Orders API Overview

This topic provides an overview of the orders and order drafts REST API components. Refer to Orders in the Guides section for more information about orders, including order attributes, order statuses, and managing orders in the Admin interface.

Orders Resource

You can interact with orders and perform order operations using the orders resource: `commerce/orders`.

Order Operations

There are a number of operations you can perform on orders and order subresources such as order items and order attributes - see the API documentation resources linked above. However, key order-level operations are:

- **GetOrder**
- **GetOrders**
- **CreateOrder**
- **CancelOrder**
- **UpdateOrder**

**Note: Credit Cards in Create Order**

There are two similar fields relating to credit card numbers - `cardNumberPartOrMask` and `ccLastFour`. Order Management-only configurations store the credit card token in the card number field, which may result in a different set of last four digits than the actual card number. Thus, they should use `ccLastFour` to record the actual last four digits of the card number so that these digits can be displayed in the order details of the Admin UI - if not provided, Kibo will display the last four digits of the tokenized number.

However, eCommerce and eCommerce+Order Management configurations use the `cardNumberPartOrMask` field as described in the Create Order API model in which its value is the actual or masked credit card number. Thus, `ccLastFour` is not required because Kibo will display the last four digits from `cardNumberPartOrMask` in the Admin UI. However, using either one or both fields is acceptable for validation - if both are provided, then `ccLastFour` will take priority.

Edit and Cancellation Restrictions

After an order has been submitted, you can use the Update Order Restrictions API to set two flags which will determine whether a customer service representative is allowed to edit or cancel the order. These flags are `restrictEdit` and `restrictCancellation`. If either is set to true, then only roles with the Override Order Update Restriction user behavior will be able to edit and/or cancel the order. If not set, then they will default to false and the order will not be restricted.

If the user has been restricted from editing the order, then it will be displayed in a read-only state on the Admin (but
they will still be able to edit internal order notes and initiate or process returns). If the user has been restricted from cancelling the order but they are still allowed to edit it, then they will be able to cancel shipment line items.

The call is made using the below format:

`.../commerce/orders//updateorderrestrictions?restrictEdit=true&restrictCancellation=true`

**Importing Orders**

There are a couple scenarios in which you may need to import orders from another system into the Kibo Composable Commerce Platform (KCCP). This process uses the Create Order API but there is different requirements and behavior to be aware of depending on your scenario.

**Import Orders for OMS-Only**

Order Management-only implementations that do not use Kibo eCommerce need to import orders from their front end. This process uses the CreateOrder API but requires the following basic information:

- Email
- Customer Account ID
- Billing Info
- Fulfillment Info
- Payments
- Pricing Fields
- Items
- The `isImport` flag should be set to "true"

A full example of an import request is provided at the end of this section, which details the required parameters for a basic Order Management import.

**Billing Information**

The `billingInfo` object should be populated for both order and payment objects, containing the billing contact, address, and card.

**Fulfillment Information**

The `fulfillmentInfo` contains the fulfillment contact and shipping address of the customer, as well as the shipping method name and code.

Supported shipping methods will depend on the configurations you enabled in your shipping settings. Each shipping method is identified with a code, which can be mapped to your front end system. If you don’t allow customers to choose their carrier or a specific shipping method on your front end, you can send in a generic Kibo code instead. For a list of these shipping method and generic codes, see the Shipment Types guide.
Payment Information

Credit card authorization will happen on the external front end, but details will need to be sent in the Create Order request to perform subsequent operations on the payment such as capturing and crediting. It is your responsibility to tokenize the credit card with the gateway provider and send the tokenized card in clear text along with authorization details.

Card details are sent in the `payment.billingInfo.card` object:

- `card.cardNumberPartOrMask` will hold the tokenized card number.
- `card.isTokenized=true` indicates that the card is tokenized.

Authorization details from the gateway are included in the interaction object:

- Set the `interactionType` to "Authorization" and the status to "Authorized."
- Set the `amount` requested field to the authorized amount on the payment.
- Send required gateway information in `gatewayResponseData`.

The below example shows a payload for the Cybersource gateway and a tokenized card.

```json
{
  "paymentType": "CreditCard",
  "paymentWorkflow": "Mozu",
  "billingInfo": {
    "paymentType": "CreditCard",
    "billingContact": {
      "email": "sam.billing@email.com",
      "firstName": "Sam",
      "middleNameOrInitial": "",
      "lastNameOrSurname": "Billing",
      "phoneNumbers": {
        "home": "1234567895",
        "mobile": "1234567895",
        "work": ""
      },
      "address": {
        "address1": "1845 Kramer Ln",
        "address2": "",
        "address3": "",
        "address4": "",
        "cityOrTown": "Austin",
        "stateOrProvince": "TX",
        "postalOrZipCode": "78758",
        "countryCode": "US",
        "addressType": "Residential",
        "isValidate": false
      }
    },
    "isSameBillingShippingAddress": false,
    "card": {
      "isUsedRecurring": false,
      "nameOnCard": "Sam Billing",
      "isCardInfoSaved": false,
      "paymentOrCardType": "VISA",
      "cardNumberPartOrMask": "0160184007701111",
      "isTokenized": true
    }
  }
}
The Authorize.net gateway adapter supports tokenization from Authorize.net using customer profiles. The required information is:

- `card.cardNumberPartOrMask`
- `customerPaymentProfileId` in `payment.data`
- `TransactionId` and `ApprovalCode (authCode)` in `interactions.gatewayResponseData`

```json
"cardNumberPartOrMask": "0160184007701111",
"isTokenized": true,
"expireMonth": 12,
"expireYear": 2021
},
"auditInfo": {
  "updateDate": "2018-09-17T17:54:21.027Z",
  "createDate": "2018-09-17T17:53:24.598Z",
  "updateBy": "355060a60a5e48eeb7f2fb8d92af2ba5",
  "createBy": "355060a60a5e48eeb7f2fb8d92af2ba5"
}
},
"status": "Authorized",
"subPayments": [],
"interactions": [
  {
    "gatewayInteractionId": 155627160,
    "paymentId": "d5abdc686f4b4247ae75a95e01271446",
    "currencyCode": "USD",
    "interactionType": "Authorization",
    "status": "Authorized",
    "paymentEntryStatus": "New",
    "isRecurring": false,
    "isManual": false,
    "gatewayTransactionId": "40018563191",
    "gatewayAuthCode": "ASQLVO",
    "gatewayAVSCodes": "Y",
    "gatewayCVV2Codes": "P",
    "gatewayResponseCode": "1",
    "gatewayResponseText": "This transaction has been approved.",
    "gatewayResponseData": [
      {
        "key": "AuthorizationRequestId",
        "value": "5642607567736184003012"
      },
      {
        "key": "AuthorizationRequestToken",
        "value": "Ahj/7wSTMeLpFwo2JfNENyzVs0ZNmDdq2bt2bZi4aMGNg7ZKNP3ock7AVGn70OSd6QNmBx/EMmkmW6QHez7QyZjxdluFGxIs4gLxmU"
      },
      {
        "key": "currencyCode",
        "value": "USD"
      }
    ],
    "amount": 192.50,
    "interactionDate": "2018-09-17T17:54:21.098Z"
  }
]
```
{
    "paymentType": "CreditCard",
    "paymentWorkflow": "Mozu",
    "billingInfo": {
        "paymentType": "CreditCard",
        "billingContact": {
            "email": "sam.billing@email.com",
            "firstName": "Sam",
            "middleNameOrInitial": "",
            "lastNameOrSurname": "Billing",
            "phoneNumbers": {
                "home": "1234567895",
                "mobile": "1234567895",
                "work": ""
            },
            "address": {
                "address1": "1845 Kramer Ln",
                "address2": "",
                "address3": "",
                "address4": "",
                "cityOrTown": "Austin",
                "stateOrProvince": "TX",
                "postalOrZipCode": "78758",
                "countryCode": "US",
                "addressType": "Residential",
                "isValidated": false
            }
        },
        "isSameBillingShippingAddress": false,
        "card": {
            "isUsedRecurring": false,
            "nameOnCard": "Sam Billing",
            "isCardInfoSaved": false,
            "paymentOrCardType": "VISA",
            "cardNumberPartOrMask": "1510780557",
            "isTokenized": true,
            "expireMonth": 12,
            "expireYear": 2021
        }
    },
    "status": "Authorized",
    "subPayments": [],
    "data": {
        "customerPaymentProfileId": "1510480795"
    },
    "interactions": [
        {
            "gatewayInteractionId": 155627160,
            "paymentId": "d5abdc686f4b4247ae75a95e01271446",
            "currencyCode": "USD",
            "interactionType": "Authorization",
            "status": "Authorized",
            "paymentEntryStatus": "New",
            "isRecurring": false,
            "isManual": false,
            "gatewayTransactionId": "40018563191",
            "gatewayAuthCode": "ASQLVO",
            "gatewayAVSCodes": "Y",
            "gatewayCVV2Codes": "P",
            "gatewayResponseCode": "1"
        }
    ]
}
gatewayResponseCode: 1,
gatewayResponseText: "This transaction has been approved.",

gatewayResponseData: [
  {
    "key": "TransactionId",
    "value": "40045193885"
  },
  {
    "key": "ApprovalCode",
    "value": "C7TZL9"
  }
],
amount: 192.5,
interactionDate: "2018-09-17T17:54:21.098Z"
],
isRecurring: false,
amountCollected: 0,
amountCredited: 0,
amountRequested: 192.5
}

Fraud Information

Fraud check happens on your external front end system. Send the fraud results in the validationResults object, such as in the below Kount example.

Set the status to either "Pass" or "Review." Review will put the order in Pending Review status.

"validationResults": [
  {
    "validationId": "630000",
    "validatorName": "KountValidator",
    "validatorType": "Fraud",
    "status": "Review",
    "createdDate": "2015-07-01T18:45:56.013Z",
    "messages": [
      {
        "messageType": "{VERS",
        "message": "0630"
      },
      {
        "messageType": " EMAILS",
        "message": "1"
      },
      {
        "messageType": " CARDS",
        "message": "1"
      },
      {
        "messageType": " VMAX",
        "message": "0"
      },
      {
        "messageType": " DEVICE_LAYERS",
        "message": "...."
      },
      {
        "messageType": " TRAN"
No-Op and Third Party Payments

For no-operation payments, send the payment as a credit card with dummy payment details and set up a no-op gateway to process different card types. This gateway will return a success response for every transaction on the payment without any real interaction with the payment gateway.

Paypal Express 2 is supported as a third party payment for imported orders. You need to have the Paypal Express 2 application installed on your tenant. The Create Order request should include:

- The “PaypalExpress2” payment type
- PayPal token in payment.externalTransactionId
- Payer ID in payment.billingInfo.data
- Auth transaction ID in payment.interaction.gatewayTransactionId

Below is a sample payment object with the PayPal payment.
{  "paymentType": "PayPalExpress2",  "paymentWorkflow": "PayPalExpress2",  "externalTransactionId": "EC-74R44913L24993252",  "billingInfo": {    "paymentType": "PayPalExpress2",    "paymentWorkflow": "PayPalExpress2",    "isSameBillingShippingAddress": false,    "billingContact": {      "email": "example@kibocommerce.com"    },    "data": {      "paypal": {        "payerId": "B373JG5S4Y388"      }    }  },  "status": "Authorized",  "subPayments": [],  "interactions": [    {      "id": "6cc37e2c8b6d41e8a2afaaaf6010c3108",      "paymentId": "b719cda9a6734d9599c3aaf6010bc5dc",      "orderid": "0eba14f72bb54200014d8a720000433f",      "target": {        "targetType": "Order",        "targetId": "0eba14f72bb54200014d8a720000433f",        "targetNumber": 2022      },      "interactionType": "Authorization",      "status": "Authorized",      "paymentEntryStatus": "New",      "isRecurring": false,      "isManual": false,      "gatewayTransactionId": "8AF73994TM546221D",      "gatewayResponseCode": "200",      "gatewayResponseText": "Success - c3b9386056e4a",      "amount": 26.10,      "interactionDate": "2019-10-30T16:16:27.329Z",      "auditInfo": {}    }  ],  "isRecurring": false,  "amountCollected": 0,  "amountCredited": 0,  "amountRequested": 26.10}

The field mappings that are taken into consideration are listed below. The sum of these fields across all items on the order should equal the order total, or else a validation error will be thrown.

- **Unit Price**: `item.product.price.tenantOverridePrice`, `item.product.price.SalePrice`, and `item.product.price.price` (pre discount)
- **Product Discount**: `item.discountTotal`
- **Product Tax**: `item.itemTaxTotal`
• Shipping Total: item.shippingTotal (raw shipping cost + shipping discounts)
• Shipping Discount: item.shippingDiscounts
• Shipping Tax: item.shippingTaxTotal
• Handling: item.handlingAmount (raw handling amount, no discounts)
• Handling Discount: item.weightedOrderHandlingFeeDiscount
• Handling Tax: item.weightedOrderHandlingFeeTax
• Duty: item.dutyAmount

**Item Product Information**

Provide as many product details from the API model as possible for the item. Some of the important fields are:

• item.product.name
• item.product.imageUrl
• item.product.price
• item.product.isTaxable
• item.product.productCode
• item.product.variationProductCode
• item.product.measurements
• item.product.stock.isOnBackOrder
• item.product.options

**OMS-Only Example**

This is a complete import order request.

```json
{
  "isTaxExempt": false,
  "email": "example@kibo.net",
  "ipAddress": "2001:0db8:85a3:0000:0000:8a2e:0370:7334",
  "type": "Offline",
  "externalId": "11182019_3",
  "handlingSubTotal": 0,
  "handlingTotal": 0,
  "notes": [],
  "items": [
    {
      "fulfillmentLocationCode": "AUS",
      "fulfillmentMethod": "Pickup",
      "product": {
        "fulfillmentTypesSupported": [
          "DirectShip",
          "InStorePickup"
        ],
        "imageAlternateText": "",
        "imageUrl": "//example.com/exampleImageURL",
        "options": [],
        "properties": []
      }
    }
  ]
}
```
{  
  "attributeFQN": "tenant~add-shipping",
  "name": "Add Shipping",
  "dataType": "String",
  "isMultiValue": false,
  "values": [
    {  
      "stringValue": "3.50",
      "value": "3.50"
    }
  ]
},
{
  "attributeFQN": "tenant~free-shipping",
  "name": "Free Shipping",
  "dataType": "Bool",
  "isMultiValue": false,
  "values": [
    {  
      "value": false
    }
  ]
}
],
"categories": [
  {  
    "id": 2
  },
  {  
    "id": 6,
    "parent": {  
      "id": 5
    }
  }
],
"price": {  
  "price": 15
},
"discountsRestricted": false,
"isTaxable": true,
"productType": "Basic Standard",
"productUsage": "Standard",
"bundledProducts": [],
"productCode": "sp_02",
"name": "Standard Product - 2",
"description": "Standard Product - 2",
"goodsType": "Physical",
"isPackagedStandAlone": false,
"measurements": {  
  "height": {  
    "unit": "in",
    "value": 1
  },
  "width": {  
    "unit": "in",
    "value": 1
  },
  "length": {  
    "unit": "in",
    "value": 1
  }
}
"value": 1
},
"weight": {
  "unit": "lbs",
  "value": 2.5
}
},
"quantity": 2,
"subtotal": 30,
"extendedTotal": 30,
"taxableTotal": 30,
"discountTotal": 0,
"discountedTotal": 30,
"itemTaxTotal": 0,
"shippingTotal": 0,
"shippingTaxTotal": 0,
"handlingAmount": 0,
"feeTotal": 0,
"total": 30,
"unitPrice": {
  "extendedAmount": 15,
  "listAmount": 15
},
"productDiscounts": [],
"shippingDiscounts": [],
"shippingAmountBeforeDiscountsAndAdjustments": 0,
"weightedOrderAdjustment": 0,
"weightedOrderDiscount": 0,
"adjustedLineItemSubtotal": 30,
"totalWithoutWeightedShippingAndHandling": 30,
"weightedOrderTax": 0,
"weightedOrderShipping": 0,
"weightedOrderShippingDiscount": 0,
"weightedOrderShippingManualAdjustment": 0,
"weightedOrderShippingTax": 0,
"weightedOrderHandlingFee": 0,
"weightedOrderHandlingFeeDiscount": 0,
"weightedOrderDuty": 0,
"totalWithWeightedShippingAndHandling": 30,
"weightedOrderHandlingAdjustment": 0
},
{
  "fulfillmentLocationCode": "AUS",
  "fulfillmentMethod": "Pickup",
  "product": {
    "fulfillmentTypesSupported": [
      "DirectShip",
      "InStorePickup"
    ],
    "imageAlternateText": "",
    "imageUrl": "//example.com/exampleImageURL",
    "options": [],
    "properties": [
      {
        "attributeFQN": "tenant~add-shipping",
        "name": "Add Shipping",
        "dataType": "String",
        "isMultiValue": false.
      }
    ]
  }
}
"status": "Authorized",
"subPayments": [],
"interactions": [
{
  "gatewayInteractionId": 155627160,
  "paymentId": "d5abdc686f4b4247ae75a95e01271446",
  "currencyCode": "USD",
  "interactionType": "Authorization",
  "status": "Authorized",
  "paymentEntryStatus": "New",
  "isRecurring": false,
  "isManual": false,
  "gatewayTransactionId": "40018563191",
  "gatewayAuthCode": "ASQLVO",
  "gatewayAVSCodes": "Y",
  "gatewayCVV2Codes": "P",
  "gatewayResponseCode": "1",
  "gatewayResponseText": "This transaction has been approved.",
  "gatewayResponseData": [
    {
      "key": "AuthorizationRequestId",
      "value": "5642607567736184003012"
    },
    {
      "key": "AuthorizationRequestToken",
      "value": "Ahj/7wSTMeLpFwo2jFnENyzVs0ZNmDdq2bt2bZi4aMGDNgxZKNP3ock7AVGn70OSd6QnBx/EMmkW6QHez7QjyZjxdluFGxl4gLxmU"
    },
    {
      "key": "currencyCode",
      "value": "USD"
    }
  ],
  "amount": 30,
  "interactionDate": "2018-09-17T17:54:21.098Z"
},
"isRecurring": false,
"amountCollected": 0,
"amountCredited": 0,
"amountRequested": 30
},
{
  "paymentType": "CreditCard",
  "paymentWorkflow": "Mozu",
  "billingInfo": {
    "paymentType": "CreditCard",
    "billingContact": {
      "email": "example@kibo.net",
      "firstName": "Ming",
      "lastNameOrSurname": "Ming",
      "phoneNumbers": {
        "mobile": "512-555-5555"
      },
      "address": {
        "address1": "123 ABC",
        "cityOrTown": "Austin",
        "stateOrProvince": "TX",
        "postalOrZipCode": "78758",
        "countryCode": "US"
      }
    }
  }
}
"countryCode": "US",
  "addressType": "Residential",
  "isValidated": false
}

"isSameBillingShippingAddress": false,
"card": {
  "isUsedRecurring": false,
  "nameOnCard": "Sam Billing",
  "isCardInfoSaved": false,
  "paymentOrCardType": "VISA",
  "cardNumberPartOrMask": "0160184007701111",
  "isTokenized": true,
  "expireMonth": 12,
  "expireYear": 2021
},

"auditInfo": {
  "updateDate": "2018-09-17T17:54:21.027Z",
  "createDate": "2018-09-17T17:53:24.598Z",
  "updateBy": "355060a60a5e48eeb7f2fb8d92af2ba5",
  "createBy": "355060a60a5e48eeb7f2fb8d92af2ba5"
}

"status": "Authorized",
"subPayments": [],
"interactions": [
  {
    "gatewayInteractionId": 155627160,
    "paymentId": "d5abdc686f4b4247ae75a95e01271446",
    "currencyCode": "USD",
    "interactionType": "Authorization",
    "status": "Authorized",
    "paymentEntryStatus": "New",
    "isRecurring": false,
    "isManual": false,
    "gatewayTransactionId": "40018563191",
    "gatewayAuthCode": "ASQLVO",
    "gatewayAVSCodes": "Y",
    "gatewayCVV2Codes": "P",
    "gatewayResponseCode": "1",
    "gatewayResponseText": "This transaction has been approved.",
    "gatewayResponseData": [
      {
        "key": "AuthorizationRequestId",
        "value": "5642607567736184003012"
      },
      {
        "key": "AuthorizationRequestToken",
        "value": "Ahj7wSTMelpFwo2FrEnyzVs0ZNmdq2bt2bZi4aMGDNgxZKNP3ock7AVGn70OSd6QnBxEMmkmW6QHez7QjyZjxdlUGxIs4gLxmU"
      },
      {
        "key": "currencyCode",
        "value": "USD"
      }
    ]
  }
],
"interactionDate": "2018-09-17T17:54:21.098Z"
"isRecurring": false,
      "amountCollected": 0,
      "amountCredited": 0,
      "amountRequested": 45
    }
  },
  "isImport": true,
  "amountAvailableForRefund": 75,
  "amountRemainingForPayment": 0,
  "amountRefunded": 0,
  "tenantId": 19523,
  "siteId": 30207,
  "channelCode": "Online",
  "currencyCode": "USD",
  "customerInteractionType": "Unknown",
  "fulfillmentInfo": {
    "fulfillmentContact": {
      "email": "example2@kibo.net",
      "firstName": "Ga",
      "lastNameOrSurname": "Ga",
      "phoneNumbers": {
        "mobile": "1-512-739-1485"
      },
      "address": {
        "address1": "1900 W Girard Ave",
        "cityOrTown": "Sheridan",
        "stateOrProvince": "TX",
        "postalOrZipCode": "78717",
        "countryCode": "US",
        "addressType": "Residential",
        "isValidated": true
      }
    },
    "shippingMethodCode": "fedex_FEDEX_2_DAY",
    "shippingMethodName": "5% of order",
    "auditInfo": {
      "updateDate": "2017-05-24T00:42:02.211Z",
      "createDate": "2017-05-24T00:21:49.617Z",
      "updateBy": "355060a60a5e48eeb7f2fb8d92af2ba5",
      "createBy": "355060a60a5e48eeb7f2fb8d92af2ba5"
    }
  },
  "isOptInForSms": true,
  "orderDiscounts": [],
  "subtotal": 195,
  "discountedSubtotal": 75,
  "discountTotal": 0,
  "discountedTotal": 75,
  "shippingTotal": 0,
  "shippingSubTotal": 0,
  "shippingTaxTotal": 0,
  "handlingTaxTotal": 0,
  "itemTaxTotal": 0,
  "taxTotal": 0,
  "feeTotal": 0,
  "total": 75,
  "lineItemSubtotalWithOrderAdjustments": 75,
  "shippingAmountBeforeDiscountsAndAdjustments": 0,
  "extendedProperties": []
}
Import Completed Orders

Implementations that include eCommerce can import past orders from a previous system that have been completed. Importing these historical orders allows you to maintain their records in KCCP and perform returns, refunds, and credits on them.

Enable Site Setting

Before importing, you may want to enable a site setting to allow making returns and credits on imported completed orders. This will allow the system to create fulfilled shipments for these orders and process returns, refunds, and credits on them.

1. Go to System > Settings > General.
2. Click the Site tab and locate the fulfiller settings section.
3. Toggle on Refund/Credit on imported orders.

Import Data

In the Create Order payload, set the isHistoricalImport flag to "true" to indicate that it is a past order. When enabled, the order can only be set to the Completed or Canceled status. You must also include payment authorization and capture interaction information to support making refunds or credits on this order, the same data as provided in the standard order import.

Shipments will be created in the Fulfilled state and the isHistoricalImport flag will be passed to them, which will not trigger any events such as status change notifications. If you do not include fulfillment location codes in the order item data, KCCP will use the default location as configured in the shipping settings. If there is no provided location code or default location, then an error will be returned and the order will not be imported.

This is a full example request for importing a completed order:

```json
{
    "isTaxExempt": false,
    "email": "example@kibocommerce.com"
}
```
"customerId": 1000,
"ipAddress": "000.11.222.333",
"status": "Completed",
"type": "Offline",
"externalId": "CCPayment",
"isEligibleForReturns": true,
"totalCollected": 0,
"attributes": [],
"shippingDiscounts": [],
"handlingDiscounts": [],
"handlingTotal": 0,
"notes": [],
"items": [
  {
    "id": "eb75a8fa9a354e66ae32ada300c9c5a1",
    "fulfillmentLocationCode": "Loc1",
    "fulfillmentMethod": "Ship",
    "lineId": 1,
    "product": {
      "fulfillmentTypesSupported": [
        "DirectShip",
        "InStorePickup"
      ],
      "options": [],
      "properties": [],
      "categories": [],
      "price": {
        "price": 10
      },
      "discountsRestricted": false,
      "isTaxable": true,
      "productType": "AllUsage",
      "productUsage": "Standard",
      "bundledProducts": [],
      "productCode": "sp_01",
      "name": "sp_01",
      "goodsType": "Physical",
      "isPackagedStandAlone": false,
      "stock": {
        "manageStock": true,
        "isOnBackOrder": false,
        "stockAvailable": 100,
        "aggregateInventory": 0
      },
      "measurements": {
        "height": {
          "unit": "in",
          "value": 1
        },
        "width": {
          "unit": "in",
          "value": 1
        },
        "length": {
          "unit": "in",
          "value": 1
        },
        "weight": {
          "unit": "lbs",
          "value": 1
        }
      }
    }
  }
]
null
"billingContact": {
  "id": 1000,
  "email": "example@kibocommerce.com",
  "firstName": "example",
  "middleNameOrInitial": "",
  "lastNameOrSurname": "customer",
  "companyOrOrganization": "Example Company",
  "phoneNumbers": {
    "home": "5129991111",
    "mobile": "",
    "work": ""
  },
  "address": {
    "address1": "717 N. Harwood St.",
    "address2": "",
    "address3": "",
    "address4": "",
    "cityOrTown": "Dallas",
    "stateOrProvince": "TX",
    "postalOrZipCode": "75201",
    "countryCode": "US",
    "addressType": "Residential",
    "isValidated": false
  }
},
"isSameBillingShippingAddress": false
},
"payments": [
  {
    "paymentType": "CreditCard",
    "paymentWorkflow": "Mozu",
    "billingInfo": {
      "paymentType": "CreditCard",
      "billingContact": {
        "email": "example.customer@email.com",
        "firstName": "Example",
        "middleNameOrInitial": "",
        "lastNameOrSurname": "Customer",
        "phoneNumbers": {
          "home": "1234567895",
          "mobile": "1234567895",
          "work": ""
        },
        "address": {
          "address1": "111 Example Ln",
          "address2": "",
          "address3": "",
          "address4": "",
          "cityOrTown": "Austin",
          "stateOrProvince": "TX",
          "postalOrZipCode": "78758",
          "countryCode": "US",
          "addressType": "Residential",
          "isValidated": false
        }
      },
      "isSameBillingShippingAddress": false,
      "card": {
        "isUsedRecurring": false,
        "billingInfo": {
          "billingContact": {
            "id": 1000,
            "email": "example@kibocommerce.com",
            "firstName": "example",
            "middleNameOrInitial": "",
            "lastNameOrSurname": "customer",
            "companyOrOrganization": "Example Company",
            "phoneNumbers": {
              "home": "5129991111",
              "mobile": "",
              "work": ""
            },
            "address": {
              "address1": "717 N. Harwood St.",
              "address2": "",
              "address3": "",
              "address4": "",
              "cityOrTown": "Dallas",
              "stateOrProvince": "TX",
              "postalOrZipCode": "75201",
              "countryCode": "US",
              "addressType": "Residential",
              "isValidated": false
            }
          },
          "isSameBillingShippingAddress": false
        },
        "isSameBillingShippingAddress": false,
        "card": {
          "isUsedRecurring": false,
          "billingInfo": {
            "billingContact": {
              "id": 1000,
              "email": "example@kibocommerce.com",
              "firstName": "example",
              "middleNameOrInitial": "",
              "lastNameOrSurname": "customer",
              "companyOrOrganization": "Example Company",
              "phoneNumbers": {
                "home": "5129991111",
                "mobile": "",
                "work": ""
              },
              "address": {
                "address1": "717 N. Harwood St.",
                "address2": "",
                "address3": "",
                "address4": "",
                "cityOrTown": "Dallas",
                "stateOrProvince": "TX",
                "postalOrZipCode": "75201",
                "countryCode": "US",
                "addressType": "Residential",
                "isValidated": false
              }
            },
            "isSameBillingShippingAddress": false
          },
          "isSameBillingShippingAddress": false
        }
      }
    }
  }
]
"isUsedRecurring": false,
"nameOnCard": "Example Customer",
"isCardInfoSaved": false,
"paymentOrCardType": "VISA",
"cardNumberPartOrMask": "4111111111111111",
"ccLastFour": "1123",
"isTokenized": true,
"expireMonth": 1,
"expireYear": 2023
},
"auditInfo": {
 "updateDate": "2018-09-17T17:54:21.027Z",
 "createDate": "2018-09-17T17:53:24.598Z",
 "updateBy": "355060a60a5e48eeb7f2fb8d9af2ba5",
 "createBy": "355060a60a5e48eeb7f2fb8d9af2ba5"
}
},
"status": "Collected",
"subPayments": [],
"interactions": [
 {
  "currencyCode": "USD",
  "interactionType": "AuthorizeAndCapture",
  "status": "Captured",
  "paymentEntryStatus": "New",
  "isRecurring": false,
  "isManual": false,
  "gatewayTransactionId": "111222333444",
  "gatewayAuthCode": "12345",
  "gatewayAVSCodes": "Y",
  "gatewayCVV2Codes": "P",
  "gatewayResponseCode": "1",
  "gatewayResponseText": "This transaction has been approved.",
  "gatewayResponseData": [
    {
      "key": "AuthorizationRequestId",
      "value": "ABC123"
    },
    {
      "key": "AuthorizationRequestToken",
      "value": "ABCDEFGHIJK1234567890"
    },
    {
      "key": "currencyCode",
      "value": "USD"
    }
  ],
  "amount": 65.1,
  "interactionDate": "2018-09-17T17:54:21.098Z"
 }
],
"isRecurring": false,
"amountCollected": 25,
"amountCredited": 0,
"amountRequested": 25
],
"refunds": [],
"credits": []
Once imported, you will be able to view the completed orders and shipments in the Admin UI and Fulfiller UI where you can also perform refunds, returns, and credits if enabled. In both interfaces, there will be a message indicating that the order/shipment is a historical import and cannot be edited.

**Order Drafts**

Order drafts let you make incremental modifications to the order without immediately committing those changes. You can use order drafts to make order item adjustments, change item quantities, or apply additional coupons or discounts. You cannot make changes to an order payment, package, shipment, or pickup using order drafts.

Kibo stores the order draft separately from the original order, but the draft maintains a link to the original order. When performing the `GetOrder` operation, you can choose to retrieve the draft or the original using the `draft` parameter.

After either you or a shopper create an order, you can use the following API order operations to create or update an order draft:

- `CreateOrder`
- `UpdateOrder`
- `DeleteOrderDraft`

When you update an order draft, you can perform any of the following actions:
<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apply to draft</td>
<td>This action saves and applies any modifications made to the existing draft. If the draft includes</td>
<td>To perform this action, run the <code>UpdateOrder</code> operation and set the <code>updateMode</code> parameter to</td>
</tr>
<tr>
<td></td>
<td>previously saved modifications, the new modifications are added to the draft.</td>
<td><code>ApplyToDraft</code>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apply to original</td>
<td>This action saves and applies any modifications made to the draft to the original order and deletes</td>
<td>To perform this action, run the <code>UpdateOrder</code> operation and set the <code>updateMode</code> parameter to</td>
</tr>
<tr>
<td></td>
<td>the existing draft. If the draft included any previously saved modifications, these modifications are lost.</td>
<td><code>ApplyToOriginal</code>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apply and commit</td>
<td>This action applies all new and prior modifications associated with the draft to the original order</td>
<td>To perform this action, run the <code>UpdateOrder</code> operation and set the <code>updateMode</code> parameter to</td>
</tr>
<tr>
<td></td>
<td>and deletes the existing draft.</td>
<td><code>ApplyAndCommit</code>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Each time you save an order draft as the original, or apply and commit the order draft, Kibo creates a new version of the order and increments the version number by one. If multiple users make simultaneous modifications to an order draft, Kibo applies the modifications to the draft based on the first user to save or commit the changes.