Amazon Personalize

Developer Guide

aws

Amazon Personalize: Developer Guide

Table of Contents

This is prerelease documentation for a service in preview release. It is subject to change.

We made breaking changes to the Amazon Personalize API and service model on 02/20/19. To continue using Amazon Personalize with the AWS Command Line Interface or AWS SDK for Python (Boto 3), update your service JSON files by doing steps 3-6 of Setting Up the AWS CLI (p. 8).

What Is Amazon Personalize?

Amazon Personalize is a machine learning service that makes it easy for developers to add individualized recommendations to customers using their applications. It reflects the over 20 years of experience that Amazon has in building personalization systems.

You can use Amazon Personalize in a variety of scenarios, such as giving users recommendations based on their preferences and behavior, personalized re-ranking of search results, and personalizing content for emails and notifications.

Amazon Personalize does not require extensive machine learning experience and you can build, train, and deploy a solution (a trained Amazon Personalize recommendation model) with the AWS console or programatically by using the AWS SDK. As the developer, you only need to do the following:

- Format input data and upload the data into an Amazon S3 bucket, and/or send real-time event data.
- Select a training algorithm (recipe) to use on the data.
- Train a solution using the recipe.
- Deploy the solution.

Amazon Personalize can capture live events from your users to achieve real-time personalization. Amazon Personalize can blend real-time user activity data with existing user profile and item information to recommend the most relevant items, according to the user's current session context and activity. You can also use Amazon Personalize to collect data for new properties, such as a brand new website, and once enough data has been collected, Amazon Personalize can start to make recommendations.

To give recommendations to your users, you can call Amazon Personalize via an API, and then create personalized experiences for them.

Amazon Personalize can improve its recommendations over time as new user activity data is collected. For example, a new movie rental event by a user, which is used in retraining of a solution, results in better movie recommendations. You can retrain an Amazon Personalize solution on a scheduled basis, or as needed.

With Amazon Personalize you can train a solution for different use cases, such as user personalization, items related to an item, and re-ranking of items. You choose a recipe based on your use case and provide the input data. A recipe does featurization of your data, and applies a choice of learning algorithms, along with default hyperparameters, and hyperparameter optimization job configuration.

Recipes in Amazon Personalize allow you to create custom personalization models without needing machine learning expertise. You can choose which recipe to use to train a solution, or let Amazon Personalize decide on the best recipe to use for your data. To help you decide which recipe to use, Amazon Personalize provides extensive metrics on the performance of a trained solution. While easy to use with pre-built recipes, Amazon Personalize is also deeply customizable and allows you to add your own recipes.

Are You a First-Time Amazon Personalize User?

If you are a first-time user of Amazon Personalize, we recommend that you read the following sections in order:

1. **How It Works (p. 3)** – This section introduces various Amazon Personalize components that you work with to create an end-to-end experience.

2. **Getting Started (p. 11)** – In this section you set up your account, and test the Amazon Personalize Console and API.
3. **Preparing and Importing Data (p. 28)** – This section describes how to prepare and import your training data into Amazon Personalize.
4. **Recording Events (p. 35)** – This section provides information about improving user recommendations by recording user events and retraining the solution.
5. **Creating a Solution (p. 41)** – This section provides information about training and deploying a solution.
6. **Creating a Campaign (p. 59)** – This section provides information about deploying a solution as a campaign.
7. **Getting Recommendations (p. 60)** – This section show how to get recommendations from a campaign.

How It Works

Amazon Personalize is a machine learning service for personalization and recommendations. To use Amazon Personalize, you first train a model from data that you supply. Your users can then receive recommendations through your application or website. For example, you might train a model to make movie recommendations based on the viewing habits of similar users. You could then use the model to make recommendations to a user who is signed in to a movie ticket sales application.

To make recommendations, Amazon Personalize uses an artificial intelligence model that is trained with your data. The data used to train the model is stored in related datasets in a dataset group. Each model is trained by using a recipe that contains algorithms for unique use cases. In Amazon Personalize, a trained model is known as a solution. A solution is deployed for use in a campaign. Users of your applications can receive recommendations through the campaign.

A dataset can grow over time and your models retrained on the new data. The data can come from new metadata and the consumption of real-time user event data. In the previous movie recommendations example, you could add new movies as they are released, as well as add a movie that is chosen by the signed-in user.

Amazon Personalize has an AWS console that you can use to create, manage, and deploy solutions. Alternatively, you can use the AWS Command Line Interface (AWS CLI) or one of the Amazon Personalize SDKs.

Amazon Personalize consists of three related components:

- Amazon Personalize – Use this to create, manage, and deploy solutions.
- Amazon Personalize Events – Use this to record user events for further training of solutions. For more information, see Recording Events (p. 35).
- Amazon Personalize Runtime – Use this to get recommendations from a campaign (deployed solution). For more information, see Getting Recommendations (p. 60).

Amazon Personalize Workflow

The workflow for training, deploying, and using a solution is:

1. Create datasets and a dataset group.
2. Get training data.
   - Import historical data to the dataset group.
   - Record user events to the dataset group.
3. Create a solution (trained model) using a recipe.
4. Evaluate the solution using metrics.
5. Create a campaign (deploy the solution).
6. Provide recommendations for users.

This guide provides Getting Started (p. 11) exercises for doing this workflow with the Amazon Personalize console and with the AWS CLI. Python examples for the workflow are included in other sections of the guide.

Datasets and Dataset Groups

Amazon Personalize requires data, stored in Amazon Personalize datasets, to train a model.

You have two means to provide the training data. You can import historical data from an Amazon S3 bucket, and you can record data as it is created.

A dataset group contains related datasets, three created by the customer (users, items, and historical interactions), and one created by Amazon Personalize for live event interactions. A dataset group can contain only one of each kind of dataset.

You can create datasets groups to serve different purposes. For example, you might have an application that provides recommendations for purchasing shoes and another that gives recommendations for places to visit in Europe. In Amazon Personalize, each application would have its own dataset group.

Historical data must be provided in a CSV file. Each dataset type has a unique schema that specifies the contents of the file.

There is a minimum amount of data that is necessary to train a model. Using existing historical data allows you to immediately start training a solution. If you ingest data as it is created, and there is no historical data, it can take a while before training can begin.

You can configure Amazon Personalize to upload training data on a pre-determined schedule. You can use the Amazon Personalize console to import data into a dataset. Alternatively you can use the AWS SDK.

For more information, see .

User Events

Amazon Personalize can consume real time user event data to be used for model training either alone or combined with historical data.

For more information, see .

Recipes and Solutions

Once enough data is available in the interactions datasets (historical and live events), the data can be used to train a model. A trained model is known as a solution. The model is trained using a recipe. A recipe is an algorithm and the data processing steps that optimize the solution for making recommendations based on your input data.

Amazon Personalize supports a number of predefined recipes. Amazon Personalize can automatically choose the most appropriate recipe based on its analysis of the training data. Alternatively, you can choose which recipe to train the model on. Each recipe has its own use case and you should choose the recipe that best fits your needs. For more information, see . You can also provide your own recipes by using the AWS SDK. For more information, see .

You can specify a schedule for how often a solution is automatically re-trained. This is useful for updating a solution using data recently added to the associated dataset. You can also configure the solution to retrain on a manual basis.

For more information, see .

Metrics

Each time you train a solution, it is assigned a new version. The version is provided to you as an Amazon Resource Name (ARN) in the response to the `CreateSolution` and `RetrainSolution` APIs. Use the solution version ARN to identify which version of the solution you want to use for your campaign.

For more information, see Evaluating Solutions (p. 57).

Campaigns

A deployed solution is known as a campaign, and is able to make recommendations for your users. To deploy a solution, you create a campaign in the console or by calling the the section called "CreateCampaign" (p. 80) API. You can choose which version of the solution to use. By default, a campaign uses the latest version of a solution.

For more information, see Creating a Campaign (p. 59).

Recommendations

After you create a campaign, you are able to get two different types of recommendations, dependent on what recipe type was used to train the model.

For user-personalization and related-items recipes, the GetRecommendations (p. 208) API returns a list of recommended items. For example, movies can be recommended for users signed in to your website.

For search-personalization recipes, the PersonalizeRanking (p. 211) API re-ranks a list of recommended items based on a specified query.

For more information, see Getting Recommendations (p. 60).

Setting Up

Before using Amazon Personalize, you must have an Amazon Web Services (AWS) account. Once you have an AWS account, you can access Amazon Personalize through the Amazon Personalize console, the AWS Command Line Interface (AWS CLI), or the AWS SDKs.

This guide includes examples for AWS CLI, Python, and JavaScript with AWS Amplify.

Topics

Sign Up for AWS

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all services in AWS, including Amazon Personalize. You are charged only for the services that you use.

If you have an AWS account already, skip to the next task. If you don't have an AWS account, use the following procedure to create one.

To sign up for AWS

1. Open https://aws.amazon.com, and then choose **Create an AWS Account**.
2. Follow the on-screen instructions to complete the account creation. Note your 12-digit AWS account number. Part of the sign-up procedure involves receiving a phone call and entering a PIN using the phone keypad.
3. Create an AWS Identity and Access Management (IAM) admin user. See Creating Your First IAM User and Group in the *AWS Identity and Access Management User Guide* for instructions.
4. Create an IAM user for use with Amazon Personalize. The account will need certain permissions. For more information, see Required Permissions (p. 7).

Endpoints

An endpoint is a URL that is the entry point for a web service. For the endpoints supported by Amazon Personalize, see Regions and Endpoints.

Setting up Permissions

Before you can use Amazon Personalize, you have to set up permissions that allow access to the Amazon Personalize console and access to the Amazon Personalize API operations. You also have to allow Amazon Personalize to perform tasks on your behalf and access resources that you own, such as the Amazon S3 Bucket that contains your training data.

Topics

- Creating an IAM Role (p. 7)

Required Permissions

Whilst you can use An IAM user with administrator permissions, we recommend the following permissions for restricting access to Amazon Personalize. You can add other add other permissions as you need them. For information about restricting access to Amazon Personalize operations, see Using Identity-Based Policies (IAM Policies) for Amazon Personalize (p. 67). The following AWS managed policies are required.

- AmazonPersonalizeFullAccess — Gives access to the Amazon Personalize console and API operations.
- AmazonS3FullAccess — Gives permissions to upload your training data into an Amazon S3 Bucket.
- IAMReadOnlyAcess — Give read-only permissions to access the IAM console.

You also need to attach the following inline policy the IAM user. The inline policy gives the IAM user permission to create an IAM service role. The service role is required to allow Amazon Personalize to perform tasks on behalf of the user.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:CreateRole",
                "iam:CreateInstanceProfile",
                "iam:UpdateAssumeRolePolicy",
                "iam:AddRoleToInstanceProfile",
                "iam:PassRole",
                "iam:PutRolePolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

For more information about AWS managed policies and inline policies, see Managed Policies and Inline Policies.

You can add permissions to directly to the IAM user. We recommend adding the permissions to an IAM group and adding the user to the IAM group. For more information, see Creating IAM Groups.

Creating an IAM Role

Amazon Personalize sometimes needs to access to your resources and perform tasks on your behalf. For example, Amazon Personalize uploads training data from an Amazon S3 Bucket that you own when performing a dataset import job. The following steps show you how to create an AWS Identity and Access Management (IAM) role that allows Amazon Personalize to perform tasks on your behalf. You also have to give Amazon Personalize access to specific resources. For example, access to an Amazon S3 Bucket for dataset import. For more information, see Uploading to an Amazon S3 Bucket (p. 31).

To create an IAM role

1. Sign in to the IAM console (https://console.aws.amazon.com/iam).
2. In the navigation pane, choose **Roles**.

3. Choose **Create role**.

4. For **Select type of trusted entity**, choose **AWS service**.

5. For **Choose the service that will use this role**, if you don't see **Amazon Personalize** listed, choose **EC2**. Otherwise, choose **Amazon Personalize**.

6. Choose **Next: Permissions**.

7. For **Attach permissions policies**, choose the check box next to the following policies. To display a policy in the list, type part of the policy name in the **Filter policies** query filter.

    a. *AmazonS3ReadOnlyAccess*

    b. *CloudWatchFullAccess*

8. Choose **Next: Tags**.

9. You don't need to add tags, so choose **Next: Review**.

10. In the **Review** section, for **Role name**, enter a name for the role (for example, `PersonalizeRole`). Update the description for the role in **Role description**, then choose **Create role**.

11. Choose the new role to open the role's details page.

12. In the **Summary**, copy the **Role ARN** value and save it. You need it to import a dataset into Amazon Personalize.

13. If you didn't choose **Amazon Personalize** as the service that will use this role, choose **Trust relationships**.

14. Choose **Edit trust relationship** and update the trust policy as follows.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "personalize.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

15. Choose **Update Trust Policy**.

Setting Up the AWS CLI

The AWS Command Line Interface (AWS CLI) is a unified developer tool for managing AWS services, including Amazon Personalize. We recommend that you install it.

1. To install the AWS CLI, follow the instructions in Installing the AWS Command Line Interface in the *AWS Command Line Interface User Guide*.

2. To configure the AWS CLI and set up a profile to call the AWS CLI, follow the instructions in Configuring the AWS CLI in the *AWS Command Line Interface User Guide*.

3. Download the following service json files (right-click each file, then select **Save As**):

    - personalize.json
    - personalize-runtime.json
    - personalize-events.json

4. Use the following commands to install the service json files you downloaded in the previous step.

    a. Amazon Personalize

```
aws configure add-model --service-name personalize --service-model file://
personalize.json
```

b.   Amazon Personalize Runtime

```
aws configure add-model --service-name personalize-runtime --service-model file://
personalize-runtime.json
```

c.   Amazon Personalize Events

```
aws configure add-model --service-name personalize-events --service-model file://
personalize-events.json
```

5.   To confirm that the AWS CLI profile is configured properly, run the following command in a command window:

```
aws configure --profile default
```

If your profile has been configured correctly, you should see output similar to the following:

```
AWS Access Key ID [****************52FQ]:
AWS Secret Access Key [****************xgyZ]:
Default region name [us-west-2]:
Default output format [json]:
```

6.   To verify that the AWS CLI is configured for use with Amazon Personalize, run the following commands.

```
aws personalize help
```

and

```
aws personalize-runtime help
```

and

```
aws personalize-events help
```

If the AWS CLI is configured correctly, you will see a list of the supported AWS CLI commands for Amazon Personalize, Amazon Personalize runtime, and Amazon Personalize events.

Setting Up the AWS SDKs

Download and install the AWS SDKs that you want to use. This guide provides examples for Python and JavaScript. For information about other AWS SDKs, see Tools for Amazon Web Services.

- AWS SDK for Python (Boto 3)

  You also need to install the service json files for Amazon Personalize (**PREVIEW RELEASE**). For more information, see Setting Up the AWS CLI (p. 8).

  To confirm that your python environment is configured correctly for use with Amazon Personalize, see Getting Started (AWS SDK for Python) (p. 26).

- AWS Amplify

Getting Started

This getting started guide shows you how to train and deploy an Amazon Personalize solution that returns movie recommendations for a given user. To train the solution, you use historical data that consists of 100,000 movie ratings on 9700 movies from 600 users.

To simplify this guide:

- We rely on the simply fact that a user saw a movie and not on what they rated the movie. This simplifies the preparation of the training data.
- We don't record live user interaction events. For information on capturing user events, see Recording Events (p. 35).

To begin, you download and prepare the training data. Next, you create an AWS Identity and Access Management (IAM) role that allows Amazon Personalize to access the data on your behalf. After creating the training data and role, you can proceed to either Getting Started (Console) (p. 12) or Getting Started (AWS CLI) (p. 18).

Prerequisites

The following steps are prerequisites for the Getting Started exercises. Individual exercises might have further prerequisites.

- Create an AWS account and an AWS Identity and Access Management user, as specified in Sign Up for AWS (p. 6).
- Ensure the IAM user that you are using has the Required Permissions (p. 7).
- Create an AWS Identity and Access Management (IAM) service role, as specified in Creating an IAM Role (p. 7). You will use the role ARN when you upload the movie training data.
- Prepare your training data and upload the data to your Amazon S3 bucket, as specified in Create the Training Data (p. 11). You will use the name of the Amazon S3 bucket when you upload the movie training data.

Create the Training Data

To create training data, you download, modify, and save the movie ratings data to a Amazon Simple Storage Service (Amazon S3) bucket. Then, you give Amazon Personalize permission to read from the bucket.

1. Download the movie ratings zip file, ml-latest-small.zip from MovieLens (under *recommended for education and development*). Unzip the file. The user-interactions data is in the file named `ratings.csv`.
2. Open the `ratings.csv` file.

    a. Delete the *rating* column.

    b. Replace the header row with the following:

    `USER_ID,ITEM_ID,TIMESTAMP`.

These headers must be exactly as shown for Amazon Personalize to recognize the data.

Save the `ratings.csv` file.

3. Upload `ratings.csv` to your Amazon S3 bucket. For more information, see Uploading Files and Folders by Using Drag and Drop in the Amazon Simple Storage Service Console User Guide.

4. Grant Amazon Personalize permission to read the data in the bucket. Attach the following policy to the bucket. Change the value of `bucketname` to the name of the Amazon S3 bucket that you used in step 3. For more information, see How Do I Add an S3 Bucket Policy? in the Amazon Simple Storage Service Console User Guide.

```
{
    "Version":"2012-10-17",
    "Id":"PersonalizeS3Bucket AccessPolicy",
    "Statement":[
        {
            "Sid": "PersonalizeS3BucketAccessPolicy",
            "Effect": "Allow",
            "Principal": {
                "Service":"personalize.amazonaws.com"
            },
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
             ],
            "Resource":[
                "arn:aws:s3:::bucketname",
                "arn:aws:s3:::bucketname/*"
            ]
        }
    ]
}
```

Getting Started (Console)

In this exercise, you use the Amazon Personalize console to create a campaign that returns movie recommendations for a given user.

Before you start this exercise, do the Getting Started Prerequisites (p. 11).

Step 1: Import Training Data

In this procedure, you import your training data into a *user-item interaction* dataset and create a group for the dataset.

To import training data

1. Sign in to the AWS Management Console and open the Amazon Personalize console at https://console.aws.amazon.com/personalize/.

2. In the **New dataset group** box, choose **Get started**.

3. Under **Dataset group details**, in the **Dataset group name** field, specify a name for your dataset group.

4. Under **Input data selection**, keep the default selection of **Upload user-item interaction data**.

   Your screen should look similar to the following:

5. Choose **Next**.

6. For **Schema selection**, choose **Create new schema**. A minimal Interactions schema is displayed in the **Schema definition** field. The schema matchs the headers you previously added to the `ratings.csv` file.

7. In the **New schema name** field, specify a name for the new schema.

    Your screen should look similar to the following:

Schema details

Schema selection Info

○ **Use existing schema**
Choose an existing schema that matches your dataset.

● **Create new schema**
Create a new schema to match your dataset

New schema name
The text you type here appears in the Schema dashboard. It can help you distinguish this schema from others.

ratings-schema

The schema name must have 2 to 256 characters. Valid characters: a-z, A-Z, 0-9, and . : + = @ % - (hyphen).

Schema definition Info
Ensure your dataset's schema matches the following schema.

```
 1  {
 2      "type": "record",
 3      "name": "Interactions",
 4      "namespace": "com.amazonaws.personalize.schema",
 5      "fields": [
 6          {
 7              "name": "USER_ID",
 8              "type": "string"
 9          },
10          {
11              "name": "ITEM_ID",
12              "type": "string"
13          },
14          {
15              "name": "TIMESTAMP",
16              "type": "long"
17          }
18      ],
19      "version": "1.0"
20  }
```

Cancel   Previous   **Next**

8. Choose **Next**.

9. In the **Dataset import job name** field, specify a name for the job.

10. Under **Automatic dataset import jobs**, keep the default selection of **Off**.

11. For **IAM Role**, choose **Enter a custom IAM role ARN**.

12. In the **Custom IAM role ARN** field, specify the role that you created in Creating an IAM Role (p. 7).

13. In the **Data location** field, specify the place that your movie data file in CSV format is stored within Amazon Simple Storage Service (S3) by using the following syntax:

    ```
    s3://<name of your S3 bucket>/<folder path>/<CSV filename>
    ```

Your screen should look similar to the following:

14. Choose **Finish**. The dataset will take a while to be uploaded.

Step 2: Create a Solution

In this procedure, use your dataset to create and train a model, which is known as a *solution*. The dataset must be uploaded before you can create a solution. For more information, see *Step 1: Import Training Data*.

To create and train a solution

1. In the navigation pane, under the dataset group that you created, choose **Dashboard**.
2. Choose **Create solutions**.
3. In the **Solution name** field, specify a name for your solution.
4. For **Recipe selection**, choose **Automatic (AutoML)**.
5. For the **Event type** and **Minimum transactions per second** fields, leave the default values.
6. Your screen should look similar to the following:

7. Choose **Finish**. The training of the solution will take a while to complete.

Step 3: Create a Campaign

In this procedure, you deploy the solution by creating a campaign. To create a campaign you must first create and train a solution. For more information, see *Step 2: Create a Solution*.

To create a campaign

1. In the navigation pane, under the dataset group that you created, choose **Dashboard**.
2. Choose **Create new campaign**.
3. In the **Campaign name** field, specify a name for your campaign.

4. In the **Solution** field, select the solution that you created in Step 2: Create a Solution (p. 15).

   Your screen should look similar to the following:

5. Choose **Create campaign**.

   After the campaign is created, a page displays where you can get recommendations (Step 4).

Step 4: Get Recommendations

In this procedure, use the campaign that you created in Step 3 to get recommendations.

To get recommendations

1. Ignore the **Campaign inference** section.

2. Under **Test campaign results**, in the **User ID** and **Item ID** fields, specify values from the *ratings* dataset. For example, specify the user ID **83** and the item ID **750** from the dataset.

3. Choose **Get recommendations**.

   Your screen should look similar to the following. The numbers in the table are the item IDs for recommended items.

Getting Started (AWS CLI)

In this exercise, you use the AWS Command Line Interface (AWS CLI) to explore Amazon Personalize. You create a campaign that returns movie recommendations for a given user ID.

Before you start this exercise, do the following prerequisites.

- The Getting Started Prerequisites (p. 11).
- Set up the AWS CLI, as specified in Setting Up the AWS CLI (p. 8).

> **Note**
> The CLI commands in this exercise were tested on Linux. For information about using the CLI
> commands on Windows, see Specifying Parameter Values for the AWS Command Line Interface
> in the *AWS Command Line Interface User Guide*.

Step 1: Import Training Data

Follow the steps to create a dataset group, add a dataset to the group, and then populate the dataset
using the movie ratings data.

1. Create a dataset group by running the following command. For more information about the API, see
   CreateDatasetGroup (p. 86).

   ```
   aws personalize create-dataset-group --name MovieRatingDatasetGroup
   ```

   The dataset group ARN is displayed, for example:

   ```
   {
     "datasetGroupArn": "arn:aws:personalize:us-west-2:acct-id:dataset-group/
   MovieRatingDatasetGroup"
   }
   ```

   You can use the `describe-dataset-group` command to display the dataset group you created,
   specifying the returned dataset group ARN.

   ```
   aws personalize describe-dataset-group \
   --dataset-group-arn arn:aws:personalize:us-west-2:acct-id:dataset-group/
   MovieRatingDatasetGroup
   ```

   The dataset group and its properties are displayed, for example:

   ```
   {
       "datasetGroup": {
           "name": "MovieRatingDatasetGroup",
           "datasetGroupArn": "arn:aws:personalize:us-west-2:acct-id:dataset-group/
   MovieRatingDatasetGroup",
           "status": "ACTIVE",
           "creationDateTime": 1542392161.262
       }
   }
   ```

   > **Note**
   > You must wait until the dataset group's `status` shows as ACTIVE before creating a dataset
   > in the group. This operation is usually quick.

   If you don't remember the dataset group ARN, you can use the `list-dataset-groups` command
   to display all the dataset groups that you created, along with their ARN's.

   ```
   aws personalize list-dataset-groups
   ```

Note

The `describe-object` and `list-objects` commands are available for most Amazon Personalize objects. These commands won't be shown in the remainder of this exercise but know that they are available.

2. Create a schema file in JSON format by saving the following code to a file named `MovieRatingSchema.json`. The schema matchs the headers you previously added to `ratings.csv`. The schema name is `Interactions`, which matches one of the three types of datasets recognized by Amazon Personalize. For more information, see Datasets and Dataset Groups (p. 4).

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
      {
          "name": "USER_ID",
          "type": "string"
      },
      {
          "name": "ITEM_ID",
          "type": "string"
      },
      {
          "name": "TIMESTAMP",
          "type": "long"
      }
  ],
  "version": "1.0"
}
```

3. Create a schema by running the following command. Specify the file you saved in the previous step. The example shows the file as belonging to the current folder. For more information about the API, see CreateSchema (p. 104).

```
aws personalize create-schema \
  --name MovieRatingSchema \
  --schema file://MovieRatingSchema.json
```

The schema Amazon Resource Name (ARN) is displayed, for example:

```
{
  "schemaArn": "arn:aws:personalize:us-west-2:acct-id:schema/MovieRatingSchema"
}
```

4. Create an empty dataset by running the following command. Provide the dataset group ARN and schema ARN that were returned in the previous steps. The `dataset-type` must match the schema type from the previous step. For more information about the API, see CreateDataset (p. 83).

```
aws personalize create-dataset \
  --dataset-group-arn arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieRatingDatasetGroup \
  --dataset-type Interactions \
  --schema-arn arn:aws:personalize:us-west-2:acct-id:schema/MovieRatingSchema
```

The dataset ARN is displayed, for example:

```
{
```

```
  "datasetArn": "arn:aws:personalize:us-west-2:acct-id:dataset/MovieRatingDatasetGroup/
INTERACTIONS"
}
```

5. Add the training data to the dataset.

    a. Create a dataset import job by running the following command. Provide the dataset ARN and
        Amazon S3 bucket name that were returned in the previous steps. Supply the AWS Identity
        and Access Management (IAM) role ARN you created in Creating an IAM Role (p. 7). For more
        information about the API, see CreateDatasetImportJob (p. 89).

```
aws personalize create-dataset-import-job \
   --job-name MovieRatingImportJob \
   --dataset-arn arn:aws:personalize:us-west-2:acct-id:dataset/
MovieRatingDatasetGroup/INTERACTIONS \
   --data-source dataLocation=s3://bucketname/ratings.csv \
   --role-arn roleArn
```

    The dataset import job ARN is displayed, for example:

```
{
   "datasetImportJobArn": "arn:aws:personalize:us-west-2:acct-id:dataset-import-job/
MovieRatingImportJob"
}
```

    b. Check the create status by using the `describe-dataset-import-job` command. Provide the
        dataset import job ARN that was returned in the previous step. For more information about the
        API, see DescribeDatasetImportJob (p. 139).

```
aws personalize describe-dataset-import-job
   --dataset-import-job-arn arn:aws:personalize:us-west-2:acct-id:dataset-import-
job/MovieRatingImportJob
```

    The properties of the dataset import job and the create status are displayed. Initially, the create
    `status` will show as CREATE PENDING, for example:

```
{
   "datasetImportJob": {
       "jobName": "MovieRatingImportJob",
       "datasetImportJobArn": "arn:aws:personalize:us-west-2:acct-id:dataset-import-
job/MovieRatingImportJob",
       "datasetArn": "arn:aws:personalize:us-west-2:acct-id:dataset/
MovieRatingDatasetGroup/INTERACTIONS",
       "dataSource": {
           "dataLocation": "s3://<bucketname>/ratings.csv"
       },
       "status": "CREATE PENDING",
       "creationDateTime": 1542392161.837
   }
}
```

    c. Check the run status of the import job. Once the create `status` shows as ACTIVE, the
        `describe-dataset-import-job` response includes the `latestDatasetImportJobRun`
        object and its run status, for example:

```
{
   "datasetImportJob": {
       "jobName": "MovieRatingImportJob",
       ...
       "status": "ACTIVE",
```

```
        "creationDateTime": 1542392161.837,
        "latestDatasetImportJobRun": {
            "datasetImportJobRunArn": "arn:aws:personalize:us-west-2:acct-id:dataset-
import-job/MovieRatingImportJob/<run-id>",
            "status": "CREATE IN_PROGRESS"
        }
    }
}
```

The dataset import is complete when the run `status` of the `latestDatasetImportJobRun` shows as ACTIVE. Then you are ready to train the model using the specified dataset.

You can check just the run `status` by using the `describe-dataset-import-job-run` command and providing the dataset import job run ARN. For more information about the API, see DescribeDatasetImportJobRun (p. 141).

> **Note**
> Importing takes time. You must wait until the dataset import is complete (`latestDatasetImportJobRun status` shows as ACTIVE) before training the model using the dataset.

Step 2: Create a Solution (Train the Model)

To create a solution you need to train a model. You need a recipe (algorithm, hyperparameters, and feature transformation) and training data. Amazon Personalize provides a set of predefined recipes that you can use for model training. For more information, see Using Predefined Recipes (p. 43). For this exercise, you use AutoML, which allows Amazon Personalize to pick the best recipe, and the dataset you created in the preceding step.

1. Train a model by running the following command. For more information about the API, see CreateSolution (p. 106).

```
aws personalize create-solution \
  --name MovieSolution \
  --perform-auto-ml \
  --dataset-group-arn arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieRatingDatasetGroup \
  --min-provisioned-tps 10
```

The solution ARN is displayed, for example:

```
{
  "solutionArn": "arn:aws:personalize:us-west-2:acct-id:solution/MovieSolution"
}
```

2. Check the create status using the `describe-solution` command. Provide the solution ARN that was returned in the previous step. For more information about the API, see DescribeSolution (p. 153).

```
aws personalize describe-solution
  --solution-arn arn:aws:personalize:us-west-2:acct-id:solution/MovieSolution
```

The properties of the solution and the create `status` are displayed. Initially, the create `status` will show as CREATE PENDING, for example:

```
{
  "solution": {
```

```
        "name": "MovieSolution",
        "solutionArn": "arn:aws:personalize:us-west-2:acct-id:solution/MovieSolution",
        "performAutoML": true,
        "datasetGroupArn": "arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieRatingDatasetGroup",
        "solutionConfig": {
            "retrainingMode": "MANUAL",
            "autoMLConfig": {
                "metricName": "precision_at_25",
                "recipeList": [
                    "arn:aws:personalize:::recipe/aws-hrnn",
                    "arn:aws:personalize:::recipe/aws-deepfm",
                    "arn:aws:personalize:::recipe/aws-ffnn"
                ]
            }
        },
        "status": "CREATE PENDING",
        "creationDateTime": 1543864685.016
    }
}
```

Note the `metricName` and `recipeList`. When Amazon Personalize performs AutoML, it chooses the recipe from the list that optimizes that metric.

3. Check the training status of the solution. Once the create `status` shows as CREATE IN_PROGRESS, the `describe-solution` response includes the `latestSolutionVersion` object and its training `status`, for example:

```
{
  "solution": {
      "name": "MovieSolution",
      ...
      "status": "CREATE IN_PROGRESS",
      "creationDateTime": 1543864685.016,
      "latestSolutionVersion": {
          "solutionVersionArn": "arn:aws:personalize:us-west-2:acct-id:solution/
MovieSolution/<version-id>",
          "status": "CREATE IN_PROGRESS"
      },
  }
}
```

You can check just the training status by using the `describe-solution-version` command and providing the solution version ARN.

4. Check the AutoML results. When the latest solution version training `status` shows as ACTIVE, the training is complete. The `describe-solution` or `describe-solution-version` response now includes the `autoMLResult` object, which shows the best recipe as determined by Amazon Personalize, for example:

```
{
  "solution": {
      "name": "MovieSolution",
      ...
      "performAutoML": true,
      "solutionConfig": {
          "autoMLConfig": {
              "metricName": "precision_at_25",
              "recipeList": [
                  "arn:aws:personalize:::recipe/aws-hrnn",
                  "arn:aws:personalize:::recipe/aws-deepfm",
                  "arn:aws:personalize:::recipe/aws-ffnn"
              ]
```

```
            }
        },
        "autoMLResult": {
            "bestRecipeArn": "arn:aws:personalize:::recipe/aws-hrnn"
        },
        "status": "ACTIVE",
        "latestSolutionVersion": {
            "solutionVersionArn": "arn:aws:personalize:us-west-2:202281833397:solution/
www-solution/37b225c4",
            "status": "ACTIVE"
        }
    }
}
```

Now you can check the solution metrics and create a campaign using the solution.

> **Note**
> Training takes time. You must wait until training is complete (`latestSolutionVersion status` shows as ACTIVE) before using this version of the solution in a campaign. For quicker training, instead of using `perform-auto-ml`, select a specific recipe using the `recipe-arn` parameter.

5. You can validate the performance of the solution by reviewing its metrics. Get the metrics for the solution by running the following command. Provide the solution ARN that was returned previously. Amazon Personalize defaults to the latest version of the solution, the one just trained, when the solution ARN is specified but the solution version ARN isn't specified. For more information about the API, see GetMetrics (p. 159).

```
aws personalize get-metrics
   --solution-arn arn:aws:personalize:us-west-2:acct-id:solution/MovieSolution
```

A sample response is shown:

```
{
  "metrics": {
      "arn:aws:personalize:us-west-2:acct-id:model/awspersonalizehrnnmodel-7923fda9": {
          "_num_evaluation_users": 91.0,
          "_num_unique_items": 1077.0,
          "_user_history_length_10_pct_quantile": 13.0,
          "_user_history_length_50_pct_quantile": 41.0,
          "_user_history_length_90_pct_quantile": 129.0,
          "_user_history_length_mean": 57.84615384615385,
          "coverage": 0.3649025069637883,
          "mean_reciprocal_rank": 0.03389927066397654,
          "normalized_discounted_cumulative_gain_at_10": 0.05161045001320517,
          "normalized_discounted_cumulative_gain_at_25": 0.07648398967771727,
          "normalized_discounted_cumulative_gain_at_5": 0.02265501441367968,
          "precision_at_10": 0.012087912087912088,
          "precision_at_25": 0.008791208791208791,
          "precision_at_5": 0.006593406593406594
      }
  }
}
```

Step 3: Create a Campaign (Deploy the Solution)

Before you can get recommendations, you must deploy the solution. Deploying a solution is also known as creating a campaign. Once you've created your campaign, your client application can get recommendations using the GetRecommendations (p. 208) API.

1. Create a campaign by running the following command. Provide the solution ARN that was returned in the previous step. Amazon Personalize defaults to the latest version of the solution, the one just trained, when the solution ARN is specified but the solution version ARN isn't specified. For more information about the API, see CreateCampaign (p. 80).

```
aws personalize create-campaign \
  --name MovieRecommendationCampaign \
  --solution-arn arn:aws:personalize:us-west-2:acct-id:solution/MovieSolution \
  --update-mode AUTO
```

A sample response is shown:

```
{
  "campaignArn": "arn:aws:personalize:us-west-2:acct-id:campaign/
MovieRecommendationCampaign"
}
```

2. Check the deployment status by running the following command. Provide the campaign ARN that was returned in the previous step. For more information about the API, see DescribeCampaign (p. 133).

```
aws personalize describe-campaign \
  --campaign-arn arn:aws:personalize:us-west-2:acct-id:campaign/
MovieRecommendationCampaign
```

A sample response is shown:

```
{
  "campaign": {
      "name": "MovieRecommendationCampaign",
      "updateMode": "true",
      "campaignArn": "arn:aws:personalize:us-west-2:acct-id:campaign/
MovieRecommendationCampaign",
      "solutionArn": "arn:aws:personalize:us-west-2:acct-id:solution/MovieSolution",
      "status": "ACTIVE"
  }
}
```

> **Note**
> You must wait until the `status` shows as ACTIVE before getting recommendations from the campaign.

Step 4: Get Recommendations

Get recommendations by running the `get-recommendations` command. Provide the campaign ARN that was returned in the previous step. In the request, you specify a user ID from the movie ratings dataset. For more information about the API, see GetRecommendations (p. 208).

> **Note**
> Not all recipes support the `GetRecommendations` API. For more information, see Using Predefined Recipes (p. 43).
> The AWS CLI command you call in this step, `personalize-runtime`, is different than in previous steps.

```
aws personalize-runtime get-recommendations \
  --campaign-arn arn:aws:personalize:us-west-2:acct-id:campaign/MovieRecommendationCampaign
 \
```

```
    --user-id 123
```

In response, the campaign returns a list of item recommendations (movie IDs) the user might like. The list is sorted in descending order of relevance for the user.

```
{
  "itemList": [
      {
          "itemId": "99"
      },
      {
          "itemId": "281"
      },
      {
          "itemId": "275"
      },
      ...
}
```

Getting Started (AWS SDK for Python)

This topic provides getting started information about programming Amazon Personalize with the AWS SDK for Python (Boto 3).

Prerequisites

The following are prerequisite steps for using the python examples in this guide. If you are experimenting with Amazon Personalize, you can use the same source data as that used in the Getting Started (Console) (p. 12) and Getting Started (AWS CLI) (p. 18) exercises. If you are using your own source data, you don't need to do the prerequisite step Create the Training Data (p. 11). For information about preparing your own source data, see Preparing and Importing Data (p. 28).

- Do the Getting Started Prerequisites (p. 11).
- Set up the AWS SDK for Python (Boto 3), as specified in Setting Up the AWS SDKs (p. 9).

Confirming your Python Environment

After you have completed the prerequisites, run the following python example to confirm that your environment is configured correctly. If successful, a list of the available recipes is displayed and you can now run the other python examples that are in this guide.

```
import boto3

if __name__ == "__main__":

    personalize = boto3.client('personalize')

    response=personalize.list_recipes()

    for recipe in response['recipes']:
        print (recipe)
```

Jupyter (iPython) Notebook

A Jupyter (iPython) notebook is also available for you to explore the Amazon Personalize APIs. With one exception, the Jupyter notebook has the same prerequisites as the python examples in this guide. The notebook uses different source data and you don't need to do the prerequisite step Create the Training Data (p. 11).

To get the Jupyter notebook, clone or download the personalize_sample_notebook.ipynb notebook from the Amazon Personalize Samples repository.

Preparing and Importing Data

Amazon Personalize uses data you provide to train a model. You can provide data from a source file, or you can collect data from live events such as user activity on a website.

If you are providing a source file for import, your data must be in comma-separated values (CSV) format and conform to a schema that Amazon Personalize can understand. You upload your CSV file into an Amazon S3 bucket from which Amazon Personalize imports your data into a dataset. Datasets belong to a dataset group that contains other related datasets. You can upload your data using the AWS Management Console or by using the AWS SDK.

This section provides information about formatting and importing your historical data into Amazon Personalize. For information about importing live events, see Recording Events (p. 35).

Topics

Datasets and Schemas

Amazon Personalize recognizes three types of historical datasets. Each dataset type has an associated schema with a *name* key whose value matches the dataset type.

- **Users** – This dataset is intended to provide metadata about your users. This includes information such as age, gender, and loyalty membership, among others, which can be important signals in personalization systems.
- **Items** – This dataset is intended to provide metadata about your items. This includes information such as price, SKU type, and availability, among others.
- **Interactions** – This dataset is intended to provide historical interaction data between users and items.

The Users and Items dataset types are known as metadata types and are only used by certain recipes. For more information, see Using Predefined Recipes (p. 43). For metadata datasets, all strings must be marked as `categorical` in the schema, as shown in the following examples.

> **Note**
> A dataset group can contain only one of each kind of dataset.

Each dataset has a set of required fields and reserved keywords, as shown in the following table.

Dataset Type	Required Fields	Reserved Keywords	
Users	USER_ID		
Items	ITEM_ID		
Interactions	USER_ID	EVENT_TYPE	
	ITEM_ID	EVENT_VALUE	
	TIMESTAMP		

Before you add a dataset to Amazon Personalize, you need to define a schema for the dataset. Schemas in Amazon Personalize are defined in the Avro format. For more information, see Apache Avro.

The following example shows an `Interactions` schema. The `EVENT_TYPE` and `EVENT_VALUE` fields are optional, and are reserved keywords recognized by the section called "Amazon Personalize Events" (p. 205).

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
      {
          "name": "USER_ID",
          "type": "string"
      },
      {
          "name": "ITEM_ID",
          "type": "string"
      },
      {
          "name": "EVENT_TYPE",
          "type": "string"
      },
      {
          "name": "EVENT_VALUE",
          "type": "string"
      },
      {
          "name": "TIMESTAMP",
          "type": "long"
      }
  ],
  "version": "1.0"
}
```

The following example shows a `Users` schema in Avro format. Only the `USER_ID` field is required.

```
{
  "type": "record",
  "name": "Users",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
      {
          "name": "USER_ID",
          "type": "string"
      },
      {
          "name": "AGE",
          "type": "int"
      },
      {
          "name": "GENDER",
          "type": "string",
          "categorical": true
      }
  ],
  "version": "1.0"
}
```

The following example shows an `Items` schema. Only the `ITEM_ID` field is required.

```
{
```

```
  "type": "record",
  "name": "Items",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
      {
          "name": "ITEM_ID",
          "type": "string"
      },
      {
          "name": "GENRE",
          "type": "string",
          "categorical": true
      }
  ],
  "version": "1.0"
}
```

If you are using the AWS console, you create a new schema when you create a dataset for your input data. You can also choose an existing schema.

If you are using the AWS CLI, see Step 1: Import Training Data (p. 19) for an example.

To create a schema using the AWS Python SDK

1. Define the Avro format schema that you want to use.
2. Save the schema in a JSON file (optional).
3. Create the schema using the following code.

```
import boto3
import io

    personalize = boto3.client('personalize')

    with open("schema.json") as f:
        createSchemaResponse = personalize.create_schema(
            name = "Schema name",
            schema = f.read()
        )

    schemaArn = createSchemaResponse['schemaArn']

    print("Schema ARN:" + schemaArn )
```

4. Amazon Personalize returns the ARN of the new schema. Store it for later use.

Amazon Personalize provides operations for managing schemas. For example, you can use the ListSchemas (p. 181) API to get a list of the schemas that you have created.

After you have created a schema, you use it with datasets that match the schema. For more information, see Formatting your Input Data (p. 30).

Formatting your Input Data

The files you use to import data into Amazon Personalize must map to the schema you are using.

Amazon Personalize imports data from files that are in CSV format. In CSV files, individual values are separated by commas. Amazon Personalize requires the first row of your CSV file to contain column headers. The column headers in your CSV file need to map to the schema used by the dataset.

For example, the following CSV data sample maps to the Interactions schema created previously in Datasets and Schemas (p. 28). This data might represent historical user activity on a website that sells movie tickets. The data can be used to train a model that gives a user movie recommendations based on the activity of other users.

```
USER_ID,ITEM_ID,EVENT_TYPE,EVENT_VALUE,TIMESTAMP
196,242,click,15,881250949
186,302,click,13,891717742
22,377,click,10,878887116
244,51,click,20,880606923
166,346,click,10,886397596
298,474,click,40,884182806
115,265,click,20,881171488
253,465,click,50,891628467
305,451,click,30,886324817
```

The associated `Interactions` schema is repeated below.

```
{
  "type": "record",
  "name": "Interactions",
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    { "name": "EVENT_TYPE",
      "type": "string"
    },
    {
      "name": "EVENT_VALUE",
      "type": "string"
    },
    {
      "name": "TIMESTAMP",
      "type": "long"
    }
  ],
  "version": "1.0"
}
```

The USER_ID, ITEM_ID, and TIMESTAMP fields are required by Amazon Personalize. USER_ID is the identifier for a user of your application. ITEM_ID is the identifier for a movie. EVENT_TYPE and EVENT_VALUE map to user activities. In the sample data, `click` might represent a movie purchase event and `15` might represent the purchase price of the movie. TIMESTAMP is the Unix time that the movie purchase took place.

After you have formatted your data, you can upload your data to an Amazon S3 bucket for import into Amazon Personalize. For more information, see Uploading to an Amazon S3 Bucket (p. 31).

Uploading to an Amazon S3 Bucket

Once you have created a CSV file with your data, you upload the file to an Amazon S3 bucket. This is the location that Amazon Personalize imports your data from. You have to give Amazon Personalize

permission to access the Amazon S3 bucket. If you are using the AWS console, Amazon Personalize can create the access policy when you create the dataset. If you are using the CLI or the AWS SDK, attach the following policy to the bucket. For more information, see How Do I Add an S3 Bucket Policy? in the Amazon Simple Storage Service Console User Guide.

For an example, see Create the Training Data (p. 11).

```
{
    "Version": "2012-10-17",
    "Id": "PersonalizeS3BucketAccessPolicy",
    "Statement": [
        {
            "Sid": "PersonalizeS3BucketAccessPolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "personalize.amazonaws.com"
            },
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name>",
                "arn:aws:s3:::bucket-name>/*"
            ]
        }
    ]
}
```

After you have uploaded your data to an Amazon S3 bucket, you can import your data into Amazon Personalize. For more information, see Importing your Data (p. 32).

Importing your Data

You import your training data into Amazon Personalize by first creating an empty dataset group and then an empty dataset in that dataset group. Next, you create an import job that populates the dataset with data from your Amazon S3 bucket. For more information about datasets, see Datasets and Schemas (p. 28).

The following steps show you how to create and populate a dataset. For the equivalent steps using the AWS CLI, see Step 1: Import Training Data (p. 19).

To Import your Data using the SDK

1. Create a dataset group using the CreateDatasetGroup (p. 86) API.

    ```
    import boto3
        personalize = boto3.client('personalize')

        response=personalize.create_dataset_group(name='Dataset group name')

        arn=response['datasetGroupArn']

        description=personalize.describe_dataset_group(datasetGroupArn=arn)
    ['datasetGroup']
        print('Name: ' + description['name'])
        print('ARN: ' + description['datasetGroupArn'])
        print('Status: ' + description['status'])
    ```

The `datasetGroupArn` is returned in the response. The response from the DescribeDatasetGroup (p. 137) API includes the status of the operation.

> **Note**
> You must wait until the status shows as ACTIVE before preceding to the next step.

2. Create a dataset using the CreateDataset (p. 83) API. Specify the `datasetGroupArn` returned in the previous step. Use the `schemaArn` created earlier in Datasets and Schemas (p. 28).

```python
import boto3

    personalize = boto3.client('personalize')

    response = personalize.create_dataset(
        schemaArn = "schema arn",
        datasetGroupArn = "dataset group arn",
        datasetType= "Interactions")

    print ("Dataset Arn: " + response['datasetArn'])
```

3. Add data to your dataset by creating and running a dataset import job using the CreateDatasetImportJob (p. 89) API. Specify the `datasetGroupArn` returned previously. Set the `dataLocation` to the Amazon S3 bucket where your training data is stored. For the `roleArn`, see Creating an IAM Role (p. 7).

```python
import boto3

    personalize = boto3.client('personalize')

    response=personalize.create_dataset_import_job(
        jobName="Import job name",
        datasetArn="Dataset arn",
        dataSource={'dataLocation':'s3://bucket/file.csv'},
        roleArn="Role")

    print ("Dataset Import Job arn: " + response['datasetImportJobArn'])

    arn=response['datasetImportJobArn']

    description=personalize.describe_dataset_import_job(datasetImportJobArn=arn)
['datasetImportJob']
    print('Name: ' + description['jobName'])
    print('ARN: ' + description['datasetImportJobArn'])
    print('Status: ' + description['status'])
    print('Import status ' + description['latestDatasetImportJobRun']['status'])
```

The `CreateDatasetImportJob` operation creates an import job that is configured to import data from an Amazon S3 bucket (`dataLocation`). The Amazon S3 bucket location is specified as an Amazon Resource Name (ARN).

The `roleArn` parameter specifies the AWS Identity and Access Management role that gives Amazon Personalize permissions to access your Amazon S3 Bucket.

`CreateDatasetImportJob` triggers a first-time import and returns the dataset import job ARN. You call the DescribeDatasetImportJob (p. 139) API to get the current status of the import job.

> **Note**
> The import isn't complete until the status of `latestDatasetImportJobRun` is `ACTIVE`.

You can run the import job at a defined frequency by specifying the `importJobSchedule` key of the `datasetImportJobConfig` parameter (not shown). For more information, see DatasetImportJobConfig (p. 238).

> **Note**
> Imports in Amazon Personalize are a full refresh of the data. Incremental updates cannot be added. If you need to update your data, pass the complete updated file.

Amazon Personalize provides operations for managing datasets, dataset groups, and dataset import jobs. For example, you can use ListDatasets (p. 172) to list the datasets in a dataset group and use DeleteDataset (p. 115) to delete a dataset.

After you have imported your data into a dataset, you can create a solution by training a model. For more information, see Creating a Solution (p. 41).

Recording Events

Amazon Personalize can make recommendations based purely on historical data as demonstrated in the Getting Started (p. 11) guides. Amazon Personalize can also make recommendations purely on real time clickstream data, or a combination of both, using the Amazon Personalize event ingestion SDK.

> **Note**
> A minimum of 1000 records of combined interaction data is required to train a model.

The event ingestion SDK includes a JavaScript library for recording events from web client applications. The SDK also includes a library for recording events in server code.

To record events, you need the following:

- A dataset group, which can be empty.
- An event tracker with the appropriate AWS Identity and Access Management (IAM) permissions. The IAM role allows Amazon CloudWatch to calculate the clickthrough rate for the events.

Topics

Creating a Dataset Group

If you went through the Getting Started (p. 11) guide, you can use the same dataset group that you created, or you can create a new dataset group as shown below. The dataset group can be empty or the group can contain any of the user defined datasets. For more information, see Datasets and Dataset Groups (p. 4).

Python

```
import boto3

personalize = boto3.client('personalize')
response = personalize.create_dataset_group(name='MovieClickGroup')
print(response['datasetGroupArn'])
```

AWS CLI

```
aws personalize create-dataset-group --name MovieClickGroup
```

The dataset group Amazon Resource Name (ARN) is displayed, for example:

```
{
    "datasetGroupArn": "arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieClickGroup"
}
```

Getting a Tracking ID

A tracking ID associates an event with a dataset group and authorizes you to send data to Amazon Personalize. You generate a tracking ID by calling the CreateEventTracker (p. 93) API. You supply the dataset group ARN and the ARN of the IAM role that you created in Creating an IAM Role (p. 7).

Python

```
import boto3

personalize = boto3.client('personalize')
response = personalize.create_event_tracker(
    name='MovieClickTracker',
    datasetGroupArn='arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieClickGroup',
    roleArn='role-arn'
)
print(response['eventTrackerArn'])
print(response['trackingId'])
```

AWS CLI

```
aws personalize create-event-tracker \
    --name MovieClickTracker \
    --dataset-group-arn arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieClickGroup \
    --role-arn role-arn
```

The event tracker ARN and tracking ID are displayed, for example:

```
{
    "eventTrackerArn": "arn:aws:personalize:us-west-2:acct-id:event-tracker/
MovieClickTracker",
    "trackingId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
}
```

Event-Interactions Dataset

When Amazon Personalize creates an event tracker, it also creates a new event-interactions dataset in the dataset group associated with the event tracker. The event-interactions dataset stores the event data from the PutEvents (p. 206) call. The contents of the dataset are not available to the user.

> **Note**
> A dataset group can contain only one user-item iterations dataset and one event-interactions dataset. You will get an error if you try to call `CreateEventTracker` using the same dataset group as an existing event tracker even if you use a different event tracker name.

To view the properties of the new dataset, call the ListDatasets (p. 172) API, supplying the
dataset group ARN. Use the dataset ARN for the EVENT_INTERACTIONS dataset to call the
DescribeDataset (p. 135) API. The following is an example response:

```
{
    "dataset": {
        "datasetArn": "arn:aws:personalize:us-west-2:acct-id:dataset/dataset-group-name/
EVENT_INTERACTIONS",
        "datasetGroupArn": "arn:aws:personalize:us-west-2:acct-id:dataset-group/
MovieClickGroup",
        "datasetType": "EVENT_INTERACTIONS",
        "schemaArn": "arn:aws:personalize:us-west-2:acct-id:schema/event-interactions-
schema",
        "status": "ACTIVE",
        "creationDateTime": 1545694802.016
    }
}
```

Event Interactions Schema

To view the schema corresponding to the event-interactions dataset, call the DescribeSchema (p. 151)
API, supplying the schema ARN from the previous listing. An example response follows. The
`event_type` field corresponds to the `eventName` parameter of the `Record` API. The `item_id` and
`event_value` fields correspond to the `id` and `value` keys of the `properties` parameter of the `Record`
API.

```
{
    "schema": {
        "name": "event-interactions-schema",
        "schemaArn": "arn:aws:personalize:us-west-2:acct-id:schema/event-interactions-
schema",
        "schema": {
          "type": "record",
          "name": "Interactions",
          "namespace": "com.amazonaws.personalize.schema",
          "fields": [
            {
              "name": "user_id",
              "type": "string"
            },
            {
              "name": "session_id",
              "type": "string"
            },
            {
              "name": "timestamp",
              "type": "long"
            }
            {
              "name": "event_type",
              "type": "string"
            },
            {
              "name": "item_id",
              "type": "string"
            },
            {
              "name": "event_value",
              "type": "string"
            },
```

```
            ],
            "version": "1.0"
        }"
    }
}
```

Calling the Record API

The following shows an example of a `Record` call. You supply the tracking ID created in the previous step. The user ID and session ID are custom to your application.

The event list is an array of the section called "Event" (p. 287) types. The event ID is custom to your application. The `sentAt` and `eventName` keys correspond to the `timestamp` and `event_type` fields of the event-interactions schema (described below).

The `properties` key is a string map (key-value pairs) of event-specific data. The `id` and `value` keys correspond to the `item_id` and `event_value` fields of the event-interactions schema.

Python

```
import boto3

personalize_events = boto3.client('personalize-events')
personalize_events.record(
    trackingId = 'tracking-id',
    userId = 'user-id',
    sessionId = 'session-id',
    eventList = [
      {
          "eventId": "event1",
          "sentAt": 1545694248,
          "eventName": "rating",
          "properties": """{\"id\": \"101\", \"value\": \"4\"}"""
      },
      {
          "eventId": "event2",
          "sentAt": 1545694251,
          "eventName": "rating",
          "properties": """{\"id\": \"311\", \"value\": \"2\"}"""
      }
    ]
)
```

AWS CLI

```
aws personalize-events record \
    --tracking-id tracking-id \
    --user-id user-id \
    --session-id session-id \
    --event-list [event1, event2, ...]
```

Event Types

Amazon Personalize recognizes any string value (subject to length constraints) as a valid event type. There are two event types that Amazon Personalize treats as special when using the Javascript SDK:

- ListView

  For the `ListView` event type, you use the `items` key with an array of objects as the value (each object having one or more key-value pairs). For example,

  ```
  {items: [{"id": "item1", "value":"value1"}, {"id": "item2",
  "value":"value2"}, {...}]}
  ```
- MediaAutoTrack

Creating an Events Solution

When training a model that uses event data, two parameters of the the section called "CreateSolution" (p. 106) API are relevant. The `eventType` parameter must be specified when multiple event types are recorded. The `eventType` indicates which type of event Amazon Personalize uses for model training.

The `eventValueThreshold` parameter of the `SolutionConfig` object creates an event filter. When this parameter is specified, only events with a value greater than or equal to the threshold are used for training the model. You must specify `eventType` when using `eventValueThreshold`.

Send Real Time Event Data using AWS Amplify

In the following procedure, you use the AWS Amplify JavaScript Library to record media event data.

1. Generate a Amazon Cognito identity pool ID.

    Amazon Personalize authenticates the user identities for the event data using the Amazon Cognito identity pool. To track authenticated users you need to create a Amazon Cognito user pool. To track anonymous users, you need to enable access to unauthenticated identities. Follow AWS Amplify Authentication for usage guidelines.

    An example of an identity pool ID is: `us-east-1:1699ebc0-7900-4099-b910-2df94f52a030`.
2. Include personalize-events.min.js (right-click to download) in your web application (**PREVIEW RELEASE**).
3. Add Amazon Personalize as a provider in Amplify.

    Add the following code to your application. Replace `identityPoolId` and `trackingId` with the identity pool ID and tracking ID generated previously.

    ```
    // Configure the plugin after adding it to the Analytics module
    const aws_config =  {
        "aws_project_region": "us-west-2",
        "aws_cognito_identity_pool_id": "identityPoolId",
        "aws_cognito_region": "us-west-2",
    };

    Analytics.addPluggable(new AmazonPersonalizeProvider("trackingId"));
    Auth.configure(aws_config);

    // Configure the plugin after adding it to the Analytics module
    Analytics.configure({
        AmazonPersonalize: {

            // OPTIONAL -  Amazon Personalize service region
            region: 'us-east-1',
    ```

```
        // OPTIONAL - The number of events to be deleted from the buffer when flushed.
        flushSize: 10,

        // OPTIONAL - The interval in milliseconds to perform a buffer check and flush
 if necessary.
        flushInterval: 5000, // 5s
    }
});
```

4. Record event data as shown in the following code.

Note the use of the `eventName` and `properties` keys. These keys map to the `EVENT_TYPE` and `EVENT_VALUE` fields in an `Interactions` dataset schema.

```
Analytics.record({
        eventName: "ListView",
        properties: {"items": [{"id": "item1"}, {"id": "item2"}], "listViewId":
 "list1"}
    },
    "AmazonPersonalize");

===Click event:=====
Analytics.record({
        eventName: "Click",
        properties: {"id": "itemIdClickedOn", "value": "itemNameClickedOn"}
    },
    "AmazonPersonalize");


=== Page event ====
Analytics.record({
        eventName: "Page",
        properties: {"id": "page id", "value": "page name"}
    },
    "AmazonPersonalize");

=== MediaAutoTrack event ====
Analytics.record({
    eventName: "MediaAutoTrack",
    properties: {
        "id": "MEDIA DOM ELEMENT ID"
    }
}, "AmazonPersonalize");
```

Creating a Solution

A solution is the term Amazon Personalize uses for a trained machine learning model that makes recommendations to customers. Creating a solution entails optimizing the model to deliver the best results for a specific business need. Amazon Personalize uses "recipes" to create these personalized solutions.

A recipe in Amazon Personalize is made up of an algorithm with hyperparameters, and a feature transformation. An algorithm is a mathematical expression that when trained becomes the model. The algorithm contains parameters whose unknown values are determined by training on input data. Hyperparameters are external to the algorithm and specify details of the training process, such as the number of training passes to run over the complete dataset or the number of hidden layers to use. For more information on algorithms, see Creating an Algorithm (p. 55). For more information on hyperparameters, see Hyperparameter Optimization (p. 56).

Feature transformation is the process of modifying raw input data into a form more suitable for model training. For example, a categorical field of strings could be modified by assigning a unique number to each category. The modified data would then be used as input for the model training. For more information, see Feature Transformation (p. 56).

Amazon Personalize provides a number of predefined recipes that allow you to make recommendations with no knowledge of machine learning. The predefined recipes are also useful for quick experimentation. For more information, see Using Predefined Recipes (p. 43). If your business has more specific needs, Amazon Personalize allows you to "Bring Your Own Recipe" (BYOR). For more information, see Bring Your Own Recipe (p. 52).

Creating a Solution

A solution is created by calling the CreateSolution (p. 106) API. A condensed, reorganized version of the `CreateSolution` request is shown. The various options and configuration objects are discussed in the following sections.

```
{
    "name": "string",
    "performAutoML": boolean,
    "recipeArn": "string",
    "solutionConfig": {
        "autoMLConfig": {
            "metricName": "string",
            "recipeList": [ "string" ]
        },
        "algorithmHyperParameters": { },
        "featureTransformationParameters": { },
        "hpoConfig": { }
    },
    "performHPO": boolean
}
```

How a recipe is chosen is shown in the following table. Either `performAutoML` or `recipeArn` must be specified but not both.

performAutoML	recipeArn	solutionConfig	Result
true	omit	omitted	Amazon Personalize chooses the recipe

performAutoML	recipeArn	solutionConfig	Result
true	omit	`autoMLConfig: metricName` and `recipeList` specified (other `solutionConfig` properties ignored)	Amazon Personalize chooses a recipe from the list to optimize the metric
omit	specified	omitted	You specify the recipe
omit	specified	specified	You specify the recipe and override the training properties

Leave it to Amazon Personalize

If you have no knowledge of machine learning, or just want to get started as quickly as possible, you can have Amazon Personalize analyze your data and decide on the best predefined recipe and options to use. In this case, you call `CreateSolution`, specify 'true' for the `performAutoML` parameter, and omit the `recipeArn` and `solutionConfig` parameters. Amazon Personalize does the rest.

You can also specify the list of recipes that Amazon Personalize examines to determine the optimal recipe, based on a metric you specify. This list can include both predefined and BYOR recipes. In this case, you call `CreateSolution`, specify 'true' for the `performAutoML` parameter, omit the `recipeArn` parameter, and include the `solutionConfig` parameter, specifying the `metricName` and `recipeList` as part of the `autoMLConfig` object.

Instead of having Amazon Personalize choosing the recipe, you can do so manually. In this case, you call `CreateSolution`, specify 'false' for `performAutoML`, and supply the `recipeArn` parameter. For the list of predefined recipes, see Using Predefined Recipes (p. 43).

Bring Your Own Recipe

If you want to use your own algorithm, Amazon Personalize offers its "Bring Your Own Recipe" feature. Your algorithm will be part of the same data pipelines, training APIs, and recommendations as Amazon Personalize's predefined algorithms. You can run metrics and benchmark your algorithms against Amazon Personalize's algorithms in order to choose the most appropriate model for your needs. For more information see Bring Your Own Recipe (p. 52).

Customize the Training

To customize the training, supply the `solutionConfig` parameter. The `SolutionConfig` object allows you to override the default solution and recipe parameters. For more information see Overriding Default Recipe Parameters (p. 55).

Evaluate the Training

After training a model, you evaluate it to determine whether its performance and accuracy allow you to achieve your business goals. You might generate multiple models using different methods and evaluate each. For example, you could apply different business rules for each model, and then apply various measures to determine each model's suitability. For more information, see Evaluating Solutions (p. 57).

To create a solution (SDK)

1. Create and import a dataset into a dataset group. For more information, see Preparing and Importing Data (p. 28).

2. Create the solution using the following code.

```
import boto3

if __name__ == "__main__":

    personalize = boto3.client('personalize', region_name='us-west-2')

    response=personalize.create_solution(
        datasetGroupArn="Dataset group arn",
        name="Solution name",
        performAutoML=True,
        minProvisionedTPS=10)

    arn=response['solutionArn']

    description=personalize.describe_solution(solutionArn=arn)['solution']
    print('Name: ' + description['name'])
    print('ARN: ' + description['solutionArn'])
    print('Status: ' + description['status'])
```

Training a model takes time to complete. To check the current status, call DescribeSolution (p. 153). Training is complete when the value of `status` is `ACTIVE`.

Now that you have a solution, you can create a campaign by deploying the solution. For more information, see Creating a Campaign (p. 59).

Using Predefined Recipes

Amazon Personalize provides predefined recipes to train a model, which enables you to create a personalization system without needing machine learning experience.

These recipes use predefined attributes of your data, predefined feature transformations, predefined algorithms, and initial parameters for the algorithms. You can override many of these parameters when creating a solution.

Amazon Personalize can automatically choose the most appropriate recipe based on its analysis of the input data. Alternatively, you can choose a specific recipe based on your experience. Each recipe has its own use case, as described below, and you should select the recipe that best fits your needs.

You can see a list of available recipes in the Amazon Personalize console or list the recipes by calling the the section called "ListRecipes" (p. 179) API. You can get information about a specific recipe by calling the the section called "DescribeRecipe" (p. 149) API.

Amazon Personalize provides three types of recipes. Besides behavioral differences, each type has different requirements for getting recommendations, as shown in the following table.

Recipe type	API	userId	itemId	inputList
SEARCH_PERSONALIZATION	PersonalizeRanking (p. 211)	required	NA	list of itemId's
RELATED_ITEMS	GetRecommendations (p. 208)	not used	required	NA
USER_PERSONALIZATION	GetRecommendations (p. 204)	required	optional	NA

Predefined Recipes

Amazon Personalize provides the following predefined recipes listed by recipe type.

SEARCH_PERSONALIZATION recipes - personalizes search

Personalized-Ranking (p. 49)

Use this recipe when you're personalizing the search results for your users, such as, personalized reranking of search results or curated lists.

RELATED_ITEMS recipes - predicts items similar to a given item

SIMS (p. 49)

Item-to-item similarities (SIMS) generates items similar to a given item based on co-occurrence of the item in user history in the user-item interaction dataset. In the absence of sufficient user behavior data for an item, the algorithm returns popular items as recommendations.

Use for improving item discoverability and in detail pages. Fast performance.

USER_PERSONALIZATION recipes - predicts items a user will interact with

DeepFM (p. 45)^

Extends a factorization machine using a deep autoencoder to achieve higher accuracy via nonlinear modeling of user item interactions.

Use to provide recommendations for a short horizon into the future. Recommended for fast training and inference, with good general performance.

FFNN (p. 46)^

A feed-forward neural network that utilizes a built-in K-nearest neighbors search to deliver recommendations based on past interactions in the user-item interactions dataset.

Recommended for fast training and inference, and scalability to large datasets.

HRNN (p. 47)^

A hierarchical recurrent neural network, which can model the temporal order of user-item interactions.

Recommended when user behavior is changing with time (the evolving intent problem).

HRNN-coldstart*

Similar to HRNN-metadata with personalized exploration of new items.

Recommended when frequently adding new items to a catalog and you want the items to show up in recommendations immediately.

HRNN-metadata*

HRNN with additional features derived from contextual metadata (user-item interactions), along with user and item metadata (user and item datasets).

Performs better than non-metadata models when high quality metadata is available. Can involve longer training times.

Popularity-Count (p. 48)

Calculates popularity of items based on a count of events against that item in the user-item interactions dataset.

Use as a baseline to compare other personalization recipes.

^ Amazon Personalize examines these recipes when performing AutoML.

* Metadata models: These recipes train on both user-interaction data and metadata (User and Item datasets).

DeepFM Recipe

Matrix factorization and related methods are the standard method used by many contemporary personalization systems, including those of Amazon.

The core idea is to represent user-item interactions as a two-dimensional matrix where each row corresponds to an item, and each column corresponds to a user. If the user has rated or interacted with the item, then the entry in the matrix is the rating. If the user has not rated the item, then the entry is missing. We denote a missing entry with the question mark symbol *?*. Often the user ratings are binary, that is, 1 if a user has consumed the item, and *?* otherwise. The goal is to predict the true value for each *?* value. Matrix factorization learns a numerical embedding vector for each customer (row) and for each item (column). The vectors are trained so that when appropriately combined, the result equals the known rating, or equals 1, as closely as possible. Then, the prediction for each *?* is the same combination of the relevant two vectors.

Matrix factorization is a linear modeling technique and only works well for average customers and on long-lived items. With recent advances in machine learning, we can now extend these ideas to non-linear, feed-forward neural network (FFNN)-based methods. For niche products and rare customers, these methods are a better model.

The input is a vector representing what a user has rated or consumed in the past. The output (the label) is a vector representing what they will rate or consume in the next time period. For example, the next week. In training, the label is the most recent past week. In production, the predictions are used to provide recommendations for a short horizon into the future. For example, one day ahead.

Mathematically, the FFNN with a hidden layer is a generalization of matrix factorization, with the additional advantage that the model is application-aware. This means that the model learns to predict the next item a customer will interact with. Within Amazon Personalize, this model is called a Deep Factorization Machine (DeepFM).

This predefined recipe has the following properties:

- **Name** – `aws-deepfm`
- **Recipe ARN** – `arn:aws:personalize:::recipe/aws-deepfm`
- **Algorithm ARN** – `arn:aws:personalize:::algorithm/aws-deepfm`
- **Feature Transformation ARN** – `arn:aws:personalize:::feature-transformation/JSON-percentile-filtering`
- **Recipe type** – `USER_PERSONALIZATION`

The following table lists the hyperparameters used in the recipe. For each hyperparameter, the following properties are provided:

- Name
- Range – [lower bound, upper bound]
- Default value
- Value type – Categorical, Continuous, Integer, Boolean, List, String
- HPO – Should be tuned during hyperparameter optimization.

Name	Range	Default value	Value type	HPO tuning	Description
nHiddenLayers	[0, 5]	2	Integer	Yes	Number of hidden layers.
embedding_size	[4, 512]	32	Integer	Yes	Number of hidden units per layer.
dropout	[0.0, 1.0]	0	Continuous	Yes	Dropout rate at each hidden layer.
epochs	[1, 32]	3	Integer	No	Number of passes over the complete dataset to run.
mini_batch_size	[2, 256]	64	Integer	No	Size (partial dataset) used for each mini-training iteration.

FFNN Recipe

This predefined recipe uses a feed-forward neural network (FFNN) with the following properties:

- **Name** – `aws-ffnn`
- **Recipe ARN** – `arn:aws:personalize:::recipe/aws-ffnn`
- **Algorithm ARN** – `arn:aws:personalize:::algorithm/aws-ffnn`
- **Feature Transformation ARN** – `arn:aws:personalize:::feature-transformation/general-interactions-recordIO-percentile-filtering`
- **Recipe type** – `USER_PERSONALIZATION`

The following table lists the hyperparameters used in the recipe. For each hyperparameter, the following properties are provided:

- Name
- Range – `[lower bound, upper bound]`
- Default value
- Value type – Categorical, Continuous, Integer, Boolean, List, String
- HPO – Should be tuned during hyperparameter optimization.

Name	Range	Default value	Value type	HPO tuning	Description
batch_size	[1, 1000]	1000	Integer	Maybe	Size (partial dataset) used for each training iteration.

Name	Range	Default value	Value type	HPO tuning	Description
dsize	[1, 128]	10	Integer	Yes	Number of hidden units per layer.
dropout	[0.0, 1.0]	0.2	Continuous	Yes	Dropout rate at each hidden layer.
learning_rate	[0.0, 1.0]	0.01	Continuous	Yes	Learning rate.
max_hist_length	[100, 5000]	500	Integer	No	Maximum number of items to consider in user's history.
min_hist_length	[1, 100]	5	Integer	No	Minimum number of items to consider in user's history.

HRNN Recipe

Amazon Personalize has implemented a hierarchical recurrent neural network (HRNN), which is able to model the changes in user behavior.

Temporal modeling is important in recommendation systems because user intent and interests tend to change or drift with time. This drifting is hard to model with traditional approaches. For example, a common approach in factorization machines is to manually come up with a discounting function for distant interactions (distance measured by time). Such manual engineering of weights is human-effort intensive and is prone to inaccuracy. The Amazon Personalize HRNN model, on the other hand, can take ordered user histories and make correct inferences. HRNN uses a gating mechanism to model the discount weights as a learnable function of the items and timestamp. The hierarchical component further improves temporal model efficiency and leads to higher accuracy.

From a usage perspective, Amazon Personalize has already derived the features for each user from the dataset you provided. If you have done real time data integration, these features are updated in real time according to user activity. Thus, you only provide *user-id* at inference. While the system does not throw an error if you also provide an *item-id*, the system neglects the value and the value does not impact the results.

This predefined recipe has the following properties:

- **Name** – `aws-hrnn`
- **Recipe ARN** – `arn:aws:personalize:::recipe/aws-hrnn`
- **Algorithm ARN** – `arn:aws:personalize:::algorithm/aws-hrnn`
- **Feature Transformation ARN** – `arn:aws:personalize:::feature-transformation/JSON-percentile-filtering`
- **Recipe type** – `USER_PERSONALIZATION`

The following table lists the hyperparameters used in the recipe. For each hyperparameter, the following properties are provided:

- Name
- Range – `[lower bound, upper bound]`
- Default value
- Value type – Categorical, Continuous, Integer, Boolean, List, String
- HPO – Should be tuned during hyperparameter optimization.

Name	Range	Default value	Value type	HPO tuning	Description
model.num_layers	[1, 3]	1	Integer	Yes	Number of hidden layers.
model.num_hidden	[32, 256]	149	Integer	Yes	Number of hidden units per layer.
model.dropout	[0.0, 1.0]	0.2	Continuous	Yes	Dropout rate at each hidden layer.
model.skip_connection	TRUE/FALSE	FALSE	Boolean	Yes	Skip connection between HRNN input and output.
training.epochs	[1, 1000)	1	Integer	No	Number of passes over the complete dataset to run.
training.batch_size	[1, 128)	31	Integer	No	Size (partial dataset) used for each training iteration.
training.early_time	[0, 86400]	43200	Integer	Yes, if time is a constraint	Limit training time (may need extra validation/ metric time).
training.eval_epoch	[0, +time		Integer	Yes, if time is a constraint	Limit validation/ metric time.

Popularity-Count Recipe

Popularity-count counts the popularity of items in a dataset. It returns the top popular items defined by the number of times each item occurs in the dataset. The recipe returns the same popular items for different users. Popularity-count is a good starting point to compare the results of other recipes against.

This predefined recipe has the following properties:

- **Name** – `aws-popularity-count`

- **Recipe ARN** – `arn:aws:personalize:::recipe/aws-popularity-count`
- **Recipe type** – `USER_PERSONALIZATION`

Popularity-count has no exposed hyperparameters.

Personalized-Ranking Recipe

Personalized ranking. This predefined recipe has the following properties:

- **Name** – `aws-personalized-ranking`
- **Recipe ARN** – `arn:aws:personalize:::recipe/aws-personalized-ranking`
- **Algorithm ARN** – `arn:aws:personalize:::algorithm/aws-personalized-ranking`
- **Feature Transformation ARN** – `arn:aws:personalize:::feature-transformation/JSON-percentile-filtering`
- **Recipe type** – `SEARCH_PERSONALIZATION`

SIMS Recipe

Item-to-item similarities (SIMS) is a model that generates items similar to a given item. In essence, the model is based on the concept of collaborative filtering, and leverages user-item interaction data to recommend items similar to a given item. In the absence of sufficient user behavior data for an item, this algorithm will return popular items as recommendations.

This predefined recipe has the following properties:

- **Name** – `aws-sims`
- **Recipe ARN** – `arn:aws:personalize:::recipe/aws-sims`
- **Algorithm ARN** – `arn:aws:personalize:::algorithm/aws-sims`
- **Feature Transformation ARN** – `arn:aws:personalize:::feature-transformation/sims`
- **Recipe type** – `RELATED_ITEMS`

The following table lists the hyperparameters used in the recipe. For each hyperparameter, the following properties are provided:

- Name
- Range – `[lower bound, upper bound]`
- Default value
- Value type – Categorical, Continuous, Integer, Boolean, List, String
- HPO – Should be tuned during hyperparameter optimization.

Name	Range	Default value	Value type	HPO tuning	Description
alpha	[0.0, 1.0]	0.5	Continuous	Yes	Balances the importance given to recent interactions versus older interactions. The higher the alpha,

Name	Range	Default value	Value type	HPO tuning	Description
					the more importance given to recent interactions.
copurchase_delta	[0, ∞)	10	Integer	No	Maximum number of days between interactions of a user with two items, for the items to be considered similar candidates to each other.
min_copurchase_count	[0, ∞)	5	Integer	No	Minimum number of users who need to have interacted with item A and item B for them to be considered similar to each other.
exponentiation_parameter	[0.0, 10]	1	Continuous	No	Determines correlation between co-purchases of an item.
epsilon_smoothing	[0, ∞)	3	Integer	No	Smoothing factor for items with insufficient co-purchase data.
history_decay_fast_half_life	[0, ∞)	90	Integer	Yes [30 - 150]	Fast decay parameter that downweights interactions based on how many days in the past they occurred.

Name	Range	Default value	Value type	HPO tuning	Description
history_decay_slow_half_life	[0, ∞)	360	Integer	Yes [120 - 720]	Slow decay parameter that downweights interactions based on how many days in the past they occurred.
in_order_decay_fast_half_life	[0, ∞)	60	Integer	Yes [20 - 120]	Fast decay parameter that downweights interactions that occurred in-order (A before B), based on how many days have elapsed between interactions.
in_order_decay_slow_half_life	[0, ∞)	360	Integer	Yes [120 - 720]	Slow decay parameter that downweights interactions that occurred in-order (A before B), based on how many days have elapsed between interactions.
out_of_order_decay_fast_half_life	[1, ∞)	30	Integer	Yes [10 - 45*]	Fast decay parameter that downweights interactions that occurred out-of-order (B before A), based on how many days have elapsed between interactions.

Name	Range	Default value	Value type	HPO tuning	Description
out_of_order_decay_slow_half_life	[1, ∞)	180	Integer	Yes [10 - 45*]	Slow decay parameter that downweights interactions that occurred out-of-order (B before A), based on how many days have elapsed between interactions.
top_n	[1, ∞)	100	Integer	No	Number of similar items to return for each item.

* The maximum value should be less than the *history_decay_slow_half_life* and *in_order_decay_slow_half_life* properties.

Bring Your Own Recipe

Bring your own Recipe (BYOR) allows you to add your own algorithms to Amazon Personalize. These added recipes will be part of the same data pipelines, training APIs, and recommendations as Personalize's own algorithms. You can run metrics and benchmark your algorithms against Amazon Personalize's algorithms in order to choose the most appropriate model for your needs.

With BYOR, you need to tell Amazon Personalize where the algorithm is stored and supply information that lets Amazon Personalize train and evaluate the algorithm. You supply default values that can be overriden when the algorithm is trained.

You use your recipe by specifying it as a parameter to the CreateSolution (p. 106) API or by adding the recipe to the list of recipes Amazon Personalize searches when performing AutoML.

To bring your own recipe, follow these steps:

1. Containerize Your Algorithm

   You first need to containerize your algorithm in an Amazon SageMaker compatible manner.

   a. Define Your Algorithm

   Follow this Amazon SageMaker example to define your algorithm: https://github.com/awslabs/amazon-sagemaker-examples/tree/master/advanced_functionality/scikit_bring_your_own/container.

   In the above example the 'train' program contains your custom algorithm training logic and `predictor.py` your inference logic. During training and inference, Amazon Personalize feeds featured data to your container. Additional transformations on the data are optional.

   You are able to export featurized data from Amazon Personalize to help you test and build your training logic through the CreateFeatureExportJob (p. 96) API.

b. Containerize Your Algorithm and Push to Amazon EC2 Container Registry (ECR)

To package your algorithm and push to ECR, you can run the `build_and_push.sh` script included in the example.

2. Allow Amazon Personalize Access to Your Container

Add the following permission to your ECR repository to allow Amazon Personalize to download your container image.

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "ECR Image download policy",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "personalize.amazonaws.com"
                ]
            },
            "Action": [
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability"
            ]
        }
    ]
}
```

3. Create a New Recipe in Amazon Personalize

A recipe in Amazon Personalize is composed of an algorithm, including hyperparameters, and feature transformation.

a. Create the Algorithm

Create the algorithm by running the following command. For more information about the API, see CreateAlgorithm (p. 77). For more information about the input parameters, see Hyperparameter Optimization (p. 56). You can override all default parameters using the solution configuration when creating a solution.

```
aws personalize create-algorithm \
    --name BYOAlgorithm \
    --algorithm-image  \
    --algorithm-data-format-type  \
    --default-hyper-parameters \
    --default-hyper-parameter-ranges \
    --default-hpo-objective \
    --default-resource-config \
    --role-arn \
    --training-input-mode
```

The algorithm ARN is displayed. For example:

```
{
    "algorithmArn": "arn:aws:personalize::acct-id:algorithm/BYOAlgorithm"
}
```

You can use the `describe-algorithm` CLI command to display the algorithm you created:

```
aws personalize describe-algorithm --algorithm-arn arn:aws:personalize::acct-
id:algorithm/BYOAlgorithm
```

The algorithm and its parameters are displayed.

You can use the `update-algorithm` CLI command to update the algorithm.

b. Create the Feature Transformation

A feature transformation specifies information about how to featurize the input data. For
example, setting maximum user history length percentile to a value between 0 and 1. You can
provide different parameters for feature transformation customization. The input interaction
data is featurized based on the values provided for the supported parameter keys. The set of
currently supported parameter overrides for personalization models are:

- max_hist_length_percentile (maximum user history length to be filtered based on percentile)

- min_hist_length_percentile (minimum user history length to be filtered based on percentile)

- train ratio

- test ratio

Create the feature transformation by running the following command. For more information
about the API, see CreateFeatureTransformation (p. 99). For more information about feature
transformation, see Feature Transformation (p. 56).

```
aws personalize create-feature-transformation \
    --name BYOFeatureTransformation, \
    --defaultParameters '{"max_hist_length_percentile": 0.95,
 "min_hist_length_percentile": 0.1}
```

The feature transformation ARN is displayed. For example:

```
{
    "featureTransformationArn": "arn:aws:personalize::acct-id:feature-
transformation/BYOFeatureTransformation"
}
```

You can use the `describe-feature-transformation` CLI command to display the feature
transformation you created.

You can use the `update-feature-transformation` CLI command to update the feature
transformation.

c. Create the Recipe

Create the recipe by running the following command and supplying the `algorithmArn` and
`featureTransformationArn` from the previous steps. For more information about the API,
see CreateRecipe (p. 101).

```
aws personalize create-recipe \
    --name BYORecipe \
    --algorithm-arn arn:aws:personalize::acct-id:algorithm/BYOAlgorithm \
    --feature-transformation-arn arn:aws:personalize::acct-id:feature-
transformation/BYOFeatureTransformation
```

The recipe ARN is displayed. For example:

```
{
    "recipeArn": "arn:aws:personalize::acct-id:recipe/BYORecipe"
}
```

You can use the `describe-recipe` CLI command to display the recipe you created.

You can use the `update-recipe` CLI command to update the recipe.

Overriding Default Recipe Parameters

A solution is created by calling the CreateSolution (p. 106) API. A slightly condensed, reorganized version of the `CreateSolution` request is shown, highlighting the solution configuration. The solution configuration is used to override the default parameters of a recipe, both predefined recipes and BYOR. All solution configuration properties, except `autoMLConfig`, are ignored when `performAutoML` is true. The various configuration objects are discussed in the following sections.

```
{
    "name": "string",
    "recipeArn": "string",
    "performAutoML": boolean,
    "performHPO": boolean,
    "eventType": "string",
    "solutionConfig": {
        "autoMLConfig": {
            "metricName": "string",
            "recipeList": [ "string" ]
        },
        "algorithmHyperParameters": {
            "string" : "string"
        },
        "featureTransformationParameters": {
            "string" : "string"
        },
        "hpoConfig": {
            "algorithmHyperParameterRanges": {
                ...
            },
            "hpoObjective": {
                "metricName": "string",
                "metricRegex": "string",
                "type": "string"
            },
            "hpoResourceConfig": {
                "maxNumberOfTrainingJobs": "string",
                "maxParallelTrainingJobs": "string"
            }
        }
        "eventValueThreshold": "string",
        "retrainingMode": "string",
        "retrainSchedule": "string"
    },
}
```

Creating an Algorithm

When you "Bring Your Own Recipe", you need to tell Amazon Personalize where the algorithm is stored and supply information that lets Amazon Personalize train and evaluate the algorithm. You supply default values that can be overriden when the algorithm is trained.

Hyperparameter Optimization

A hyperparameter is used to optimize the algorithm training process and is set before training begins. This contrasts with a model parameter whose value is determined by the training process.

Hyperparameter optimization (HPO), or tuning, is the task of choosing optimal hyperparameters for a specific learning objective.

Hyperparameters can be categorical, continuous, or integer-valued.

Hyperparameters and their properties are specified as part of the SolutionConfig (p. 278) object passed to the CreateSolution (p. 106) and UpdateSolution (p. 203) APIs.

For more information, see Automatic Model Tuning.

Feature Transformation

Feature transformation is the process of modifying raw input data into a form more suitable for model training. For example, a categorical field of strings could be modified by assigning a unique number to each category. The modified data would then be used as input for the model training.

Amazon Personalize provides two methods to support feature transformation, CreateFeatureTransformation (p. 99) and CreateFeatureExportJob (p. 96). You use `CreateFeatureTransformation` to create a list of key/value pairs. Next you call `CreateFeatureExportJob`, specifying the `FeatureTransformation` object. `CreateFeatureExportJob` reads the input data, replaces each key with it's corresponding value, and then exports the modified data to be used in model training.

Evaluating Solutions

Metrics are a great way to evaluate the performance of a recommendation system before applying it to live online traffic. A personalization recipe has good performance when it shows significantly better metrics than the popularity baseline recipe – a recipe that only recommends the top K most popular items. Metrics can also be used to distinguish the performance between different recipes and to choose the best one. Since the differences may sometimes be small, we report multiple types of metrics and pick the one that scores top on the most items.

Disclaimer: Offline metrics are subject to data collection artifacts such as seasonal and sales events. While we spend a lot of research on calibrating offline metrics to online performance and obtained some better than state-of-art results, the true performance can only be unbiasedly evaluated by online A/B tests.

The following metrics are accompanied with an example, where the system generates a list of 4 recommendations and the second and fourth are clicked by the user, that is, the relevance vector is [0,1,0,1]. Some metrics consider positional effects because items that appear at the top of the list naturally attract more traffic. All of the metrics are truncated by top K recommendations.

Name	Example	Explanation
precision@K	2/4 = 0.5	Total relevant items divided by total recommended items.
mean reciprocal ranks (MRR@K)	mean(1/2 + 1/4) = 0.375	Considers positional effects by computing the mean of the inverse positions of all relevant items.
normalized discounted cumulative gains (NDCG@K)	(1/log(1 + 2) + 1/log(1 + 4)) / (1/log(1 + 1) + 1/log(1 + 2)) = 0.65	Considers positional effects by applying inverse logarithmic weights based on the positions of relevant items, normalized by the largest possible scores from ideal recommendations.
average precision (AP@K)	mean(1/2 + 2/4) = 0.5	Average precision@K where K is the position of every relevant item.

Amazon Personalize provides the following metrics (for each of these, higher numbers are better):

- **Precision@K**: number of relevant items / number of total recommendations (K). This metric rewards precise recommendation of the relevant items.
- **DCG@K**: discounted cumulative gain for relevant recommendations. This metric rewards relevant items that appear early on the list, because the earlier parts of the list usually draw more attention. Mathematically, it assigns weight = $1/[\log_2 (1 + position)]$ to relevant items and zero to irrelevant items.
- **NDCG@K**: normalized discounted cumulative gain. This divides DCG by the ideal DCG (all relevant items at the top), so that the results are between 0 and 1.
- **Average precision**: sum of precisions@K, where each K is the position of a relevant item on the recommendation list.

Note

@K means that the metrics only look at the top-*K* recommendations.

Creating a Campaign

You create a campaign by deploying a solution. You can create and update a campaign by using the AWS console or by using the AWS SDK.

To create a campaign with the SDK call the section called "CreateCampaign" (p. 80) and pass the following:

- A name for the campaign.
- The Amazon Resource Name (ARN) of the solution to deploy.
- An update mode.

If the update mode is AUTO, the latest version of the solution is automatically deployed whenever the solution is updated. If it is MANUAL, the campaign must be manually updated by calling UpdateCampaign (p. 192). If you want to create a campaign using a specific version of a solution, specify the optional `SolutionVersionArn` input parameter.

To create a campaign

1. Create a solution to deploy. For more information, see Creating a Solution (p. 41).
2. Use the following code to create a campaign.

```python
import boto3

if __name__ == "__main__":

    personalize = boto3.client('personalize', region_name='us-west-2')

    response=personalize.create_campaign(
        name='PythonCampaign',
        solutionArn='Solution arn',
        updateMode='AUTO' )

    arn=response['campaignArn']

    description=personalize.describe_campaign(campaignArn=arn)['campaign']
    print('Name: ' + description['name'])
    print('ARN: ' + description['campaignArn'])
    print('Status: ' + description['status'])
```

The campaign isn't ready for use until its status is active. To get the current status, call DescribeCampaign (p. 133) and check that the `status` field is `ACTIVE`.

Amazon Personalize provides operations for managing campaigns such as ListCampaigns (p. 161) to list the campaigns you have created. You can delete a campaign by calling DeleteCampaign (p. 113). If you delete a campaign, the solutions that are part of the campaign are not deleted.

After you have created your campaign, your deployed solution is able to make recommendations. For more information, see Getting Recommendations (p. 60).

Getting Recommendations

Amazon Personalize can get recommendations and personalized rankings from a campaign. For example, if you have a campaign that is trained to give movie recommendations, you can use the following operations to give movie recommendations to users signed into your application or website. For an example, see Getting Started (AWS CLI) (p. 18).

The Amazon Personalize console details page for a campaign includes example code that you can use to get recommendations.

Recommendations

Once you have created your campaign, you can use it in your applications to get recommendations.

To get recommendations, you call the GetRecommendations (p. 208) API. You supply the Amazon Resource Name (ARN) of the required campaign, the user ID, and optionally, the item ID. `GetRecommendations` returns a list of recommended items for the user.

To get a recommendation

1. Use the following code to get a recommendation. Change the value of `campaignArn` to the ARN of a valid campaign. Change the value of `userId` and `itemId` to a user ID and item ID that are in the data you used to train the solution.

```python
import boto3

if __name__ == "__main__":

    personalizert = boto3.client('personalize-runtime', region_name='us-west-2')

    response=personalizert.get_recommendations(
        campaignArn="Campaign ARN",
        userId='User ID',
        itemId='Item ID')

    print("Recommended items")

    for item in response['itemList']:
            print (item['itemId'])
```

2. Run the code. A list of recommended items for the user is displayed.

Personalized Rankings

A personalized ranking is a list of recommended items that are re-ranked for a specific user. To get personalized rankings, call the PersonalizeRanking (p. 211) API. You supply the ARN of the required campaign, the user ID, and a list of recommended items.

Note
The solution backing the campaign must have been created using a recipe of type SEARCH_PERSONALIZATION. For more information, see Using Predefined Recipes (p. 43).

To get a personalized ranking

1. Use the following code to get a personalized ranking. Change the value of `campaignArn` to the ARN of a valid campaign. Change the value of `userId` and `inputList` to a user ID and item list that are in the data you used to train the solution.

```python
import boto3

if __name__ == "__main__":

    personalizert = boto3.client('personalize-runtime', region_name='us-west-2')

    response=personalizert.personalize_ranking(
        campaignArn="Campaign arn",
        userId='UserID',
        inputList=['ItemID1','ItemID2'])

    print("Personalized Ranking")
    for item in response['personalizedRanking']:
        print (item['itemId'])
```

2. Run the code. A list of ranked recommendations is displayed. The first item in the list is considered by Amazon Personalize to be of most interest to the user.

Amazon Personalize Security

At AWS, security is the highest priority. The following topics explain how to help secure your Amazon Personalize resources and data.

Topics

- Authentication and Access Control for Amazon Personalize (p. 62)
- Logging Amazon Personalize API Calls with AWS CloudTrail (p. 70)

Authentication and Access Control for Amazon Personalize

Access to Amazon Personalize requires credentials. Those credentials must have permissions to access AWS resources, such as an Amazon Personalize dataset. The following sections provide details on how you can use AWS Identity and Access Management (IAM) and Amazon Personalize to help secure access to your resources.

- Authentication (p. 62)
- Access Control (p. 63)

Authentication

You can access AWS as any of the following types of identities:

- **AWS account root user** – When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

- **IAM user** – An IAM user is an identity within your AWS account that has specific custom permissions (for example, permissions to create a dataset in Amazon Personalize). You can use an IAM user name and password to sign in to secure AWS webpages like the AWS Management Console, AWS Discussion Forums, or the AWS Support Center.

  In addition to a user name and password, you can also generate access keys for each user. You can use these keys when you access AWS services programmatically, either through one of the several SDKs or by using the AWS Command Line Interface (CLI). The SDK and CLI tools use the access keys to cryptographically sign your request. If you don't use AWS tools, you must sign the request yourself. Amazon Personalize supports *Signature Version 4*, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 Signing Process in the *AWS General Reference*.

- **IAM role** – An IAM role is an IAM identity that you can create in your account that has specific permissions. It is similar to an *IAM user*, but it is not associated with a specific person. An IAM role

enables you to obtain temporary access keys that can be used to access AWS services and resources. IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – Instead of creating an IAM user, you can use existing user identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as *federated users*. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated Users and Roles in the *IAM User Guide*.

- **AWS service access** – You can use an IAM role in your account to grant an AWS service permissions to access your account's resources. For example, you can create a role that allows Amazon Redshift to access an Amazon S3 bucket on your behalf and then load data from that bucket into an Amazon Redshift cluster. For more information, see Creating a Role to Delegate Permissions to an AWS Service in the *IAM User Guide*.

- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM Role to Grant Permissions to Applications Running on Amazon EC2 Instances in the *IAM User Guide*.

Access Control

You can have valid credentials to authenticate your requests, but unless you have permissions you cannot create or access Amazon Personalize resources. For example, you must have permissions to create an Amazon Personalize dataset.

The following sections describe how to manage permissions for Amazon Personalize. We recommend that you read the overview first.

- Overview of Managing Access Permissions to Your Amazon Personalize Resources (p. 63)
- Using Identity-Based Policies (IAM Policies) for Amazon Personalize (p. 67)
- Amazon Personalize API Permissions: Actions, Permissions, and Resources Reference (p. 70)

Overview of Managing Access Permissions to Your Amazon Personalize Resources

Every AWS resource is owned by an AWS account, and permissions to create or access a resource are governed by permissions policies. An account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles), and some services (such as AWS Lambda) also support attaching permissions policies to resources.

> **Note**
> An *account administrator* (or administrator user) is a user with administrator privileges. For more information, see IAM Best Practices in the *IAM User Guide*.

When granting permissions, you decide who is getting the permissions, the resources they get permissions for, and the specific actions that you want to allow on those resources.

Topics

Amazon Personalize Resources and Operations

In Amazon Personalize, there are resource types for *an algorithm*, a *campaign*, *a dataset*, a *dataset group*, a *dataset import job*, an *event tracker*, a *feature export job*, a *feature transformation*, a *recipe*, and a *schema*. In a policy, you use an Amazon Resource Name (ARN) to identify the resource that the policy applies to.

These resources have unique Amazon Resource Names (ARNs) associated with them, as shown in the following table.

Resource Type	ARN Format
Algorithm ARN	`arn:aws:personalize:`*`region`*`:`*`account-id`*`:`*`algorithm/ algorithm-name`*
Campaign ARN	`arn:aws:personalize:`*`region`*`:`*`account-id`*`:`*`campaign/ campaign-name`*
Dataset ARN	`arn:aws:personalize:`*`region`*`:`*`account-id`*`:`*`dataset/dataset- name`*
Dataset group ARN	`arn:aws:personalize:`*`region`*`:`*`account-id`*`:`*`dataset-group/ dataset-group-name`*
Dataset import job ARN	`arn:aws:personalize:`*`region`*`:`*`account-id`*`:`*`dataset-import- job/dataset-import-job-name`*
Event tracker ARN	`arn:aws:personalize:`*`region`*`:`*`account-id`*`:`*`event-tracker/ event-tracker-name`*
Feature export job ARN	`arn:aws:personalize:`*`region`*`:`*`account-id`*`:`*`feature-export- job/feature-export-job-name`*
Feature Transformation ARN	`arn:aws:personalize:`*`region`*`:`*`account-id`*`:`*`feature- transformation/feature-transformation-name`*
Recipe ARN	`arn:aws:personalize:`*`region`*`:`*`account-id`*`:`*`recipe/recipe- name`*
Schema ARN	`arn:aws:personalize:`*`region`*`:`*`account-id`*`:`*`schema/schema- name`*
Solution ARN	`arn:aws:personalize:`*`region`*`:`*`account-id`*`:`*`solution/ solution-name`*
Solution Version ARN	`arn:aws:personalize:`*`region`*`:`*`account-id`*`:`*`solution-version/ solution-version-name`*

Amazon Personalize provides a set of operations to work with Amazon Personalize resources. For a list of available operations, see Amazon Personalize API Permissions Reference (p. 70).

Understanding Resource Ownership

The AWS account owns the resources that are created in the account, regardless of who created the resources. Specifically, the resource owner is the AWS account of the principal entity (that is, the root account or an IAM user) that authenticates the resource creation request. The following examples illustrate how this works:

- If you use the root account credentials of your AWS account to create a dataset, your AWS account is the owner of the resource (in Amazon Personalize, the resource is a dataset).
- If you create an IAM user in your AWS account and grant permissions to create a dataset to that user, the user can create a dataset. However, your AWS account, to which the user belongs, owns the dataset resource.

Managing Access to Resources

A *permissions policy* describes who has access to what. The following section explains the available options for creating permissions policies.

> **Note**
> This section discusses using IAM in the context of Amazon Personalize. It doesn't provide detailed information about the IAM service. For complete IAM documentation, see What Is IAM? in the *IAM User Guide*. For information about IAM policy syntax and descriptions, see AWS IAM Policy Reference in the *IAM User Guide*.

Policies attached to an IAM identity are referred to as *identity-based* policies (IAM polices) and policies attached to a resource are referred to as *resource-based* policies. Amazon Personalize supports identity-based policies.

Topics
- Identity-Based Policies (IAM Policies) (p. 65)
- Resource-Based Policies (p. 66)

Identity-Based Policies (IAM Policies)

You can attach policies to IAM identities. For example, you can do the following:

- **Attach a permissions policy to a user or a group in your account** – To grant a user permissions to create an Amazon Personalize resource, such as a dataset, you can attach a permissions policy to a user or group that the user belongs to.
- **Attach a permissions policy to a role (grant cross-account permissions)** – You can attach an identity-based permissions policy to an IAM role to grant cross-account permissions. For example, the administrator in account A can create a role to grant cross-account permissions to another AWS account (for example, account B) or an AWS service as follows:
  1. Account A administrator creates an IAM role and attaches a permissions policy to the role that grants permissions on resources in account A.
  2. Account A administrator attaches a trust policy to the role identifying account B as the principal who can assume the role.
  3. Account B administrator can then share the role with users in account B. Doing this allows users in account B to create or access resources in account A on behalf of account A. The principal in the trust policy can also be an AWS service principal if you want to grant an AWS service permissions to assume the role.

For more information about using IAM to delegate permissions, see Access Management in the *IAM User Guide*.

The following is an example policy that lists all dataset groups.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "personalize:CreateDatasetGroup"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "personalize:CreateDataset"
            ],
            "Resource": [
                "arn:aws:personalize:region:acct-id:dataset-group/*",
                "arn:aws:personalize:region:acct-id:schema/*"
            ]
        }
    ]
}
```

The policy has two statements:

- The first statement grants permission for the `personalize:CreateDatasetGroup` action. Because the CreateDatasetGroup operation does not act on any particular Amazon Personalize resource, the action grants user permission to create any dataset group.
- The second statement grants permission for the `personalize:CreateDataset` action. To create a dataset, you need to specify a dataset group and schema. The policy gives access to all dataset groups and schema resources in the AWS account and AWS Region.

The policy doesn't specify the `Principal` element because in an identity-based policy you don't specify the principal who gets the permission. When you attach the policy to a user, the user is the implicit principal. When you attach a permissions policy to an IAM role, the principal identified in the role's trust policy gets the permissions.

For more information about using identity-based policies with Amazon Personalize, see Using Identity-Based Policies (IAM Policies) for Amazon Personalize (p. 67). For more information about users, groups, roles, and permissions, see Identities (Users, Groups, and Roles) in the *IAM User Guide*.

Resource-Based Policies

Other services, such as Amazon S3, also support resource-based permissions policies. For example, you can attach a policy to an S3 bucket to manage access permissions to that bucket. Amazon Personalize supports resource-based policies.

Specifying Policy Elements: Actions, Effects, and Principals

For each Amazon Personalize resource, the service defines a set of API operations. To grant permissions for these API operations, Amazon Personalize defines a set of actions that you can specify in a policy. Some API operations can require permissions for more than one action in order to perform the API

operation. For more information about resources and API operations, see Amazon Personalize Resources and Operations (p. 64) and Amazon Personalize API Permissions Reference (p. 70).

The following are the most basic policy elements:

- **Resource** – You use an Amazon Resource Name (ARN) to identify the resource that the policy applies to. For more information, see Amazon Personalize Resources and Operations (p. 64).
- **Action** – You use action keywords to identify resource operations that you want to allow or deny. For example, you can use `personalize:ListDatasets` to list datasets.
- **Effect** – You specify the effect, either allow or deny, when the user requests the specific action. If you don't explicitly grant access to (allow) a resource, access is implicitly denied. You can also explicitly deny access to a resource, which you might do to make sure that a user cannot access it, even if a different policy grants access.
- **Principal** – In identity-based policies (IAM policies), the user that the policy is attached to is the implicit principal. For resource-based policies, you specify the user, account, service, or other entity that you want to receive permissions (applies to resource-based policies only). Amazon Personalize doesn't support resource-based policies.

To learn more about IAM policy syntax and descriptions, see AWS IAM Policy Reference in the *IAM User Guide*.

For a list showing all of the Amazon Personalize API operations and the resources that they apply to, see Amazon Personalize API Permissions: Actions, Permissions, and Resources Reference (p. 70).

Specifying Conditions in a Policy

When you grant permissions, you can use the access policy language to specify the conditions when a policy should take effect. For example, you might want a policy to be applied only after a specific date. For more information about specifying conditions in a policy language, see Condition in the *IAM User Guide*.

To express conditions, you use predefined condition keys. There are no condition keys specific to Amazon Personalize. However, there are AWS-wide condition keys that you can use as appropriate. For a complete list of AWS-wide keys, see Available Keys for Conditions in the *IAM User Guide*.

Specifying a Service Role

Certain Amazon Personalize operations require permissions to to access AWS services on your behalf. For example, to import your data into a dataset with the section called "CreateDatasetImportJob" (p. 89), Amazon Personalize needs access to the Amazon S3 Bucket that contains the data. To give permissions you create an IAM service role. For more information, see Creating an IAM Role (p. 7).

Using Identity-Based Policies (IAM Policies) for Amazon Personalize

This topic provides examples of identity-based policies that demonstrate how an account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles) and thereby grant permissions to perform operations on Amazon Personalize resources.

> **Important**
> We recommend that you first review the introductory topics that explain the basic concepts and options available to manage access to your Amazon Personalize resources. For more information, see Overview of Managing Access Permissions to Your Amazon Personalize Resources (p. 63).

Topics

The following shows an example of a permissions policy.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "personalize:ListDatasetGroups"
            ],
            "Resource": "*"
        }
    ]
}
```

This policy example grants a user permission to list the dataset groups in their account.

For a table showing all of the Amazon Personalize API operations and the resources that they apply to, see Amazon Personalize API Permissions: Actions, Permissions, and Resources Reference (p. 70).

Permissions Required to Use the Amazon Personalize Console

The permissions reference table lists the Amazon Personalize API operations and shows the required permissions for each operation. For more information about Amazon Personalize API operations, see Amazon Personalize API Permissions: Actions, Permissions, and Resources Reference (p. 70). Amazon Personalize does not require any additional permissions when working with the Amazon Personalize console.

AWS Managed (Predefined) Policies for Amazon Personalize

AWS addresses many common use cases by providing standalone IAM policies that are created and administered by AWS. These AWS managed policies grant necessary permissions for common use cases so that you can avoid having to investigate what permissions are needed. For more information, see AWS Managed Policies in the *IAM User Guide*.

The following AWS managed policies, which you can attach to users in your account, are specific to Amazon Personalize:

- **AmazonPersonalizeFullAccess** – Grants full access to Amazon Personalize resources.

    Note
    You can review these permissions policies by signing in to the IAM console and searching for specific policies there.
    These policies work when you are using AWS SDKs or the AWS CLI.

You can also create your own custom IAM policies to allow permissions for Amazon Personalize actions and resources. You can attach these custom policies to the IAM users or groups that require those permissions.

Customer Managed Policy Examples

In this section, you can find example user policies that grant permissions for various Amazon Personalize actions. These policies work when you are using AWS SDKs or the AWS CLI. When you are using

the console, you need to grant additional permissions specific to the console, which is discussed in
Permissions Required to Use the Amazon Personalize Console (p. 68).

Examples

- Example 1: Allow a User Read-Only Access to Resources (p. 69)
- Example 2: Allow a User Full Access to Resources (p. 69)

Example 1: Allow a User Read-Only Access to Resources

The following example grants read-only access to Amazon Personalize resources.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "personalize:DescribeAlgorithm",
                "personalize:DescribeCampaign",
                "personalize:DescribeDataset",
                "personalize:DescribeDatasetGroup",
                "personalize:DescribeDatasetImportJob",
                "personalize:DescribeDatasetImportJobRun",
                "personalize:DescribeEventTracker",
                "personalize:DescribeFeatureExportJob",
                "personalize:DescribeFeatureTransformation",
                "personalize:DescribeRecipe",
                "personalize:DescribeSchema",
                "personalize:DescribeSolution",
                "personalize:DescribeSolutionVersion",
                "personalize:GetMetrics",
                "personalize:ListCampaigns",
                "personalize:ListDatasetGroups",
                "personalize:ListDatasetImportJobRuns",
                "personalize:ListDatasetImportJobs",
                "personalize:ListDatasets",
                "personalize:ListEventTrackers",
                "personalize:ListFeatureExportJobs",
                "personalize:ListRecipes",
                "personalize:ListSchemas",
                "personalize:ListSolutions",
                "personalize:ListSolutionVersions"
            ],
            "Resource": "*"
        }
    ]
}
```

Example 2: Allow a User Full Access to Resources

The following example grants full access to all Amazon Personalize resources.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "personalize:*"
            ],
            "Resource": "*"
```

```
            }
        ]
}
```

Amazon Personalize API Permissions: Actions, Permissions, and Resources Reference

When you are setting up Access Control (p. 63) and writing a permissions policy that you can attach to an IAM identity (identity-based policies), you can use the following list as a reference. The list includes each Amazon Personalize API operation, the corresponding actions for which you can grant permissions to perform the action, and the AWS resource for which you can grant the permissions. You specify the actions in the policy's `Action` field, and you specify the resource value in the policy's `Resource` field.

To express conditions in your Amazon Personalize policies, you can use AWS-wide condition keys. For a complete list of AWS-wide keys, see Available Keys in the *IAM User Guide*.

> **Note**
> To specify an action, use the `personalize` prefix followed by the API operation name (for example, `personalize:DeleteDataset`).

Logging Amazon Personalize API Calls with AWS CloudTrail

Amazon Personalize is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon Personalize. CloudTrail captures a subset of API calls for Amazon Personalize as events, including calls from the Amazon Personalize console and from code calls to the Amazon Personalize APIs. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Amazon Personalize. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to Amazon Personalize, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, including how to configure and enable it, see the AWS CloudTrail User Guide.

Amazon Personalize Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When supported event activity occurs in Amazon Personalize, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for Amazon Personalize, create a trail. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all regions. The trail logs events from all regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations

- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

Amazon Personalize supports logging every action (API operation) as an event in CloudTrail log files. For more information, see Actions (p. 74).

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or IAM user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity Element.

Example: Amazon Personalize Log File Entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files are not an ordered stack trace of the public API calls, so they do not appear in any specific order.

The following example shows a CloudTrail log entry with actions for the ListDatasetGroups API operation:

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "principal-id",
        "arn": "arn:aws:iam::user-arn",
        "accountId": "account-id",
        "accessKeyId": "access-key,
        "userName": "user-name"
    },
    "eventTime": "2018-11-22T02:18:03Z",
    "eventSource": "personalize.amazonaws.com",
    "eventName": "ListDatasetGroups",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "source-ip-address",
    "userAgent": "aws-cli/1.11.16 Python/2.7.11 Darwin/15.6.0 botocore/1.4.73",
    "requestParameters": null,
    "responseElements": {
        "datasetGroups": [
            {
                "name": "testdatasetgroup",
                "datasetGroupArn": "arn:aws:personalize:us-west-2:nnnnnnnnnn:dataset-group/
testdataset",
                "status": "ACTIVE",
                "creationDateTime": "Nov 5, 2018 6:06:01 AM"
            }
        ]
    },
    "requestID": "request-id",
    "eventID": "event-id",
    "eventType": "AwsApiCall",
    "recipientAccountId": "recipient-account-id"
```

```
}
```

Limits in Amazon Personalize

The following is a list of limits in Amazon Personalize.

Maximum amount of data for an individual dataset	100 GB
Minimum increment for a cron schedule (importing data, training a model)	30 minutes
Minimum number of data points required for training a model (creating a solution)	1000
Maximum number of items that are considered by a model during training	1 million
Maximum number of points in history that are considered by a model during training	500 million

API Reference

This section provides documentation for the Amazon Personalize API operations.

Actions

The following actions are supported by Amazon Personalize:

The following actions are supported by Amazon Personalize Events:

The following actions are supported by Amazon Personalize Runtime:

Amazon Personalize

The following actions are supported by Amazon Personalize:

CreateAlgorithm

Service: Amazon Personalize

Creates a custom algorithm for use with the CreateRecipe (p. 101) API.

Request Syntax

```
{
    "algorithmImage": {
        "dockerURI": "string",
        "name": "string"
    },
    "defaultHyperParameterRanges": {
        "categoricalHyperParameterRanges": [
            {
                "isTunable": boolean,
                "name": "string",
                "values": [ "string" ]
            }
        ],
        "continuousHyperParameterRanges": [
            {
                "isTunable": boolean,
                "maxValue": number,
                "minValue": number,
                "name": "string"
            }
        ],
        "integerHyperParameterRanges": [
            {
                "isTunable": boolean,
                "maxValue": number,
                "minValue": number,
                "name": "string"
            }
        ]
    },
    "defaultHyperParameters": {
        "string" : "string"
    },
    "defaultResourceConfig": {
        "string" : "string"
    },
    "name": "string",
    "roleArn": "string",
    "trainingInputMode": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

algorithmImage (p. 77)

Specifies the URI of the Docker container for the algorithm image.

Type: AlgorithmImage (p. 218) object

Required: Yes

defaultHyperParameterRanges (p. 77)

Specifies the default hyperparameters, their ranges, and whether they are tunable. A tunable hyperparameter can have its value determined during hyperparameter optimization (HPO).

Type: DefaultHyperParameterRanges (p. 253) object

Required: Yes

defaultHyperParameters (p. 77)

A list of hyperparameters and their default values.

Type: String to string map

Key Length Constraints: Maximum length of 256.

Value Length Constraints: Maximum length of 1000.

Required: Yes

defaultResourceConfig (p. 77)

Specifies the default maximum number of training jobs and the maximum number of parallel training jobs.

Type: String to string map

Key Length Constraints: Maximum length of 256.

Value Length Constraints: Maximum length of 1000.

Required: Yes

name (p. 77)

The name for the algorithm.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: Yes

roleArn (p. 77)

The Amazon Resource Name (ARN) of the role.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Required: Yes

trainingInputMode (p. 77)

Type: String

Length Constraints: Maximum length of 256.

Required: Yes

Response Syntax

```
{
```

```
    "algorithmArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

algorithmArn (p. 78)

The ARN of the created algorithm.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

LimitExceededException

The limit on the number of requests per second has been exceeded.

HTTP Status Code: 400

ResourceAlreadyExistsException

The specified resource already exists.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

CreateCampaign

Service: Amazon Personalize

Creates a campaign by deploying a solution. When a client calls the GetRecommendations (p. 208) API, a campaign is specified in the request.

Status

A campaign can be in one of the following states:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -or- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS > DELETE FAILED

To get the campaign status, call DescribeCampaign (p. 133).

> **Note**
> You must wait until the `status` of the campaign is `ACTIVE` before asking the campaign for recommendations.

Scheduling

To have the campaign automatically updated every time the soluton is updated, set the `updateMode` to `AUTO`. When the `updateMode` is set to `MANUAL`, you must call UpdateCampaign (p. 192) for the campaign to use an updated solution.

Related APIs

- ListCampaigns (p. 161)
- DescribeCampaign (p. 133)
- UpdateCampaign (p. 192)
- DeleteCampaign (p. 113)

Request Syntax

```
{
   "name": "string",
   "solutionVersionArn": "string",
   "updateMode": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

name (p. 80)

A name for the new campaign. The campaign name must be unique within your account.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: Yes

solutionVersionArn (p. 80)

The Amazon Resource Name (ARN) of the solution version to deploy.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

updateMode (p. 80)

When the mode is AUTO, the campaign is automatically updated every time the solution is updated.

When the mode is MANUAL, you must call UpdateCampaign (p. 192) to update the campaign.

Type: String

Length Constraints: Maximum length of 256.

Valid Values: `AUTO | MANUAL`

Required: Yes

Response Syntax

```
{
    "campaignArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

campaignArn (p. 81)

The Amazon Resource Name (ARN) of the campaign.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

LimitExceededException

The limit on the number of requests per second has been exceeded.

HTTP Status Code: 400

ResourceAlreadyExistsException

The specified resource already exists.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

CreateDataset

Service: Amazon Personalize

Creates an empty dataset and adds it to the specified dataset group. You use
CreateDatasetImportJob (p. 89) to import your training data to a dataset.

There are three types of datasets:

- Interactions
- Items
- Users

Each dataset type has an associated schema with required field types. Only the `Interactions` dataset
is required to train a model (create a solution).

Related APIs

- CreateDatasetGroup (p. 86)
- ListDatasets (p. 172)
- DescribeDataset (p. 135)
- UpdateDataset (p. 194)
- DeleteDataset (p. 115)

Request Syntax

```
{
   "datasetGroupArn": "string",
   "datasetType": "string",
   "name": "string",
   "schemaArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

datasetGroupArn (p. 83)

> The Amazon Resource Name (ARN) of the dataset group to add the dataset to.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: Yes

datasetType (p. 83)

> The type of dataset.
>
> One of the following (case insensitive) values:
> - Interactions
> - Items

- Users

    Type: String

    Length Constraints: Maximum length of 256.

    Required: Yes

name (p. 83)

    The name for the dataset.

    Type: String

    Length Constraints: Minimum length of 1. Maximum length of 63.

    Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

    Required: Yes

schemaArn (p. 83)

    The ARN of the schema to associate with the dataset. The schema defines the dataset's fields.

    Type: String

    Length Constraints: Maximum length of 256.

    Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

    Required: Yes

Response Syntax

```
{
    "datasetArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

datasetArn (p. 84)

    The ARN of the dataset.

    Type: String

    Length Constraints: Maximum length of 256.

    Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errors

InvalidInputException

    Provide a valid value for the field or parameter.

    HTTP Status Code: 400

LimitExceededException

The limit on the number of requests per second has been exceeded.

HTTP Status Code: 400

ResourceAlreadyExistsException

The specified resource already exists.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

CreateDatasetGroup

Service: Amazon Personalize

Creates an empty dataset group. A dataset group contains related datasets that supply data for training a model. A dataset group can contain at most three datasets, one for each type of dataset:

- Interactions
- Items
- Users

To train a model (create a solution), a dataset group containing a dataset of type `Interactions` is required. You call CreateDataset (p. 83) to add a dataset to the group.

A dataset group can be in one of the following states:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -or- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS > DELETE FAILED

To get the status of the dataset group, call DescribeDatasetGroup (p. 137). If the status shows as CREATE FAILED, the response includes a `failureReason` key, which describes why the creation failed.

> **Note**
> You must wait until the `status` of the dataset group is `ACTIVE` before adding a dataset to the group.

You can specify an AWS Key Management Service (KMS) key to encrypt the datasets in the group. If you specify a KMS key, you must also include an AWS Identity and Access Management (IAM) role that has permission to access the key.

APIs that require a dataset group ARN in the request

- CreateDataset (p. 83)
- CreateEventTracker (p. 93)
- CreateFeatureExportJob (p. 96)
- CreateSolution (p. 106)

Related APIs

- ListDatasetGroups (p. 164)
- DescribeDatasetGroup (p. 137)
- DeleteDatasetGroup (p. 117)

Request Syntax

```
{
   "kmsKeyArn": "string",
   "name": "string",
   "roleArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

kmsKeyArn (p. 86)

> The Amazon Resource Name (ARN) of a KMS key used to encrypt the datasets.
>
> Type: String
>
> Required: No

name (p. 86)

> The name for the new dataset group.
>
> Type: String
>
> Length Constraints: Minimum length of 1. Maximum length of 63.
>
> Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`
>
> Required: Yes

roleArn (p. 86)

> The ARN of the IAM role that has permissions to access the KMS key. Supplying an IAM role is only valid when also specifying a KMS key.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`
>
> Required: No

Response Syntax

```
{
    "datasetGroupArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

datasetGroupArn (p. 87)

> The Amazon Resource Name (ARN) of the new dataset group.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errors

InvalidInputException

> Provide a valid value for the field or parameter.

HTTP Status Code: 400

LimitExceededException

The limit on the number of requests per second has been exceeded.

HTTP Status Code: 400

ResourceAlreadyExistsException

The specified resource already exists.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

CreateDatasetImportJob

Service: Amazon Personalize

Creates a job that imports training data from your data source (an Amazon S3 bucket) to an Amazon Personalize dataset. To allow Amazon Personalize to import the training data, you must specify an AWS Identity and Access Management (IAM) role that has permission to read from the data source.

> **Important**
> The dataset import job replaces any previous data in the dataset.

This operation occurs in two steps. First, the import job itself is created. Second, upon successful creation, a run of the import job is initiated.

Status

A dataset import job can be in one of the following states:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -or- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS > DELETE FAILED

To get the *create* status of the import job, call DescribeDatasetImportJob (p. 139) providing the Amazon Resource Name (ARN) of the dataset import job. Once the create status shows as ACTIVE, the `DescribeDatasetImportJob` response includes the `latestDatasetImportJobRun` object and its *run* status. The dataset import is complete when the run status of `latestDatasetImportJobRun` shows as ACTIVE. You can check just the run status by calling DescribeDatasetImportJobRun (p. 141). If the run status shows as CREATE FAILED, the response includes a `failureReason` key, which describes why the job failed.

> **Note**
> Importing takes time. You must wait until the `status` of `latestDatasetImportJobRun` shows as ACTIVE before training a model using the dataset.

Scheduling

To run the import job on a schedule, include the `datasetImportJobConfig` parameter. A new run of the import job is repeated at the frequency defined by its `importJobSchedule` key. You specify the frequency using a cron expression. For more information, see Cron Expressions in the *Amazon CloudWatch User Guide*. To stop the scheduled imports, call UpdateDatasetImportJob (p. 196) and set an empty string for the `importJobSchedule` key. The following is the AWS CLI shorthand syntax for importing the data every day at 12:00pm UTC:

```
--dataset-import-job-config importJobSchedule="cron(0 12 * * ? *)"
```

When `datasetImportJobConfig` is not specified, you must call UpdateDatasetImportJob (p. 196) to create a new run. Each run uses a snapshot of the source data at the time of the call and is given a unique ARN.

Related APIs

- ListDatasetImportJobs (p. 169)
- DescribeDatasetImportJob (p. 139)
- UpdateDatasetImportJob (p. 196)
- DeleteDatasetImportJob (p. 119)

- ListDatasetImportJobRuns (p. 166)
- DescribeDatasetImportJobRun (p. 141)

Request Syntax

```
{
   "datasetArn": "string",
   "datasetImportJobConfig": {
      "importJobSchedule": "string"
   },
   "dataSource": {
      "dataLocation": "string"
   },
   "jobName": "string",
   "roleArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

datasetArn (p. 90)

> The ARN of the dataset that receives the imported data.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: Yes

datasetImportJobConfig (p. 90)

> The dataset import job configuration properties, including the import update frequency.
>
> Type: DatasetImportJobConfig (p. 238) object
>
> Required: No

dataSource (p. 90)

> The Amazon S3 bucket that contains the training data to import.
>
> Type: DataSource (p. 250) object
>
> Required: Yes

jobName (p. 90)

> The user supplied name of the dataset import job.
>
> Type: String
>
> Length Constraints: Minimum length of 1. Maximum length of 63.
>
> Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`
>
> Required: Yes

roleArn (p. 90)

> The ARN of the IAM role that has permissions to read from the Amazon S3 data source.
>
> Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Required: Yes

Response Syntax

```
{
    "datasetImportJobArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

datasetImportJobArn (p. 91)

The ARN of the dataset import job.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

LimitExceededException

The limit on the number of requests per second has been exceeded.

HTTP Status Code: 400

ResourceAlreadyExistsException

The specified resource already exists.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

CreateEventTracker

Service: Amazon Personalize

Creates an event tracker that you use when sending event data to the specified dataset group using the PutEvents (p. 206) API.

When Amazon Personalize creates an event tracker, it also creates an *event-interactions* dataset in the dataset group associated with the event tracker. The event-interactions dataset stores the event data from the `PutEvents` call. The contents of this dataset are not available to the user.

> **Note**
> A dataset group can contain only one event-interactions dataset. You will get an error if you call `CreateEventTracker` using the same dataset group as an existing event tracker, even if you use a different event tracker name.

You must specify an AWS Identity and Access Management (IAM) role that has *CloudWatchFullAccess* permission. The IAM role allows Amazon CloudWatch to calculate the clickthrough rate for the events.

When you send event data you include your tracking ID. The tracking ID identifies the customer and authorizes the customer to send the data.

The event tracker can be in one of the following states:

- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED

To get the status of the event tracker, call DescribeEventTracker (p. 143).

> **Note**
> The event tracker must be in the ACTIVE state before using the tracking ID.

Related APIs

- ListEventTrackers (p. 175)
- DescribeEventTracker (p. 143)
- DeleteEventTracker (p. 121)

Request Syntax

```
{
    "datasetGroupArn": "string",
    "name": "string",
    "roleArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

datasetGroupArn (p. 93)

The Amazon Resource Name (ARN) of the dataset group that receives the event data.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

name (p. 93)

The name for the event tracker.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: Yes

roleArn (p. 93)

The ARN of an IAM role that has *CloudWatchFullAccess* permission.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Required: Yes

Response Syntax

```
{
    "eventTrackerArn": "string",
    "trackingId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

eventTrackerArn (p. 94)

The ARN of the event tracker.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

trackingId (p. 94)

The ID of the event tracker. Include this ID in requests to the PutEvents (p. 206) API.

Type: String

Length Constraints: Maximum length of 256.

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

LimitExceededException

The limit on the number of requests per second has been exceeded.

HTTP Status Code: 400

ResourceAlreadyExistsException

The specified resource already exists.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

CreateFeatureExportJob

Service: Amazon Personalize

Featurizes training data using the information from the specified feature transformation object. The results are stored in an Amazon S3 bucket.

Request Syntax

```
{
    "dataDestination": "string",
    "datasetGroupArn": "string",
    "featureTransformationArn": "string",
    "featureTransformationOverrides": {
        "string" : "string"
    },
    "jobName": "string",
    "roleArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

dataDestination (p. 96)

> The Amazon S3 bucket to store the results.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Required: Yes

datasetGroupArn (p. 96)

> The Amazon Resource Name (ARN) of the dataset group containing the data to featurize.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: Yes

featureTransformationArn (p. 96)

> The ARN of the feature transformation object.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: Yes

featureTransformationOverrides (p. 96)

> Provides values that override the default values in the feature transformation object.
>
> Type: String to string map

Key Length Constraints: Maximum length of 256.

Value Length Constraints: Maximum length of 1000.

Required: No

jobName (p. 96)

The name for the export job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: Yes

roleArn (p. 96)

The ARN of the IAM role that has Amazon S3 permissions.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Required: Yes

Response Syntax

```
{
    "featureExportJobArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

featureExportJobArn (p. 97)

The ARN of the export job.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

LimitExceededException

The limit on the number of requests per second has been exceeded.

HTTP Status Code: 400

ResourceAlreadyExistsException

The specified resource already exists.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

CreateFeatureTransformation

Service: Amazon Personalize

Creates a feature transformation object that contains information on how to featurize the raw input data.

Request Syntax

```
{
   "defaultParameters": {
      "string" : "string"
   },
   "name": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

defaultParameters (p. 99)

Specifies the default parameters for feature transformation.

Type: String to string map

Key Length Constraints: Maximum length of 256.

Value Length Constraints: Maximum length of 1000.

Required: Yes

name (p. 99)

The name for the feature transformation object.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: Yes

Response Syntax

```
{
   "featureTransformationArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

featureTransformationArn (p. 99)

The Amazon Resource Name (ARN) of the FeatureTransformation object.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

LimitExceededException

The limit on the number of requests per second has been exceeded.

HTTP Status Code: 400

ResourceAlreadyExistsException

The specified resource already exists.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

CreateRecipe

Service: Amazon Personalize

Creates a recipe used to train a solution. Call the CreateAlgorithm (p. 77) and CreateFeatureTransformation (p. 99) APIs to obtain the request parameters.

To get a listing of the recipe's properties, call the DescribeRecipe (p. 149) API. To update the recipe, call the UpdateRecipe (p. 201) API.

Request Syntax

```
{
   "algorithmArn": "string",
   "featureTransformationArn": "string",
   "name": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

algorithmArn (p. 101)

The Amazon Resource Name (ARN) of the algorithm used in the recipe.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

featureTransformationArn (p. 101)

The ARN of the feature transformation object to be used in the recipe.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

name (p. 101)

The name of the recipe.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: Yes

Response Syntax

```
{
```

```
    "recipeArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

recipeArn (p. 101)

> The ARN of the recipe.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errors

InvalidInputException

> Provide a valid value for the field or parameter.
>
> HTTP Status Code: 400

LimitExceededException

> The limit on the number of requests per second has been exceeded.
>
> HTTP Status Code: 400

ResourceAlreadyExistsException

> The specified resource already exists.
>
> HTTP Status Code: 400

ResourceNotFoundException

> Could not find the specified resource.
>
> HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python

- [AWS SDK for Ruby V2](#)

CreateSchema

Service: Amazon Personalize

Creates an Amazon Personalize schema from the specified schema string. The schema you create must be in AVRO JSON format.

Amazon Personalize recognizes three schema variants. Each schema is associated with a dataset type and has a set of required field and keywords. You specify a schema when you call CreateDataset (p. 83).

For more information on schemas, see Datasets and Schemas.

Related APIs

- ListSchemas (p. 181)
- DescribeSchema (p. 151)
- DeleteSchema (p. 127)

Request Syntax

```
{
    "name": "string",
    "schema": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

name (p. 104)

> The name for the schema.
>
> Type: String
>
> Length Constraints: Minimum length of 1. Maximum length of 63.
>
> Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`
>
> Required: Yes

schema (p. 104)

> A schema in AVRO JSON format.
>
> Type: String
>
> Length Constraints: Maximum length of 10000.
>
> Required: Yes

Response Syntax

```
{
    "schemaArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

schemaArn (p. 104)

The Amazon Resource Name (ARN) of the created schema.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

LimitExceededException

The limit on the number of requests per second has been exceeded.

HTTP Status Code: 400

ResourceAlreadyExistsException

The specified resource already exists.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

CreateSolution

Service: Amazon Personalize

Creates a solution by training a model. After creating a solution, you check its accuracy by calling GetMetrics (p. 159). When you are satisfied with the solution, you deploy it using CreateCampaign (p. 80). The campaign provides recommendations to a client through the GetRecommendations (p. 208) API.

To train a model, Amazon Personalize requires training data and a recipe. The training data comes from the dataset group that you provide in the request. A recipe specifies the training algorithm and a feature transformation. You can specify one of the predefined recipes provided by Amazon Personalize or provide a custom recipe. Alternatively, you can specify `performAutoML` and Amazon Personalize will analyze your data and select the optimum recipe for you.

You must request the minimum provisioned number of recommendations (transactions) per second that Amazon Personalize will support by using the `minProvisionedTPS` parameter.

Status

A solution can be in one of the following states:

- CREATE PENDING > CREATE IN_PROGRESS > ACTIVE -or- CREATE FAILED
- DELETE PENDING > DELETE IN_PROGRESS > DELETE FAILED

To get the *create* status of the solution, call DescribeSolution (p. 153) providing the Amazon Resource Name (ARN) of the solution. Once the create status shows as CREATE IN_PROGRESS, the `DescribeSolution` response includes the `latestSolutionVersion` object and its *training* status. Training is complete when the training status of the `latestSolutionVersion` shows as ACTIVE. You can check just the training status by calling DescribeSolutionVersion (p. 156). If the training status shows as CREATE FAILED, the response includes a `failureReason` key, which describes why the job failed.

> **Note**
> Before deploying the solution as a campaign, the following conditions must be met:
>
> - The create `status` of the solution is ACTIVE.
> - At least one version of the solution has attained a training `status` of ACTIVE.

Scheduling

To run the training on a schedule, include the `solutionJobConfig` object and set its `retrainingMode` key to `MANUAL`. Training is repeated at the frequency defined by its `retrainSchedule` key. You specify the frequency using a cron expression. For more information, see Cron Expressions in the *Amazon CloudWatch User Guide*. To stop the scheduled trainings, call UpdateSolution (p. 203) and set an empty string for the `retrainSchedule` key.

Related APIs

- ListSolutions (p. 183)
- UpdateSolution (p. 203)
- DescribeSolution (p. 153)
- DeleteSolution (p. 129)

- ListSolutionVersions (p. 186)
- DescribeSolutionVersion (p. 156)

Request Syntax

```
{
   "datasetGroupArn": "string",
   "eventType": "string",
   "minProvisionedTPS": number,
   "name": "string",
   "performAutoML": boolean,
   "performHPO": boolean,
   "recipeArn": "string",
   "solutionConfig": {
      "algorithmHyperParameters": {
         "string" : "string"
      },
      "autoMLConfig": {
         "metricName": "string",
         "recipeList": [ "string" ]
      },
      "eventValueThreshold": "string",
      "featureTransformationParameters": {
         "string" : "string"
      },
      "hpoConfig": {
         "algorithmHyperParameterRanges": {
            "categoricalHyperParameterRanges": [
               {
                  "name": "string",
                  "values": [ "string" ]
               }
            ],
            "continuousHyperParameterRanges": [
               {
                  "maxValue": number,
                  "minValue": number,
                  "name": "string"
               }
            ],
            "integerHyperParameterRanges": [
               {
                  "maxValue": number,
                  "minValue": number,
                  "name": "string"
               }
            ]
         },
         "hpoObjective": {
            "metricName": "string",
            "metricRegex": "string",
            "type": "string"
         },
         "hpoResourceConfig": {
            "maxNumberOfTrainingJobs": "string",
            "maxParallelTrainingJobs": "string"
         }
      },
      "retrainingMode": "string",
      "retrainSchedule": "string"
   }
}
```

Request Parameters

The request accepts the following data in JSON format.

datasetGroupArn (p. 107)

The Amazon Resource Name (ARN) of the dataset group that provides the training data.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

eventType (p. 107)

When your schema includes an `EVENT_TYPE` field, this parameter specifies which type of event (for example, 'click' or 'like') is used as the label for training.

Type: String

Length Constraints: Maximum length of 256.

Required: No

minProvisionedTPS (p. 107)

Specifies the requested minimum provisioned transactions (recommendations) per second that Amazon Personalize will support.

Type: Integer

Required: Yes

name (p. 107)

The name for the new solution. All versions of the solution have a reference to this name.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: Yes

performAutoML (p. 107)

Whether to perform automated machine learning (AutoML). The default is `false`. For this case, you must specify `recipeArn`.

When set to `true`, Amazon Personalize analyzes your training data and selects the optimal recipe for you. In this case, you must omit `recipeArn`. This will lengthen the training process as compared to selecting a specific recipe.

Type: Boolean

Required: No

performHPO (p. 107)

The default is `true`. For this case, Amazon Personalize performs hyperparameter optimization (HPO) on the specified or selected recipe.

To turn off HPO, set this parameter to false (`--no-perform-hpo` when using the AWS CLI). When performing AutoML, this parameter is always `true` and you should not set it to `false`.

Type: Boolean

Required: No

recipeArn (p. 107)

The ARN of the recipe to use for model training. Only specified when `performAutoML` is false.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

solutionConfig (p. 107)

The configuration to use with the solution. When `performAutoML` is set to true, Amazon Personalize only evaluates the `autoMLConfig` section, and `retrainingMode` and `retrainSchedule` parameters of the solution configuration.

Type: SolutionConfig (p. 278) object

Required: No

Response Syntax

```
{
    "solutionArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

solutionArn (p. 109)

The ARN of the solution.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

LimitExceededException

The limit on the number of requests per second has been exceeded.

HTTP Status Code: 400

ResourceAlreadyExistsException

The specified resource already exists.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DeleteAlgorithm

Service: Amazon Personalize

Deletes the given algorithm.

Request Syntax

```
{
    "algorithmArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

algorithmArn (p. 111)

The Amazon Resource Name (ARN) of the algorithm to delete.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceInUseException

The specified resource is in use.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DeleteCampaign

Service: Amazon Personalize

Removes a campaign by deleting the solution deployment. The solution that the campaign is based on is not deleted and can be redeployed when needed. A deleted campaign can no longer be specified in a GetRecommendations (p. 208) request. For more information on campaigns, see CreateCampaign (p. 80).

Request Syntax

```
{
    "campaignArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

campaignArn (p. 113)

> The Amazon Resource Name (ARN) of the campaign to delete.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

InvalidInputException

> Provide a valid value for the field or parameter.
>
> HTTP Status Code: 400

ResourceInUseException

> The specified resource is in use.
>
> HTTP Status Code: 400

ResourceNotFoundException

> Could not find the specified resource.
>
> HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DeleteDataset

Service: Amazon Personalize

Deletes the given dataset. For more information on datasets, see CreateDataset (p. 83).

Request Syntax

```
{
    "datasetArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

datasetArn (p. 115)

The Amazon Resource Name (ARN) of the dataset to delete.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceInUseException

The specified resource is in use.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DeleteDatasetGroup

Service: Amazon Personalize

Deletes the given dataset group. Before you delete a dataset group, you must delete all datasets belonging to the group. To find all datasets that belong to the group, call and provide the Amazon Resource Name (ARN) for the dataset group you want to delete.

When you try to delete a non-empty dataset group, an `InvalidInputException` occurs, which lists the datasets that need to be deleted. If you try to delete a dataset group immediately after deleting the group's datasets, you might still get the `InvalidInputException` error. It takes a moment to delete the datasets. If you do get the error, try the call again. For more information on dataset groups, see .

Request Syntax

```
{
    "datasetGroupArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

datasetGroupArn

> The ARN of the dataset group to delete.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

InvalidInputException

> Provide a valid value for the field or parameter.
>
> HTTP Status Code: 400

ResourceInUseException

> The specified resource is in use.
>
> HTTP Status Code: 400

ResourceNotFoundException

> Could not find the specified resource.
>
> HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DeleteDatasetImportJob

Service: Amazon Personalize

Deletes the dataset import job. Your campaigns use a cached copy of the data and continue to function. These cached copies are deleted only when you call DeleteSolution (p. 129). For more information on dataset import jobs, see CreateDatasetImportJob (p. 89).

Request Syntax

```
{
    "datasetImportJobArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

datasetImportJobArn (p. 119)

The Amazon Resource Name (ARN) of the import job to delete.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceInUseException

The specified resource is in use.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DeleteEventTracker

Service: Amazon Personalize

Deletes the given event tracker. Does not delete the clickstream dataset from the associated dataset group. For more information on event trackers, see CreateEventTracker (p. 93).

Request Syntax

```
{
    "eventTrackerArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

eventTrackerArn (p. 121)

The Amazon Resource Name (ARN) of the event tracker to delete.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceInUseException

The specified resource is in use.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DeleteFeatureTransformation

Service: Amazon Personalize

Deletes the given feature transformation.

Request Syntax

```
{
    "featureTransformationArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

featureTransformationArn (p. 123)

The Amazon Resource Name (ARN) of the feature transformation to delete.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceInUseException

The specified resource is in use.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DeleteRecipe

Service: Amazon Personalize

Deletes the given recipe.

Request Syntax

```
{
    "recipeArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

recipeArn (p. 125)

The Amazon Resource Name (ARN) of the recipe.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceInUseException

The specified resource is in use.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DeleteSchema

Service: Amazon Personalize

Deletes the given schema. For more information on schemas, see CreateSchema (p. 104).

Request Syntax

```
{
    "schemaArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

schemaArn (p. 127)

The Amazon Resource Name (ARN) of the schema to delete.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceInUseException

The specified resource is in use.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DeleteSolution

Service: Amazon Personalize

Deletes a solution. A solution cannot be deleted if it is part of a campaign. To determine what campaigns are using the solution, call ListCampaigns (p. 161) and supply the Amazon Resource Name (ARN) of the solution. For more information on solutions, see CreateSolution (p. 106).

Request Syntax

```
{
    "solutionArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

solutionArn (p. 129)

> The ARN of the solution to delete.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

InvalidInputException

> Provide a valid value for the field or parameter.
>
> HTTP Status Code: 400

ResourceInUseException

> The specified resource is in use.
>
> HTTP Status Code: 400

ResourceNotFoundException

> Could not find the specified resource.
>
> HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DescribeAlgorithm

Service: Amazon Personalize

Describes the given algorithm.

Request Syntax

```
{
    "algorithmArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

algorithmArn (p. 131)

The Amazon Resource Name (ARN) of the algorithm to describe.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

Response Syntax

```
{
    "algorithm": {
        "algorithmArn": "string",
        "algorithmImage": {
            "dockerURI": "string",
            "name": "string"
        },
        "creationDateTime": number,
        "defaultHyperParameterRanges": {
            "categoricalHyperParameterRanges": [
                {
                    "isTunable": boolean,
                    "name": "string",
                    "values": [ "string" ]
                }
            ],
            "continuousHyperParameterRanges": [
                {
                    "isTunable": boolean,
                    "maxValue": number,
                    "minValue": number,
                    "name": "string"
                }
            ],
            "integerHyperParameterRanges": [
                {
                    "isTunable": boolean,
                    "maxValue": number,
                    "minValue": number,
                    "name": "string"
                }
```

```
        ]
    },
    "defaultHyperParameters": {
        "string" : "string"
    },
    "defaultResourceConfig": {
        "string" : "string"
    },
    "lastUpdatedDateTime": number,
    "name": "string",
    "roleArn": "string",
    "trainingInputMode": "string"
  }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

algorithm (p. 131)

> A listing of the properties of the algorithm.
>
> Type: Algorithm (p. 216) object

Errors

InvalidInputException

> Provide a valid value for the field or parameter.
>
> HTTP Status Code: 400

ResourceNotFoundException

> Could not find the specified resource.
>
> HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DescribeCampaign

Service: Amazon Personalize

Describes the given campaign, including its status.

A campaign can be in one of the following states:

- CREATE PENDING => CREATE IN_PROGRESS => ACTIVE -or- CREATE FAILED
- DELETE PENDING => DELETE IN_PROGRESS => DELETE FAILED

When the `status` is `CREATE FAILED`, the response includes the `failureReason` key, which describes why.

For more information on campaigns, see .

Request Syntax

```
{
    "campaignArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

campaignArn (p. 133)

The Amazon Resource Name (ARN) of the campaign.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

Response Syntax

```
{
    "campaign": {
        "campaignArn": "string",
        "creationDateTime": number,
        "failureReason": "string",
        "lastUpdatedDateTime": number,
        "latestCampaignUpdate": {
            "creationDateTime": number,
            "failureReason": "string",
            "lastUpdatedDateTime": number,
            "solutionVersionArn": "string",
            "status": "string"
        },
        "name": "string",
        "solutionArn": "string",
        "solutionVersionArn": "string",
        "status": "string",
        "updateMode": "string"
    }
```

```
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

campaign (p. 133)

> The properties of the campaign.
>
> Type: Campaign (p. 221) object

Errors

InvalidInputException

> Provide a valid value for the field or parameter.
>
> HTTP Status Code: 400

ResourceNotFoundException

> Could not find the specified resource.
>
> HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DescribeDataset

Service: Amazon Personalize

Describes the given dataset. For more information on datasets, see CreateDataset (p. 83).

Request Syntax

```
{
    "datasetArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

datasetArn (p. 135)

The Amazon Resource Name (ARN) of the dataset to describe.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

Response Syntax

```
{
    "dataset": {
        "creationDateTime": number,
        "datasetArn": "string",
        "datasetGroupArn": "string",
        "datasetType": "string",
        "lastUpdatedDateTime": number,
        "name": "string",
        "schemaArn": "string",
        "status": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

dataset (p. 135)

A listing of the dataset's properties.

Type: Dataset (p. 230) object

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DescribeDatasetGroup

Service: Amazon Personalize

Describes the given dataset group. For more information on dataset groups, see
CreateDatasetGroup (p. 86).

Request Syntax

```
{
    "datasetGroupArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

datasetGroupArn (p. 137)

> The Amazon Resource Name (ARN) of the dataset group to describe.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: Yes

Response Syntax

```
{
    "datasetGroup": {
        "creationDateTime": number,
        "datasetGroupArn": "string",
        "failureReason": "string",
        "kmsKeyArn": "string",
        "lastUpdatedDateTime": number,
        "name": "string",
        "roleArn": "string",
        "status": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

datasetGroup (p. 137)

> A listing of the dataset group's properties.
>
> Type: DatasetGroup (p. 232) object

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DescribeDatasetImportJob

Service: Amazon Personalize

Describes the dataset import job created by CreateDatasetImportJob (p. 89).

Call this API to determine the import job's create status. When the import job is successfully created, the properties and run status of the most current run of the import job are also described. To get information about a specific run, call DescribeDatasetImportJobRun (p. 141).

Request Syntax

```
{
    "datasetImportJobArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

datasetImportJobArn (p. 139)

> The Amazon Resource Name (ARN) of the dataset import job to describe.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: Yes

Response Syntax

```
{
    "datasetImportJob": {
        "creationDateTime": number,
        "datasetArn": "string",
        "datasetImportJobArn": "string",
        "datasetImportJobConfig": {
            "importJobSchedule": "string"
        },
        "dataSource": {
            "dataLocation": "string"
        },
        "jobName": "string",
        "lastUpdatedDateTime": number,
        "latestDatasetImportJobRun": {
            "creationDateTime": number,
            "datasetImportJobRunArn": "string",
            "failureReason": "string",
            "lastUpdatedDateTime": number,
            "status": "string"
        },
        "roleArn": "string",
        "status": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

datasetImportJob (p. 139)

Information about the dataset import job, including the create status.

The create status is one of the following values:
- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED

Type: DatasetImportJob (p. 236) object

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DescribeDatasetImportJobRun

Service: Amazon Personalize

Describes a specific run of a dataset import job.

A new run of a dataset import job is automatically started at a frequency defined by the dataset import job or manually with a call to UpdateDatasetImportJob (p. 196). Each run uses a snapshot of the source data at the time of the call and is given a unique Amazon Resource Name (ARN). For more information on dataset import jobs, see CreateDatasetImportJob (p. 89).

Related APIs

- ListDatasetImportJobRuns (p. 166)

Request Syntax

```
{
    "datasetImportJobRunArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

datasetImportJobRunArn (p. 141)

> The Amazon Resource Name (ARN) of the dataset import job to describe.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: Yes

Response Syntax

```
{
    "datasetImportJobRun": {
        "creationDateTime": number,
        "datasetArn": "string",
        "datasetImportJobArn": "string",
        "datasetImportJobConfig": {
            "importJobSchedule": "string"
        },
        "datasetImportJobRunArn": "string",
        "dataSource": {
            "dataLocation": "string"
        },
        "failureReason": "string",
        "lastUpdatedDateTime": number,
        "roleArn": "string",
        "status": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

datasetImportJobRun (p. 141)

Information about the dataset import job run, including the run status. Only when the run `status` shows as ACTIVE, can the data be used to train a solution.

The run status is one of the following values:
- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED

Type: DatasetImportJobRun (p. 239) object

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DescribeEventTracker

Service: Amazon Personalize

Describes an event tracker.

Call this API to ensure that the `status` of your event tracker is `ACTIVE`. For more information on event trackers, see CreateEventTracker (p. 93).

Request Syntax

```
{
    "eventTrackerArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

eventTrackerArn (p. 143)

The Amazon Resource Name (ARN) of the event tracker to describe.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

Response Syntax

```
{
    "eventTracker": {
        "accountId": "string",
        "creationDateTime": number,
        "datasetGroupArn": "string",
        "eventTrackerArn": "string",
        "lastUpdatedDateTime": number,
        "name": "string",
        "roleArn": "string",
        "status": "string",
        "trackingId": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

eventTracker (p. 143)

An object that describes the event tracker.

Type: EventTracker (p. 255) object

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DescribeFeatureExportJob

Service: Amazon Personalize

Describes the feature export job.

Request Syntax

```
{
   "featureExportJobArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

featureExportJobArn (p. 145)

>   The Amazon Resource Name (ARN) of the export job.
>
>   Type: String
>
>   Length Constraints: Maximum length of 256.
>
>   Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
>   Required: Yes

Response Syntax

```
{
   "featureExportJob": {
      "creationDateTime": number,
      "dataDestination": "string",
      "datasetGroupArn": "string",
      "failureReason": "string",
      "featureExportJobArn": "string",
      "featureTransformationArn": "string",
      "featureTransformationOverrides": {
         "string" : "string"
      },
      "jobName": "string",
      "lastUpdatedDateTime": number,
      "roleArn": "string",
      "status": "string"
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

featureExportJob (p. 145)

>   A listing of the FeatureExportJob properties.
>
>   Type: FeatureExportJob (p. 259) object

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DescribeFeatureTransformation

Service: Amazon Personalize

Describes the given feature transformation.

Request Syntax

```
{
    "featureTransformationArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

featureTransformationArn (p. 147)

The Amazon Resource Name (ARN) of the feature transformation to describe.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

Response Syntax

```
{
    "featureTransformation": {
        "creationDateTime": number,
        "defaultParameters": {
            "string" : "string"
        },
        "featureTransformationArn": "string",
        "lastUpdatedDateTime": number,
        "name": "string",
        "status": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

featureTransformation (p. 147)

A listing of the FeatureTransformation properties.

Type: FeatureTransformation (p. 264) object

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DescribeRecipe

Service: Amazon Personalize

Describes a recipe created by the CreateRecipe (p. 101) API.

A recipe contains an algorithm that trains a model, hyperparameters that govern the training, and feature transformation information for modifying the input data before training.

Amazon Personalize provides a set of predefined recipes. You specify a recipe when you create a solution using the CreateSolution (p. 106) API. `CreateSolution` trains a model using the algorithm in the specified recipe and a training dataset. The solution, when deployed as a campaign, can provide recommendations using the GetRecommendations (p. 208) API.

Request Syntax

```
{
    "recipeArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

recipeArn (p. 149)

   The Amazon Resource Name (ARN) of the recipe to describe.

   Type: String

   Length Constraints: Maximum length of 256.

   Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

   Required: Yes

Response Syntax

```
{
    "recipe": {
        "algorithmArn": "string",
        "creationDateTime": number,
        "description": "string",
        "featureTransformationArn": "string",
        "lastUpdatedDateTime": number,
        "name": "string",
        "recipeArn": "string",
        "recipeType": "string",
        "status": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

recipe (p. 149)

An object that describes the recipe.

Type: Recipe (p. 271) object

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DescribeSchema

Service: Amazon Personalize

Describes a schema. For more information on schemas, see CreateSchema (p. 104).

Request Syntax

```
{
    "schemaArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

schemaArn (p. 151)

> The Amazon Resource Name (ARN) of the schema to retrieve.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: Yes

Response Syntax

```
{
    "schema": {
        "creationDateTime": number,
        "lastUpdatedDateTime": number,
        "name": "string",
        "schema": "string",
        "schemaArn": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

schema (p. 151)

> The requested schema.
>
> Type: DatasetSchema (p. 245) object

Errors

InvalidInputException

> Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DescribeSolution

Service: Amazon Personalize

Describes a solution. For more information on solutions, see CreateSolution (p. 106).

Request Syntax

```
{
    "solutionArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

solutionArn (p. 153)

The Amazon Resource Name (ARN) of the solution to describe.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

Response Syntax

```
{
    "solution": {
        "autoMLResult": {
            "bestRecipeArn": "string"
        },
        "creationDateTime": number,
        "datasetGroupArn": "string",
        "eventType": "string",
        "lastUpdatedDateTime": number,
        "latestSolutionVersion": {
            "creationDateTime": number,
            "failureReason": "string",
            "lastUpdatedDateTime": number,
            "solutionVersionArn": "string",
            "status": "string"
        },
        "minProvisionedTPS": number,
        "name": "string",
        "performAutoML": boolean,
        "performHPO": boolean,
        "recipeArn": "string",
        "solutionArn": "string",
        "solutionConfig": {
            "algorithmHyperParameters": {
                "string" : "string"
            },
            "autoMLConfig": {
                "metricName": "string",
                "recipeList": [ "string" ]
            },
            "eventValueThreshold": "string",
```

```
            "featureTransformationParameters": {
                "string" : "string"
            },
            "hpoConfig": {
                "algorithmHyperParameterRanges": {
                    "categoricalHyperParameterRanges": [
                        {
                            "name": "string",
                            "values": [ "string" ]
                        }
                    ],
                    "continuousHyperParameterRanges": [
                        {
                            "maxValue": number,
                            "minValue": number,
                            "name": "string"
                        }
                    ],
                    "integerHyperParameterRanges": [
                        {
                            "maxValue": number,
                            "minValue": number,
                            "name": "string"
                        }
                    ]
                },
                "hpoObjective": {
                    "metricName": "string",
                    "metricRegex": "string",
                    "type": "string"
                },
                "hpoResourceConfig": {
                    "maxNumberOfTrainingJobs": "string",
                    "maxParallelTrainingJobs": "string"
                }
            },
            "retrainingMode": "string",
            "retrainSchedule": "string"
        },
        "status": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

solution (p. 153)

An object that describes the solution.

Type: Solution (p. 275) object

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DescribeSolutionVersion

Service: Amazon Personalize

Describes a specific version of a solution. For more information on solutions, see CreateSolution (p. 106).

Request Syntax

```
{
    "solutionVersionArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

solutionVersionArn (p. 156)

    The Amazon Resource Name (ARN) of the solution version.

    Type: String

    Length Constraints: Maximum length of 256.

    Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

    Required: Yes

Response Syntax

```
{
    "solutionVersion": {
        "creationDateTime": number,
        "datasetGroupArn": "string",
        "eventType": "string",
        "failureReason": "string",
        "lastUpdatedDateTime": number,
        "performAutoML": boolean,
        "performHPO": boolean,
        "recipeArn": "string",
        "solutionArn": "string",
        "solutionConfig": {
            "algorithmHyperParameters": {
                "string" : "string"
            },
            "autoMLConfig": {
                "metricName": "string",
                "recipeList": [ "string" ]
            },
            "eventValueThreshold": "string",
            "featureTransformationParameters": {
                "string" : "string"
            },
            "hpoConfig": {
                "algorithmHyperParameterRanges": {
                    "categoricalHyperParameterRanges": [
                        {
                            "name": "string",
                            "values": [ "string" ]
                        }
                    ],
```

```
                "continuousHyperParameterRanges": [
                    {
                        "maxValue": number,
                        "minValue": number,
                        "name": "string"
                    }
                ],
                "integerHyperParameterRanges": [
                    {
                        "maxValue": number,
                        "minValue": number,
                        "name": "string"
                    }
                ]
            },
            "hpoObjective": {
                "metricName": "string",
                "metricRegex": "string",
                "type": "string"
            },
            "hpoResourceConfig": {
                "maxNumberOfTrainingJobs": "string",
                "maxParallelTrainingJobs": "string"
            }
        },
        "retrainingMode": "string",
        "retrainSchedule": "string"
    },
    "solutionVersionArn": "string",
    "status": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

solutionVersion (p. 156)

> The solution version.

> Type: SolutionVersion (p. 282) object

Errors

InvalidInputException

> Provide a valid value for the field or parameter.

> HTTP Status Code: 400

ResourceNotFoundException

> Could not find the specified resource.

> HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

GetMetrics

Service: Amazon Personalize

Gets the metrics for the specified solution or solution version. When the solution ARN is specified and the solution version ARN isn't specified, the metrics for the latest solution version are returned.

Request Syntax

```
{
    "solutionArn": "string",
    "solutionVersionArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

solutionArn (p. 159)

> The Amazon Resource Name (ARN) of the solution for which to get metrics.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: Yes

solutionVersionArn (p. 159)

> The ARN of the solution version for which to get metrics.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: No

Response Syntax

```
{
    "metrics": {
        "string" : {
            "string" : number
        }
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

metrics (p. 159)

> The metrics for the solution versions.

Type: String to string to double map map

Key Length Constraints: Maximum length of 256.

Key Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Key Length Constraints: Maximum length of 256.

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

ListCampaigns

Service: Amazon Personalize

Returns a list of campaigns that use the given solution. When a solution is not specified, all the campaigns associated with the account are listed. The response provides the properties for each campaign, including the Amazon Resource Name (ARN). For more information on campaigns, see CreateCampaign (p. 80).

Request Syntax

```
{
    "maxResults": number,
    "nextToken": "string",
    "solutionArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

maxResults (p. 161)

The maximum number of campaigns to return.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

nextToken (p. 161)

A token returned from the previous call to `ListCampaigns` for getting the next set of campaigns (if they exist).

Type: String

Length Constraints: Maximum length of 1300.

Required: No

solutionArn (p. 161)

The Amazon Resource Name (ARN) of the solution to list the campaigns for. When a solution is not specified, all the campaigns associated with the account are listed.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

Response Syntax

```
{
    "campaigns": [
        {
            "campaignArn": "string",
```

```
        "creationDateTime": number,
        "failureReason": "string",
        "lastUpdatedDateTime": number,
        "name": "string",
        "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

campaigns (p. 161)

A list of the campaigns.

Type: Array of CampaignSummary (p. 224) objects

Array Members: Maximum number of 100 items.

nextToken (p. 161)

A token for getting the next set of campaigns (if they exist).

Type: String

Length Constraints: Maximum length of 1300.

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

InvalidNextTokenException

The token is not valid.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3

- AWS SDK for Python
- AWS SDK for Ruby V2

ListDatasetGroups

Service: Amazon Personalize

Returns a list of dataset groups. The response provides the properties for each dataset group, including the Amazon Resource Name (ARN). For more information on dataset groups, see CreateDatasetGroup (p. 86).

Request Syntax

```
{
    "maxResults": number,
    "nextToken": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

maxResults (p. 164)

> The maximum number of dataset groups to return.
>
> Type: Integer
>
> Valid Range: Minimum value of 1. Maximum value of 100.
>
> Required: No

nextToken (p. 164)

> A token returned from the previous call to `ListDatasetGroups` for getting the next set of dataset groups (if they exist).
>
> Type: String
>
> Length Constraints: Maximum length of 1300.
>
> Required: No

Response Syntax

```
{
    "datasetGroups": [
        {
            "creationDateTime": number,
            "datasetGroupArn": "string",
            "failureReason": "string",
            "lastUpdatedDateTime": number,
            "name": "string",
            "status": "string"
        }
    ],
    "nextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

datasetGroups (p. 164)

The list of your dataset groups.

Type: Array of DatasetGroupSummary (p. 234) objects

Array Members: Maximum number of 100 items.

nextToken (p. 164)

A token for getting the next set of dataset groups (if they exist).

Type: String

Length Constraints: Maximum length of 1300.

Errors

InvalidNextTokenException

The token is not valid.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

ListDatasetImportJobRuns

Service: Amazon Personalize

Returns a list of import job runs for the given dataset import job. When a dataset import job is not specified, all the dataset import jobs associated with the account are listed. The response provides the properties for each dataset import job run, including the Amazon Resource Name (ARN). A dataset import job can be run on a schedule, or manually using UpdateDatasetImportJob (p. 196). Each time the import job is run, a new dataset snapshot is created and given its own Amazon Resource Name (ARN). For more information on dataset import jobs, see CreateDatasetImportJob (p. 89).

Related APIs

- DescribeDatasetImportJobRun (p. 141)

Request Syntax

```
{
   "datasetImportJobArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

datasetImportJobArn (p. 166)

>   The Amazon Resource Name (ARN) of the dataset import job to list the dataset import jobs runs for.

>   Type: String

>   Length Constraints: Maximum length of 256.

>   Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

>   Required: No

maxResults (p. 166)

>   The maximum number of results to return.

>   Type: Integer

>   Valid Range: Minimum value of 1. Maximum value of 100.

>   Required: No

nextToken (p. 166)

>   A token returned from the previous call to `ListDatasetImportJobRuns` for getting the next set of dataset import job runs (if they exist).

>   Type: String

>   Length Constraints: Maximum length of 1300.

>   Required: No

Response Syntax

```
{
   "datasetImportJobRuns": [
      {
         "creationDateTime": number,
         "datasetImportJobRunArn": "string",
         "failureReason": "string",
         "lastUpdatedDateTime": number,
         "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

datasetImportJobRuns (p. 167)

The list of dataset import job runs.

Type: Array of DatasetImportJobRunSummary (p. 241) objects

Array Members: Maximum number of 100 items.

nextToken (p. 167)

A token for getting the next set of import job runs (if they exist).

Type: String

Length Constraints: Maximum length of 1300.

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

InvalidNextTokenException

The token is not valid.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

ListDatasetImportJobs

Service: Amazon Personalize

Returns a list of dataset import jobs that use the given dataset. When a dataset is not specified, all the dataset import jobs associated with the account are listed. The response provides the properties for each dataset import job, including the Amazon Resource Name (ARN). For more information on dataset import jobs, see CreateDatasetImportJob (p. 89). For more information on datasets, see CreateDataset (p. 83).

Request Syntax

```
{
    "datasetArn": "string",
    "maxResults": number,
    "nextToken": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

datasetArn (p. 169)

> The Amazon Resource Name (ARN) of the dataset to list the dataset import jobs for.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: No

maxResults (p. 169)

> The maximum number of dataset import jobs to return.
>
> Type: Integer
>
> Valid Range: Minimum value of 1. Maximum value of 100.
>
> Required: No

nextToken (p. 169)

> A token returned from the previous call to `ListDatasetImportJobs` for getting the next set of dataset import jobs (if they exist).
>
> Type: String
>
> Length Constraints: Maximum length of 1300.
>
> Required: No

Response Syntax

```
{
    "datasetImportJobs": [
        {
            "creationDateTime": number,
```

```
        "datasetImportJobArn": "string",
        "jobName": "string",
        "lastUpdatedDateTime": number,
        "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

datasetImportJobs (p. 169)

The list of dataset import jobs.

Type: Array of DatasetImportJobSummary (p. 243) objects

Array Members: Maximum number of 100 items.

nextToken (p. 169)

A token for getting the next set of dataset import jobs (if they exist).

Type: String

Length Constraints: Maximum length of 1300.

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

InvalidNextTokenException

The token is not valid.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot

- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

ListDatasets

Service: Amazon Personalize

Returns the list of datasets contained in the given dataset group. The response provides the properties for each dataset, including the Amazon Resource Name (ARN). For more information on datasets, see CreateDataset (p. 83).

Request Syntax

```
{
    "datasetGroupArn": "string",
    "maxResults": number,
    "nextToken": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

datasetGroupArn (p. 172)

> The Amazon Resource Name (ARN) of the dataset group that contains the datasets to list.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: No

maxResults (p. 172)

> The maximum number of datasets to return.
>
> Type: Integer
>
> Valid Range: Minimum value of 1. Maximum value of 100.
>
> Required: No

nextToken (p. 172)

> A token returned from the previous call to `ListDatasetImportJobs` for getting the next set of dataset import jobs (if they exist).
>
> Type: String
>
> Length Constraints: Maximum length of 1300.
>
> Required: No

Response Syntax

```
{
    "datasets": [
        {
            "creationDateTime": number,
            "datasetArn": "string",
            "datasetType": "string",
```

```
        "lastUpdatedDateTime": number,
        "name": "string",
        "status": "string"
      }
   ],
   "nextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

datasets (p. 172)

An array of `Dataset` objects. Each object provides metadata information.

Type: Array of DatasetSummary (p. 248) objects

Array Members: Maximum number of 100 items.

nextToken (p. 172)

A token for getting the next set of datasets (if they exist).

Type: String

Length Constraints: Maximum length of 1300.

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

InvalidNextTokenException

The token is not valid.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java

- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

ListEventTrackers

Service: Amazon Personalize

Returns the list of event trackers associated with the account. The response provides the properties for each event tracker, including the Amazon Resource Name (ARN) and tracking ID. For more information on event trackers, see CreateEventTracker (p. 93).

Request Syntax

```
{
   "datasetGroupArn": "string",
   "maxResults": number,
   "nextToken": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

datasetGroupArn (p. 175)

> The ARN of a dataset group used to filter the response.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: No

maxResults (p. 175)

> The maximum number of event trackers to return.
>
> Type: Integer
>
> Valid Range: Minimum value of 1. Maximum value of 100.
>
> Required: No

nextToken (p. 175)

> A token returned from the previous call to `ListEventTrackers` for getting the next set of event trackers (if they exist).
>
> Type: String
>
> Length Constraints: Maximum length of 1300.
>
> Required: No

Response Syntax

```
{
   "eventTrackers": [
      {
         "creationDateTime": number,
         "eventTrackerArn": "string",
         "lastUpdatedDateTime": number,
```

```
            "name": "string",
            "status": "string"
        }
    ],
    "nextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

eventTrackers (p. 175)

A list of event trackers.

Type: Array of EventTrackerSummary (p. 257) objects

Array Members: Maximum number of 100 items.

nextToken (p. 175)

A token for getting the next set of event trackers (if they exist).

Type: String

Length Constraints: Maximum length of 1300.

Errors

InvalidNextTokenException

The token is not valid.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

ListFeatureExportJobs

Service: Amazon Personalize

Returns the list of available feature export jobs. The response provides the properties for each feature export job, including the Amazon Resource Name (ARN).

Request Syntax

```
{
    "maxResults": number,
    "nextToken": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

maxResults (p. 177)

The maximum number of feature export jobs to return.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

nextToken (p. 177)

A token returned from the previous call to `ListFeatureExportJobs` for getting the next set of feature export jobs (if they exist).

Type: String

Length Constraints: Maximum length of 1300.

Required: No

Response Syntax

```
{
    "featureExportJobs": [
        {
            "creationDateTime": number,
            "failureReason": "string",
            "featureExportJobArn": "string",
            "jobName": "string",
            "lastUpdatedDateTime": number,
            "status": "string"
        }
    ],
    "nextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

featureExportJobs (p. 177)

A list of feature export jobs.

Type: Array of FeatureExportJobSummary (p. 262) objects

Array Members: Maximum number of 100 items.

nextToken (p. 177)

A token for getting the next set of feature export jobs (if they exist).

Type: String

Length Constraints: Maximum length of 1300.

Errors

InvalidNextTokenException

The token is not valid.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

ListRecipes

Service: Amazon Personalize

Returns a list of available recipes. The response provides the properties for each recipe, including the recipe's Amazon Resource Name (ARN).

The `recipeProvider` determines which recipes are included in the list. Valid values are:

- Service - only predefined Amazon Personalize recipes
- BYOR - only Bring Your Own Recipes (customer defined)
- All (default) - both BYOR and Service recipes, with BYORs listed first

Request Syntax

```
{
    "maxResults": number,
    "nextToken": "string",
    "recipeProvider": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

maxResults (p. 179)

> The maximum number of recipes to return.
>
> Type: Integer
>
> Valid Range: Minimum value of 1. Maximum value of 100.
>
> Required: No

nextToken (p. 179)

> A token returned from the previous call to `ListRecipes` for getting the next set of recipes (if they exist).
>
> Type: String
>
> Length Constraints: Maximum length of 1300.
>
> Required: No

recipeProvider (p. 179)

> The default is `All`.
>
> Type: String
>
> Valid Values: `Service | BYOR | All`
>
> Required: No

Response Syntax

```
{
    "nextToken": "string",
```

```
    "recipes": [
      {
        "creationDateTime": number,
        "lastUpdatedDateTime": number,
        "name": "string",
        "recipeArn": "string",
        "status": "string"
      }
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

nextToken (p. 179)

A token for getting the next set of recipes.

Type: String

Length Constraints: Maximum length of 1300.

recipes (p. 179)

The list of available recipes.

Type: Array of RecipeSummary (p. 273) objects

Array Members: Maximum number of 100 items.

Errors

InvalidNextTokenException

The token is not valid.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

ListSchemas

Service: Amazon Personalize

Returns the list of schemas associated with the account. The response provides the properties for each schema, including the Amazon Resource Name (ARN). For more information on schemas, see CreateSchema (p. 104).

Request Syntax

```
{
    "maxResults": number,
    "nextToken": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

maxResults (p. 181)

The maximum number of schemas to return.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

nextToken (p. 181)

A token returned from the previous call to `ListSchemas` for getting the next set of schemas (if they exist).

Type: String

Length Constraints: Maximum length of 1300.

Required: No

Response Syntax

```
{
    "nextToken": "string",
    "schemas": [
        {
            "creationDateTime": number,
            "lastUpdatedDateTime": number,
            "name": "string",
            "schemaArn": "string"
        }
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

nextToken (p. 181)

A token used to get the next set of schemas (if they exist).

Type: String

Length Constraints: Maximum length of 1300.

schemas (p. 181)

A list of schemas.

Type: Array of DatasetSchemaSummary (p. 247) objects

Array Members: Maximum number of 100 items.

Errors

InvalidNextTokenException

The token is not valid.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

ListSolutions

Service: Amazon Personalize

Returns a list of solutions that use the given dataset group. When a dataset group is not specified, all the solutions associated with the account are listed. The response provides the properties for each solution, including the Amazon Resource Name (ARN). For more information on solutions, see CreateSolution (p. 106).

Request Syntax

```
{
    "datasetGroupArn": "string",
    "maxResults": number,
    "nextToken": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

datasetGroupArn (p. 183)

   The Amazon Resource Name (ARN) of the dataset group.

   Type: String

   Length Constraints: Maximum length of 256.

   Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

   Required: No

maxResults (p. 183)

   The maximum number of solutions to return.

   Type: Integer

   Valid Range: Minimum value of 1. Maximum value of 100.

   Required: No

nextToken (p. 183)

   A token returned from the previous call to `ListSolutions` for getting the next set of solutions (if they exist).

   Type: String

   Length Constraints: Maximum length of 1300.

   Required: No

Response Syntax

```
{
    "nextToken": "string",
    "solutions": [
        {
```

```
            "creationDateTime": number,
            "lastUpdatedDateTime": number,
            "name": "string",
            "solutionArn": "string",
            "status": "string"
        }
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

nextToken (p. 183)

A token for getting the next set of solutions (if they exist).

Type: String

Length Constraints: Maximum length of 1300.

solutions (p. 183)

A list of the current solutions.

Type: Array of SolutionSummary (p. 280) objects

Array Members: Maximum number of 100 items.

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

InvalidNextTokenException

The token is not valid.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python

- [AWS SDK for Ruby V2](#)

ListSolutionVersions

Service: Amazon Personalize

Returns a list of solution versions for the given solution. When a solution is not specified, all the solution versions associated with the account are listed. The response provides the properties for each solution version, including the Amazon Resource Name (ARN). For more information on solutions, see CreateSolution (p. 106).

Request Syntax

```
{
    "maxResults": number,
    "nextToken": "string",
    "solutionArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

maxResults (p. 186)

> The maximum number of solution versions to return.
>
> Type: Integer
>
> Valid Range: Minimum value of 1. Maximum value of 100.
>
> Required: No

nextToken (p. 186)

> A token returned from the previous call to `ListSolutionVersions` for getting the next set of solution versions (if they exist).
>
> Type: String
>
> Length Constraints: Maximum length of 1300.
>
> Required: No

solutionArn (p. 186)

> The Amazon Resource Name (ARN) of the solution.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: No

Response Syntax

```
{
    "nextToken": "string",
    "solutionVersions": [
        {
```

```
            "creationDateTime": number,
            "failureReason": "string",
            "lastUpdatedDateTime": number,
            "solutionVersionArn": "string",
            "status": "string"
        }
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

nextToken (p. 186)

A token for getting the next set of solution versions (if they exist).

Type: String

Length Constraints: Maximum length of 1300.

solutionVersions (p. 186)

A list of solution versions describing the version properties.

Type: Array of SolutionVersionSummary (p. 285) objects

Array Members: Maximum number of 100 items.

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python

- AWS SDK for Ruby V2

UpdateAlgorithm

Service: Amazon Personalize

Updates the given algorithm. Only the algorithm ARN is required.

Request Syntax

```
{
   "algorithmArn": "string",
   "algorithmImage": {
      "dockerURI": "string",
      "name": "string"
   },
   "defaultHyperParameterRanges": {
      "categoricalHyperParameterRanges": [
         {
            "isTunable": boolean,
            "name": "string",
            "values": [ "string" ]
         }
      ],
      "continuousHyperParameterRanges": [
         {
            "isTunable": boolean,
            "maxValue": number,
            "minValue": number,
            "name": "string"
         }
      ],
      "integerHyperParameterRanges": [
         {
            "isTunable": boolean,
            "maxValue": number,
            "minValue": number,
            "name": "string"
         }
      ]
   },
   "defaultHyperParameters": {
      "string" : "string"
   },
   "defaultResourceConfig": {
      "string" : "string"
   },
   "trainingInputMode": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

algorithmArn (p. 189)

The Amazon Resource Name (ARN) of the algorithm to update.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

algorithmImage (p. 189)

Specifies a new URI for the Docker container for the algorithm image.

Type: AlgorithmImage (p. 218) object

Required: No

defaultHyperParameterRanges (p. 189)

Specifies new default hyperparameter ranges.

Type: DefaultHyperParameterRanges (p. 253) object

Required: No

defaultHyperParameters (p. 189)

Specifies new default hyperparameters.

Type: String to string map

Key Length Constraints: Maximum length of 256.

Value Length Constraints: Maximum length of 1000.

Required: No

defaultResourceConfig (p. 189)

Specifies a new default resource configuration.

Type: String to string map

Key Length Constraints: Maximum length of 256.

Value Length Constraints: Maximum length of 1000.

Required: No

trainingInputMode (p. 189)

Specifies a new training input mode.

Type: String

Length Constraints: Maximum length of 256.

Required: No

Response Syntax

```
{
   "algorithmArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

algorithmArn (p. 190)

The same algorithm ARN as given in the request.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

UpdateCampaign

Service: Amazon Personalize

Updates a campaign by either deploying a new solution or changing the campaign's update mode.

> **Note**
> When deploying a new solution, you must wait until the `status` of the updated campaign is
> `ACTIVE` before asking the campaign for recommendations.

For more information on campaigns, see CreateCampaign (p. 80).

Request Syntax

```
{
   "campaignArn": "string",
   "solutionVersionArn": "string",
   "updateMode": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

campaignArn (p. 192)

The Amazon Resource Name (ARN) of the campaign.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

solutionVersionArn (p. 192)

The ARN of a new solution version to deploy.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

updateMode (p. 192)

When the mode is AUTO, the campaign is automatically updated every time the solution is updated.
When the mode is MANUAL, you must call UpdateCampaign to update the campaign.

Type: String

Length Constraints: Maximum length of 256.

Valid Values: `AUTO` | `MANUAL`

Required: No

Response Syntax

```
{
    "campaignArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

campaignArn (p. 193)

The same campaign ARN as given in the request.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

UpdateDataset

Service: Amazon Personalize

Updates the given dataset to a new schema. For more information on datasets, see CreateDataset (p. 83).

Request Syntax

```
{
    "datasetArn": "string",
    "schemaArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

datasetArn (p. 194)

> The Amazon Resource Name (ARN) of the dataset to update.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: Yes

schemaArn (p. 194)

> The ARN of the new schema to associate with the dataset.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: No

Response Syntax

```
{
    "datasetArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

datasetArn (p. 194)

> The same dataset ARN as given in the request.
>
> Type: String
>
> Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceInUseException

The specified resource is in use.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

UpdateDatasetImportJob

Service: Amazon Personalize

Updates the specified dataset import job. Calling this operation can have three effects:

- Optionally changes the location of the training data to import.
- Optionally updates the import job repeat schedule.
- Starts a new import job.

For more information on dataset import jobs, see CreateDatasetImportJob (p. 89).

> **Important**
> The dataset import job replaces any previous data in the dataset.

A new import job cannot be created for a given dataset while another import job for that same dataset is running.

- ListDatasetImportJobRuns (p. 166)
- DescribeDatasetImportJobRun (p. 141)

Request Syntax

```
{
   "datasetImportJobArn": "string",
   "datasetImportJobConfig": {
      "importJobSchedule": "string"
   },
   "dataSource": {
      "dataLocation": "string"
   }
}
```

Request Parameters

The request accepts the following data in JSON format.

datasetImportJobArn (p. 196)

The Amazon Resource Name (ARN) of the dataset import job to update.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

datasetImportJobConfig (p. 196)

The configuration properties for the dataset import job.

Type: DatasetImportJobConfig (p. 238) object

Required: No

dataSource (p. 196)

The Amazon S3 bucket that contains the new data to use for training the model.

Type: DataSource (p. 250) object

Required: No

Response Syntax

```
{
    "datasetImportJobArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

datasetImportJobArn (p. 197)

The same dataset import job ARN as given in the request.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

UpdateFeatureTransformation

Service: Amazon Personalize

Updates the default parameters of the given feature transformation.

Request Syntax

```
{
   "defaultParameters": {
      "string" : "string"
   },
   "featureTransformationArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

defaultParameters (p. 199)

> The default parameters used to update the feature transformation.
>
> Type: String to string map
>
> Key Length Constraints: Maximum length of 256.
>
> Value Length Constraints: Maximum length of 1000.
>
> Required: No

featureTransformationArn (p. 199)

> The Amazon Resource Name (ARN) of the feature transformation to update.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: Yes

Response Syntax

```
{
   "featureTransformationArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

featureTransformationArn (p. 199)

> The same feature transformation ARN as given in the request.
>
> Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceNotFoundException

Could not find the specified resource.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

UpdateRecipe

Service: Amazon Personalize

Updates a recipe created by the CreateRecipe (p. 101) API to use a new algorithm or feature transformation.

To get a listing of the current recipe's properties, call the DescribeRecipe (p. 149) API.

Request Syntax

```
{
   "algorithmArn": "string",
   "featureTransformationArn": "string",
   "recipeArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

algorithmArn (p. 201)

> The Amazon Resource Name (ARN) of the algorithm to use in the updated recipe.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: No

featureTransformationArn (p. 201)

> The ARN of the feature transformation object to use in the updated recipe.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: No

recipeArn (p. 201)

> The ARN of the recipe to update.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: Yes

Response Syntax

```
{
   "recipeArn": "string"
```

```
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

recipeArn (p. 201)

>   The same recipe ARN as given in the request.
>
>   Type: String
>
>   Length Constraints: Maximum length of 256.
>
>   Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errors

InvalidInputException

>   Provide a valid value for the field or parameter.
>
>   HTTP Status Code: 400

ResourceNotFoundException

>   Could not find the specified resource.
>
>   HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

UpdateSolution

Service: Amazon Personalize

Updates an existing solution to use a different solution configuration. The solution must be in the ACTIVE state. To get the training status, call the DescribeSolution (p. 153) API. This API doesn't trigger retraining of the solution with the updated settings. For more information on solutions, see CreateSolution (p. 106).

Request Syntax

```
{
    "solutionArn": "string",
    "solutionConfig": {
        "algorithmHyperParameters": {
            "string" : "string"
        },
        "autoMLConfig": {
            "metricName": "string",
            "recipeList": [ "string" ]
        },
        "eventValueThreshold": "string",
        "featureTransformationParameters": {
            "string" : "string"
        },
        "hpoConfig": {
            "algorithmHyperParameterRanges": {
                "categoricalHyperParameterRanges": [
                    {
                        "name": "string",
                        "values": [ "string" ]
                    }
                ],
                "continuousHyperParameterRanges": [
                    {
                        "maxValue": number,
                        "minValue": number,
                        "name": "string"
                    }
                ],
                "integerHyperParameterRanges": [
                    {
                        "maxValue": number,
                        "minValue": number,
                        "name": "string"
                    }
                ]
            },
            "hpoObjective": {
                "metricName": "string",
                "metricRegex": "string",
                "type": "string"
            },
            "hpoResourceConfig": {
                "maxNumberOfTrainingJobs": "string",
                "maxParallelTrainingJobs": "string"
            }
        },
        "retrainingMode": "string",
        "retrainSchedule": "string"
    }
}
```

Request Parameters

The request accepts the following data in JSON format.

solutionArn (p. 203)

    The ARN of the solution to be updated.

    Type: String

    Length Constraints: Maximum length of 256.

    Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

    Required: Yes

solutionConfig (p. 203)

    The new solution configuration.

    Type: SolutionConfig (p. 278) object

    Required: No

Response Syntax

```
{
    "solutionArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

solutionArn (p. 204)

    The same solution ARN as given in the request.

    Type: String

    Length Constraints: Maximum length of 256.

    Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Errors

InvalidInputException

    Provide a valid value for the field or parameter.

    HTTP Status Code: 400

ResourceNotFoundException

    Could not find the specified resource.

    HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

Amazon Personalize Events

The following actions are supported by Amazon Personalize Events:

- PutEvents (p. 206)

PutEvents

Service: Amazon Personalize Events

Records user interaction event data.

Request Syntax

```
POST /events HTTP/1.1
Content-type: application/json

{
   "eventList": [
      {
         "context": {
            "string" : "string"
         },
         "eventId": "string",
         "eventType": "string",
         "properties": blob,
         "sentAt": number
      }
   ],
   "sessionId": "string",
   "trackingId": "string",
   "userId": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

eventList (p. 206)

> A list of event data from the session.
>
> Type: Array of Event (p. 287) objects
>
> Array Members: Minimum number of 1 item. Maximum number of 100 items.
>
> Required: Yes

sessionId (p. 206)

> The session ID associated with the user's visit.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Required: Yes

trackingId (p. 206)

> The tracking ID for the event. The ID is generated by a call to CreateEventTracker (p. 93).
>
> Type: String
>
> Length Constraints: Maximum length of 256.

Required: Yes

userId (p. 206)

The user associated with the event.

Type: String

Length Constraints: Maximum length of 256.

Required: No

Response Syntax

```
HTTP/1.1 200
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

Amazon Personalize Runtime

The following actions are supported by Amazon Personalize Runtime:

- GetRecommendations (p. 208)
- PersonalizeRanking (p. 211)

GetRecommendations

Service: Amazon Personalize Runtime

Returns a list of recommended items. The required input depends on the recipe type used to create the solution backing the campaign, as follows:

- RELATED_ITEMS - `itemId` required, `userId` not used
- USER_PERSONALIZATION - `itemId` optional, `userId` required

> **Note**
> Campaigns that are backed by a solution created using a recipe of type
> SEARCH_PERSONALIZATION use the PersonalizeRanking (p. 211) API.

Request Syntax

```
POST /recommendations HTTP/1.1
Content-type: application/json

{
    "campaignArn": "string",
    "itemId": "string",
    "numResults": number,
    "userId": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

campaignArn (p. 208)

The Amazon Resource Name (ARN) of the campaign to use for getting recommendations.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

itemId (p. 208)

The item ID to provide recommendations for.

Required for `RELATED_ITEMS` recipe type.

Type: String

Length Constraints: Maximum length of 100.

Required: No

numResults (p. 208)

The number of results to return. The default is 25. The maximum is 100.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

userId (p. 208)

The user ID to provide recommendations for.

Required for `USER_PERSONALIZATION` recipe type.

Type: String

Length Constraints: Maximum length of 100.

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "itemList": [
        {
            "itemId": "string"
        }
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

itemList (p. 209)

A list of recommendations.

Type: Array of PredictedItem (p. 289) objects

Array Members: Maximum number of 100 items.

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceNotFoundException

The specified resource does not exist.

HTTP Status Code: 404

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

PersonalizeRanking

Service: Amazon Personalize Runtime

Re-ranks a list of recommended items for the given user, filtered using the specified query. The first item in the list is deemed the most likely item to be of interest to the user.

> **Note**
> The solution backing the campaign must have been created using a recipe of type SEARCH_PERSONALIZATION.

Request Syntax

```
POST /personalize-ranking HTTP/1.1
Content-type: application/json

{
    "campaignArn": "string",
    "inputList": [ "string" ],
    "searchQuery": "string",
    "userId": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

campaignArn (p. 211)

The Amazon Resource Name (ARN) of the campaign to use for generating the personalized ranking.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: Yes

inputList (p. 211)

A list of items to rank.

Type: Array of strings

Array Members: Maximum number of 100 items.

Length Constraints: Maximum length of 100.

Required: Yes

searchQuery (p. 211)

A query used to filter the returned results.

Type: String

Length Constraints: Maximum length of 1024.

Pattern: `^[a-zA-Z0-9](-*[a-zA-Z0-9])*`

Required: No

userId (p. 211)

The user for which you want the campaign to provide a personalized ranking.

Type: String

Length Constraints: Maximum length of 100.

Required: Yes

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "personalizedRanking": [
      {
         "itemId": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

personalizedRanking (p. 212)

A list of items in order of most likely interest to the user.

Type: Array of PredictedItem (p. 289) objects

Array Members: Maximum number of 100 items.

Errors

InvalidInputException

Provide a valid value for the field or parameter.

HTTP Status Code: 400

ResourceNotFoundException

The specified resource does not exist.

HTTP Status Code: 404

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

Data Types

The following data types are supported by Amazon Personalize:

The following data types are supported by Amazon Personalize Events:

The following data types are supported by Amazon Personalize Runtime:

Amazon Personalize

The following data types are supported by Amazon Personalize:

Algorithm

Service: Amazon Personalize

Describes a custom algorithm for use with the CreateRecipe (p. 101) API.

Contents

algorithmArn

The Amazon Resource Name (ARN) of the algorithm.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

algorithmImage

The URI of the Docker container for the algorithm image.

Type: AlgorithmImage (p. 218) object

Required: No

creationDateTime

The date and time (in Unix time) that the algorithm was created.

Type: Timestamp

Required: No

defaultHyperParameterRanges

Specifies the default hyperparameters, their ranges, and whether they are tunable. A tunable hyperparameter can have its value determined during hyperparameter optimization (HPO).

Type: DefaultHyperParameterRanges (p. 253) object

Required: No

defaultHyperParameters

Specifies the default hyperparameters.

Type: String to string map

Key Length Constraints: Maximum length of 256.

Value Length Constraints: Maximum length of 1000.

Required: No

defaultResourceConfig

Specifies the default maximum number of training jobs and parallel training jobs.

Type: String to string map

Key Length Constraints: Maximum length of 256.

Value Length Constraints: Maximum length of 1000.

Required: No

lastUpdatedDateTime

The date and time (in Unix time) that the algorithm was last updated.

Type: Timestamp

Required: No

name

The name of the algorithm.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

roleArn

The Amazon Resource Name (ARN) of the role.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

trainingInputMode

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

AlgorithmImage

Service: Amazon Personalize

Contents

dockerURI

The URI of the Docker container for the algorithm image.

Type: String

Length Constraints: Maximum length of 256.

Required: Yes

name

The name of the algorithm image.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

AutoMLConfig

Service: Amazon Personalize

When the solution performs AutoML (`performAutoML` is true), Amazon Personalize determines which recipe from the specified list optimizes the given metric. Amazon Personalize then uses that recipe for the solution.

Contents

metricName

> The metric to optimize.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Required: No

recipeList

> The list of candidate recipes.
>
> Type: Array of strings
>
> Array Members: Maximum number of 100 items.
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

AutoMLResult

Service: Amazon Personalize

Specifies the recipe that best optimized the specified metric.

Contents

bestRecipeArn

The Amazon Resource Name (ARN) of the best recipe.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

Campaign

Service: Amazon Personalize

Describes a deployed solution, otherwise known as a campaign. For more information on campaigns, see CreateCampaign (p. 80).

Contents

campaignArn

The Amazon Resource Name (ARN) of the campaign.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

creationDateTime

The date and time (in Unix format) that the campaign was created.

Type: Timestamp

Required: No

failureReason

When a campaign fails, the reason behind the failure.

Type: String

Required: No

lastUpdatedDateTime

The date and time (in Unix format) that the campaign was last updated.

Type: Timestamp

Required: No

latestCampaignUpdate

Provides a summary of the properties of a campaign update. For a complete listing, call the DescribeCampaign (p. 133) API.

Type: CampaignUpdateSummary (p. 226) object

Required: No

name

The name of the campaign.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

solutionArn

The Amazon Resource Name (ARN) of the solution.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

solutionVersionArn

The Amazon Resource Name (ARN) of a specific version of the solution.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

status

The status of the campaign.

The status is one of the following values:
- CREATE PENDING => CREATE IN_PROGRESS => ACTIVE -or- CREATE FAILED
- DELETE PENDING => DELETE IN_PROGRESS => DELETE FAILED

Type: String

Length Constraints: Maximum length of 256.

Required: No

updateMode

When the mode is AUTO, the campaign is automatically updated every time the solution is updated.

When the mode is MANUAL, UpdateCampaign (p. 192) must be called to update the campaign.

Type: String

Length Constraints: Maximum length of 256.

Valid Values: `AUTO | MANUAL`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

CampaignSummary

Service: Amazon Personalize

Provides a summary of the properties of a campaign. For a complete listing, call the
DescribeCampaign (p. 133) API.

Contents

campaignArn

The Amazon Resource Name (ARN) of the campaign.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

creationDateTime

The date and time (in Unix time) that the campaign was created.

Type: Timestamp

Required: No

failureReason

When a campaign fails, the reason behind the failure.

Type: String

Required: No

lastUpdatedDateTime

The date and time (in Unix time) that the campaign was last updated.

Type: Timestamp

Required: No

name

The name of the campaign.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

The status of the campaign.

The status is one of the following values:
- CREATE PENDING => CREATE IN_PROGRESS => ACTIVE -or- CREATE FAILED
- DELETE PENDING => DELETE IN_PROGRESS => DELETE FAILED

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

CampaignUpdateSummary

Service: Amazon Personalize

Provides a summary of the properties of a campaign update. For a complete listing, call the DescribeCampaign (p. 133) API.

Contents

creationDateTime

The date and time (in Unix time) that the campaign update was created.

Type: Timestamp

Required: No

failureReason

When a campaign update fails, the reason behind the failure.

Type: String

Required: No

lastUpdatedDateTime

The date and time (in Unix time) that the campaign update was last updated.

Type: Timestamp

Required: No

solutionVersionArn

The Amazon Resource Name (ARN) of the deployed solution version.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

status

The status of the campaign update.

The status is one of the following values:

- CREATE PENDING => CREATE IN_PROGRESS => ACTIVE -or- CREATE FAILED
- DELETE PENDING => DELETE IN_PROGRESS => DELETE FAILED

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

CategoricalHyperParameterRange

Service: Amazon Personalize

Provides the name and range of a categorical hyperparameter.

Contents

name

The name of the hyperparameter.

Type: String

Length Constraints: Maximum length of 256.

Required: No

values

A list of the categories for the hyperparameter.

Type: Array of strings

Array Members: Maximum number of 100 items.

Length Constraints: Maximum length of 1000.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

ContinuousHyperParameterRange

Service: Amazon Personalize

Provides the name and range of a continuous hyperparameter.

Contents

maxValue

The maximum allowable value for the hyperparameter.

Type: Double

Valid Range: Minimum value of -1000000.

Required: No

minValue

The minimum allowable value for the hyperparameter.

Type: Double

Valid Range: Minimum value of -1000000.

Required: No

name

The name of the hyperparameter.

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

Dataset

Service: Amazon Personalize

Provides metadata for a dataset.

Contents

creationDateTime

The creation date and time (in Unix time) of the dataset.

Type: Timestamp

Required: No

datasetArn

The Amazon Resource Name (ARN) of the dataset that you want metadata for.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

datasetGroupArn

The Amazon Resource Name (ARN) of the dataset group.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

datasetType

One of the following values:
- Interactions
- Items
- Users

Type: String

Length Constraints: Maximum length of 256.

Required: No

lastUpdatedDateTime

A time stamp that shows when the dataset was updated.

Type: Timestamp

Required: No

name

The name of the dataset.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

schemaArn

The ARN of the associated schema.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

status

The status of the dataset.

The status is one of the following values:

- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

DatasetGroup

Service: Amazon Personalize

A dataset group is a collection of related datasets (Interactions, User, and Item). You create a dataset group by calling CreateDatasetGroup (p. 86). You then create a dataset and add it to a dataset group by calling CreateDataset (p. 83). The dataset group is used to create and train a solution by calling CreateSolution (p. 106). A dataset group can contain only one of each type of dataset.

You can specify an AWS Key Management Service (KMS) key to encrypt the datasets in the group.

Contents

creationDateTime

> The creation date and time (in Unix time) of the dataset group.

> Type: Timestamp

> Required: No

datasetGroupArn

> The Amazon Resource Name (ARN) of the dataset group.

> Type: String

> Length Constraints: Maximum length of 256.

> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

> Required: No

failureReason

> If creating a dataset group fails, provides the reason why.

> Type: String

> Required: No

kmsKeyArn

> The Amazon Resource Name (ARN) of the KMS key used to encrypt the datasets.

> Type: String

> Required: No

lastUpdatedDateTime

> The last update date and time (in Unix time) of the dataset group.

> Type: Timestamp

> Required: No

name

> The name of the dataset group.

> Type: String

> Length Constraints: Minimum length of 1. Maximum length of 63.

> Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

roleArn

The ARN of the IAM role that has permissions to create the dataset group.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Required: No

status

The current status of the dataset group.

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

DatasetGroupSummary

Service: Amazon Personalize

Provides a summary of the properties of a dataset group. For a complete listing, call the API.

Contents

creationDateTime

The date and time (in Unix time) that the dataset group was created.

Type: Timestamp

Required: No

datasetGroupArn

The Amazon Resource Name (ARN) of the dataset group.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

failureReason

When a creating a dataset group fails, the reason behind the failure.

Type: String

Required: No

lastUpdatedDateTime

The date and time (in Unix time) that the dataset group was last updated.

Type: Timestamp

Required: No

name

The name of the dataset group.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

The status of the dataset group.

The status is one of the following values:

- CREATE PENDING => CREATE IN_PROGRESS => ACTIVE -or- CREATE FAILED
- DELETE PENDING => DELETE IN_PROGRESS => DELETE FAILED

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

DatasetImportJob

Service: Amazon Personalize

Describes a job that imports training data from a data source (Amazon S3 bucket) to an Amazon Personalize dataset. For more information, see CreateDatasetImportJob (p. 89).

Contents

creationDateTime

The creation date and time (in Unix time) of the dataset import job.

Type: Timestamp

Required: No

datasetArn

The Amazon Resource Name (ARN) of the dataset that receives the imported data.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

datasetImportJobArn

The ARN of the dataset import job.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

datasetImportJobConfig

The dataset import job configuration properties, including the import update frequency.

Type: DatasetImportJobConfig (p. 238) object

Required: No

dataSource

The Amazon S3 bucket that contains the training data to import.

Type: DataSource (p. 250) object

Required: No

jobName

The name of the import job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

lastUpdatedDateTime

The date and time (in Unix time) of the last update run.

Type: Timestamp

Required: No

latestDatasetImportJobRun

Describes the latest run of the dataset import job.

Type: DatasetImportJobRunSummary (p. 241) object

Required: No

roleArn

The ARN of the AWS Identity and Access Management (IAM) role that has permissions to read from the Amazon S3 data source.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

status

The create status of the dataset import job.

The values can be
- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

DatasetImportJobConfig

Service: Amazon Personalize

The dataset import job configuration properties, including the import update frequency.

Contents

importJobSchedule

A cron expression that specifies the frequency for running the dataset import job. For more information, see Cron Expressions in the *Amazon CloudWatch User Guide*.

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

DatasetImportJobRun

Service: Amazon Personalize

Describes a specific run of a dataset import job. For more information, see
CreateDatasetImportJob (p. 89).

Contents

creationDateTime

The creation date and time (in Unix time) of the dataset import job run.

Type: Timestamp

Required: No

datasetArn

The Amazon Resource Name (ARN) of the dataset that receives the imported data.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

datasetImportJobArn

The ARN of the dataset import job.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

datasetImportJobConfig

The dataset import job configuration properties, including the import update frequency.

Type: DatasetImportJobConfig (p. 238) object

Required: No

datasetImportJobRunArn

The ARN of this run of the dataset import job.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

dataSource

The Amazon S3 bucket that contains the training data to import.

Type: DataSource (p. 250) object

Required: No

failureReason

When a dataset import job fails, the reason behind the failure.

Type: String

Required: No

lastUpdatedDateTime

The date and time (in Unix time) of the last import job run.

Type: Timestamp

Required: No

roleArn

The ARN of the AWS Identity and Access Management (IAM) role that has permissions to read from the Amazon S3 data source.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

status

The run status of the dataset import job.

The values can be
- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

DatasetImportJobRunSummary

Service: Amazon Personalize

Provides a summary of the properties of a dataset import job run. For a complete listing, call the
API.

Contents

creationDateTime

The date and time (in Unix time) that this run of the dataset import job was created.

Type: Timestamp

Required: No

datasetImportJobRunArn

The Amazon Resource Name (ARN) of the dataset import job run.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

failureReason

When a dataset import job run fails, the reason behind the failure.

Type: String

Required: No

lastUpdatedDateTime

The date and time (in Unix time) that the dataset import job run was last updated.

Type: Timestamp

Required: No

status

The status of the dataset import job run.

The status is one of the following values:

- CREATE PENDING => CREATE IN_PROGRESS => ACTIVE -or- CREATE FAILED
- DELETE PENDING => DELETE IN_PROGRESS => DELETE FAILED

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

DatasetImportJobSummary

Service: Amazon Personalize

Provides a summary of the properties of a dataset import job. For a complete listing, call the DescribeDatasetImportJob (p. 139) API.

Contents

creationDateTime

The date and time (in Unix time) that the dataset import job was created.

Type: Timestamp

Required: No

datasetImportJobArn

The Amazon Resource Name (ARN) of the dataset import job.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

jobName

The name of the dataset import job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

lastUpdatedDateTime

The date and time (in Unix time) that the dataset import job was last updated.

Type: Timestamp

Required: No

status

The status of the dataset import job.

The status is one of the following values:

- CREATE PENDING => CREATE IN_PROGRESS => ACTIVE -or- CREATE FAILED
- DELETE PENDING => DELETE IN_PROGRESS => DELETE FAILED

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

DatasetSchema

Service: Amazon Personalize

Describes the schema for a dataset. For more information on schemas, see .

Contents

creationDateTime

The date and time (in Unix time) that the schema was created.

Type: Timestamp

Required: No

lastUpdatedDateTime

The date and time (in Unix time) that the schema was last updated.

Type: Timestamp

Required: No

name

The name of the schema.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

schema

The schema.

Type: String

Length Constraints: Maximum length of 10000.

Required: No

schemaArn

The Amazon Resource Name (ARN) of the schema.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

DatasetSchemaSummary

Service: Amazon Personalize

Provides a summary of the properties of a dataset schema. For a complete listing, call the
DescribeSchema (p. 151) API.

Contents

creationDateTime

The date and time (in Unix time) that the schema was created.

Type: Timestamp

Required: No

lastUpdatedDateTime

The date and time (in Unix time) that the schema was last updated.

Type: Timestamp

Required: No

name

The name of the schema.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

schemaArn

The Amazon Resource Name (ARN) of the schema.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

DatasetSummary

Service: Amazon Personalize

Provides a summary of the properties of a dataset. For a complete listing, call the
DescribeDataset (p. 135) API.

Contents

creationDateTime

The date and time (in Unix time) that the dataset was created.

Type: Timestamp

Required: No

datasetArn

The Amazon Resource Name (ARN) of the dataset.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

datasetType

The dataset type. One of the following values:

- Interactions
- Items
- Users
- Event-Interactions

Type: String

Length Constraints: Maximum length of 256.

Required: No

lastUpdatedDateTime

The date and time (in Unix time) that the dataset was last updated.

Type: Timestamp

Required: No

name

The name of the dataset.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

The status of the dataset.

The status is one of the following values:

- CREATE PENDING => CREATE IN_PROGRESS => ACTIVE -or- CREATE FAILED
- DELETE PENDING => DELETE IN_PROGRESS => DELETE FAILED

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

DataSource

Service: Amazon Personalize

Describes the data source that contains the data to upload to a dataset.

Contents

dataLocation

The path to the Amazon S3 bucket where the data that you want to upload to your dataset is stored. For example:

```
s3://bucket-name/training-data.csv
```

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

DefaultCategoricalHyperParameterRange

Service: Amazon Personalize

Provides the name and default range of a categorical hyperparameter and whether the hyperparameter is tunable. A tunable hyperparameter can have its value determined during hyperparameter optimization (HPO).

Contents

isTunable

Whether the hyperparameter is tunable.

Type: Boolean

Required: No

name

The name of the hyperparameter.

Type: String

Length Constraints: Maximum length of 256.

Required: No

values

A list of the categories for the hyperparameter.

Type: Array of strings

Array Members: Maximum number of 100 items.

Length Constraints: Maximum length of 1000.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

DefaultContinuousHyperParameterRange

Service: Amazon Personalize

Provides the name and default range of a continuous hyperparameter and whether the hyperparameter is tunable. A tunable hyperparameter can have its value determined during hyperparameter optimization (HPO).

Contents

isTunable

Whether the hyperparameter is tunable.

Type: Boolean

Required: No

maxValue

The maximum allowable value for the hyperparameter.

Type: Double

Valid Range: Minimum value of -1000000.

Required: No

minValue

The minimum allowable value for the hyperparameter.

Type: Double

Valid Range: Minimum value of -1000000.

Required: No

name

The name of the hyperparameter.

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

DefaultHyperParameterRanges

Service: Amazon Personalize

Specifies the hyperparameters and their default ranges. Hyperparameters can be categorical, continuous, or integer-valued.

Contents

categoricalHyperParameterRanges

The categorical hyperparameters and their default ranges.

Type: Array of DefaultCategoricalHyperParameterRange (p. 251) objects

Array Members: Maximum number of 100 items.

Required: No

continuousHyperParameterRanges

The continuous hyperparameters and their default ranges.

Type: Array of DefaultContinuousHyperParameterRange (p. 252) objects

Array Members: Maximum number of 100 items.

Required: No

integerHyperParameterRanges

The integer-valued hyperparameters and their default ranges.

Type: Array of DefaultIntegerHyperParameterRange (p. 254) objects

Array Members: Maximum number of 100 items.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

DefaultIntegerHyperParameterRange

Service: Amazon Personalize

Provides the name and default range of a integer-valued hyperparameter and whether the hyperparameter is tunable. A tunable hyperparameter can have its value determined during hyperparameter optimization (HPO).

Contents

isTunable

Indicates whether the hyperparameter is tunable.

Type: Boolean

Required: No

maxValue

The maximum allowable value for the hyperparameter.

Type: Integer

Valid Range: Maximum value of 1000000.

Required: No

minValue

The minimum allowable value for the hyperparameter.

Type: Integer

Valid Range: Minimum value of -1000000.

Required: No

name

The name of the hyperparameter.

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

EventTracker

Service: Amazon Personalize

Provides information about an event tracker.

Contents

accountId

The Amazon AWS account that owns the event tracker.

Type: String

Length Constraints: Maximum length of 256.

Required: No

creationDateTime

The date and time (in Unix format) that the event tracker was created.

Type: Timestamp

Required: No

datasetGroupArn

The Amazon Resource Name (ARN) of the dataset group that receives the event data.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

eventTrackerArn

The ARN of the event tracker.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

lastUpdatedDateTime

The date and time (in Unix time) that the event tracker was last updated.

Type: Timestamp

Required: No

name

The name of the event tracker.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

roleArn

The ARN of the role that has *CloudWatchFullAccess* permission.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

status

The status of the event tracker.

The status is one of the following values:

- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED

Type: String

Length Constraints: Maximum length of 256.

Required: No

trackingId

The ID of the event tracker. Include this ID in requests to the PutEvents (p. 206) API.

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

EventTrackerSummary

Service: Amazon Personalize

Provides a summary of the properties of an event tracker. For a complete listing, call the DescribeEventTracker (p. 143) API.

Contents

creationDateTime

The date and time (in Unix time) that the event tracker was created.

Type: Timestamp

Required: No

eventTrackerArn

The Amazon Resource Name (ARN) of the event tracker.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

lastUpdatedDateTime

The date and time (in Unix time) that the event tracker was last updated.

Type: Timestamp

Required: No

name

The name of the event tracker.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

The status of the event tracker.

The status is one of the following values:

- CREATE PENDING => CREATE IN_PROGRESS => ACTIVE -or- CREATE FAILED
- DELETE PENDING => DELETE IN_PROGRESS => DELETE FAILED

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

FeatureExportJob

Service: Amazon Personalize

Creates a task to perform feature transformation. feature transformation is the process of modifying raw input data into a form more suitable for model training.

Contents

creationDateTime

The creation date and time (in Unix time) of the export job.

Type: Timestamp

Required: No

dataDestination

The Amazon S3 bucket that stores the results.

Type: String

Length Constraints: Maximum length of 256.

Required: No

datasetGroupArn

The Amazon Resource Name (ARN) of the dataset group containing the data to featurize.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

failureReason

When a feature export job fails, the reason behind the failure.

Type: String

Required: No

featureExportJobArn

The ARN of the export job.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

featureTransformationArn

The ARN of the FeatureTransformation object.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

featureTransformationOverrides

The values that override the default values in the feature transformation object.

Type: String to string map

Key Length Constraints: Maximum length of 256.

Value Length Constraints: Maximum length of 1000.

Required: No

jobName

The name of the export job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

lastUpdatedDateTime

The last update date and time (in Unix time) of the export job.

Type: Timestamp

Required: No

roleArn

The IAM role with permissions to the Amazon S3 bucket.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

status

The status of the feature export job.

The status is one of the following values:
- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

FeatureExportJobSummary

Service: Amazon Personalize

Provides a summary of the properties of a feature export job. For a complete listing, call the
DescribeFeatureExportJob (p. 145) API.

Contents

creationDateTime

The date and time (in Unix time) that the feature export job was created.

Type: Timestamp

Required: No

failureReason

When a feature export job fails, the reason behind the failure.

Type: String

Required: No

featureExportJobArn

The Amazon Resource Name (ARN) of the feature export job.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

jobName

The name of the feature export job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

lastUpdatedDateTime

The date and time (in Unix time) that the feature export job was last updated.

Type: Timestamp

Required: No

status

The status of the campaign.

The status is one of the following values:
- CREATE PENDING => CREATE IN_PROGRESS => ACTIVE -or- CREATE FAILED
- DELETE PENDING => DELETE IN_PROGRESS => DELETE FAILED

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

FeatureTransformation

Service: Amazon Personalize

Provides feature transformation information. feature transformation is the process of modifying raw input data into a form more suitable for model training.

Contents

creationDateTime

The creation date and time (in Unix time) of the feature transformation.

Type: Timestamp

Required: No

defaultParameters

Provides the default parameters for feature transformation.

Type: String to string map

Key Length Constraints: Maximum length of 256.

Value Length Constraints: Maximum length of 1000.

Required: No

featureTransformationArn

The Amazon Resource Name (ARN) of the FeatureTransformation object.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

lastUpdatedDateTime

The last update date and time (in Unix time) of the feature transformation.

Type: Timestamp

Required: No

name

The name of the feature transformation.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

status

The status of the feature transformation.

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

HPOConfig

Service: Amazon Personalize

Describes the properties for hyperparameter optimization (HPO). For use with the bring-your-own-recipe feature. Do not use for Amazon Personalize native recipes.

Contents

algorithmHyperParameterRanges

The hyperparameters and their allowable ranges.

Type: HyperParameterRanges (p. 269) object

Required: No

hpoObjective

The metric to optimize during HPO.

Type: HPOObjective (p. 267) object

Required: No

hpoResourceConfig

Describes the resource configuration for HPO.

Type: HPOResourceConfig (p. 268) object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

HPOObjective

Service: Amazon Personalize

The metric to optimize during hyperparameter optimization (HPO).

Contents

metricName

The name of the metric.

Type: String

Length Constraints: Maximum length of 256.

Required: No

metricRegex

A regular expression for finding the metric in the training job logs.

Type: String

Length Constraints: Maximum length of 256.

Required: No

type

The data type of the metric.

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

HPOResourceConfig

Service: Amazon Personalize

Describes the resource configuration for hyperparameter optimization (HPO).

Contents

maxNumberOfTrainingJobs

The maximum number of training jobs.

Type: String

Length Constraints: Maximum length of 256.

Required: No

maxParallelTrainingJobs

The maximum number of parallel training jobs.

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

HyperParameterRanges

Service: Amazon Personalize

Specifies the hyperparameters and their ranges. Hyperparameters can be categorical, continuous, or integer-valued.

Contents

categoricalHyperParameterRanges

The categorical hyperparameters and their ranges.

Type: Array of CategoricalHyperParameterRange (p. 228) objects

Array Members: Maximum number of 100 items.

Required: No

continuousHyperParameterRanges

The continuous hyperparameters and their ranges.

Type: Array of ContinuousHyperParameterRange (p. 229) objects

Array Members: Maximum number of 100 items.

Required: No

integerHyperParameterRanges

The integer-valued hyperparameters and their ranges.

Type: Array of IntegerHyperParameterRange (p. 270) objects

Array Members: Maximum number of 100 items.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

IntegerHyperParameterRange

Service: Amazon Personalize

Provides the name and range of an integer-valued hyperparameter.

Contents

maxValue

The maximum allowable value for the hyperparameter.

Type: Integer

Valid Range: Maximum value of 1000000.

Required: No

minValue

The minimum allowable value for the hyperparameter.

Type: Integer

Valid Range: Minimum value of -1000000.

Required: No

name

The name of the hyperparameter.

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

Recipe

Service: Amazon Personalize

Provides information about a recipe. Each recipe provides an algorithm that Amazon Personalize uses in model training when you use the CreateSolution (p. 106) API to create a solution.

Contents

algorithmArn

The Amazon Resource Name (ARN) of the algorithm that Amazon Personalize uses to train the model.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

creationDateTime

The date and time (in Unix format) that the recipe was created.

Type: Timestamp

Required: No

description

The description of the recipe.

Type: String

Required: No

featureTransformationArn

The ARN of the FeatureTransformation object.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

lastUpdatedDateTime

The date and time (in Unix format) that the recipe was last updated.

Type: Timestamp

Required: No

name

The name of the recipe.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

recipeArn

The Amazon Resource Name (ARN) of the recipe.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

recipeType

One of the following values:

- RELATED_ITEMS
- SEARCH_PERSONALIZATION
- USER_PERSONALIZATION

Type: String

Length Constraints: Maximum length of 256.

Required: No

status

The status of the recipe.

The status is one of the following values:

- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

RecipeSummary

Service: Amazon Personalize

Provides a summary of the properties of a recipe. For a complete listing, call the DescribeRecipe (p. 149) API.

Contents

creationDateTime

The date and time (in Unix time) that the recipe was created.

Type: Timestamp

Required: No

lastUpdatedDateTime

The date and time (in Unix time) that the recipe was last updated.

Type: Timestamp

Required: No

name

The name of the recipe.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

recipeArn

The Amazon Resource Name (ARN) of the recipe.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

status

The status of the recipe.

The status is one of the following values:

- CREATE PENDING => CREATE IN_PROGRESS => ACTIVE -or- CREATE FAILED
- DELETE PENDING => DELETE IN_PROGRESS => DELETE FAILED

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

Solution

Service: Amazon Personalize

An object that provides information about a solution. A solution is a trained model that is deployed as a campaign.

Contents

autoMLResult

> When `performAutoML` is true, specifies the best recipe found.
>
> Type: AutoMLResult (p. 220) object
>
> Required: No

creationDateTime

> The creation date and time (in Unix time) of the solution.
>
> Type: Timestamp
>
> Required: No

datasetGroupArn

> The Amazon Resource Name (ARN) of the dataset group that provides the training data.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`
>
> Required: No

eventType

> The event type (for example, 'click' or 'like') that is used as the label for training.
>
> Type: String
>
> Length Constraints: Maximum length of 256.
>
> Required: No

lastUpdatedDateTime

> The date and time (in Unix time) that the solution was last updated.
>
> Type: Timestamp
>
> Required: No

latestSolutionVersion

> Describes the latest version of the solution, including the status and the ARN.
>
> Type: SolutionVersionSummary (p. 285) object
>
> Required: No

minProvisionedTPS

> Specifies the requested minimum provisioned transactions (recommendations) per second.

Type: Integer

Required: No

name

The name of the solution.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

performAutoML

When true, Amazon Personalize performs a search for the best recipe from the list specified in the solution configuration (`recipeArn` is ignored). When false, Amazon Personalize uses `recipeArn`.

Type: Boolean

Required: No

performHPO

Whether to perform hyperparameter optimization (HPO) on the chosen recipe.

Type: Boolean

Required: No

recipeArn

The ARN of the recipe used to create the solution.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

solutionArn

The ARN of the solution.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

solutionConfig

Describes the configuration properties for the solution.

Type: SolutionConfig (p. 278) object

Required: No

status

The status of the solution.

The status is one of the following values:

- CREATE PENDING => CREATE IN_PROGRESS => ACTIVE -or- CREATE FAILED
- DELETE PENDING => DELETE IN_PROGRESS => DELETE FAILED

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

SolutionConfig

Service: Amazon Personalize

Describes the configuration properties for the solution.

Contents

algorithmHyperParameters

Lists the hyperparameter names and ranges.

Type: String to string map

Key Length Constraints: Maximum length of 256.

Value Length Constraints: Maximum length of 1000.

Required: No

autoMLConfig

The AutoMLConfig (p. 219) object containing a list of recipes to search when AutoML is performed.

Type: AutoMLConfig (p. 219) object

Required: No

eventValueThreshold

Only events with a value greater than or equal to this threshold are used for training a model.

Type: String

Length Constraints: Maximum length of 256.

Required: No

featureTransformationParameters

Lists the feature transformation parameters.

Type: String to string map

Key Length Constraints: Maximum length of 256.

Value Length Constraints: Maximum length of 1000.

Required: No

hpoConfig

Describes the properties for hyperparameter optimization (HPO). For use with the bring-your-own-recipe feature. Not used with Amazon Personalize predefined recipes.

Type: HPOConfig (p. 266) object

Required: No

retrainingMode

Setting the mode to 'AUTO' retrains the solution with a predefined default for the recipe. In this case, a retraining schedule is not required. Setting the mode to 'MANUAL' retrains the solution as specified in `retrainSchedule`. If `retrainSchedule` is not provided, the mode defaults to 'AUTO'.

Type: String

Length Constraints: Maximum length of 256.

Valid Values: `AUTO | MANUAL`

Required: No

retrainSchedule

When `retrainingMode` is set to 'MANUAL', the solution is retrained according to the specified cron expression. For more information, see Cron Expressions in the *Amazon CloudWatch User Guide*.

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

SolutionSummary

Service: Amazon Personalize

Provides a summary of the properties of a solution. For a complete listing, call the
API.

Contents

creationDateTime

The date and time (in Unix time) that the solution was created.

Type: Timestamp

Required: No

lastUpdatedDateTime

The date and time (in Unix time) that the solution was last updated.

Type: Timestamp

Required: No

name

The name of the solution.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9][a-zA-Z0-9\-_]*`

Required: No

solutionArn

The Amazon Resource Name (ARN) of the solution.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

status

The status of the solution.

The status is one of the following values:

- CREATE PENDING => CREATE IN_PROGRESS => ACTIVE -or- CREATE FAILED
- DELETE PENDING => DELETE IN_PROGRESS => DELETE FAILED

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

SolutionVersion

Service: Amazon Personalize

An object that provides information about a specific version of a Solution (p. 275).

Contents

creationDateTime

The date and time (in Unix time) that this version of the solution was created.

Type: Timestamp

Required: No

datasetGroupArn

The Amazon Resource Name (ARN) of the dataset group providing the training data.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

eventType

The event type (for example, clicks or watches) that is used as the label for training.

Type: String

Length Constraints: Maximum length of 256.

Required: No

failureReason

When a solution fails, the reason behind the failure.

Type: String

Required: No

lastUpdatedDateTime

The date and time (in Unix time) that this version of the solution was last updated.

Type: Timestamp

Required: No

performAutoML

When true, Amazon Personalize performs a search for the most optimal recipe according to the solution configuration. When false, Amazon Personalize uses `recipeArn`.

Type: Boolean

Required: No

performHPO

Whether to perform hyperparameter optimization (HPO) on the chosen recipe.

Type: Boolean

Required: No

recipeArn

The ARN of the recipe used in the solution.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

solutionArn

The ARN of the solution.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

solutionConfig

Describes the configuration properties for the solution.

Type: SolutionConfig (p. 278) object

Required: No

solutionVersionArn

The ARN of the solution version.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

status

The status of the solution version.

The status is one of the following values:

- CREATE PENDING
- CREATE IN_PROGRESS
- ACTIVE
- CREATE FAILED

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

SolutionVersionSummary

Service: Amazon Personalize

Provides a summary of the properties of a solution version. For a complete listing, call the
DescribeSolutionVersion (p. 156) API.

Contents

creationDateTime

The date and time (in Unix time) that this version of a solution was created.

Type: Timestamp

Required: No

failureReason

When a solution version fails, the reason behind the failure.

Type: String

Required: No

lastUpdatedDateTime

The date and time (in Unix time) that the solution version was last updated.

Type: Timestamp

Required: No

solutionVersionArn

The Amazon Resource Name (ARN) of the solution version.

Type: String

Length Constraints: Maximum length of 256.

Pattern: `arn:([a-z\d-]+):personalize:.*:.*:.+`

Required: No

status

The status of the solution version.

The status is one of the following values:

- CREATE PENDING => CREATE IN_PROGRESS => ACTIVE -or- CREATE FAILED
- DELETE PENDING => DELETE IN_PROGRESS => DELETE FAILED

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

Amazon Personalize Events

The following data types are supported by Amazon Personalize Events:

- Event (p. 287)

Event

Service: Amazon Personalize Events

Represents user interaction event information.

Contents

context

Provides event context information as a key-value pair. For example, browser and version used.

Type: String to string map

Key Length Constraints: Maximum length of 256.

Value Length Constraints: Maximum length of 256.

Required: No

eventId

An ID associated with the event.

Type: String

Length Constraints: Maximum length of 256.

Required: Yes

eventType

The type of event. This property corresponds to the `event_type` field of the event-interactions schema.

Type: String

Length Constraints: Maximum length of 256.

Required: Yes

properties

A string map of event-specific data that you might choose to record. For example, if a user chooses a shoe icon on your site, you might send size, brand, and color information of the shoe.

Each item in the map consists of one or two key-value pairs. For example,

`{"id": "itemid1"}`

`{"id": "itemid1", "value": "itemname"}`

The `id` and `value` keys correspond to the `item_id` and `event_value` fields of the event-interactions schema.

When using the JavaScript SDK, two special event types are available:

- ListView

  For this event type, you use `items` as the key with an array of objects as the value (each object has one or more key-value pairs). For example,

  `{items: [{"id": "item1", "value":"value1"}, {...}]]}`
- MediaAutoTrack

Type: Base64-encoded binary data object

Required: Yes

sentAt

The timestamp on the client side when the event occurred.

Type: Timestamp

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

Amazon Personalize Runtime

The following data types are supported by Amazon Personalize Runtime:

- PredictedItem (p. 289)

PredictedItem

Service: Amazon Personalize Runtime

An object that identifies an item.

The GetRecommendations (p. 208) and PersonalizeRanking (p. 211) APIs return a list of `PredictedItem`s.

Contents

itemId

The recommended item ID.

Type: String

Length Constraints: Maximum length of 100.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 400

InternalFailure

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

InvalidAction

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 400

InvalidClientTokenId

The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

InvalidParameterCombination

Parameters that must not be used together were used together.

HTTP Status Code: 400

InvalidParameterValue

An invalid or out-of-range value was supplied for the input parameter.

HTTP Status Code: 400

InvalidQueryParameter

The AWS query string is malformed or does not adhere to AWS standards.

HTTP Status Code: 400

MalformedQueryString

The query string contains a syntax error.

HTTP Status Code: 404

MissingAction

The request is missing an action or a required parameter.

HTTP Status Code: 400

MissingAuthenticationToken

The request must contain either a valid (registered) AWS access key ID or X.509 certificate.

HTTP Status Code: 403

MissingParameter

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

OptInRequired

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

RequestExpired

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

ServiceUnavailable

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 400

ValidationError

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400

Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information about Signature Version 4, see Signature Version 4 Signing Process in the *Amazon Web Services General Reference*.

Action

The action to be performed.

Type: string

Required: Yes

Version

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Type: string

Required: Yes

X-Amz-Algorithm

The hash algorithm that you used to create the request signature.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Valid Values: `AWS4-HMAC-SHA256`

Required: Conditional

X-Amz-Credential

The credential scope value, which is a string that includes your access key, the date, the region you are targeting, the service you are requesting, and a termination string ("aws4_request"). The value is expressed in the following format: *access_key*/*YYYYMMDD*/*region*/*service*/aws4_request.

For more information, see Task 2: Create a String to Sign for Signature Version 4 in the *Amazon Web Services General Reference*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-Date

The date that is used to create the signature. The format must be ISO 8601 basic format (YYYYMMDD'T'HHMMSS'Z'). For example, the following date time is a valid X-Amz-Date value: `20120325T120000Z`.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see Handling Dates in Signature Version 4 in the *Amazon Web Services General Reference*.

Type: string

Required: Conditional

X-Amz-Security-Token

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS Security Token Service, go to AWS Services That Work with IAM in the *IAM User Guide*.

Condition: If you're using temporary security credentials from the AWS Security Token Service, you must include the security token.

Type: string

Required: Conditional

X-Amz-Signature

Specifies the hex-encoded signature that was calculated from the string to sign and the derived signing key.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-SignedHeaders

Specifies all the HTTP headers that were included as part of the canonical request. For more information about specifying signed headers, see Task 1: Create a Canonical Request For Signature Version 4 in the *Amazon Web Services General Reference*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

Document History for Amazon Personalize

The following table describes important changes in each release of the *Amazon Personalize Developer Guide*. For notification about updates to this documentation, you can subscribe to an RSS feed.

- **Latest documentation update:** November 28th, 2018

update-history-change	update-history-description	update-history-date
Amazon Personalize Released. (p. 293)	This is the first release of the documentation for Amazon Personalize.	November 28, 2018

AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the *AWS General Reference*.