FWS to MWS Fulfillment API Sections Migration Guide

Contents

Migrating from Amazon Fulfillment Web Service (Amazon FWS) to the Amazon Marketplace Web Service (Amazon MWS) Fulfillment API sections

Amazon Marketplace Web Service (Amazon MWS) is Amazon's newest API for sellers. It offers extensive documentation and support, and client libraries in Java, C#, and PHP for a fast and easy way to integrate with Amazon. Amazon MWS supports seller API operations that are broader in scope and functionality than any APIs Amazon has previously offered sellers.

The Good News

Moving your code from using Amazon FWS to Amazon MWS might not be as bad as you think. Many of the operations that you used in Amazon FWS are the same as the operations your will find in the Amazon MWS Fulfillment API sections (the Inbound Shipment, Inventory, and Outbound Shipment API sections). The following are some of the improvements you will find using the Amazon MWS Fulfillment API sections:

- A single point of registration and authentication for all Amazon MWS API sections, including the Amazon MWS Fulfillment API sections.
- A single source of customer support for all Amazon MWS API sections, including the Amazon MWS Fulfillment API sections.
- More protection against erroneously creating duplicate shipments.
- More regular call patterns that are easier to learn and more consistent within the Amazon MWS API.
- The Amazon MWS Fulfillment API sections work the same way in all regions where Fulfillment by Amazon is supported.

If you used HTTP query requests with Amazon FWS, you will find the Amazon MWS query request structure to be very similar to what you are used to. If you used SOAP requests with Amazon FWS, you will need to adapt your requests to the Amazon MWS HTTP query request structure. Note that while Amazon FWS provided some support for SOAP using unsupported third-party code libraries, Amazon MWS supports only HTTP query requests. Note, too, that while Amazon FWS returned responses wrapped in SOAP envelopes, Amazon MWS returns XML responses without SOAP.

Each Amazon MWS API section has its own client library that contains code for doing many common tasks for that API section when working with Amazon MWS. By using the code in the Amazon MWS client libraries, you save time and you know the request you send is correctly formatted.

The first two tasks you must accomplish to use Amazon MWS are to register for Amazon MWS and to identify the correct endpoint for your marketplace. From there you need to construct a valid request string, then sign that request string with your Secret Key. Finally, you create a URL with all the information required and submit it to the endpoint. These steps will be discussed in depth in this document.

Amazon MWS Registration

To use the Amazon MWS Fulfillment API sections, you need to register. You can register at these locations, depending on your marketplace:

CN: *http://developer.amazonservices.com.cn*

DE: *http://developer.amazonservices.de*

FR: *http://developer.amazonservices.fr*

IT: *http://developer.amazonservices.it*

JP: *http://developer.amazonservices.jp*

UK: *http://developer.amazonservices.co.uk*

US: *http://developer.amazonservices.com*

To register for Amazon MWS, you must have an Amazon MWS-eligible seller account. These accounts can include:

- A non-individual seller account
- An Amazon WebStore account
- An Amazon Product Ads account
- A Checkout by Amazon account

You simply enter your seller account credentials and Amazon MWS returns several important values that you will use to authenticate your requests to Amazon MWS. For more information on registering for Amazon MWS, see the *Amazon MWS Developer Guide*.

Note: You must register for Amazon MWS in the same marketplace as your seller account. For example, an EU seller must register for Amazon MWS using one of the EU URLs. An EU seller cannot register for Amazon MWS using the US URL.

Authentication

Using Amazon MWS, you pass several values with your query requests that are used for authentication. These values are provided to you when you register for Amazon MWS.

Authentication occurs when you submit a request that includes a signature that you create using the Secret Key you are given when you register for Amazon MWS. How to create this signature and how to format a request is covered in the *Amazon MWS Developer Guide*.

Endpoints

To access the Amazon MWS Fulfillment APIs, use one of the following market-specific endpoints:

Amazon Marketplace	MWS Endpoint
CN	https://mws.amazonservices.com.cn
DE	https://mws.amazonservices.de
FR	https://mws.amazonservices.fr
IT	https://mws.amazonservices.it
JP	https://mws.amazonservices.jp
UK	https://mws.amazonservices.co.uk
US	https://mws.amazonservices.com

Understanding Throttling in Amazon MWS

To use Amazon Marketplace Web Service (Amazon MWS) effectively, you need to understand the concept of throttling. Throttling is the process of limiting the number of requests you can submit in a given amount of time. A request can be when you update an inbound shipment or when you create a fulfillment order. Throttling protects the web service from being overwhelmed with requests and ensures all authorized developers have access to the web service.

Definitions of three values that control Amazon MWS throttling are:

- Request quota - The number of requests that you can submit at one time without throttling. The request quota decreases with each request you submit, and increases at the restore rate.

- Restore rate (also called the recovery rate) - The rate at which your request quota increases over time, up to the maximum request quota.

- Maximum request quota (also called the burst rate) - The maximum size that the request quota can reach.

To apply these ideas, consider this example. Say you want to use the `SubmitFeed` operation, from the MWS Feeds API section, to submit 25 inventory update feeds. The `SubmitFeed` operation has a maximum request quota of 15 and a restore rate of one new request every two minutes. If you submit all 25 feed requests at once, your requests will be throttled after 15 requests. You would then have to resubmit 10 feed requests once the request quota had been restored. Since the restore rate is one request every two minutes, it would take 20 minutes for you to be able to submit the remaining 10 feed requests. So, instead of submitting all the requests and having to resubmit the requests that were throttled, you could automate your process to submit feed requests incrementally.

For example, you could submit 10 feed requests (out of your original 25 feeds), and the request quota would still have five requests left over. You could then wait 10 minutes, and the restore rate would have increased the request quota to 10 (one request every two minutes for 10 minutes gives you five new requests). You could then submit 10 more feed requests. For the remaining five feed requests, you could wait ten more minutes and then submit them. If all things go well, you would have submitted all 25 of your inventory feeds in about 20 minutes.

You should consider automating your requests and have a “back off” process where, if throttling occurs because you reached the maximum request quota or the web service experienced high traffic volumes, you could slow down the number of requests you make and resubmit requests that initially failed.

Creating an Amazon MWS Request

Amazon MWS supports query requests for calling web service actions. Query requests are simple HTTP requests, using the GET or POST method with query parameters in the URL or HTTP body, respectively. Amazon MWS requires the use of HTTPS 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. Query requests must contain an Action parameter to indicate the action to be performed. The response is an XML document.

The *Amazon MWS Developer Guide* goes into detail on how to create a valid HTTP query request. Note that Amazon MWS Scratchpad is a helpful tool for understanding how to create correctly formed HTTP query requests. Amazon MWS Scratchpad automatically generates and displays many of the inputs that are required in a HTTP query request to Amazon MWS. Examining the Amazon MWS responses that Amazon MWS Scratchpad displays for each of your query requests can help you build successful parsing logic.

For more information about Amazon MWS and Amazon MWS Scratchpad, see the links on the Amazon MWS portal page for your region:

CN: *http://developer.amazonservices.com.cn*

DE: *http://developer.amazonservices.de*

FR: *http://developer.amazonservices.fr*

IT: *http://developer.amazonservices.it*

JP: *http://developer.amazonservices.jp*

UK: *http://developer.amazonservices.co.uk*

US: *http://developer.amazonservices.com*

Mapping Operations and Parameters from Amazon FWS to Amazon MWS

The three Amazon MWS Fulfillment API sections (the Inbound Shipment, Inventory, and Outbound Shipment API sections) are different from the three Amazon FWS APIs (the Inbound, Inventory, and Outbound APIs) in the following respects:

- Some operations, input parameters, and response elements were renamed. For example, **MerchantSKU** was renamed **SellerSKU**.

- The functionality of some operations in Amazon FWS was consolidated into fewer operations in the Amazon MWS Fulfillment API sections. For example, the functionality of the `GetFulfillmentIdentifierForMSKU` and the `GetInboundShipmentPreview` operations of the Amazon FWS Inbound API were consolidated into the `CreateInboundShipmentPlan` operation of the Amazon MWS Inbound Shipment API section.

- Some operations in Amazon FWS were deprecated because their functionality was implemented in a different Amazon MWS API section. For example, the `GetFulfillmentIdentifier`, `GetFulfillmentItemByFNSKU`, `GetFulfillmentItemByMSKU`, `ListAllFulfillmentItems`, and `ListAllFulfillmentItemsByNextToken` operations were deprecated. The functionality of these operations can be found in the Amazon MWS Reports API section, using the **FBA Inventory Report** (_GET_AFN_INVENTORY_DATA_).

The following sections provide the details about these changes.

Inbound

This section explains how the Amazon MWS Inbound Shipment API section differs from the Amazon FWS Inbound API.

Operation changes

The following table shows how operations have changed in the Amazon MWS Inbound Shipment API section:

Amazon FWS Inbound operation	Amazon MWS Inbound Shipment operation
`GetInboundShipmentPreview`	`CreateInboundShipmentPlan`
`GetFulfillmentIdentifierForMSKU`	`CreateInboundShipmentPlan` You can also use the **FBA Inventory Report** (_GET_AFN_INVENTORY_DATA_) from the Amazon MWS Reports API section.
`PutInboundShipment`	`CreateInboundShipment` or `UpdateInboundShipment`
`ListInboundShipments`	`ListInboundShipments`: • **ShipmentStatusList** request parameter replaces **ShipmentStatus** request parameter and includes more valid values • **LastUpdatedAfter** request parameter replaces **CreatedAfter** request parameter • **LastUpdatedBefore** request parameter replaces **CreatedBefore** request parameter • **MaxCount** request parameter was removed • **ShipmentName** and **LabelPrepType** response elements were added • **ShipToAddress** and **CreateDate** response elements were removed.
`GetInboundShipmentData`	`ListInboundShipments`

`ListInboundShipmentItems`	`ListInboundShipmentItems:` • **ShipmentId** response element was added.
`PutInboundShipmentItems`	`UpdateInboundShipment`
`PutInboundShipmentData`	`UpdateInboundShipment`
`SetInboundShipmentStatus`	`UpdateInboundShipment`
`DeleteInboundShipmentItems`	`UpdateInboundShipment`
`GetFulfillmentIdentifier`	`CreateInboundShipmentPlan` You can also use the **FBA Inventory Report** (_GET_AFN_INVENTORY_DATA_) from the Amazon MWS Reports API section.
`GetFulfillmentItemByMSKU`	*Same as above*
`GetFulfillmentItemByFNSKU`	*Deprecated*. Functionality can be found in the Amazon MWS Reports API section, using the **FBA Inventory Report** (_GET_AFN_INVENTORY_DATA_).
`ListAllFulfillmentItems`	*Same as above*
`GetServiceStatus (Inbound)`	`GetServiceStatus`

Parameter and element changes

The following table shows how parameters and elements have changed or were renamed in the Amazon MWS Inbound Shipment API section:

Amazon FWS Inbound parameter or element	Amazon MWS Inbound Shipment parameter or element
MerchantSKU	**SellerSKU**
DestinationFulfillmentCenter	**DestinationFulfillmentCenterId**
MerchantSKUQuantityItem	**InboundShipmentPlanRequestItem**
ShipmentPreview	**InboundShipmentPlan**
MerchantSKUQuantityItem	**Items**
ItemCondition	**Condition**
	DistrictOrCounty: New element in the **Address** datatype

Inventory

This section explains how the Amazon MWS Inventory API section differs from the Amazon FWS Inventory API.

Operation changes

The following table shows how operations have changed in the Amazon MWS Inventory API section:

Amazon FWS Inventory operation	Amazon MWS Inventory operation
`GetInventorySupply`	`ListInventorySupply`
`ListUpdatedInventorySupply`	`ListInventorySupply`
`ListUpdatedInventorySupplyByNextToken`	`ListInventorySupplyByNextToken`
`GetServiceStatus (Inventory)`	`GetServiceStatus`

Parameter and element changes

The following table shows how parameters and elements were renamed in the Amazon MWS Inventory API section:

Amazon FWS Inventory parameter or element	Amazon MWS Inventory parameter or element
MerchantSKU	**SellerSkus**
MerchantSKUSupply	**InventorySupply**
FulfillmentNetworkSKU	**FNSKU**
SupplyDetail	**InventorySupplyDetail**
MerchantItem	**InboundShipmentPlanRequestItem**
ItemCondition	**Condition**

Outbound

This section explains how the Amazon MWS Outbound Shipment API section differs from the Amazon FWS Outbound API.

Operation changes

The following table shows how operations have changed in the Amazon MWS Outbound API section:

Amazon FWS Outbound operation	Amazon MWS Outbound Shipment operation
`ListAllFulfillmentOrders`	`ListAllFulfillmentOrders`: • **FulfillmentMethod** request parameter was added • **NumberOfResultsRequested** request parameter was removed
`GetServiceStatus (Outbound)`	`GetServiceStatus`

Parameter and element changes

The following table shows how parameters and elements have changed or were renamed in the Amazon MWS Outbound Shipment API section:

Amazon FWS Outbound parameter or element	Amazon MWS Outbound Shipment parameter or element
MerchantSKU	**SellerSkus**
MerchantFulfillmentOrderId	**SellerFulfillmentOrderId**
MerchantFulfillmentOrderItemId	**SellerFulfillmentOrderItemId**
FulfillmentOrder	**FulfillmentOrders**
	DistrictOrCounty: New element in the **Address** datatype
	EstimatedFees: Removed from the **CreateFulfillmentOrderItem** datatype

NextToken behavior

The **HasNext** response element is not used with the “ByNextToken” operations of the Amazon MWS Fulfillment API sections. Instead, **NextToken** is returned only when there remain more pages of data to return. If **NextToken** is not returned, there are no more pages to return.

GetServiceStatus

The GetServiceStatus operations of Amazon MWS Fulfillment API sections can return response elements that Amazon FWS did not return, such as the **MessageId** element and a message in the **Text** element.

Shipping items to the Amazon Fulfillment Network (AFN)

The following table shows how the process of sending items to the AFN has changed using the Amazon MWS Inbound Shipment API section:

Task	Amazon FWS action	Amazon MWS action
Get a fulfillment identifier for each item that you want to send to the AFN	GetFulfillmentIdentifierForMSKU	CreateInboundShipmentPlan
Get **ShipmentId**, **DestinationFulfillmentCenter**, and **ShipToAddress**	GetInboundShipmentPreview	CreateInboundShipmentPlan
Create inbound shipment	PutInboundShipment	CreateInboundShipment

Making changes to an Inbound Shipment

The following table shows how making changes to an inbound shipment has changed using the Amazon MWS Inbound Shipment API section:

Task	Amazon FWS action	Amazon MWS action
Add items to an inbound shipment	PutInboundShipmentItems	UpdateInboundShipment. For each item that you want to add to an inbound shipment, include **SellerSKU** for the new item.
Delete items from an inbound shipment	DeleteInboundShipmentItems	UpdateInboundShipment. For each item that you want to delete from an inbound shipment, include **SellerSKU** for that item and set **QuantityShipped** to zero.
Change item quantities in an inbound shipment	PutInboundShipmentItems	UpdateInboundShipment. For each item in an inbound shipment for which you want to change the quantity, include **SellerSKU** and change **QuantityShipped** as appropriate.
Change the shipment	SetInboundShipmentStatus	UpdateInboundShipment.

status of an inbound shipment		Change **ShipmentStatus** to WORKING, SHIPPED, or CANCELLED, as appropriate.