Reports API Use Case Guide

Version: 1.0

Date: 6/17/2020

Contents

What is the Reports API?

With the Selling Partner API for Reports (Reports API), you can build applications that enable sellers to get reports from Amazon that helps them manage their selling business. There are reports for a wide variety of use cases, such as monitoring inventory, tracking orders for fulfillment, getting tax information, tracking returns and seller performance, managing a selling business with Fulfillment by Amazon, and more. See Reports Datatypes for a complete list of report types.

The two principal workflows for getting reports are requesting a report and scheduling a report. Only order reports can be scheduled.

Requesting a report

You can request any available report type using the **requestReport** operation. See Tutorial: Request a report for instructions for directly requesting reports in this way.

Scheduling an order report (for driving an order fulfillment process)

To build an application that helps sellers drive their fulfillment process for seller-fulfilled customer orders, Amazon recommends setting up order report scheduling. See Schedule order reports for instructions for scheduling order reports.

Terminology

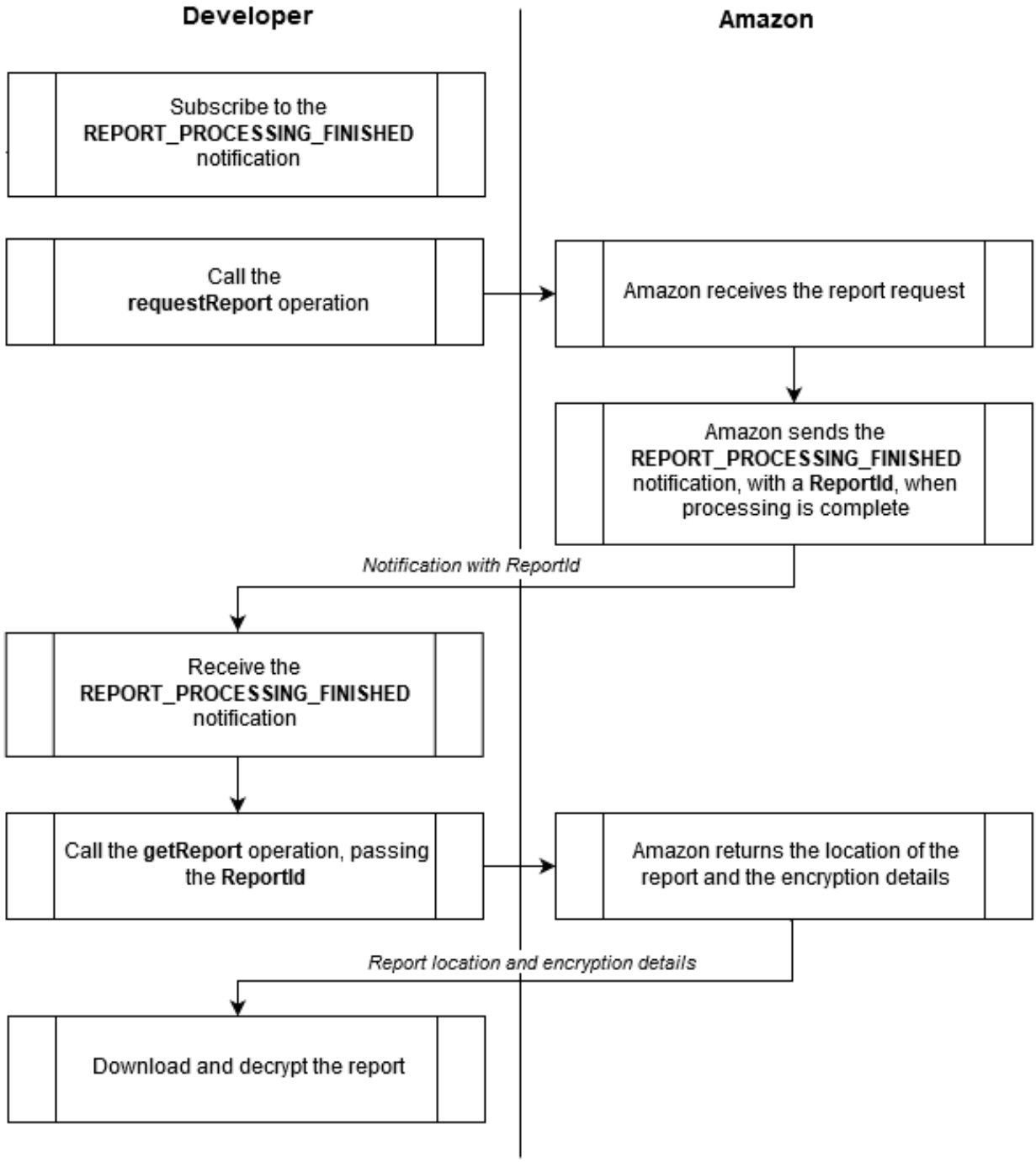
- **Cipher block chaining**. Cipher block chaining is an algorithm that uses a block cipher to provide information security such as confidentiality or authenticity. This algorithm uses an initialization vector and a key to encrypt the data.
- **S3 presigned URL**. A URL for an AWS S3 bucket from which you can download an object without AWS security credentials or permissions. You get an S3 presigned URL in Task 1. Get location and encryption information.

Tutorial: Request a report

Here are the high-level steps for requesting a report:

1. Subscribe to the **REPORT_PROCESSING_FINISHED** notification. This is a one-time task.
2. Call the **requestReport** operation, specifying the type of report that you are requesting and any optional parameters that you want.
3. Amazon receives the report request.
4. Wait for the **REPORT_PROCESSING_FINISHED** notification.
5. Amazon sends you the **REPORT_PROCESSING_FINISHED** notification indicating that report processing is complete. The notification includes a **ReportId** value.
6. Call the **getReport** operation, passing the **ReportId** value from the previous step.
7. Amazon returns the location of the report along with the encryption details.
8. Download and decrypt the report.

The following flowchart shows the process for requesting a report:



Prerequisites

To complete this tutorial, you will need:

- A report to request. See ReportType enumeration in the Amazon Marketplace Web Service documentation for a list of the available report types.

- Authorization from the seller for whom you are making calls. See the Selling Partner API Developer Guide for more information.
- A working Java Development Kit (JDK) installation, including the javax.crypto library.
- An understanding of client-side encryption using the cipher block chaining (CBC). For definitions, see Terminology.

Java code samples

This tutorial contains Java code samples that can help you build a Java application that gets reports from Amazon. You can use principles demonstrated in these code samples to guide you in building applications in other programming languages.

For the detailed steps for requesting a report, start at Step 1. Request a report.

Steps

Step 1. Request a report

Step 2. Confirm report processing

Step 3. Get the report

Supplemental Java code

This section contains the `CryptoHelper` class and the `CryptoProvider` interface, which are referenced in the sample Java code in Task 2. Download and decrypt the report.

```
/** CryptoProvider.java **/
package com.amazon.spapi;

import javax.crypto.Cipher;
import io.swagger.client.model.EncryptionDetails;

 @ FunctionalInterface
public interface CryptoProvider
{
  Cipher getInitializedCipher(int mode, EncryptionDetails
encryptionDetails);
}

/** CryptoHelper.java **/
package com.amazon.spapi;

import io.swagger.client.model.EncryptionDetails;

import javax.crypto.Cipher;
import javax.crypto.CipherInputStream;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;
import java.io.InputStream;
import java.security.GeneralSecurityException;
import java.security.Key;
import java.security.SecureRandom;
```

```
import java.util.Base64;

public class CryptoHelper
{

  static final String AES = "AES";
  static final Base64.Decoder BASE64_DECODER = Base64.getDecoder();

  static final CryptoProvider AES_CRYPTO_PROVIDER =
UploadToDestinationExample::getInitializedCipher;

  static InputStream buildCipherInputStream(EncryptionDetails
encryptionDetails, InputStream stream, int mode)
  {
    return new CipherInputStream(stream,
AES_CRYPTO_PROVIDER.getInitializedCipher(mode, encryptionDetails));
  }

  static Cipher getInitializedCipher(int mode, EncryptionDetails
details)
  {

    Cipher cipher;
    try
    {
      cipher = Cipher.getInstance(AES);
      Key key = new
SecretKeySpec(BASE64_DECODER.decode(details.getKey()), AES);
      byte[]iv =
BASE64_DECODER.decode(details.getInitializationVector());
      IvParameterSpec ivParameterSpec = new IvParameterSpec(iv);
      cipher.init(mode, key, ivParameterSpec, new SecureRandom());
    }
    catch (GeneralSecurityException e)
    {
      throw new IllegalStateException("Could not create Cipher for
key-iv pair", e);
    }

    return cipher;
  }

}
```

Step 1. Request a report

Call the **requestReport** operation to specify the type of report that you are requesting and any optional parameters that you want.

- Call the **requestReport** operation, passing the following parameters:

Path parameters:

Name	Description	Required
reportType	The type of report that you are requesting. For **reportType** values, see ReportType enumeration in the Amazon Marketplace Web Service documentation. Type: string	Yes

Body parameters:

Name	Description	Required
requestReportBody	Additional information to pass if a report accepts report options. Type: object	No

Query parameters:

Name	Description	Required
StartDate	The start of a date range, in ISO 8601 date time format, used for selecting the data to report. The default is now. The value must be prior to or equal to the current date and time. Type: string	No
EndDate	The end of a date range, in ISO 8601 date time format, used for selecting the data to report. The default is now. The value must be prior to or equal to the current date and time. Type: string	No
MarketplaceIds	A list of one or more marketplace identifiers for the marketplaces that the seller is registered to sell in. The resulting report will include information for all marketplaces you specify. The default is the first marketplace that the seller registered to sell in. Type: array[string]	No

Request example:

```
POST https://sellingpartnerapi-
na.amazon.com/reports/v0/request/_GET_MERCHANT_LISTINGS_ALL_DATA_?Star
tDate=2019-12-
10T20:11:24.000Z&MarketplaceIds=A1PA6795UKMFR9,ATVPDKIKX0DER
```

Response

A successful response includes the following property:

Name	Description
ReportRequestInfo	Detailed information about a report request. Type: ReportRequestInfo

Response example:

```
{
  "payload":
  {
    "ReportRequestInfo":
    {
      "ReportRequestId": "ID323",
      "ReportType": "_GET_MERCHANT_LISTINGS_ALL_DATA_",
      "StartDate": "2019-12-10T14:53:45.962Z",
      "EndDate": "2019-12-12T14:53:45.962Z",
      "Scheduled": false,
      "SubmittedDate": "2019-12-11T14:53:45.962Z",
    }
  }
}
```

Step 2. Confirm report processing

After you call the requestReport operation you need to wait for confirmation that we have processed your report before you can continue. Amazon recommends subscribing to the **REPORT_PROCESSING_FINISHED** notification to get this confirmation. After you subscribe, we will send you a push notification when we finish processing any report that you submit. To subscribe to the **REPORT_PROCESSING_FINISHED**, see the Notifications use case guide. Subscribing to the **REPORT_PROCESSING_FINISHED** notification is a one-time task.

To confirm report processing

1. Be sure that you are subscribed to the **REPORT_PROCESSING_FINISHED** notification. To subscribe to this notification, see the Notifications use case guide.
2. After Step 1. Request a report, wait for the **REPORT_PROCESSING_FINISHED** notification.

   When report processing is complete, you receive the **REPORT_PROCESSING_FINISHED** notification with the **ReportProcessingStatus** element set to one of these values:

   - *DONE* - The report was successfully generated. Use the **ReportId** value included in the notification as input for the **getReport** operation in Task 1. Get location and encryption information.

- `CANCELLED` - The report was cancelled. See the "Why are my reports cancelled?" section of the Downloading Reports article in the Amazon Marketplace Web Service (Amazon MWS) Knowledge Base for reasons why reports are cancelled.
- `DONE_NO_DATA` - The report was generated but there was no data to report. This happens when there is no new data to report between consecutive report requests. Retry until there is data to report.

Note. An alternative way to confirm report processing is to poll the **getReportRequestList** operation until the response indicates that report processing is complete. A downside to polling is that repeated calls the **GetReportRequestList** operation could make you exceed throttling limits. If polling is the better option for you, however, see What you should know about the Amazon MWS Reports API section in the Amazon MWS documentation. The workflow for polling the **getReportRequestList** operation using Amazon MWS is nearly identical to the polling workflow using Selling Partner API. The major difference is that with the Selling Partner API, after calling the **getReport** operation you must download and decrypt the report.

Step 3. Get the report

Get your report by first getting location and encryption information and then downloading and decrypting the report.

Tasks

Task 1. Get location and encryption information

Call the **getReport** operation to get the location of your report and the information you will need to decrypt it.

1. Call the **getReport** operation using the following parameters:

Path parameters:

Name	Description	Required
reportId	The identifier of the report. This identifier is included in the **REPORT_PROCESSING_FINISHED** notification. See Step 2. Confirm report processing. Type: string	Yes

Request example:

```
GET https://sellingpartnerapi-
na.amazon.com/reports/v0/reports/ReportIdExample
```

Response

A successful response includes the following elements:

Name	Description
destination	The location of the report content. Type: destination
encryptionDetails	Encryption details required for decrypting the report content. Type: EncryptionDetails
isGzipped	When *true*, the report is compressed using Gzip compression. Type: boolean

Response example:

```
{
  "payload":
  {
    "destination":
    {
      "channel": "S3",
      "url": "https://s3.amazonaws.com/%2FNinetyDays/Order-report-
test3.94241e3b-125d-44c6-a0b4- 712c08ee9f99.T1RC7FCPQ663P6?X-Amz-
Algorithm=AWS4-HMAC-SHA256&X-Amz- Date=20200101T240000Z&X-Amz-
SignedHeaders=host&X-Amz-Expires=300&X- Amz-
Credential=AKIA5U6MO6WEL7IJWREL%2F20200101%2Fus-east-
1%2Fs3%2Faws4_request&X-Amz-
Signature=f34f7410c4c2effd974875172944w6wsf3b319132a9f83608c3ca0e9b6c4
c8b"
    },
    "encryptionDetails":
    {
      "standard": "AES",
      "initializationVector": "SAMPLE_InitializationVector",
      "key": "SAMPLE_Key"
    },
    "": false
  }
  isGzipped
}
```

2. Save the **destination**, **encryptionDetails**, and **isGzipped** values to pass in Task 2. Download and decrypt the report.

Task 2. Download and decrypt the report

The Java sample code in this task contains logic for downloading and decrypting the report. This sample code uses the Apache HTTP client. See Supplemental Java code for the types referenced in the sample code.

1. Use the following as inputs for the sample code:
   - The **destination** value that you saved in Task 1. Get location and encryption information is the argument for the `destination` parameter of the `downloadAndDecryptReportContent` method of the `DecryptReportContentExample` class.
   - The **isGzipped** value that you saved in Task 1. Get location and encryption information is the argument for the `isGzipped` parameter of the `downloadAndDecryptReportContent` method of the `DecryptReportContentExample` class.
   - The **encryptionDetails** value that you saved in Task 1. Get location and encryption information is the argument for the `encryptionDetails` parameter of the `InputStream` method of the `CryptoHelper` class. See Supplemental Java code for the `CryptoHelper` class.
2. Save the `result` value. This is your decrypted report.

Sample Java code

```
package com.amazon.spapi;

import io.swagger.client.model.EncryptionDetails;
import io.swagger.client.model.UploadDestination;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.HttpStatus;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.HttpClients;

import javax.crypto.Cipher;
import java.io.IOException;
import java.io.InputStream;
import java.util.zip.GZIPInputStream;

import static com.amazon.spapi.CryptoHelper.buildCipherInputStream;

public class DecryptReportContentExample
{

  public InputStream downloadAndDecryptReportContent(UploadDestination
destination, boolean isGzipped)
  throws IOException
  {
    InputStream result = null;
    HttpResponse httpResponse = null;
    String url = destination.getUrl();
```

```
    // Acquire the file
    HttpClient httpClient = HttpClients.createDefault();
    HttpGet httpGet = new HttpGet(url);
    httpResponse = httpClient.execute(httpGet);
    if (httpResponse == null ||
httpResponse.getStatusLine().getStatusCode() ==
HttpStatus.SC_NOT_FOUND)
    {
      throw new IllegalArgumentException("Could not find result at
destination.");
    }
    HttpEntity entity = httpResponse.getEntity();
    if (entity == null)
    {
      throw new RuntimeException("The HTTP store returned success but
no document.");
    }
    result = decryptReportContent(entity.getContent(),
destination.getEncryptionDetails(), isGzipped);
    return result;
  }

  private InputStream decryptReportContent(
    InputStream input, EncryptionDetails encryptionDetails, boolean
isGzipped)throws IOException
  {
    InputStream resultStream = input;
    // If encrypted, decipher the stream
    if (encryptionDetails != null &&
EncryptionDetails.StandardEnum.AES.equals(encryptionDetails.getStandar
d()))
    {
      resultStream = buildCipherInputStream(encryptionDetails,
resultStream, Cipher.DECRYPT_MODE);
    }

    // Determine if the stream should be unzipped as well
    if (isGzipped)
    {
      resultStream = new GZIPInputStream(resultStream);
    }

    return resultStream;
  }
}
```

Schedule order reports

You can schedule requests for order reports so that they are submitted periodically, using the **ManageReportSchedule** operation. Use the Schedule enumeration to specify the time period for requesting the order requests. For a list of order reports that can be scheduled, see Order Reports in the

Amazon Marketplace Web Service (Amazon MWS) documentation. For more information about working with order reports, see Requesting order reports.

Here are the high-level steps for scheduling order reports:

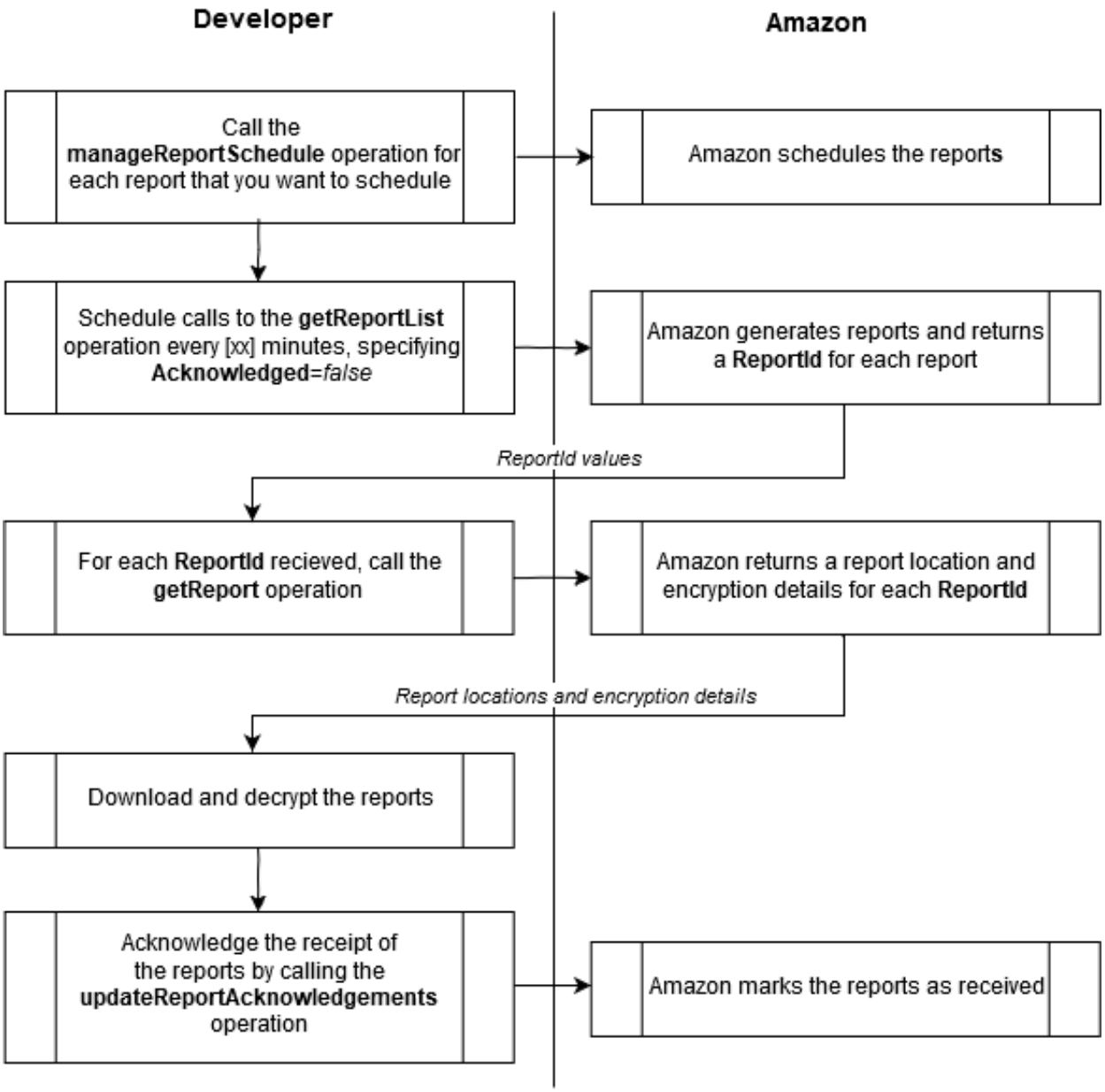
1. Call the **manageReportSchedule** operation to create a schedule for periodically submitting order report requests. Specify **ReportType** and **Schedule** values for the report type and time period that you want. For **ReportType** values, see ReportType enumeration in the Amazon MWS documentation. For **Schedule** values, see Schedule enumeration. Note the following behaviors:
    - If no combination of **ReportType** and **Schedule** exists from a previous call to **manageReportSchedule**, then Amazon creates a new report request schedule.
    - If the specified **ReportType** is already scheduled but with a different **Schedule** value, then Amazon updates the report request schedule to use the new **Schedule** value.
    - If you pass a **ReportType** value and set the **Schedule** value to *_NEVER_*, Amazon deletes the report request schedule for that report type.
2. Schedule calls to the **getReportList** operation using an interval that is similar to the schedule that you configured in the previous step. Specify *false* for the **Acknowledged** parameter. Each call to the **getReportList** operation determines whether Amazon has generated any new order reports since the previous call to **getReportList**.
3. Capture any **ReportId** values that are returned after calls to the **getReportList** operation. If no reports were generated since the last call to the **getReportList** operation, no **ReportId** values are returned.
4. For each **ReportId** value that is returned, submit the **getReport** operation, passing the **ReportId** value. Amazon returns the specified report.
5. Acknowledge receipt of the reports by calling the **updateReportAcknowledgements** operation. Specify the **ReportId** values for the reports that you want to acknowledge. Specify *true* for the **Acknowledged** parameter. Do this after downloading your reports to ensure that the **ReportId** values for these reports are not returned for your next call to the **getReportList** operation.

Note. Report scheduling is for seller-fulfilled orders only. For more information about tracking Fulfillment by Amazon orders, see Monitoring and tracking FBA orders in the Amazon MWS documentation.

Duplicate orders

Scheduling order reports helps to ensure that each generated order report (as identified by its **ReportId**) is returned only once. In most cases this means that individual orders (as identified by **OrderId**) will be returned only once. In rare cases, however, an order might be returned more than once. In addition, you should expect duplicate orders when you manually request order reports using the **RequestReport** operation. For these reasons you should design your automated systems to handle duplicate orders in ways that make sense for your situation.

The following flowchart shows the process for scheduling a report:



Schedule enumeration

The **Schedule** enumeration contains units of time by which you can schedule report requests. You specify a **Schedule** value when calling the **manageReportSchedule** operation. For more information, see Schedule order reports.

Schedule enumeration

Time period	Enumeration value
Every 15 minutes	_15_MINUTES_
Every 30 minutes	_30_MINUTES_
Every hour	_1_HOUR_
Every two hours	_2_HOURS_
Every four hours	_4_HOURS_
Every eight hours	_8_HOURS_
Every 12 hours	_12_HOURS_
Every day	_1_DAY_
Every two days	_2_DAYS_
Every three days	_72_HOURS_
Every week	_1_WEEK_
Every 14 days	_14_DAYS_
Every 15 days	_15_DAYS_
Every 30 days	_30_DAYS_
Delete a previously created report request schedule	_NEVER_

Report behavior

Requesting order reports

You can ask Amazon to generate an order report either by directly requesting the report with the **requestReport** operation (see Tutorial: Request a report) or by scheduling the report (see Schedule order reports). In either case, Amazon will only generate a report if there are orders to report.

In the case of directly requesting an order report using the **RequestReport** operation, if there were no orders placed in the time frame specified in your request, the operation returns *_DONE_NO_DATA_* to indicate there is no order report to return. In the case of scheduled order reports, if there were no orders placed between the time the last order report was generated and the current request, no report is generated.

For the list of order reports, see Order Reports in the Amazon MWS documentation.

Using Multiple MarketplaceId values when requesting a report

Behavior of reports when submitting multiple MarketplaceId values

When you specify multiple **MarketplaceId** values when submitting a report request, report processing behavior is more complex than when specifying a single **MarketplaceId** value. You can expect the following behavior when submitting a report request with multiple **MarketplaceId** values:

- If you specify more than one **MarketplaceId** value and one or more of those **MarketplaceId** values fail validation, for whatever reason, then then Amazon returns an error and the submission fails.
- If you don't specify a **MarketplaceId** value, reports that are not Listings Reports show all marketplaces the seller is registered in. You must specify a **MarketplaceId** value for Listings Reports.
- If a seller does not have a status of **Active** in a marketplace, only Settlement Reports are available for that marketplace. For example, if a seller's status in a marketplace is Pending because they have not completed the registration process, that seller would only be able to retrieve Settlement Reports.
- Note that Amazon validates a report request before it is queued for processing. A report request passes validation when it contains appropriate **MarketplaceId** values for the **ReportType** requested. Passing validation does not mean that the report will be processed successfully.

Behavior of Listings Reports when submitting multiple MarketplaceId values

Listings Reports can only provide information for one marketplace per request.

Behavior of Order Reports when submitting multiple MarketplaceId values

Flat file Order Reports can be used with the **ShowSalesChannel** parameter to show an additional column of sales channel information. For example: `&ReportOptions=ShowSalesChannel%3Dtrue`. For a list of Order Reports, see Order Reports in the Amazon Marketplace Web Service documentation.

Behavior of Order Tracking Reports when submitting multiple MarketplaceId values

Order Tracking Reports return orders from all of the marketplaces that the seller is registered in even if you specify only a subset of the marketplaces that they are registered in when you submit a report request. For a list of Order Tracking Reports, see Order Tracking Reports in the Amazon Marketplace Web Service documentation.

Behavior of Settlement Reports when submitting multiple MarketplaceId values

Settlement Reports can be retrieved regardless of the status of the seller in the marketplace. For a list of Settlement Reports, see Settlement Reports in the Amazon Marketplace Web Service documentation.

Best practices

Expect changes to reports

Amazon periodically adds new fields and field values to reports. Be sure that any report parsers that you build into your applications can gracefully handle these types of report updates.

Reports data types

The following data types are used in the Reports API:

Data type	Description
destination	The location of the report content.
EncryptionDetails	Encryption details required for decrypting the report content.
ReportInfo	Detailed information about a report.
ReportRequestInfo	Detailed information about a report request.
ReportSchedule	Detailed information about a report schedule.

destination

The location of the report content.

Name	Description	Required
channel	The distribution channel used to retrieve the report content. Type: string	Yes
url	The URL of the report content. Type: string	Yes

EncryptionDetails

Encryption details required for decrypting the report content.

Name	Description
standard	The encryption standard used to encrypt the report. AES implies AES256 with CBC (Cipher Block Chaining) Type: string
initializationVector	The vector to decrypt the content using Cipher Block Chaining (CBC). Type: string
key	The encryption key used to decrypt the content. Type: string

ReportInfo

Detailed information about a report.

Name	Description
ReportId	The report identifier. Type: string
ReportType	The report type identifier. See ReportType enumeration in the Amazon Marketplace Web Service documentation for a list of the available report types. Type: string
ReportRequestId	The report request identifier. Type: string
AvailableDate	The date that the report is available. In ISO 8601 date time format. Type: string
Acknowledged	A Boolean value that indicates if the report was acknowledged by this call to the **updateReportAcknowledgements** operation. The value is *true* if the report was acknowledged; otherwise *false*. Type: boolean
AcknowledgedDate	The date the report was acknowledged. In ISO 8601 date time format. Type: string

ReportRequestInfo

Detailed information about a report request.

Name	Description
ReportRequestId	The report request identifier. Type: string
ReportType	The report type identifier. See ReportType enumeration in the Amazon Marketplace Web Service documentation for a list of the available report types. Type: string
StartDate	The start of a date range used for selecting the data to report, in ISO 8601 date time format. Type: string
EndDate	The end of a date range used for selecting the data to report, in ISO 8601 date time format. Type: string
Scheduled	Indicates if a report is scheduled. When *true*, the report is scheduled. Type: boolean
SubmittedDate	The date when the report was submitted, in ISO 8601 date time format. Type: string
ReportProcessingStatus	The processing status of the report. Enum: _CANCELLED_, _DONE_, _IN_PROGRESS_, _SUBMITTED_, _DONE_NO_DATA_ Type: string
GeneratedReportId	A report identifier used to retrieve a report. Type: string
StartedProcessingDate	The date when the report processing started, in ISO 8601 date time format. Type: string
CompletedDate	The date when the report processing completed, in ISO 8601 date time format. Type: string

ReportSchedule

Detailed information about a report schedule.

Name	Description
ReportType	The report type. See ReportType enumeration in the Amazon Marketplace Web Service documentation for a list of the available report types. Type: string
Schedule	Indicates how often a report should be requested. See Schedule enumeration for values. Type: string
ScheduledDate	The date when the next report request is scheduled, in ISO 8601 date time format. Type: string