Sabancı University

CS310 Mobile Application Development

2023 Fall Term - Project Backend Report

"CashSwing"

Berke Ayyıldızlı	Beyza Balota	Damla Aydın	Dila Karataş
31018	31232	30825	28852

Summary of Backend:

CashSwing's backend was created using Spring Tool Suite. The backend is mainly based on converting currencies which are Turkish Lira, Euro, US Dollar, Great British Pound, and Japanese Yen. Four classes are created:

Currency.java class is part of the 'com.example' package. It imports necessary classes from both MongoDB and Spring Data. Id, mainCurrency, currencyImage, exchangeRate, max and min are the respected parameters of a Currency object. Getters and setters are also implemented in this file.

CurrencyExchangeController.java imports necessary packages from Spring Framework. Three methods were used in this class; two are a type of GET, and the other is a type of POST. Respected exceptions and results are also implemented inside of the function bodies.

CurrencyRepository.java imports the MongoDB repository, and the interface extends that repository. Since the findByMainCurrency( ) and findAll( ) methods are unfamiliar to Spring, the methods are declared.

ProjBackendVol3Application.java is the main class of CashSwing's backend. It imports the Spring Boot Application for running purposes.

Moreover, in order to Spring to access our database, application.properties file was created, and it contains these lines:

spring.data.mongodb.uri=mongodb://admin:pass@localhost:27017

spring.data.mongodb.database=currency_converter

API's and Endpoints:

1 - ***Get a list of all the currencies on the database - "/getAllCurrencies"***

Description: Lists all the currencies, id, mainCurrency(name), currencyImage(image name), exchangeRate(rate), max(maximum value in 5 years) and min (minimum value in 5 years).

Address: http://localhost:8080/api/exchangerates/getAllCurrencies

Method: GET

Returns: JSON Array:


```
[
  {
    "id": 1,
    "mainCurrency": "TRY/USD",
    "exchangeRate": 0.035,
    "currencyImage": "TRY_USD.jpeg",
    "max": 0.19,
    "min": 0.03
  },
  {
    "id": 2,
    "mainCurrency": "USD/TRY",
    "exchangeRate": 28.93,
    "currencyImage": "USD_TRY.jpeg",
    "max": 28.92,
    "min": 5.30
  },
…
…
…
```

2 - ***Save a new currency to the MongDB database - "/save"***

Description: Takes the details of the new currency in the JSON format, i.e., all the respective fields with the corresponding data types. If the insertion is successful, it throws the “Currency saved successfully!” prompt; if not, it throws "Failed to save currency: xxxxxx ".

Address: http://localhost:8080/api/exchangerates/save

Method: POST

Accepts: JSON file:

```
{
   "id": 99,
   "mainCurrency": "Test/AnotherTest",
   "exchangeRate": 99,
   "currencyImage": "TEST_ANOTHERTEST.jpeg",
   "max": 0.19,
   "min": 0.03
 }
```

Returns: String:

“Currency saved successfully!” or "Failed to save currency: Test/AnotherTest "

3 - ***Get the wanted currency by its name - "/get/{mainCurrency}"***

Description: Takes the wanted currency as a parameter to the query, and returns the respected values of the currency. If the value is indeed in the dataset, it prints the respected currency, if not, it throws "No currency found for {mainCurrency}".

Address: http://localhost:8080/api/exchangerates/get/{mainCurrency}

Method: GET

Returns: JSON Array or String:


```
{
   "id": 999,
   "mainCurrency": "test/anotherTestTest",
   "exchangeRate": 999,
   "currencyImage": "TEST_ANOTHERTESTTEST.jpeg",
   "max": 0.99,
   "min": 0.33
 }
```


Or,

“No currency found for test/anotherTestTest”

Conclusion:

The backend side of the “CashSwing” app is complete. The next stage is going to be about implementing mentioned functions into our frontend side.