**Grocery Shop Till Documentation**

**Overview**

The Grocery Shop Till project is a Java-based application made to simulate the operations of a grocery store. It includes functionalities such as adding, deleting and retrieving items and special deals, and scanning items for purchase.

**Endpoints**

**Get All Products:**

**URL:** http://localhost:8080/products

**Method:** GET

**Description:** Retrieve all grocery products in the store.

**Add Product:**

**URL:** http://localhost:8080/products

**Method:** POST

**Description:** Add a new grocery product to the store.

**Delete Product:**

**URL:** http://localhost:8080/products

**Method:** DELETE

**Description:** Delete grocery product from the store.

**Get All Deals:**

**URL**: http://localhost:8080/deals

**Method:** GET

**Description:** Retrieve all special deals available.

**Add Deal:**

**URL:** http://localhost:8080/deals

**Method:** POST

**Description:** Add a new special deal.

**Delete Deal:**

**URL:** http://localhost:8080/deals

**Method:** DELETE

**Description:** Delete special deal.

**Get Basket:**

**URL:** http://localhost:8080/baskets/{id}

**Method:** GET

**Description:** Retrieve the basket’s products.

**Scan Items:**

**URL:** http://localhost:8080/baskets/{id}/scan

**Method:** POST

**Description:** Scan a list of products and calculate the total cost.

**Create Basket:**

**URL:** http://localhost:8080/baskets

**Method:** POST

**Description:** Create a new basket.

**Clear Basket:**

**URL:** http://localhost:8080/baskets/{id}/clear

**Method:** DELETE

**Description:** Removes all items from the basket after checkout.

**UI**

The project includes a basic user interface for interacting with the above endpoints. It consists of index.html page, along with script.css forming the style of the page and script.js file which serve as the client-side responsible for handling user interactions, making requests to the server, handling the response and DOM manipulating.

**Folder Structure**

The structure consists of several packages divided by responsibilities.

-**controller/ :** Contains the controller classes responsible for handling HTTP requests.

-**dto/ :** Contains class which maps deal type to list of products as well as retrieving basket’s Id for data manipulation.

-**model/:** Contains entity classes representing the data structure of the application.

- **repository/:** Contains interfaces for CRUD operations, sorting etc.

**-service/:** Contains service class that implement the business logic of the application.

**-src/main/resources/:** This directory contains all non-Java resources like the index.html, style.css and script.js files

**Dependencies**

-MySQL

-Spring Boot

-Maven

-Lombok