

Reserve Inventory in Cart

You can reserve inventory for products in a shopper's cart for a specified amount of time, during which the inventory will be allocated for that customer even if the order is not yet placed. This allows you to guarantee inventory for low inventory items, such as concert tickets or other popular items.

This feature is available for both OMS-only and implementations with eCommerce. It does not include any core theme updates. Instead, OMS-only implementations must implement this feature using the Reservation APIs to initiate a reservation from an external checkout.

The Reservation Process

When an item is added to the cart and is enabled for reservation, KCCP receives a reservation request and allocates inventory at the appropriate fulfillment location for a certain amount of time. If this timer expires before the order is submitted, the inventory will be deallocated and the line items will be treated as regular non-reserved items.

If there is no inventory available at the selected pickup location for a BOPIS item, Order Routing will determine a transfer location instead and the inventory will be reserved there (however, an error will be returned if transfers are not enabled).

Once the order is submitted, the reserved inventory will be kept available for that order throughout the fulfillment process. Any non-reserved items will be handled per the standard inventory process.

Reservation Updates

If changes are made in the cart, reservations will update accordingly (OMS-only implementations must make the appropriate update call to the [Reservation APIs](#)) and the timer will restart on any updated allocations.

- If the quantity is updated or another item enabled for reservation is added, inventory will be allocated or deallocated accordingly. For example, if you change a quantity from 1 to 3 then a separate allocation will be created for the additional quantity. If an item or quantity is removed, that inventory will be deallocated.
- If the fulfillment method is changed to STH, the allocation behavior will depend on whether a zip code or location code is provided.
 - If a zip code sent in the update request is different from the existing zip code in the cart, then the new zip code will be ignored.
 - If a new location code is sent with the request, then inventory will be allocated at that location regardless of the zip code.
- If the fulfillment method is changed to BOPIS, a location code must be passed. Otherwise, an

error will be returned. Inventory will be allocated at this location for pickup if possible.

- If the pickup location of a BOPIS item is changed, inventory will be deallocated at the previous location and allocated at the new location if possible.

Reservations with Future Inventory

[Future inventory](#) is supported alongside reservations to allocate from upcoming inventory if there isn't enough quantity currently available at a fulfillment location.

- For Ship to Home orders, inventory can be reserved at the location as long as the sum of its available and future inventory meets the requested quantity. Inventory will always be allocated from the current inventory levels first, and then any remaining quantity will be allocated from the future inventory.
- If a Buy Online Pickup in Store location does not have the full quantity available to reserve from current inventory, Order Routing will first look for any transfer locations that can provide the remaining amount with current inventory. If BOPIS transfers are not enabled or the quantity cannot be satisfied from transfer locations, then the system will reserve from future inventory levels at the pickup location.

Enable Reserved Inventory

You must enable reserved inventory at both the product/item level and site level. The exact method by which you enable this for items depends on your implementation type.

eCommerce Settings

If your implementation includes eCommerce, you should first enable Reserve Inventory in Cart for each product in the catalog you want to offer this feature for.

1. Go to **Main > Catalog > Products**.
2. Click the product in the table that you want to enable reservations for, or click **Edit** from the item's dropdown menu.
3. Go to the Properties section of the product configurations.
4. Enable **Reserve Inventory in Cart**.

Reserve Inventory In Cart



5. Click **Save**.

Then, use site settings to specify the time in minutes for which the inventory should be reserved in the cart.

1. Go to **System > Settings > General Settings**.
2. Under Inventory Settings, set the **Reserve Inventory in Cart Interval** amount. If not set, the timer will default to 15 minutes.

Inventory Settings

Reserve inventory in Cart interval ⓘ mins

3. Click **Save**.

OMS-Only Settings

If your implementation is Order Management-only, you cannot set product-level configurations. However, you will still use site settings to specify the time in minutes for which the inventory should be reserved in the cart.

1. Go to **System > Settings > General Settings**.
2. Under Inventory Settings, set the **Reserve Inventory in Cart Interval** amount. If not set, the timer will default to 15 minutes.

Inventory Settings

Reserve inventory in Cart interval ⓘ mins

3. Click **Save**.

API Integration

OMS-only implementations must first integrate the [Reservation APIs](#) with the external cart to create reservations before placing an order.

The [Create Reservation API](#) (POST `commerce/reservation/`) will allocate inventory but not validate the existence of the cart or whether reservation is enabled on the product or not. This allows you to force reservation regardless of the cart or product settings, though it will still initiate a timer as configured in your site settings.

See the example below. In the item data, `linelid` is a required field and you should also provide the Cart Item ID in the `id` field. This helps OMS map inventory allocations from reservations to shipments.

```
{
  "siteId": 22398,
  "tenantId": 18023,
  "userId": "58638e3fb9dc4bea9d9c45d602889a3e",
  "items": [
    {
      "linelid": "1",
      "id": "056a9883c9ea498597e1aea700f81d63",
      "product": {
        "productCode": "Ring",
        "name": "Ring",
        "goodsType": "Physical",
        "properties": [],
        "price": {
          "price": 200,
          "salePrice": 199.5,
          "priceListCode": "default_pl",
          "priceListEntryMode": "Bulk"
        },
        "productType": "STD_Prod_Type",
        "productUsage": "Standard",
        "mfgPartNumber": "",
        "sku": "",
        "upc": "",
        "variationProductCode": ""
      },
      "quantity": "1",
      "fulfillmentLocationCode": "",
      "fulfillmentMethod": "Ship",
      "allocationIds": [],
      "allowsBackOrder": false
    }
  ],
  "changeMessages": [],
  "cartId": "68638e3fb9dc4bea9d9c45d602889a3e",
  "zipCode": "411062",
  "status": "Active"
}
```

After creating the reservation, call the [Get Allocation Status API](#) to check whether or not the allocation was successful. The status will be Pending, Success, or Failed. Based on this status, you should take the appropriate action such as removing the item from the cart in the event of a failed

allocation.

Create Order with Reserved Inventory

After the APIs have been integrated and a reservation has been made, you can create the order in OMS. Be sure to include the Reservation ID at the order level, as well as the following data at the order item level:

- Cart Item ID
- Item
- Quantity
- Fulfillment Type
- Location (if applicable)
- Shipping Zip Code (if applicable)
- `isReservationEnabled` set to true

This is an example of an OMS-only [Create Order call](#) that includes reserved inventory:

```
{
  "version": "1",
  "originalCartId": "68638e3fb9dc4bea9d9c45d602889b5c",
  "reservationId": "1408f4eb21e7000001ece3f500004638",
  "customerAccountId": 1004,
  "isTaxExempt": false,
  "email": "example@example.com",
  "ipAddress": "111.11.111.111",
  "type": "Offline",
  "attributes": [],
  "shippingDiscounts": [],
  "handlingDiscounts": [],
  "handlingAmount": 20,
  "handlingSubTotal": 22,
  "handlingTotal": 22,
  "fulfillmentStatus": "NotFulfilled",
  "isFulfillable": false,
  "notes": [],
  "items": [
    {
      "id": "ca28d594d51e420886a7aef90061f1f4",
      "originalCartItemid": "68638e3fb9dc4bea9d9c45d602889b5a",
      "fulfillmentLocationCode": "W-1",
      "fulfillmentMethod": "Ship",
      "lineId": 1,
      "product": {
        "fulfillmentTypesSupported": [
          "DirectShip",
          "InStorePickup"
        ],
        "properties": [
        ],
      },
      "categories": [],
      "price": {
```

```
"price": 2
},
"discountsRestricted": false,
"isTaxable": true,
"productType": "PW_STDProduct_Extras_1",
"productUsage": "Standard",
"productCode": "sp_01",
"name": "standard product 1",
"goodsType": "Physical",
"isPackagedStandAlone": false,
"stock": {
  "manageStock": true,
  "isOnBackOrder": false
},
"measurements": {
  "height": {
    "unit": "in",
    "value": 1
  },
  "width": {
    "unit": "in",
    "value": 1
  },
  "length": {
    "unit": "in",
    "value": 1
  },
  "weight": {
    "unit": "lbs",
    "value": 1
  }
}
},
"quantity": 1,
"discountTotal": 0,
"itemTaxTotal": 0.17,
"shippingTaxTotal": 0.5,
"shippingTotal": 6,
"handlingAmount": 2,
"feeTotal": 0,
"total": 8.67,
"unitPrice": {
  "extendedAmount": 2,
  "listAmount": 2
},
"productDiscounts": [],
"shippingDiscounts": [],
"isAssemblyRequired": false,
"isReservationEnabled": true
}
],
"validationResults": [],
"billingInfo": {
  "billingContact": {
    "id": 1123,
    "email": "example@example.com",
    "firstName": "Example",
    "middleNameOrInitial": "",
    "lastNameOrSurname": "Customer",
    "companyOrOrganization": "BCT"
```

```
companyOrOrganization": "BCT",
"phoneNumbers": {
  "home": "1111111111111111",
  "mobile": "",
  "work": ""
},
"address": {
  "address1": "717 N HARWOOD ST",
  "address2": "",
  "address3": "",
  "address4": "",
  "cityOrTown": "DALLAS",
  "stateOrProvince": "TX",
  "postalOrZipCode": "75201-6501",
  "countryCode": "US",
  "addressType": "Commercial",
  "isValidated": true
}
},
"isSameBillingShippingAddress": false,
"isRecurring": false
},
"payments": [],
"refunds": [],
"credits": [],
"packages": [],
"pickups": [],
"digitalPackages": [],
"isImport": true,
"isHistoricalImport": false,
"isUnified": true,
"couponCodes": [],
"invalidCoupons": [],
"readyToCapture": false,
"isOptInForSms": false,
"continuityOrderOrdinal": 0,
"userId": "62b5d02b512749c2991a754a3825ba8b",
"id": "1403767e5f48a100019dd2b8000045fa",
"channelCode": "CH_001",
"currencyCode": "USD",
"customerInteractionType": "Unknown",
"fulfillmentInfo": {
  "fulfillmentContact": {
    "id": 1123,
    "email": "example@example.com",
    "firstName": "Example",
    "middleNameOrInitial": "",
    "lastNameOrSurname": "Customer",
    "companyOrOrganization": "BCT",
    "phoneNumbers": {
      "home": "1111111111111111",
      "mobile": "",
      "work": ""
    },
    "address": {
      "address1": "717 N HARWOOD ST",
      "address2": "",
      "address3": "",
      "address4": "",
      "cityOrTown": "DALLAS",
```

```
"stateOrProvince": "TX",
"postalOrZipCode": "75201-6501",
"countryCode": "US",
"addressType": "Commercial",
"isValidated": true
}
},
"shippingMethodCode": "fedex_INTERNATIONAL_ECONOMY"
},
"orderDiscounts": [],
"suggestedDiscounts": [],
"total": 12.49
}
```

Send Zip Code

Once enabled, both eCommerce and Order Management implementations should collect zip code information from the shopper's cart if at least one reserved item will be fulfilled via Ship to Home. This will be used by order routing to provide a location suggestion, and thus is not needed for In-Store Pickup items (as order routing uses the pickup location to determine transfer locations if needed).

The cart will always only accept one zip code in the billing address. You must collect this data and send it to KCCP when requesting a reservation, whether through the Create Order API as shown above or the Create Reservation example in the API integration section. You can change the zip code if needed via the Update Zip Code API.

If the shipping zip code in the checkout is different from the shipping zip code in the cart, inventory will be deallocated from the original location and reallocated to the new location if possible.

Expiration and Pending Items

There are two scenarios in which a reservation may not be guaranteed: if the timer expires or if an item in the cart does not have inventory available for reservation. In both cases, an [event notification](#) will be received.

Expiration

Expiration is the more common scenario, and happens when the reservation timer runs out before the order is submitted. When this happens, you should remove the item from the shopper's cart and display a message that the timer expired. If the shopper is currently in checkout, direct them back to the cart to re-add the item and create a new reservation.

The event will provide the Reservation and Cart IDs, as well as the date and time that the expiration occurred.


```
{
  "eventId":"0000000-0000-0000-0000-0000000000",
  "extendedProperties":[
    {
      "key":"reservationId",
      "value":"1234567890"
    },
    {
      "key":"cartId",
      "value":"0987654321"
    },
    {
      "key":"expirationDateTime",
      "value":"2022-08-21T21:03:10.666+00:00";
    }
  ],
  "topic":"reservation.expired",
  "entityId":"000000000000000000",
  "timestamp":"2022-08-21T21:03:23.666+00:00",
  "correlationId":"000000000000000000000000",
  "isTest":false
}
```

Pending Items

The Kibo Composable Commerce Platform performs inventory checks when the shopper is viewing product detail pages and adding items to their cart, so when an item enabled for reservations is added then there is usually enough in stock to successfully reserve. However, if an error does occur then it's possible that the item may not actually have inventory available for allocation. In this case, the item will still be added to the cart but a pending item will be created for it.

When this occurs, you will receive a Pending Item Created event as shown below. OMS-only implementations should then remove the item from the cart or notify the shopper that the item is out of stock. For eCommerce implementations, KCCP will automatically mark the item out of stock but you will need to customize your theme to display a notice to the shopper on the storefront.

```

{
  "eventId": "3255f036-b4fd-4e3d-b504-af0f016711c5",
  "extendedProperties": [
    {
      "key": "cartId",
      "value": "141642ae19a0a400016e8d8000004635"
    },
    {
      "key": "cartItemId",
      "value": "2f542a9e18bf4500830baf0900d2901d"
    },
    {
      "key": "ucp",
      "value": "res_02"
    },
    {
      "key": "quantity",
      "value": "1"
    }
  ],
  "topic": "inventory.cart_pending_items_create",
  "entityId": "141642ae19a0a400016e8d8000004635",
  "timestamp": "2022-09-07T12:46:41+00:00",
  "correlationId": "00000000000000000000000000000000",
  "isTest": false
}

```

Additional Reservation APIs

After creating a reservation, you can [update its details](#), [restart its timer](#), and [change its zip code](#) as needed via API before the order has been placed. You can also check its status at any time with a [Get Reservation request](#). The status indicates whether inventory is currently allocated and whether the order has been placed yet.

- **Active:** The order has not yet been placed and reserved inventory is allocated.
- **Expired:** The time limit has been reached and the inventory has been deallocated.
- **Closed:** The order with a reservation has been submitted via the Create Order API.

Add items, change quantities, and make other changes to existing reservations before order placement with the Reservation Items API. If adding items, you must include the zip code in the endpoint (`/commerce/reservation/{reservationId}/items?zipCode={zipCode}`).

For the full list of available endpoints and their schemas, see [the API documentation](#).