Suggestion Logs and Debug

The Suggestion Logs and Suggestion Debug pages of the Order Routing interface help you make calls with the Order Routing REST APIs to retrieve the record of logical decision made during routing and test your configurations.

Suggestion Logs

The Suggestion Logs screen allows you to place a Get Suggestion Log API request via the interface. By using the Candidate Suggestion and Routing Suggestion calls, you can test your order routing configurations by retrieving the list of possible location candidates for a given order and Order Routing's analysis of the best assignment location.

With a Suggestion ID and Order ID, you can then retrieve the log of all logical decisions that Order Routing made to achieve that suggestion result.

Suggestion Debug

The Suggestion Debug page allows further testing of the Routing Suggestion logic based on specific item and inventory data. The page will then process the call and return a breakdown of the result, listing location candidates and the logical steps that were taken to determine the routing result.

Suggestion Request Input

To begin configuring the call, you must start by identifying the items being routed. You can do this is one of two ways:

- Enter item information as table rows by providing identifying information and quantities, clicking **+** to add another row to the table. In this case, you must click **Generate JSON** after entering all item data. The page will then create the request body based on the items you provided.
- Click **Type in JSON Code** to manually write or copy/paste a request body in the text box.

Next, you must configure the inventory information that Order Routing will calculate its suggestion based on. This can use either real inventory or mock data - enable or disable the **Use Real Inventory** option depending on what you intend to use. If this option is enabled, then the text box for manually entering JSON will be disabled.

- If using real inventory, then you can simply insert desired locations for Order Routing to consider in the table and click **Generate JSON** as when configuring the item data. This does not perform any allocation, deallocate, or other inventory adjustment calls - only reads current inventory from your site.
- If using mock inventory, you must provide the full inventory data in the **Inventory Response** text box. This could be a copy/paste of a real Get Inventory API response with some fields adjusted for testing purposes. This gives you the flexibility to test Order Routing's suggestion logic in different scenarios without the need to modify actual inventory data.

Evaluating the Request

When the suggestion request and inventory data are configured, click Run Test. The page will validate the JSON and if there are no errors, it will process the request and display the results in a series of sections that appear at the bottom of the page.

The first information displayed is either the suggested assignment location(s) or a state change, such as if it was moved to Customer Care due to failover actions (shown below). After this initial result report, the suggestion log is displayed as a breakdown of data points and JSON requests starting with the initial request information.

After that initial data, each section describes the logical decisions that Order Routing made as well as listing any relevant failover after-actions. It begins with selecting a routing strategy for the items.

Next, it selects a scenario that it will try to assign the items within.

The locations in the selected scenario are then sorted based on the sort settings configured in Order Routing - however, note that this example only contains one location for testing purposes so it does not reflect how a longer list would be sorted.

After analyzing those locations, any after-actions that had to be taken due to inability to route the items are reported. This example took the ASSIGN_TO_CUSTOMER_CARE action, hence the Customer Care state change reported in the first section.

The results will then conclude with a copy of the summarized response returned from the API, the inventory data response, and the suggestion log in standard JSON format (which can be expanded and hidden with the **+/-** icon).