**Account API**

The *Account API* lets you manage the different policies that are associated with an eBay seller's account. This includes a seller's return, payment, and fulfillment policies. In addition, sellers can create an manage inventory and store locations, hours of operations (for brick-and-mortar stores). Sellers can also use the Account API to set sales-tax tables for the different jurisdictions in which they operate. Lastly, sellers can opt in to and out of seller programs, as well as review their seller privileges (such as their selling limits).

**Technical overview**

The Account API interfaces with several resources, as outline by the following figure:

Main objects in the Account API:

* fulfillment policy
* payment policy
* return policy
* inventory location
* seller programs
* seller privileges
* eBay marketplace jurisdictions
* tax tables

**Inventory API**

The Inventory API is used to create and manage product offers on eBay marketplaces. The Inventory API has the following entities:

* **Location**: a seller must have at least one inventory location set up before that seller can start creating and publishing offers with the Inventory API. Every inventory location must also have a seller-defined merchant location key value.
* **Inventory Item**: before a product can be sold in an offer on an eBay marketplace, an inventory item record must exist for that product. An inventory item record contains such things as product details, item condition, quantity available. Every Inventory Item must also have a seller-defined SKU value.
* **Offer**: an offer is what becomes a live eBay listing. An offer is first created, and then it is published with the Inventory API. Each offer must be associated with an eBay marketplace, an inventory item, an inventory location, and a category ID. The offer will contain quantity available for the offer, the listing description, an offer price. Every offer must also reference a payment, a fulfillment, and a return business policy.
* **Inventory Item Group**: an inventory item group is necessary if the seller would like to create multiple-variation listings through the Inventory API. An inventory item group is a collection of similar inventory items that might vary on a couple of aspects like color and size.
* **Compatible Products**: the product compatibility calls allow the seller to create a compatible vehicle list, or vehicles that are compatible with a motor vehicle part or accessory.

**Technical overview**

The Inventory API is more-or-less a "standalone" REST-based API, but it does have the following relationships with other APIs:

* Trading API: sellers may use the Trading API to get some listing-related metadata, and may also want to use **GetItem** call to view the listing that they have created through the Offer calls
* Account API: sellers can use the Account API to opt in to Business Policies, and to create and manage business policies (that are used/referenced with the **Create Offer** call in the Inventory API).
* Fulfillment API: sellers can use the Fulfillment API to get order details and manage shipment tracking.

# Fulfillment API

The outcome of a buyer's eBay checkout process is an *order*. Order payments are often processed immediately. However, in some cases (e.g. COD), order payments will be pending until other activities complete.

The Fulfillment API enables sellers to manage the completion of an order in accordance with the payment method and timing specified at checkout. The line items in the order are grouped into one or more packages. As the seller addresses, handles, and ships each package, you use the Fulfillment API to facilitate these activities. The set of specifications for this process is known as a *fulfillment*.

This API covers only the transactions that have completed the checkout process. This includes all direct pay Fixed Price purchases, and any delayed payment purchases (like Auctions) where the buyer has completed the checkout.

For more information about using REST APIs, see [Working with REST](http://developer.ebay.com/devzone/rest/wwr/content/wwr-landing.html).

## Technical overview

The checkout process produces an order and generates its order identifier, which serves as the starting point for the Fulfillment API.

### Primary Fulfillment API objects

Objects are as follows:

Order

The Order object contains information about an order that requires shipping, including:

* Information about the buyer and seller
* Information about the order’s line items
* The plans for packaging, addressing and shipping the order
* The status of payment, packaging, addressing, and shipping the order
* A summary of monetary amounts specific to the order, such as pricing, payments, and shipping costs

Line Item

The LineItem object contains details about one or more units of a sold item in an order, including:

* Information about the item's eBay listing, including listing title, promotions, and the seller's SKU
* All monetary amounts specific to the line item such as pricing, promotions, delivery costs, fees, taxes, and refunds
* The status of packaging, addressing and shipping the line item

Shipping Fulfillment

The ShippingFulfillment object contains information about one package and its delivery, including:

* The plans for packaging, addressing and shipping a set of line items
* The history of the fulfillment
* Shipping and tracking details of the fulfillment
* Estimated and actual delivery dates

Fulfillment Start Instruction

The FulfillmentStartInstruction object contains recommendations for fulfilling an order, including:

* The type of fulfillment
* The shipping carrier and service
* Recipient contact and address information
* The estimated delivery window

**Marketing API Overview**

The Marketing API lets sellers and merchants manage the life cycle of promoted listings and item promotions.

**Technical overview**

The following lists the main objects of the Marketing API.

* CAMPAIGN
  + Ads (the promoted listings)
  + Campaign ID
  + Marketplace ID (site where campaign is hosted)
  + Campaign Name
  + Funding strategy (seller fee)
  + Start and end dates
* AD - Ads are generated in one of 3 ways:
  + From a set of rules that select the listings from the seller's inventory (brands, categoryIds, conditionIds, maxPrice, and minPrice)
  + A list of seller inventory listing IDs
  + A list of eBay listing IDs
* Listing Id
  + An ID of a specific item
  + An ID of the parent of an item with variations, such as a shirt that is available in multiple sizes and colors
* ITEM\_PROMOTION
  + Discount rules
  + Inventory criterion
  + Start and end dates
  + Marketplace ID (site where promotion is hosted)
  + Promotion name and description
  + Image URL
  + Promotion status
* PROMOTION
  + Promotion Id
* PROMOTION\_SUMMARY\_REPORT

**Analytics API**

The **Analytics API** provides key information about an individual seller’s business performance.

Understanding business health is crucial whether a seller wishes to maintain steady business or grow to new heights. Currently the API enables sellers to understand how their buyers are engaging with their listings (GET Traffic Report) and whether sellers are meeting their buyers’ expectations (GET Seller Standards Profile).

Informing on how a seller’s business health changes over time, such as understanding seasonal trends in traffic flow, multiple years of data is available for some metrics. Refer to each API document to understand supported time ranges for reports.

**Technical overview**

The following lists the main objects of the Performance API.

* **Traffic Report**
  + *dimensionKeys* - aggregation method for the report, such as days
  + *metrics* - ype of data for the report, such as total listing page views
  + *records* - individual records of the report, bucketed by dimensionKeys
* **Seller Standards Profile**
  + *program* - marketplace region where a seller conducts business

*cycle* - current or projected evaluation cycle

**The Metadata API**

The *Metadata API* lets you ... .

**Technical overview**

The Metadata API interface ...

Main objects in the Metadata API are:

* fulfillment policy