David Svistea, Maria Tentea

**Coffeeshop Application: USER MANUAL**

**Table of Contents:**

1 Customer

2 Employee

3 Manager

4 Location

5 Event

6 Product

7 Order

**Introduction:**

**Setup:**

1. Download/Clone the the project from GitHUB

2. Create a new database in MySql and create all tables by running all queries from src/tablequeries.sql

3. Adjust the connection properties from JDBC database connection in src/main/resources/application.properties

4. Run the application

**Interaction with the application:**

The application is REST-Service based and you interact by sending httpqueries with Postman or directly in IntelliJ IDEA ULTIMATE (thanks to its build in http query support)

For examples see in src/main/java/map/project/CoffeeShop/demorequests.http

**CRUD-Operations**

**Create:**

In order to Create a new Entity you have to type

POST localhost:8080/ENTITYNAME/create

And you must pass a JSON object with All its attributes, ID is not important as it is controlled by the APP.

POST localhost:8080/customer/create  
*Content-Type*: application/json  
  
{  
 "id": 133,  
 "firstName": "Bob",  
 "lastName": "Customer",  
 "address": "CustomerAddress 123 Customercity 22333"  
}

If you have an entity with other entities as attributes, you only have to specify the Id of the entity it is related to.

POST localhost:8080/order/create  
Content-Type: application/json  
  
{  
 "id": 1,  
 "location": {  
 "id": 1  
 },  
  
 "customer": {  
 "id": 1  
 },  
 "date\_time": "2024.01.2:06:13"  
}

**Read:**

By default, every entity has a readById and readAll option.

GET localhost:8080/order/4

GET localhost:8080/order/all

The general syntax should speak for itself.

**Update:**

Is done by using a concatenation of Delete and Create. But some entities have special methods to manipulate their attributes.

**Delete:**

DELETE localhost:8080/order/4

**IMPORTANT:** By deleting specific entities, it is possible that other entities are going to be deleted as well.

**1.Customer**

Attributes:

int id;  
  
String firstName;  
  
String lastName;  
  
String address;

**2.Employee**

Attributes:

int id;

Location location;  
  
String firstName;

String lastName;

String address;

float salary;

String title;

**3.Manager**

Attributes:

int id;  
  
Location location;  
  
String firstName;

String lastName;

String address;

float salary;

**4.Location**

Attributes:

int id;  
  
Manager manager;  
  
String name;  
  
String address;  
  
boolean active;  
  
List<Employee> employees;

List<Order> orders;  
  
List<Event> events;  
  
List<LocationProduct> locationProducts;

Operations:

Add Employee:

PUT localhost:8080/location/addEmployee/{locationId}/{employeeId}

Remove Employee:

PUT localhost:8080/location/removeEmployee/{locationId}/{employeeId}

Set Manager:

PUT localhost:8080/location/setManager/{locationId}/{managerId}

Remove Manager:

PUT localhost:8080/location/removeManager/{locationId}/{managerId}

Close Location:

PUT localhost:8080/location/closeLocation/{locationId}

Add Product to Location Stock:

PUT localhost:8080/location/addProductToStock  
Content-Type: application/json  
  
{  
 "location": {  
 "id": LOCATIONID  
 },  
  
 "product": {  
 "id": PRODUCTID  
 },  
 "quantity": 40  
}

Remove Product from Location Stock:

POST localhost:8080/location/removeProductFromStock/{locationId}/{productId}/{quantity}

Show all available Products at Location:

GET localhost:8080/location/allAvailableProducts/{locationId}

**5.Event**

Attributes:

int id;  
  
Location location;  
  
String name;

String host;

float profit;

Set Location:

PUT localhost:8080/event/setLocation/{eventId}/{locationId

**6.Product**

Attributes:

int id;  
  
String name;  
  
float price;  
  
int size;  
  
String unit;

Show all Food/Drinks:

GET localhost:8080/products/food/all

GET localhost:8080/products/drinks/all

Sort Food/Drinks by Price ASC/DESC:

GET localhost:8080/products/food/sortByPrice/asc

GET localhost:8080/products/food/sortByPrice/desc

GET localhost:8080/products/drinks/sortByPrice/asc

GET localhost:8080/products/drinks/sortByPrice/desc

**7.Order**

Attributes:

int id;  
  
String date\_time;  
  
Location location;  
  
Customer customer;

List<OrderProduct> orderProducts;

Add Products to Order:

PUT localhost:8080/order/addProduct/{orderId}/{productId}/{count}

**IMPORTANT:** In order to be able to add products to your order, there have to be enough stock at the location the order is associated to.

Calculate total Price:

GET localhost:8080/order/getTotalPrice/ORDERID