Amazon Marketplace Web Service

Developer Guide (Version 2009-01-01)

10/31/2010

Amazon Marketplace Web Service: Developer Guide (Version 2009-01-01)

Table of Contents

Welcome

Topics

This is the Amazon Marketplace Web Service Developer Guide. This section describes who should read this guide, how the guide is organized, and other resources related to Amazon Marketplace Web Service.

Amazon Marketplace Web Service will occasionally be referred to within this guide as simply "Amazon MWS"; all copyrights and legal protections still apply.

Audience

This guide is intended for developers who are building web-enabled applications that will use Amazon Marketplace Web Service to handle product listing, download orders for fulfillment, confirm shipments, and other management tasks.

To use Amazon MWS, you must have access to an eligible Amazon account and be signed up to use Amazon MWS and have your MWS account identifier and credentials. See the Authentication and Authorization (p. 25) topic for more information.

Required Knowledge and Skills

Use of this guide assumes you are familiar with the following:

- XML (For an overview, go to the *W3 Schools XML Tutorial*)
- Basic understanding of web services. For an overview, go to the *W3 Schools Web Services Tutorial*
- A programming language for consuming a Web service and any related tools.
- Basic knowledge about selling on Amazon. Comprehensive information about selling on Amazon is located at:

- DE: https://sellercentral.amazon.de/gp/help/
- FR: https://sellercentral.amazon.fr/gp/help/
- JP: https://sellercentral-japan.amazon.com/gp/help/home.html
- UK: https://sellercentral.amazon.co.uk/gp/help/
- US: *https://sellercentral.amazon.com/gp/help*

How This Guide Is Organized

This guide is organized into several major sections described in the table below.

Information	Relevant Sections
General information about Amazon Marketplace Web Service	Introduction to Amazon Marketplace Web Service (p. 6)
Introduces the key concepts for programming Amazon Marketplace Web Service including information about requests, authentication and authorization, and responses as well as detailed information, procedures, and examples for using the API	Programming Guide (p. 17)
Elemental reference to the Batch Data Exchange API	API Reference (p. 37)
Glossary of terms	Glossary (p. 116)
Index	Index (p. 118)

Related Resources

The table below lists related resources that you'll find useful as you work with Amazon MWS.

Description	Resource
MWS primary Web page for MWS registration, which also contains the MWS documentation, the client libraries, and FAQ.	• DE: *http://developer.amazonservices.de* • FR: *http://developer.amazonservices.fr* • JP: *http://developer.amazonservices.jp* • UK: *http://developer.amazonservices.co.uk* • US: *http://developer.amazonservices.com*
The Selling on Amazon Guide to XML contains important information about using XML for Amazon product feeds and reports.	https://images-na.ssl-images-amazon.com/images/G/01/rainier/help/XML_Documentation_Intl.pdf
Comprehensive information about selling on Amazon.	• DE: https://sellercentral.amazon.de/gp/help/ • FR: https://sellercentral.amazon.fr/gp/help/ • JP: https://sellercentral-japan.amazon.com/gp/help/home.html • UK: https://sellercentral.amazon.co.uk/gp/help/ • US: *https://sellercentral.amazon.com/gp/help*

Description	Resource
Community-based forum for developers to discuss technical questions related to Amazon MWS.	http://www.amazonsellercommunity.com/forums/forum.jspa?forumID=43

What's New

This What's New is associated with the 2009-01-01 release of Amazon Marketplace Web Service. This guide was last updated on 31 October 2010.

The following table describes the important changes to this document since the last release of the Amazon Marketplace Web Service Developer Guide.

Change	Description	Release Date
Eighth Release API Version 2009-01-01	Added four new report types for FBA to the ReportType (p. 95) topic. Also removed references to only Pro Merchants being eligible to register for Amazon MWS.	October 2010
Seventh Release API Version 2009-01-01	Rewrote several topics to describe the MWS registration process and the developer account identifier and credentials. Updated guide to refer to MWS developer account identifier. Added new report types for Product Ads and FBA.	June 2010
Sixth Release API Version 2009-01-01	Changed U.S. MWS Website address from http://mws.amazon.com to http://developer.amazonservices.com. Also changed the U.S. MWS endpoint from https://mws.amazonaws.com to https://mws.amazonservices.com.	March 2010
Fifth Release API Version 2009-01-01	Removed all references to Beta in this guide.	February 2010
Fourth Release Public Beta	Added new tab-delimited flat file FBA fulfilled shipments report: _GET_AMAZON_FULFILLED_SHIPMENTS_DATA_ .	January 2010
Third Release Public Beta	Added new report and feed types, updated API functions with throttling limits, updated sample code for listing functions, and other small changes.	December 2009

Change	Description	Release Date
Second Release Public Beta	Updated the ReportType (p. 95) topic with additional information.	July 2009
First Release Public Beta	This is the first public release of the Amazon Marketplace Web Service Developer Guide.	April 2009

Introduction to Amazon Marketplace Web Service

This section offers an introduction to Amazon Marketplace Web Service and the concepts you need to know to be successful.

What Is Amazon MWS?

Topics

This introduction to Amazon Marketplace Web Service is intended to give you a detailed summary. After reading this section, you should have a good idea of what Amazon MWS offers and how it can fit in with your business.

Overview of Amazon Marketplace Web Service

The Amazon MWS API functions are designed to facilitate and automate the stages of the business process for selling on Amazon. All eligible Amazon sellers can use Amazon MWS to manage their online business on Amazon.

Using Amazon MWS you can create applications that look up products for sale, download orders for fulfillment, confirm shipment, and schedule and receive reports. The Amazon MWS API functions are accessible by using a REST-like interface.

Amazon MWS is a secure environment that uses signatures for authentication and lets sellers delegate calling rights to developers by using the MWS authorization service. To use MWS with an Amazon Pro Merchant seller account, complete your registration at:

DE: *http://developer.amazonservices.de*
FR: *http://developer.amazonservices.fr*
JP: *http://developer.amazonservices.jp*
UK: *http://developer.amazonservices.co.uk*
US: *http://developer.amazonservices.com*

Amazon MWS provides the following major features:

- **Inventory Management—**You can perform batch uploads of inventory, add products, check inventory levels, examine pricing information, and other inventory management tasks.
- **Order Management—**You can download order information, obtain payment data, acknowledge and adjust orders, and schedule reports.
- **Reports—**You can use Amazon MWS to request generation of a variety of reports, you can query the status of these reports, and then download them.

Key Amazon Seller Concepts

Amazon sellers can list the items they have for sale directly on the product detail page where Amazon sells the same item new or they can create files that contain product information and upload the files to Amazon using Amazon MWS. Sellers can list their items in a variety of categories.

For additional information, see Related Resources (p. 2).

Sellers can make use of the inventory reporting functionality at Amazon. This feature allows sellers to download a text or XML version of their open listings, sold items, order fulfillment (items sold and paid via Amazon Payments), and listings canceled by Amazon. Sellers can choose date ranges of the last 15, 30, or 60 days for sold listings and order fulfillment data.

What Is a Feed?

Feeds are used to exchange data files with Amazon. To use Amazon MWS successfully, it is important that you understand the types of feeds and reports that Amazon supports. Data files come in two types: text files, which you build from Excel spreadsheets, and XML documents. Seller Central contains templates and examples to help you build these files.

For additional information, see Related Resources (p. 2).

Note

Whether a seller receives reports in XML or flat file format is determined by the seller account configuration. By default, all reports will be in flat file format. In order to receive XML reports, contact Amazon by going to the appropriate site for your locale:

- DE: https://sellercentral.amazon.de/gp/help/
- FR: https://sellercentral.amazon.fr/gp/help/
- JP: https://sellercentral-japan.amazon.com/gp/help/home.html
- UK: https://sellercentral.amazon.co.uk/gp/help/
- US: *https://sellercentral.amazon.com/gp/help*

There are two main categories of feeds:

- Product Feeds (p. 9)
- Order Feeds (p. 9)

Product Feeds

Sellers use SubmitFeed (p. 39) to manage their inventory on Amazon by using a combination of each of the following product-related feeds.

- Product feed, which is the primary feed for products, and contains the core information about each product (such as the title, SKU, description, and other elements).
- Inventory feed, which shows the current inventory status of the products in the product feed.
- Overrides feed, which lets you set your shipping charges for the products in the product feed.
- Pricing feed, which sets the current price for the products in your product feed.
- Product image feed, which provides a link to your local store of product images.
- Relationships feed, which establishes the parent-and-child relationship between products in your product feed.

Order Feeds

Sellers can use Amazon MWS to manage all aspects of their orders on Amazon. By using the MWS reporting functionality, sellers can download order reports. More information about using MWS to manage orders is found in the topic, Managing Orders (p. 13). For information about requesting reports, see the topic, Managing Reports (p. 15).

By calling `SubmitFeed`, sellers can upload communications to Amazon about their orders.

MWS supports the following order report (p. 95)and feed (p. 93) types.

- Order report feed, which contains a list of all the orders of the seller's products since the last order report was created.

- Order acknowledgement feed, which allows a seller's system to update the Amazon e-commerce system regarding acceptance or rejection of individual orders for further processing.
- Order adjustment feed, which allows the seller to inform Amazon of any changes to be made to a transaction.
- Order fulfillment feed, which allows the seller to inform Amazon regarding order fulfillment status.
- Processing feed, which provides sellers with information about the status of a specified feed.
- Fulfillment Center feed, which accepts data from sellers about their store and/or warehouse.
- Settlement report, which Amazon provides to the seller that lists payment events related to each item in a customer order, including all subsequent adjustment events.

Managing Inventory

You can use Amazon MWS to upload your inventory to Amazon for products that already exist in Amazon's catalog. For products that do not exist in Amazon's catalog, first add those products to the catalog by creating a new product detail page.

With Amazon MWS, you can add new items to your existing available inventory on Amazon, or modify current items, and also delete items from your inventory. You can use Amazon MWS to add photos to your products and manage variations such as different colors and sizes for your products. You can perform batch uploads of inventory, check inventory levels, examine pricing information, and other inventory management tasks.

You need to use either flat file templates or XML feeds to send batch files to Amazon using Amazon MWS. Flat file templates can be manually edited using Microsoft Excel and other spreadsheet applications. They can also be automatically generated using software, as long as it follows the same tab-separated layout as dictated by the downloadable templates.

For additional information, see Related Resources (p. 2).

Note

Whether a seller receives reports in XML or flat file format is determined by the seller account configuration. By default, all reports will be in flat file format. In order to receive XML reports, contact Amazon by going to the appropriate site for your locale:

- DE: https://sellercentral.amazon.de/gp/help/
- FR: https://sellercentral.amazon.fr/gp/help/
- JP: https://sellercentral-japan.amazon.com/gp/help/home.html
- UK: https://sellercentral.amazon.co.uk/gp/help/
- US: *https://sellercentral.amazon.com/gp/help*

In order to create an XML file that will process successfully, you must include the information contained within the core schemas. XML feeds need to validate against the XML schemas, which are provided in the forms of XSDs. There are different XSDs for different categories; however, all three of the following core schemas are required:

- **Base—**All other XSDs reference the base XSD elements and data types.
- **Envelope—**Used to wrap all other data with message-level protocol data.

  The envelope consists of a header and one or more messages, each of which contains the specified data object. While an envelope can contain more than one message, each message in the same envelope must be of the same type, as specified by the message type element.
- **Header—**Used by the envelope to specify universal data related to the feed or message in the feed.

  Versioning of feeds is controlled by the *`DocumentVersion`* element, and the seller for a given feed is identified by the *`MerchantId`*, which is assigned by Amazon when the seller account is established.

Use SubmitFeed (p. 39) to upload your batch file as an XML or flat file document, specifying the FeedType (p. 93) in your call.

When you call `SubmitFeed`, Amazon returns a *`FeedProcessingId`* so you can periodically check the status of the upload using GetFeedSubmissionList (p. 45). When the MWS system finishes processing your batch file, you can call GetFeedSubmissionResult (p. 55) to get a processing report that describes which records in the file were successful and which records generated errors. If there are errors in the batch file, you can fix the records that had errors and upload changes.

Uploading Inventory Files

1	To create a new inventory catalog, or to update existing inventory items, the seller posts a file containing inventory information to Amazon using the `SubmitFeed` operation.
2	When Amazon receives the file, the *`FeedSubmissionId`* is returned.
3	Using the *`FeedSubmissionId`* to identify the file, the seller periodically checks the processing status using the *`GetFeedSubmissionList`* operation.
4	If Amazon is still processing the request, the *`FeedProcessingStatusList`* parameter of the *`GetFeedSubmissionList`* operation returns a status of `_IN_PROGRESS_`. If the processing is complete, a status of `_DONE_` is returned.
5	The seller calls the *`GetFeedSubmissionResult`* operation to determine whether there are errors in the transmission.
6	Amazon returns the processing report.
7	The seller analyzes the processing report.
8	If the processing report indicates there were errors in the transmission, the seller corrects the errors and again calls `SubmitFeed`, repeating the entire process again.
9	When the processing report is error-free, the transmission is complete.

Related Topics

- SubmitFeed (p. 39)

Managing Orders

When a customer places an order on Amazon for an item that a seller has available in their online inventory, the seller's available inventory is decremented by the quantity of the item that the customer ordered. After the order, Amazon performs a series of checks. If the order passes all checks then it is forwarded to the seller, otherwise the order is canceled and the inventory levels are adjusted accordingly. If the order goes through, the following list describes the order processing steps.

Order processing steps

1. **Receive the order—**Amazon generates an order report. You can request the report, or schedule order reports for regular processing.
2. **Acknowledge receipt of the order—**This step acknowledges that the seller has received the order; it does not mean the order has been fulfilled yet. During this step, sellers can cancel the order or simply acknowledge it.
3. **Ship and confirm the order—**Once the order has been picked, packed, and shipped, notify Amazon by sending a shipping confirmation, which signals Amazon to complete the financial transaction and notify the buyer that the item(s) are on the way. If Amazon does not receive the confirmation within 30 days of the order being placed, the order will be canceled automatically and the seller will not be paid.
4. **Adjust the order—**Sometimes it is necessary to issue a refund for all or part of an order, or process a return.
5. **Get paid—**Receiving payment is the final part of the order cycle. Amazon generates a settlement report, which provides information about all financial tractions for the settlement period. Depending on your seller agreement, settlement may be as frequent as every week, but for most sellers settlement reports are automatically generated every 14 days. The settlement report is a detailed reconciliation of all orders and adjustments for the settlement period. The report can also serve as the detail (or receipt) of the actual deposit made to the Seller’s bank account. Sellers can expect to receive a deposit to their banking account 3-4 days after the settlement report was generated.

You can use Amazon MWS to download order information, acknowledge orders, obtain payment data, and schedule reports. You can use Amazon MWS to request generation of order fulfillment reports, query the status of these reports, and then download them. Order fulfillment reports contain information about the orders for your items. Depending on its type, these reports can be regularly scheduled or manually generated.

Use the RequestReport (p. 57) operation to request the generation of your report. Specify the type of report you want generated by choosing the appropriate ReportType (p. 95). For example, use `_GET_FLAT_FILE_ORDERS_DATA_` to generate an order fulfillment report, or use `_GET_FLAT_FILE_OPEN_LISTINGS_DATA_` to get an open listings report.

For additional information, see Related Resources (p. 2).

Note

Whether a seller receives reports in XML or flat file format is determined by the seller account configuration. By default, all reports will be in flat file format. In order to receive XML reports, contact Amazon by going to the appropriate site for your locale:

- DE: https://sellercentral.amazon.de/gp/help/
- FR: https://sellercentral.amazon.fr/gp/help/
- JP: https://sellercentral-japan.amazon.com/gp/help/home.html
- UK: https://sellercentral.amazon.co.uk/gp/help/
- US: *https://sellercentral.amazon.com/gp/help*

You'll receive a response that includes the processing status of the report. You can continue to request the processing status, using GetReportRequestList (p. 59), which returns a list of report request IDs and the status of the reports.

To retrieve your report, use the *`GetReportList`* operation, which returns a list of report IDs of all the reports that are available for download. You can constrain that list by specifying parameters such as the report request ID, or the report request date, or the type of report. Use the GetReport (p. 79) operation, including the *`ReportId`*, which returns the contents of your report.

Managing Reports

You can use Amazon MWS to request order fulfillment and open listings reports, you can query the status of these reports, and then download them. Order fulfillment reports contain information about the orders for your items. Open listings reports contain information about your listings on Amazon. These reports can be regularly scheduled or manually generated.

Use the RequestReport (p. 57) operation to request the generation of your report. Specify the type of report you want generated by choosing the appropriate ReportType (p. 95). For example, use `_GET_FLAT_FILE_ORDERS_DATA_` to generate an order fulfillment report, or use `_GET_FLAT_FILE_OPEN_LISTINGS_DATA_` to get an open listings report.

For additional information, see Related Resources (p. 2).

Note

Whether a seller receives reports in XML or flat file format is determined by the seller account configuration. By default, all reports will be in flat file format. In order to receive XML reports, contact Amazon by going to the appropriate site for your locale:

- DE: https://sellercentral.amazon.de/gp/help/
- FR: https://sellercentral.amazon.fr/gp/help/
- JP: https://sellercentral-japan.amazon.com/gp/help/home.html
- UK: https://sellercentral.amazon.co.uk/gp/help/
- US: *https://sellercentral.amazon.com/gp/help*

You'll receive a response that includes the processing status of the report. You can continue to request the processing status, using GetReportRequestList (p. 59), which returns a list of report request IDs and the status of the reports.

To retrieve your report, use the GetReportList (p. 70) operation, which returns a list of report IDs of all the reports that are available for download. You can constrain that list by specifying parameters such as the report request ID, or the report request date, or the type of report. Use the GetReport (p. 79) operation, including the *ReportId*, which returns the contents of your report.

Requesting Reports

1	To create a request that a report be generated, the seller calls the *RequestReport* operation.
2	When Amazon receives the report request, the *ReportRequestId* is returned.
3	Using the *ReportRequestId* to identify the report, the seller periodically checks the processing status using the *GetReportRequestList* operation.
4	If Amazon is still processing the request, the *ReportProcessingStatusList* parameter of the *GetReportRequestList* operation returns a status of `_IN_PROGRESS_`. If the processing is complete, a status of `_DONE_` is returned.
5	The seller calls the *GetReportList* operation to get a list of the available reports, using the `ReportRequestId` parameter to limit the number of reports returned in the list.
6	Amazon returns the list of available reports that match the query parameters.
7	The seller calls the *GetReport* operation to return the contents of the report, using the *ReportId* to limit which reports are returned.

8	Amazon returns the contents of the report, or multiple reports if more than one *`ReportId`* was listed in the *`GetReport`* operation.
9	The report request is complete.

Programming Guide

Topics

This section offers information about how to formulate requests and receive responses with Amazon MWS, including detailed instructions to help you with authorization and authentication, and procedures on how to use the API.

Understanding Requests and Responses

Topics

This section addresses how you formulate API requests. After reading this section, you should understand the components and requirements of a request, know how to authenticate a request, and understand the content of responses.

Requests

Amazon MWS supports Query requests for calling service actions. Query requests are simple HTTP requests, using the GET or POST method with query parameters in the URL or HTTP body, respectively. We require HTTPS in order to prevent third-party eavesdropping on your communication with Amazon.

Each of the HTTP header lines must be terminated with a carriage return and a line feed. For more information, go to Section 2.2 of the *HTTP/1.1 Specification*. Query requests must contain an *Action* parameter to indicate the action to be performed. The response is an XML document.

Amazon MWS limits calls to 1,000 total calls per hour per the combination of developer account and Amazon seller account. Also, some of the MWS API functions have additional throttles, which are detailed in each description of the function as well as shown in the topic, MWS Account Throttling (p. 22) .

Structure of a Query Request

Amazon MWS Query requests are URLs. The URL consists of:

- **Endpoint—**The domain name of the request, such as `https://mws.amazonservices.com/`.

  For a list of endpoints for each Amazon marketplace, see MWS Endpoints (p. 21).
  After the endpoint is a question mark (?), which separates the endpoint from the parameters.
- **Action—**The action you want to perform on the endpoint, such as *GetFeedSubmissionResult*.
- **Parameters—**Any request parameters.

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.

Example Query Request

Because the format is hard to read if left complete as a long string, (and may be cut off when viewing in certain formats), and although the Amazon MWS client's actual query string must not contain any line breaks or white spaces between the query string parameters, all Query examples in this guide are presented in the following parsed format.

The following is an example Query request that downloads a document created as the result of submitting a feed.

```
https://mws.amazonservices.com/
      ?AWSAccessKeyId=0PExampleR2
      &Action=GetFeedSubmissionResult
      &FeedSubmissionId=20Example76
      &Marketplace=ATExampleER
      &Merchant=A1ExampleE6
      &SignatureVersion=2
      &Signature=CNExampleQ%3D
      &Timestamp=2009-02-04T17%3A44%3A33.500Z
      &Version=2009-01-01
```

MWS Endpoints

Amazo Market	MWS Endpoint	MWS Website
DE	https://mws.amazonservices.de	http://developer.amazonservices.de
FR	https://mws.amazonservices.fr	http://developer.amazonservices.fr
JP	https://mws.amazonservices.jp	http://developer.amazonservices.jp
UK	https://mws.amazonservices.co.uk	http://developer.amazonservices.co.uk
US	https://mws.amazonservices.com	http://developer.amazonservices.com

MWS Account Throttling

Amazon MWS limits calls to 1,000 total calls per hour per the combination of a developer account and Amazon seller account. Also, some of the MWS API functions have additional throttles, as shown in the following table.

Function Name	Throttling Limit
SubmitFeed (p. 39)	30 requests per hour, also subject to overall limit of 1,000 per account per hour
GetFeedSubmissionList (p. 45)	1 request per minute, also subject to overall limit of 1,000 per account per hour
GetFeedSubmissionListByNextToken (p. 48)	Unlimited other than being subject to overall limit of 1,000 per account per hour
GetFeedSubmissionCount (p. 50)	1 request per minute, also subject to overall limit of 1,000 per account per hour
CancelFeedSubmissions (p. 52)	1 request per minute, also subject to overall limit of 1,000 per account per hour
GetFeedSubmissionResult (p. 55)	60 requests per hour, also subject to overall limit of 1,000 per account per hour
RequestReport (p. 57)	30 requests per hour, also subject to overall limit of 1,000 per account per hour
GetReportRequestList (p. 59)	1 request per minute, also subject to overall limit of 1,000 per account per hour
GetReportRequestListByNextToken (p. 62)	Unlimited other than being subject to overall limit of 1,000 per account per hour
GetReportRequestCount (p. 65)	1 request per minute, also subject to overall limit of 1,000 per account per hour
CancelReportRequests (p. 67)	1 request per minute, also subject to overall limit of 1,000 per account per hour
GetReportList (p. 70)	1 request per minute, also subject to overall limit of 1,000 per account per hour
GetReportListByNextToken (p. 73)	Unlimited other than being subject to overall limit of 1,000 per account per hour
GetReportCount (p. 76)	1 request per minute, also subject to overall limit of 1,000 per account per hour
GetReport (p. 79)	60 requests per hour, also subject to overall limit of 1,000 per account per hour
ManageReportSchedule (p. 81)	1 request per minute, also subject to overall limit of 1,000 per account per hour
GetReportScheduleList (p. 83)	1 request per minute, also subject to overall limit of 1,000 per account per hour
GetReportScheduleListByNextToken (p. 85)	Unlimited other than being subject to overall limit of 1,000 per account per hour

Function Name	Throttling Limit
GetReportScheduleCount (p. 87)	1 request per minute, also subject to overall limit of 1,000 per account per hour
UpdateReportAcknowledgements (p. 89)	1 request per minute, also subject to overall limit of 1,000 per account per hour

Responses

Response Messages

In response to an action request, Amazon MWS returns an XML data structure or a flat file that contains the results of the request.

The Structure of a Response

If a request is successful, the response will be returned with the data requested.

The following example shows a successful response.

```
<?xml version="1.0"?>
<RequestReportResponse xmlns="http://mws.amazonaws.com/doc/2009-01-01/">
  <RequestReportResult>
    <ReportRequestInfo>
      <ReportRequestId>2291326454</ReportRequestId>
      <ReportType>_GET_MERCHANT_LISTINGS_DATA_</ReportType>
      <StartDate>2009-01-21T02:10:39+00:00</StartDate>
      <EndDate>2009-02-13T02:10:39+00:00</EndDate>
      <Scheduled>false</Scheduled>
      <SubmittedDate>2009-02-20T02:10:39+00:00</SubmittedDate>
      <ReportProcessingStatus>_SUBMITTED_</ReportProcessingStatus>
    </ReportRequestInfo>
  </RequestReportResult>
  <ResponseMetadata>
    <RequestId>88faca76-b600-46d2-b53c-0c8c4533e43a</RequestId>
  </ResponseMetadata>
</RequestReportResponse>
```

If a request is unsuccessful, the main response element is named `ErrorResponse`, irrespective of the action requested. This element contains one or more `Error` child elements. Each `Error` includes:

- An error code that identifies the type of error that occurred
- A message code that describes the error condition in a human-readable form
- An error type, identifying either the receiver or the sender as the error originator

The following example shows an error response.

```
<ErrorResponse xmlns="http://mws.amazonaws.com/doc/2009-01-01/">
    <Error>
        <Type>Sender</Type>
        <Code>InvalidClientTokenId</Code>
        <Message>The AWS Access Key Id you provided does not exist in our
 records.</Message>
        <Detail>com.amazonaws.mws.model.Error$Detail@17b6643</Detail>
    </Error>
    <RequestID>b7afc6c3-6f75-4707-bcf4-0475ad23162c</RequestID>
</ErrorResponse>
```

Authentication and Authorization

Topics

This section describes authentication, which is the process for identifying and verifying who is sending a request, and authorization, which determines whether you have the credentials to perform a specific action.

What Is Authentication?

Authentication is a process for identifying and verifying who is sending a request. The following steps outline the general authentication process.

General Process of Authentication

1	The sender obtains the necessary credential.
2	The sender sends a request with the credential to the recipient.
3	The recipient uses the credential to verify the sender truly sent the request.
4	If yes, the recipient processes the request. If no, the recipient rejects the request and responds accordingly.

During authentication, Amazon Marketplace Web Service verifies both the identity of the sender and whether the sender is registered to use services offered by Amazon MWS. If either test fails, the request is not processed further.

For further discussion of authentication, go to the techencylopedia.com entry for authentication. For definitions of common industry terms related to authentication, go to the RSA Laboratories Glossary.

MWS Credentials

When you register for MWS you are issued a pair of developer credentials, which you need to make successful requests to Amazon MWS, as well as a developer account identifier, which is necessary for you to retain for future use. If you are developing MWS applications or otherwise providing MWS-related development services to other sellers, you will need to provide this developer account identifier to those sellers so that they can authorize you to access their Amazon seller accounts with MWS.

These are your MWS developer credentials.

- Access Key ID (a 20-character, alphanumeric sequence)
  For example: 022QF0EXAMPLEH9DHM02
- Secret Access Key (a 40-character sequence)
  For example: kWcrlEXAMPLEM/LtmEENI/aVmYvHNif5zB+d9+ct

⚠ **Caution**

Your Secret Access Key is a secret that only you and MWS should know. It is important to keep it confidential to protect your account. Never include it in your requests to MWS, and never e-mail it to anyone. Do not share it outside your organization, even if an inquiry appears to come from MWS or anyone else at Amazon. No one who legitimately represents Amazon will ever ask you for your Secret Access Key.

The Access Key ID is associated with your MWS registration. You include it in Amazon MWS requests to identify yourself as the sender of the request.

The Access Key ID is not a secret, and anyone could use your Access Key ID in requests to Amazon MWS. To provide proof that you truly are the sender of the request, you must also include a digital signature. For all requests except those generated using the Amazon MWS client libraries, you calculate the signature using your Secret Access Key. MWS uses the Access Key ID in the request to look up your Secret Access Key and then calculates a digital signature with the key. If the signature MWS calculates matches the signature you sent, the request is considered authentic. Otherwise, the request fails authentication and is not processed.

Viewing Your MWS Credentials and Identifier

Your Access Key ID, Secret Access Key, and Account Number are displayed to you when you register to use MWS. They are not e-mailed to you. You should print this page or save it to your hard drive.
If you need to see the credentials and identifier again, you can repeat the MWS registration process. Your developer account credentials and identifier are displayed on the final page of the registration process.

Calculating Signatures

The Amazon Marketplace Web Service client libraries calculate signatures for you. If you are using one of our libraries and sending in messages, you do not need to calculate your signature or time stamp.

When accessing Amazon MWS, you must provide the following items so the request can be authenticated:

- **AWSAccessKeyId—**Your MWS account is identified by your Access Key ID, which MWS uses to look up your Secret Access Key.
- **Signature—**Each request must contain a valid request signature, or the request is rejected. A request signature is calculated using your Secret Access Key, which is a shared secret known only to you and MWS.
- **Signature Version—**Which signature version is being used. This is MWS-specific information that tells MWS the algorithm you used to form the string that is the basis of the signature. For Amazon MWS, this will always be *`SignatureVersion=2`*.
- **SignatureMethod—**Which HMAC hash algorithm is being used to calculate your signature, either SHA256 or SHA1.
- **Timestamp—**Each request must contain the time stamp of the request. Depending on the API function you're using, you can provide an expiration date and time for the request instead of or in addition to the time stamp. See the particular API function for details of what is required and allowed for that function.

Following is the series of tasks required to authenticate requests to MWS. It is assumed you have already registered for MWS and received an Access Key ID and Secret Access Key.

You perform the first three tasks.

Process for Authentication: Tasks You Perform

1	Construct a request to MWS.
2	Calculate a keyed-hash message authentication code (HMAC-SHA) signature using your Secret Access Key (for information about HMAC, go to http://www.faqs.org/rfcs/rfc2104.html)
3	Include the signature and your Access Key ID in the request, and then send the request to MWS.

MWS performs the next three tasks.

Process for Authentication: Tasks MWS Performs

4	MWS uses the Access Key ID to look up your Secret Access Key.
5	MWS generates a signature from the request data and the Secret Access Key using the same algorithm you used to calculate the signature you sent in the request.
6	If the signature generated by MWS matches the one you sent in the request, the request is considered authentic. If the comparison fails, the request is discarded, and MWS returns an error response.

Calculating a Signature

The request must include an HMAC-SHA signature. The signature is used as the value for the *`Signature`* parameter in the request URL being constructed.

The string you use to compute the HMAC signature is constructed using the method described in the following procedure.

To create the signature

1. Create the canonicalized query string that you need later in this procedure:
   a. Sort the UTF-8 query string components by parameter name with natural byte ordering. The parameters can come from the GET URI or from the POST body (when Content-Type is application/x-www-form-urlencoded).
   b. URL encode the parameter name and values according to the following rules:
      - Do not URL encode any of the unreserved characters that RFC 3986 defines. These unreserved characters are A-Z, a-z, 0-9, hyphen (-), underscore (_), period (.), and tilde (~).
      - Percent encode all other characters with %XY, where X and Y are hex characters 0-9 and uppercase A-F.
      - Percent encode extended UTF-8 characters in the form %XY%ZA....
      - Percent encode the space character as %20 (and not +, as common encoding schemes do).
   c. Separate the encoded parameter names from their encoded values with the equals sign (=) (ASCII character 61), even if the parameter value is empty.
   d. Separate the name-value pairs with an ampersand (&) (ASCII code 38).
2. Create the string to sign according to the following pseudo-grammar (the "\n" represents an ASCII newline).

```
StringToSign = HTTPVerb + "\n" +
               ValueOfHostHeaderInLowercase + "\n" +
               HTTPRequestURI + "\n" +
               CanonicalizedQueryString <from the preceding step>
```

   The HTTPRequestURI component is the HTTP absolute path component of the URI up to, but not including, the query string. If the HTTPRequestURI is empty, use a forward slash (/).
   The following example shows a string to sign.

```
POST
mws.amazonservices.com
/
AWSAccessKeyId=0PExampleR2
&Action=SubmitFeed
&FeedType=_POST_INVENTORY_AVAILABILITY_DATA_
&Marketplace=ATExampleER
&Merchant=A1ExampleE6
&SignatureVersion=2
&Timestamp=2009-08-20T01%3A10%3A27.607Z
&Version=2009-01-01
```

3. Calculate an RFC 2104-compliant HMAC with the string you just created, your Secret Access Key as the key, and SHA256 or SHA1 as the hash algorithm.
4. Convert the resulting value to base64.

5. Use the resulting value as the value of the *`Signature`* request parameter.

About the Time Stamp

The time stamp (or expiration time) you use in the request must be a `dateTime` object. For more information, go to http://www.w3.org/TR/xmlschema-2/#dateTime. Although it is not required, we recommend that you provide the time stamp in the Coordinated Universal Time (Greenwich Mean Time) time zone. For example: "2009-03-03T18:12:22Z" or "2009-02-23T18:12:22.093-07:00". The *`Timestamp`* field must contain the client's machine time in ISO8601 format; requests with a time stamp significantly different (15 minutes) than the receiving machine's clock will be rejected to help prevent replay attacks.

Network Time Protocol (NTP) is a protocol designed to synchronize the clocks of computers over a network. For more information, go to http://www.ntp.org/.

Every request from Amazon MWS will include a *`Date`* header in its HTTP response that you can use to check whether your local machine's time matches our server's time: `Date: Tue, 24 Mar 2009 20:34:28 GMT`

You can also load https://mws.amazonservices.com/ in any Web browser (no *`Signature`* is required) to check on the MWS server time:

```
<?xml version="1.0"?>
<PingResponse>
<Timestamp timestamp="2009-03-24T20:29:19:22Z"/>
</PingResponse>
```

In order to allow us to extend the content of the `PingResponse`, any software you write to parse out the *`Timestamp`* should not break if sibling XML tags start to appear. Generally, you should ignore unknown tags in any XML we send you, as per the Web architectural principle in Section 5.2 of http://www.w3.org/TR/webarch/.

If you specify a time stamp (instead of an expiration time), the request automatically expires 15 minutes after the time stamp (in other words, MWS does not process a request if the request time stamp is more than 15 minutes earlier than the current time on MWS servers). Make sure your server's time is set correctly.

Important

If you are using .NET you must not send overly specific time stamps, due to different interpretations of how extra time precision should be dropped. To avoid overly specific time stamps, manually construct `dateTime` objects with no more than millisecond precision.

Authorization

Authentication determines who you are, while authorization determines whether you are allowed to perform a specific action. Authentication is performed using your Access Key ID to locate your Secret Key, which you've used to create your signature. Authorization is determined by whether you've registered to use MWS. Only a request that is successfully identified as originating from a developer who has registered to use MWS will be successfully processed.

Every request to MWS must contain:

- **Access Key ID**
- **Merchant ID**
- **Marketplace ID**

The Merchant ID is used to specify which seller a request is intended to act on behalf of. Again, the MWS developer account must be authorized for this seller in order for the request to succeed.

The Marketplace ID is the logical location a seller's online business is registered in. In order for an MWS request to pass authorization, both the Merchant ID and Marketplace ID are required in every request.

Requests to MWS are authenticated and authorized by verifying information contained within the request. This verification is performed using the information in the following table, which *must* be included in every Amazon MWS request.

Name	Description	Required
AWSAccessKeyId	The sender's developer account is identified by the Access Key ID, which is used to look up the Secret Access Key. Type: xs:string	Yes
Signature	Each request to a web service that requires authenticated requests must contain a valid request signature, or the request is rejected. A request signature is calculated using the Secret Access Key that has been assigned to the developer's account, which is a shared secret known only to MWS and the developer. Type: xs:string	Yes
SignatureVersion	Specifies the version you want to use to calculate your signature. Version 2 is the only supported version. Type: xs:string	Yes
SignatureMethod	Specifies the method used to calculate your signature. Can be either HmacSHA256 or HmacSHA1. Type: xs:string	Yes
Timestamp	The timestamp of when the request is sent. Used to ensure old requests cannot be played back by a malicious third party. Type: xs:datetime	Yes

Name	Description	Required
Merchant	The Merchant ID that represents the specific seller account to which the API action should be applied. Type: xs:string	Yes
Marketplace	The Marketplace ID for the specific marketplace to which the API action should be applied. Type: xs:string	Yes
Version	The specific version of Amazon MWS being called. This must always be set to `2009-01-01`, although we may support additional versions in the future. Type: xs:string	Yes

Example Query Snippet

```
?AWSAccessKeyId=0PExampleR2
&Signature=0RExample0%3D
&SignatureVersion=2
&SignatureMethod=HmacSHA256
&Timestamp=2009-02-04T17%3A34%3A14.203Z
&Merchant=A1ExampleE6
&Marketplace=ATExampleER
&Version=2009-01-01
```

Required Headers

Topics

This section explains how to formulate the User-Agent field in the HTTP header, which is a required field for all MWS calls, and the Content-MD5 HTTP header, required for all calls to the SubmitFeed (p. 39) function.

User-Agent Header

The User-Agent field in the HTTP header is a required field for all MWS calls, and is used to identify your application, its version number, and programming language. The User-Agent header information enables us to identify problems with particular applications, application versions, and programming languages.

In the future, the User-Agent field will also allow us to give you periodic summary information regarding your Amazon MWS usage, broken out by your application identifiers, version identifiers, programming languages, and by other tags of your choice that you supplied in the User-Agent header. (This reporting functionality is not yet available, but is planned for a later version of Amazon MWS.)

The Amazon MWS client libraries provide an easy-to-use method for passing the User-Agent parameter with every MWS request. When you initialize the client library, you add the Application or Company Name and the Version Number. Other HTTP libraries also provide easy methods for constructing User-Agent headers, but if you have any difficulties with this requirement, please request assistance. You can contact us by using the contact form at:

DE: *http://developer.amazonservices.de*
FR: *http://developer.amazonservices.fr*
JP: *http://developer.amazonservices.jp*
UK: *http://developer.amazonservices.co.uk*
US: *http://developer.amazonservices.com*

To meet the requirements, begin with the name of your application, followed by a forward slash, followed by the version of the application, followed by a space, an opening parenthesis, the *Language* name value pair, and a closing paranthesis. The *Language* parameter is a required attribute, but you can add additional attributes separated by semi-colons.

The following example illustrates a minimally acceptable User-Agent header.

```
AppId/AppVersionId (Language=LanguageNameAndOptionallyVersion)
```

If you are a third-party application integrator, you might want a User-Agent header like the following.

```
My Desktop Seller Tool/2.0 (Language=Java/1.6.0.11; Platform=Windows/XP)
```

If you are a large seller who is integrating through your own IT department, you might want a User-Agent header like the following, so we could help you troubleshoot using the *Host* attribute.

```
MyCompanyName/build1611 (Language=Perl; Host=jane.laptop.example.com)
```

To specify additional attributes, use the format *AttributeName*=*Value;*, separating each name value pair with a semi-colon. Should you wish to use a backslash (\), quote it with another backslash (\\). Similarly, quote a forward slash in the application name (\/), an opening parenthesis in the application version (|(), an equal sign in the attribute name (\=), and both a closing parenthesis (\)), and a semicolon (\;) in attribute values.

Tip

Because the User-Agent header is transmitted in every request, we suggest not exceeding 200 characters so you don't consume unnecessary bandwidth. We will reject a User-Agent header only if it is longer than 500 characters, but suggest you stay far below this hard limit.

Using the Content-MD5 Header with SubmitFeed

For the SubmitFeed (p. 39) function, we require that you pass the Content-MD5 HTTP header, which contains the MD5 hash of the HTTP entity body (see Section 14.15 of RFC 2616, the HTTP/1.1 specification), so we can check if the feed we stored for processing is bit for bit identical with what you sent, protecting you from corrupted descriptive or pricing product data appearing on Amazon.

The MWS client libraries provide an easy-to-use method for passing the Content-MD5 header with every MWS request, as long as you send data that has first been stored on your disk.

Your software should be prepared to transmit a feed that is larger than your available computer memory, even though it might fit into main memory now. You can work around this barrier by the following process.

Transmitting Large Feeds

1	Store a feed on disk before transmitting it to us.
2	Compute the Content-MD5 of the file and store it in a companion file.
3	Construct a SubmitFeed (p. 39) call, pass in the stored Content-MD5, and attach the file contents in a streaming fashion.

The following Java code sample illustrates how to compute the Content-MD5:

```
   /**
    * Consume the stream and return its Base-64 encoded MD5 checksum.
    */
   public static String computeContentMD5Header(InputStream inputStream) {

       // Consume the stream to compute the MD5 as a side effect.
       DigestInputStream s;
       try {
           s = new DigestInputStream( inputStream,
MessageDigest.getInstance("MD5"));

           // drain the buffer, as the digest is computed as a side-effect
           byte[] buffer = new byte[8192];
           while(s.read(buffer) > 0);

           return new String(

org.apache.commons.codec.binary.Base64.encodeBase64(s.getMessageDigest().digest()),
                   "UTF-8");
       } catch (NoSuchAlgorithmException e) {
           throw new RuntimeException(e);
       } catch (IOException e) {
           throw new RuntimeException(e);
       }
   }
```

Related Topics

- SubmitFeed (p. 39)

API Reference

Amazon Marketplace Web Service contains one set of API functions: Batch Data Exchange, for inventory and order management as well as reporting and analytical tasks.

Batch Data Exchange API

Topics

The Batch Data Exchange API provides the services for you to upload inventory and order data to Amazon, and to request reports about your inventory and orders.

You can submit an XML or flat file document together with related metadata, such as the FeedType (p. 93), to Amazon using SubmitFeed (p. 39).

Amazon returns a *`FeedProcessingId`* that allows you to periodically check the status of the feed, using GetFeedSubmissionList (p. 45) or GetFeedSubmissionCount (p. 50).

When the feed system finishes processing a feed, you will receive a processing report that describes which records in the feed were successful and which records generated errors. You can either check the status of multiple batches, or retrieve error logs and/or quick fix files for specific batch IDs.

Operations

Topics

The Batch Data Exchange operations provide the services for you to upload data to Amazon and to request reports.

You can submit an XML or flat file document together with related metadata, such as the *`FeedType`* to Amazon.

When the feed system finishes processing a feed, you'll receive a processing report that describes which records in the feed were successful and which generated errors.

SubmitFeed

Description

The `SubmitFeed` operation uploads a file for processing together with the necessary metadata to process the file.

Amazon MWS limits calls to 1,000 total calls per hour per seller account. For best performance, you should limit your calls to `SubmitFeed` to no more than three feeds per hour per seller account, although you can successfully call `SubmitFeed` up to 30 times per hour. Feed size is limited to 2,147,483,647 bytes (2^31 -1) per feed. If you have a large amount of data to post, however, we recommend when possible that you submit feeds smaller than this limit; submit feeds when you have 30,000 records/items or four hours have passed since your last submittal, whichever comes first. This ensures optimal feed processing performance.

The client must transmit a User-Agent header line so that we can diagnose problematic HTTP client software. For more information about the User-Agent header line, see the topic, User-Agent Header (p. 34).

The Content-MD5 HTTP header is required when calling `SubmitFeed`. It must be computed as per section 14.15 of the *HTTP/1.1 Specification*(http://www.ietf.org/rfc/rfc2616.txt). For more information, see the topic, Using the Content-MD5 Header with SubmitFeed (p. 36).

The actual format of the *`FeedContent`* in the HTTP body of the `SubmitFeed` call varies by marketplace, seller, product category, and by other factors.

For additional information, see Related Resources (p. 2).

In North America and Europe, transmit a Content-Type of "text/tab-separated-values; charset=iso-8859-1". In Japan, "text/tab-separated-values; charset=Shift_JIS".

Request Parameters

Name	Description	Required
FeedContent	The actual content of the feed itself, in XML or flat file format. You must include the *FeedContent* in the body of the HTTP request. Type: HTTP-BODY Default: none	Yes
FeedType	The FeedType (p. 93) being submitted, which indicates how the data should be processed. Type: xs:string Default: none	Yes
PurgeAndReplace	Set to `true` to enable purge and replace functionality. Only applicable to product-related flat file feed types, which don't have a mechanism for specifying purge and replace in the feed body. Use this parameter only for exceptional cases. Usage is throttled to allow only one purge and replace within a 24-hour period. Type: xs:boolean default: `false`	No

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.
- See Using the Content-MD5 Header with SubmitFeed (p. 36) for information about the Content-MD5 header line, required in every `SubmitFeed` call.

Response Elements

The `SubmitFeed` operation returns a *SubmitFeed* response, which is an aggregated element with child elements described in the following table.

Name	Description
FeedSubmissionId	A unique identifier for the feed submission. Type: xs:string
FeedType	The type of feed submitted, as provided by the FeedType (p. 93) parameter of `SubmitFeed`.
SubmittedDate	The date and time when the feed was submitted. Type: xs:datetime
FeedProcessingStatus	The processing status of the feed submission.

Examples

Example Query Request

```
POST /?Marketplace=ATVExampleDER
        & Action=SubmitFeed
        &Merchant=A1XExample5E6
        &FeedType=_POST_PRODUCT_DATA_
        &AWSAccessKeyId=0PB842ExampleN4ZTR2
        &Version=2009-01-01
        &Signature=SvSExamplefZpSignaturex2cs%3D
        &SignatureVersion=2
        &SignatureMethod=HmacSHA256
        &Timestamp=2009-01-26T23%3A51%3A31.315Z  HTTP/1.1
Content-Type: text/xml; charset=iso-8859-1
Content-MD5: ExampleMd5HashOfHttpBodyAsPerRfc2616Example
User-Agent: MWSTestsuite/2009-03-05 (Language=Java/1.6.0_11/50.0/
Sun Microsystems Inc.; Platform=Linux/i386/2.4.21-50a6smp;
 MWSClientVersion=2009-03-09)
Host: mws.amazonservices.com
Transfer-Encoding: chunked
```

Example HTTP Body

The following is an example HTTP body for a `SubmitFeed` request for a health-related product. Keep in mind that your XML format can differ as per the previous Content-Type discussion.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<AmazonEnvelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="amzn-envelope.xsd">
  <Header>
    <DocumentVersion>1.01</DocumentVersion>
    <MerchantIdentifier>M_EXAMPLE_123456</MerchantIdentifier>
  </Header>
  <MessageType>Product</MessageType>
  <PurgeAndReplace>false</PurgeAndReplace>
  <Message>
    <MessageID>1</MessageID>
    <OperationType>Update</OperationType>
    <Product>
      <SKU>56789</SKU>
      <StandardProductID>
        <Type>ASIN</Type>
        <Value>B0EXAMPLEG</Value>
      </StandardProductID>
      <ProductTaxCode>A_GEN_NOTAX</ProductTaxCode>
      <DescriptionData>
        <Title>Example Product Title</Title>
        <Brand>Example Product Brand</Brand>
        <Description>This is an example product description.</Description>
        <BulletPoint>Example Bullet Point 1</BulletPoint>
        <BulletPoint>Example Bullet Point 2</BulletPoint>
        <MSRP currency="USD">25.19</MSRP>
        <Manufacturer>Example Product Manufacturer</Manufacturer>
        <ItemType>example-item-type</ItemType>
      </DescriptionData>
      <ProductData>
        <Health>
          <ProductType>
            <HealthMisc>
              <Ingredients>Example Ingredients</Ingredients>
              <Directions>Example Directions</Directions>
            </HealthMisc>
          </ProductType>
        </Health>
      </ProductData>
    </Product>
  </Message>
</AmazonEnvelope>
```

Example Response

MWS will respond with the following headers:

```
HTTP/1.1 200 OK
Content-Type: text/xml
```

MWS will respond with an HTTP body like the following:

```
<?xml version="1.0"?>
<SubmitFeedResponse xmlns="http://mws.amazonaws.com/doc/2009-01-01/">
  <SubmitFeedResult>
    <FeedSubmissionInfo>
      <FeedSubmissionId>2291326430</FeedSubmissionId>
      <FeedType>_POST_PRODUCT_DATA_</FeedType>
      <SubmittedDate>2009-02-20T02:10:35+00:00</SubmittedDate>
      <FeedProcessingStatus>_SUBMITTED_</FeedProcessingStatus>
    </FeedSubmissionInfo>
  </SubmitFeedResult>
  <ResponseMetadata>
    <RequestId>75424a38-f333-4105-98f0-2aa9592d665c</RequestId>
  </ResponseMetadata>
</SubmitFeedResponse>
```

Related Topics

- Managing Inventory (p. 11)
- Using the Content-MD5 Header with SubmitFeed (p. 36)
- User-Agent Header (p. 34)

GetFeedSubmissionList

Description

The `GetFeedSubmissionList` operation returns the total list of feed submissions within the previous 90 days that match the query parameters.

Calls to `GetFeedSubmissionList` are limited to 1 request per minute, included within the overall limit of 1,000 calls per seller account per hour.

The maximum number of results that will be returned in one call is one hundred. If there are additional results to return, *`HasNext`* will be returned in the response with a `true` value. To retrieve all the results, you can use the value of the *`NextToken`* parameter to call GetFeedSubmissionListByNextToken (p. 48) until *`HasNext`* is `false`.

Request Parameters

Name	Description	Required
`FeedSubmissionIdList`	A structured list of feed submission IDs. If you pass in explicit IDs in this call, the other conditions, if specified, will be ignored. Type: xs:string Default: All	No
`MaxCount`	Maximum number of feed submissions to return in the list. If you specify a number greater than 100, the call will be rejected. Type: xs:nonNegativeInteger Default: 10	No
`FeedTypeList`	A structured list of one or more FeedType (p. 93) constants by which to filter feed submissions. Type: xs:string Default: All types	No
`FeedProcessingStatusList`	A structured list of one or more feed processing statuses by which to filter feed submissions. Valid values are: _SUBMITTED_ _IN_PROGRESS_ _CANCELLED_ _DONE_ Type: xs:string Default: All types	No

Name	Description	Required
SubmittedFromDate	The earliest submission date you are looking for, in ISO8601 date format (for example, "2008-07-03T18:12:22Z" or "2008-07-03T18:12:22.093-07:00"). Type: xs:datetime Default: 30 days ago	No
SubmittedToDate	The latest submission date you are looking for, in ISO8601 date format (for example, "2008-07-03T18:12:22Z" or "2008-07-03T18:12:22.093-07:00"). Type: xs:datetime Default: Now	No

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.

Response Elements

The `GetFeedSubmissionList` operation returns a *`GetFeedSubmissionList`* response, which is an aggregated element with child elements described in the following table.

Name	Description
NextToken	Used to pass information to another call. Use the *`NextToken`* to call `GetFeedSubmissionListByNextToken` if the value of *`HasNext`* is `true`. Type: xs:string
HasNext	Indicates whether there are more items to return, requiring additional calls to `GetFeedSubmissionListByNextToken` to retrieve them all. `True` means there are more items to retrieve. `False` means there are no more items to retrieve. Type: xs:boolean
FeedSubmissionId	A unique identifier for the feed submission. Type: xs:string
FeedType	The type of feed submitted, as provided by the FeedType (p. 93) parameter of SubmitFeed (p. 39).
SubmittedDate	The date and time when the feed was submitted. Type: xs:datetime
FeedProcessingStatus	The processing status of the feed submission.

Examples

Example Query Request

```
https://mws.amazonservices.com/
        ?AWSAccessKeyId=0PExampleR2
        &Action=GetFeedSubmissionList

 &FeedSubmissionIdList.Id.1=1058369303&FeedSubmissionIdList.Id.2=1228369302
        &FeedTypeList.Type.1=_POST_PRODUCT_DATA_& FeedTypeList.Type.2=
 _POST_PRODUCT_PRICING_DATA_
        &FeedProcessingStatusList.Status.1=_DONE_
        &Marketplace=ATExampleER
        &Merchant=A1ExampleE6
        &Signature=BXExampleo%3D
        &SignatureVersion=2
        &SignatureMethod=HmacSHA256
        &Timestamp=2009-02-04T15%3A51%3A49.015Z
        &Version=2009-01-01
```

Example Response

```
<?xml version="1.0"?>
<GetFeedSubmissionListResponse xmlns="http://mws.amazonaws.com/
doc/2009-01-01/">
  <GetFeedSubmissionListResult>
    <NextToken>2YgYW55IGNhcm5hbCBwbGVhc3VyZS4=</NextToken>
    <HasNext>true</HasNext>
    <FeedSubmissionInfo>
      <FeedSubmissionId>2291326430</FeedSubmissionId>
      <FeedType>_POST_PRODUCT_DATA_</FeedType>
      <SubmittedDate>2009-02-20T02:10:35+00:00</SubmittedDate>
      <FeedProcessingStatus>_SUBMITTED_</FeedProcessingStatus>
    </FeedSubmissionInfo>
  </GetFeedSubmissionListResult>
  <ResponseMetadata>
    <RequestId>1105b931-6f1c-4480-8e97-f3b467840a9e</RequestId>
  </ResponseMetadata>
</GetFeedSubmissionListResponse>
```

GetFeedSubmissionListByNextToken

Description

The GetFeedSubmissionListByNextToken operation returns a list of feed submissions that match the query parameters, using the *NextToken*, which was supplied by a previous call to either GetFeedSubmissionListByNextToken or a call to GetFeedSubmissionList (p. 45), where the value of *HasNext* was true in that previous call.

Calls to GetFeedSubmissionListByNextToken do not have a specific limitation, but are included in the overall limit of 1,000 requests per hour per seller account.

Request Parameters

Name	Description	Required
NextToken	Token returned in a previous call to either GetFeedSubmissionList or GetFeedSubmissionListByNextToken when the value of *HasNext* was true. Type: xs:string	Yes

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.

Response Elements

The GetFeedSubmissionListByNextToken operation returns a *GetFeedSubmissionListByNextToken* response, which is an aggregated element with child elements described in the following table.

Name	Description
NextToken	Used to pass information to another call. Use the *NextToken* to call GetFeedSubmissionListByNextToken again if the value of *HasNext* is true. Type: xs:string
HasNext	Indicates whether there are more items to return, requiring additional calls to GetFeedSubmissionListByNextToken to retrieve them all. true means there are more items to retrieve. *False* means there are no more items to retrieve. Type: xs:boolean
FeedSubmissionId	A unique identifier for the feed submission. Type: xs:string

Name	Description
FeedType	The type of feed submitted, as provided by the FeedType (p. 93) parameter of SubmitFeed (p. 39).
SubmittedDate	The date and time when the feed was submitted. Type: xs:datetime
FeedProcessingStatus	The processing status of the feed submission.

Examples

Example Query Request

```
https://mws.amazonservices.com/
        ?AWSAccessKeyId=0PExampleR2
        &Action=GetFeedSubmissionListByNextToken
        &NextToken=2YgYW55IGNhcm5hbCBwbGVhc3VyZS4=
        &Marketplace=ATExampleER
        &Merchant=A1ExampleE6
        &Signature=BXExampleo%3D
        &SignatureVersion=2
        &SignatureMethod=HmacSHA256
        &Timestamp=2009-02-04T15%3A51%3A49.015Z
        &Version=2009-01-01
```

Example Response

```
<?xml version="1.0"?>
<GetFeedSubmissionListByNextTokenResponse xmlns="http://mws.amazonaws.com/
doc/2009-01-01/">
<GetFeedSubmissionListByNextTokenResult>
    <NextToken>none</NextToken>
    <HasNext>false</HasNext>
    <FeedSubmissionInfo>
      <FeedSubmissionId>2291326430</FeedSubmissionId>
      <FeedType>_POST_PRODUCT_DATA_</FeedType>
      <SubmittedDate>2009-02-20T02:10:35+00:00</SubmittedDate>
      <FeedProcessingStatus>_SUBMITTED_</FeedProcessingStatus>
    </FeedSubmissionInfo>
    </GetFeedSubmissionListByNextTokenResult>
  <ResponseMetadata>
    <RequestId>1105b931-6f1c-4480-8e97-f3b467840a9e</RequestId>
  </ResponseMetadata>
  </GetFeedSubmissionListByNextTokenResponse>
```

GetFeedSubmissionCount

Description

The `GetFeedsubmissionCount` operation returns a count of the total number of feed submissions within the previous 90 days.

Calls to `GetFeedSubmissionCount` are limited to one request per minute, included within the overall limit of 1,000 calls per seller account per hour.

Request Parameters

Name	Description	Required
FeedTypeList	A structured list of one or more FeedType (p. 93) constants by which to filter feed submissions. Type: xs:string Default: All types	No
FeedProcessingStatusList	A structured list of one or more feed processing statuses by which to filter feed submissions. Valid values are: _SUBMITTED_ _IN_PROGRESS_ _CANCELLED_ _DONE_ Type: xs:string Default: All types	No
SubmittedFromDate	The earliest submission date you are looking for, in ISO8601 date format (for example, "2008-07-03T18:12:22Z" or "2008-07-03T18:12:22.093-07:00"). Type: xs:datetime Default: 30 days ago	No
SubmittedToDate	The latest submission date you are looking for, in ISO8601 date format (for example, "2008-07-03T18:12:22Z" or "2008-07-03T18:12:22.093-07:00"). Type: xs:datetime Default: Now	No

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.

Response Elements

The `GetFeedSubmissionCount` operation returns a *`GetFeedSubmissionCount`* response, which is a single element with a child element described in the following table.

Name	Description
`Count`	The count of matching feed submissions. Type: tns: nonNegativeInteger

Examples

Example Query Request

```
https://mws.amazonservices.com/
        ?AWSAccessKeyId=0PExampleR2
        &Action=GetFeedSubmissionCount
        &FeedTypeList.Type.1=_POST_PRODUCT_DATA_

 &FeedProcessingStatusList.Status.1=_DONE_&FeedProcessingStatusList.Status.2=_CANCELLED_

        &Marketplace=ATExampleER
        &Merchant=A1ExampleE6
        &Signature=ewExampleU%3D
        &SignatureVersion=2
        &SignatureMethod=HmacSHA256
        &Timestamp=2009-02-04T15%3A51%3A49.312Z
        &Version=2009-01-01
```

Example Response

```
<?xml version="1.0"?>
<GetFeedSubmissionCountResponse
  xmlns="http://mws.amazonaws.com/doc/2009-01-01/">
  <GetFeedSubmissionCountResult>
    <Count>463</Count>
  </GetFeedSubmissionCountResult>
  <ResponseMetadata>
    <RequestId>21e482a8-15c7-4da3-91a4-424995ed0756</RequestId>
  </ResponseMetadata>
</GetFeedSubmissionCountResponse>
```

CancelFeedSubmissions

Description

The `CancelFeedSubmissions` operation cancels one or more feed submissions, returning the count of the canceled feed submissions and the feed submission information. You can specify a number to cancel of greater than one hundred, but information will only be returned about the first one hundred feed submissions in the list. To return metadata about a greater number of canceled feed submissions, you can call GetFeedSubmissionList (p. 45). If feeds have already begun processing, they cannot be canceled.

Calls to `CancelFeedSubmissions` are limited to one request per minute, included within the overall limit of 1,000 calls per seller account per hour.

Request Parameters

Name	Description	Required
FeedSubmissionIdList	A structured list of feed submission IDs. If you pass in explicit IDs in this call, the other conditions, if specified, will be ignored. Type: xs:string Default: All	No
FeedTypeList	A structured list of one or more FeedType (p. 93) constants by which to filter feed submissions. Type: xs:string Default: All types	No
SubmittedFromDate	The earliest submission date you are looking for, in ISO8601 date format (for example, "2008-07-03T18:12:22Z" or "2008-07-03T18:12:22.093-07:00"). Type: xs:datetime Default: 30 days ago	No
SubmittedToDate	The latest submission date you are looking for, in ISO8601 date format (for example, "2008-07-03T18:12:22Z" or "2008-07-03T18:12:22.093-07:00"). Type: xs:datetime Default: Now	No

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.

Response Elements

The CancelFeedSubmissions operation returns a *CancelFeedSubmissions* response, which is an aggregated element with child elements described in the following table.

Name	Description
Count	The total number of feed submissions that matched the query parameters. Type: tns: nonNegativeInteger
FeedSubmissionId	A unique identifier for the feed submission. Type: xs:string
FeedType	The type of feed submitted, as provided by the FeedType (p. 93) parameter of SubmitFeed (p. 39).
SubmittedDate	The date and time when the feed was submitted. Type: xs:datetime
FeedProcessingStatus	The processing status of the feed submission.

Examples

Example Query Request

```
https://mws.amazonservices.com/
        ? AWSAccessKeyId=0PExampleR2
        &Action=CancelFeedSubmissions
        &FeedSubmissionIdList.Id.1=1058369303
        &FeedTypeList.Type.1=_POST_PRODUCT_DATA_& FeedTypeList.Type.2=
 _POST_PRODUCT_PRICING_DATA_
        &Marketplace=ATExampleER
        &Merchant=A1ExampleE6
        &Signature=0RExample0%3D
        &SignatureVersion=2
        &SignatureMethod=HmacSHA256
        &Timestamp=2009-02-04T17%3A34%3A14.203Z
        &Version=2009-01-01
```

Example Response

```
<?xml version="1.0"?>
<CancelFeedSubmissionsResponse
  xmlns="http://mws.amazonaws.com/doc/2009-01-01/">
  <CancelFeedSubmissionsResult>
    <Count>1</Count>
    <FeedSubmissionInfo>
      <FeedSubmissionId>2291326430</FeedSubmissionId>
      <FeedType>_POST_PRODUCT_DATA_</FeedType>
      <SubmittedDate>2009-02-20T02:10:35+00:00</SubmittedDate>
      <FeedProcessingStatus>_CANCELLED_</FeedProcessingStatus>
    </FeedSubmissionInfo>
  </CancelFeedSubmissionsResult>
  <ResponseMetadata>
    <RequestId>18e78983-bbf9-43aa-a661-ae7696cb49d4</RequestId>
  </ResponseMetadata>
</CancelFeedSubmissionsResponse>
```

GetFeedSubmissionResult

Description

The `GetFeedSubmissionResult` operation returns the feed processing report and the Content-MD5 header for the returned body.

Calls to `GetFeedSubmissionResult` are limited to 60 requests per hour, included within the overall limit of 1,000 calls per seller account per hour.

You should compute the MD5 hash of the HTTP body that we returned to you, and compare that with the Content-MD5 header value that we returned. If they do not match, which means the body was corrupted during transmission, you should discard the result and automatically retry the call for up to three more times. Please notify us if you ever see such a corrupted body. You can contact us by using the contact form at:

- DE: *http://developer.amazonservices.de*
- FR: *http://developer.amazonservices.fr*
- JP: *http://developer.amazonservices.jp*
- UK: *http://developer.amazonservices.co.uk*
- US: *http://developer.amazonservices.com*

For more information on computing the MD5, see Using the Content-MD5 Header with SubmitFeed (p. 36).

Request Parameters

Name	Description	Required
`FeedSubmissionId`	The identifier of the feed submission to get results for. Obtained by a call to GetFeedSubmissionList (p. 45). Type: xs:string	Yes

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.

Response Elements

The `GetFeedSubmissionResult` operation returns the feed processing report and the Content-MD5 header for the returned body.

Examples

Example Query Request

```
https://mws.amazonservices.com/
        ?AWSAccessKeyId=0PExampleR2
        &Action=GetFeedSubmissionResult
        &FeedSubmissionId=20Example76
        &Marketplace=ATExampleER
        &Merchant=A1ExampleE6
        &Signature=CNExampleQ%3D
        &SignatureVersion=2
        &SignatureMethod=HmacSHA256
        &Timestamp=2009-02-04T17%3A44%3A33.500Z
        &Version=2009-01-01
```

Example Response

```
<?xml version="1.0" encoding="UTF-8"?>
<AmazonEnvelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="amzn-envelope.xsd">
 <Header>
  <DocumentVersion>1.02</DocumentVersion>
  <MerchantIdentifier>M_EXAMPLE_9876543210</MerchantIdentifier>
 </Header>
 <MessageType>ProcessingReport</MessageType>
 <Message>
  <MessageID>1</MessageID>
  <ProcessingReport>
   <DocumentTransactionID>2060950676</DocumentTransactionID>
   <StatusCode>Complete</StatusCode>
   <ProcessingSummary>
    <MessagesProcessed>0</MessagesProcessed>
    <MessagesSuccessful>0</MessagesSuccessful>
    <MessagesWithError>1</MessagesWithError>
    <MessagesWithWarning>0</MessagesWithWarning>
   </ProcessingSummary>
   <Result>
    <MessageID>0</MessageID>
    <ResultCode>Error</ResultCode>
    <ResultMessageCode>6001</ResultMessageCode>
    <ResultDescription>XML parsing fatal error at line 1, column 1: Invalid
 document structure</ResultDescription>
    <AdditionalInfo>
     <SKU>0</SKU>
    </AdditionalInfo>
   </Result>
  </ProcessingReport>
 </Message>
</AmazonEnvelope>
```

Related Topics

- Using the Content-MD5 Header with SubmitFeed (p. 36)

RequestReport

Description

The RequestReport operation requests the generation of a report, which creates a report request. Reports are retained for 90 days.

Calls to RequestReport are limited to 30 requests per hour, included within the overall limit of 1,000 calls per seller account per hour.

Request Parameters

Name	Description	Required
ReportType (p. 95)	The type of report to request. Type: xs:string	Yes
StartDate	Start of a date range used for selecting the data to report. Type: xs:datetime Default: Now	No
EndDate	End of a date range used for selecting the data to report. Type: xs:datetime Default: Now	No

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.

Response Elements

The RequestReport operation returns a *RequestReport* response, which is an aggregated element with child elements described in the following table.

Name	Description
ReportRequestId	A unique identifier for the report request. Type: xs:string
ReportType	The ReportType (p. 95) requested.
StartDate	Start of a date range used for selecting the data to report. Type: xs:datetime
EndDate	End of a date range used for selecting the data to report. Type: xs:datetime

Name	Description
Scheduled	Whether or not this report was scheduled. Type: xs:boolean
SubmittedDate	The submission date of the report. Type: xs:datetime
ReportProcessingStatus	The processing status of the report.

Examples

Example Query Request

```
https://mws.amazonservices.com/
        ?AWSAccessKeyId=0PB842EXAMPLE7N4ZTR2
        &Action=RequestReport
        &EndDate=2008-06-26T18%3A12%3A21
        &Marketplace=ATVPDKIKX0DER
        &Merchant=A1XEXAMPLE5E6
        &ReportType=_GET_MERCHANT_LISTINGS_DATA_
        &Signature=ZQLpf8vEXAMPLE0iC265pf18n0%3D
        &SignatureVersion=2
        &SignatureMethod=HmacSHA256
        &StartDate=2009-01-03T18%3A12%3A21
        &Timestamp=2009-02-04T18%3A12%3A21.687Z
        &Version=2009-01-01
```

Example Response

```
<?xml version="1.0"?>
<RequestReportResponse xmlns="http://mws.amazonaws.com/doc/2009-01-01/">
  <RequestReportResult>
    <ReportRequestInfo>
      <ReportRequestId>2291326454</ReportRequestId>
      <ReportType>_GET_MERCHANT_LISTINGS_DATA_</ReportType>
      <StartDate>2009-01-21T02:10:39+00:00</StartDate>
      <EndDate>2009-02-13T02:10:39+00:00</EndDate>
      <Scheduled>false</Scheduled>
      <SubmittedDate>2009-02-20T02:10:39+00:00</SubmittedDate>
      <ReportProcessingStatus>_SUBMITTED_</ReportProcessingStatus>
    </ReportRequestInfo>
  </RequestReportResult>
  <ResponseMetadata>
    <RequestId>88faca76-b600-46d2-b53c-0c8c4533e43a</RequestId>
  </ResponseMetadata>
</RequestReportResponse>
```

GetReportRequestList

Description

The `GetReportRequestList` operation returns a list of report requests that match the query parameters.

Calls to `GetReportRequestList` are limited to one request per minute, included within the overall limit of 1,000 calls per seller account per hour.

The maximum number of results that will be returned in one call is one hundred. If there are additional results to return, *`HasNext`* will be returned in the response with a `true` value. To retrieve all the results, you can use the value of the *`NextToken`* parameter to call GetReportRequestListByNextToken (p. 62) until *`HasNext`* is `false`.

Request Parameters

Name	Description	Required
`ReportRequestIdList`	A structured list of report request IDs. If you pass in explicit IDs in this call, the other conditions, if specified, will be ignored. Type: xs:string Default: All	No
`ReportTypeList`	A structured ReportType (p. 95) list by which to filter reports. Type: xs:string Default: All	No
`ReportProcessingStatusList`	A structured list of report processing statuses by which to filter report requests. ReportProcessingStatus values: _SUBMITTED_ _IN_PROGRESS_ _CANCELLED_ _DONE_ _DONE_NO_DATA_ Type: xs:string Default: All	No
`MaxCount`	Maximum number of reports to return in the list. If you specify a number greater than 100, the call will be rejected. Type: xs:nonNegativeInteger Default: 10	No

Name	Description	Required
RequestedFromDate	The earliest date you are looking for, in ISO8601 date format (for example, "2008-07-03T18:12:22Z" or "2008-07-03T18:12:22.093-07:00"). Type: xs:datetime Default: 90 days ago	No
RequestedToDate	The most recent date you are looking for. Type: xs:datetime Default: Now	No

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.

Response Elements

The `GetReportRequestList` operation returns a *GetReportRequestList* response, which is an aggregated element with child elements described in the following table.

Name	Description
NextToken	Used to pass information to another call. Use the *NextToken* to call *GetReportRequestListByNextToken* if the value of *HasNext* is `true`. Type: xs:string
HasNext	Indicates whether there are more items to return, requiring additional calls to *GetReportRequestListByNextToken* to retrieve them all. `True` means there are more items to retrieve. `False` means there are no more items to retrieve. Type: xs:boolean
ReportRequestId	A unique identifier for the report request. Type: xs:string
ReportType	The ReportType (p. 95) requested.
StartDate	Start of a date range used for selecting the data to report. Type: xs:datetime
EndDate	End of a date range used for selecting the data to report. Type: xs:datetime

Name	Description
Scheduled	Whether or not this report was scheduled. Type: xs:boolean
SubmittedDate	The date when the report was submitted. Type: xs:datetime
ReportProcessingStatus	The processing status of the report.

Examples

Example Query Request

```
https://mws.amazonservices.com/
        ?AWSAccessKeyId=0PB842EXAMPLE7N4ZTR2
        &Action=GetReportRequestList
        &ReportRequestIdList.Id.1= 2291326454
        &ReportTypeList.Type.1= _GET_ORDERS_DATA_& ReportTypeList.Type.2=
 _GET_MERCHANT_LISTINGS_DATA_
        &ReportProcessingStatusList.Status.1=_DONE_
        &Marketplace=ATVPDKIKX0DER
        &Merchant=A1XEXAMPLE5E6
        &Signature=pBixmXKBaS%2Bq3EbPzgFhv%2BDf6do%3D
        &SignatureVersion=2
        &SignatureMethod=HmacSHA256
        &Timestamp=2009-02-04T18%3A12%3A21.921Z
        &Version=2009-01-01
```

Example Response

```
<?xml version="1.0"?>
<GetReportRequestListResponse xmlns="http://mws.amazonaws.com/
doc/2009-01-01/">
  <GetReportRequestListResult>
  <NextToken>2YgYW55IPQhcm5hbCBwbGVhc3VyZS4=</NextToken>
    <HasNext>true</HasNext>
    <ReportRequestInfo>
      <ReportRequestId>2291326454</ReportRequestId>
      <ReportType>_GET_MERCHANT_LISTINGS_DATA_</ReportType>
      <StartDate>2009-01-21T02:10:39+00:00</StartDate>
      <EndDate>2009-02-13T02:10:39+00:00</EndDate>
      <Scheduled>false</Scheduled>
      <SubmittedDate>2009-02-20T02:10:39+00:00</SubmittedDate>
      <ReportProcessingStatus>_SUBMITTED_
      </ReportProcessingStatus>
    </ReportRequestInfo>
  </GetReportRequestListResult>
  <ResponseMetadata>
    <RequestId>732480cb-84a8-4c15-9084-a46bd9a0889b</RequestId>
  </ResponseMetadata>
</GetReportRequestListResponse>
```

GetReportRequestListByNextToken

Description

The GetReportRequestListByNextToken operation returns a list of report requests that match the query parameters, using the *NextToken*, which was supplied by a previous call to either *GetReportRequestListByNextToken* or a call to GetReportRequestList (p. 59), where the value of *HasNext* was true in that previous call.

Calls to GetReportRequestListByNextToken do not have a specific limitation, but are included in the overall limit of 1,000 requests per hour per seller account.

Request Parameters

Name	Description	Required
NextToken	Token returned in a previous call to either GetReportRequestList or GetReportRequestListByNextToken when the value of *HasNext* was true. Type: xs:string	Yes

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.

Response Elements

The GetReportRequestListByNextToken operation returns a *GetReportRequestListByNextToken* response, which is an aggregated element with child elements described in the following table.

Name	Description
NextToken	Used to pass information to another call. Use the *NextToken* to call the operation again if the value of *HasNext* is true. Type: xs:string
HasNext	Indicates whether there are more items to return, requiring additional calls to GetReportRequestListByNextToken to retrieve them all. True means there are more items to retrieve. False means there are no more items to retrieve. Type: xs:boolean
ReportRequestId	A unique identifier for the report request. Type: xs:string

Name	Description
ReportType	The ReportType (p. 95) requested.
StartDate	Start of a date range used for selecting the data to report. Type: xs:datetime
EndDate	End of a date range used for selecting the data to report. Type: xs:datetime
Scheduled	Whether or not this report was scheduled. Type: xs:boolean
SubmittedDate	The date when the report was submitted. Type: xs:datetime
ReportProcessingStatus	The processing status of the report.

Examples

Example Query Request

```
https://mws.amazonservices.com/
        ?AWSAccessKeyId=0PB842EXAMPLE7N4ZTR2
        &Action=GetReportRequestListByNextToken
        &NextToken=2YgYW55IPQhcm5hbCBwbGVhc3VyZS4=
        &Marketplace=ATVPDKIKX0DER
        &Merchant=A1XEXAMPLE5E6
        &Signature=pBixmXKBaS%2Bq3EbPzgFhv%2BDf6do%3D
        &SignatureVersion=2
        &SignatureMethod=HmacSHA256
        &Timestamp=2009-02-04T18%3A12%3A21.921Z
        &Version=2009-01-01
```

Example Response

```
<?xml version="1.0"?>
<GetReportRequestListByNextTokenResponse xmlns="http://mws.amazonaws.com/
doc/2009-01-01/">
<GetReportRequestLisByNextTokentResult>
    <NextToken>none</NextToken>
    <HasNext>false</HasNext>
    <ReportRequestInfo>
      <ReportRequestId>2291326454</ReportRequestId>
      <ReportType>_GET_MERCHANT_LISTINGS_DATA_</ReportType>
      <StartDate>2009-01-21T02:10:39+00:00</StartDate>
      <EndDate>2009-02-13T02:10:39+00:00</EndDate>
      <Scheduled>false</Scheduled>
      <SubmittedDate>2009-02-20T02:10:39+00:00</SubmittedDate>
      <ReportProcessingStatus>_SUBMITTED_
      </ReportProcessingStatus>
    </ReportRequestInfo>
    </GetReportRequestListByNextTokenResult>
  <ResponseMetadata>
    <RequestId>732480cb-84a8-4c15-9084-a46bd9a0889b</RequestId>
  </ResponseMetadata>
  </GetReportRequestListByNextTokenResponse>
```

GetReportRequestCount

Description

The `GetReportRequestCount` returns a count of report requests.

Calls to `GetReportRequestCount` are limited to one request per minute, included within the overall limit of 1,000 calls per seller account per hour.

Request Parameters

Name	Description	Required
ReportTypeList	A structured ReportType (p. 95) list by which to filter reports. Type: xs:string Default: All	No
ReportProcessingStatusList	A structured list of report processing statuses by which to filter report requests. ReportProcessingStatus values: _SUBMITTED_ _IN_PROGRESS_ _CANCELLED_ _DONE_ _DONE_NO_DATA_ Type: xs:string Default: All	No
RequestedFromDate	The earliest date you are looking for, in ISO8601 date format (for example, "2008-07-03T18:12:22Z" or "2008-07-03T18:12:22.093-07:00"). Type: xs:datetime Default: 90 days ago	No
RequestedToDate	The most recent date you are looking for. Type: xs:datetime Default: Now	No

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.

Response Elements

The `GetReportRequestCount` operation returns a *`GetReportRequestCount`* response, which is a single element with a child element described in the following table.

Name	Description
`Count`	Total number of report requests. Type: tns: nonNegativeInteger

Examples

Example Query Request

```
https://mws.amazonservices.com/
        ?AWSAccessKeyId=0PB842EXAMPLE7N4ZTR2
        &Action=GetReportRequestCount
        &ReportTypeList.Type.1= _GET_ORDERS_DATA_& ReportTypeList.Type.2=
 _GET_MERCHANT_LISTINGS_DATA_
        &ReportProcessingStatusList.Status.1=_DONE_
        &Marketplace=ATVPDKIKX0DER
        &Merchant=A1XEXAMPLE5E6
        &Signature=oqxKULFnyvOMXmrsqerBjOJlW0U%3D
        &SignatureVersion=2
        &SignatureMethod=HmacSHA256
        &Timestamp=2009-02-04T18%3A12%3A22.093Z
        &Version=2009-01-01
```

Example Response

```
<GetReportRequestCountResponse xmlns="http://mws.amazonaws.com/
doc/2009-01-01/">
  <GetReportRequestCountResult>
    <Count>1276</Count>
  </GetReportRequestCountResult>
  <ResponseMetadata>
    <RequestId>7e155027-3741-4422-95a7-1de12703c13e</RequestId>
  </ResponseMetadata>
</GetReportRequestCountResponse>
```

CancelReportRequests

Description

The `CancelReportRequests` operation cancels one or more report requests, returning the count of the canceled report requests and the report request information. You can specify a number to cancel of greater than one hundred, but information will only be returned about the first one hundred report requests in the list. To return metadata about a greater number of canceled report requests, you can call GetReportRequestList (p. 59). If report requests have already begun processing, they cannot be canceled.

Calls to `CancelReportRequests` are limited to one request per minute, included within the overall limit of 1,000 calls per seller account per hour.

Request Parameters

Name	Description	Required
ReportRequestIdList	A structured list of report request IDs. If you pass in explicit IDs in this call, the other conditions, if specified, will be ignored. Type: xs:string Default: All	No
ReportTypeList	A structured ReportType (p. 95) list by which to filter reports. Type: xs:string Default: All	No
ReportProcessingStatusList	A structured list of report processing statuses by which to filter report requests. ReportProcessingStatus values: _SUBMITTED_ _IN_PROGRESS_ _CANCELLED_ _DONE_ _DONE_NO_DATA_ Type: xs:string Default: All	No
RequestedFromDate	The earliest date you are looking for, in ISO8601 date format (for example, "2008-07-03T18:12:22Z" or "2008-07-03T18:12:22.093-07:00"). Type: xs:datetime Default: 90 days ago	No

Name	Description	Required
RequestedToDate	The most recent date you are looking for. Type: xs:datetime Default: Now	No

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.

Response Elements

The `CancelReportRequests` operation returns a *CancelReportRequests* response, which is an aggregated element with child elements described in the following table.

Name	Description
Count	Total number of report requests. Type: tns: nonNegativeInteger
StartDate	Start of a date range used for selecting the data to report. Type: xs:datetime
EndDate	End of a date range used for selecting the data to report. Type: xs:datetime
Scheduled	Whether or not this report was scheduled. Type: xs:boolean
SubmittedDate	The submission date of the report. Type: xs:datetime
ReportProcessingStatus	The processing status of the report.

Examples

Example Query Request

```
https://mws.amazonservices.com/
        ?AWSAccessKeyId=0PB842EXAMPLE7N4ZTR2
        &Action=CancelReportRequests
        &ReportRequestIdList.Id.1= 2291326454
        &ReportTypeList.Type.1= _GET_ORDERS_DATA_& ReportTypeList.Type.2=
 _GET_MERCHANT_LISTINGS_DATA_
        &ReportProcessingStatusList.Status.1=_DONE_
        &Marketplace=ATVPDKIKX0DER
        &Merchant=A1XEXAMPLE5E6
        &Signature=wZFasNUpTth2GA5Xzuamb1XGVFY%3D
        &SignatureVersion=2
        &SignatureMethod=HmacSHA256
        &Timestamp=2009-02-04T18%3A12%3A22.421Z
        &Version=2009-01-01
```

Example Response

```
<?xml version="1.0"?>
<CancelReportRequestsResponse xmlns="http://mws.amazonaws.com/
doc/2009-01-01/">
  <CancelReportRequestsResult>
    <Count>10</Count>
    <ReportRequestInfo>
      <ReportRequestId>2291326454</ReportRequestId>
      <ReportType>_GET_MERCHANT_LISTINGS_DATA_</ReportType>
      <StartDate>2009-01-21T02:10:39+00:00</StartDate>
      <EndDate>2009-02-13T02:10:39+00:00</EndDate>
      <Scheduled>false</Scheduled>
      <SubmittedDate>2009-02-20T02:10:39+00:00</SubmittedDate>
      <ReportProcessingStatus>_CANCELLED_</ReportProcessingStatus>
    </ReportRequestInfo>
  </CancelReportRequestsResult>
  <ResponseMetadata>
    <RequestId>a720f9d6-83e9-4684-bc35-065b41ed5ca4</RequestId>
  </ResponseMetadata>
</CancelReportRequestsResponse>
```

GetReportList

Description

The GetReportList operation returns a list of reports within the previous 90 days that match the query parameters. The maximum number of results that will be returned in one call is one hundred. If there are additional results to return, *HasNext* will be returned in the response with a true value. To retrieve all the results, you can use the value of the *NextToken* parameter to call GetReportListByNextToken (p. 73) until *HasNext* is false.

Calls to GetReportList are limited to one request per minute, included within the overall limit of 1,000 calls per seller account per hour.

Request Parameters

Name	Description	Required
MaxCount	Maximum number of reports to return in the list. If you specify a number greater than 100, the call will be rejected. Type: xs:nonNegativeInteger Default: 10	No
ReportTypeList	A structured ReportType (p. 95) list by which to filter reports. Type: xs:string Default: All	No
Acknowledged	Set to true to list order reports that have been acknowledged with a prior call to UpdateReportAcknowledgements (p. 89). Set to false to list order reports that have not been acknowledged. This filter is valid only with order reports; it does not work with listing reports. Type: xs:boolean Default: either	No
AvailableFromDate	The earliest date you are looking for, in ISO8601 date format (for example, "2008-07-03T18:12:22Z" or "2008-07-03T18:12:22.093-07:00"). Type: xs:datetime Default: 90 days ago	No
AvailableToDate	The most recent date you are looking for. Type: xs:datetime Default: Now	No

Name	Description	Required
ReportRequestIdList	A structured list of report request IDs. If you pass in explicit IDs in this call, the other conditions, if specified, will be ignored. Type: xs:string Default: All	No

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.

Response Elements

The GetReportList operation returns a *GetReportList* response, which is an aggregated element with child elements described in the following table.

Name	Description
NextToken	Used to pass information to another call. Use the *NextToken* to call the operation again if the value of *HasNext* is true. Type: xs:string
HasNext	Indicates whether there are more items to return, requiring additional calls to GetReportListByNextToken to retrieve them all. True means there are more items to retrieve. False means there are no more items to retrieve. Type: xs:boolean
ReportId	A unique identifier for the report . Type: xs:string
ReportType	The ReportType (p. 95) requested.
ReportRequestId	A unique identifier for the report request. Type: xs:string
AvailableDate	The date the report is available. Type: xs:datetime
Acknowledged	Whether or not this report was acknowledged. Type: xs:boolean

Examples

Example Query Request

```
https://mws.amazonservices.com/
        ?AWSAccessKeyId=0PB842EXAMPLE7N4ZTR2
        &Acknowledged=false
        &Action=GetReportList
        &ReportRequestIdList.Id.1= 2291326454& ReportRequestIdList.Id.2=
 2294446454
        &ReportTypeList.Type.1= _GET_ORDERS_DATA_
        &Marketplace=ATVPDKIKX0DER
        &Merchant=A1XEXAMPLE5E6
        &ReportTypeList=_GET_MERCHANT_LISTINGS_DATA_
        &Signature=3yvUqWWBpLDld9CCx0ANjVU95ks%3D
        &SignatureVersion=2
        &SignatureMethod=HmacSHA256
        &Timestamp=2009-02-04T18%3A12%3A19.796Z
&Version=2009-01-01
```

Example Response

```
<?xml version="1.0"?>
<GetReportListResponse xmlns="http://mws.amazonaws.com/doc/2009-01-01/">
  <GetReportListResult>
  <NextToken>2YgYW55IPQhvu5hbCBwbGVhc3VyZS4=</NextToken>
    <HasNext>true</HasNext>
    <ReportInfo>
      <ReportId>898899473</ReportId>
      <ReportType>_GET_MERCHANT_LISTINGS_DATA_</ReportType>
      <ReportRequestId>2278662938</ReportRequestId>
      <AvailableDate>2009-02-10T09:22:33+00:00</AvailableDate>
      <Acknowledged>false</Acknowledged>
    </ReportInfo>
  </GetReportListResult>
  <ResponseMetadata>
    <RequestId>fbf677c1-dcee-4110-bc88-2ba3702e331b</RequestId>
  </ResponseMetadata>
</GetReportListResponse
```

Related Topics

- UpdateReportAcknowledgements (p. 89)

GetReportListByNextToken

Description

The `GetReportListByNextToken` operation returns a list of reports that match the query parameters, using the *`NextToken`*, which was supplied by a previous call to either `GetReportListByNextToken` or a call to GetReportList (p. 70), where the value of *`HasNext`* was `true` in that previous call.

Calls to `GetReportListByNextToken` do not have a specific limitation, but are included in the overall limit of 1,000 requests per hour per seller account.

Request Parameters

Name	Description	Required
NextToken	Token returned in a previous call to either `GetReportList` or `GetReportListByNextToken` when the value of *`HasNext`* was `true`. Type: xs:string	Yes

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.

Response Elements

The `GetReportListByNextToken` operation returns a *`GetReportListByNextToken`* response, which is an aggregated element with child elements described in the following table.

Name	Description
`NextToken`	Used to pass information to another call. Use the *`NextToken`* to call the operation again if the value of *`HasNext`* is `true`. Type: xs:string
`HasNext`	Indicates whether there are more items to return, requiring additional calls to `GetReportListByNextToken` to retrieve them all. `True` means there are more items to retrieve. `False` means there are no more items to retrieve. Type: xs:boolean
`ReportId`	A unique identifier for the report . Type: xs:string
`ReportType`	The ReportType (p. 95) requested.

Name	Description
ReportRequestId	A unique identifier for the report request. Type: xs:string
AvailableDate	The date the report is available. Type: xs:datetime
Acknowledged	Whether or not this report was acknowledged. Type: xs:boolean

Examples

Example Query Request

```
https://mws.amazonservices.com/
        ?AWSAccessKeyId=0PB842EXAMPLE7N4ZTR2
        &Acknowledged=false
        &Action=GetReportListByNextToken
        &NextToken=2YgYW55IPQhvu5hbCBwbGVhc3VyZS4=
        &Marketplace=ATVPDKIKX0DER
        &Merchant=A1XEXAMPLE5E6
        &ReportTypeList=_GET_MERCHANT_LISTINGS_DATA_
        &Signature=3yvUqWWBpLDld9CCx0ANjVU95ks%3D
        &SignatureVersion=2
        &SignatureMethod=HmacSHA256
        &Timestamp=2009-02-04T18%3A12%3A19.796Z
&Version=2009-01-01
```

Example Response

```
<?xml version="1.0"?>
<GetReportListByNextTokenResponse xmlns="http://mws.amazonaws.com/
doc/2009-01-01/">
<GetReportListByNextTokenResult>
    <NextToken>none</NextToken>
    <HasNext>false</HasNext>
    <ReportInfo>
      <ReportId>898899473</ReportId>
      <ReportType>_GET_MERCHANT_LISTINGS_DATA_</ReportType>
      <ReportRequestId>2278662938</ReportRequestId>
      <AvailableDate>2009-02-10T09:22:33+00:00</AvailableDate>
      <Acknowledged>false</Acknowledged>
    </ReportInfo>
    </GetReportListByNextTokenResult>
  <ResponseMetadata>
    <RequestId>fbf677c1-dcee-4110-bc88-2ba3702e331b</RequestId>
  </ResponseMetadata>
  </GetReportListByNextTokenResponse
```

GetReportCount

Description

The `GetReportCount` operation returns a count of reports within the previous 90 days that are available for the seller to download.

Calls to `GetReportCount` are limited to one request per minute, included within the overall limit of 1,000 calls per seller account per hour.

Request Parameters

Name	Description	Required
`ReportTypeList`	A structured ReportType (p. 95) list by which to filter reports. Type: xs:string Default: All	No
`Acknowledged`	Set to `true` to list order reports that have been acknowledged with a prior call to UpdateReportAcknowledgements (p. 89). Set to `false` to list order reports that have not been acknowledged. This filter is valid only with order reports; it does not work with listing reports. Type: xs:boolean Default: `either`	No
`AvailableFromDate`	The earliest date you are looking for, in ISO8601 date format (for example, "2008-07-03T18:12:22Z" or "2008-07-03T18:12:22.093-07:00"). Type: xs:datetime Default: 90 days ago	No
`AvailableToDate`	The most recent date you are looking for. Type: xs:datetime Default: Now	No

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.

Response Elements

The `GetReportCount` operation returns a *`GetReportCount`* response, which is a single element with a child element described in the following table.

Name	Description
Count	Total number of reports. Type: tns: nonNegativeInteger

Examples

Example Query Request

```
https://mws.amazonservices.com/
        ?AWSAccessKeyId=0PB842EXAMPLE7N4ZTR2
        &Action=GetReportCount
        &ReportTypeList.Type.1= _GET_ORDERS_DATA_
        &Marketplace=ATVPDKIKX0DER
        &Merchant=A1XEXAMPLE5E6
        &Signature=AV6JVsC3JvgEXAMPLERL685cP64%3D
        &SignatureVersion=2
        &SignatureMethod=HmacSHA256
        &Timestamp=2009-02-04T18%3A12%3A20.296Z
        &Version=2009-01-01
```

Example Response

```
<?xml version="1.0"?>
<GetReportCountResponse xmlns="http://mws.amazonaws.com/doc/2009-01-01/">
  <GetReportCountResult>
    <Count>166</Count>
  </GetReportCountResult>
  <ResponseMetadata>
    <RequestId>a497aadb-5ea1-49bf-aa14-dabe914465e3</RequestId>
  </ResponseMetadata>
</GetReportCountResponse>
```

GetReport

Description

The `GetReport` operation returns the contents of a report and the Content-MD5 header for the returned body. Reports are retained for 90 days from the time they have been generated.

You should compute the MD5 hash of the HTTP body and compare that with the returned Content-MD5 header value. If they do not match, which means the body was corrupted during transmission, you should discard the result and automatically retry the call for up to three more times. Please notify us if you ever see such a corrupted body. You can contact us by using the contact form at:

- DE: *http://developer.amazonservices.de*
- FR: *http://developer.amazonservices.fr*
- JP: *http://developer.amazonservices.jp*
- UK: *http://developer.amazonservices.co.uk*
- US: *http://developer.amazonservices.com*

For more information, see Using the Content-MD5 Header with SubmitFeed (p. 36).

Calls to `GetReport` are limited to 60 requests per hour, included within the overall limit of 1,000 calls per seller account per hour.

Request Parameters

Name	Description	Required
ReportId	A unique identifier of the report to download, as obtained from GetReportList (p. 70) or the *GeneratedReportId* of a *ReportRequest*. Type: xs:nonNegativeInteger Default: none	Yes

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.

Response Elements

Name	Description
Report	The contents of the report document. Type: xs:string

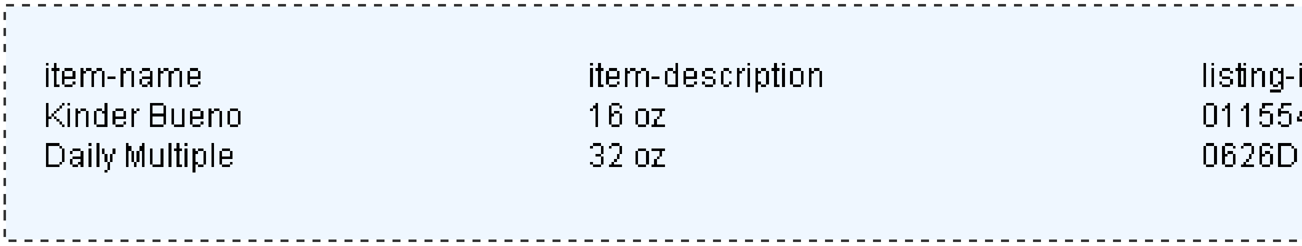
Examples

Example Query Request

```
http://mws.amazonservices.com/
        ?AWSAccessKeyId=0PB842EXAMPLE7N4ZTR2
        &Action=GetReport
        &Marketplace=ATVPDKIKX0DER
        &Merchant=A1XEXAMPLE5E6
        &ReportId=624169093
        &Signature=sY%2BEJFLA1gmz78dEOofUcBWSM44%3D
        &SignatureVersion=2
        &SignatureMethod=HmacSHA256
        &Timestamp=2009-02-04T18%3A12%3A20.015Z
        &Version=2009-01-01
```

Example Response

This is an example of a response in tab-delimited format.



```
item-name              item-description          listing-i
Kinder Bueno           16 oz                     01155
Daily Multiple         32 oz                     0626D
```

ManageReportSchedule

Description

The ManageReportSchedule operation creates, updates, or deletes a report schedule for a particular report type. Currently, only order reports can be scheduled.

Calls to ManageReportSchedule are limited to one request per minute, included within the overall limit of 1,000 calls per seller account per hour.

Request Parameters

Name	Description	Required
ReportType	The type of reports that you want to schedule generation of. Currently, only order reports can be scheduled. Type: xs:string Default: none	Yes
Schedule	A string that describes how often a *ReportRequest* should be created. The list of enumerated values is found in the enumeration topic, Schedule (p. 113). Type: xs:string Default: none	Yes
ScheduledDate	The date when the next report is scheduled to run. Limited to no more than 366 days in the future. Type: xs:datetime Default: Now	No

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.

Response Elements

The ManageReportSchedule operation returns a *ManageReportSchedule* response, which is an aggregated element with child elements described in the following table.

Name	Description
Count	Total number of reports that matched the query parameters. Type: tns: nonNegativeInteger
ReportType	The ReportType (p. 95) to be scheduled.
Schedule	The Schedule (p. 113) that determines how often a report will be requested.

Name	Description
ScheduledDate	The date when the next report is scheduled to run.

Examples

Example Query Request

```
https://mws.amazonservices.com
        ?AWSAccessKeyId=06Example02
        &Action=ManageReportSchedule
        &Marketplace=ATVPDKIKX0DER
        &Merchant=A3Example4D
        &ReportType=_GET_ORDERS_DATA_
        &Schedule=_30_DAYS_
        &Signature=RuExample0%3D
        &SignatureVersion=2
        &SignatureMethod=HmacSHA256
        &Timestamp=2009-01-14T20%3A50%3A30.218Z
        &Version=2009-01-01
```

Example Response

```
 <?xml version="1.0"?>
<ManageReportScheduleResponse xmlns="http://mws.amazonaws.com/
doc/2009-01-01/">
  <ManageReportScheduleResult>
    <Count>1</Count>
    <ReportSchedule>
      <ReportType>_GET_ORDERS_DATA_</ReportType>
      <Schedule>_30_DAYS_</Schedule>
      <ScheduledDate>2009-02-20T02:10:42+00:00</ScheduledDate>
    </ReportSchedule>
  </ManageReportScheduleResult>
  <ResponseMetadata>
    <RequestId>7ee1bc50-5a13-4db1-afd7-1386e481984e</RequestId>
  </ResponseMetadata>
</ManageReportScheduleResponse>
```

GetReportScheduleList

Description

The `GetReportScheduleList` operation returns a list of report schedules that match the query parameters. Currently, only order reports can be scheduled.

The maximum number of results that will be returned in one call is one hundred. If there are additional results to return, *`HasNext`* will be returned in the response with a `true` value. To retrieve all the results, you can use the value of the *`NextToken`* parameter to call GetReportScheduleListByNextToken (p. 85) until *`HasNext`* is `false`.

Note

For this release of Amazon MWS, only order reports can be scheduled, so *`HasNext`* will always be `False`.

Calls to `GetReportScheduleList` are limited to one request per minute, included within the overall limit of 1,000 calls per seller account per hour.

Request Parameters

Name	Description	Required
`ReportTypeList`	A structured ReportType (p. 95) list by which to filter reports. Currently, only order reports can be scheduled. Type: xs:string Default: All	No

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.

Response Elements

The `GetReportScheduleList` operation returns a *`GetReportScheduleList`* response, which is an aggregated element with child elements described in the following table.

Name	Description
`NextToken`	Used to pass information to another call. Use the *`NextToken`* to call the operation again if the value of *`HasNext`* is `true`. Type: xs:string
`HasNext`	Indicates whether there are more items to return, requiring additional calls to `GetReportScheduleListByNextToken` to retrieve them all. `True` means there are more items to retrieve. `False` means there are no more items to retrieve. Type: xs:boolean

Name	Description
ReportType	The ReportType (p. 95) to be scheduled.
Schedule	The Schedule (p. 113) that determines how often a report will be requested.
ScheduledDate	The date when the next report is scheduled to run.

Examples

Example Query Request

```
https://mws.amazonservices.com/
        ?AWSAccessKeyId=0PExampleR2
        &Action=GetReportScheduleList
        &ReportTypeList.Type.1= _GET_ORDERS_DATA_& ReportTypeList.Type.2=
 _GET_MERCHANT_LISTINGS_DATA
        &Marketplace=ATExampleER
        &Merchant=A1ExampleE6
        &Signature=ltExample8%3D
        &SignatureVersion=2
        &SignatureMethod=HmacSHA256
        &Timestamp=2009-01-07T19%3A12%3A13.859Z
        &Version=2009-01-01
```

Example Response

```
<?xml version="1.0"?>
<GetReportScheduleListResponse xmlns="http://mws.amazonaws.com/
doc/2009-01-01/">
  <GetReportScheduleListResult>
  <NextToken>4XgYW55IPQhcm5hbCBwbGVhc3VyZS4=</NextToken>
    <HasNext>true</HasNext>
    <ReportSchedule>
      <ReportType>_GET_ORDERS_DATA_</ReportType>
      <Schedule>_30_DAYS_</Schedule>
      <ScheduledDate>2009-02-20T02:10:42+00:00</ScheduledDate>
    </ReportSchedule>
  </GetReportScheduleListResult>
  <ResponseMetadata>
    <RequestId>c0464157-b74f-4e52-bd1a-4ebf4bc7e5aa</RequestId>
  </ResponseMetadata>
</GetReportScheduleListResponse>
```

GetReportScheduleListByNextToken

Description

The GetReportScheduleListByNextToken operation returns a list of report schedules that match the query parameters, using the *NextToken*, which was supplied by a previous call to either GetReportScheduleListByNextToken or a call to GetReportScheduleList (p. 83), where the value of *HasNext* was true in that previous call.

Note

For this release of Amazon MWS, only order reports can be scheduled, so *HasNext* will always be False.

Calls to GetReportScheduleListByNextToken do not have a specific limitation, but are included in the overall limit of 1,000 requests per hour per seller account.

Request Parameters

Name	Description	Required
NextToken	Token returned in a previous call to either GetReportScheduleList or GetReportScheduleListByNextToken when the value of *HasNext* was true. Type: xs:string	Yes

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.

Response Elements

The GetReportScheduleListByNextToken operation returns a *GetReportScheduleListByNextToken* response, which is an aggregated element with child elements described in the following table.

Name	Description
NextToken	Used to pass information to another call. Use the *NextToken* to call the operation again if the value of *HasNext* is true. Type: xs:string
HasNext	Indicates whether there are more items to display if you call the operation again. True means there are more items to retrieve. False means there are no more items to retrieve. Type: xs:boolean

Name	Description
ReportType	The ReportType (p. 95) to be scheduled.
Schedule	The Schedule (p. 113) that determines how often a report will be requested.
ScheduledDate	The date when the next report is scheduled to run.

Examples

Example Query Request

```
https://mws.amazonservices.com/
        ?AWSAccessKeyId=0PExampleR2
        &Marketplace=ATExampleER
        &Merchant=A1ExampleE6
        &Action=GetReportScheduleListByNextToken
        &NextToken=4XgYW55IPQhcm5hbCBwbGVhc3VyZS4=
        &Signature=ltExample8%3D
        &SignatureVersion=2
        &SignatureMethod=HmacSHA256
        &Timestamp=2009-01-07T19%3A12%3A13.859Z
        &Version=2009-01-01
```

Example Response

```
<?xml version="1.0"?>
<GetReportScheduleListByNextTokenResponse xmlns="http://mws.amazonaws.com/
doc/2009-01-01/">
<GetReportScheduleListByNextTokenResult>
    <NextToken>none</NextToken>
    <HasNext>false</HasNext>
    <ReportSchedule>
      <ReportType>_GET_ORDERS_DATA_</ReportType>
      <Schedule>_30_DAYS_</Schedule>
      <ScheduledDate>2009-02-20T02:10:42+00:00</ScheduledDate>
    </ReportSchedule>
    </GetReportScheduleListByNextTokenResult>
  <ResponseMetadata>
    <RequestId>c0464157-b74f-4e52-bd1a-4ebf4bc7e5aa</RequestId>
  </ResponseMetadata>
  </GetReportScheduleListByNextTokenResponse>
```

GetReportScheduleCount

Description

The GetReportScheduleCount operation returns a count of report schedules. Currently, only order reports can be scheduled.

Calls to GetReportScheduleCount are limited to one request per minute, included within the overall limit of 1,000 calls per seller account per hour.

Request Parameters

Name	Description	Required
ReportTypeList	A structured ReportType (p. 95) list by which to filter reports. Currently, only order reports can be scheduled. Type: xs:string Default: All	No

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.

Response Elements

The GetReportScheduleCount operation returns a *GetReportScheduleCount* response, which is a single element with a child element described in the following table.

Name	Description
Count	The count of matching report schedules. Type: tns: nonNegativeInteger

Examples

Example Query Request

```
https://mws.amazonservices.com/
        ?AWSAccessKeyId=0PExampleR2
        &Action=GetReportScheduleCount
        &ReportTypeList.Type.1= _GET_MERCHANT_LISTINGS_DATA_
        &Marketplace=ATExampleER
        &Merchant=A1ExampleE6
        &Signature=ltExample8%3D
        &SignatureVersion=2
        &SignatureMethod=HmacSHA256
        &Timestamp=2009-01-07T19%3A12%3A13.859Z
        &Version=2009-01-01
```

Example Response

```
<?xml version="1.0"?>
<GetReportScheduleCountResponse
  xmlns="http://mws.amazonaws.com/doc/2009-01-01/">
  <GetReportScheduleCountResult>
    <Count>18</Count>
    </GetReportScheduleCountResult>
  <ResponseMetadata>
    <RequestId>21e482a8-15c7-4da3-91a4-424995ed0756</RequestId>
  </ResponseMetadata>
  </GetReportScheduleCountResponse>
```

UpdateReportAcknowledgements

Description

The `UpdateReportAcknowledgements` operation is an optional function that you should use only if you want Amazon to remember the *`Acknowledged`* status of your reports. `UpdateReportAcknowledgements` updates the acknowledged status of one or more reports. To keep track of which reports you have already received, it is a good practice to acknowledge reports after you have received and stored them successfully. Then, when you call GetReportList (p. 70)you can specify to receive only reports that have not yet been acknowledged.

You can also use this function to retrieve reports that have been lost, possibly because of a hard disk failure, by setting *`Acknowledged`* to `false` and then calling `GetReportList`, which returns a list of reports within the previous 90 days that match the query parameters.

Calls to `UpdateReportAcknowledgements` are limited to one request per minute, included within the overall limit of 1,000 calls per seller account per hour.

Request Parameters

Name	Description	Required
`ReportIdList`	A structured list of Report Ids. The maximum number of reports that can be specified is 100. Type: xs:string Default: none	Yes
`Acknowledged`	Set to `true` to list reports that have been acknowledged. Set to `false` to list reports that have not been acknowledged. Type: xs:boolean Default: `either`	No

Note

- See Authorization (p. 31) for information about parameters that you *must* include in every Amazon MWS Query request.
- See User-Agent Header (p. 34) for information about the User-Agent header line, required in every request.

Response Elements

The `UpdateReportAcknowledgements` operation returns an *`UpdateReportAcknowledgements`* response, which is an aggregated element with child elements described in the following table.

Name	Description
`Count`	Total number of reports that matched the query parameters. Type: tns: nonNegativeInteger
`ReportId`	A unique identifier for the report. Type: xs:string

Name	Description
ReportType	The ReportType (p. 95) requested.
ReportRequestId	A unique identifier for the report request. Type: xs:string
AvailableDate	Date the report is available. Type: xs:datetime
Acknowledged	Whether or not this report was acknowledged. Type: xs:boolean
AcknowledgedDate	Date the report was acknowledged. Type: xs:datetime

Examples

Example Query Request

```
https://mws.amazonservices.com/
        ?AWSAccessKeyId=0PB842EXAMPLE7N4ZTR2
        &Action=UpdateReportAcknowledgements
        &ReportIdList.Id.1= 841997483&ReportIdList.id.2= 843337483
        &Acknowledged=true
        &Marketplace=ATVPDKIKX0DER
        &Merchant=A1XEXAMPLE5E6
        &ReportIdList=624169093
        &Signature=cE8%2FUgE8BspmM%2B26UTy7oVEdBk4%3D
        &SignatureVersion=2
        &SignatureMethod=HmacSHA256
        &Timestamp=2009-02-04T18%3A12%3A20.718Z
        &Version=2009-01-01
```

Example Response

```
<?xml version="1.0"?>
<UpdateReportAcknowledgementsResponse
  xmlns="http://mws.amazonaws.com/doc/2009-01-01/">
  <UpdateReportAcknowledgementsResult>
    <Count>1</Count>
    <ReportInfo>
      <ReportId>841997483</ReportId>
      <ReportType>_GET_MERCHANT_LISTINGS_DATA_</ReportType>
      <ReportRequestId>2234038326</ReportRequestId>
      <AvailableDate>2009-01-06T03:48:36+00:00</AvailableDate>
      <Acknowledged>true</Acknowledged>
      <AcknowledgedDate>2009-02-20T02:10:41+00:00</AcknowledgedDate>
    </ReportInfo>
  </UpdateReportAcknowledgementsResult>
  <ResponseMetadata>
    <RequestId>42a578a7-ed92-486b-ac67-5de7464fcdfa</RequestId>
  </ResponseMetadata>
</UpdateReportAcknowledgementsResponse>
```

Enumerations

FeedType

The *FeedType* enumeration is used in feed-related API functions.

For additional information, see Related Resources (p. 2).

Name	Enumeration	Format
Product Feed	_POST_PRODUCT_DATA_	XML
Relationships Feed	_POST_PRODUCT_RELATIONSHIP_DATA_	XML
Single Format Item Feed	_POST_ITEM_DATA_	XML
Shipping Override Feed	_POST_PRODUCT_OVERRIDES_DATA_	XML
Product Images Feed	_POST_PRODUCT_IMAGE_DATA_	XML
Pricing Feed	_POST_PRODUCT_PRICING_DATA_	XML
Inventory Feed	_POST_INVENTORY_AVAILABILITY_DATA_	XML
Order Acknowledgement Feed	_POST_ORDER_ACKNOWLEDGEMENT_DATA_	XML
Order Fulfillment Feed	_POST_ORDER_FULFILLMENT_DATA_	XML
FBA Shipment Injection Fulfillment Feed	_POST_FULFILLMENT_ORDER_REQUEST_DATA_	XML
FBA Shipment Injection Cancellation Feed	_POST_FULFILLMENT_ORDER_CANCELLATION_REQUEST_DATA_	XML
Order Adjustment Feed	_POST_PAYMENT_ADJUSTMENT_DATA_	XML
Flat File Listings Feed	_POST_FLAT_FILE_LISTINGS_DATA_	Tab delimited
Flat File Order Acknowledgement Feed	_POST_FLAT_FILE_ORDER_ACKNOWLEDGEMENT_DATA_	Tab delimited
Flat File Order Fulfillment Feed	_POST_FLAT_FILE_FULFILLMENT_DATA_	Tab delimited
Flat File Order Adjustment Feed	_POST_FLAT_FILE_PAYMENT_ADJUSTMENT_DATA_	Tab delimited
Flat File Inventory Loader Feed	_POST_FLAT_FILE_INVLOADER_DATA_	Tab delimited
Flat File Music Loader File	_POST_FLAT_FILE_CONVERGENCE_LISTINGS_DATA_	Tab delimited
Flat File Book Loader File	_POST_FLAT_FILE_BOOKLOADER_DATA_	Tab delimited

Name	Enumeration	Format
Flat File Price and Quantity Update File	_POST_FLAT_FILE_PRICEANDQUANTITYONLY_UPDATE_DATA_	Tab delimited
UIEE Inventory File	_POST_UIEE_BOOKLOADER_DATA_	Universal Information Exchange Environment (UIEE)

ReportType

The `ReportType` enumeration is used in report-related API functions.

For additional information, see Related Resources (p. 2).

Listings Reports		
Name	**Enumeration/API Function**	**Description**
Open Listings Report	_GET_FLAT_FILE_OPEN_LISTINGS_DATA_ API Function: RequestReport (p. 57)	Tab-delimited flat file open listings report that contains the SKU, ASIN, Price, and Quantity fields. For Marketplace and Seller Central.
Merchant Listings Report	_GET_MERCHANT_LISTINGS_DATA_ API Function: RequestReport (p. 57)	Tab-delimited flat file detailed active listings report for up to 50,000 listings. For Marketplace and Seller Central.
Merchant Listings Lite Report	_GET_MERCHANT_LISTINGS_DATA_LITE_ API Function: RequestReport (p. 57)	Tab-delimited flat file active listings report that contains only the SKU, ASIN, Price, and Quantity fields for items that have a quantity greater than zero. You can use this report for more than 50,000 listings. For Marketplace and Seller Central.

Listings Reports		
Name	**Enumeration/API Function**	**Description**
Merchant Listings Liter Report	_GET_MERCHANT_LISTINGS_DATA_LITER_ API Function: RequestReport (p. 57)	Tab-delimited flat file active listings report that contains only the SKU and Quantity fields for items that have a quantity greater than zero. You can use this report for more than 50,000 listings. For Marketplace and Seller Central.
Canceled Listings Report	_GET_MERCHANT_CANCELLED_LISTINGS_DATA_ API Function: RequestReport (p. 57)	Tab-delimited flat file canceled listings report. For Marketplace sellers only.

Order Reports		
Name	**Enumeration/API Function**	**Description**
Unshipped Orders Report	_GET_FLAT_FILE_ACTIONABLE_ORDER_DATA_ API Function: RequestReport (p. 57)	Tab-delimited flat file report that contains only orders that are not confirmed as shipped. Cannot be scheduled. For Marketplace and Seller Central.
Scheduled XML Order Report	_GET_ORDERS_DATA_ API Function: ManageReportSchedule (p. 81)	Scheduled XML order report. For Seller Central sellers only.

Order Reports		
Name	**Enumeration/API Function**	**Description**
Scheduled Flat File Order Report	_GET_FLAT_FILE_ORDER_REPORT_DATA_ API Function: ManageReportSchedule (p. 81)	Scheduled tab-delimited flat file order report. For Seller Central sellers only.
Flat File Order Report	_GET_FLAT_FILE_ORDERS_DATA_ API Function: RequestReport (p. 57)	Tab-delimited flat file order report that can be requested. For Seller Central sellers only.
Flat File Order Report	_GET_CONVERGED_FLAT_FILE_ORDER_REPORT_DATA_ API Functions: ManageReportSchedule (p. 81) and RequestReport (p. 57)	Tab-delimited flat file order report that can be both scheduled and requested. For Marketplace sellers only.

Settlement Reports		
Name	**Enumeration/API Function**	**Description**
Flat File Settlement Report	_GET_FLAT_FILE_PAYMENT_SETTLEMENT_DATA_ API Function: GetReportList (p. 70)	Tab-delimited flat file settlement report that is automatically scheduled; it cannot be requested through `RequestReport`. For Seller Central sellers only.

Settlement Reports		
Name	**Enumeration/API Function**	**Description**
XML Settlement Report	_GET_PAYMENT_SETTLEMENT_DATA_ API Function: GetReportList (p. 70)	XML file settlement report that is automatically scheduled; it cannot be requested through `RequestReport`. For Seller Central sellers only.
Flat File V2 Settlement Report	_GET_ALT_FLAT_FILE_PAYMENT_SETTLEMENT_DATA_ API Function: GetReportList (p. 70)	Tab-delimited flat file alternate version of the Flat File Settlement Report. Price columns are condensed into three general purpose columns: amount-type, amount-description, and amount. This report is automatically scheduled for FBA sellers; it cannot be requested through `RequestReport`. For Seller Central only.

FBA Reports		
Name	**Enumeration/API Function**	**Description**
Flat File All Orders Report by Last Update	_GET_FLAT_FILE_ALL_ORDERS_DATA_BY_LAST_UPDATE_ API Function: RequestReport (p. 57)	Tab-delimited flat file order report that returns all orders updated in the specified date range regardless of fulfillment channel or shipment status. This report is intended for order tracking, not to drive your fulfillment process; it does not include customer-identifying information and scheduling is not supported. For all Amazon sellers.
Flat File All Orders Report by Order Date	_GET_FLAT_FILE_ALL_ORDERS_DATA_BY_ORDER_DATE_ API Function: RequestReport (p. 57)	Tab-delimited flat file order report that returns all orders placed in the specified date range regardless of fulfillment channel or shipment status. This report is intended for order tracking, not to drive your fulfillment process; it does not include customer-identifying information and scheduling is not supported. For all Amazon sellers.

FBA Reports		
Name	**Enumeration/API Function**	**Description**
XML All Orders Report by Last Update	_GET _XML_ALL_ORDERS _DATA_BY_LAST_UPDATE_ API Function: RequestReport (p. 57)	XML file order report that returns all orders updated in the specified date range regardless of fulfillment channel or shipment status. This report is intended for order tracking, not to drive your fulfillment process; it does not include customer-identifying information and scheduling is not supported. For all Amazon sellers.
XML All Orders Report by Order Date	_GET _XML_ALL_ORDERS _DATA_BY_ORDER_DATE_ API Function: RequestReport (p. 57)	XML file order report that returns all orders placed in the specified date range regardless of fulfillment channel or shipment status. This report is intended for order tracking, not to drive your fulfillment process; it does not include customer-identifying information and scheduling is not supported. For all Amazon sellers.

FBA Reports		
Name	**Enumeration/API Function**	**Description**
FBA Inventory Report	_GET_AFN_INVENTORY_DATA_ API Function: RequestReport (p. 57)	Tab-delimited flat file FBA inventory report. For FBA sellers only. For Marketplace and Seller Central.
FBA Fulfilled Shipments Report	_GET_AMAZON_FULFILLED_SHIPMENTS_DATA_ API Function: RequestReport (p. 57)	Tab-delimited flat file Amazon-fulfilled shipments report. Contains detailed order/ shipment/item information including price, address, and tracking data. For FBA sellers only. For Marketplace and Seller Central.
FBA Returns Report	_GET_FBA_FULFILLMENT_CUSTOMER_RETURNS_DATA_ API Function: RequestReport (p. 57)	Tab-delimited flat file FBA customer returns report. Contains customer returned items received at an Amazon fulfillment center, including Return Reason and Disposition. For FBA sellers only. For Marketplace and Seller Central.

FBA Reports

Name	Enumeration/API Function	Description
FBA Customer Shipment Sales Report	_GET_FBA_FULFILLMENT_CUSTOMER_SHIPMENT_SALES_DATA_ API Function: RequestReport (p. 57)	Tab-delimited flat file Amazon-fulfilled shipments report. Contains condensed item level data on shipped FBA customer orders including price, quantity, and ship to location. For FBA sellers only. For Marketplace and Seller Central.
FBA Promotions Report	_GET_FBA_FULFILLMENT_CUSTOMER_SHIPMENT_PROMOTION_DATA_ API Function: RequestReport (p. 57)	Tab-delimited flat file FBA customer order promotions report. Contains promotions applied to FBA customer orders sold through Amazon; e.g. Super Saver Shipping. For FBA sellers only. For Marketplace and Seller Central.

FBA Reports		
Name	**Enumeration/API Function**	**Description**
FBA Daily Inventory History Report	_GET_FBA_FULFILLMENT_CURRENT_INVENTORY_DATA_ API Function: RequestReport (p. 57)	Tab-delimited flat file FBA daily inventory report. Contains historical daily snapshots of your available inventory in Amazon's fulfillment centers including quantity, location and disposition. For FBA sellers only. For Marketplace and Seller Central.
FBA Monthly Inventory History Report	_GET_FBA_FULFILLMENT_MONTHLY_INVENTORY_DATA_ API Function: RequestReport (p. 57)	Tab-delimited flat file monthly inventory report. Contains historical monthly snapshots of your available inventory in Amazon's fulfillment centers including average and end-of-month quantity, location and disposition. For FBA sellers only. For Marketplace and Seller Central.

FBA Reports		
Name	**Enumeration/API Function**	**Description**
FBA Received Inventory Report	_GET_FBA_FULFILLMENT_INVENTORY_RECEIPTS_DATA_ API Function: RequestReport (p. 57)	Tab-delimited flat file FBA received inventory report. Contains inventory that has completed the receive process at Amazon’s fulfillment centers. For FBA sellers only. For Marketplace and Seller Central.
FBA Inventory Event Detail Report	_GET_FBA_FULFILLMENT_INVENTORY_SUMMARY_DATA_ API Function: RequestReport (p. 57)	Tab-delimited flat file FBA inventory events report. Contains history of inventory events (e.g. receipts, shipments, adjustments etc.) by SKU and Fulfillment Center. For FBA sellers only. For Marketplace and Seller Central.

FBA Reports		
Name	**Enumeration/API Function**	**Description**
FBA Inventory Adjustments Report	_GET_FBA_FULFILLMENT_INVENTORY_ADJUSTMENTS_DATA_ API Function: RequestReport (p. 57)	Tab-delimited flat file FBA inventory adjustment report. Contains corrections and updates to your inventory in response to issues such as damage, loss, receiving discrepancies, etc. For FBA sellers only. For Marketplace and Seller Central.
FBA Inventory Age Report	_GET_FBA_FULFILLMENT_INVENTORY_AGE_DATA_ API Function: RequestReport (p. 57)	Tab-delimited flat file FBA inventory age report. Contains the number of days that items have been stored at Amazon, by SKU. Used for identifying slowly- moving inventory. For FBA sellers only. For Marketplace and Seller Central.

FBA Reports		
Name	**Enumeration/API Function**	**Description**
FBA Replacements Report	_GET_FBA_FULFILLMENT_CUSTOMER_SHIPMENT_REPLACEMENT_DATA_ API Function: RequestReport (p. 57)	Tab-delimited flat file FBA order replacements report. Contains replacements that have been issued to customers for completed orders. For FBA sellers only. For Marketplace and Seller Central.

Product Ads Reports		
Name	**Enumeration/API Function**	**Description**
Product Ads Listings Report	_GET_NEMO_MERCHANT_LISTINGS_DATA_ API Function: RequestReport (p. 57)	Tab-delimited flat file detailed active listings report for up to 50,000 listings. For Product Ads sellers only.

Product Ads Reports		
Name	**Enumeration/API Function**	**Description**
Product Ads Daily Performance by SKU Report, flat file	_GET_PADS_PRODUCT_PERFORMANCE_OVER_TIME_DAILY_DATA_TSV_ API Functions: ManageReportSchedule (p. 81) and RequestReport (p. 57)	Currently available only in the U.S. Tab-delimited flat file containing impression and click counts for every SKU that received traffic. SKUs not seen by customers will not be shown. This will result in a blank report if no SKUs received traffic for the specified day. This report is aggregated by day; requesting more than one day will result in a line per SKU per day. Reports are available by 4am (Pacific Time) the following day and can be requested for dates within the last 90 days. For Product Ads sellers only.

Product Ads Reports		
Name	**Enumeration/API Function**	**Description**
Product Ads Daily Performance by SKU Report, XML	_GET_PADS_PRODUCT_PERFORMANCE_OVER_TIME_DAILY_DATA_XML_ API Functions: ManageReportSchedule (p. 81) and RequestReport (p. 57)	Currently available only in the U.S. XML file containing impression and click counts for every SKU that received traffic. SKUs not seen by customers will not be shown. This will result in a blank report if no SKUs received traffic for the specified day. This report is aggregated by day; requesting more than one day will result in a line per SKU per day. Reports are available by 4am (Pacific Time) the following day and can be requested for dates within the last 90 days. For Product Ads sellers only.

Product Ads Reports		
Name	**Enumeration/API Function**	**Description**
Product Ads Weekly Performance by SKU Report, flat file	_GET_PADS_PRODUCT_PERFORMANCE_OVER_TIME_WEEKLY_DATA_TSV_ API Functions: ManageReportSchedule (p. 81) and RequestReport (p. 57)	Currently available only in the U.S. Tab-delimited flat file containing impression and click counts for every SKU that received traffic. SKUs not seen by customers will not be shown. This will result in a blank report if no SKUs received traffic for the specified week. The weekly reports run from Sunday to Saturday. This report is aggregated by week; requesting more than one week will result in a line per SKU per week. Reports are available by 4am (Pacific Time) the following Monday and can be requested for dates within the last 90 days. For Product Ads sellers only.

Product Ads Reports

Name	Enumeration/API Function	Description
Product Ads Weekly Performance by SKU Report, XML	_GET_PADS_PRODUCT_PERFORMANCE_OVER_TIME_WEEKLY_DATA_XML_ API Functions: ManageReportSchedule (p. 81) and RequestReport (p. 57)	Currently available only in the U.S. XML file containing impression and click counts for every SKU that received traffic. SKUs not seen by customers will not be shown. This will result in a blank report if no SKUs received traffic for the specified week. The weekly reports run from Sunday to Saturday. This report is aggregated by week; requesting more than one week will result in a line per SKU per week. Reports are available by 4am (Pacific Time) the following Monday and can be requested for dates within the last 90 days. For Product Ads sellers only.

Product Ads Reports		
Name	**Enumeration/API Function**	**Description**
Product Ads Monthly Performance by SKU Report, flat file	_GET_PADS_PRODUCT_PERFORMANCE_OVER_TIME_MONTHLY_DATA_TSV_ API Functions: ManageReportSchedule (p. 81) and RequestReport (p. 57)	Currently available only in the U.S. Tab-delimited flat file containing impression and click counts for every SKU that received traffic. SKUs not seen by customers will not be shown. This can result in a blank report if no SKUs received traffic for the specified month. The monthly reports run from the 1st day of the month to the last. This report is aggregated by month; requesting more than one month will result in a line per SKU per month. Reports are available by 4am (Pacific Time) on the 5th day of the following month and can be requested for dates within the last 90 days. For Product Ads sellers only.

Product Ads Reports

Name	Enumeration/API Function	Description
Product Ads Monthly Performance by SKU Report, XML	_GET_PADS_PRODUCT_PERFORMANCE_OVER_TIME_MONTHLY_DATA_XML_ API Functions: ManageReportSchedule (p. 81) and RequestReport (p. 57)	Currently available only in the U.S. XML flat file containing impression and click counts for every SKU that received traffic. SKUs not seen by customers will not be shown. This can result in a blank report if no SKUs received traffic for the specified month. The monthly reports run from the 1st day of the month to the last. This report is aggregated by month; requesting more than one month will result in a line per SKU per month. Reports are available by 4am (Pacific Time) on the 5th day of the following month and can be requested for dates within the last 90 days. For Product Ads sellers only.

Schedule

The `Schedule` enumeration is used in the ManageReportSchedule (p. 81) API function.

Schedule Description	Enumeration
Every 15 minutes	_15_MINUTES_
Every 30 minutes	_30_MINUTES_
Every hour	_1_HOUR_
Every 2 hours	_2_HOURS_
Every 4 hours	_4_HOURS_
Every 8 hours	_8_HOURS_
Every 12 hours	_12_HOURS_
Every day	_1_DAY_
Every 2 days	_2_DAYS_
Every 3 days	_72_HOURS_
Every 7 days	_7_DAYS_
Every 14 days	_14_DAYS_
Every 15 days	_15_DAYS_
Every 30 days	_30_DAYS_
Delete a previously created report schedule	_NEVER_

Error Codes

The following table describes the Amazon MWS error codes. Additional errors, which might be returned due to problems with your feeds, are detailed in the Seller Central Help topics.

Error Code	Description
AccessDenied	Client tried connecting to MWS through HTTP rather than HTTPS.
AccessToFeedProcessingResultDenied	Insufficient privileges to access the feed processing result.
AccessToReportDenied	Insufficient privileges to access the requested report.
ContentMD5Missing	The Content-MD5 header value was missing.
ContentMD5DoesNotMatch	The calculated MD5 hash value doesn’t match the provided Content-MD5 value.
FeedCanceled	Returned for a request for a processing report of a canceled feed.
FeedProcessingResultNoLongerAvailable	The feed processing result is no longer available for download.
FeedProcessingResultNotReady	Processing report not yet generated.
InputDataError	Feed content contained errors.
InternalError	Unspecified server error occurred.
InvalidFeedSubmissionId	Provided Feed Submission Id was invalid.
InvalidFeedType	Submitted Feed Type was invalid.
InvalidParameterValue	Provided query parameter was invalid. For example, the format of the `Timestamp` parameter was malformed.
InvalidQueryParameter	Superfluous parameter submitted.
InvalidReportId	Provided Report Id was invalid.
InvalidReportType	Submitted Report Type was invalid.
InvalidScheduleFrequency	Submitted schedule frequency was invalid.
MissingClientTokenId	Either the Merchant Id or Marketplace Id parameter was empty or missing.
MissingParameter	Required parameter was missing from the query.
ReportNoLongerAvailable	The specified report is no longer available for download.
ReportNotReady	Report not yet generated.
SignatureDoesNotMatch	The provided request signature does not match the server's calculated signature value.

Error Code	Description
UserAgentHeaderLanguageAttributeMissing	The User-Agent header Language attribute was missing.
UserAgentHeaderMalformed	The User-Agent value did not comply with the expected format. See the topic, User-Agent Header (p. 34).
UserAgentHeaderMaximumLengthExceeded	The User-Agent value exceeded 500 characters.
UserAgentHeaderMissing	The User-Agent header value was missing.

and the military transports it overseas by cargo carrier. You can use normal domestic postage on mail to these addresses. Because the cost for APO/FPO addresses is the same as regular domestic mail, it is not possible to set separate, higher shipping rates.

Glossary

catalog — The Amazon catalog is essentially considered to be all product data available to sellers. The output of that data appears on Amazon in various stores, such as Books, Apparel, and Electronics. The catalog includes products offered exclusively by Amazon and products offered by sellers. It also includes any product variations that exist, such as a blouse that comes in red, blue, and green. A variation product might have its own product detail page or not, which is usually determined by category. If the Amazon Web site were a paper catalog, each product would have its own page, except in cases where variations were collected together and shown on one page. Every product has its own catalog identifier, known as an ASIN. The catalog can also contain products that are currently out of stock or that lack product details.

category — Amazon categories are listed on the main Web site tab. Some categories have their own tabs, such as Books and Electronics. Other categories, such as Pet Supplies, have their own storefront but are contained within another, top-level category. How you categorize a product determines which area of the Web site the product will appear.

endpoint — The first line of a request represents the endpoint of the request. This is the domain name of the request. After the endpoint is a question mark (?), which separates the endpoint from the parameters.

feed — [illegible] Seller [illegible] Amazon Services. For [illegible]

flat file — [illegible] you can submit your inventory data using a product inventory feed.

HMAC-SHA1 — A hash message authentication code (HMAC) is a short piece of information used to authenticate a message, calculated using a cryptographic hash function (in this case, the SHA1 function) and a secret key. For more information, go to HMAC: Keyed-Hashing for Message Authentication.

inventory — The total product offerings of a single seller, or the total quantity of a unique product/SKU. The quantity of a product that you list on Amazon must represent the actual number you can fulfill within the expectations set by the shipping time.

inventory file — A tab-delimited text file containing all product-related data that Amazon needs to create a product detail page. Also known as a text file template or product inventory file.

inventory report — A TXT (tab) version of a seller's open listings, sold items, order fulfillment (items sold and paid via Amazon Payments), and listings canceled by Amazon. XML reports can be available if the seller account is configured to receive them.

Item Classification Guide — Documents that are specific to each category that provide valid values for setting up your products. These values take advantage of the browse structure and search functionality on Amazon. These documents are available a:t

- DE: https://sellercentral.amazon.de/gp/help/
- FR: https://sellercentral.amazon.fr/gp/help/
- JP: https://sellercentral-japan.amazon.com/gp/help/home.html
- UK: https://sellercentral.amazon.co.uk/gp/help/
- US: https://sellercentral.amazon.com/gp/help

listing information — Details that represent the products that an individual seller sells; these details can differ from other sellers' listings. For example, you might sell the same Bose speakers as another seller, so the product information can be the same, but your condition note, price and/or handling time, for example, could be different.

offer listing page — The offer listing page displays sales listings for a product offered by a variety of sellers. Depending on the product category, listings can include offers from Amazon Marketplace sellers, Amazon, and other third-party sellers. In certain categories, listings can also be offered in a variety of condition types (New, Used, and so on). Customers reach the offer listing page from the product detail page or from links provided in search results.

product detail page — Each product in Amazon's catalog appears on a product detail page. When you contribute a new product detail page, it becomes part of the Amazon catalog, and other sellers can then offer the same product on that same page. We create the best possible detail pages by combining information provided by our many contributors. There is no guarantee that the product information you provide will appear on Amazon, but your price, availability, and shipping fees will be associated with your listing. Sellers cannot delete product detail pages. If you make an error and are not able to edit it, or if you think a page should be deleted, submit your request through the Suggestion Box at the bottom of the detail page. Our catalog department will review your submission and decide whether or not to make the requested change. The product detail page displays the product ASIN, and possibly a picture, reviews, and links to related products.

product information — Details that are common to every instance of the product on Amazon. For example, several sellers sell Bose speakers on Amazon, and their listing information can differ, but the general details about the product, such as the product name and model number, are the same.

Secret Access Key — A key assigned to you by MWS when you register to use MWS. Used for request authentication. To learn more about request signatures, including when to use them see Calculating Signatures (p. 28).

SKU

A stock keeping unit is a specific seller's product identifier. The SKU is a critical piece of data in every inventory file that you submit to us. Amazon uses the SKUs in your inventory file to associate your products with the appropriate product detail page in our catalog (if one already exists). This means the following:

- Every product in your inventory file must have a unique SKU.
- An existing SKU cannot be changed; it remains in the catalog until you delete it. That is, you cannot change SKU "123" so that it reads "ABC" in our system simply by changing the SKU in your inventory file.

stock keeping unit

See SKU.

tab-delimited file

A text file with data elements that are separated using the tab character. For example, when you upload a product feed, the file must be formatted as a tab-delimited text file.

text file

Sellers use text files to load multiple products at the same time. You can create these files using a program such as Microsoft Excel. After you create a spreadsheet with the data you want in it, you save the file as a *tab-delimited text file*, and then upload this file to Seller Central.

The text files you use in Seller Central are set up using predefined templates. For *inventory files*, these templates are based on the product category.

web service

A web service is an interface to an application.

Index