Documentation

Developed by Iderkhangai Amarkhuu (iderhangai2014@gmail.com)

Description:

Design and implement a back-end Java application that simulates a vending machine: the user provides the amount of change and the available coins in the vending machine. Then the vending machine gives back the change. The number of the coins of the change should be as few as possible. If there are not enough coins in the vending machine, error should be given back instead. The valid coins are: 5, 10, 20, 50, 100 and 200.
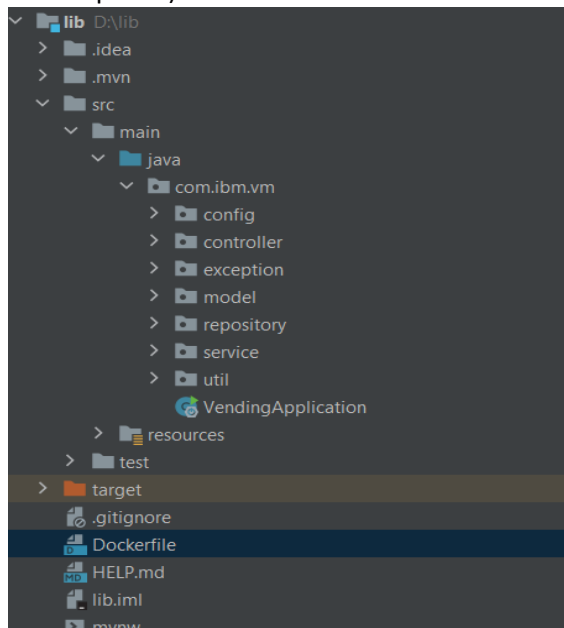
Technologies used

- Spring Boot – Back-End
- ReactJS –Front-End (Material UI)

Architecture:

Spring Boot (Java 11) :

- **Controller** – Main rest controller resides here
- **Exception** – Global exception handler which handles errors.
- **Model** – Service parameters (request and response)
- **Util** – Helper classes
- **Service** – Service that does main functionality
- **Config** – Configuration class (CORS handling)

File structure

ReactJS

- **App.js** – Main functionality that calls API and render UI elements.
- **App.css** – Styles

API (Spring Boot) – `mvn install (port:8080)`
URL: http://localhost:8080/api/v1/vm

Request:

```json
{
    "amount": "500"
}
```

Successful response:

```json
{
    "code": 0,
    "timestamp": "2021-05-17 12:28:21",
    "changes": [
        200,
        200,
        100
    ]
}
```

Request 2:

```json
{
    "amount": ""
}
----------------------------------------------------------------------------
{
    "amount": "11"
}
```

Error Response:

```json
{
    "code": 422,
    "message": "Validation error. Check 'errors' field and Try again.",
    "timestamp": "2021-05-17 12:35:41",
```

```
    "errors": [
        {
            "field": "amount",
            "message": "must be greater than or equal to 10"
        }
    ]
}
------------------------------------------------------------------------------------
{
    "code": 500,
    "message": "Invalid Coin Please Enter Valid Coin!",
    "timestamp": "2021-05-17 12:36:03"
}
```

Front-End (ReactJS)
To run the file: `npm start` (port:3000)

Insert a one of the following coins in the input then submit

💰 Coin Machine

The Following Coins are available in your machine

(5) (10) (20) (50) (100) (200)

Please enter the amount

[Enter your amount]

SUBMIT

Server returns possible minimum number of available coins in the machine:

🛍️ Coin Machine

Please enter the amount

Enter your amount
580

SUBMIT

(200) (200) (100) (50) (20) (10)

Testing with invalid coins

🛍️ Coin Machine

⚠️ Invalid Coin Please Enter Valid Coin!

Please enter the amount

Enter your amount
11

SUBMIT

Creating docker image for the project

```
FROM java:11
EXPOSE 8080
ADD target/ibm-vm.jar ibm-vm.jar
ENTRYPOINT ["java", "-jar", "ibm-vm.jar"]
```