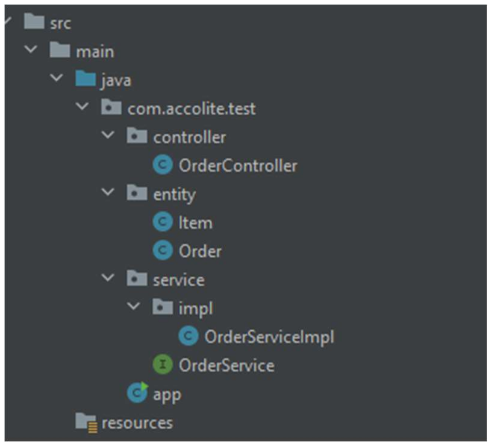
Spring MVC Assignment

By Raj Vignesh Karunakaran

To create an order processing system which follows MVC Architecture

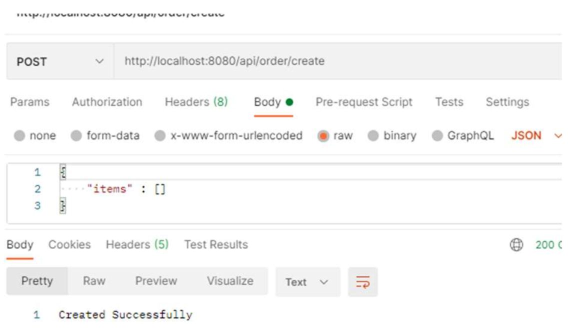
File Structure:



```
src
  main
    java
      com.accolite.test
        controller
          OrderController
        entity
          Item
          Order
        service
          impl
            OrderServiceImpl
          OrderService
        app
    resources
```

Check GitHub for the source code.

1. Create and get Order.
   a. POST method to create an order (OrderId is automatically assigned)



POST http://localhost:8080/api/order/create

Params | Authorization | Headers (8) | Body ● | Pre-request Script | Tests | Settings

none | form-data | x-www-form-urlencoded | [✓ raw] | binary | GraphQL | JSON

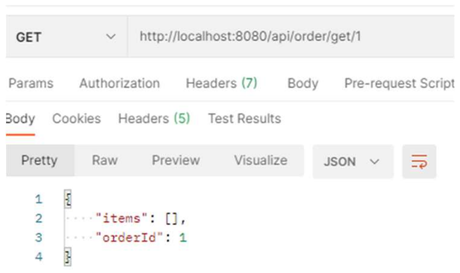
```
{
    "items" : []
}
```

Body | Cookies | Headers (5) | Test Results — 200

Pretty | Raw | Preview | Visualize | Text

```
Created Successfully
```

   b. GET method to get the order by Id.



GET http://localhost:8080/api/order/get/1

Params | Authorization | Headers (7) | Body | Pre-request Script

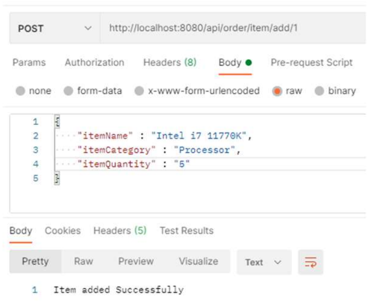
Body | Cookies | Headers (5) | Test Results

Pretty | Raw | Preview | Visualize | JSON

```
{
    "items": [],
    "orderId": 1
}
```

2. Item creation and Manipulation

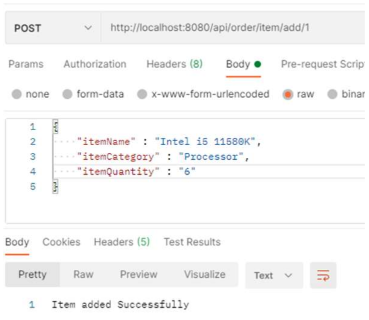
   a. Create and add an item to the order (Throws error if Order id is not found and the ItemId is assigned automatically)



POST http://localhost:8080/api/order/item/add/1

```
{
    "itemName" : "Intel i7 11770K",
    "itemCategory" : "Processor",
    "itemQuantity" : "5"
}
```

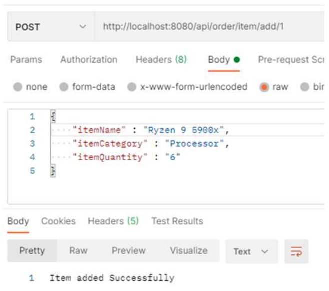
```
Item added Successfully
```



POST http://localhost:8080/api/order/item/add/1

```
{
    "itemName" : "Intel i5 11580K",
    "itemCategory" : "Processor",
    "itemQuantity" : "6"
}
```

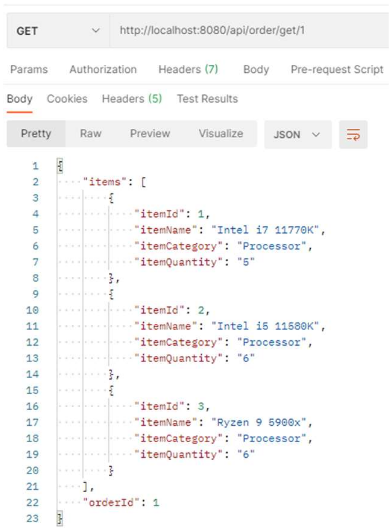
```
Item added Successfully
```



POST http://localhost:8080/api/order/item/add/1

```
{
    "itemName" : "Ryzen 9 5900x",
    "itemCategory" : "Processor",
    "itemQuantity" : "6"
}
```

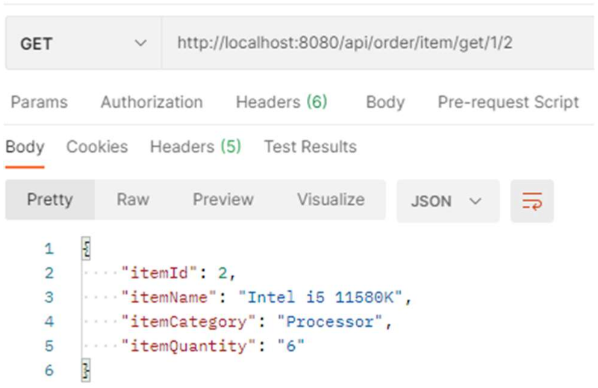
```
Item added Successfully
```



GET http://localhost:8080/api/order/get/1

```
{
    "items": [
        {
            "itemId": 1,
            "itemName": "Intel i7 11770K",
            "itemCategory": "Processor",
            "itemQuantity": "5"
        },
        {
            "itemId": 2,
            "itemName": "Intel i5 11580K",
            "itemCategory": "Processor",
            "itemQuantity": "6"
        },
        {
            "itemId": 3,
            "itemName": "Ryzen 9 5900x",
            "itemCategory": "Processor",
            "itemQuantity": "6"
        }
    ],
    "orderId": 1
}
```

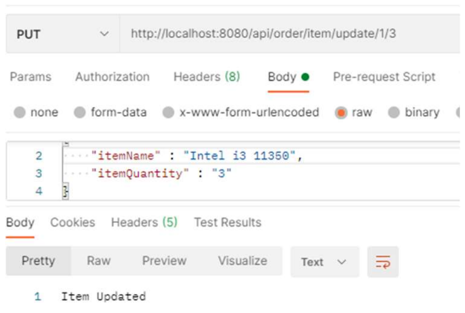
   b. Get item by id from an order (Throws error if the orderId, itemId is not found)



GET http://localhost:8080/api/order/item/get/1/2

```
{
    "itemId": 2,
    "itemName": "Intel i5 11580K",
    "itemCategory": "Processor",
    "itemQuantity": "6"
}
```

c. Update an item in an order (Throws error if the orderId, itemId is not found, Also can update any key at any time. Throws an error if the key is not present in item)

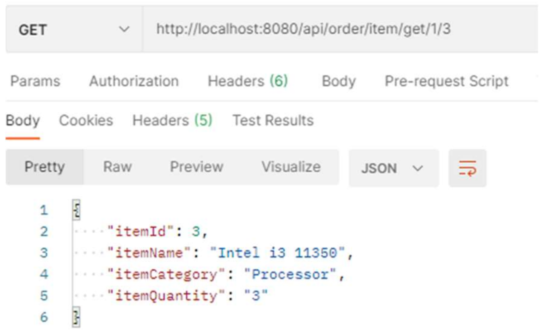


```
PUT http://localhost:8080/api/order/item/update/1/3
```

```
    "itemName" : "Intel i3 11350",
    "itemQuantity" : "3"
}
```

```
Item Updated
```

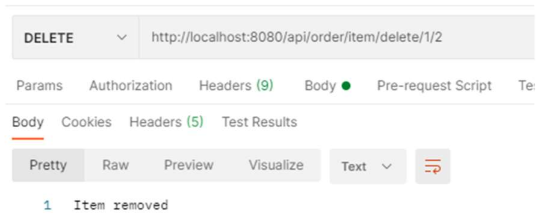
After updating details:



```
GET http://localhost:8080/api/order/item/get/1/3
```

```
{
    "itemId": 3,
    "itemName": "Intel i3 11350",
    "itemCategory": "Processor",
    "itemQuantity": "3"
}
```

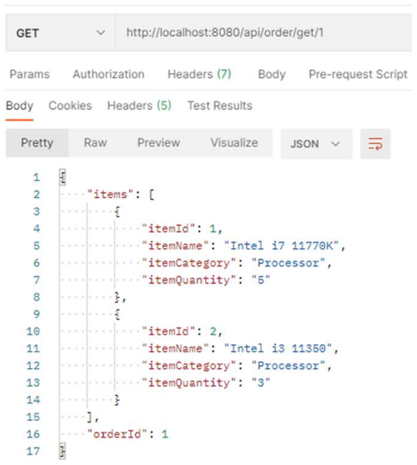
d. Delete an Item using itemId (Throws error if the orderId, itemId is not found, Also updates the ItemId of other elements accordingly)



```
DELETE http://localhost:8080/api/order/item/delete/1/2
```

```
Item removed
```

After Deletion of Item #2:



```
GET http://localhost:8080/api/order/get/1
```

```
{
    "items": [
        {
            "itemId": 1,
            "itemName": "Intel i7 11770K",
            "itemCategory": "Processor",
            "itemQuantity": "5"
        },
        {
            "itemId": 2,
            "itemName": "Intel i3 11350",
            "itemCategory": "Processor",
            "itemQuantity": "3"
        }
    ],
    "orderId": 1
}
```