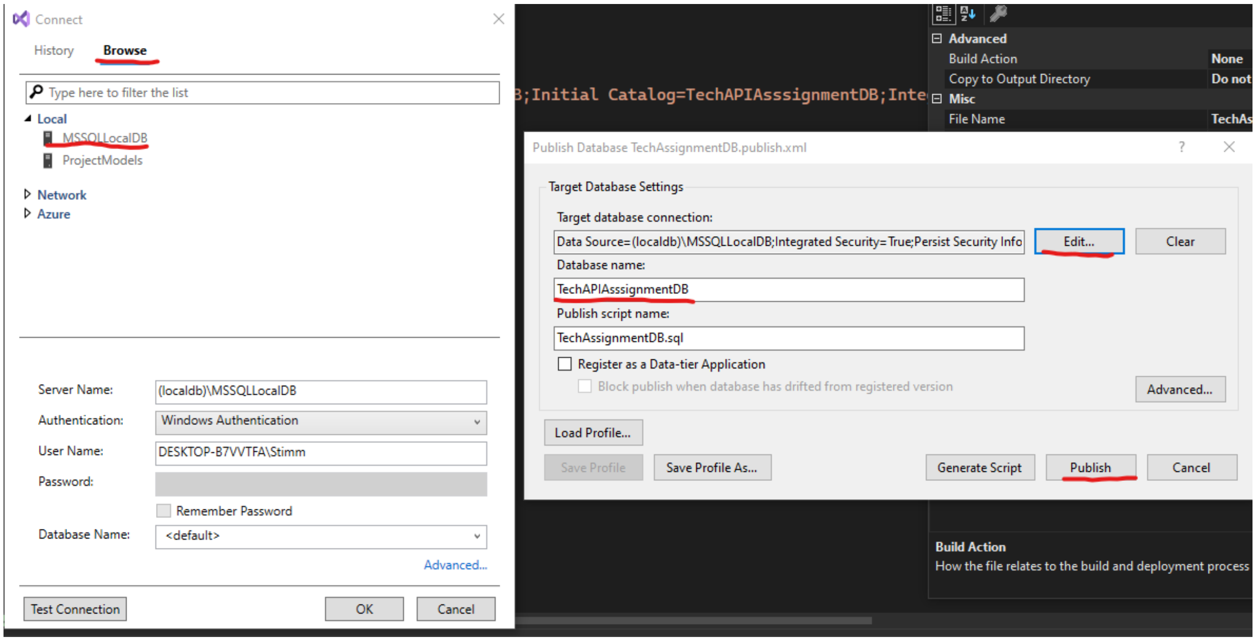
Tech Assessment README

Setup:

To set up the project make sure that the SQL DB is published and that its connection string is in the appsettings.json

Publish the DB:

If the DB is not yet published simply Right Click on the TechAssignmentDB Project -> Publish -> Edit the Target database connection -> Browse -> Local -> MSSQLLocalDB.
Database name: TechAPIAsssignmentDB
Publish.



Server Name: (localdb)\MSSQLLocalDB
Authentication: Windows Authentication
User Name: DESKTOP-B7VVTFA\Stimm
Password:
Database Name: <default>

```
;Initial Catalog=TechAPIAsssignmentDB;Inte
```

Build Action: None
Copy to Output Directory: Do not
File Name: TechAs

Target database connection: Data Source=(localdb)\MSSQLLocalDB;Integrated Security=True;Persist Security Info
Database name: TechAPIAsssignmentDB
Publish script name: TechAssignmentDB.sql

Build Action
How the file relates to the build and deployment process

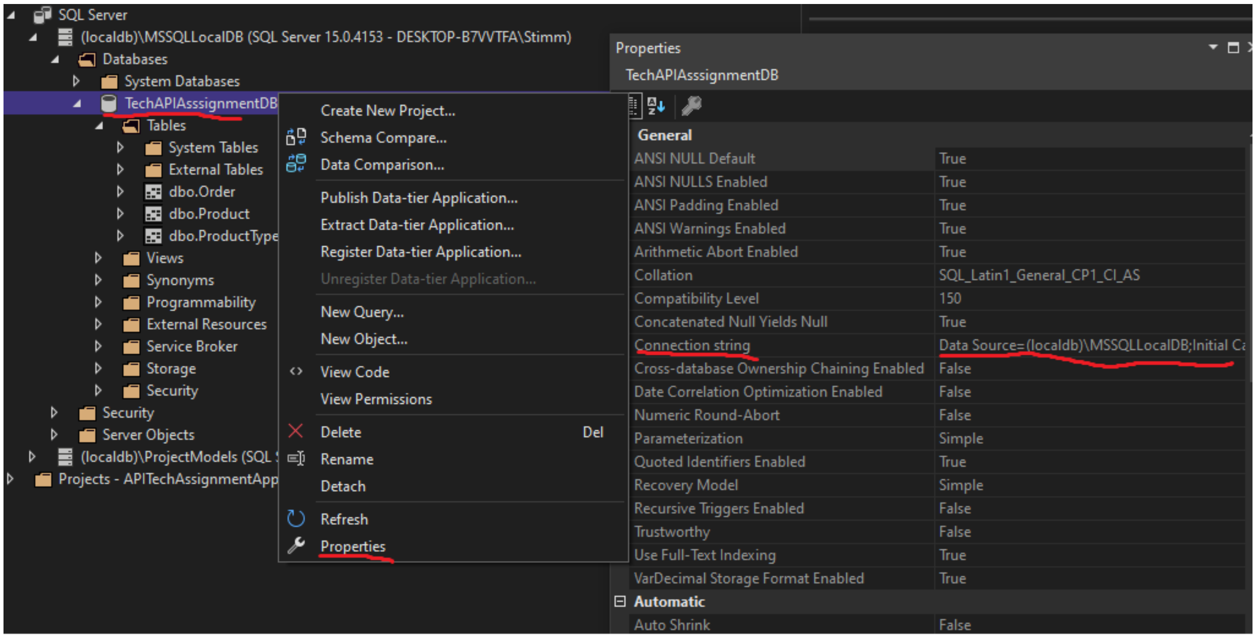
This should then run a post Deployment script that will populate the DB with some basic data.

Set up the connection string:

We need to get the connection string and add it to the appsettings.json file

Get the connection string from SQL Server object Explorer:

View -> SQL Server object Explorer ->(Localdb)MMSQLLocalDB ->Right click TechhAPIAssignment -> connection string

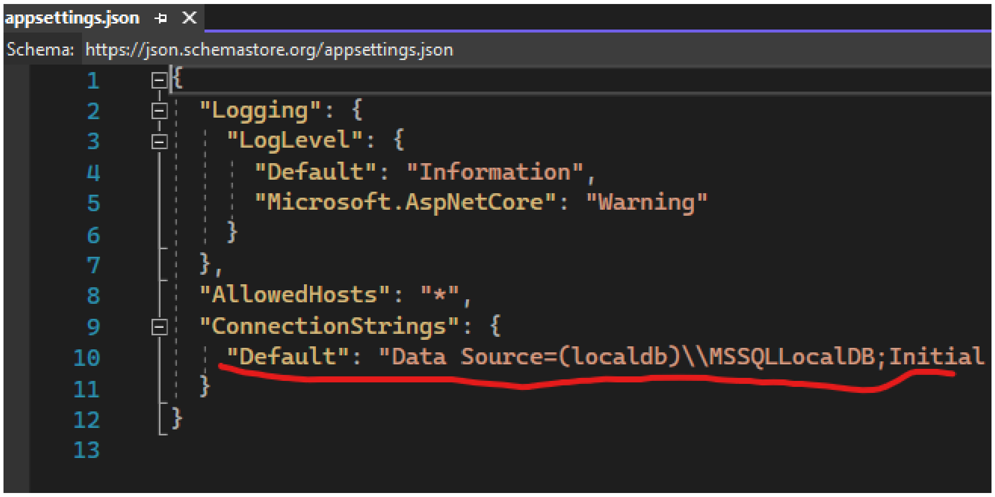


Properties

TechAPIAsssignmentDB

General	
ANSI NULL Default	True
ANSI NULLS Enabled	True
ANSI Padding Enabled	True
ANSI Warnings Enabled	True
Arithmetic Abort Enabled	True
Collation	SQL_Latin1_General_CP1_CI_AS
Compatibility Level	150
Concatenated Null Yields Null	True
Connection string	Data Source=(localdb)\MSSQLLocalDB;Initial Ca
Cross-database Ownership Chaining Enabled	False
Date Correlation Optimization Enabled	False
Numeric Round-Abort	False
Parameterization	Simple
Quoted Identifiers Enabled	True
Recovery Model	Simple
Recursive Triggers Enabled	False
Trustworthy	False
Use Full-Text Indexing	True
VarDecimal Storage Format Enabled	True
Automatic	
Auto Shrink	False

And then add it to the Default connection string:



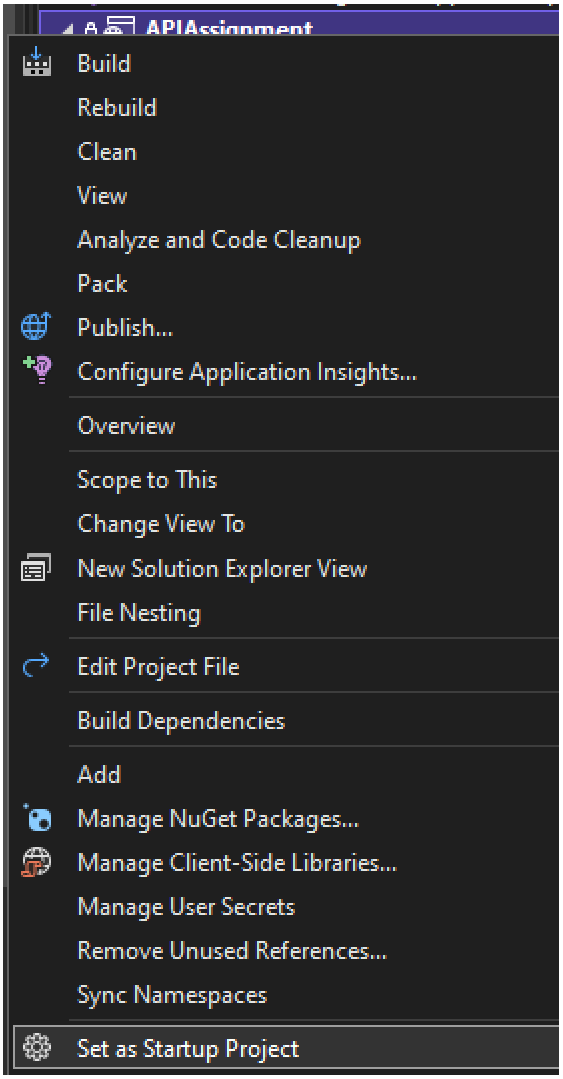
```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "Default": "Data Source=(localdb)\\MSSQLLocalDB;Initial
  }
}
```

Distributed .Net Engineer Tech Assignment

Create a production ready API solution implementing the follow actions:

Launch swagger:

Simply set the APIAssignment project as StartUp Project



And then run the application

Distributed .Net Engineer Tech Assignment

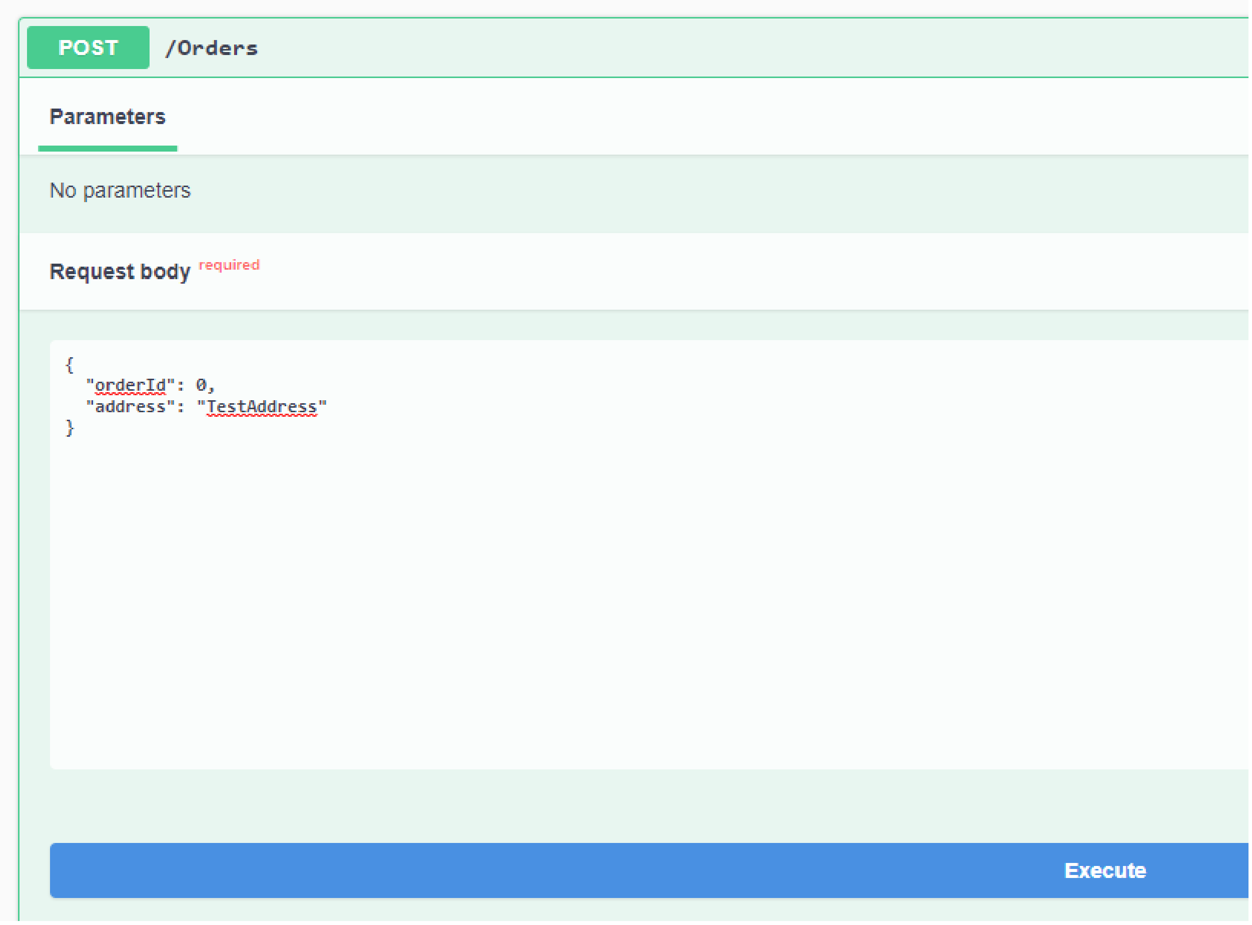
- Create a new order
- Update the order delivery address
- Update the order items
- Cancel an order
- Retrieve a single order
- Retrieve a paginated list of orders

Create a new order

Post/Orders:

Post call to set up new orders.

Leave OrderId as 0 and set the Address to be max 60 characters



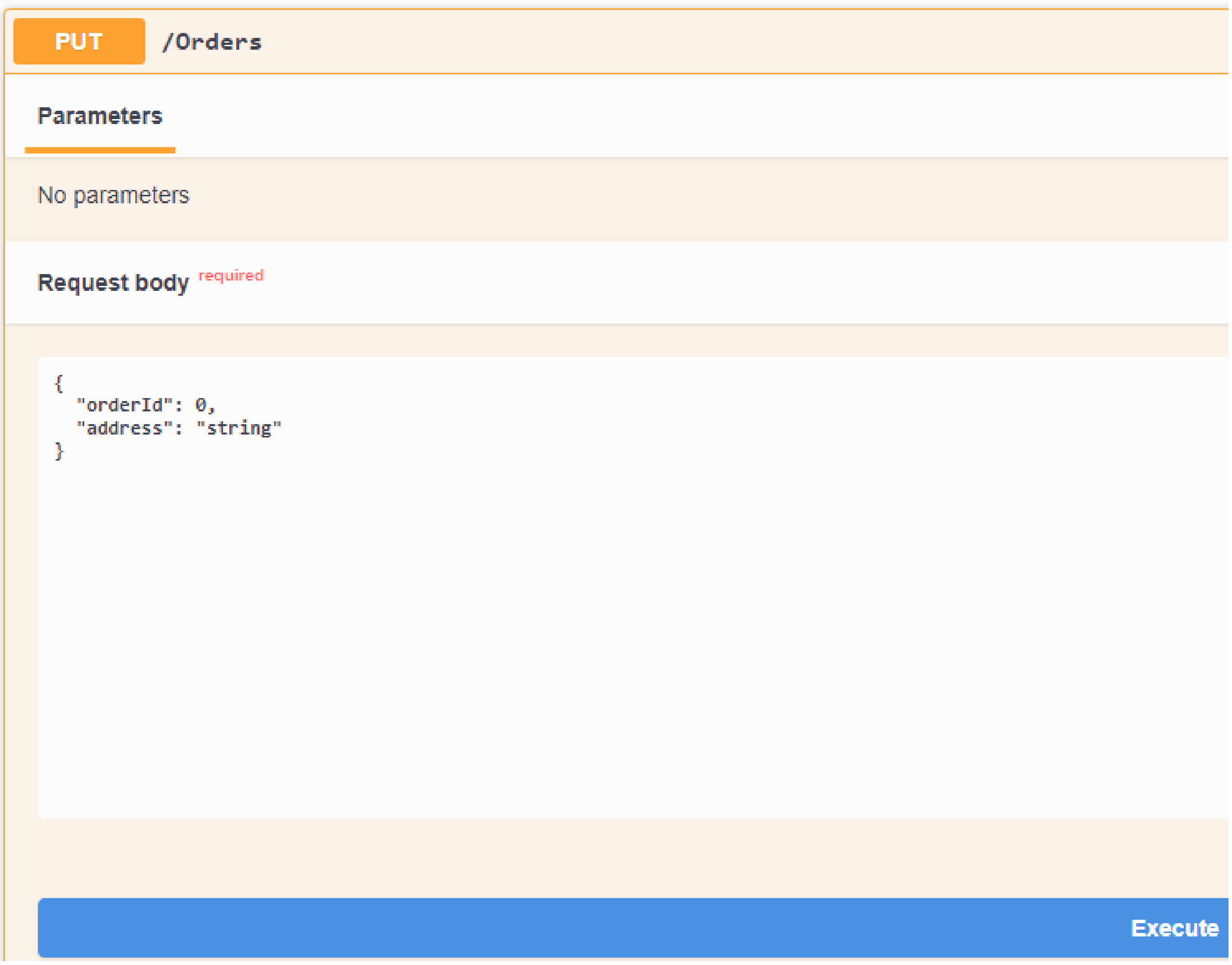
No parameters

```
{
  "orderId": 0,
  "address": "TestAddress"
}
```

Update the order delivery address

Put/Orders

Put call that allows you to change the address of an order
Set OrderID to be the Order you wish to change
Set Address to be the new address.



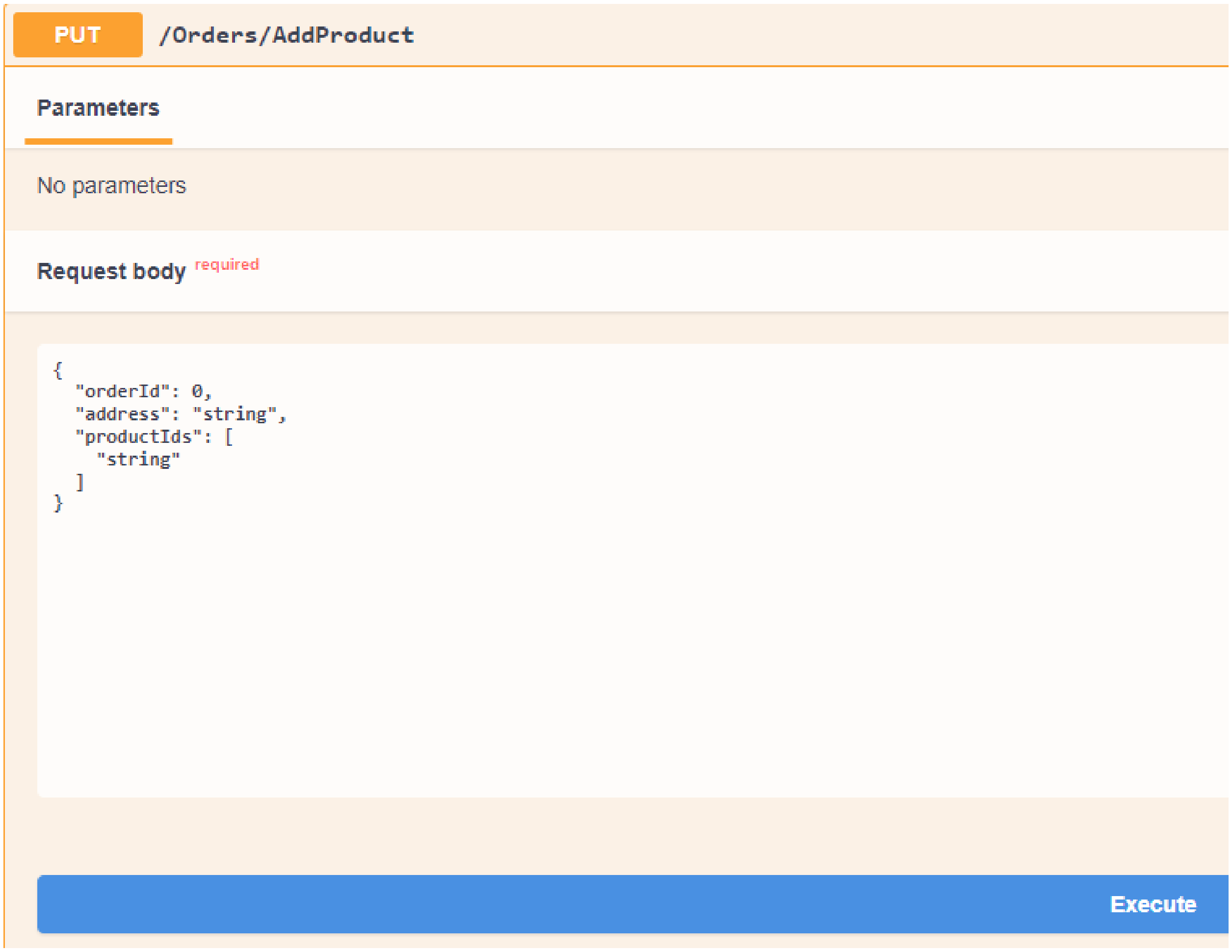
No parameters

```
{
  "orderId": 0,
  "address": "string"
}
```

Update the order items

Put/Orders/AddProducts

Put call that allows you to add a list of ProductID’s to add them to an order.
Set OrderID to be the Order you wish to add the product to
Fill the ProductIDs list with the GUID’s of products you wish to add



No parameters

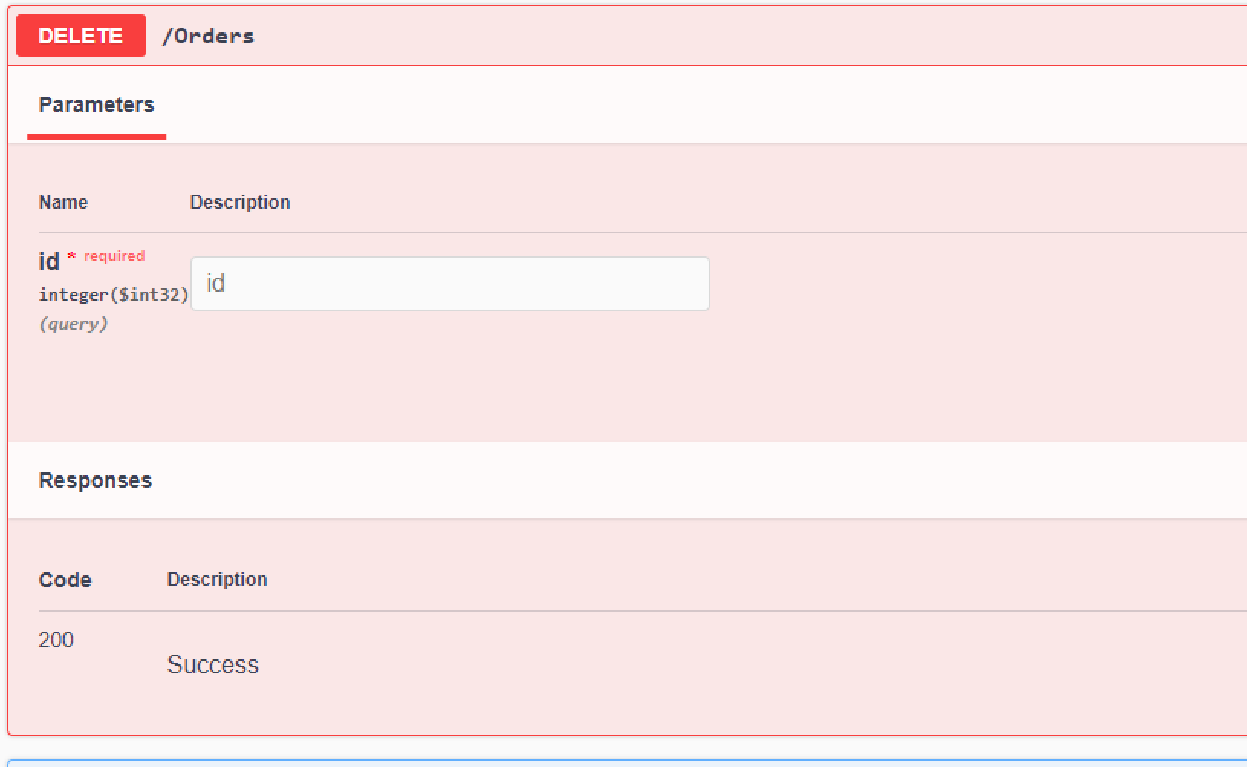
```
{
  "orderId": 0,
  "address": "string",
  "productIds": [
    "string"
  ]
}
```

Cancel an order

Delete/Orders

Delete an Order and makes the tagged products available for reordering

Set the ID field to be the Order you wish to delete



DELETE /Orders

Parameters

Name	Description
id * required integer($int32) *(query)*	id

Responses

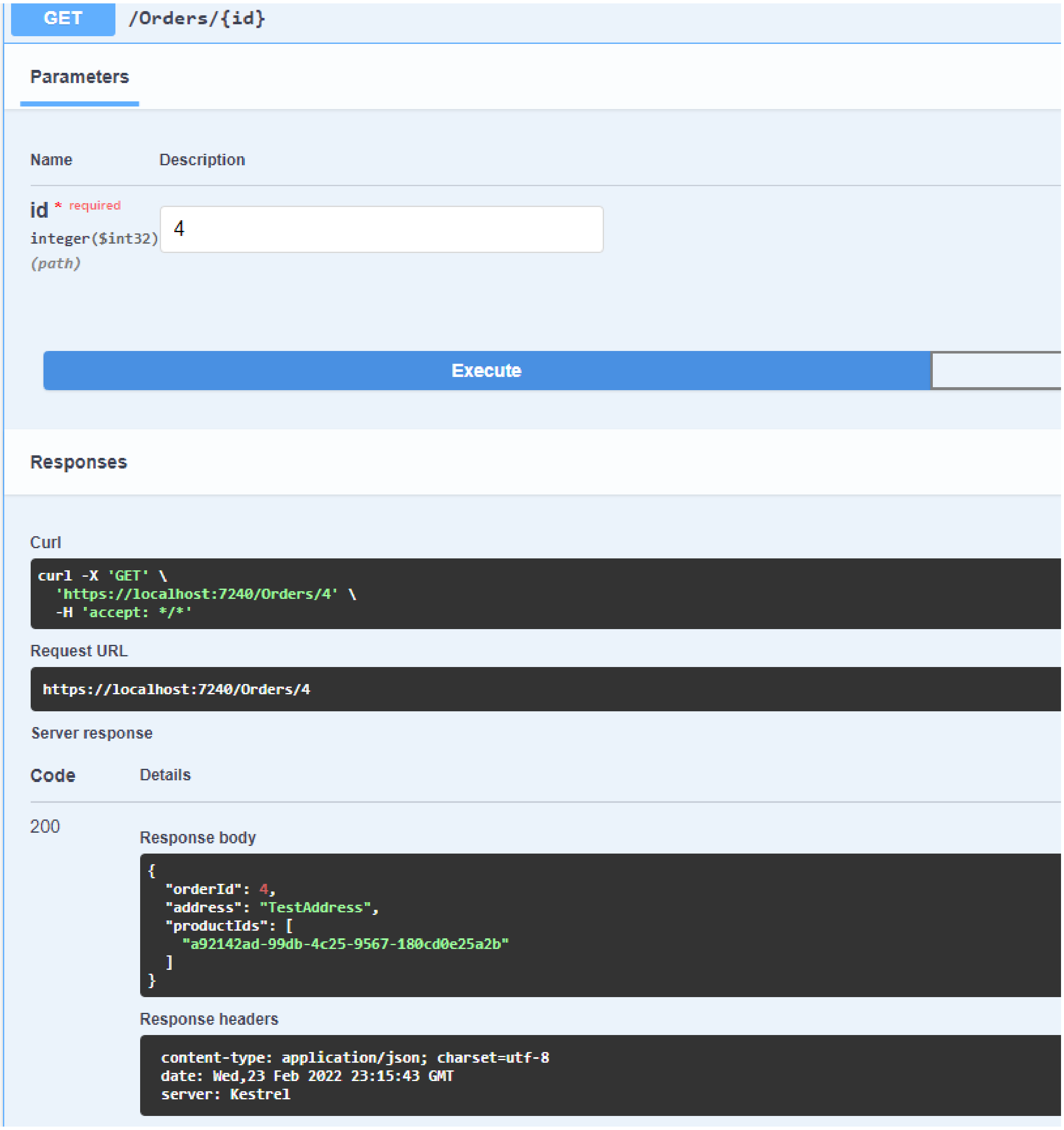
Code	Description
200	Success

Retrieve a single order

Get/Orders/{id}

Returns a Order and its related products

Set Id to be the OrderID that you wish to view



id: 4

Curl

```
curl -X 'GET' \
  'https://localhost:7240/Orders/4' \
  -H 'accept: */*'
```

Request URL

```
https://localhost:7240/Orders/4
```

Server response

Code	Details
200	Response body

```
{
  "orderId": 4,
  "address": "TestAddress",
  "productIds": [
    "a92142ad-99db-4c25-9567-180cd0e25a2b"
  ]
}
```

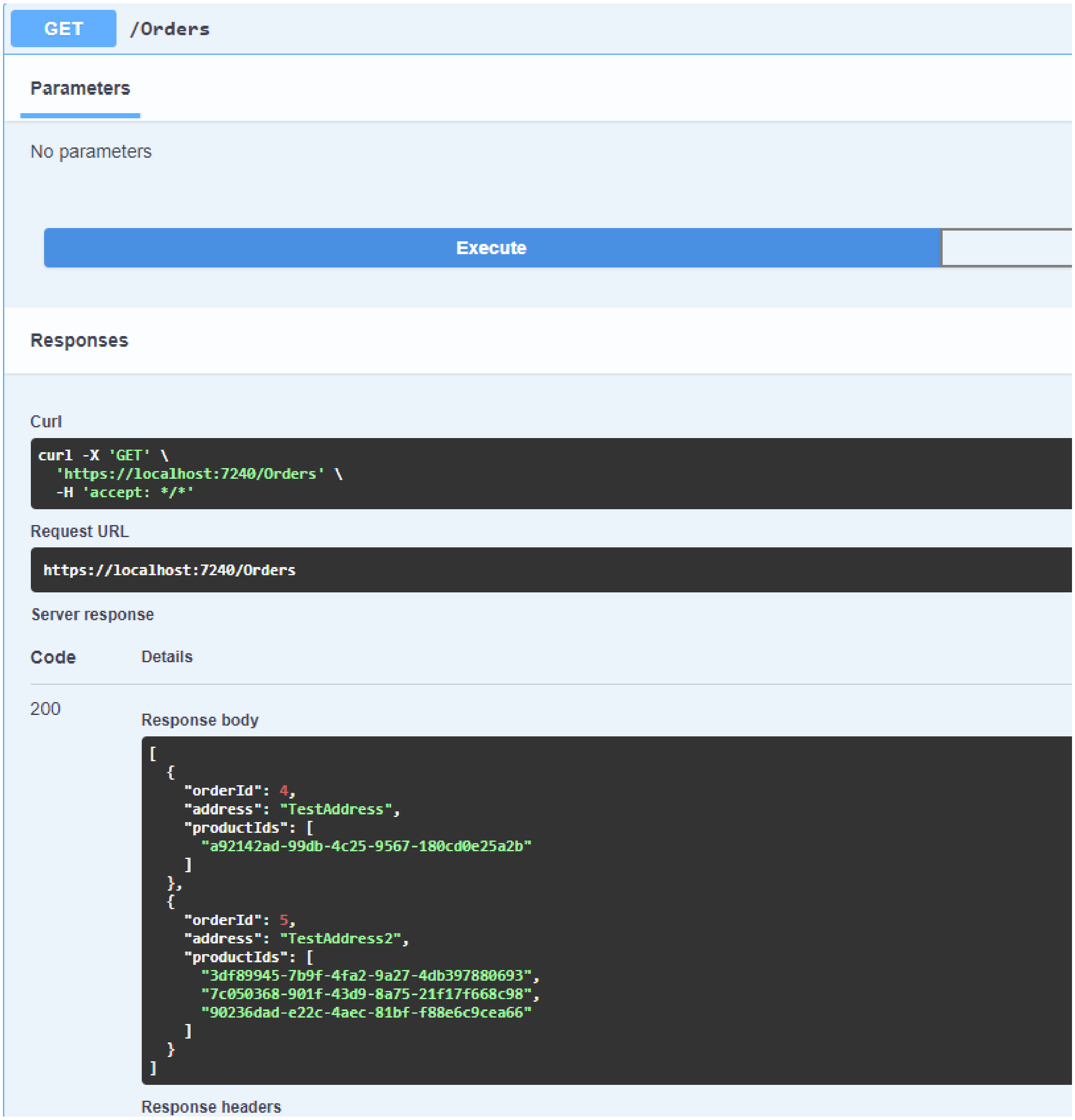
Response headers

```
 content-type: application/json; charset=utf-8
 date: Wed,23 Feb 2022 23:15:43 GMT
 server: Kestrel
```

Retrieve a paginated list of orders

Get/Orders

Returns every Order and is related products



No parameters

```
curl -X 'GET' \
  'https://localhost:7240/Orders' \
  -H 'accept: */*'
```

```
https://localhost:7240/Orders
```

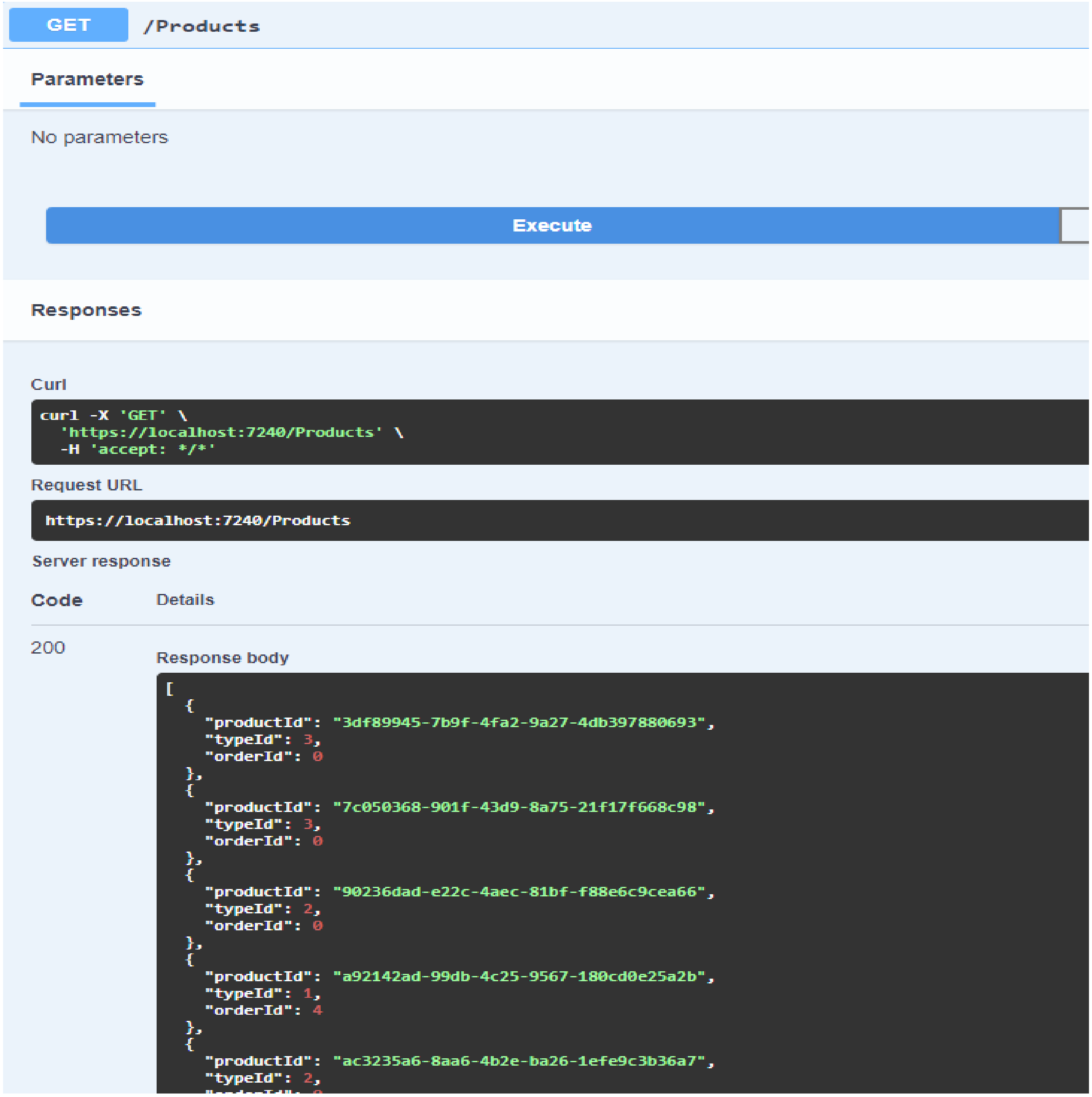
200

```
[
  {
    "orderId": 4,
    "address": "TestAddress",
    "productIds": [
      "a92142ad-99db-4c25-9567-180cd0e25a2b"
    ]
  },
  {
    "orderId": 5,
    "address": "TestAddress2",
    "productIds": [
      "3df89945-7b9f-4fa2-9a27-4db397880693",
      "7c050368-901f-43d9-8a75-21f17f668c98",
      "90236dad-e22c-4aec-81bf-f88e6c9cea66"
    ]
  }
]
```

Useful API calls

Get/Products

Returns all products, their ProductID’s and whether or not the product is currently assigned to a existing Order



Curl

```
curl -X 'GET' \
  'https://localhost:7240/Products' \
  -H 'accept: */*'
```

Request URL

```
https://localhost:7240/Products
```

Server response

Code Details

200 Response body

```
[
  {
    "productId": "3df89945-7b9f-4fa2-9a27-4db397880693",
    "typeId": 3,
    "orderId": 0
  },
  {
    "productId": "7c050368-901f-43d9-8a75-21f17f668c98",
    "typeId": 3,
    "orderId": 0
  },
  {
    "productId": "90236dad-e22c-4aec-81bf-f88e6c9cea66",
    "typeId": 2,
    "orderId": 0
  },
  {
    "productId": "a92142ad-99db-4c25-9567-180cd0e25a2b",
    "typeId": 1,
    "orderId": 4
  },
  {
    "productId": "ac3235a6-8aa6-4b2e-ba26-1efe9c3b36a7",
    "typeId": 2,
    "orderId": 0
```