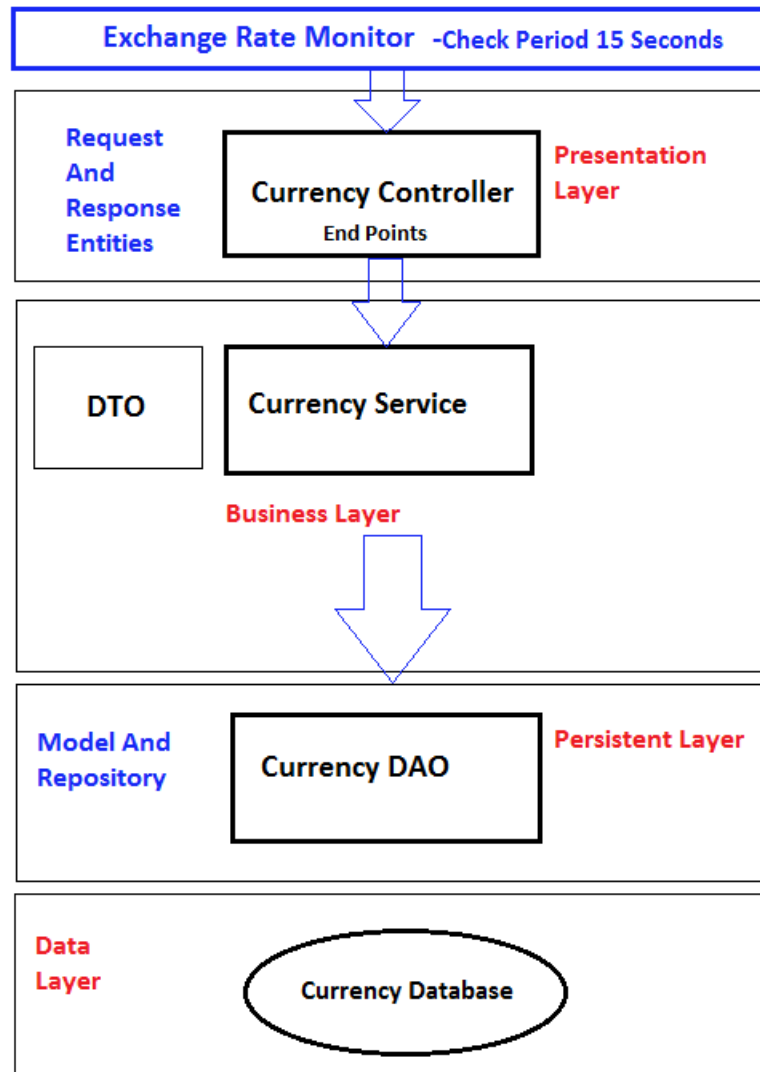


Currency Service API Documentation

1. Scope

- Currency Service has 2 Rest End points.
- Currency Service has one currency monitor component and is scheduled to keep checking latest exchange rate as per configured check period.

2. Architecture Diagram

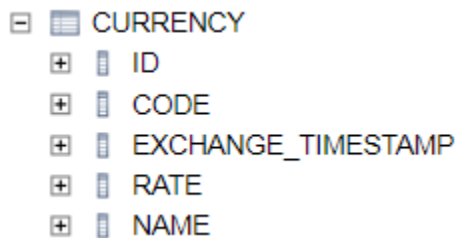


- Currency Monitor checks for latest exchange rate for 1 BTC in every 15 seconds using REST end point `getLatestRatesByBaseCurrency` – base Currency is defaulted to BTC.
- After successfully retrieving currency details -Currency Service persist Currency in database table CURRENCY.
- Currency Exchange history end point retrieves details from database.

3. Use Cases

- a. Get Latest Currency Exchange Rate
Request parameter Base currency – Default Value BTC
- b. Get Historical Currency Exchange Rate
Request Parameter Start Date and End Date
- c. Currency Monitor
Scheduled to run on every 15 Seconds -Configurable Check Interval

4. Entity Relationship Diagram



5. Interface

- a. Currency Service API can be seen using swagger interface
- b. Link to Swagger <http://localhost:1010/swagger-ui.html>
- c. Link to H2 Database Console <http://localhost:1010/h2/login.do>
Driver Class org.h2.Driver
JDBC URL jdbc:h2:file:~/test
Username sa
Password
There's no password to the H2 Database console.

6. Usage

- a. Pre-requisite – Maven should be available in the system.
- b. Run shell script from bash terminal by typing `./run-currency-service.sh`
- c. This service will start checking latest exchange rate for 1BTC to USD
- d. Check log file **currency-service.log** to verify interval

7. Sample Request and Response

- a. `getLatestRates`

Request

```
curl -X GET "http://localhost:1010/currency/getLatestRates?baseCurrency=BTC" -H "accept: application/json"
```

Response

```
{
  "responseMessage": "Successfully retrieved Current USD For 1 BTC",
  "currencyDTO": {
    "code": "USD",
    "name": "US Dollar",
    "rate": 3602.47
  }
}
```

b. `getHistoricalRatesByDate`

Request

```
curl -X GET "http://localhost:1010/currency/getHistoricalRatesByDate?startDate=2019-02-10T11%3A00%3A49&endDate=2019-02-10T16%3A00%3A49" -H "accept: */*"
```

Response

```
{
  "responseMessage": "Successfully retrieved historical currency exchange rate between
LocalDateTime 2019-02-10T11:00:49And 2019-02-10T16:00:49",
  "currencyDTOs": [
    {
      "code": "USD",
      "rate": 3607.25,
      "time": "2019-02-10T12:52:02.597"
    },
    {
      "code": "USD",
      "rate": 3609.63,
      "time": "2019-02-10T13:05:43.564"
    },
    {
      "code": "USD",
      "rate": 3590.32,
      "time": "2019-02-10T15:32:44.861"
    },
    {
      "code": "USD",
      "rate": 3595.95,
      "time": "2019-02-10T15:41:49.767"
    },
    {
      "code": "USD",
      "rate": 3596.28,
      "time": "2019-02-10T16:00:13.009"
    },
    {
      "code": "USD",
      "rate": 3596.28,
      "time": "2019-02-10T16:00:15.931"
    }
  ]
}
```

8. Technologies and Framework

JDK 8, Spring Boot 2.0, JPA, Spring Restful Service, and H2 Database.
Spring fox Swagger 2.0, Lombok, and Spring Scheduler.