

Curbside Fulfillment Steps

This guide applies to standalone Curbside Delivery implementations. Order Management users integrating Curbside as an additional fulfillment type can use the standard shipment fulfillment API calls.

After a Curbside order has been imported into Kibo, it will go through several fulfillment steps marked by certain actions taken by either the retailer or customer – Validate Stock, Customer En Route, Customer At Curbside, and Provide to Customer – before it is marked complete. Each action is performed as an API request to advance the order through the fulfillment process. At any point in this process, the order/shipment can be canceled. In that case, refer to the [cancellation guide](#).

See the steps below for descriptions of the appropriate API endpoints and calls. Ask the Kibo team for additional examples and Postman samples if desired.

Validate Stock

A PUT call should be made to the following endpoint after the retail associate has determined how much stock is available at their location to fulfill the order. This API is used when there is complete or partial quantity for all order items available for fulfillment. If there is zero inventory available across the entire order, then the whole order should be canceled through a [Cancel Order or Shipment API](#) instead.

Call Attribute	Description
Endpoint	https://{tenantId}. {host}/api/commerce/shipments/{shipmentNumber}/tasks/Validate%20Stock/completed
Supported Formats	JSON
HTTP Method	PUT

There are two objects supported in the request body, *taskBody* and *handleOption*. The former should always be provided whenever a validate stock call is being made, but the latter is only used in the case of partial pickup.

Parameter	Type	Description
taskBody	object	An object that only contains one string parameter, <i>stockLevel</i> , to indicate whether the entire quantity of all items in the order are in stock (“IN_STOCK”) or if only partial inventory is available (“PARTIAL_STOCK”).
handleOption	object	An object that contains an <i>items</i> array to indicate how much quantity was unavailable for particular items. The quantities in this array will be canceled from the order, while the remaining quantity will continue through fulfillment.

Each entry in the *items* array of *handleOption* should include the following parameters to define unavailable quantity:

Parameter	Type	Description
lineId	integer	The particular line item of the order that did not have the entire quantity available.
quantity	integer	The quantity of the item that is unavailable; or how much should be canceled from the order.
reasonCode	string	Included within a <i>reason</i> object, this should be set to “NoInventory.”
moreInfo	string	Included within a <i>reason</i> object, this is an optional field to provide more explanation about the lack of inventory.

If all item quantities can be fulfilled, the below request body would be used to indicate that all items are in stock. There is no data populated in *handleOption* because there are no items that need to be canceled.

```
{
  "taskBody": {
    "stockLevel": "IN_STOCK"
  },
  "handleOption": {}
}
```

In the case of partial pickup, in which there is some quantity available, then *handleOption* is populated with the unavailable stock. Note that the quantity values listed here are the amounts that will be canceled from the order and not fulfilled. For example, the below request body cancels 3 quantity of Item 1, and 5 quantity of Item 2.

```
{
  "taskBody": {
    "stockLevel": "PARTIAL_STOCK"
  },
  "handleOption": {
    "items": [
      {
        "lineId": 1,
        "quantity": 3,
        "reason": {
          "reasonCode": "NoInventory",
          "moreInfo": ""
        }
      },
      {
        "lineId": 2,
        "quantity": 5,
        "reason": {
          "reasonCode": "NoInventory",
          "moreInfo": ""
        }
      }
    ]
  }
}
```

In both of these cases, the API will return a 200 OK response code if the validation request was successful.

Customer En Route

When the customer clicks the link in their email or SMS notification to indicate that they are on their way to the pick-up location, a PUT call with an empty request body is sent to the below endpoint. There is no data to provide; this API request is used to trigger a notification to the retailer to let them know that the customer is on their way and they should expect to deliver the order soon.

Call Attribute	Description
Endpoint	https://{tenantId}.{host}/api/commerce/shipments/{shipmentNumber}/canceled
Supported Formats	JSON
HTTP Method	PUT

As said above, there is no payload for this call.

```
{}
```

A 200 OK response code should be returned if the API successfully accepts the call.

Customer At Curbside

When the customer has arrived at the fulfillment location, they click the appropriate button from their notifications and are directed to a page that asks them to provide one or more pieces of information to help the store associated deliver their order. The exact fields may vary depending on the retailer's implementation but generally include a parking spot number, method of delivery (e.g. which door or window to place the order in), vehicle make and model, and/or license plate number. The submission of that information is an API call to the following endpoint:

Call Attribute	Description
Endpoint	https://{tenantId}.{host}/api/commerce/shipments/{shipmentNumber}/customerAtCurbside
Supported Formats	JSON
HTTP Method	PUT

The request body will always consist of strings named for each field that the retailer asks the customer for, and the associated values are the customer's input. For the common fields described above, the request may look like this:

```
{  
  "Parking Spot Number": "3",  
  "Method of Delivery": "Place Items In Trunk",  
  "Vehicle Color, Make, Model": "Blue Toyota Prius",  
  "License Plate": "ABC-123"  
}
```

A 200 OK response code can be expected from the API when this information is accepted.

Provide to Customer

When the store associate brings the order out to the curb, the customer will either accept or reject the order. If they accept the order, then it can be marked complete. If they reject the order, then it should be canceled with the [Cancel Shipment API](#). Either case ends the fulfillment process.

To indicate that the customer accepted the shipment, make a call to the following endpoint:

Call Attribute	Description
Endpoint	https://{tenantId}. {host}/api/commerce/shipments/{shipmentNumber}/tasks/Provide%20To%20Customer/completed
Supported Formats	JSON
HTTP Method	PUT

Much like for stock validation, the request body includes *taskBody* and *handleOption* objects. If the order was accepted by the customer, then *taskBody* will indicate that acceptance and no data should be provided in *handleOption*.

Parameter	Type	Description
taskBody	object	An object that only contains one boolean parameter, <i>customerAccepted</i> , to indicate whether the customer accepted the order.
handleOption	object	An object that is left null when the order was accepted.

The request body will always have this payload after the customer accepts the order:

```
{
  "taskBody": {
    "customerAccepted": true
  },
  "handleOption": {}
}
```

A 200 OK response code will be returned if the call is successful.

Order Complete

No special API call needs to be made to mark the order complete after the Provide to Customer step. Once the previous API call has been made, email and SMS events are automatically triggered to send confirmation messages to the

customer and the order/shipment will be marked as completed and fulfilled.