Multiple Addresses API Overview

This topic outlines how to use APIs to manage shipping to multiple destinations. Note that this process does not serve the same function as a standard order with one fulfillment address (single ship). Instead, it allows order items to be fulfilled to more than one address.

Note that the standard Order model used for single ship identifies a single destination in the FulfillmentInfo.Contact.Address field. The Multiple Shipments API instead uses a Checkout model that has a collection of Destinations to represent the fulfillment addresses.

Introduction to Multiple Destinations

Shipping to multiple destinations allows shoppers to send items within an order to different destinations, or to use a combination of fulfillment methods for the items, such as direct ship, in-store pickup, or gift card (digital) delivery. For example, a shopper can purchase three items and choose to send those items to three different destinations. Each destination the shopper chooses to send items to can use a different fulfillment method, such as standard versus expedited shipping. In addition, shoppers can choose to direct ship a subset of order items, while marking other items for in-store pickup or gift card delivery.

To learn more about this feature, including its limitations, refer to theShip to Multiple Addresses topic.

API Structure

The multiple destinations feature applies on the site level (rather than applying to all sites in a tenant). On the theme side, sites with the feature enabled include modified themes with new order confirmation pages that allow shoppers to ship to multiple addresses and use different fulfillment methods. These sites also include order confirmation emails to send to shoppers. On the API side, sites with the feature enabled take advantage of an API resource called Checkout, which manages order items, their intended destinations, and their logical groupings prior to order submission.

Checkout Resource

During the checkout process, the API creates the Checkout resource to track a shopper's order items and their intended destinations. The Checkout resource is active until the shopper submits the order, at which point one or many orders are created based on the data contained in the Checkout resource.

The Checkout resource includes the GetCheckout and GetCheckouts operations, which let you read checkouts available on a site.

Important elements within the Checkout resource include the Destinations subresource and the Groupings collection, which help structure how orders generate after a shopper hits Submit.

Destinations

Each order item contains a destination field that specifies where the item ships to. Direct ship items point to a physical address, digital items point to an email, and in-store pickup items point to a blank field. At the Checkout level, a Destinations collection specifies the full list of destinations for the checkout.

Groupings

Groupings bundle items together that have the same fulfillment type and destination. Direct ship items going to the same destination are grouped together, in-store pickup items are grouped together, and gift card items are grouped together. For example, suppose a shopper adds six items to their cart. Items A and B are direct ship items to different destinations, Items C and D are in-store pickup items, and Items E and F are gift cards. The following groupings result (the exact order of grouping numbers is not important):

Grouping 1

- Item A (direct ship item to Destination A)

Grouping 2

- Item B (direct ship item to Destination B)

Grouping 3

- Item C (in-store pickup item)
- Item D (in-store pickup item)

Grouping 4

- Item E (gift card)
- Item F (gift card)

Order Creation Based on Groupings

After a shopper submits the order (finishing their checkout process), one or many Order resources are created based on the groupings. Groupings that contain direct ship items are placed within their own orders (because the items are shipping to distinct destinations). In-store pickup and gift card groupings are bundled into the first direct ship order created, if it exists (otherwise a new order is created and the groupings are placed within it).

For example, to visualize how orders are created based on groupings, take the same example where a shopper purchases six items. Items A and B are direct ship items to different destinations, Items C and D are in-store pickup items, and Items E and F are gift cards. In this case, two orders will be created based on the four aforementioned groupings, as follows:

Order 1

- Item A (direct ship item to Destination A)
- Item C (in-store pickup item bundled into the first order created)
- Item D (in-store pickup item bundled into the first order created)
- Item E (gift card bundled into the first order created)
- Item F (gift card bundled into the first order created)

Order 2

- Item B (direct ship item to Destination B)

Effect on Discounts, Shipping, and Payment

Because of the grouping and order breakdown for multiple destinations, order-level discounts apply at the shipment level that results from the order creation. For more information about the effect on discounts, and how you must be careful when creating discounts for a site that supports multiple addresses, refer to the Guides topic.

During checkout, the Shipping runtime adds common shipping method prices together across the shipments that will result from the groupings, and displays the total price to the shopper on the confirmation page. After a shopper enters payment information and submits their order, payments are split according to the resulting orders.

Effect on Fraud Check

Fraud checks run on each order created after the shopper submits their checkout. For example, if an order with multiple destinations results in three orders generated, fraud checks run on all three orders individually.

Specify Different Shipment Methods for Items with the Same Destination

Currently, the API bundles direct ship items going to the same destination into one grouping. This means that all items going to the same destination must use the same shipment method. If you want items going to the same destination to use different shipment methods, so that some items arrive faster than others, you need to use Arc.js. In the future, you will have access to an Arc.js action that can further subdivide groupings, allowing you to implement finer control over shipment methods to the same destination (and other such customizations).