳細資訊，請參閱 適用於 JavaScript 的 AWS SDK API 參考中的 SendVoiceMessage。

Python

適用於 Python 的 SDK (Boto3)

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
import logging
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


def send_voice_message(
    sms_voice_client,
    origination_number,
    caller_id,
    destination_number,
```

```
    language_code,
    voice_id,
    ssml_message,
):
    """
    Sends a voice message using speech synthesis provided by Amazon Polly.

    :param sms_voice_client: A Boto3 PinpointSMSVoice client.
    :param origination_number: The phone number that the message is sent from.
                               The phone number must be associated with your
Amazon
                               Pinpoint account and be in E.164 format.
    :param caller_id: The phone number that you want to appear on the recipient's
                      device. The phone number must be associated with your
Amazon
                      Pinpoint account and be in E.164 format.
    :param destination_number: The recipient's phone number. Specify the phone
                               number in E.164 format.
    :param language_code: The language to use when sending the message.
    :param voice_id: The Amazon Polly voice that you want to use to send the
message.
    :param ssml_message: The content of the message. This example uses SSML to
control
                         certain aspects of the message, such as the volume and
the
                         speech rate. The message must not contain line breaks.
    :return: The ID of the message.
    """
    try:
        response = sms_voice_client.send_voice_message(
            DestinationPhoneNumber=destination_number,
            OriginationPhoneNumber=origination_number,
            CallerId=caller_id,
            Content={
                "SSMLMessage": {
                    "LanguageCode": language_code,
                    "VoiceId": voice_id,
                    "Text": ssml_message,
                }
            },
        )
    except ClientError:
        logger.exception(
            "Couldn't send message from %s to %s.",
```

```
            origination_number,
            destination_number,
        )
        raise
    else:
        return response["MessageId"]


def main():
    origination_number = "+12065550110"
    caller_id = "+12065550199"
    destination_number = "+12065550142"
    language_code = "en-US"
    voice_id = "Matthew"
    ssml_message = (
        "<speak>"
        "This is a test message sent from <emphasis>Amazon Pinpoint</emphasis> "
        "using the <break strength='weak'/>AWS SDK for Python (Boto3). "
        "<amazon:effect phonation='soft'>Thank you for listening."
        "</amazon:effect>"
        "</speak>"
    )
    print(f"Sending voice message from {origination_number} to
 {destination_number}.")
    message_id = send_voice_message(
        boto3.client("pinpoint-sms-voice"),
        origination_number,
        caller_id,
        destination_number,
        language_code,
        voice_id,
        ssml_message,
    )
    print(f"Message sent!\nMessage ID: {message_id}")


if __name__ == "__main__":
    main()
```

- 如需 API 詳細資訊，請參閱 AWS SDK for Python (Boto3) API 參考中的 SendVoiceMessage。

SAP ABAP

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Create SSML message type object with voice parameters
        DATA(lo_ssml_message) = NEW /aws1/cl_ppsssmlmessagetype(
          iv_languagecode = iv_language_code    " e.g., 'en-US'
          iv_voiceid = iv_voice_id              " e.g., 'Matthew'
          iv_text = iv_ssml_message             " SSML formatted message text
        ).

        " Create voice message content with the SSML message
        DATA(lo_content) = NEW /aws1/cl_ppsvoicemessagecont(
          io_ssmlmessage = lo_ssml_message
        ).

        " Send the voice message
        DATA(lo_result) = lo_pps->sendvoicemessage(
          iv_originationphonenumber = iv_origination_number  " e.g.,
'+12065550110'
          iv_callerid = iv_caller_id                         " e.g.,
'+12065550199'
          iv_destinationphonenumber = iv_destination_number  " e.g.,
'+12065550142'
          io_content = lo_content
        ).

        " Retrieve the message ID from the response
        ov_message_id = lo_result->get_messageid( ).

        MESSAGE 'Voice message sent successfully.' TYPE 'I'.

      CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
        MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_bad_request_ex.
```

```
      CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
        MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_internal_error_ex.
      CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
        MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
        RAISE EXCEPTION lo_too_many_requests_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 SendVoiceMessage。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

UpdateConfigurationSetEventDestination 搭配 AWS SDK 使用

以下程式碼範例顯示如何使用 UpdateConfigurationSetEventDestination。

SAP ABAP

適用於 SAP ABAP 的開發套件

> **Note**
>
> GitHub 上提供更多範例。尋找完整範例，並了解如何在 AWS 程式碼範例儲存庫中設定和執行。

```
    TRY.
        " Update the event destination
        lo_pps->updateconfseteventdst(
          iv_configurationsetname = iv_configuration_set_name    " e.g., 'my-
config-set'
          iv_eventdestinationname = iv_event_destination_name    " e.g., 'my-
event-dest'
          io_eventdestination = io_event_destination
        ).

        MESSAGE 'Event destination updated successfully.' TYPE 'I'.
```

```
    CATCH /aws1/cx_ppsnotfoundexception INTO DATA(lo_not_found_ex).
      MESSAGE lo_not_found_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_not_found_ex.
    CATCH /aws1/cx_ppsbadrequestex INTO DATA(lo_bad_request_ex).
      MESSAGE lo_bad_request_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_bad_request_ex.
    CATCH /aws1/cx_ppsinternalsvcerrorex INTO DATA(lo_internal_error_ex).
      MESSAGE lo_internal_error_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_internal_error_ex.
    CATCH /aws1/cx_ppstoomanyrequestsex INTO DATA(lo_too_many_requests_ex).
      MESSAGE lo_too_many_requests_ex->get_text( ) TYPE 'I'.
      RAISE EXCEPTION lo_too_many_requests_ex.
  ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 UpdateConfigurationSetEventDestination。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 搭配 AWS SDK 使用 Amazon Pinpoint。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

Amazon Pinpoint 的安全性

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構專為滿足最安全敏感組織的需求而建置。

安全性是 AWS 與您之間的共同責任。共同責任模式將其描述為雲端的安全性，和雲端中的安全性：

- 雲端的安全性 – AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。 AWS 也為您提供可安全使用的服務。作為AWS 合規計劃的一部分，第三方稽核人員會定期測試和驗證我們安全的有效性。若要了解適用於 Amazon Pinpoint 的合規計劃，請參閱AWS 合規計劃的 服務範圍。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件有助您了解如何在使用 Amazon Pinpoint 時，套用共同責任模式。以下主題說明如何將 Amazon Pinpoint 設成符合您的安全性與合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 Amazon Pinpoint 資源。

如需詳細資訊，請參閱有關參考架構的資訊，請參閱 Amazon Pinpoint 彈性參考架構。

主題

- Amazon Pinpoint 中的資料保護
- Amazon Pinpoint 的身分和存取管理
- Amazon Pinpoint 中的記錄和監控
- Amazon Pinpoint 的合規驗證
- Amazon Pinpoint 的恢復能力
- Amazon Pinpoint 的基礎設施安全性
- Amazon Pinpoint 中的組態與漏洞分析
- Amazon Pinpoint 的安全最佳做法

Amazon Pinpoint 中的資料保護

共同責任模型適用於 AWS Amazon Pinpoint 中的資料保護。如此模型所述， AWS 負責保護執行所有 的全域基礎設施 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務 的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。 如需歐洲資料保護的詳細資訊，請參閱一般資料保護規則 (GDPR) 中心。

基於資料保護目的，建議您保護 AWS 帳戶 登入資料，並使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 設定個別使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用 設定 API 和使用者活動記錄 AWS CloudTrail。如需有關使用 CloudTrail 追蹤擷取 AWS 活動的資訊，請參閱AWS CloudTrail 《 使用者指南》中的使用 CloudTrail 追蹤。
- 使用 AWS 加密解決方案，以及其中的所有預設安全控制 AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在 AWS 透過命令列界面或 API 存取 時需要 FIPS 140-3 驗證的密碼編譯模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱聯邦資訊處理標準 (FIPS) 140-3。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用 Amazon Pinpoint 或使用主控台、API AWS CLI或 AWS SDKs的其他 AWS 服務 時。您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

Amazon Pinpoint 可能會儲存您或您客戶的以下個人資料類型，視您設定及使用服務的方式而定：

組態資料

這包括憑證和設定等專案組態資料，這些專案組態資料定義了 Amazon Pinpoint 透過支援的管道傳送訊息的方式和時間，以及訊息傳送的使用者客群。若要傳送訊息，此資料可包括電子郵件訊息專用的 IP 地址、SMS 文字訊息的短碼和寄件者 ID，以及與推播通知服務通訊的憑證，例如 Apple 推播通知服務 (APN) 和 Firebase Cloud Messaging (FCM)。

使用者和端點資料

這包括您用來儲存並管理 Amazon Pinpoint 專案使用者和端點相關資料的標準和自訂屬性。屬性可以儲存特定使用者的相關資訊 (例如使用者名稱) 或使用者特定端點的相關資訊 (例如使用者的電子郵件地址、行動電話號碼或行動裝置字符)。此資料也可以包括將 Amazon Pinpoint 專案的使用者與外部系統中的使用者相互關聯的外部使用者 ID (例如客戶關係管理系統)。如需此資料可包含哪些內容的詳細資訊，請參閱 Amazon Pinpoint API Reference 中的使用者和端點結構描述。

分析資料

這包括指標的資料 (也稱為關鍵績效指標 (KPI))，可讓您深入了解 Amazon Pinpoint 專案在使用者互動和購買活動等領域的表現。這也包括可深入了解專案使用者人口統計資料的指標資料。資料可

以從使用者和端點的標準和自訂屬性衍生出來 (例如使用者居住的城市)。它也可以從事件衍生，例如您為專案傳送之電子郵件訊息的開啟和點按事件。

匯入的資料

這包括您從外部來源新增或匯入，並在 Amazon Pinpoint 中使用的任何使用者、區隔和分析資料。例如您匯入 Amazon Pinpoint (直接透過主控台或從 Amazon S3 儲存貯體匯入)，以建置靜態客群的 JSON 檔案。其他範例還有以程式化方式新增並建立動態客群的端點資料、接收直接訊息的端點地址，以及您將應用程式設為向 Amazon Pinpoint 回報的事件。

主題

- 資料加密
- 網際網路流量隱私權
- 為 Amazon Pinpoint 建立介面 VPC 端點

資料加密

Amazon Pinpoint 會對傳輸中的資料和靜態資料加密。若將資料提交給 Amazon Pinpoint，Amazon Pinpoint 在接收資料和儲存資料時，都會對資料加密。您從 Amazon Pinpoint 擷取資料時，Amazon Pinpoint 會使用目前的安全協定將資料傳輸給您。

靜態加密

Amazon Pinpoint 為您儲存的所有資料都會經過加密。這包括組態資料、使用者和端點資料、分析資料，以及您新增或匯入 Amazon Pinpoint 的任何資料。為了加密您的資料，Amazon Pinpoint 會使用服務代表您擁有和維護的內部 AWS Key Management Service (AWS KMS) 金鑰。我們會定期輪換這些金鑰。如需 的詳細資訊 AWS KMS，請參閱 AWS Key Management Service 開發人員指南。

傳輸中加密

Amazon Pinpoint 使用 HTTPS 和 Transport Layer Security (TLS) 1.2 (含) 以上版本，與您的用戶端和應用程式通訊。為了與其他 AWS 服務通訊，Amazon Pinpoint 使用 HTTPS 和 TLS 1.2。此外，當您使用 主控台、 AWS 開發套件或 建立和管理 Amazon Pinpoint 資源時 AWS Command Line Interface，所有通訊都會使用 HTTPS 和 TLS 1.2 進行保護。

金鑰管理

為了加密 Amazon Pinpoint 資料，Amazon Pinpoint 會使用服務代表您擁有和維護的內部 AWS KMS 金鑰。我們會定期輪換這些金鑰。您無法佈建和使用自己的 AWS KMS 金鑰或其他金鑰來加密存放在 Amazon Pinpoint 中的資料。

網際網路流量隱私權

網際網路流量隱私權是指保護 Amazon Pinpoint 與內部部署用戶端和應用程式之間的連線和流量，以及 Amazon Pinpoint 與相同 AWS 區域中的其他 AWS 資源之間的連線和流量。以下功能和實務可協助您確保 Amazon Pinpoint 的網際網路流量隱私權。

Amazon Pinpoint 和內部部署用戶端與應用程式之間的流量

若要在內部部署網路上建立 Amazon Pinpoint 與用戶端和應用程式之間的私有連線，您可以使用 Direct Connect。這可讓您使用標準光纖乙太網路纜線將網路連結至某個 AWS Direct Connect 位置。纜線的一端連接到路由器。另一端連接到 Direct Connect 路由器。如需詳細資訊，請參閱《 Direct Connect 使用者指南》中的什麼是Direct Connect ？。

為了透過已發布的 API 安全存取 Amazon Pinpoint，進行 API 呼叫時，建議遵守 Amazon Pinpoint 要求。Amazon Pinpoint 要求用戶端使用 Transport Layer Security (TLS) 1.2 (含) 以上版本。用戶端也必須支援具備完整轉寄密碼 (PFS) 的密碼套件，例如暫時性 Diffie-Hellman (DHE) 或橢圓曲線 Diffie-Hellman Ephemeral (ECDHE)。現代系統 (如 Java 7 和更新版本) 大多會支援這些模式。

此外，必須使用存取金鑰 ID 和與您 AWS 帳戶之 AWS Identity and Access Management (IAM) 主體相關聯的私密存取金鑰來簽署請求。或者，您可以使用 AWS Security Token Service (AWS STS) 來產生暫時安全登入資料來簽署請求。

Amazon Pinpoint 與其他 AWS 資源之間的流量

為了保護 Amazon Pinpoint 與相同 AWS 區域中其他 AWS 資源之間的通訊，Amazon Pinpoint 預設會使用 HTTPS 和 TLS 1.2。

為 Amazon Pinpoint 建立介面 VPC 端點

建立介面 VPC 端點，就能在虛擬私有雲端 (VPC) 和 Amazon Pinpoint 的端點之間建立私有連線。

介面端點採用 AWS PrivateLink技術，可讓您在沒有網際網路閘道、NAT 裝置、VPN 連接或 的情況下，私下存取 Amazon Pinpoint APIs Direct Connect。VPC 中的執行個體不需要公有 IP 地址，即可透過與 AWS PrivateLink整合的 Amazon Pinpoint API 通訊。

如需詳細資訊，請參閱 AWS PrivateLink 指南。

建立介面 VPC 端點

您可以使用 Amazon VPC 主控台或 AWS Command Line Interface () 建立介面端點AWS CLI。如需詳細資訊，請參閱《 AWS PrivateLink 指南》中的建立介面端點。

Amazon Pinpoint 支援以下服務名稱：

- `com.amazonaws.region.pinpoint`
- `com.amazonaws.region.pinpoint-sms-voice-v2`

如果您開啟介面端點的私有 DNS，您可以使用 的預設 DNS 名稱向 Amazon Pinpoint 提出 API 請求 AWS 區域，例如 `com.amazonaws.us-east-1.pinpoint`。詳情請參閱 AWS PrivateLink 指南中的 DNS 主機名稱。

如需目前可使用 Amazon Pinpoint 的所有區域和端點的清單，請參閱 Amazon Web Services 一般參考中的 AWS 服務端點。

建立 VPC 端點政策

您可以將端點政策連接到控制存取權的 VPC 端點。此政策會指定下列資訊：

- 可執行動作的主體。
- 可執行的動作。
- 可供執行動作的資源。

如需詳細資訊，請參閱《AWS PrivateLink 指南》中的「使用端點政策控制對服務的存取」。

範例：VPC 端點政策

以下 VPC 端點政策為所有資源上的所有主體授予存取權，讓他們存取所列的 Amazon Pinpoint 動作。

```
{
"Statement": [
    {
      "Principal": "*",
      "Action": [
```

```
        "mobiletargeting:CreateCampaign",
        "mobiletargeting:CreateApp",
        "mobiletargeting:DeleteApp",
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Amazon Pinpoint 的身分和存取管理

AWS Identity and Access Management (IAM) 是一種 AWS 服務 ，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員可控制哪些人員符合已進行身分驗證 (已登入) 身分及已完成身分驗證 (具有許可) 身分，可以使用 Amazon Pinpoint 資源。IAM 是您可以免費使用 AWS 服務 的 。

主題

- 目標對象
- 使用身分驗證
- 使用政策管理存取權
- Amazon Pinpoint 如何搭配 IAM 運作
- IAM 政策的 Amazon Pinpoint 動作
- Amazon Pinpoint 身分型政策範例
- 一般 Amazon Pinpoint 任務的 IAM 角色
- 疑難排解 Amazon Pinpoint 身分和存取管理

目標對象

使用方式 AWS Identity and Access Management (IAM) 會根據您的角色而有所不同：

- 服務使用者 — 若無法存取某些功能，請向管理員申請所需許可 (請參閱 疑難排解 Amazon Pinpoint 身分和存取管理)
- 服務管理員 — 負責設定使用者存取權並提交相關許可請求 (請參閱 Amazon Pinpoint 如何搭配 IAM 運作)
- IAM 管理員 — 撰寫政策以管理存取控制 (請參閱 Amazon Pinpoint 身分型政策範例)

使用身分驗證

身分驗證是您 AWS 使用身分憑證登入 的方式。您必須以 AWS 帳戶根使用者、IAM 使用者或擔任 IAM 角色身分進行身分驗證。

您可以使用身分來源的登入資料，例如 AWS IAM Identity Center (IAM Identity Center)、單一登入身分驗證或 Google/Facebook 登入資料，以聯合身分的形式登入。如需有關登入的詳細資訊，請參閱《AWS 登入 使用者指南》中的如何登入您的 AWS 帳戶。

對於程式設計存取， AWS 提供 SDK 和 CLI 以密碼編譯方式簽署請求。如需詳細資訊，請參閱《IAM 使用者指南》中的 API 請求的AWS 第 4 版簽署程序。

AWS 帳戶 根使用者

當您建立 時 AWS 帳戶，您會從一個名為 AWS 帳戶 theroot 使用者的登入身分開始，該身分可完整存取所有 AWS 服務 和資源。強烈建議不要使用根使用者來執行日常任務。有關需要根使用者憑證的任務，請參閱《IAM 使用者指南》中的需要根使用者憑證的任務。

IAM 使用者和群組

IAM 使用者https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users.html是一種身分具備單人或應用程式的特定許可權。建議以臨時憑證取代具備長期憑證的 IAM 使用者。如需詳細資訊，請參閱《IAM 使用者指南》中的要求人類使用者使用聯合身分提供者來 AWS 使用臨時憑證存取 。

IAM 群組會指定 IAM 使用者集合，使管理大量使用者的許可權更加輕鬆。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 使用者的使用案例。

IAM 角色

IAM 角色https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html的身分具有特定許可權，其可以提供臨時憑證。您可以透過從使用者切換到 IAM 角色 （主控台） 或呼叫 AWS CLI 或 AWS API 操作來擔任角色。如需詳細資訊，請參閱《IAM 使用者指南》中的擔任角色的方法。

IAM 角色適用於聯合身分使用者存取、臨時 IAM 使用者許可、跨帳戶存取權與跨服務存取，以及在 Amazon EC2 執行的應用程式。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 中的快帳戶資源存取。

使用政策管理存取權

您可以透過建立政策並將其連接到身分或資源 AWS 來控制 AWS 中的存取。政策定義與身分或資源相關聯的許可。當委託人提出請求時 AWS ， 會評估這些政策。大多數政策會以 JSON 文件 AWS 形式存放在 中。如需進一步了解 JSON 政策文件，請參閱《IAM 使用者指南》中的 JSON 政策概觀。

管理員會使用政策，透過定義哪些主體可在哪些條件下對哪些資源執行動作，以指定可存取的範圍。

預設情況下，使用者和角色沒有許可。IAM 管理員會建立 IAM 政策並將其新增至角色，供使用者後續擔任。IAM 政策定義動作的許可，無論採用何種方式執行。

身分型政策

身分型政策是附加至身分 (使用者、使用者群組或角色) 的 JSON 許可政策文件。這類政策控制身分可對哪些資源執行哪些動作，以及適用的條件。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

身分型政策可分為內嵌政策 (直接內嵌於單一身分) 與受管政策 (可附加至多個身分的獨立政策)。如需了解如何在受管政策及內嵌政策之間做選擇，請參閱《IAM 使用者指南》中的在受管政策與內嵌政策之間選擇。

Amazon Pinpoint 支援以身分型政策控制 Amazon Pinpoint 資源的存取權。

資源型政策

資源型政策是附加到資源的 JSON 政策文件。範例包括 IAM 角色信任政策與 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。您必須在資源型政策中指定主體。

資源型政策是位於該服務中的內嵌政策。您無法在資源型政策中使用來自 IAM 的 AWS 受管政策。

Amazon Pinpoint 支援以資源型政策控制 Amazon Pinpoint 資源的存取權。

存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

Amazon S3 AWS WAF和 Amazon VPC 是支援 ACLs的服務範例。如需進一步了解 ACL，請參閱《Amazon Simple Storage Service 開發人員指南》中的存取控制清單 (ACL) 概觀。

Amazon Pinpoint 不支援以 ACL 控制 Amazon Pinpoint 資源的存取權。

其他政策類型

AWS 支援其他政策類型，可設定更多常見政策類型授予的最大許可：

- 許可界限 — 設定身分型政策可授與 IAM 實體的最大許可。如需詳細資訊，請參閱《 IAM 使用者指南》中的 IAM 實體許可界限。
- 服務控制政策 (SCP) — 為 AWS Organizations中的組織或組織單位指定最大許可。如需詳細資訊，請參閱《AWS Organizations 使用者指南》中的服務控制政策。
- 資源控制政策 (RCP) — 設定您帳戶中資源可用許可的上限。如需詳細資訊，請參閱《AWS Organizations 使用者指南》中的資源控制政策 (RCP)。
- 工作階段政策 — 在以程式設計方式為角色或聯合身分使用者建立臨時工作階段時，以參數形式傳遞的進階政策。如需詳細資訊，請參《IAM 使用者指南》中的工作階段政策。

Amazon Pinpoint 支援以這類政策控制 Amazon Pinpoint 資源的存取權。

多種政策類型

當多種類型的政策適用於請求時，產生的許可會更複雜而無法理解。若要了解如何 AWS 在涉及多個政策類型時決定是否允許請求，請參閱《IAM 使用者指南》中的政策評估邏輯。

Amazon Pinpoint 如何搭配 IAM 運作

若要使用 Amazon Pinpoint，您 AWS 帳戶中的使用者需要允許他們檢視分析資料、建立專案、定義使用者客群、部署行銷活動等的許可。如果您將行動或 Web 應用程式整合至 Amazon Pinpoint，您應用程式的使用者也需要 Amazon Pinpoint 的存取權。此存取權可讓您的應用程式註冊端點，並向 Amazon Pinpoint 回報用量資料。若要授予 Amazon Pinpoint 功能的存取權，請建立 AWS Identity and Access Management (IAM) 政策，以允許 IAM 身分或 Amazon Pinpoint 資源的 Amazon Pinpoint 動作。

IAM 是一項服務，可協助管理員安全地控制對 AWS 資源的存取。IAM 政策包含的陳述式，允許或拒絕特定使用者執行特定動作，或允許或拒絕對資源執行特定動作。Amazon Pinpoint 提供一組動作，讓您在 IAM 政策中使用，為 Amazon Pinpoint 使用者和資源指定細微許可。這表示您可以授予適當的存取權給 Amazon Pinpoint，以避免建立過度的許可政策，而可能暴露重要資料或損及您的資源。例如您可以授予不受限制的存取權給 Amazon Pinpoint 管理員，並向只需要存取特定專案的個人授予唯讀存取權。

使用 IAM 管理 Amazon Pinpoint 的存取權之前，您應該了解哪些 IAM 功能可與 Amazon Pinpoint 搭配使用。若要全面了解 Amazon Pinpoint 和其他 AWS 服務如何與 IAM 搭配使用，請參閱《AWS IAM 使用者指南》中的與 IAM 搭配使用的 服務。

主題

- Amazon Pinpoint 身分型政策
- Amazon Pinpoint 資源型許可的政策
- 基於 Amazon Pinpoint 標籤的授權
- Amazon Pinpoint IAM 角色

Amazon Pinpoint 身分型政策

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。Amazon Pinpoint 支援特定動作、資源和條件索引鍵。若要了解您可以在 JSON 政策中使用的所有元素，請參閱 IAM 使用者指南中的 IAM JSON 政策元素參考。

動作

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

JSON 政策的 `Action` 元素描述您可以用來允許或拒絕政策中存取的動作。政策會使用動作來授予執行相關聯動作的許可。

這表示政策動作能夠控制使用者可以在 Amazon Pinpoint 主控台執行的動作，他們也可以直接使用 AWS SDKs、 AWS Command Line Interface (AWS CLI) 或 Amazon Pinpoint APIs，以程式設計方式控制使用者可以執行的動作。

Amazon Pinpoint 中的政策動作使用以下前置詞：

- **`mobiletargeting`** – 適用於衍生自 Amazon Pinpoint API 的動作，這是 Amazon Pinpoint 的主要 API。
- **`sms-voice`** – 適用於衍生自 Amazon Pinpoint SMS 和語音 API 的動作，Amazon Pinpoint SMS 和語音 API 是個補充 API，提供進階選項，讓您使用和管理 Amazon Pinpoint 中的 SMS 和語音管道。

例如，若要授與某人檢視專案所有客群相關資訊的許可 (相當於 Amazon Pinpoint API 中 `GetSegments` 操作的動作)，請在政策中加入 `mobiletargeting:GetSegments` 動作。政策陳述式

必須包含 `Action` 或 `NotAction` 元素。Amazon Pinpoint 定義了自己的一組動作，並說明使用者可以使用這些動作執行的任務。

若要在單一陳述式中指定多個動作，請用逗號分隔：

```
"Action": [
      "mobiletargeting:action1",
      "mobiletargeting:action2"
```

您也可以使用萬用字元 (*) 指定多個動作。例如，若要指定開頭是 `Get` 文字的所有動作，請包含以下動作：

```
"Action": "mobiletargeting:Get*"
```

但是，根據最佳實務，您應該定義遵循「最低權限」原則的政策。換句話說，您應建立其中只包含執行特定動作所需許可的政策。

關於可在 IAM 政策中使用的 Amazon Pinpoint 動作，請參閱 IAM 政策的 Amazon Pinpoint 動作 查看完整清單。

Resources

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

`Resource` JSON 政策元素可指定要套用動作的物件。最佳實務是使用其 Amazon Resource Name (ARN) 來指定資源。若動作不支援資源層級許可，使用萬用字元 (*) 表示該陳述式適用於所有資源。

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

例如 `mobiletargeting:GetSegments` 動作會針對與特定 Amazon Pinpoint 專案相關聯的所有客群，擷取相關資訊。您可以使用下列格式的 ARN 來識別專案：

```
arn:aws:mobiletargeting:${Region}:${Account}:apps/${projectId}
```

如需有關 ARN 格式的詳細資訊，請參閱《AWS 一般參考》中的 Amazon Resource Name (ARN)。

在 IAM 政策中，您可以為以下 Amazon Pinpoint 資源類型指定 ARN：

- 行銷活動
- 旅程
- 訊息範本 (在某些情況下稱為範本)
- 專案 (在某些情況下稱為 App 或應用程式)
- 建議者模型 (在某些情況下稱為建議者)
- 客群

例如，若要為具有專案 ID 810c7aab86d42fb2b56c8c966example 的專案建立政策陳述式，請使用下列 ARN：

```
"Resource": "arn:aws:mobiletargeting:us-
east-1:123456789012:apps/810c7aab86d42fb2b56c8c966example"
```

若要指定所有屬於特定帳戶的所有專案，請使用萬用字元 (*)：

```
"Resource": "arn:aws:mobiletargeting:us-east-1:123456789012:apps/*"
```

有些 Amazon Pinpoint 動作 (例如建立資源的特定動作)，不能在特定資源上執行。在那些情況下，您必須使用萬用字元 (*)：

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

在 IAM 政策中，您也可以為以下類型的 Amazon Pinpoint SMS 和語音資源指定 ARN：

- 組態集合
- 選擇退出清單
- 電話號碼
- 集區
- 寄件者 Id

例如要為有電話號碼 ID 為 phone-1234567890123456789012345678901 2 的電話號碼建立政策陳述式，請使用以下 ARN：

```
"Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/
phone-1234567890123456789012345678901 2"
```

若要指定屬於特定帳戶的所有電話號碼，請使用萬用字元 (*) 代替電話號碼 ID：

```
"Resource": "arn:aws:sms-voice:us-east-1:123456789012:phone-number/*"
```

有些 Amazon Pinpoint SMS 和語音動作，不能在特定資源上執行，例如用於管理帳戶層級設定 (例如支出限制) 的資源。在那些情況下，您必須使用萬用字元 (*)：

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

有些 Amazon Pinpoint API 動作會用到多項資源。例如，`TagResource` 動作可以將標籤加入多個專案。若要在單一陳述式中指定多項資源，請使用逗號分隔 ARN：

```
"Resource": [
      "resource1",
      "resource2"
```

若要查看 Amazon Pinpoint 資源類型及其 ARN 的清單，請參閱 IAM 使用者指南中的 Amazon Pinpoint 定義的資源。如需了解每種資源類型的 ARN 可用來指定哪些操作，請參閱 IAM 使用者指南中的 Amazon Pinpoint 定義的動作。

條件索引鍵

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

`Condition` 元素會根據定義的條件，指定陳述式的執行時機。您可以建立使用條件運算子的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。若要查看所有 AWS 全域條件索引鍵，請參閱《IAM 使用者指南》中的AWS 全域條件內容索引鍵。

Amazon Pinpoint 會定義自己的條件索引鍵，也支援一些全域條件索引鍵。若要查看所有 AWS 全域條件索引鍵的清單，請參閱《IAM 使用者指南》中的AWS 全域條件內容索引鍵。若要查看 Amazon Pinpoint 條件索引鍵的清單，請參閱 IAM 使用者指南中的 Amazon Pinpoint 條件索引鍵。若要了解條件索引鍵可與哪些動作和資源搭配使用，請參閱 IAM 使用者指南中的 Amazon Pinpoint 定義的動作。

範例

若要檢視 Amazon Pinpoint 身分型政策的範例，請參閱 Amazon Pinpoint 身分型政策範例。

Amazon Pinpoint 資源型許可的政策

資源型許可政策是 JSON 政策文件，內容說明指定的主體可在哪些條件下對 Amazon Pinpoint 資源執行哪些動作。Amazon Pinpoint 支援行銷活動、旅程、訊息範本 (範本)、推薦者模型 (推薦者)、專案 (應用程式) 和客群的資源型許可政策。

範例

若要檢視 Amazon Pinpoint 資源型政策的範例，請參閱 the section called “身分型政策範例”。

基於 Amazon Pinpoint 標籤的授權

您可以將標籤與特定類型的 Amazon Pinpoint 資源相關聯，或將請求中的標籤傳遞給 Amazon Pinpoint。如需根據標籤控制存取，請使用 `aws:ResourceTag/${TagKey}`、`aws:RequestTag/${TagKey}` 或 `aws:TagKeys` 條件索引鍵，在政策的條件元素中，提供標籤資訊。

如需標記 Amazon Pinpoint 資源相關資訊 (包括 IAM 政策範例)，請參閱 管理 Amazon Pinpoint 資源標籤。

Amazon Pinpoint IAM 角色

IAM 角色是您 AWS 帳戶中具備特定許可的實體。

使用臨時憑證搭配 Amazon Pinpoint

您可以搭配聯合使用暫時登入資料登入、擔任 IAM 角色，或是擔任跨帳戶角色。您取得暫時安全登入資料的方式是透過呼叫 AWS Security Token Service (AWS STS) API 操作 (例如，AssumeRole 或 GetFederationToken)。

Amazon Pinpoint 支援使用臨時憑證。

服務連結角色

服務連結角色可讓 AWS 服務存取其他服務中的資源，以代表您完成 動作。服務連結角色會顯示在您的 IAM 帳戶中，並由該服務所擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

Amazon Pinpoint 不使用服務連結角色。

服務角色

此功能可讓服務代表您擔任服務角色。此角色可讓服務存取其他服務中的資源，以代表您完成動作。服務角色會出現在您的 IAM 帳戶中，且由該帳戶所擁有。這表示 IAM 管理員可以變更此角色的許可。不過，這樣可能會破壞此服務的功能。

Amazon Pinpoint 支援使用服務角色。

IAM 政策的 Amazon Pinpoint 動作

若要管理您 AWS 帳戶中 Amazon Pinpoint 資源的存取權，您可以將 Amazon Pinpoint 動作新增至 AWS Identity and Access Management (IAM) 政策。使用政策中的動作，能夠控制使用者可以在 Amazon Pinpoint 主控台執行的動作。您也可以直接使用 AWS SDKs、 AWS Command Line Interface (AWS CLI) 或 Amazon Pinpoint APIs，以程式設計方式控制使用者可以執行的動作。

在政策中，指定每個動作時請使用適當的 Amazon Pinpoint 命名空間，後面加上冒號和動作名稱，例如 `GetSegments`。大多數的動作相當於使用特定 URI 和 HTTP 方法，對 Amazon Pinpoint API 提出的請求。例如，如果您允許使用者政策中的 `mobiletargeting:GetSegments` 動作，這會允許使用者提交 HTTP GET 請求至 `/apps/projectId/segments` URI，以擷取專案所有區段的相關資訊。此政策也允許使用者在主控台上檢視該資訊，並使用 AWS SDK 或 擷取該資訊 AWS CLI。

每個動作都是針對特定的 Amazon Pinpoint 資源執行，您可以在政策聲明中根據其 Amazon Resource Name (ARN) 加以識別。例如，對特定專案執行 `mobiletargeting:GetSegments` 動作，您會使用 ARN `arn:aws:mobiletargeting:region:accountId:apps/projectId` 來識別此專案。

本主題介紹了您可以在 AWS 帳戶的 IAM 政策中加入的 Amazon Pinpoint 動作。若要查看範例，了解如何使用政策中的動作，管理 Amazon Pinpoint 資源的存取權，請參閱 Amazon Pinpoint 身分型政策範例。

主題

- Amazon Pinpoint API 動作
- Amazon Pinpoint SMS 和語音第 1 版 API 動作

Amazon Pinpoint API 動作

本節介紹 Amazon Pinpoint API 可用功能的動作，Amazon Pinpoint API 是 Amazon Pinpoint 的主要 API。若要進一步了解此 API，請參閱 Amazon Pinpoint API 參考。

類別：

- 分析和指標
- 行銷活動
- 頻道
- 端點

- 事件串流
- 事件
- 匯出任務
- 匯入任務
- 旅程
- 訊息範本
- 訊息
- 一次性密碼
- 電話號碼驗證
- 專案
- 建議者模型
- 客群
- Tags (標籤)
- 使用者

分析和指標

以下許可是關於在 Amazon Pinpoint 主控台檢視分析資料。這些許可也和擷取 (查詢) 標準指標的彙整資料相關；這些資料稱為「關鍵績效指標 (KPI)」，適用於專案、行銷活動和旅程。

`mobiletargeting:GetReports`

在 Amazon Pinpoint 主控台上檢視分析資料。您也需要此許可才能使用 Amazon Pinpoint 主控台建立包含自訂屬性的客群。此外也需要取得 Amazon Pinpoint 主控台內客群規模的估算值。

- URI – 不適用
- 方法 – 不適用
- 資源 ARN – `arn:aws:mobiletargeting:`*`region`*`:`*`accountId`*`:*`

`mobiletargeting:GetApplicationDateRangeKpi`

擷取 (查詢) 標準應用程式指標的彙整資料。這是適用於與專案相關聯所有行銷活動或交易訊息的指標。

- URI – `/apps/`*`projectId`*`/kpis/daterange/`*`kpi-name`*
- 方法 – GET

- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/kpis/daterange/*kpi-name*

mobiletargeting:GetCampaignDateRangeKpi

擷取 (查詢) 標準行銷活動指標的彙整資料。這是適用於個別行銷活動的指標。

- URI – /apps/*projectId*/campaigns/*campaignId*/kpis/daterange/*kpi-name*
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*/kpis/daterange/*kpi-name*

mobiletargeting:GetJourneyDateRangeKpi

擷取 (查詢) 標準旅程參與指標的彙整資料。這個參與指標適用於個別旅程，例如在旅程的所有活動中，參與者已開啟的訊息數。

- URI – /apps/*projectId*/journeys/*journeyId*/kpis/daterange/*kpi-name*
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*/kpis/daterange/*kpi-name*

mobiletargeting:GetJourneyExecutionMetrics

擷取 (查詢) 適用於個別旅程的標準執行指標的彙總資料，例如積極執行旅程中所有活動的參與者人數。

- URI – /apps/*projectId*/journeys/*journeyId*/execution-metrics
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*/execution-metrics

mobiletargeting:GetJourneyExecutionActivityMetrics

擷取 (查詢) 適用於個別旅程的標準執行指標的彙總資料，例如開始或完成活動的參與者人數。

- URI – /apps/*projectId*/journeys/*journeyId*/activities/*journey-activity-id*/execution-metrics
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*/activities/*journey-activity-id*/execution-metrics

行銷活動

以下許可與管理您 Amazon Pinpoint 帳戶中的行銷活動有關。

mobiletargeting:CreateCampaign

建立專案的行銷活動。

- URI – /apps/*projectId*/campaigns
- 方法 – POST
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns

mobiletargeting:DeleteCampaign

刪除特定的行銷活動。

- URI – /apps/*projectId*/campaigns/*campaignId*
- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:GetCampaign

擷取特定行銷活動的相關資訊。

- URI – /apps/*projectId*/campaigns/*campaignId*
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:GetCampaignActivities

擷取行銷活動所執行的活動的相關資訊。

- URI – /apps/*projectId*/campaigns/*campaignId*/activities
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:GetCampaigns

擷取專案的所有行銷活動的相關資訊。

- URI – /apps/*projectId*/campaigns
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:GetCampaignVersion

擷取特定行銷活動版本的相關資訊。

- URI – /apps/*projectId*/campaigns/*campaignId*/versions/*versionId*
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:GetCampaignVersions

擷取行銷活動的目前和之前版本的相關資訊。

- URI – /apps/*projectId*/campaigns/*campaignId*/versions
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

mobiletargeting:UpdateCampaign

更新特定的行銷活動。

- URI – /apps/*projectId*/campaigns/*campaignId*
- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/campaigns/*campaignId*

頻道

以下許可與管理您 Amazon Pinpoint 帳戶中的管道有關。在 Amazon Pinpoint 中，管道是指您用於聯絡客戶的方法，例如傳送電子郵件、SMS 訊息或推播通知。

mobiletargeting:DeleteAdmChannel

停用專案的 Amazon Device Messaging (ADM) 管道。

- URI – /apps/*projectId*/channels/adm
- 方法 – DELETE

- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/adm

mobiletargeting:GetAdmChannel

擷取專案的 ADM 管道的相關資訊。

- URI – /apps/*projectId*/channels/adm
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/adm

mobiletargeting:UpdateAdmChannel

啟用或更新專案的 ADM 管道。

- URI – /apps/*projectId*/channels/adm
- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/adm

mobiletargeting:DeleteApnsChannel

停用專案的 Apple 推播通知服務 (APNs) 管道。

- URI – /apps/*projectId*/channels/apns
- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns

mobiletargeting:GetApnsChannel

擷取專案的 APNs 管道的相關資訊。

- URI – /apps/*projectId*/channels/apns
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns

mobiletargeting:UpdateApnsChannel

啟用或更新專案的 APN 管道。

- URI – /apps/*projectId*/channels/apns

- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns

mobiletargeting:DeleteApnsSandboxChannel

停用專案的 APNs 沙盒管道。

- URI – /apps/*projectId*/channels/apns_sandbox
- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_sandbox

mobiletargeting:GetApnsSandboxChannel

擷取專案的 APNs 沙盒管道的相關資訊。

- URI – /apps/*projectId*/channels/apns_sandbox
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_sandbox

mobiletargeting:UpdateApnsSandboxChannel

啟用或更新專案的 APN 沙盒管道。

- URI – /apps/*projectId*/channels/apns_sandbox
- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_sandbox

mobiletargeting:DeleteApnsVoipChannel

停用專案的 APNs VoIP 管道。

- URI – /apps/*projectId*/channels/apns_voip
- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_voip

mobiletargeting:GetApnsVoipChannel

擷取專案的 APNs VoIP 管道的相關資訊。

- URI – /apps/*projectId*/channels/apns_voip

- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_voip

mobiletargeting:UpdateApnsVoipChannel

啟用或更新專案的 APN VoIP 管道。

- URI – /apps/*projectId*/channels/apns_voip
- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_voip

mobiletargeting:DeleteApnsVoipSandboxChannel

停用專案的 APNs VoIP 沙盒管道。

- URI – /apps/*projectId*/channels/apns_voip_sandbox
- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_voip_sandbox

mobiletargeting:GetApnsVoipSandboxChannel

擷取專案的 APNs VoIP 沙盒管道的相關資訊。

- URI – /apps/*projectId*/channels/apns_voip_sandbox
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_voip_sandbox

mobiletargeting:UpdateApnsVoipSandboxChannel

啟用或更新專案的 APN VoIP 沙盒管道。

- URI – /apps/*projectId*/channels/apns_voip_sandbox
- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/apns_voip_sandbox

mobiletargeting:DeleteBaiduChannel

停用專案的百度雲推送管道。

- URI – /apps/*projectId*/channels/baidu

- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/baidu

mobiletargeting:GetBaiduChannel

擷取專案之百度雲推送管道的相關資訊。

- URI – /apps/*projectId*/channels/baidu
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/baidu

mobiletargeting:UpdateBaiduChannel

啟用或更新專案的百度雲推送管道。

- URI – /apps/*projectId*/channels/baidu
- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/baidu

mobiletargeting:DeleteEmailChannel

停用專案的電子郵件管道。

- URI – /apps/*projectId*/channels/email
- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/email

mobiletargeting:GetEmailChannel

擷取專案之電子郵件管道的相關資訊。

- URI – /apps/*projectId*/channels/email
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/email

mobiletargeting:UpdateEmailChannel

啟用或更新專案的電子郵件管道。

- URI – /apps/*projectId*/channels/email

- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/email

mobiletargeting:DeleteGcmChannel

為專案停用 Firebase Cloud Messaging (FCM) 管道。此管道允許 Amazon Pinpoint 透過 FCM 服務，傳送推播通知到 Android 應用程式；FCM 服務取代了 Google Cloud Messaging (GCM) 服務。

- URI – /apps/*projectId*/channels/gcm
- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/gcm

mobiletargeting:GetGcmChannel

擷取專案的 FCM 管道相關資訊。此管道允許 Amazon Pinpoint 透過 FCM 服務，傳送推播通知到 Android 應用程式；FCM 服務取代了 Google Cloud Messaging (GCM) 服務。

- URI – /apps/*projectId*/channels/gcm
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/gcm

mobiletargeting:UpdateGcmChannel

啟用或更新專案的 FCM 管道。此管道允許 Amazon Pinpoint 透過 FCM 服務，傳送推播通知到 Android 應用程式；FCM 服務取代了 Google Cloud Messaging (GCM) 服務。

- URI – /apps/*projectId*/channels/gcm
- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/gcm

mobiletargeting:DeleteSmsChannel

停用專案的簡訊管道。

- URI – /apps/*projectId*/channels/sms
- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/sms

mobiletargeting:GetSmsChannel

擷取專案之簡訊管道的相關資訊。

- URI – /apps/*projectId*/channels/sms
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/sms

mobiletargeting:UpdateSmsChannel

啟用或更新專案的簡訊管道。

- URI – /apps/*projectId*/channels/sms
- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/channels/sms

mobiletargeting:GetChannels

擷取應用程式每個管道的歷史記錄和狀態相關資訊。

- URI – /apps/*application-id*/channels
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:region:*accountId*:apps/*projectId*/channels

mobiletargeting:DeleteVoiceChannel

停用應用程式的語音管道，並刪除該管道的所有現有設定。

- URI – /apps/*application-id*/channels/voice
- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectid*/channels/voice

mobiletargeting:GetVoiceChannel

擷取應用程式語音管道的狀態和設定相關資訊。

- URI – /apps/*application-id*/channels/voice
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectid*/channels/voice

mobiletargeting:UpdateVoiceChannel

啟用應用程式的語音管道，或更新應用程式的語音管道狀態和設定。

- URI – /apps/*application-id*/channels/voice
- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectid*/channels/voice

端點

以下許可與管理您 Amazon Pinpoint 帳戶中的端點有關。在 Amazon Pinpoint 中，端點是訊息的單一目的地。例如，端點可以是客戶的電子郵件地址、電話號碼或行動裝置字符。

mobiletargeting:DeleteEndpoint

刪除端點。

- URI – /apps/*projectId*/endpoints/*endpointId*
- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/endpoints/*endpointId*

mobiletargeting:GetEndpoint

擷取特定端點的相關資訊。

- URI – /apps/*projectId*/endpoints/*endpointId*
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/endpoints/*endpointId*

mobiletargeting:RemoveAttributes

從與應用程式相關聯的所有端點，移除一個或多個具相同屬性類型的屬性。

- URI – apps/*application-id*/attributes/*attribute-type*
- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:region:*accountId*:apps/*projectId*/attributes/*attribute-type*

mobiletargeting:UpdateEndpoint

建立端點或更新端點的資訊。

- URI – /apps/*projectId*/endpoints/*endpointId*
- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/endpoints/*endpointId*

mobiletargeting:UpdateEndpointsBatch

以批次操作來建立或更新端點。

- URI – /apps/*projectId*/endpoints
- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

事件串流

以下許可與管理您 Amazon Pinpoint 帳戶中的事件串流有關。

mobiletargeting:DeleteEventStream

刪除專案的事件串流。

- URI – /apps/*projectId*/eventstream/
- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/eventstream

mobiletargeting:GetEventStream

擷取專案的事件串流的相關資訊。

- URI – /apps/*projectId*/eventstream/
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/eventstream

mobiletargeting:PutEventStream

建立或更新專案的事件串流。

- URI – /apps/*projectId*/eventstream/

- 方法 – POST
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/eventstream

事件

以下許可與管理您 Amazon Pinpoint 帳戶中的事件任務有關。您可以在 Amazon Pinpoint 中建立匯入任務，並根據存放在 Amazon S3 儲存貯體的端點定義來建立客群。

mobiletargeting:PutEvents

建立要記錄端點的新事件，或者建立或更新與現有事件相關聯的端點資料。

- URI – /apps/*application-id*/events
- 方法 – POST
- 資源 ARN – arn:aws:mobiletargeting:region:*accountId*:apps/*projectId*/events

匯出任務

以下許可與管理您 Amazon Pinpoint 帳戶中的匯出任務有關。您可以在 Amazon Pinpoint 中建立匯出任務，將端點相關資訊傳送到 Amazon S3 儲存貯體來儲存或分析。

mobiletargeting:CreateExportJob

建立匯出任務，將端點定義匯出到 Amazon S3。

- URI – /apps/*projectId*/jobs/export
- 方法 – POST
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/jobs/export

mobiletargeting:GetExportJob

擷取有關特定的匯出任務的專案。

- URI – /apps/*projectId*/jobs/export/*jobId*
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/jobs/export/*jobId*

mobiletargeting:GetExportJobs

擷取專案的所有匯出任務的清單。

- URI – /apps/*projectId*/jobs/export
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/jobs/export

匯入任務

以下許可與管理您 Amazon Pinpoint 帳戶中的匯入任務有關。您可以在 Amazon Pinpoint 中建立匯入任務，並根據存放在 Amazon S3 儲存貯體的端點定義來建立客群。

mobiletargeting:CreateImportJob

從 Amazon S3 匯入端點定義以建立客群。

- URI – /apps/*projectId*/jobs/import
- 方法 – POST
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:GetImportJob

擷取專案之特定匯入任務的相關資訊。

- URI – /apps/*projectId*/jobs/import/*jobId*
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/jobs/import/*jobId*

mobiletargeting:GetImportJobs

擷取專案的所有匯入任務的相關資訊。

- URI – /apps/*projectId*/jobs/import
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

旅程

以下許可與管理您 Amazon Pinpoint 帳戶中的旅程有關。

mobiletargeting:CreateJourney

為專案建立旅程。

- URI – /apps/*projectId*/journeys
- 方法 – POST
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys

mobiletargeting:GetJourney

擷取特定行程的相關資訊。

- URI – /apps/*projectId*/journeys/*journeyId*
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*

mobiletargeting:ListJourneys

擷取專案所有旅程的資訊。

- URI – /apps/*projectId*/journeys
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys

mobiletargeting:UpdateJourney

更新特定旅程的組態和其他設定。

- URI – /apps/*projectId*/journeys/*journeyId*
- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*

mobiletargeting:UpdateJourneyState

取消作用中的旅程。

- URI – /apps/*projectId*/journeys/*journeyId*/state
- 方法 – PUT

- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*/state

mobiletargeting:DeleteJourney

刪除特定的行程。

- URI – /apps/*projectId*/journeys/*journeyId*
- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/journeys/*journeyId*

訊息範本

以下許可與建立並管理您 Amazon Pinpoint 帳戶中的訊息範本有關。訊息範本是您可在訊息中定義、儲存和重複使用的內容和設定，方便您將訊息傳送到任何 Amazon Pinpoint 專案。

mobiletargeting:ListTemplates

擷取與您 Amazon Pinpoint 帳戶相關聯的所有訊息範本相關資訊。

- URI – /templates
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates

mobiletargeting:ListTemplateVersions

擷取特定訊息範本的所有版本相關資訊。

- URI – /templates/*template-name*/*template-type*/versions
- 方法 – GET
- 資源 ARN – 不適用

mobiletargeting:UpdateTemplateActiveVersion

將特定的訊息範本版本指定為作用中的範本版本。

- URI – /templates/*template-name*/*template-type*/active-version
- 方法 – GET
- 資源 ARN – 不適用

mobiletargeting:GetEmailTemplate

擷取透過電子郵件管道傳送的訊息範本相關資訊。

- URI – /templates/*template-name*/email
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting:CreateEmailTemplate

建立透過電子郵件管道傳送訊息的訊息範本。

- URI – /templates/*template-name*/email
- 方法 – POST
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting:UpdateEmailTemplate

更新透過電子郵件管道傳送訊息的現有訊息範本。

- URI – /templates/*template-name*/email
- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting:DeleteEmailTemplate

刪除透過電子郵件管道傳送訊息的訊息範本。

- URI – /templates/*template-name*/email
- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/EMAIL

mobiletargeting:GetPushTemplate

擷取透過推送通知管道傳送訊息的訊息範本相關資訊。

- URI – /templates/*template-name*/push
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/PUSH

mobiletargeting:CreatePushTemplate

建立透過推送通知管道傳送訊息的訊息範本。

- URI – /templates/*template-name*/push
- 方法 – POST
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/PUSH

mobiletargeting:UpdatePushTemplate

更新透過推送通知管道傳送訊息的現有訊息範本。

- URI – /templates/*template-name*/push
- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/PUSH

mobiletargeting:DeletePushTemplate

刪除透過推送通知管道傳送訊息的訊息範本。

- URI – /templates/*template-name*/push
- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/PUSH

mobiletargeting:GetSmsTemplate

擷取透過簡訊管道傳送的訊息範本相關資訊。

- URI – /templates/*template-name*/sms

- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/SMS

mobiletargeting:CreateSmsTemplate

建立透過簡訊管道傳送訊息的訊息範本。

- URI – /templates/*template-name*/sms
- 方法 – POST
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/SMS

mobiletargeting:UpdateSmsTemplate

更新透過簡訊管道傳送訊息的現有訊息範本。

- URI – /templates/*template-name*/sms
- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/SMS

mobiletargeting:DeleteSmsTemplate

刪除透過簡訊管道傳送訊息的訊息範本。

- URI – /templates/*template-name*/sms
- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/SMS

mobiletargeting:GetVoiceTemplate

擷取透過語音頻道傳送的訊息範本相關資訊。

- URI – /templates/*template-name*/voice
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/VOICE

mobiletargeting:CreateVoiceTemplate

建立透過語音頻道傳送訊息的訊息範本。

- URI – /templates/*template-name*/voice
- 方法 – POST
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/VOICE

mobiletargeting:UpdateVoiceTemplate

更新透過語音頻道傳送訊息的現有訊息範本。

- URI – /templates/*template-name*/voice
- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/VOICE

mobiletargeting:DeleteVoiceTemplate

刪除透過語音頻道傳送訊息的訊息範本。

- URI – /templates/*template-name*/voice
- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:templates/*template-name*/VOICE

訊息

以下許可有關於從您的 Amazon Pinpoint 帳戶傳送訊息和推播通知。您可以使用 SendMessages 和 SendUsersMessages 操作將訊息傳送到特定端點，而不需要先建立客群和行銷活動。

mobiletargeting:SendMessages

向特定的端點傳送訊息或推送通知。

- URI – /apps/*projectId*/messages
- 方法 – POST
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/messages

mobiletargeting:SendUsersMessages

向與特定使用者 ID 相關聯的所有端點傳送訊息或推送通知。

- URI – /apps/*projectId*/users-messages
- 方法 – POST
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/messages

一次性密碼

以下許可有關於在 Amazon Pinpoint 中傳送和驗證一次性密碼 (OTP) 有關。

mobiletargeting:SendOTPMessage

傳送包含一次性密碼的文字訊息。

- URI – /apps/*projectId*/otp
- 方法 – POST
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/otp

mobiletargeting:VerifyOTPMessage

針對使用 SendOTPMessage 操作產生的一次性密碼 (OTP) 查看有效性。

- URI – /apps/*projectId*/verify-otp
- 方法 – POST
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/verify-otp

電話號碼驗證

以下許可有關於在 Amazon Pinpoint 中使用電話號碼驗證服務。

mobiletargeting:PhoneNumberValidate

擷取電話號碼的相關資訊。

- URI – /phone/number/validate
- 方法 – POST

- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:phone/number/validate

專案

以下許可與管理您 Amazon Pinpoint 帳戶中的專案有關。起初，專案稱為應用程式。就操作目的而言，Amazon Pinpoint 應用程式與 Amazon Pinpoint 專案相同。

mobiletargeting:CreateApp

建立 Amazon Pinpoint 專案。

- URI – /apps
- 方法 – POST
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps

mobiletargeting:DeleteApp

刪除 Amazon Pinpoint 專案。

- URI – /apps/*projectId*
- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:GetApp

擷取 Amazon Pinpoint 專案相關資訊。

- URI – /apps/*projectId*
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:GetApps

擷取與您 Amazon Pinpoint 帳戶相關聯的所有專案相關資訊。

- URI – /apps
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps

mobiletargeting:GetApplicationSettings

擷取 Amazon Pinpoint 專案的預設設定。

- URI – /apps/*projectId*/settings

- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:UpdateApplicationSettings

更新 Amazon Pinpoint 專案的預設設定。

- URI – /apps/*projectId*/settings
- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

建議者模型

以下許可與管理 Amazon Pinpoint 組態相關，讓您從推薦者模型擷取和處理建議資料。建議者模型是一種機器學習模型，它可以透過在資料中尋找模式，來預測並產生個人化的建議。

mobiletargeting:CreateRecommenderConfiguration

為推薦模型建立 Amazon Pinpoint 組態。

- URI – /recommenders
- 方法 – POST
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:recommenders

mobiletargeting:GetRecommenderConfigurations

擷取與您 Amazon Pinpoint 帳戶相關聯的所有推薦者模型組態相關資訊。

- URI – /recommenders
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:recommenders

mobiletargeting:GetRecommenderConfiguration

擷取推薦者模型的個別 Amazon Pinpoint 組態相關資訊。

- URI – /recommenders/*recommenderId*
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:recommenders/*recommenderId*

mobiletargeting:UpdateRecommenderConfiguration

更新推薦模型的 Amazon Pinpoint 組態。

- URI – /recommenders/*recommenderId*
- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:recommenders/*recommenderId*

mobiletargeting:DeleteRecommenderConfiguration

刪除推薦模型的 Amazon Pinpoint 組態。

- URI – /recommenders/*recommenderId*
- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:recommenders/*recommenderId*

客群

以下許可與管理您 Amazon Pinpoint 帳戶中的客群有關。在 Amazon Pinpoint 中，客群是您行銷活動的收件人群組，共用您定義的某些屬性。

mobiletargeting:CreateSegment

建立客群。若要讓使用者從 Amazon Pinpoint 以外的地方匯入端點資料以建立客群，請允許 mobiletargeting:CreateImportJob 動作。

- URI – /apps/*projectId*/segments
- 方法 – POST
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:DeleteSegment

刪除客群。

- URI – /apps/*projectId*/segments/*segmentId*
- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:GetSegment

擷取特定客群的相關資訊。

- URI – /apps/*projectId*/segments/*segmentId*

- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:GetSegmentExportJobs

擷取匯出客群端點定義之任務的相關資訊。

- URI – /apps/*projectId*/segments/*segmentId*/jobs/export
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*/jobs/export

mobiletargeting:GetSegments

擷取專案所有區段的相關資訊。

- URI – /apps/*projectId*/segments
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*

mobiletargeting:GetSegmentImportJobs

針對從 Amazon S3 匯入端點定義而建立客群的任務，擷取相關資訊。

- URI – /apps/*projectId*/segments/*segmentId*/jobs/import
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:GetSegmentVersion

擷取特定客群版本的相關資訊。

- URI – /apps/*projectId*/segments/*segmentId*/versions/*versionId*
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:GetSegmentVersions

擷取客群的目前和之前版本的相關資訊。

- URI – /apps/*projectId*/segments/*segmentId*/versions
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

mobiletargeting:UpdateSegment

更新特定的客群。

- URI – /apps/*projectId*/segments/*segmentId*
- 方法 – PUT
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/segments/*segmentId*

Tags (標籤)

以下許可與檢視和管理 Amazon Pinpoint 資源的標籤有關。

mobiletargeting:ListTagsForResource

擷取與專案、行銷活動、訊息範本或區段相關聯標籤的相關資訊。

- URI – /tags/*resource-arn*
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:*

mobiletargeting:TagResource

將一或多個標籤新增到專案、行銷活動、訊息範本或區段。

- URI – /tags/*resource-arn*
- 方法 – POST
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:*

mobiletargeting:UntagResource

從專案、行銷活動、訊息範本或區段中移除一或多個標籤。

- URI – /tags/*resource-arn*

- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:*

使用者

以下許可與管理使用者有關。在 Amazon Pinpoint 中，使用者相當於接收您訊息的個人。單一使用者可能與多個端點相關聯。

mobiletargeting:DeleteUserEndpoints

刪除與使用者 ID 相關聯的所有端點。

- URI – /apps/*projectId*/users/*userId*
- 方法 – DELETE
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/users/*userId*

mobiletargeting:GetUserEndpoints

擷取與使用者 ID 相關聯之所有端點的資訊。

- URI – /apps/*projectId*/users/*userId*
- 方法 – GET
- 資源 ARN – arn:aws:mobiletargeting:*region*:*accountId*:apps/*projectId*/users/*userId*

Amazon Pinpoint SMS 和語音第 1 版 API 動作

本節介紹 Amazon Pinpoint SMS 和語音 API 中可用功能的動作。這是補充 API，提供進階選項，讓您使用和管理 Amazon Pinpoint 中的 SMS 和語音管道。若要進一步了解此 API，請參閱 Amazon Pinpoint SMS 和語音 API 參考。

sms-voice:CreateConfigurationSet

建立組態集來傳送語音訊息。

- URI – /sms-voice/configuration-sets
- 方法 – POST
- 資源 ARN – 不能使用。請使用 *。

sms-voice:DeleteConfigurationSet

刪除傳送語音訊息的組態集。

- URI – /sms-voice/configuration-sets/*ConfigurationSetName*
- 方法 – DELETE
- 資源 ARN – 不能使用。請使用 *。

sms-voice:GetConfigurationSetEventDestinations

擷取組態集及其所包含事件目標的相關資訊。

- URI – /sms-voice/configuration-sets/*ConfigurationSetName*/event-destinations
- 方法 – GET
- 資源 ARN – 不能使用。請使用 *。

sms-voice:CreateConfigurationSetEventDestination

建立語音事件的事件目的地。

- URI – /sms-voice/configuration-sets/*ConfigurationSetName*/event-destinations
- 方法 – POST
- 資源 ARN – 不能使用。請使用 *。

sms-voice:UpdateConfigurationSetEventDestination

更新語音事件的事件目的地。

- URI – /sms-voice/configuration-sets/*ConfigurationSetName*/event-destinations/*EventDestinationName*
- 方法 – PUT
- 資源 ARN – 不能使用。請使用 *。

sms-voice:DeleteConfigurationSetEventDestination

刪除語音事件的事件目的地。

- URI – /sms-voice/configuration-sets/*ConfigurationSetName*/event-destinations/*EventDestinationName*
- 方法 – DELETE

- 資源 ARN – 不能使用。請使用 *。

`sms-voice:SendVoiceMessage`

建立和傳送語音訊息。

- URI – /sms-voice/voice/message
- 方法 – POST
- 資源 ARN – 不能使用。請使用 *。

Amazon Pinpoint 身分型政策範例

使用者和角色預設不具備建立或修改 Amazon Pinpoint 資源的許可。他們也無法使用 AWS 管理主控台 AWS CLI或 AWS API 執行任務。IAM 管理員必須建立 IAM 政策，授予使用者和角色許可，讓他們對所需的資源執行特定的 API 操作。管理員接著必須將這些政策連接至需要這些許可的使用者或群組。

若要了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的在 JSON 標籤上建立政策。

主題

- 政策最佳實務
- 使用 Amazon Pinpoint 主控台
- 範例：存取單一 Amazon Pinpoint 專案
- 範例：根據標籤檢視 Amazon Pinpoint 資源
- 範例：允許使用者檢視他們自己的許可
- 範例：提供 Amazon Pinpoint API 動作的存取權
- 範例：提供 Amazon Pinpoint SMS 和語音 API 動作的存取權
- 範例：限制 Amazon Pinpoint 專案存取特定 IP 地址
- 範例：根據標籤限制 Amazon Pinpoint 存取權
- 範例：允許 Amazon Pinpoint 使用在 Amazon SES 中驗證的身分傳送電子郵件

政策最佳實務

身分型政策會判斷您帳戶中的某個人員是否可以建立、存取或刪除 Amazon Pinpoint 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管政策並邁向最低權限許可 – 若要開始將許可授予您的使用者和工作負載，請使用將許可授予許多常見使用案例的 AWS 受管政策。它們可在您的 中使用 AWS 帳戶。我們建議您定義特定於使用案例 AWS 的客戶受管政策，進一步減少許可。如需更多資訊，請參閱《IAM 使用者指南》中的 AWS 受管政策或任務職能的AWS 受管政策。
- 套用最低權限許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的許可。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱《IAM 使用者指南》中的 IAM 中的政策和許可。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。如果透過特定 例如 使用服務動作 AWS 服務，您也可以使用條件來授予其存取權 CloudFormation。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素：條件。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您撰寫安全且實用的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM Access Analyzer 驗證政策。
- 需要多重要素驗證 (MFA) – 如果您的案例需要 IAM 使用者或 中的根使用者 AWS 帳戶，請開啟 MFA 以提高安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的透過 MFA 的安全 API 存取。

如需 IAM 中最佳實務的相關資訊，請參閱《IAM 使用者指南》中的 IAM 安全最佳實務。

使用 Amazon Pinpoint 主控台

若要存取 Amazon Pinpoint 主控台，您必須擁有最基本的許可。這些許可必須允許您列出和檢視 AWS 帳戶中 Amazon Pinpoint 資源的詳細資訊。如果您建立的身分型政策，套用的許可比所需的最低權限更嚴格，那麼使用該政策的實體 (使用者或角色)，其主控台將不能正常運作。若要確保這些實體可以使用 Amazon Pinpoint 主控台，請將政策連接到實體。如需詳細資訊，請參閱《IAM 使用者指南》中的新增許可到使用者。

下列範例政策提供特定 AWS 區域中 Amazon Pinpoint 主控台的唯讀存取權。該唯讀存取權限包含了 Amazon Pinpoint 主控台仰賴的其他服務的唯讀存取權限，例如 Amazon Simple Email Service (Amazon SES)、IAM 和 Amazon Kinesis。

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "UseConsole",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": "arn:aws:mobiletargeting:us-east-1:111122223333:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "firehose:ListDeliveryStreams",
                "iam:ListRoles",
                "kinesis:ListStreams",
                "s3:List*",
                "ses:Describe*",
                "ses:Get*",
                "ses:List*",
                "sns:ListTopics"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                }
            }
        }
    ]
}
```

在上述政策範例中，將*##*取代為 AWS 區域名稱，並將 *accountId* 取代為 AWS 您的帳戶 ID。

對於僅呼叫 AWS CLI 或 AWS API 的使用者，您不需要允許最低主控台許可。反之，只需允許存取符合他們嘗試執行之 API 操作的動作就可以了。

範例：存取單一 Amazon Pinpoint 專案

您也可以建立唯讀政策，只存取特定專案。以下範例政策可讓使用者登入主控台，並檢視專案清單。它也讓使用者檢視 Amazon Pinpoint 主控台仰賴的其他 AWS 服務 (例如 Amazon SES、IAM 和 Amazon

Kinesis) 的相關資源的資訊。不過，此政策只讓使用者檢視政策中所指定之專案的相關資訊。您可以修改此政策，以允許存取其他專案或 AWS 區域。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ViewProject",
            "Effect": "Allow",
            "Action": "mobiletargeting:GetApps",
            "Resource": "arn:aws:mobiletargeting:us-east-1:111122223333:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": [
            "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/*",
    "arn:aws:mobiletargeting:us-east-1:111122223333:reports"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ses:Get*",
                "kinesis:ListStreams",
                "firehose:ListDeliveryStreams",
                "iam:ListRoles",
                "ses:List*",
                "sns:ListTopics",
                "ses:Describe*",
                "s3:List*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                }
```

```
            }
        }
    ]
}
```

在上述範例中，將*##*取代為 AWS 區域名稱、將 *accountId* 取代為 AWS 您的帳戶 ID，並將 *projectId* 取代為您要提供存取權的 Amazon Pinpoint 專案 ID。

同樣地，您可以建立政策，授予 AWS 您帳戶中對其中一個 Amazon Pinpoint 專案具有有限寫入存取權的使用者，例如具有專案 ID 的`810c7aab86d42fb2b56c8c966example`專案。在此情況下，您希望允許使用者檢視、新增和更新專案元件 (例如區段和促銷活動)，但不能刪除任何元件。

除了授與 `mobiletargeting:Get` 和 `mobiletargeting:List` 動作的許可之外，也請建立授與下列動作許可的政策：`mobiletargeting:Create`、`mobiletargeting:Update` 和 `mobiletargeting:Put`。這些是建立和管理大多數專案元件所需的額外許可。例如：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "LimitedWriteProject",
            "Effect": "Allow",
            "Action": "mobiletargeting:GetApps",
            "Resource": "arn:aws:mobiletargeting:us-east-1:111122223333:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*",
                "mobiletargeting:Create*",
                "mobiletargeting:Update*",
                "mobiletargeting:Put*"
            ],
            "Resource": [
            "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/810c7aab86d42fb2b56c8c966example",
    "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/810c7aab86d42fb2b56c8c966example/*",
```

```
                "arn:aws:mobiletargeting:us-east-1:111122223333:reports"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ses:Get*",
                "kinesis:ListStreams",
                "firehose:ListDeliveryStreams",
                "iam:ListRoles",
                "ses:List*",
                "sns:ListTopics",
                "ses:Describe*",
                "s3:List*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                }
            }
        }
    ]
}
```

範例：根據標籤檢視 Amazon Pinpoint 資源

您可以在身分型政策中使用條件，並根據標籤控制 Amazon Pinpoint 資源的存取權。此範例政策說明如何建立這類政策，才能允許檢視 Amazon Pinpoint 資源。但是，只有在資源標籤 `Owner` 的值是該使用者的使用者名稱時，才會授予該許可。此政策也會授予在主控台完成此動作的必要許可。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ListResources",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
```

```
                "mobiletargeting:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ViewResourceIfOwner",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": "arn:aws:mobiletargeting:*:*:*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Owner": "userName"
                },
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:mobiletargeting:us-
east-1:111122223333:*"
                }
            }
        }
    ]
}
```

您可以將此政策類型連接到您帳戶中的 使用者。如果 `richard-roe` 使用者嘗試檢視 Amazon Pinpoint 資源，必須將資源標記為 `Owner=richard-roe` 或 `owner=richard-roe`。否則他便會被拒絕存取。條件標籤鍵 `Owner` 符合 `Owner` 和 `owner`，因為條件索引鍵名稱不區分大小寫。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素：條件。

範例：允許使用者檢視他們自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視連接到他們使用者身分的內嵌及受管政策。此政策包含在主控台或使用 或 AWS CLI AWS API 以程式設計方式完成此動作的許可。

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

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

範例：提供 Amazon Pinpoint API 動作的存取權

本節提供的範例政策，可讓您存取 Amazon Pinpoint API 中各項功能的範例政策；Amazon Pinpoint API 是 Amazon Pinpoint 的主要 API。若要進一步了解此 API，請參閱 Amazon Pinpoint API 參考。

唯讀存取

下列範例政策允許唯讀存取特定 AWS 區域中 Amazon Pinpoint 帳戶中的所有資源。

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "ViewAllResources",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:Get*",
                "mobiletargeting:List*"
            ],
            "Resource": "arn:aws:mobiletargeting:us-east-1:111122223333:*"
        }
    ]
}
```

在上述範例中，將##取代為 AWS 區域名稱，並將 *accountId* 取代為您的帳戶 AWS ID。

管理員存取

以下範例政策可完全存取您 Amazon Pinpoint 帳戶中的所有 Amazon Pinpoint 動作和資源：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "FullAccess",
            "Effect": "Allow",
            "Action": [
                "mobiletargeting:*"
            ],
            "Resource": "arn:aws:mobiletargeting:us-east-1:111122223333:*"
        }
    ]
}
```

請在以上的範例中，將 *accountId* 替換成您的 AWS 帳戶 ID。

範例：提供 Amazon Pinpoint SMS 和語音 API 動作的存取權

本節提供的範例政策，可讓您存取 Amazon Pinpoint SMS 和語音 API 中各項功能。這是補充 API，提供進階選項，讓您使用和管理 Amazon Pinpoint 中的 SMS 和語音管道。若要進一步了解此 API，請參閱 Amazon Pinpoint SMS 和語音 API 參考。

唯讀存取

下列範例政策允許唯讀存取您 AWS 帳戶中的所有 Amazon Pinpoint SMS 和語音 API 動作和資源：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "SMSVoiceReadOnly",
            "Effect": "Allow",
            "Action": [
                "sms-voice:Get*",
                "sms-voice:List*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:sms-voice:us-east-1:111122223333:*"
                }
            }
        }
    ]
}
```

管理員存取

下列範例政策允許完整存取您 AWS 帳戶中的所有 Amazon Pinpoint SMS 和語音 API 動作和資源：

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "SMSVoiceFullAccess",
            "Effect": "Allow",
            "Action": [
                "sms-voice:*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:sms-voice:us-east-1:111122223333:*"
                }
            }
        }
    ]
}
```

範例：限制 Amazon Pinpoint 專案存取特定 IP 地址

以下範例政策允許任何使用者對指定項目 (*projectId*) 執行任何 Amazon Pinpoint 動作。不過，要求必須源自於條件中所指定的 IP 地址範圍。

此陳述式中的條件會識別允許之 Internet Protocol version 4 (IPv4) 地址的範圍 `54.240.143.*`，但有一個例外：`54.240.143.188`。`Condition` 區塊使用 `IpAddress`和 `NotIpAddress`條件和 `aws:SourceIp` 條件金鑰，這是 AWS全局條件金鑰。如需這些條件索引鍵的詳細資訊，請參閱政策 IAM 使用者指南中的指定條件。 `aws:SourceIp` IPv4 值會使用標準 CIDR 表示法。如需詳細資訊，請參閱《IAM 使用者指南》中的 IP 位址條件運算子。

JSON

```
{
    "Version":"2012-10-17",
    "Id":"AMZPinpointPolicyId1",
```

```
    "Statement":[
        {
            "Sid":"IPAllow",
            "Effect":"Allow",
            "Principal":"*",
            "Action":"mobiletargeting:*",
            "Resource":[
                "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId",
                "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/*"
            ],
            "Condition":{
                "IpAddress":{
                    "aws:SourceIp":"54.240.143.0/24"
                },
                "NotIpAddress":{
                    "aws:SourceIp":"54.240.143.188/32"
                }
            }
        }
    ]
}
```

範例：根據標籤限制 Amazon Pinpoint 存取權

以下範例政策授予對指定項目 (*projectId*) 執行任何 Amazon Pinpoint 動作的許可。不過，只有當請求來自其名稱是專案 `Owner` 資源標籤中之值 (如條件中所指定) 的使用者時，才會授與許可。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ModifyResourceIfOwner",
            "Effect": "Allow",
            "Action": "mobiletargeting:*",
            "Resource": [
            "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId",
    "arn:aws:mobiletargeting:us-east-1:111122223333:apps/projectId/*"
                ],
            "Condition": {
```

```
                "StringEquals": {
                    "aws:ResourceTag/Owner": "userName"
                }
            }
        }
    ]
}
```

範例：允許 Amazon Pinpoint 使用在 Amazon SES 中驗證的身分傳送電子郵件

透過 Amazon Pinpoint 主控台驗證電子郵件身份 (例如電子郵件地址或網域) 時，系統會設定讓 Amazon Pinpoint 和 Amazon SES 都可以自動使用該身分。但如果您透過 Amazon SES 驗證電子郵件身分，而且想將該身分與 Amazon Pinpoint 搭配使用，就必須對該身分套用政策。

以下範例政策授予 Amazon Pinpoint 使用透過 Amazon SES 驗證的電子郵件身分傳送電子郵件的權限。

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"PinpointEmail",
            "Effect":"Allow",
            "Principal":{
                "Service":"pinpoint.amazonaws.com"
            },
            "Action":"ses:*",
            "Resource":"arn:aws:ses:us-east-1:111122223333:identity/emailId",
            "Condition":{
                "StringEquals":{
                    "aws:SourceAccount":"111122223333"
                },
                "ArnLike":{
                    "aws:SourceArn":"arn:aws:mobiletargeting:us-
east-1:111122223333:apps/*"
                }
            }
        }
    ]
```

```
}
```

如果您在 AWS GovCloud （美國西部） 區域使用 Amazon Pinpoint，請改用下列政策範例：

JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"PinpointEmail",
            "Effect":"Allow",
            "Principal":{
                "Service":"pinpoint.amazonaws.com"
            },
            "Action":"ses:*",
            "Resource":"arn:aws-us-gov:ses:us-gov-
west-1:111122223333:identity/emailId",
            "Condition":{
                "StringEquals":{
                    "aws:SourceAccount":"111122223333"
                },
                "ArnLike":{
                    "aws:SourceArn":"arn:aws-us-gov:mobiletargeting:us-gov-
west-1:111122223333:apps/*"
                }
            }
        }
    ]
}
```

一般 Amazon Pinpoint 任務的 IAM 角色

IAM 角色是您可以在 AWS 帳戶中建立並授予特定許可的 AWS Identity and Access Management (IAM) 身分。IAM 角色是具有許可政策的 AWS 身分，可決定身分可以和不可以執行的動作 AWS。但角色不該只與一個人特別建立關聯，而是可以由任何有需要的人擔任。

此外角色沒有與之相關聯的標準長期憑證 (密碼或存取金鑰)。相反地，它提供工作階段的暫時性安全登入資料。您可以使用 IAM 角色將存取權委派給通常無法存取 AWS 資源的使用者、應用程式、應用程式或服務。

因此您可以使用 IAM 角色，將 Amazon Pinpoint 與您帳戶的特定 AWS 服務和資源整合。例如您可能想讓 Amazon Pinpoint 存取儲存在 Amazon Simple Storage Service (Amazon S3) 儲存貯體中的端點定義，並想將 Amazon Pinpoint 用於客群。或者您可能想讓 Amazon Pinpoint 將事件資料串流到帳戶的 Amazon Kinesis 串流。同樣地，您可能想要使用 IAM 角色來允許 Web 或行動應用程式註冊端點或報告 Amazon Pinpoint 專案的用量資料，而無需在應用程式中嵌入 AWS 金鑰 （在其中可能很難輪換，使用者可以擷取）。

對於這些案例，您可以使用 IAM 角色委派 Amazon Pinpoint 的存取權。本節說明並提供使用 IAM 角色搭配其他 AWS 服務的常見 Amazon Pinpoint 任務範例。如需將 IAM 角色搭配 Web 和行動應用程式使用的詳細資訊，請參閱 IAM 使用者指南中的提供存取權給通過外部驗證的使用者 (聯合身分)。

主題

- 用於匯入端點或客群的 IAM 角色
- 用於匯出端點或客群的 IAM 角色
- 從 Amazon Personalize 擷取建議的 IAM 角色
- 將事件串流到 Kinesis 的 IAM 角色
- 使用 Amazon SES 傳送電子郵件的 IAM 角色

用於匯入端點或客群的 IAM 角色

使用 Amazon Pinpoint，您可以從 AWS 帳戶中的 Amazon Simple Storage Service (Amazon S3) 儲存貯體匯入端點定義，以定義使用者客群。匯入之前，必須先將所需的許可委派給 Amazon Pinpoint。若要這樣做，您可以建立 AWS Identity and Access Management (IAM) 角色，並將下列政策連接至角色：

- `AmazonS3ReadOnlyAccess` AWS 受管政策。此政策由 建立和管理 AWS，並授予 Amazon S3 儲存貯體的唯讀存取權。
- 允許 Amazon Pinpoint 擔任角色的信任政策。

建立角色後，您可以使用 Amazon Pinpoint 從 Amazon S3 儲存貯體匯入客群。如需使用主控台建立儲存貯體、建立端點檔案以及匯入客群的相關資訊，請參閱 Amazon Pinpoint 使用者指南中的匯入

客群。如需如何使用 以程式設計方式匯入客群的範例 適用於 Java 的 AWS SDK，請參閱本指南在 Amazon Pinpoint 中匯入客群中的 。

建立 IAM 角色 (AWS CLI)

使用 AWS Command Line Interface () 完成下列步驟以建立 IAM 角色AWS CLI。如果您尚未安裝 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》中的安裝 AWS CLI 。

使用 建立 IAM 角色 AWS CLI

1. 建立 JSON 檔案，其中包含您的角色的信任政策，並將檔案儲存在本機。您可以使用下列信任政策。

   JSON

```
{
     "Version":"2012-10-17",
     "Statement": [
          {
               "Action": "sts:AssumeRole",
               "Effect": "Allow",
               "Principal": {
                    "Service": "pinpoint.amazonaws.com"
               },
               "Condition": {
                    "StringEquals": {
                         "aws:SourceAccount": "111122223333"
                    },
               "ArnLike": {
               "aws:SourceArn": "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/application-id"
                    }
               }
          }
     ]
}
```

   在上述範例中，執行下列動作：

   - 將##取代為您使用 Amazon Pinpoint AWS 的區域。
   - 將 *accountId* 取代為您 AWS 帳戶的唯一 ID。

- 將 *application-id* 取代為專案的唯一 ID。

2. 在命令行中，使用 create-role 命令來建立角色，並連接信任政策：

```
aws iam create-role --role-name PinpointSegmentImport --assume-role-policy-document
 file://PinpointImportTrustPolicy.json
```

在 file:// 字首後面，指定包含信任政策的 JSON 檔案的路徑。

執行這個命令之後，您會在終端機中看到類似下列的輸出：

3. 使用 attach-role-policy命令將 AmazonS3ReadOnlyAccess AWS 受管政策連接至角色：

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonS3ReadOnlyAccess --role-name PinpointSegmentImport
```

用於匯出端點或客群的 IAM 角色

您可以建立匯出任務以取得端點清單。建立匯出任務時，必須指定專案 ID，客群 ID 則可不指定。Amazon Pinpoint 會將與專案或客群相關聯的端點清單，匯出到 Amazon Simple Storage Service (Amazon S3) 儲存貯體。產生的檔案包含端點及其屬性的 JSON 格式清單，例如管道、地址、加入/退出狀態、建立日期和端點 ID。

若要建立匯出任務，您必須設定 IAM 角色允許 Amazon Pinpoint 寫入 Amazon S3 儲存貯體。設定角色的程序包含兩個步驟：

1. 建立一個允許實體 (就本案例而言是 Amazon Pinpoint) 寫入特定 Amazon S3 儲存貯體的 IAM 政策。
2. 建立 IAM 角色，並將政策連接到該角色。

此主題包含完成這兩個步驟的程序。這些程序假設您已建立 Amazon S3 儲存貯體，以及該儲存貯體內的資料夾，以用於存放匯出的客群。如需建立儲存貯體的詳細資訊，請參閱 Amazon Simple Storage Service 使用者指南中的建立儲存貯體。

這些程序也假設您已安裝並設定 AWS Command Line Interface (AWS CLI)。如需設定 的資訊 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》中的安裝 AWS CLI 。

步驟 1：建立 IAM 政策

IAM 政策定義了實體的許可，例如身分或資源。若要建立一個角色來匯出 Amazon Pinpoint 端點，必須先在特定 Amazon S3 儲存貯體中，建立一個授予特定資料夾寫入許可的政策。以下政策範例遵循了授予最低權限的安全性實務—也就是只授予執行單一任務所需的許可。

若要建立 IAM 政策

1. 在文字編輯器中，建立新檔案。將以下程式碼貼到檔案：

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowUserToSeeBucketListInTheConsole",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:GetBucketLocation"
            ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::*" ]
        },
        {
            "Sid": "AllowRootAndHomeListingOfBucket",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::amzn-s3-demo-bucket-example-
bucket" ],
            "Condition": {
                "StringEquals": {
                    "s3:delimiter": [ "/" ],
                    "s3:prefix": [
                        "",
                        "Exports/"
                    ]
                }
            }
        },
```

```
        {
            "Sid": "AllowListingOfUserFolder",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::amzn-s3-demo-bucket-example-
bucket" ],
            "Condition": {
                "StringLike": {
                    "s3:prefix": [
                        "Exports/*"
                    ]
                }
            }
        },
        {
            "Sid": "AllowAllS3ActionsInUserFolder",
            "Action": [ "s3:*" ],
            "Effect": "Allow",
            "Resource": [ "arn:aws:s3:::amzn-s3-demo-bucket-example-
bucket/Exports/*" ]
        }
    ]
}
```

在上述程式碼中，將 *amzn-s3-demo-bucket-example-bucket* 的所有執行個體取代為 Amazon S3 儲存貯體的名稱，其中包含您要匯出區段資訊的資料夾。此外，請將 *Exports* 的所有執行個體取代為資料夾本身的名稱。

完成後，請將檔案儲存為 s3policy.json。

2. 透過使用 AWS CLI，導覽至 s3policy.json 檔案所在的目錄。然後，輸入以下命令建立政策：

```
aws iam create-policy --policy-name s3ExportPolicy --policy-document
 file://s3policy.json
```

如果成功建立政策，您會看到類似以下的輸出：

```
{
    "Policy": {
        "CreateDate": "2018-04-11T18:44:34.805Z",
```

```
        "IsAttachable": true,
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PolicyId": "ANPAJ2YJQRJCG3EXAMPLE",
        "UpdateDate": "2018-04-11T18:44:34.805Z",
        "Arn": "arn:aws:iam::123456789012:policy/s3ExportPolicy",
        "PolicyName": "s3ExportPolicy",
        "Path": "/"
    }
}
```

複製政策的 Amazon Resource Name (ARN) (上述範例中的 `arn:aws:iam::123456789012:policy/s3ExportPolicy`)。在下一節，您必須在建立角色時提供此 ARN。

> **Note**
>
> 如果出現訊息，說明您的帳戶無權執行 `CreatePolicy` 操作，您需要將政策與您的使用者連接，這樣才能建立新的 IAM 政策和角色。如需詳細資訊，請參閱《IAM 使用者指南》中的新增和移除 IAM 身分許可。

步驟 2：建立 IAM 角色

現在您已建立 IAM 政策，接下來就能建立角色並連接該 IAM 政策。每個 IAM 角色包含了信任政策—這套規則指定了哪些實體可擔任該角色。本節中，您需要建立允許 Amazon Pinpoint 擔任角色的信任政策。接著，您需要建立角色本身，然後連接您在上一節建立的政策。

建立 IAM 角色

1. 在文字編輯器中，建立新檔案。將以下程式碼貼到檔案：

   JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Effect":"Allow",
            "Principal":{
```

```
                "Service":"pinpoint.amazonaws.com"
            },
            "Action":"sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                },
                "ArnLike": {
                "aws:SourceArn": "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/applicationId"
                }
            }
        }
    ]
}
```

儲存檔案為 `trustpolicy.json`。

2. 透過使用 AWS CLI，導覽至 `trustpolicy.json` 檔案所在的目錄。輸入以下命令來建立新的角色：

```
aws iam create-role --role-name s3ExportRole --assume-role-policy-document
 file://trustpolicy.json
```

3. 在命令列，輸入下列命令，將您在上一節建立的政策連接到您剛建立的角色：

```
aws iam attach-role-policy --policy-arn arn:aws:iam::123456789012:policy/
s3ExportPolicy --role-name s3ExportRole
```

在上述命令中，將 *arn:aws:iam::123456789012:policy/s3ExportPolicy* 替換成您在上一節建立的政策的 ARN。

從 Amazon Personalize 擷取建議的 IAM 角色

您可以設定 Amazon Pinpoint 從已部署為 Amazon Personalize 行銷活動的 Amazon Personalize 解決方案擷取建議資料。您可以使用此資料，根據每個收件人的屬性和行為，將個人化建議傳送給訊息收件人。如需詳細資訊，請參閱 Amazon Pinpoint 使用者指南中的機器學習模型。

從 Amazon Personalize 行銷活動擷取建議資料之前，必須先建立一個允許 Amazon Pinpoint 從行銷活動擷取資料的 AWS Identity and Access Management (IAM) 角色。您使用主控台在 Amazon Pinpoint 中設定推薦者模型時，Amazon Pinpoint 會自動為您建立這個角色。或者，您可以手動建立此角色。

若要手動建立角色，請使用 IAM API 完成以下步驟：

1. 建立一個允許實體 (就本案例而言是 Amazon Pinpoint)，從 Amazon Personalize 行銷活動擷取建議資料的 IAM 政策。
2. 建立 IAM 角色，並將 IAM 政策連接到該角色。

本主題說明如何使用 AWS Command Line Interface () 完成這些步驟AWS CLI。它假設您已建立 Amazon Personalize 解決方案，並部署為 Amazon Personalize 行銷活動。如需建立和部署行銷活動的相關資訊，請參閱 Amazon Personalize 開發人員指南中的建立行銷活動。

此主題也假設您已安裝並設定 AWS CLI。如需設定 的資訊 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》中的安裝 AWS CLI 。

步驟 1：建立 IAM 政策

IAM 政策定義了實體的許可，例如身分或資源。若要建立允許 Amazon Pinpoint 從 Amazon Personalize 行銷活動中擷取建議資料的角色，您必須先建立該角色的 IAM 政策。此政策需要允許 Amazon Pinpoint 執行以下操作：

- 擷取行銷活動所部署之解決方案的組態資訊 (`DescribeSolution`)。
- 檢查行銷活動的狀態 (`DescribeCampaign`)。
- 從行銷活動擷取建議資料 (`GetRecommendations`)。

在以下程序中，範例政策授予存取權給特定 Amazon Personalize 行銷活動部署的特定 Amazon Personalize 解決方案。

若要建立 IAM 政策

1. 在文字編輯器中，建立新檔案。將以下程式碼貼到檔案：

   JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "RetrieveRecommendationsOneCampaign",
            "Effect": "Allow",
            "Action": [
                "personalize:DescribeSolution",
                "personalize:DescribeCampaign",
                "personalize:GetRecommendations"
            ],
            "Resource": [
                "arn:aws:personalize:us-
east-1:111122223333:solution/solutionId",
                "arn:aws:personalize:us-
east-1:111122223333:campaign/campaignId"
            ]
        }
    ]
}
```

在上述範例中，以您的資訊取代*##*文字：

- *##* – 託管 Amazon Personalize 解決方案和行銷活動的 AWS 區域的名稱。
- *accountId* – 您的 AWS 帳戶 ID。
- *solutionId* – 行銷活動部署的 Amazon Personalize 解決方案的唯一資源 ID。
- *campaignId* – Amazon Personalize 行銷活動擷取建議資料的唯一資源 ID。

2. 完成後，請將檔案儲存為 RetrieveRecommendationsPolicy.json。
3. 透過使用命令列界面，導覽至 RetrieveRecommendationsPolicy.json 檔案儲存所在的目錄。
4. 輸入下列命令，以建立政策並將其命名為 RetrieveRecommendationsPolicy。若要使用不同的名稱，請將 *RetrieveRecommendationsPolicy* 變更為您要的名稱。

```
aws iam create-policy --policy-name RetrieveRecommendationsPolicy --policy-document
 file://RetrieveRecommendationsPolicy.json
```

> **Note**
>
> 如果出現訊息，說明您的帳戶無權執行 `CreatePolicy` 操作，您需要將政策與您的使用者連接，這樣才能為您的帳戶建立新的 IAM 政策和角色。如需詳細資訊，請參閱《 IAM 使用者指南》中的新增和移除 IAM 身分許可。

5. 複製政策的 Amazon Resource Name (ARN) (上述範例中的 `arn:aws:iam::123456789012:policy/RetrieveRecommendationsPolicy`)。您需要此 ARN 才能在下一節建立 IAM 角色。

步驟 2：建立 IAM 角色

建立 IAM 政策之後，您可以建立 IAM 角色並連接該 IAM 政策。

每個 IAM 角色都有一個信任政策，這套規則指定了哪些實體可擔任該角色。本節中，您需要建立允許 Amazon Pinpoint 擔任角色的信任政策。接下來，您可以建立角色本身。接著，您將政策連接至該角色。

建立 IAM 角色

1. 在文字編輯器中，建立新檔案。將以下程式碼貼到檔案：

   JSON

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "pinpoint.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                "AWS:SourceAccount": "111122223333"
                },
                "ArnLike": {
                "AWS:SourceArn": "arn:aws:mobiletargeting:us-
east-1:444455556666:apps/*"
```

```
                }
            }
        }
    ]
}
```

2. 儲存檔案為 RecommendationsTrustPolicy.json。
3. 透過使用命令列界面，導覽至 RecommendationsTrustPolicy.json 檔案儲存所在的目錄。
4. 輸入下列命令，以建立新角色並將其命名為 PinpointRoleforPersonalize。若要使用不同的名稱，請將 *PinpointRoleforPersonalize* 變更為您要的名稱。

```
aws iam create-role --role-name PinpointRoleforPersonalize --assume-role-policy-
document file://RecommendationsTrustPolicy.json
```

5. 輸入下列命令，將您在上一節建立的政策連接到您剛建立的角色：

```
aws iam attach-role-policy --policy-arn arn:aws:iam::123456789012:policy/
RetrieveRecommendationsPolicy --role-name PinpointRoleforPersonalize
```

在上述命令中，將 *arn:aws:iam::123456789012:policy/RetrieveRecommendationsPolicy* 替換成您在上一節建立的政策的 ARN。此外，如果您為角色指定不同的名稱，請以您在步驟 4 中指定的角色名稱取代 *PinpointRoleforPersonalize*。

將事件串流到 Kinesis 的 IAM 角色

Amazon Pinpoint 可以自動將應用程式用量資料或事件資料從您的應用程式傳送至 AWS 帳戶中的 Amazon Kinesis 資料串流或 Amazon Data Firehose 交付串流。您必須先將所需許可委派給 Amazon Pinpoint，Amazon Pinpoint 才能開始串流事件資料。

如果您使用主控台設定事件串流，Amazon Pinpoint 會自動建立具有必要許可的 AWS Identity and Access Management (IAM) 角色。如需詳細資訊，請參閱《Amazon Pinpoint 使用者指南》中的將 Amazon Pinpoint 事件串流至 Kinesis。 Amazon Pinpoint

如果您要手動建立角色，請將以下政策連接到角色：

- 允許 Amazon Pinpoint 將事件資料傳送到您串流的許可政策。
- 允許 Amazon Pinpoint 擔任角色的信任政策。

建立角色後，可以設定 Amazon Pinpoint 自動將事件傳送到您的串流。如需詳細資訊，請參閱本指南中的 使用 Amazon Pinpoint 透過 Kinesis 和 Firehose 串流應用程式事件資料。

建立 IAM 角色 (AWS CLI)

完成以下步驟來使用 AWS Command Line Interface (AWS CLI) 手動建立 IAM 角色。若要了解如何使用 Amazon Pinpoint 主控台建立角色，請參閱 Amazon Pinpoint 使用者指南中的將 Amazon Pinpoint 事件串流到 Amazon Kinesis。

如果您尚未安裝 AWS CLI，請參閱AWS Command Line Interface 《 使用者指南》中的安裝 AWS CLI 。您也需要建立 Kinesis 串流或 Firehose 串流。如需有關建立這些資源的資訊，請參閱《Amazon Kinesis Data Streams 開發人員指南》中的建立和管理串流，或《Amazon Data Firehose 開發人員指南》中的建立 Amazon Data Firehose 交付串流。

使用 建立 IAM 角色 AWS CLI

1. 建立新檔案。將下列政策貼到文件中，並進行下列變更：

   - 將*##*取代為您使用 Amazon Pinpoint AWS 的區域。
   - 將 *accountId* 取代為您 AWS 帳戶的唯一 ID。
   - 將 *applicationId* 取代為專案的唯一 ID。

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "pinpoint.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "111122223333"
                },
                "ArnLike": {
                "aws:SourceArn": "arn:aws:mobiletargeting:us-
east-1:111122223333:apps/applicationId"
```

```
                }
            }
        }
    ]
}
```

完成後，請將檔案儲存為 PinpointEventStreamTrustPolicy.json。

2. 使用 create-role 命令來建立角色，並連接信任政策：

```
aws iam create-role --role-name PinpointEventStreamRole --assume-role-policy-
document file://PinpointEventStreamTrustPolicy.json
```

3. 建立一個新檔案，內含您角色的許可政策。

如果您要設定 Amazon Pinpoint 將資料傳送至 Kinesis 串流，請將下列政策貼入 檔案，並取代下列項目：

- 將##取代為您使用 Amazon Pinpoint AWS 的區域。
- 將 *accountId* 取代為您 AWS 帳戶的唯一 ID。
- 以 Kinesis 串流的名稱取代 *streamName*。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Action": [
            "kinesis:PutRecords",
            "kinesis:DescribeStream"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:kinesis:us-east-1:111122223333:stream/streamName"
        ]
    }
}
```

或者，如果您設定 Amazon Pinpoint 將資料傳送至 Firehose 串流，請將下列政策貼入 檔案，並取代下列項目：

- 將*##*取代為您使用 Amazon Pinpoint AWS 的區域。
- 將 *accountId* 取代為您 AWS 帳戶的唯一 ID。
- 將 *delivery-stream-name* 取代為您 Firehose 串流的名稱。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "firehose:PutRecordBatch",
            "firehose:DescribeDeliveryStream"
        ],
        "Resource": [
            "arn:aws:firehose:us-east-1:111122223333:deliverystream/delivery-
stream-name"
        ]
    }
}
```

完成後，請將檔案儲存為 PinpointEventStreamPermissionsPolicy.json。

4. 使用 put-role-policy 命令將許可政策連接到角色：

```
aws iam put-role-policy --role-name PinpointEventStreamRole --policy-
name PinpointEventStreamPermissionsPolicy --policy-document file://
PinpointEventStreamPermissionsPolicy.json
```

使用 Amazon SES 傳送電子郵件的 IAM 角色

Amazon Pinpoint 使用您的 Amazon SES 資源為您的行銷活動或旅程傳送電子郵件。在 Amazon Pinpoint 可以使用您的 Amazon SES 資源傳送電子郵件之前，您必須將必要的許可授予 Amazon

Pinpoint。您的帳戶必須具有 iam:PutRolePolicy和 iam:UpdateAssumeRolePolicy許可，才能更新或建立 IAM 角色。

Amazon Pinpoint 主控台可以自動建立具有所需許可的 AWS Identity and Access Management (IAM) 角色。如需詳細資訊，請參閱《Amazon Pinpoint 使用者指南》中的建立電子郵件協同運作傳送角色。

如果您要手動建立角色，請將以下政策連接到角色：

- 授予 Amazon Pinpoint 存取 Amazon SES 資源的許可政策。
- 允許 Amazon Pinpoint 擔任角色的信任政策。

建立角色之後，您可以將 Amazon Pinpoint 設定為使用您的 Amazon SES 資源。

您可以使用 IAM 政策模擬器來測試 IAM 政策。如需詳細資訊，請參閱《 IAM 使用者指南》https://docs.aws.amazon.com/IAM/latest/UserGuide/中的使用 IAM 政策模擬器測試 IAM 政策。

建立 IAM 角色 (AWS 管理主控台)

完成下列步驟，為行銷活動或傳送電子郵件的旅程手動建立 IAM 角色。

1. 依照《IAM 使用者指南》中的使用 JSON 編輯器建立政策中的指示建立新的許可政策。
    - 在步驟 5 中，針對 IAM 角色使用下列許可政策。
        - 將###取代為資源所在的分割區。對於標準 AWS 區域，分割區為 aws。如果您有其他分割區的資源，則該分割區為 aws-partitionname。例如， AWS GovCloud (US-West) 中資源的分割區為 aws-us-gov。
        - 將##取代為 AWS 區域 託管 Amazon Pinpoint 專案的 名稱。
        - 將 *accountId* 取代為 的唯一 ID AWS 帳戶。

        JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "PinpointUsesSESForEmailSends",
            "Effect": "Allow",
            "Action": [
```

```
                "ses:SendEmail",
                "ses:SendRawEmail"
            ],
            "Resource": [
            "arn:aws:ses:us-east-1:111122223333:identity/*",
    "arn:aws:ses:us-east-1:111122223333:configuration-set/*"
            ]
        }
    ]
}
```

2. 依照《IAM 使用者指南》中的使用自訂信任政策建立角色中的指示建立新的信任政策。 https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_create_for-custom.html

   a. 在步驟 4 中，使用下列信任政策。

      - 將 *accountId* 取代為 的唯一 ID AWS 帳戶。

      JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPinpoint",
            "Effect": "Allow",
            "Principal": {
                "Service": "pinpoint.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "accountId"
                }
            }
        }
    ]
}
```

   b. 在步驟 11 中，新增您在上一個步驟中建立的許可政策。

疑難排解 Amazon Pinpoint 身分和存取管理

請參考以下資訊診斷及修正使用 Amazon Pinpoint 和 IAM 時可能遇到的常見問題。

主題

- 我無權在 Amazon Pinpoint 中執行操作
- 我未獲授權執行 iam:PassRole
- 我想要允許 AWS 帳戶外的人員存取我的 Amazon Pinpoint 資源

我無權在 Amazon Pinpoint 中執行操作

如果 AWS 管理主控台 告訴您無權執行 動作，則必須聯絡您的管理員尋求協助。您的管理員就是向您提供登入憑證的人。

`mateojackson` 使用者嘗試使用主控台檢視專案的詳細資訊，但卻沒有 `mobiletargeting:GetApp` 許可時，會出現以下範例錯誤。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 mobiletargeting:GetApp on resource: my-example-project
```

在此情況下，Mateo 會請求管理員更新他的政策，允許他使用 *my-example-project* 動作存取 `mobiletargeting:GetApp` 資源。

我未獲授權執行 iam:PassRole

如果您收到錯誤，告知您無權執行 `iam:PassRole` 動作，您必須更新政策，允許您將角色傳遞給 Amazon Pinpoint。

有些 AWS 服務 可讓您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

當 `marymajor` IAM 使用者嘗試使用主控台在 Amazon Pinpoint 中執行動作時，發生以下範例錯誤。但是，動作請求服務具備服務角色授予的許可。Mary 沒有將角色傳遞給服務的許可。

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

在這種情況下，Mary 的政策必須更新，允許她執行 `iam:PassRole` 動作。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我想要允許 AWS 帳戶外的人員存取我的 Amazon Pinpoint 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要了解 Amazon Pinpoint 是否支援這些功能，請參閱 Amazon Pinpoint 如何搭配 IAM 運作。
- 若要了解如何 AWS 帳戶 在您擁有的 資源之間提供存取權，請參閱《IAM 使用者指南》中的在您擁有 AWS 帳戶 的另一個 中提供存取權給 IAM 使用者。
- 若要了解如何將資源的存取權提供給第三方 AWS 帳戶，請參閱《IAM 使用者指南》中的將存取權提供給第三方 AWS 帳戶 擁有。
- 如需了解如何透過聯合身分提供存取權，請參閱《IAM 使用者指南》中的將存取權提供給在外部進行身分驗證的使用者 (聯合身分)。
- 如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的 IAM 中的跨帳戶資源存取。

Amazon Pinpoint 中的記錄和監控

為了維護 Amazon Pinpoint 專案和 Amazon Pinpoint 資源其他類型的可靠性、可用性和效能，記錄和監控十分重要。您應該從 Amazon Pinpoint 專案和資源的所有部分記錄和收集監控資料，以便在發生多點故障時更輕鬆地偵錯。 AWS 提供多種工具，可協助您記錄和收集這些資料，並回應潛在事件：

AWS CloudTrail

Amazon Pinpoint 與 整合 AWS CloudTrail，此服務提供使用者、角色或其他 AWS 服務在 Amazon Pinpoint 中所採取動作的記錄。這包括 Amazon Pinpoint 主控台的動作，以及對 Amazon Pinpoint API 執作的程式呼叫。使用透過 CloudTrail 收集的資訊，可以確定曾向 Amazon Pinpoint 提出哪些請求。對於每個請求，您可以識別提出時間、提出請求的 IP 地址、提出請求者，以及其他詳細資料。如需詳細資訊，請參閱本指南中的 使用 記錄 Amazon Pinpoint API 呼叫 AWS CloudTrail。

Amazon CloudWatch

您可以使用 Amazon CloudWatch 收集、檢視和分析與您 Amazon Pinpoint 帳戶和專案有關的幾個重要指標。您也可以使用 CloudWatch 建立警示，在指標值達到特定條件，且低於或超過您定義的閾值時通知您。如果您建立提示，CloudWatch 會將通知傳送到您指定的 Amazon Simple

Notification Service (Amazon SNS) 主題。詳情請參閱 Amazon Pinpoint 使用者指南中的使用 Amazon CloudWatch 監控 Amazon Pinpoint。

AWS Health 儀表板

透過使用 AWS Health 儀表板，您可以檢查和監控 Amazon Pinpoint 環境的狀態。若要檢查 Amazon Pinpoint 服務的整體狀態，請使用 AWS 服務運作狀態儀表板。若要檢查、監控和檢視可能更具體地影響您 AWS 環境的任何事件或問題的歷史資料，請使用 AWS 個人運作狀態儀表板。若要進一步了解這些儀表板，請參閱AWS Health 使用者指南。

AWS Trusted Advisor

AWS Trusted Advisor 會檢查您的 AWS 環境，並針對解決安全漏洞、改善系統可用性和效能，以及節省成本的機會提供建議。所有 AWS 客戶都可以存取一組核心 Trusted Advisor 檢查。擁有商業或企業支援計劃的客戶可以存取其他 Trusted Advisor 檢查。

其中許多檢查可協助您評估 Amazon Pinpoint 資源的安全狀態，做為整體 AWS 帳戶的一部分。例如，核心 Trusted Advisor 檢查組包括下列項目：

- 針對每個支援的 AWS 區域，記錄您 AWS 帳戶的組態。
- Amazon Simple Storage Service (Amazon S3) 儲存貯體的存取權限，可能包含您為了建置客群而匯入 Amazon Pinpoint 的檔案。
- 使用 AWS Identity and Access Management 使用者、群組和角色來控制對 Amazon Pinpoint 資源的存取。
- 可能危及 AWS 環境和 Amazon Pinpoint 資源安全性的 IAM 組態和政策設定。

如需詳細資訊，請參閱《支援 使用者指南》中的 AWS Trusted Advisor。

Amazon Pinpoint 的合規驗證

在多個 AWS 合規計畫中，第三方稽核人員會評估 Amazon Pinpoint 的安全性與合規。其中包括用於安全管理控制的 AWS 系統和組織控制 (SOC)、FedRAMP、HIPAA、ISO/IEC 27001：2013、用於雲端特定控制的 ISO/IEC 27017：2015、用於個人資料保護的 ISO/IEC 27018：2014、用於品質管理系統的 ISO/IEC 9001：2015 等。

如需在特定合規計劃範圍內 AWS 的服務清單，請參閱AWS 合規計劃範圍內的服務。如需一般資訊，請參閱AWS 合規計劃。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱在 AWS Artifact 中下載報告在 。

您使用 Amazon Pinpoint 時的合規責任取決於資料的敏感度、您公司的合規目標，以及適用的法律和法規。 AWS 提供下列資源來協助合規：

- 安全與合規快速入門指南 – 這些部署指南討論架構考量，並提供部署以安全與合規為中心之基準環境的步驟 AWS。
- HIPAA 安全與合規架構白皮書 – 此白皮書說明公司如何使用 AWS 來建立符合 HIPAA 規範的應用程式。
- AWS 合規資源 – 此工作手冊和指南集合可能適用於您的產業和位置。
- AWS Config 開發人員指南中的使用規則評估資源 – AWS Config 服務會評估資源組態符合內部實務、產業準則和法規的程度。
- AWS Security Hub CSPM – AWS 此服務提供 內安全狀態的全方位檢視 AWS ，可協助您檢查是否符合安全產業標準和最佳實務。

當客戶使用適當的通訊管道時，Amazon Pinpoint 是符合 AWS HIPAA 資格的服務。如果您要使用 Amazon Pinpoint 執行包含 HIPAA 及相關法規所定義的受保護健康資訊 (PHI) 的工作負載，您應該使用電子郵件管道、推播通知通道或 SMS 管道，傳送含有 PHI 的訊息。如果您使用簡訊管道傳送包含 PHI 的訊息，您應該從您為 AWS 帳戶請求的專用短碼傳送這些訊息，以明確傳送將包含或可能包含 PHI 的訊息。語音頻道不符合 AWS HIPAA 資格；請勿使用語音頻道來傳送包含 PHI 的訊息。

Amazon Pinpoint 的恢復能力

AWS 全球基礎設施是以 AWS 區域和可用區域為基礎建置。 AWS 區域提供多個實體隔離和隔離的可用區域，這些可用區域與低延遲、高輸送量和高備援聯網連接。透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴展能力，均較單一或多個資料中心的傳統基礎設施還高。

如需詳細資訊，請參閱有關參考架構的資訊，請參閱 Amazon Pinpoint 彈性參考架構。

如需 AWS 區域和可用區域的詳細資訊，請參閱 AWS 全球基礎設施。

Amazon Pinpoint 的基礎設施安全性

Amazon Pinpoint 是受管服務，受到 AWS 全球網路安全的保護。如需 AWS 安全服務以及 如何 AWS 保護基礎設施的相關資訊，請參閱AWS 雲端安全。若要使用基礎設施安全最佳實務來設計您的 AWS 環境，請參閱安全支柱 AWS Well-Architected Framework 中的基礎設施保護。

您可以使用 AWS 發佈的 API 呼叫，透過網路存取 Amazon Pinpoint。使用者端必須支援下列專案：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密(PFS)的密碼套件，例如 DHE (Ephemeral Diffie-Hellman)或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

您可以從任何網路位置進行 API 呼叫，但 Amazon Pinpoint 支援資源型存取政策。這些政策可以包含以來源 IP 地址為基礎的限制。若要深入了解此類政策，請參閱使用政策管理存取權。

此外，您可以設定和使用各種 AWS 安全功能，從與 Amazon Pinpoint 整合的任何行動或 Web 應用程式控制對 Amazon Pinpoint 資源的存取。這包括對任務的 API 呼叫限制，例如新增端點、更新端點資料、提交事件資料和報告使用資料。

若要使用這些功能，我們建議您使用 AWS Mobile SDKs或 AWS Amplify JavaScript 程式庫，將行動和 Web 應用程式與 Amazon Pinpoint 整合。對於 Android 或 iOS 應用程式，我們建議您 AWS Mobile SDK for iOS分別使用 適用於 Android 的 AWS Mobile SDK 或 。對於以 JavaScript 為基礎的行動或 Web 應用程式，我們建議您使用 Web 版 AWS Amplify JavaScript Library 或 React Native 版 AWS Amplify JavaScript Library。若要進一步了解這些資源，請參閱AWS 行動SDKs入門、適用於 Web 的 AWS Amplify 程式庫入門，以及適用於原生反應的 AWS Amplify 程式庫入門。

Amazon Pinpoint 中的組態與漏洞分析

作為受管服務，Amazon Pinpoint 受到 Amazon Web Services：安全程序概觀白皮書中所述的 AWS 全球網路安全程序的保護。這表示 會 AWS 管理和執行基本安全任務和程序，以強化、修補、更新和以其他方式維護 Amazon Pinpoint 帳戶和資源的基礎基礎設施。這些程序已由適當的第三方進行檢閱並認證。

如需詳細資訊，請參閱下列資源：

- Amazon Pinpoint 的合規驗證
- 共同責任模型
- Amazon Web Services：安全程序概觀 (白皮書)

Amazon Pinpoint 的安全最佳做法

使用 AWS Identity and Access Management (IAM) 帳戶來控制對 API 操作的 Amazon Pinpoint 存取，尤其是建立、修改或刪除 Amazon Pinpoint 資源的操作。對於 Amazon Pinpoint API，這類資源包括專案、行銷活動和旅程。 Amazon Pinpoint SMS 和語音 API 這類資源包括電話號碼、集區和組態集。

- 為每個管理 Amazon Pinpoint 資源的人員建立個別使用者，包括您自己。請勿使用 AWS 根登入資料來管理 Amazon Pinpoint 資源。
- 授予每個使用者執行其職責所需最低程度的許可。
- 使用 IAM 群組來有效管理多個使用者的許可。
- 定期輪替您的 IAM 登入資料。

如需 Amazon Pinpoint 安全性的詳細資訊，請參閱 Amazon Pinpoint 中的安全性。如需 IAM 的相關資訊，請參閱 AWS Identity and Access Management。如需 IAM 最佳實務的資訊，請參閱 IAM 最佳實務。

Amazon Pinpoint 配額

下節列出並說明適用於 Amazon Pinpoint 資源和操作的配額 (以前稱為限制)。有些配額可以增加，有些則無法增加。若要判斷是否可以請求增加配額，請參閱每個區段的 是否可增加 欄位或陳述式。

主題

- 專案配額
- API 請求配額
- 行銷活動配額
- 電子郵件配額
- 端點配額
- 端點匯入配額
- 事件擷取配額
- 行程配額
- Lambda 配額
- Machine Learning 配額
- 訊息範本配額
- 推播通知配額
- 應用程式內訊息配額
- 區段配額
- 簡訊配額
- 10DLC 配額
- 語音配額
- 請求提高配額

專案配額

下表列出的配額，與 Amazon Pinpoint 中的專案相關。

資源	預設配額	是否可增加？
專案	在每個專案中 AWS 區域，您最多可以有 100 個專案。	否

API 請求配額

Amazon Pinpoint 實作配額，限制您可以從 AWS 您的帳戶對 Amazon Pinpoint API 提出的請求大小和數量。

除非特定資源類型另有指定，否則呼叫 (請求和回應) 承載的大小上限為 7 MB。若要判斷資源是否可有不同的配額，請參閱該資源類型有關本主題的適當章節。

請求數目上限，視配額類型和 API 操作而異。Amazon Pinpoint 對 API 請求實行兩種配額：

- 速率配額 – 也稱為速率限制，這類配額定義了特定操作每秒可以提出的請求數目上限，也控制了每個帳戶傳送或接收請求的速率。
- 高載配額 – 也稱為高載限制或高載容量，此類配額定義了帳戶可以同時進行的請求數目上限。

下表列出 Amazon Pinpoint API 的速率配額和爆量配額。

作業	預設高載/速率配額 (每秒請求數)
CreateCampaign	25
CreateEmailTemplate	10
CreateInAppTemplate	10
CreateImportJob	300
CreatePushTemplate	10
CreateSegment	25
CreateSmsTemplate	10
CreateVoiceTemplate	10

作業	預設高載/速率配額 (每秒請求數)
DeleteCampaign	25
DeleteEndpoint	5
DeleteSegment	25
GetEndpoint	10
PhoneNumberValidate	20
PutEvents	15
SendMessages	4,000
SendUsersMessages	6,000
UpdateCampaign	25
UpdateEmailTemplate	10
UpdateEndpoint	10
UpdateEndpointsBatch	2
UpdateInAppTemplate	10
UpdatePushTemplate	10
UpdateSegment	25
UpdateSmsTemplate	10
UpdateVoiceTemplate	10
所有其他操作	300

下表列出 `CreateImportJob` 的檔案匯入配額。

作業	預設配額	是否可增加？
匯入檔案的數目上限	每個匯入任務 10,000 個檔案	否

若超出任一配額，Amazon Pinpoint 會對該請求進行限流—也就是拒絕其他有效的請求，並傳回 `TooManyRequests` 錯誤。您從帳戶對特定 AWS 區域的特定操作提出的請求總數，會影響限流與否。此外，每項操作也會獨立計算調節決策。假設 Amazon Pinpoint 對 `SendMessages` 操作的請求限流，對 `UpdateEndpoint` 操作的並行請求就會成功完成。

行銷活動配額

以下配額適用於 Amazon Pinpoint API 的行銷活動資源。

下列配額適用於 AWS 區域 ，有些可以增加。詳情請參閱 Service Quotas 使用者指南中的請求增加配額。

資源	預設配額	是否可增加？
作用中的行銷活動	每個帳戶 200 個 **Note** 作用中行銷活動是尚未完成或失敗的行銷活動。作用中行銷活動具有 `SCHEDULED` 、`EXECUTING` 或 `PENDING_NEXT_RUN` 的狀態。	否
區段大小上限	匯入的客群：每個行銷活動 100,000,000 個 動態區段：無限制	否

資源	預設配額	是否可增加？
基於事件的行銷活動	每個專案最多可以包含 25 個在事件發生時傳送的行銷活動。 使用基於事件的觸發的行銷活動必須使用動態客群。它們無法使用匯入的客群。 如果您使用 AWS Mobile SDK 將應用程式與 Amazon Pinpoint 整合，則來自事件型行銷活動的訊息只會傳送給應用程式執行 適用於 Android 的 AWS Mobile SDK 版本 2.7.2 或更新版本，或 AWS Mobile SDK for iOS 版本 2.6.30 或更新版本的客戶。 如果 Amazon Pinpoint 不能在五分鐘內交付事件型行銷活動的訊息，Amazon Pinpoint 將放棄該訊息，且不會嘗試重新交付。	否

電子郵件配額

下列各節中的配額適用於電子郵件管道。

電子郵件訊息配額

資源	預設配額	是否可增加？
訊息大小上限 (包括附件)	每則訊息 10 MB	否
驗證的身分數量	10,000 個身分	否

資源	預設配額	是否可增加？
	Note 身分是指電子郵件地址或網域，或兩者的任何組合。您使用 Amazon Pinpoint 傳送的每封電子郵件，都必須經由驗證過的身分。	

電子郵件寄件者和收件人配額

資源	預設配額	是否可增加？
寄件者地址	必須驗證所有傳送地址或網域。	否
收件人地址	如果您的帳戶在沙盒內，則必須驗證所有收件人電子郵件地址或網域。 如果您的帳戶在沙盒外，您可以傳送到任何有效的地址。	是
每則訊息的收件人數量	每則訊息 50 個收件人	否
您可以驗證的身分數量	每個 AWS 區域 10，000 個身分 **Note** 身分是指電子郵件地址或網域，或兩者的任何組合。您使用 Amazon Pinpoint 傳送的每封電	否

資源	預設配額	是否可增加？
	子郵件，都必須經由驗證過的身分。	

電子郵件傳送份額

同一區域的兩個服務之間，傳送配額、傳送速率和沙盒限制都相同。如果已將您移除沙盒，而您曾提高配額/速率，您在 us-east-1 使用 Amazon SES 時，這些變更將套用到您在 us-east-1 的 Pinpoint 帳戶。

資源	預設配額	是否可增加？
每 24 小時期間內可傳送的電子郵件數量 (傳送份額)	若您的帳戶位於沙盒內，則配額為每 24 個小時 200 封電子郵件。 如果您的帳戶在沙盒外，配額會根據您的特定使用案例而不同。 **Note** 此配額是根據收件人數量，而不是傳送的唯一訊息數量。收件人是「收件者：」列中的任何電子郵件地址。	是
每秒可傳送的電子郵件數量 (傳送率)	若您的帳戶位於沙盒內，則配額為每秒 1 封電子郵件。 如果您的帳戶在沙盒外，速率會根據您的特定使用案例而不同。	是

資源	預設配額	是否可增加？
	Note 此速率是根據收件人數量，而不是傳送的唯一訊息數量。收件人是「收件者：」列中的任何電子郵件地址。	

端點配額

以下配額適用於 Amazon Pinpoint API 的端點資源。

每個端點支援的屬性數目上限是 250，端點大小上限是 15 KB。但屬性的數量可能受到端點總大小 (包括所有屬性) 的限制。如果將屬性加入範本時遇到問題，請減少每個屬性中的資料量或減少屬性數。

資源	預設配額	是否可增加？
端點大小	大小上限 15 KB	否
共同指派給 `Attributes` 、`Metrics` 和 `UserAttributes` 參數的屬性	每個應用程式的所有屬性參數是 250	否
指派給 `Attributes` 參數的屬性	每個應用程式的所有屬性參數是 250	否
指派給 `Metrics` 參數的屬性	每個應用程式的所有屬性參數是 250	否
指派給 `UserAttributes` 參數的屬性	每個應用程式的所有屬性參數是 250	否
屬性名稱長度	50 個字元	否
屬性值長度	100 個字元	否

資源	預設配額	是否可增加？
EndpointBatchRequest 承載中的EndpointBatchItem 物件	每個承載 100 個。承載大小不能超過 7 MB。	否
具有相同使用者 ID 的端點	每個使用者 ID 15 個唯一端點	否
指派給 Attributes 參數屬性的值	每個屬性 50 個	否
指派給 UserAttributes 參數屬性的值	每個屬性 50 個	否

端點匯入配額

以下配額適用於將端點匯入 Amazon Pinpoint。

資源	預設配額	是否可增加？
作用中匯入任務	每個帳戶 10 個 匯入任務只在執行時才計入此配額。匯入任務完成後，將不再計入此配額。	否
匯入大小	每個匯入任務 1 GB 例如，若每個端點的大小為 4 KB 或小於 4 KB，您便可以匯入 250,000 個端點。	否

事件擷取配額

下列配額適用於使用 AWS Mobile SDKs 和 Amazon Pinpoint API 的事件資源擷取事件。

資源	預設配額	是否可增加？
自訂事件類型的數量上限	每個應用程式 1,500 個	否
自訂屬性索引鍵的數量上限	每個應用程式 500 個	否
每個屬性的自訂屬性值數量上限	100，000。超過 100,000 個的號碼仍可註冊，但不能在 Amazon Pinpoint 分析主控台使用。	否
每個屬性索引鍵的字元數上限	50	否
每個屬性值的字元數上限	200. 如果超過 200 個字元，將捨棄該事件。	否
自訂指標索引鍵的數量上限	每個應用程式 500 個	否
請求中的事件數量上限	每個請求 100 個	否
請求的大小上限	4 MB	否
個別事件的大小上限	1,000 KB	否
每個事件的屬性索引鍵和指標索引鍵數量上限	每個請求 40 個	否

行程配額

下列配額適用於行程。

下列配額適用於 AWS 區域 ，有些可以增加。詳情請參閱 Service Quotas 使用者指南中的請求增加配額。

資源	預設配額	是否可增加？
Maximum number of active journeys (作用中行程數目上限)	每個 帳戶 50 個	否
作用中的 EventTriggeredJourneys 數目上限	每個帳戶 20 個	否
行程活動次數上限	每行程 40 項	否
區段大小上限	匯入的客群：每個旅程 100,000,000 個。 動態區段：無限制	否
聯絡中心活動上限	每個旅程 3 個	否
關閉天數上限規則	每個頻道 20 個	否
關閉日規則名稱的最大長度	150 個字元	否
關閉天數規則開始和結束間的天數上限	7 天	否
開放小時數上限規則	每天 4 個	否

Lambda 配額

以下配額適用於專門擷取及處理 Lambda 資料的 Amazon Pinpoint 組態。

資源	預設配額	是否可增加？
Lambda 函數的調用承載 (請求和回應) 的大小上限	6 MB	否
等待 Lambda 函數處理資料的時間上限	15 秒	否

資源	預設配額	是否可增加？
每個端點的事件屬性數目上限	5	否
事件屬性名稱的字元數目上限	128 個字元	否
事件屬性值的字元數目上限	128 個字元	否
每個旅程可以執行的天數上限	540 天	否

Machine Learning 配額

以下配額適用於專門擷取及處理機器學習 (ML) 模型資料的 Amazon Pinpoint 組態。

資源	預設配額	是否可增加？
模型組態的最大數目	每個訊息範本 1 個 每個帳戶 100 個	否
建議數目上限	每端點或使用者 5 個	否
每個端點或使用者的建議屬性數目上限	1，如果 AWS Lambda 函數沒有處理屬性值 10，如果 AWS Lambda 函數有處理屬性值	否
建議屬性名稱的最大長度	屬性名稱為 50 個字元 屬性顯示名稱為 25 個字元 (顯示在主控台 Attribute finder (屬性搜尋工具) 中的名稱)	否
從 Amazon Personalize 擷取的建議屬性值的長度上限	100 個字元	否
Lambda 函數的調用承載 (請求和回應) 的大小上限	6 MB	否

資源	預設配額	是否可增加？
等待 Lambda 函數處理資料的時間上限	15 秒	否
嘗試調用 Lambda 函數的次數上次	3 次嘗試	否

可能適用額外配額，視您設定 Amazon Pinpoint 使用 ML 模型的方式而定。若要進一步了解 Amazon Personalize 配額，請參閱 Amazon Personalize 開發人員指南的配額。若要進一步了解 AWS Lambda 配額，請參閱 AWS Lambda 開發人員指南中的配額。

訊息範本配額

以下配額適用於您 Amazon Pinpoint 帳戶的訊息範本。

資源	預設配額	是否可增加？
訊息範本數量上限	每個帳戶 20,000 個	否
版本數量上限	每個範本 5,000	否
電子郵件範本中的字元數上限	600,000 個字元	否
應用程式內範本的字元數目上限	200,000 個字元	否
推送通知範本中預設範本部分內的字元數上限	4,000 個字元	否
推送通知範本中 ADM 限定範本部分內的字元數上限	6,000 個字元	否
推播通知範本的 APN 專用範本部分的字元數上限	4,000 個字元	否
推送通知範本中百度限定範本部分內的字元數上限	4,000 個字元	否

資源	預設配額	是否可增加？
推送通知範本中 FCM 專用範本部分的字元數上限	4,000 個字元	否
簡訊範本中的字元數上限	1,600 個字元	否
語音範本中的字元數上限	10,000 個字元	否

推播通知配額

以下配額適用於 Amazon Pinpoint 透過推播通知管道傳送的訊息。

資源	預設配額	是否可增加？
行銷活動中每秒可傳送的推播通知數目上限	每秒 25,000 個通知	是
Amazon Device Messaging (ADM) 訊息承載大小	每則訊息 6 KB	否
Apple 推播通知服務 (APN) 訊息承載大小	每則訊息 4 KB	否
APN 沙盒訊息承載大小	每則訊息 4 KB	否
百度雲推送訊息承載大小	每則訊息 4 KB	否
Firebase Cloud Messaging (FCM) 訊息承載大小	每則訊息 4 KB	否

應用程式內訊息配額

以下配額適用於您透過 Amazon Pinpoint 管理的應用程式內訊息。

下列配額適用於 AWS 區域 ，有些可以增加。詳情請參閱 Service Quotas 使用者指南中的請求增加配額。

資源	預設配額	是否可增加？
每秒可呼叫 GetInAppMessages API 的次數目上限。	每秒 5,000 個請求	是
應用內訊息傳訊行銷活動	每個專案最多可以包含 25 個使用應用程式內傳訊管道的行銷活動。	是，請參閱 Service Quotas 使用者指南中的請求增加配額

區段配額

以下配額適用於 Amazon Pinpoint API 的客群資源。

資源	預設配額	是否可增加？
可用來建立客群的維度數量上限	每個客群 100 個	否
每個客群的客群群組數目上限	5	否
每個客群的來源客群數目上限	5	否
來源客群的最大深度。 假設客群的來源客群也有來源客群，深度鏈不得超過此限制。	5	否

簡訊配額

如需簡訊配額，請參閱 AWS 《最終使用者簡訊使用者指南》中的簡訊配額。

10DLC 配額

如需 10DLC 配額，請參閱 AWS 最終使用者傳訊簡訊使用者指南中的 10DLC 配額。

語音配額

如需語音配額，請參閱 AWS 《最終使用者傳訊簡訊使用者指南》中的語音配額。

請求提高配額

若上述任何表格中是否可以增加 欄內的值為是，您便可以請求提高該配額。

請求提高配額

1. 在 https AWS 管理主控台 ：//https://console.aws.amazon.com/ 登入 。
2. 在 https：//https://console.aws.amazon.com/support/home#/case/create 建立新的 AWS 支援案例。
3. 在您的支援案例窗格中，選擇建立案例。
4. 選擇希望提高服務限制？ 連結。
5. 在增加服務配額下，針對服務，選擇下列其中一個選項：
   - 若要請求提高與電子郵件通路相關的配額，請選擇 Pinpoint 電子郵件。
   - 若要請求提高 SMS 支出限制或 SMS 傳送率的配額，請選擇 Pinpoint SMS。針對所有其他 SMS 配額增加，請選擇 Pinpoint
   - 若要請求提高與語音通路相關的配額，請選擇Pinpoint 語音。
   - 若要請求提高與任何其他 Amazon Pinpoint 功能相關的配額，請選擇 Pinpoint。
6. 視您選擇的服務而定，系統可能會要求您輸入下列項目：
   - (選用) 對於提供將傳送簡訊的網站或應用程式連結，提供有關將傳送簡訊之網站、應用程式或服務的資訊。
   - (選用) 對於您打算傳送的訊息類型，請選擇您計劃使用長碼傳送的訊息類型。
     - 一次性密碼 - 提供密碼給客戶以向網站或應用程式進行身分驗證的訊息。
     - Promotional (促銷) - 提升您的業務或服務的非重要訊息，例如，特殊優惠或公告。
     - Transactional (交易) - 支援客戶交易的重要資訊訊息，例如訂單確認或帳戶提醒。交易訊息不得包含促銷或行銷內容。
   - （選用）針對 AWS 您要傳送訊息的區域，選擇您要傳送訊息的區域。
   - (選用) 對於您打算傳送訊息的國家/地區，輸入您要購買短碼的國家或區域。
   - (選用) 在您的客戶如何選擇接收您發送的訊息中，提供有關您選擇加入程序的詳細資訊。
   - (選用) 在請提供您計劃用來傳送訊息給客戶的訊息範本欄位中，包含您將使用的範本。

7. 在 Requests (請求) 下，執行下列動作：
   - 針對區域選擇您的 AWS 區域。
   - 針對資源類型，選擇一般限制。資源類型欄位僅適用於某些 服務。
   - 對於配額，選擇要變更的配額。
   - 對於新配額值，輸入配額的新值。
   - 若要請求增加其他 中的相同配額 AWS 區域，請選擇新增另一個請求，然後選擇其他 AWS 區域請求並填寫新的請求。
8. 選擇您想要提高的配額，然後輸入您要的新配額值。
9. 在案例描述下，釐清您請求提高配額的原因。
10. 在聯絡選項下，針對偏好的聯絡語言，選擇與 AWS 支援團隊通訊時偏好使用的語言。
11. 針對聯絡方法，選擇您偏好的與 AWS 支援團隊通訊方法。
12. 選擇提交。

AWS 支援團隊會在 24 小時內對您的請求提供初始回應。

為了避免使用我們的系統被用來傳送未經要求或惡意的內容，我們必須仔細考慮每個請求。如果我們能夠這麼做，我們在此 24 小時的期間內准許您的請求。不過，如果我們需要向您取得其他資訊，則可能需要更長的時間來解決您的請求。

如果您的使用案例不符合我們的政策，我們可能無法批准您的請求。

Amazon Pinpoint 的文件歷史記錄

下表說明 2018 年 12 月後，Amazon Pinpoint 開發人員指南每個版本的重要變更。如需有關此文件更新的通知，您可以訂閱 RSS 訂閱源。

- 文件最新更新時間：2023 年 11 月 16 日

變更	描述	日期
終止支援通知	終止支援通知：自 2026 年 10 月 30 日起， AWS 將終止對 Amazon Pinpoint 的支援。2026 年 10 月 30 日之後，您將無法再存取 Amazon Pinpoint 主控台或 Amazon Pinpoint 資源 (端點、區段、行銷活動、旅程和分析)。如需詳細資訊，請參閱 Amazon Pinpoint 終止支援。注意：與 SMS、語音、行動推播、OTP 和電話號碼驗證相關的 APIs 不受此變更影響，並受 AWS 最終使用者傳訊支援。	2025 年 5 月 20 日
新增傳送訊息的 CloudTrail 支援	新增 PutEvents、SendUserMessages 和 SendMessages API 呼叫的 CloudTrail 記錄支援，請參閱 CloudTrail 日誌檔案中支援的 Amazon Pinpoint API 動作。	2025 年 2 月 17 日
Amazon Pinpoint 已經更新了用戶指南文件	若要取得如何建立、設定和管理推播資源的最新資訊，請參閱新的最終使用者傳訊推播使用者指南。	2024 年 7 月 22 日

電子郵件標頭	您可以將電子郵件標頭新增至電子郵件訊息。如需詳細資訊，請參閱使用取消訂閱標頭傳送電子郵件。	2024 年 5 月 7 日
電子郵件協同運作	Amazon Pinpoint 已更新使用 Amazon SES 資源傳送電子郵件的方式。如需詳細資訊，請參閱使用 Amazon SES 傳送電子郵件的 IAM 角色。	2024 年 4 月 30 日
Amazon Pinpoint 已經更新了用戶指南文件	SMS 和語音資源管理主題現在會重新導向至AWS 最終使用者傳訊簡訊使用者指南。如需詳細資訊，請參閱AWS 最終使用者傳訊簡訊使用者指南。	2024 年 2 月 8 日
Amazon Pinpoint 配額	新增配額：關閉天數上限規則、關閉天數上限規則名稱、關閉天數規則開始和結束間的天數上限、以及開放小時數上限規則。詳情請參閱 Amazon Pinpoint 配額。	2023 年 12 月 19 日
Amazon Pinpoint 已經更新了用戶指南文件	若要取得如何建立、設定和管理 AWS 最終使用者簡訊簡訊和語音資源的最新資訊，請參閱新的AWS 最終使用者簡訊簡訊使用者指南。	2023 年 11 月 16 日
Amazon Pinpoint 配額	更新了 UpdateEndpointsBatch、UpdateEndpoint、PutEvents、DeleteEndpoint 和 GetEndpoint 的配額。詳情請參閱 Amazon Pinpoint 配額。	2023 年 9 月 22 日

Amazon Pinpoint 配額	更新了 CreateEmailTemplate、CreateSmsTemplate、CreatePushTemplate、CreateInAppTemplate、CreateVoiceTemplate、UpdateEmailTemplate、UpdateSmsTemplate、UpdatePushTemplate、UpdateInAppTemplate、UpdateVoiceTemplate 和 CreateImportJob 的配額。詳情請參閱 Amazon Pinpoint 配額。	2023 年 9 月 12 日
旅程和行銷活動執行指標	為旅程和行銷活動新增了新的分析指標。詳情請參閱旅程和行銷活動執行指標。	2023 年 4 月 25 日
為 Amazon Pinpoint 建立介面 VPC 端點	Amazon Pinpoint 現在支援介面 VPC 端點。詳情請參閱為 Amazon Pinpoint 建立介面 VPC 端點。	2023 年 4 月 11 日
傳輸中加密	自 2023 年 3 月 22 日起，Amazon Pinpoint 將不再支援 TLS 1.0，但您仍然可以使用 TLS 1.2 (含) 以上版本。如需詳細資訊，請參閱傳輸中加密。	2023 年 3 月 20 日
Amazon Pinpoint 配額	更新了針對行銷活動、旅程和應用程式內訊息，請求增加配額的程序。詳情請參閱 Amazon Pinpoint 配額。	2022 年 12 月 16 日

區域可用性	Amazon Pinpoint 現已在以下區域推出：美國東部 (俄亥俄)。	2022 年 10 月 5 日
IAM 角色範例更新	更新了整份文件中的多個 IAM 角色範例，以進一步配合安全最佳做法。	2022 年 5 月 27 日
SMS 和語音 API 第 2 版	Amazon Pinpoint 現在包含用於傳送 SMS 和語音訊息的專用 API。此 API 包含組態集、集區和選擇不接收清單等新功能，這對於傳送交易 SMS 和語音訊息的客戶而言很有幫助。詳情請參閱使用 Amazon Pinpoint SMS 和語音 API。	2022 年 4 月 1 日
一次性密碼建立與驗證	Amazon Pinpoint 現在包含可產生一次性密碼 (OTP) 的功能，並以 SMS 訊息傳送給您的使用者。Amazon Pinpoint 也包含一個 API，當您的使用者將 OTP 密碼輸入應用程式或網站時，可為其驗證。詳情請參閱傳送和驗證一次性密碼 (OTP)。	2021 年 11 月 26 日
應用程式內訊息	新增了 Amazon Pinpoint 的應用程式內傳訊功能已與您應用程式整合的相關資訊。	2021 年 11 月 10 日
程式碼範例	新增了 Amazon Pinpoint 常見操作的程式碼範例庫。	2021 年 11 月 3 日
專案配額	Amazon Pinpoint 專案的最大數量維持在 100，但現在可以使用 開啟服務限制提高請求來提高此配額 支援。	2021 年 10 月 11 日

Lambda 政策更新。	某些 Lambda 許可政策現在必須包含 `AWS:SourceAccount` 條件。更新了在 Amazon Pinpoint 中建立自訂頻道和使用 AWS Lambda自訂客群主題中的範例政策，以符合此要求。	2021 年 10 月 7 日
UpdateEndpoint	Amazon Pinpoint UpdateEndpoint API 現在由 CloudTrail 記錄。	2020 年 11 月 16 日
自訂屬性	Amazon Pinpoint 現在支援電子郵件傳訊範本的 250 個屬性。請參閱配額。	2020 年 9 月 18 日
區域可用性	Amazon Pinpoint 現已在以下區域推出：亞太區域 (東京)、歐洲 (倫敦) 和加拿大 (中部)。請注意，Amazon Pinpoint SMS 和語音 API 在這些區域不可用。	2020 年 9 月 10 日
區域可用性	Amazon Pinpoint 現已在亞太區域 (東京) 推出。請注意，Amazon Pinpoint SMS 和語音 API 在此區域不支援語音。	2020 年 9 月 2 日
行銷活動事件	在行銷活動事件中，新增了新行銷活動事件 `delivery_type` 參數的相關資訊。	2020 年 8 月 2 日
區域可用性	Amazon Pinpoint 現已在亞太區域 (首爾) 可以使用。請注意，Amazon Pinpoint API 在此區域不支援語音或 SMS。	2020 年 7 月 31 日

區域可用性	Amazon Pinpoint 現在可在 AWS GovCloud (US) 區域中使用。	2020 年 4 月 30 日
自訂頻道	更新了使用 Lambda 函數或 Webhook 建立自訂頻道的相關資訊。	2020 年 4 月 23 日
機器學習	新增從推薦者模型擷取個人化建議的相關資訊，以及使用 AWS Lambda 函數選擇性地增強這些建議。	2020 年 3 月 4 日
安全性	新增了安全性章節，針對 Amazon Pinpoint 各種安全性控制項和功能，提供相關資訊。	2020 年 2 月 4 日
旅程	新增了使用 Amazon Pinpoint 旅程開發自動化工作流程的相關資訊，該自動化工作流程可為專案執行傳訊活動。也新增針對適用於旅程的指標子集查詢分析資料的相關資訊。	2019 年 10 月 31 日
分析	新增程序，說明如何針對行銷活動和交易訊息查詢分析資料，並新增使用查詢結果的相關資訊。	2019 年 10 月 17 日
分析	已針對適用於交易電子郵件和簡訊的指標子集新增查詢分析資料的相關資訊。	2019 年 9 月 6 日
程式碼範例	新增了程式碼範例，您可以利用這些範例，使用 Amazon Pinpoint 支援的所有服務傳送交易推播通知。	2019 年 7 月 30 日

分析	新增針對適用於專案 (應用程式) 和行銷活動的指標子集查詢分析資料的相關資訊。	2019 年 7 月 24 日
客群	新增了一個教學課程，介紹從外部系統 (例如 Salesforce 或 Marketo) 將客戶資料匯入 Amazon Pinpoint 的解決方案。	2019 年 5 月 14 日
區域可用性	Amazon Pinpoint 現已在 AWS 亞太區域（孟買）和亞太區域（雪梨）區域提供。	2019 年 4 月 25 日
將 Postman 與 Amazon Pinpoint 搭配使用	新增了一個教學課程，介紹如何使用 Postman 與 Amazon Pinpoint API 互動。	2019 年 4 月 8 日
標記	新增了標記 Amazon Pinpoint 資源的相關資訊。	2019 年 2 月 27 日
SMS 註冊	已新增一個教學課程章節，並新增了一個教學課程，描述如何建立一個處理 SMS 使用者註冊的解決方案。	2019 年 2 月 27 日
程式碼範例	已新增數種程式設計語言的程式碼範例，展示如何以程式設計方式傳送電子郵件、SMS 和語音訊息。	2019 年 2 月 6 日

舊版更新

下表說明截至 2018 年 12 月，Amazon Pinpoint 開發人員指南每個版本的重要變更。

變更	描述	日期
區域可用性	Amazon Pinpoint 現已在美國 AWS 西部（奧勒岡）和歐洲（法蘭克福）區域提供。	2018 年 12 月 21 日
語音管道	您可以使用新的 Amazon Pinpoint 語音管道建立語音訊息，然後透過電話將語音訊息傳遞給客戶。目前您只能使用 Amazon Pinpoint SMS 和語音 API 傳送語音訊息。	2018 年 11 月 15 日
歐洲 (愛爾蘭) 可用性	Amazon Pinpoint 現已在歐洲 AWS（愛爾蘭）區域提供。	2018 年 10 月 25 日
事件 API	使用 Amazon Pinpoint API 記錄事件，並將 Amazon Pinpoint API 與端點建立關聯。	2018 年 8 月 7 日
定義和查詢端點的程式碼範例	增加程式碼範例，以示範如何定義、更新、刪除和查詢端點。為 AWS CLI 適用於 Java 的 AWS SDK、 和 Amazon Pinpoint API 提供範例。如需詳細資訊，請參閱使用端點在 Amazon Pinpoint 中代表您的對象。	2018 年 8 月 7 日
端點匯出許可	設定可讓您將 Amazon Pinpoint 端點匯出到 Amazon S3 儲存貯體的 IAM 政策。	2018 年 5 月 1 日
簡訊的電話號碼驗證	使用 Amazon Pinpoint API 驗證電話號碼，判斷是否為 SMS 訊息的有效目標。	2018 年 4 月 23 日

變更	描述	日期
Amazon Pinpoint 整合的更新主題	使用 AWS SDKs 或程式庫，將 Amazon Pinpoint 與您的 Android、iOS 或 JavaScript 應用程式整合。 JavaScript	2018 年 3 月 23 日
AWS CloudTrail 記錄	新增了使用 CloudTrail 記錄 Amazon Pinpoint API 呼叫的相關資訊。	2018 年 2 月 6 日
更新的服務配額	以電子郵件配額的其他相關資訊來更新 配額。	2018 年 1 月 19 日
Amazon Pinpoint 擴充功能的公開測試版	使用 AWS Lambda 函數來自訂客群或建立自訂傳訊管道。	2017 年 11 月 28 日
推送通知承載配額	配額包括行動推送訊息的承載大小。	2017 年 10 月 25 日
更新的服務配額	將簡訊和電子郵件管道資訊新增到 配額。	2017 年 10 月 9 日
ADM 和百度行動推送	更新您的應用程式碼以處理來自百度和 ADM 行動推送管道的推送通知。	2017 年 9 月 27 日
使用者 ID 和驗證事件與 Amazon Cognito 使用者集區。	如果您使用 Amazon Cognito 使用者集區來管理行動應用程式中的使用者登入，Amazon Cognito 會將使用者 ID 指派給端點，並向 Amazon Pinpoint 回報驗證事件。	2017 年 9 月 26 日
使用者 ID	將使用者 ID 指派給端點，以監控個別使用者的應用程式使用情況。提供了 AWS Mobile SDK 和 SDK for Java 的範例。	2017 年 8 月 31 日

變更	描述	日期
身分驗證事件	報告身分驗證事件，以了解使用者向您的應用程式進行驗證的頻率。在您的應用程式中報告 Amazon Pinpoint 事件中提供範例。	2017 年 8 月 31 日
更新範例事件	範例事件包括 Amazon Pinpoint 為電子郵件和 SMS 活動串流的事件。	2017 年 6 月 8 日
Android 工作階段管理	使用 AWS Mobile Hub 範例應用程式提供的類別，在 Android 應用程式中管理工作階段。	2017 年 4 月 20 日
更新獲利事件範例	更新範本程式碼以報告獲利事件。 。	2017 年 3 月 31 日
事件串流	您可以設定 Amazon Pinpoint 將您的應用程式和行銷活動事件傳送到 Kinesis 串流。	2017 年 3 月 24 日
許可	Amazon Pinpoint 如何搭配 IAM 運作 如需為 帳戶中 AWS 的使用者和行動應用程式的使用者授予 Amazon Pinpoint 存取權的相關資訊，請參閱 。	2017 年 1 月 12 日
Amazon Pinpoint 一般可用性	此版本引入了 Amazon Pinpoint。	2016 年 12 月 1 日