Amazon Marketplace Web Service Feeds API Section Reference (API Version 2009-01-01)

Contents

Amazon Marketplace Web Service Feeds API Section Reference (API Version 2009-01-01)

What's New

Below is the history of this document.

Change	Description	Release Date
Second Release, API Version 2009-01-01	New section added to the overview section to support multiple marketplaces and EU Global Sellers selling in multiple marketplaces. New request parameter **MarketplaceIdList** added. The **Marketplace** request parameter has been deprecated and is no longer used for authentication. Added throttling limit change from 1,000 to 10,000 requests per hour. Text corrected and improved. Flowchart added for feeds process.	June 2011
First Release, API Version 2009-01-01	The Feeds API section reference is removed from the Amazon MWS Developer Guide and made into its own stand-alone document. Text corrected and improved.	January 2011

What you should know about the Amazon MWS Feeds API section

Overview

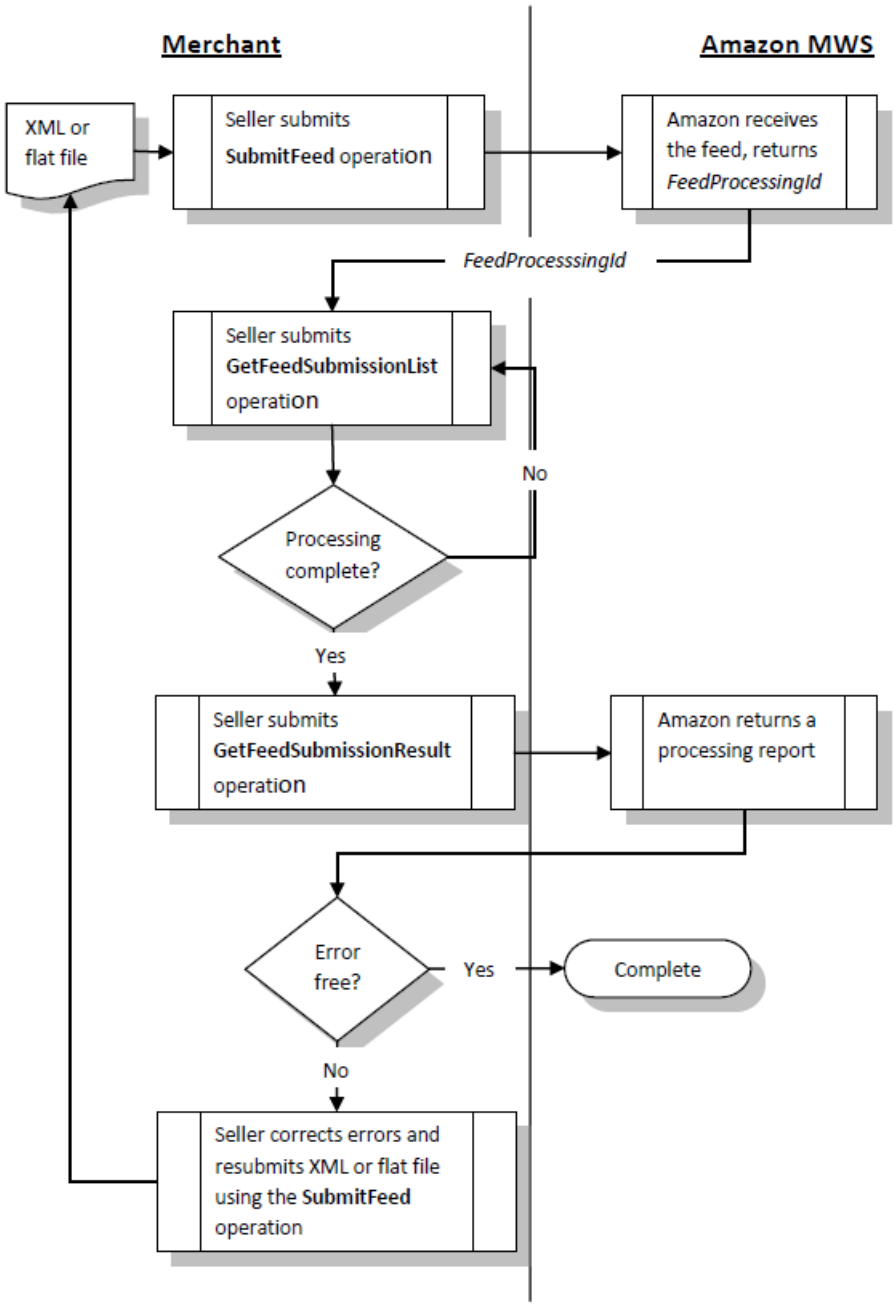
The Amazon MWS Feeds API section of the Amazon Marketplace Web Service (Amazon MWS) API lets you upload inventory and order data to Amazon. You can also use the Amazon MWS Feeds API section to get information about the processing of feeds.

The process for submitting feeds is as follows:

1. Submit an XML or flat file using the `SubmitFeed` operation along with an encrypted header and all required metadata, including a value from the *FeedType enumeration*. As with all submissions to Amazon MWS, you must also include authentication information. The `SubmitFeed` operation returns a *FeedProcessingId*, which you can use to periodically check the status of the feed using the `GetFeedSubmissionList` operation.
2. If Amazon MWS is still processing a request, the **FeedProcessingStatus** element of the `GetFeedSubmissionList` operation returns a status of *_IN_PROGRESS_*. If the processing is complete, a status of *_DONE_* is returned.
3. When the feed processing is complete, you can use the `GetFeedSubmissionResult` operation to receive a processing report that describes which records in the feed were successful and which records generated errors. Note that you have to set up a stream that Amazon MWS uses to write out the report when you submit the `GetFeedSubmissionResult` operation. Use the Amazon MWS Feeds API section client library code for the `GetFeedSubmissionResult` operation to create the stream.
4. Analyze the processing report, correct any errors in the file or transmission, and resubmit the feed using the `SubmitFeed` operation. Repeat the process until there are no errors in the processing report. When the processing report is error free, the transmission is complete.

The following flowchart shows the process for submitting a feed:

Flowchart of the feed submission process in Amazon MWS



Feed Types

Feeds come in two types: flat text files, which you can build using a spreadsheet application, and XML documents. You can find templates for creating flat text files and XML files for inventory and orders on Seller Central 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

If you are using XML-formatted feeds, see the guide *Selling on Amazon Guide to XML* for details about the schemas for the various feed types. For example, page 12 of the guide provides information about the Product Feed schema. If you need to determine the correct ItemType for a feed, see the *Category-Specific XSDs table* in the Seller Central help.

Throttling

Amazon MWS limits requests to 10,000 total requests per hour per the combination of a developer account and Amazon seller account.

The following list explains Amazon MWS throttling concepts:

Request quota	The number of requests that you can submit at one time without throttling. The request quota decreases with every request you submit. The request quota increases at the restore rate.
Restore rate	The rate at which your request quota increases over time, up to the maximum request quota.
Maximum request quota	The maximum size that the request quota can reach.

Most of the operations in the Amazon MWS Feeds API section have a maximum request quota of 10 requests and a restore rate of one request per minute. Two Amazon MWS Feeds API operations have a maximum quota limit and a restore rate that you should be aware of:

- The `SubmitFeed` operation has a maximum request quota of 15 requests before throttling occurs. The restore rate is one request every two minutes.
- The `GetFeedSubmissionResult` operation has a maximum request quota of 15 requests before throttling occurs. The restore rate is one request every minute.

For a complete explanation of throttling, see the Amazon MWS Developer Guide.

The Feeds API section contains the following operations:

Operation	Short Description
SubmitFeed	Uploads a feed for processing by Amazon MWS.
GetFeedSubmissionList	Returns a list of all feed submissions submitted in the previous 90 days.
GetFeedSubmissionListByNextToken	Returns a list of feed submissions using the **NextToken** parameter.
GetFeedSubmissionCount	Returns a count of the feeds submitted in the previous 90 days.
CancelFeedSubmissions	Cancels one or more feed submissions and returns a count of the feed submissions that were canceled.
GetFeedSubmissionResult	Returns the feed processing report and the Content-MD5 header.

Using the Content-MD5 Header with the SubmitFeed and GetFeedSubmissionResult operations

The Content-MD5 header is used as a message integrity check to verify that the decoded data received is the same data that was initially sent. When you submit a feed using the `SubmitFeed` operation, or when you receive a feed report using the `GetFeedSubmissionResult` operation, you must work with a Content-MD5 header.

When you use the `SubmitFeed` operation, you must create a Content-MD5 header, which contains the MD5 hash of the HTTP entity body, and include it in your request. This lets Amazon MWS compare the MD5 hash you create with the MD5 hash it creates when it receives the feed. This process lets Amazon MWS determine if the feed submitted for processing is identical to the feed that was received. This process prevents corrupted descriptive data or pricing product data from appearing on Amazon.

When you use the `GetFeedSubmissionResult` operation, you must calculate an MD5 hash for the received feeds report and compare that with the Content-MD5 header that is included in the response. If the two match, the report was not corrupted in transmission.

The Amazon MWS client libraries provide an easy way to pass in the Content-MD5 header with every Amazon MWS request, as long as you send data that has been stored on disk and an MD5 hash has been created for the data. For more information about working with the Content-MD5 header, see the Amazon MWS Developer Guide.

Using Multiple Marketplaces

If an Amazon seller is registered in multiple marketplaces, then the seller has multiple marketplace Ids associated with their merchant/seller Id. These multiple marketplace Ids can be, for example, an Amazon seller account, an Amazon WebStore, or Checkout by Amazon, where those services are available. Such a seller can submit a feed that is applied to one or several marketplace Ids. Amazon sellers in the EU region can submit feeds to support multiple EU marketplaces where they have registered using a single, unified seller account.

Sellers who sell in multiple marketplaces can manage their inventory levels using the same SKUs across multiple marketplaces. This eliminates the need for the seller to manually keep inventory levels across several marketplaces in-sync. For example, an Amazon seller who also had an Amazon WebStore could have one inventory to manage for the two marketplaces.

With multiple marketplaces comes the concept of a default marketplace. The default marketplace is based on the country that you first registered as a seller in. For example, if you used the DE Seller Central website to register, your default marketplace would be DE and your default marketplace Id would be A1PA6795UKMFR9 (see the table below for seller marketplace Id values).

Note: There are no changes in the feeds process for sellers who sell only their primary marketplace.

Note that the term "global" refers to the marketplaces of the unified EU Seller Central: France, Germany, Italy, and the UK.

The behavior of some feeds have changed for EU sellers registered in multiple marketplaces. For example, when `SubmitFeed` is submitted with **PurgeAndReplace** set to *true*, the purge will be applied to all EU marketplaces specified by the EU seller. When no EU marketplace is specified, then for backwards compatibility all marketplaces within a country will be updated.

Changes in how the Marketplace parameter and a marketplace Id are used

For EU sellers, you do not use a specific country endpoint, such as https://mws.amazonservices.de, to indicate what marketplace a feed is to be applied to. EU sellers can apply changes to a given EU marketplace by specifying that marketplace Id when submitting a feed. You specify what marketplaces you want a feed to apply to by supplying a list of marketplace Ids to the **MarketplaceIdList** parameter. This parameter is used instead of the **Marketplace** parameter in the request; the **Marketplace** parameter is no longer used for authentication and is deprecated, but it remains in the request for backward compatibility.

For example, an EU multiple marketplace seller could specify that a feed be applied to both its FR and DE marketplaces by specifying the **MarketplaceIdList** parameter as follows:

```
&MarketplaceIdList.Id.1=A13V1IB3VIYZZH&MarketplaceIdList.Id.2=A1PA6795UKMFR9
```

Note: If no marketplace Ids are specified when submitting a feed, the feed is applied to all marketplaces that the seller is registered in and that are in the same country as the seller's original marketplace registration.

The following table shows EU marketplace Id values:

Table 1: EU MarketplaceId Values

Marketplace	Domain	Marketplace Id
United Kingdom	www.amazon.co.uk	A1F83G8C2ARO7P
Germany	www.amazon.de	A1PA6795UKMFR9
France	www.amazon.fr	A13V1IB3VIYZZH
Italy	www.amazon.it	APJ6JRA9NG5V4

You can also provide a Webstore marketplace Id or a Checkout By Amazon marketplace Id.

Behavior of Feeds When Submitting Multiple Marketplace Ids

If you include multiple marketplace Ids when submitting a feed request, feeds processing has a more complex behavior. The following are some general rules when submitting a feed request with multiple marketplace Ids:

- Flat file feeds can only be applied to a single country. For example, a flat file feed submission would be accepted if the marketplace Ids were for a UK seller account and a UK WebStore account, but would not be accepted if the marketplace Ids were, for example, for a DE seller account and an FR seller account.
- Feed behavior can vary, depending on the `FeedType` and marketplace Ids submitted. See the table below for information on how a particular `FeedType` behaves when submitted with multiple marketplace Ids.
- If more than one marketplace Id is submitted and one or more of those marketplace Ids fail validation for whatever reason (currency mismatch, language mismatch, country mismatch, one blocked and the other valid), then Amazon MWS returns an error and the submission fails.
- Note that Amazon MWS validates a feed submission before it can be queued for processing. A feed submission passes validation when it contains appropriate marketplace Ids for the `FeedType` submitted. Passing validation does not mean that the feed is correctly formatted or that it will process successfully.

All product feeds in the table below have the same behavior. A feed submission that provides a list of marketplace Ids must provide marketplaces that all share the **same language code** or the feed is rejected at submission time. If no marketplace Ids are specified, the feed is applied to all marketplaces that the seller is registered in and that share the same language code as the seller's default marketplace. An FBA marketplace Id is not a valid marketplace Id for this type of feed.

Table 2: Behavior of Product Feeds when used with multiple marketplace Ids

Feed Type	Description	File Type
_POST_PRODUCT_DATA_	Product Feed	XML
_POST_PRODUCT_RELATIONSHIP_DATA_	Item Relationship Feeds	XML
_POST_ITEM_DATA_	Item Feeds	XML
_POST_OFFER_ONLY_DATA_	Offer Only Feeds	XML
_POST_WEBSTORE_ITEM_DATA_	WebStore Item Feeds	XML

All image feeds in the table below have the same behavior. Image feeds map images to ASINs in the provided marketplaces. If no marketplace Ids are specified, the feed is applied to all marketplaces that the seller is registered in and that are in the same country as the seller's original marketplace registration.

Table 3: Behavior of Image Feeds when used with multiple marketplace Ids

Feed Type	Description	File Type	
_POST_PRODUCT_IMAGE_DATA_	Image Feeds	XML	

All pricing feeds in the table below have the same behavior. A price feed submission that provides a list of marketplace Ids must provide marketplace Ids that all share the **same currency code** or the feed is rejected at submission time. If no marketplace Ids are specified, the feed is applied to all marketplaces that the seller is registered in that share the same currency code as the seller's default marketplace. An FBA marketplace Id is not a valid marketplace Id for this type of feed.

Table 4: Behavior of Pricing Feeds when used with multiple marketplace Ids

Feed Type	Description	File Type	
_POST_PRODUCT_PRICING_DATA_	Price Feeds	XML	

All inventory availability feeds in the table below have the same behavior. Quantity is a global value in relationship with a SKU, so changes to stock levels are reflected in all marketplaces that the SKU is active in. If multiple inventory feeds are processed for the same SKU in different marketplaces, then the quantity of the last uploaded inventory feed from the seller reflects the global inventory level. Setting the item inventory level to 0 effectively sets the item quantity to 0 in all marketplaces and makes the item non-buyable. All listing information is still maintained in the system. An FBA marketplace Id is not a valid marketplace Id for this type of feed.

Table 5: Behavior of Inventory Availability Feeds when used with multiple marketplace Ids

Feed Type	Description	File Type
_POST_INVENTORY_AVAILABILITY_DATA_	Inventory Feeds	XML

All order fulfillment feeds in the table below have the same behavior. All post-order feeds (order acknowledgement, shipment confirmation, refund) refer to an Amazon order Id, which is a globally unique identifier. Therefore, post-order feeds are not marketplace-specific.

Table 6: Behavior of Post-Order Feeds when used with multiple marketplace Ids

Feed Type	Description	File Type
_POST_ORDER_FULFILLMENT_DATA_	shipment confirmation	XML

All order fulfillment cancellation feeds can only be applied to **one country**, though they can apply to multiple marketplaces in that country, such as an Amazon seller marketplace Id, an Amazon WebStore marketplace Id, and a Checkout by Amazon marketplace Id. An FBA marketplace Id is not a valid marketplace Id for this type of feed.

Table 7: Behavior of Order Fulfillment Cancellation Feeds when used with multiple marketplace Ids

Feed Type	Description	File Type
_POST_FULFILLMENT_ORDER_CANCELLATION_REQUEST_DATA_	Fulfillment Order Cancellation Request	XML

All shipping override feeds in the table below have the same behavior. Only a single marketplace can be specified for these feeds. An FBA marketplace Id is not a valid marketplace Id for this type of feed.

Table 8: Behavior of Shipping Override Feeds when used with multiple marketplace Ids

Feed Type	Description	File Type	
_POST_PRODUCT_OVERRIDES_DATA_	Product Shipping Override Feeds	XML	
_POST_SHIPPING_OVERRIDE_DATA_	Shipping Override Feeds	XML	

All flat file feeds in the table below have the same behavior. Flat-file feeds can only be applied to one country, though they can apply to multiple marketplaces in that country, such as an Amazon seller marketplace Id, an Amazon WebStore marketplace Id, and a Checkout by Amazon marketplace Id. An FBA marketplace Id is not a valid marketplace Id for this type of feed.

Table 9: Behavior of Flat File Feeds when used with multiple marketplace Ids

Feed Type	Description	File Type
_POST_FLAT_FILE_LISTINGS_DATA_	Inventory Loader	flat file
_POST_FLAT_FILE_INV_DATA_	Inventory Data	flat file
_POST_FLAT_FILE_INVLOADER_DATA_	Converged flat file Inventory Loader	flat file
_POST_FLAT_FILE_BOOKLOADER_DATA_	Book Loader	flat file
_POST_FLAT_FILE_PRICEANDQUANTITYONLY_ UPDATE_DATA_	Price and Quantity Only Loader	flat file
_POST_FLAT_FILE_CONVERGENCE_LISTINGS_DATA_	Convergence Listings	flat file
_POST_UIEE_BOOKLOADER_DATA_	UIEE Bookloader	flat file

New error messages when submitting multiple marketplace Ids

There are several new error messages that have been added when submitting requests with multiple marketplace Ids:

Error Message	Description
All provided marketplaces for this feed type must have the same default language code. [ABCD], [EFGH] have different default language codes.	Some feeds, such as the _POST_PRODUCT_DATA_ feed, can only be applied to marketplaces that share the same language. The provided marketplace Ids do not share the same language.
All provided marketplaces for this feed type must have the same default currency code. [ABCD], [EFGH] have different default currency codes.	Some feeds, especially those that deal with pricing such as the _POST_PRODUCT_PRICING_DATA_ feed, can only be applied to marketplaces that share the same currency. The provided marketplace Ids do not share the same currency.
All provided marketplaces for this feed type must be based in the same country. [ABCD], [EFGH] have different default country codes.	Flat-file feeds can only be applied to marketplace Ids that are registered in the same country. The provided marketplace Ids do not share the same default country.
The provided marketplaces are correctly associated with your account, but you are prevented from performing this action in the following marketplaces: [ABCD], [ABCD]. Please contact Seller Support in your default marketplace for more information about your account.	There is some issue with your account and the marketplace Id you provided. You can get this error message for several reasons, including not completing a marketplace registration. Contact Seller Support in your home marketplace to clear up the issue.
One or more of the provided marketplaces is an FBA marketplace and cannot be used for this feed/report type: [ABCD], [ABCD].	An FBA marketplace Id is not a valid marketplace Id for this type of feed.
Your feed could not be applied to any marketplaces.	Since you did not provide a marketplace Id, Amazon MWS attempted to determine an appropriate marketplace to use. It was unable to find a marketplace associated with your account that could be used to fulfill your request.

SubmitFeed

Uploads a feed for processing by Amazon MWS.

Description

The `SubmitFeed` operation uploads a file and any necessary metadata for processing. Note that you must calculate a Content-MD5 header for the submitted file. For more information about creating a Content-MD5 header, see *What you should know about the Amazon MWS Feeds API section*.

The `SubmitFeed` operation has a maximum request quota of 15 and a restore rate of one request every two minutes. Requests are included within the overall limit of 10,000 requests per hour for each Amazon seller account and Amazon MWS developer account pair. For definitions of throttling terminology,see *What you should know about the Amazon MWS Feeds API section*.

Feed size is limited to 2,147,483,647 bytes (2^31 -1) per feed. If you have a large amount of data to submit, you should submit feeds smaller than the feed size limit by breaking up the data, or submit the feeds over a period of time. One good practice is to submit feeds with a size limit of 30,000 records/items or submit feeds over a period of time, such as every few hours.

Setting the Content-Type for a feed

Your feeds must be in a valid encoding based on your marketplace and file type, and that encoding must be specified as an HTTP Content-Type header. The following table shows the HTTP Content-Type header you should use for flat files and XML files for each marketplace:

Marketplace	Flat File Content-Type	XML Content-Type
North America and Europe	text/tab-separated-values; charset=iso-8859-1	text/xml
Japan	text/tab-separated-values; charset=Shift_JIS	text/xml
China	text/tab-separated-values;charset=UTF-8 or text/tab-separated-values;charset=UTF-16	text/xml

Submitting Feeds with Multiple Marketplace Ids

If an Amazon seller is registered in multiple marketplaces, then the seller has multiple marketplace Ids associated with their merchant/seller Id. For more information on submitting a feed using multiple marketplace Ids, see *Using Multiple Marketplaces*.

Request parameters

Name	Description	Required	Valid values
FeedContent	The actual content of the feed itself, in XML or flat	Yes	Default: none

Name	Description	Required	Valid values
	file format. You must include the **FeedContent** in the body of the HTTP request. Type: HTTP-BODY		
FeedType	A *FeedType enumeration* value indicating how the data should be processed. Type: xs:string	Yes	Default: None
PurgeAndReplace	A Boolean value that enables the purge and replace functionality. Set to *true* to purge and replace the existing data; otherwise *false*. This value only applies to product-related flat file feed types, which do not have a mechanism for specifying purge and replace in the feed body. **Use this parameter only in exceptional cases**. Usage is throttled to allow only one purge and replace within a 24-hour period. Type: xs:boolean	No	Default: false

Response elements

Name	Description
FeedSubmissionId	A unique identifier for the feed submission. Type: xs:string
FeedType	The type of feed submitted. This is the *FeedType enumeration* value that was provided to the `SubmitFeed` operation.
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 /?&AWSAccessKeyId=0PB842ExampleN4ZTR2
 &Action=SubmitFeed
```

```
  &MarketplaceIdList.Id.1=ATVPDKIKX0DER
  &Merchant=A1XExample5E6
  &FeedType=_POST_PRODUCT_DATA_
  &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
```

The following is an example HTTP body for a `SubmitFeed` request for a health-related product. See the note above on **FeedContent**. The HTTP body in unencrypted form looks like the following:

```
<?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

Amazon MWS responds with the following headers:

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

Amazon MWS responds with an HTTP body like the following:

```
<?xml version="1.0"?>
<SubmitFeedResponse xmlns="http://mws.amazonservices.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

What you should know about the Amazon MWS Feeds API section

GetFeedSubmissionList

GetFeedSubmissionResult

GetFeedSubmissionList

Returns a list of all feed submissions submitted in the previous 90 days.

Description

The `GetFeedSubmissionList` operation returns a list of feed submissions submitted in the previous 90 days that match the query parameters. Use this operation to determine the status of a feed submission by passing in the **FeedProcessingId** that was returned by the `SubmitFeed` operation.

The `GetFeedSubmissionList` request can return a maximum of 100 results. If there are additional results to return, **HasNext** is returned in the response with a *true* value. To retrieve all the results, you can pass the value of the **NextToken** parameter to the `GetFeedSubmissionListByNextToken` operation and repeat until **HasNext** is *false*.

The `GetFeedSubmissionList` operation has a maximum request quota of 10 and a restore rate of one request per minute. Requests are included within the overall limit of 10,000 requests per hour for each Amazon seller account and Amazon MWS developer account pair. For definitions of throttling terminology,see *What you should know about the Amazon MWS Feeds API section*.

Request parameters

Name	Description	Required	Valid Values
FeedSubmissionIdList	A structured list of **FeedSubmmissionId** values. If you pass in **FeedSubmmissionId** values in a request, other query conditions are ignored. Type: xs:string	No	Default: All
MaxCount	A non-negative integer that indicates the maximum number of feed submissions to return in the list. If you specify a number greater than 100, the request is rejected. Type: xs:nonNegativeInteger	No	Default: 10
FeedTypeList	A structured list of one or more *FeedType enumeration* values by which to filter the list of feed submissions. Type: xs:string	No	Default: All feed types

Name	Description	Required	Valid Values
FeedProcessingStatusList	A structured list of one or more feed processing statuses by which to filter the list of feed submissions. Type: xs:string	No	Default: All feed types _SUBMITTED_, _IN_PROGRESS_, _CANCELLED_, _DONE_
SubmittedFromDate	The earliest submission date that you are looking for, in ISO8601 date format. xs:datetime	No	Default: 30 days ago
SubmittedToDate	The latest submission date that you are looking for, in ISO8601 date format. Type: xs:datetime	No	Default: Now

Response elements

Name	Description
NextToken	A generated string used to pass information to another call. Pass the **NextToken** value to the `GetFeedSubmissionListByNextToken` operation if the value of **HasNext** is *true*. Type: xs:string
HasNext	A Boolean value that indicates whether there are more items to retrieve, requiring additional requests to `GetGetFeedSubmissionListByNextToken` to retrieve them. The value *true* means there are more items to retrieve; otherwise *false*. Type: xs:boolean
FeedSubmissionId	A unique identifier for the feed submission. Type: xs:string
FeedType	The type of feed submitted. This is the *FeedType enumeration* value that was provided to the `SubmitFeed` operation.
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.amazonservices.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>
```

Related topics

What you should know about the Amazon MWS Feeds API section

GetFeedSubmissionListByNextToken

GetFeedSubmissionListByNextToken

Returns a list of feed submissions using the **NextToken** parameter.

Description

The `GetFeedSubmissionListByNextToken` operation returns a list of feed submissions that match the query parameters. It uses the **NextToken**, which was supplied in a previous request to either the `GetFeedSubmissionListByNextToken` operation or the `GetFeedSubmissionList` operation where the value of **HasNext** was *true*. `GetFeedSubmissionListByNextToken` requests do not have a specific request limit, but are included in the overall limit of 10,000 requests per hour per seller account.

Request parameters

Name	Description	Required	Valid values
NextToken	A string token returned by a previous request to either `GetFeedSubmissionList` or `GetFeedSubmissionListByNextToken` where the value of **HasNext** was *true*. Type: xs:string	Yes	

Response elements

Name	Description
NextToken	A generated string used to pass information to another call. Pass the **NextToken** value to the `GetFeedSubmissionListByNextToken` operation if the value of **HasNext** is *true*. Type: xs:string
HasNext	A Boolean value that indicates whether there are more items to retrieve, requiring additional requests to `GetGetFeedSubmissionListByNextToken` to retrieve them. The value *true* means there are more items to retrieve; otherwise *false*. Type: xs:boolean
FeedSubmissionId	A unique identifier for the feed submission. Type: xs:string

Name	Description
FeedType	The type of feed submitted. This is the *FeedType enumeration* value that was provided to the `SubmitFeed` operation.
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.amazonservices.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>
```

Related topics

What you should know about the Amazon MWS Feeds API section

GetFeedSubmissionList

GetFeedSubmissionCount

Returns a count of the feeds submitted in the previous 90 days.

Description

The `GetFeedSubmissionCount` operation returns a count of the total number of feeds submitted in the previous 90 days.

The `GetFeedSubmissionCount` operation has a maximum request quota of 10 and a restore rate of one request per minute. Requests are included within the overall limit of 10,000 requests per hour for each Amazon seller account and Amazon MWS developer account pair. For definitions of throttling terminology,see *What you should know about the Amazon MWS Feeds API section*.

Request parameters

Name	Description	Required	
FeedTypeList	A structured list of one or more *FeedType enumeration* values by which to filter the list of feed submissions. Type: xs:string	No	Default: All feed types
FeedProcessingStatusList	A structured list of one or more feed processing statuses by which to filter the list of feed submissions. Type: xs:string	No	Default: All feed types _SUBMITTED_, _IN_PROGRESS_, _CANCELLED_, _DONE_
SubmittedFromDate	The earliest submission date that you are looking for, in ISO8601 date format. xs:datetime	No	Default: 30 days ago
SubmittedToDate	The latest submission date that you are looking for, in ISO8601 date format. Type: xs:datetime	No	Default: Now

Response elements

Name	Description
Count	The total number of feed submissions that match the request parameters. 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.amazonservices.com/doc/2009-01-01/">
  <GetFeedSubmissionCountResult>
    <Count>463</Count>
  </GetFeedSubmissionCountResult>
  <ResponseMetadata>
    <RequestId>21e482a8-15c7-4da3-91a4-424995ed0756</RequestId>
    </ResponseMetadata>
</GetFeedSubmissionCountResponse>
```

Related topics

What you should know about the Amazon MWS Feeds API section

GetFeedSubmissionList

CancelFeedSubmissions

Cancels one or more feed submissions and returns a count of the feed submissions that were canceled.

Description

The `CancelFeedSubmissions` operation cancels one or more feed submissions and returns a count of the canceled feed submissions and the feed submission information. Note that if you do not specify a **FeedSubmmissionId**, all feed submissions are canceled.

Information is returned for the first 100 feed submissions in the list. To return information for more than 100 canceled feed submissions, use the `GetFeedSubmissionList` operation.

If a feed has begun processing, it cannot be canceled.

The `CancelFeedSubmissions` operation has a maximum request quota of 10 and a restore rate of one request per minute. Requests are included within the overall limit of 10,000 requests per hour for each Amazon seller account and Amazon MWS developer account pair. For definitions of throttling terminology,see *What you should know about the Amazon MWS Feeds API section*.

Request parameters

Name	Description	Required	Valid values
FeedSubmissionIdList	A structured list of **FeedSubmmissionId** values. If you pass in **FeedSubmmissionId** values in a request, other query conditions are ignored. Type: xs:string	No	Default: All
FeedTypeList	A structured list of one or more *FeedType enumeration* values by which to filter the list of feed submissions. Type: xs:string	No	Default: All feed types
SubmittedFromDate	The earliest submission date that you are looking for, in ISO8601 date format. xs:datetime	No	Default: 30 days ago
SubmittedToDate	The latest submission date that you are looking for, in ISO8601 date format. Type: xs:datetime	No	Default: Now

Response elements

Name	Description
Count	The total number of feed submissions that match the request parameters. Type: tns: nonNegativeInteger
FeedSubmissionId	A unique identifier for the feed submission. Type: xs:string
FeedType	The type of feed submitted. This is the *FeedType enumeration* value that was provided to the `SubmitFeed` operation.
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.amazonservices.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>
```

Related topics

What you should know about the Amazon MWS Feeds API section

GetFeedSubmissionResult

GetFeedSubmissionResult

Returns the feed processing report and the Content-MD5 header.

Description

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

You should compute the MD5 hash of the HTTP body of the report that Amazon MWS returned to you, and compare that with the Content-MD5 header value that is returned. If the computed hash value and the returned hash value do not match, the report body was corrupted during transmission. You should discard the result and automatically retry the request for up to three more times. Please notify Amazon MWS if you receive a corrupted report body. For more information on the Content-MD5 header, see the Amazon MWS Developer Guide.

The `GetFeedSubmissionResult` operation has a maximum request quota of 15 and a restore rate of one request per minute. Requests are included within the overall limit of 10,000 requests per hour for each Amazon seller account and Amazon MWS developer account pair. For definitions of throttling terminology,see *What you should know about the Amazon MWS Feeds API section*.

Request parameters

Name	Description	Required	Valid values
FeedSubmissionId	The identifier of the feed submission you are requesting a feed processing report for. You can get the **FeedSubmissionId** for a feed using the `GetFeedSubmissionList` operation. Type: xs:string	Yes	A **FeedSubmissionId** value.

Response elements

The `GetFeedSubmissionResult` operation returns the feed processing report and the Content-MD5 header for the returned HTTP 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>T_M_GOOD_83835495</MerchantIdentifier>
 </Header>
 <MessageType>ProcessingReport</MessageType>
 <Message>
  <MessageID>1</MessageID>
  <ProcessingReport>
   <DocumentTransactionID>4319742521</DocumentTransactionID>
  </ProcessingReport>
 </Message>
</AmazonEnvelope>
```

Related topics

What you should know about the Amazon MWS Feeds API section

GetFeedSubmissionList

FeedType enumeration

Enumerates all the feed types that are available through the Feeds API section.

Description

The `FeedType` enumeration includes all the feed types that you can submit using the operations in Amazon MWS Feeds API section.

FeedType Enumeration

Name	Enumeration Value	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

Name	Enumeration Value	Format
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
Flat File Video Loader File	_POST_FLAT_FILE_LISTINGS_DATA_	Tab delimited
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)

Related topics

What you should know about the Amazon MWS Feeds API section

SubmitFeed