E-Commerce Application – Backend

Frameworks used :

- Spring Boot (REST, JPA)
- Hibernate.

Database : MySQL

Entities :

- Customer
- Item
- Order
- Stock

NOTE : In the placeholder {} the value must bbe passed.

REST API – Customer (/customer)

- **Retrieving all Customer :**
  - API : http://localhost:8080/customer
  - METHOD : GET
  - Response Example :

```
[
  {
    "id": 1,
    "name": "SAVK",
    "address": "MyAddress",
    "contactinfo": "1234",
    "ssn": "777"
  },
  {
    "id": 2,
    "name": "Customer2",
    "address": "MyAddres2s",
    "contactinfo": "1234",
    "ssn": "127"
  }
]
```

- Retrieving a specific Customer :
  - API : http://localhost:8080/customer/{id}
  - METHOD : GET
  - Response Example :

```
{
  "id": 1,
  "name": "SAVK",
  "address": "MyAddress",
  "contactinfo": "1234",
  "ssn": "777"
}
```

- Creating a new Customer :
    - API : http://localhost:8080/customer
    - METHOD : POST
    - Request Body :

```
{
        "name": "SAVK",
        "address": "MyAddress",
        "contactinfo": "1234",
        "ssn": "777"
}
```

- Updating a Customer ::
    - API : http://localhost:8080/customer
    - METHOD : PUT
    - Request Body :

```
{
        "id": 1,
         "name": "SAVK-Edit",
        "address": "MyAddress",
        "contactinfo": "1234",
        "ssn": "777"
}
```

- Deleting a Customer :
    - API : http://localhost:8080/customer/{id}
    - METHOD : DELETE

REST API – Items (/item)

- **Retrieving all Items :**
    - API : http://localhost:8080/item
    - METHOD : GET
    - Response Example :

```
[
  {
    "id": 1,
    "name": "SAVK-1-djksnv",
    "description": "SAVK-1-Edited",
    "price": 1234
  },
  {
    "id": 2,
    "name": "New Item 2",
    "description": "Desc 2",
    "price": 50
  }
]
```

- Retrieving a specific Item :
    - API : http://localhost:8080/item/{id}
    - METHOD : GET
    - Response Example :

      ```
      {
        "id": 2,
        "name": "New Item 2",
        "description": "Desc 2",
        "price": 50
      }
      ```

- Creating a Item :
    - API : http://localhost:8080/item
    - METHOD : POST
    - Request Body :

      ```
      {
        "name": "New Item",
        "description": "Description goes here",
        "price": 123
      }
      ```

- Deleting a Item :
    - API : http://localhost:8080/item/{id}
    - METHOD : DELETE

- Updating an Item :
    - API : http://localhost:8080/item
    - METHOD : PUT
    - Request Body :

    ```
    {
      "id": 2,
      "name": "New Item - Name edit",
      "description": "Desc 2 - edited",
      "price": 50
    }
    ```

- Getting stock detail for an Item :
    - API : http://localhost:8080/item/{itemId}/getStock
    - METHOD : GET
    - Response Body :

    ```
    {
      "id": 1,
      "item": {
        "id": 1,
        "name": "SAVK-1-djksnv",
        "description": "SAVK-1-Edited",
        "price": 1234
      },
      "available": 0
    ```

```
}
```

- Updating Stock for an Item :
    - API : http://localhost:8080/item/1/updateStock?value=100
    - METHOD : PUT

REST API – Order

- **Retrieving all Order :**
    - API : http://localhost:8080/order
    - METHOD : GET
    - Response Example :

      ```
      [
        {
          "id": 1,
          "datetime": "2019-05-15T09:04:30.000+0000",
          "paymentmode": "",
          "total": 0,
          "orderEntries": [],
          "incart": true,
          "customer": {
            "id": 1,
            "name": "SAVK",
            "address": "MyAddress",
            "contactinfo": "1234",
            "ssn": "777"
          }
        }
      ]
      ```

- Getting Order Details :
    - API : http://localhost:8080/order/{orderId}
    - METHOD : GET
    - Response Example :

      ```
      {
        "id": 1,
        "datetime": "2019-05-15T09:04:30.000+0000",
        "paymentmode": "",
        "total": 0,
        "orderEntries": [],
        "incart": true,
        "customer": {
          "id": 1,
          "name": "SAVK",
          "address": "MyAddress",
          "contactinfo": "1234",
      ```

```
        "ssn": "777"
    }
}
```

- Creating a new order for a Customer :
    - API : http://localhost:8080/order/createOrderForCustomer/{customerId}
    - METHOD : POST
    - Response Example :

```
{
    "id": 1,
    "datetime": "2019-05-15T09:04:30.000+0000",
    "paymentmode": "",
    "total": 0,
    "orderEntries": [],
    "incart": true,
    "customer": {
        "id": 1,
        "name": "SAVK",
        "address": "MyAddress",
        "contactinfo": "1234",
        "ssn": "777"
    }
}
```

- Adding Items to an Order :
    - API : http://localhost:8080/order/1/addItemsToTheOrder?itemId={itemId}&quantity={qty}&customerId={custId}
    - METHOD : PUT
    - Response :

```
{
    "id": 1,
    "datetime": "2019-05-15T09:04:30.000+0000",
    "paymentmode": "",
    "total": 0,
    "orderEntries": [
        {
            "id": 5,
            "order": null,
            "item": {
                "id": 1,
                "name": "SAVK-1-djksnv",
                "description": "SAVK-1-Edited",
                "price": 1234
            },
            "quantity": 10
        },
        {
            "id": 6,
            "order": null,
            "item": {
                "id": 2,
        "name": "New Item - Name edit",
        "description": "Desc 2 - edited",
        "price": 50
      },
      "quantity": 10
    }
  ],
  "incart": true,
  "customer": {
    "id": 1,
    "name": "SAVK",
    "address": "MyAddress",
    "contactinfo": "1234",
    "ssn": "777"
  }
}
```

- Checkout Order :
  - API : http://localhost:8080/order/{orderId}/checkout? paymentMode=online
  - METHOD : PUT
  - **If there is out of stock then this error response is thrown :**

```
{
  "timestamp": "2019-05-15T10:41:44.142+0000",
  "status": 500,
  "error": "Internal Server Error",
  "message": "Out of stock for Item :: SAVK-1-djksnv",
  "path": "/order/1/checkout"
}
```