

Developer Guide

Demos & Integrations

How-to Guides

How-to guides

Get an access token

Customers

Manage customers

Extend customer resources

Products and Catalogs

Extend product resources

# Extend cart and order resources

Extend the order resource to add shipping provider and a consignment number.

## Prerequisites

If you want to follow along, you need the following items:

- A Commerce Cloud account and the Client ID and Client Secret of your store
- Basic understanding of object-oriented programming or JavaScript
- An access token

## Summary of steps required

### Create a custom Flow for shipping providers

- Create shipping provider flow
- Create a name and cost fields

### Extend orders Flow

- Create consignment id field
- Create shipping provider relationship field that links to the orders flow.

### Create shipping providers

- Create entries that will store your shipping providers.

### Extend a cart

- Extend carts flow
- Create shipping provider relationship field that will link to the carts flow

### Create an order

- Get a cart
- Add an item to a cart
- Add a custom item that contains shipping charges to a cart
- Checkout your cart
- Create shipping provider relationship field that links to the orders flow
- Update the order to add shipping details

## Step-by-step walkthrough

The walkthrough shows you how to:

- Create a custom Flow to store shipping providers.
- Extend the existing resource `carts`, so that you can charge your customer for shipping.
- Extend an existing resource `orders`, so it displays shipping provider details and the consignment number for each order charged with a shipping fee.

### Create a custom Flow (shipping provider)

We start with creating a custom Flow that will store our shipping providers.

```
curl -X POST https://api.moltin.com/v2/flows \
-H "Authorization: XXXX" \
-H "Content-Type: application/json" \
-d $'{
  "data": {
    "type": "flow",
    "name": "shipping_provider",
    "slug": "shipping-provider",
    "description": "Creates a new object",
    "enabled": true
  }
}'
```

### Create Fields for Shipping Provider Flow

The Fields will be returned when you call the object you created. Replace `Flow-ID` with the ID that was generated for the Flow (see step 2). In this scenario we will create two fields: `Name` for storing each shipping providers name and `Cost` to store the shipping charges. Code example below shows you how to add the Name field. Repeat the request for the Cost field, changing the details accordingly.

```
curl -X POST https://api.moltin.com/v2/fields \
-H "Authorization: XXXX" \
-H "Content-Type: application/json" \
-d $'{
  "data": {
    "type": "field",
    "name": "Name",
    "slug": "name",
    "field_type": "string",
    "description": "Shipping providers name",
    "required": false,
    "unique": false,
    "default": 0,
    "enabled": true,
    "order": 1,
    "relationships": {
      "flow": {
        "data": {
          "type": "flow",
          "id": "SHIPPING-PROVIDER-FLOW-ID"
        }
      }
    }
  }
}'
```

### Create a core Flow for orders

Create a new core Flow that will extend your orders resource. The `slug` must match the plural name of the resource you're extending, such as `orders`.

NOTE

If you already have a Flow for your existing resource, skip this step. You need the ID of the orders Flow to continue.

```
curl -X POST https://api.moltin.com/v2/flows \
-H "Authorization: XXXX" \
-H "Content-Type: application/json" \
-d $'{
  "data": {
    "type": "flow",
    "name": "orders",
    "slug": "orders",
    "description": "Flow description",
    "enabled": true
  }
}'
```

Take note of the Flow ID that is returned. You need the ID to create a Flow field.

### Create a Consignment ID Field

The Field will be returned when you call the object you extended. Replace `Flow-ID` with the ID that was generated for the Flow. We will create the **Consignment ID** field that will store and display the consignment number for each order. This scenario assumes that every order can only have one shipping provider associated with it. To cater to this requirement, we will need to add a `one-to-one` relationship as part of the validation rules.

```
curl -X POST https://api.moltin.com/v2/fields \
-H "Authorization: XXXX" \
-H "Content-Type: application/json" \
-d $'{
  "data": {
    "type": "field",
    "name": "Consignment ID",
    "slug": "consignment-id",
    "field_type": "string",
    "description": "Field description",
    "required": false,
    "unique": false,
    "default": 0,
    "enabled": true,
    "order": 1,
    "relationships": {
      "flow": {
        "data": {
          "type": "flow",
          "id": "ORDERS-FLOW-ID"
        }
      }
    }
  }
}'
```

### Create a Shipping Provider relationship Field

Now we need to create a relationship between orders and shipping provider Flows. To do this, we need to add a relationship type of a field\_type.

```
curl -X POST https://api.moltin.com/v2/flows \
-H "Authorization: XXXX" \
-H "Content-Type: application/json" \
-d $'{
  "data": {
    "type": "field",
    "name": "Shipping Provider",
    "slug": "shipping-provider",
    "description": "The shipping provider used to send the order",
    "required": false,
    "unique": false,
    "field_type": "relationship",
    "relationships": {
      "flow": {
        "data": {
          "type": "flow",
          "id": "ORDERS-FLOW-ID"
        }
      }
    },
    "validation_rules": [
      {
        "type": "one-to-one",
        "to": "shipping provider"
      }
    ]
  }
}'
```

### Create shipping providers Entries

Entries represent the actual content each field will contain. Create an Entry for every Field added. In this example, we will add Entries to the Shipping Provider Flow Fields we created in step 3. These were Name and Cost, and this is exactly what we'll be adding a name of a shipping provider and a cost of shipping.

```
curl -X POST https://api.moltin.com/v2/flows/flow-slug/entries \
-H "Authorization: XXXX" \
-H "Content-Type: application/json" \
-d $'{
  "data": {
    "type": "entry",
    "name": "Ralphio 720",
    "cost": 7.2
  }
}'
```

### Create a core Flow for carts

Create a new core Flow that will extend your carts resource. Note that slug must match the plural name of the resource you're extending, i.e. `carts`.

If you already have a Flow for your existing resource, skip this step. You need the ID of the carts Flow to continue.

```
curl -X POST https://api.moltin.com/v2/flows \
-H "Authorization: XXXX" \
-H "Content-Type: application/json" \
-d $'{
  "data": {
    "type": "flow",
    "name": "carts",
    "slug": "carts",
    "description": "Flow description",
    "enabled": true
  }
}'
```

### Create a Shipping Provider relationship

Now it's time to create a relationship between the Shipping Provider Flow and the carts Flow, just like we did in step 6 for the orders Flow.

```
curl -X POST https://api.moltin.com/v2/flows \
-H "Authorization: XXXX" \
-H "Content-Type: application/json" \
-d $'{
  "data": {
    "type": "field",
    "name": "Shipping Provider",
    "slug": "shipping-provider",
    "description": "The shipping provider used to send the order",
    "required": false,
    "unique": false,
    "field_type": "relationship",
    "relationships": {
      "flow": {
        "data": {
          "type": "flow",
          "id": "CARTS-FLOW-ID"
        }
      }
    },
    "validation_rules": [
      {
        "type": "one-to-one",
        "to": "shipping provider"
      }
    ]
  }
}'
```

### Create an order

This scenario assumes you already have your store set up with at least products and payment details. The order we will create here will be unpaid, but you typically need to authorize and capture the payment.

Here, we'll just show steps necessary to apply a shipping charge and to display shipping details on the order:

- Get a cart
- Add an item to the cart (you need a product id).
- Add a custom item (shipping charge) to the cart.
- Checkout the cart with the customer object.
- Update the order with the Consignment ID returned by the shipper.
- Get the order with the shipping details.

### Get a cart

Make sure you replace the `:cart-reference` with a value, such as 1234.

```
curl -X GET https://api.moltin.com/v2/carts/:cart-reference \
-H "Authorization: Bearer XXXX"
```

### Add an item to the cart

Here, you need the product id that you have previously added to your store.

```
curl -X POST https://api.moltin.com/v2/carts/:cart-reference/items \
-H "Authorization: Bearer XXXX" \
-H "Content-Type: application/json" \
-d $'{
  "data": {
    "id": "PRODUCT-ID",
    "type": "cart_item",
    "quantity": 1
  }
}'
```

### Add a custom item to the cart

A custom item allows you to add additional charges like a shipping fee. Here, we use one of the shipping provider Flow Entries we created in step 7.

```
curl -X POST https://api.moltin.com/v2/carts/:cart-reference/items \
-H "Authorization: Bearer XXXX" \
-H "Content-Type: application/json" \
-d $'{
  "data": {
    "type": "custom_item",
    "name": "Ralphio 720",
    "slug": "Ralphio-720",
    "description": "Addis charges for shipping",
    "quantity": 1,
    "price": {
      "amount": 720
    }
  }
}'
```

### Checkout the cart with the customer object

The customer object allows for anonymous checkout, i.e. you don't have to have any customer details previously added to your store.

```
curl -X POST https://api.moltin.com/v2/carts/:cart-reference/checkout \
-H "Authorization: Bearer XXXX" \
-H "Content-Type: application/json" \
-d $'{
  "data": {
    "customer": {
      "email": "leslie.knope@gov.com",
      "name": "Leslie Knope"
    },
    "billing_address": {
      "first_name": "Leslie",
      "last_name": "Knope",
      "company_name": "Parks and Recreations Dept.",
      "line_1": "2nd Floor British India House",
      "line_2": "15 Carloli Square",
      "city": "Paamea",
      "postcode": "NE1 6UP",
      "country": "Indiana",
      "country": "US"
    },
    "shipping_address": {
      "first_name": "Leslie",
      "last_name": "Knope",
      "company_name": "Parks and Recreations Dept.",
      "phone_number": "(555) 555-1234",
      "line_1": "2nd Floor British India House",
      "line_2": "15 Carloli Square",
      "city": "Paamea",
      "postcode": "NE1 6UP",
      "country": "Indiana",
      "country": "US",
      "instructions": "Leave with Ron Swanson"
    }
  }
}'
```

At this point the order is created. Make note of the order id sent in the response.

### Update the order with the Consignment ID

Typically, once the order is sent for shipping, the shipping provider returns a consignment id by which you can track the shipment. We have already create a field for that, so all we need to do at this point is to add this information to our order.

```
curl -X PUT https://api.moltin.com/v2/orders/:order-id \
-H "Authorization: Bearer XXXX" \
-H "Content-Type: application/json" \
-d $'{
  "data": {
    "type": "order",
    "shipping_provider": "Ralphio 720",
    "consignment_id": "JRTU720666"
  }
}'
```

### Get the order with shipping details

To review shipping details, all you need to do is to send `GET` for the order you wish to view. The order displays the shipping details.

```
curl -X GET https://api.moltin.com/v2/orders/:order-id \
-H "Authorization: Bearer XXXX"
```

The response that you should get:

```
{
  "data": {
    "type": "order",
    "id": "110674f7-b04e-d483-bd9d-aef790ccaf3a",
    "status": "incomplete",
    "payment": "unpaid",
    "shipping": "unfulfilled",
    "customer": {
      "name": "Leslie Knope",
      "email": "leslie.knope@gov.com"
    },
    "shipping_address": {
      "first_name": "leslie",
      "last_name": "Knope",
      "phone_number": "(555) 555-1234",
      "company_name": "Parks and Recreations Dept.",
      "line_1": "2nd Floor British India House",
      "line_2": "15 Carloli Square",
      "city": "Paamea",
      "postcode": "NE1 6UP",
      "country": "Indiana",
      "country": "US",
      "instructions": ""
    },
    "billing_address": {
      "first_name": "leslie",
      "last_name": "Knope",
      "company_name": "Parks and Recreations Dept.",
      "phone_number": "(555) 555-1234",
      "line_1": "2nd Floor British India House",
      "line_2": "15 Carloli Square",
      "city": "Paamea",
      "postcode": "NE1 6UP",
      "country": "Indiana",
      "country": "US"
    },
    "links": {},
    "meta": {
      "display_price": {
        "with_tax": {
          "amount": 20220,
          "currency": "USD",
          "formatted": "$20,220"
        },
        "without_tax": {
          "amount": 27000,
          "currency": "USD",
          "formatted": "$207.20"
        }
      },
      "tax": {
        "amount": 0,
        "currency": "USD",
        "formatted": "$0.00"
      },
      "timestamps": {
        "created_at": "2019-05-17T10:20:20Z",
        "updated_at": "2019-05-17T10:21:48Z"
      }
    },
    "relationships": {
      "shipping_provider": {
        "data": {
          "type": "shipping_provider",
          "id": "0053ba418-96d2-4a25-ab73-409f2b76558A"
        }
      },
      "items": {
        "data": [
          {
            "type": "item",
            "id": "1505aa28-1350-45ec-bc1f-7f627e7aed85"
          },
          {
            "type": "item",
            "id": "922bb7b0e-cba2-4b72-ab9a-fc28a43d6c9d"
          }
        ]
      },
      "shipping_provider": "Ralphio 720",
      "consignment_id": "JRTU720666"
    }
  }
}
```

### Further reading

- Custom Data (Flows)
- Carts
- Orders
- Checkout workflow
- Custom Data (Flows) API
- Extend a resource with custom data
- The Power of Flows (Blog post)
- Create and extend your own custom data (Video)

PREVIOUS

APPLY PROMOTIONS →

Prerequisites

Summary of steps required

Create a custom Flow for shipping providers

Extend orders Flow

Create shipping providers

Extend a cart

Create an order

Step-by-step walkthrough

Create a custom Flow (Shipping provider)

Create Fields for Shipping Provider Flow

Why Elastic Path

For Telecommunication

For B2B Manufacturing

Read About Our Customers

Our Platform

Commerce Cloud

Commerce for B2B

Commerce for B2C

Reference Experiences

For Developers

What Is Hypermedia API

Modern Architecture

Ecommerce Framework

Documentation Integration

Documentation

Company

About us

Press Releases

In the news

Events

Careers

Terms of use

Privacy Policy

Connect with us

Customer Support

Email

Book a meeting →

North American toll free: +1 800 942 5282

Europe: +44 (0)980 260 5600

Worldwide: +1 604 408 8078

Contact Support Here

Talk to our sales team

Media Inquiries

Copyright © 2020, Elastic Path Software Inc. All rights reserved.

Follow Us:    