**Order Management System**

Contents

[**OrderItem Service** 1](#_Toc54636535)

[***1.*** ***Save order item (/orderitem)*** 1](#_Toc54636536)

[***2.*** ***Get all order items(/orderitem)*** 2](#_Toc54636537)

[***3.*** ***Get order items by id(/orderitem/{id})*** 2](#_Toc54636538)

[***4.*** ***Delete order item by id(/orderitem/{id})*** 2](#_Toc54636539)

[**Exception Handled:** 2](#_Toc54636540)

[**Order Service** 2](#_Toc54636541)

[***1.*** ***SaveOrder (/order)*** 2](#_Toc54636542)

[***2.*** ***GetOrder(/order)*** 3](#_Toc54636543)

[***3.*** ***Get order by id (order/{id})*** 3](#_Toc54636544)

[***4.*** ***GetOrderItem by id (orderitembyid/{id})*** 3](#_Toc54636545)

[***5.*** ***Delete order (order/{id})*** 3](#_Toc54636546)

[**Exception Handled:** 3](#_Toc54636547)

This system is a combination of 2 microsrvices, namely: orderitem and orderservice

## **OrderItem Service**

The purpose of the order item service is to create and retrieve the order items from the database. The service is exposed at port *http://localhost:8080*

The service exposes 4 apis to perform the different operations:

1. ***Save order item (/orderitem)***

The api will help the user to save the items in the database. The user will be able to save the name and the quantity of the object (order).

Sample data:

{

    "name": "television",

    "quantity": "4"

}

1. ***Get all order items(/orderitem)***

This is a get call to get all the items saved in the database.

1. ***Get order items by id(/orderitem/{id})***

If the user is aware of the autogenerated id of the item, he can make a search by providing the specific id to fetch the details.

1. ***Delete order item by id(/orderitem/{id})***

This post method will allow the user to delete the items based on the id.

### **Exception Handled:**

In the case if the user tried to fetch or delete an item with an id which is not present, the system will throw the OrderItemNotFoundException and will provide a custom message to the user.

## **Order Service**

The purpose of the order service to create and retrieve orders.

It connects to the orderitem service using the *feignclient* to fetch the order items from the service. The service is exposed at *http://localhost:8081*

The following apis are exposed by the service:

1. ***SaveOrder (/order)***

The api will allow the user to create new order by providing the required details of the order.

Sample data:

{

    "customerName": "Ashish Gupta",

    "orderId": "1",

    "shippingaddress": "Singapore",

    "orderDate": **“**2020-10-26T10:10:10.100+0000**”**,

    "price": "256",

    "orderItems": [

        {

            "name": "television",

            "quantity": "1"

        }

    ]

}

1. ***GetOrder(/order)***

This api will allow the user to fetch all the orders present in the database.

1. ***Get order by id (order/{id})***

If the user wants to find a specific order, he can use this method to fetch the order by the id.

1. ***GetOrderItem by id (orderitembyid/{id})***

The user can also search the order item using this order api call to fetch the data.

1. ***Delete order (order/{id})***

This api will allow the user to delete the order by id.

### **Exception Handled:**

In the case if the user tried to fetch or delete an item with an id which is not present, the system will throw the OrderNotFoundException and will provide a custom message to the user.