Returns API Overview

This topic outlines how to use the API to create and process returns.

Create a Return

To create a return, use the CreateReturn operation. When creating a return, make sure to include all required fields as detailed in the API reference topic.

Authorize a Return

After you create a return, you can choose to authorize it.

To authorize a return, use the PerformReturnActions operation. Supply a returnID and actionName in the request body, set to "Authorize" as in the below example:

```
{
  "actionName": "Authorize",
  "returnIDs": ["00000"]
}
```

When successful, the operation returns a breakdown of the return you updated. The return transitions to the **Authorized** state.

Process a Return

After a return is authorized, you can update the quantity of return items received or the items eligible for restocking, refund the return, and/or create a replacement order for the return. You can perform these actions in any order or combination.

It is not required to call any operation to change the return state after completing any of the steps in this section. The action/state transitions of **Await/Pending**, **Receive/Received**, **Restock/Restocked**, **Refund/Refunded**, and **Ship/Shipped** are deprecated (but still supported for backwards-compatibility with legacy applications). Now, a return can directly transition from **Authorized** to **Closed/Cancelled/Rejected**.

Receive and Restock

To update the received or restockable item quantities for an existing return object, update the `returnItem.quantityReceived` or `returnItem.quantityRestockable` properties. The RestockReturnItems operation will restock any returned items.

To obtain an existing return object, use the response from the CreateReturn operation, or use another operation that retrieves a return, such as the GetReturn operation.

Refund

To refund a return:

1. Use the GetPayments operation to retrieve the payments available on the return, which includes payment IDs from the parent order.
2. Refund the payment based on whether you are crediting an existing payment or issuing a new store credit or check:
   - **(Credit Existing Payment)**—Use the PerformPaymentActionForReturn operation, specifying the payment ID you noted in Step 1 as a URI parameter and including a request body similar to:

     ```
     {
       "actionName": "CreditPayment",
       "amount": 10.0
     }
     ```

   - **(Issue New Store Credit or Check)**—Use the CreatePaymentActionForReturn operation, including a request body similar to:

     ```
     {
       "actionName": "CreditPayment",
       "amount": 10.0,
       "newBillingInfo": {
          "paymentType": "StoreCredit" or "Check"
       }
     }
     ```

3. Verify that the responses from the payment operations include return objects that contain the new payment interactions.

Replace

To ship a replacement item, use the CreateReturnShippingOrder operation to create a replacement order for the return. The request body (a collection of `ReturnItemSpecifier`) is optional. If the body is empty, the operation replaces all remaining items marked for replace. If you want to replace only a specific item(s) marked for return, you must specify the item(s) in the request body, such as the one given in the following code block:

```
[
   {
      "quantity": 2,
      "returnItemId": "79bb306d265c4c618077a74301029114"
   },
   {
      "quantity": 1,
      "returnItemId": "97bb03d662c5c41608777a3410201941"
   }
]
```

If successful, the operation returns the new child order. At this point, you must proceed to complete (or cancel) the new child order. You cannot close the parent return if a child order is open.

Handle Returns

If the Reverse Logistics feature is enabled, you can call the Disposition API to mark returned items as handled.

Both the return and handling locations are indicated as a `locationCode` in return item data, with the former being at the return item level and the latter within a `dispositionInfo` object as shown below. This object is an array, allowing you to split quantity into multiple handling cases with different location codes. The `status` field should be either Restock or Other.

```
[{
   "returnItemId": "Item1",
   "quantity": "5",
   "locationCode": "location1",
   "condition":"Good",
   "status": "Restock",
   "dispositionInfo": [
      {
       "quantity": "5",
       "locationCode": "location2",
       "condition": "Good",
       "status": "Restock"
      }
    ]
}]
```

Close a Return

Before you can close a return, you must:

- Authorize the return.
- Refund a payment or return an item on the return.
- Close or cancel any replacement orders created on the return.

To close a return, use the PerformReturnActions operation, using a request body similar to the one shown in the following code block. You can also close returns in bulk using this call. To do so, include additional comma-separated return IDs within the brackets of the `ReturnIds` field shown below.

```
{
   "actionName": "Close",
   "returnIds": [ "09d9d6f55a5734309859974000043da" ]
}
```

When successful, the operation returns a collection of items containing the return(s) you closed. The return transitions to the **Closed** state.