

BankAccountService

BankAccountService is a micro-service that mimics some operations of a bank account

A: Functional Requirements

BankAccountService implements 3 APIs

1. Balance – return the outstanding balance
2. Deposit – credits the account with the specified amount
3. Withdrawal – debits the account with the specified amount

A.1 Function Restrictions

Deposits:

1. Maximum sum of deposits per day is \$150K
2. Maximum amount per deposit is \$40K
3. Maximum deposit frequency is 4 transactions per day

Withdrawal:

1. Maximum sum of withdrawals per day is 50K
2. Maximum amount per withdrawal is 20K
3. Maximum withdrawal frequency is 3 transactions per day

B: Components

The service requires:

1. Scala 2.12.6
2. Java 8
3. SBT 1.1.6
4. Play
5. Slick
6. Guice
7. H2 database (MySQL – optional)
8. Scoverage
9. ScalaTestPlus
10. JUnit

C: Setup:

C.1 Database (H2)

The service uses a H2 database in embedded mode. The database file (**acctservice_db.mv.db**) can be found in:

/bankaccountservice_0/resources directory. This directory also contains (**init.sql**) file, for creation of the tables. The application expects these two files in the home directory (**~/<user>**) of the user running the application. If no database file is found, one is created automatically. However, the **init.sql** file must be copied manually to the same location.

Before running the application, copy the two files to a location on your system and configure the new absolute path in the **application.conf** file in the property below

```
slick.dbs.default.db.url = "jdbc:h2:~/acctservice_db;INIT=RUNSCRIPT FROM
'~/init.sql';MODE=MYSQL;DB_CLOSE_DELAY=-1;DATABASE_TO_UPPER=FALSE"
```

If not copied, the database will not connect and the application will fail.

D: Usage

Note: On the first run, the database is empty. Use this method to insert a dummy record

<http://<host>:<port>/xyzbank/createTestRecord>

Use debug to get a dump of accounts and transactions in the database. Use the account_ids for Tests and Unit Tests

<http://<host>:<port>/xyzbank/debug>

Balance:

http://<host>:<port>/xyzbank/account/balance/{account_id}

The default port is 9000.

A sample success Json output is:

```
{
  "Status": "Ok",
  "Message": "current account balance is 28310.0"
}
```

Deposit

Construct a Json Request and send it via PUT to

<http://<host>:<port>/xyzbank/account/deposit/depositPut>

A sample Json input is:

```
{
  "id": "1532560028767",
  "txn_type": "deposit",
  "account_id": "1532627127911",
  "amount": 20,
  "event_time": 1532560028767
}
```

A sample success Json output is:

```
{
  "Status": "Ok",
  "Message": "Deposit successful"
}
```

Withdrawal

Construct a Json Request and send it via PUT to

<http://<host>:<port>/xyzbank/account/withdraw/withdrawPut>

A sample Json input is:

```
{
  "id": "1532560028767",
  "txn_type": "withdraw",
  "account_id": "1532627127911",
  "amount": 15,
  "event_time": 1532560028767
}
```

A sample success Json output is:

```
{
  "Status": "Ok",
  "Message": "Withdrawal successful"
}
```

E: Running

Clone the repository to your local machine

Change directory to the root folder containing the source (/bankservice_0)

Execute **sbt compile** to compile and download dependencies

Execute this command in the environment:

```
sbt run <port>
or
sbt run
to use the default port 9000
```

If there are no errors Play framework starts. See Section D (Usage) on REST commands

F: Running Test Cases

Execute the command:

```
sbt test
```

G: Running Coverage

Run the commands

```
sbt coverage test
sbt coverageReport
```

The report data will be written to a directory (<yourpath>/target/scoverage-report) which will be shown on the execution console. They are in HTML and XML formats

H: Load Testing

A JMeter test plan (BankServiceTestPlan.jmx) and test data (testData.txt) is included. Import the test plan and into JMeter from the public/LoadTests.

Configure the “CSV Data Set Config” config element <Filename> to point to the location where the testData.txt is saved. Alternatively, copy testData.txt to the /bin directory of JMeter and configure just the filename and not the full path

I: MySQL Database (This section applies if MySQL database is used)

Configurations for MySQL have also been provided. To use MySQL, comment H2 database settings and uncomment MySQL settings in application.conf

Change settings for MySQL in conf/application.conf and apply the relevant values for your environment for

- <url>
- <user>
- <password>

Create the account and transaction tables

```
CREATE SCHEMA IF NOT EXISTS test_db

CREATE TABLE IF NOT EXISTS account (
  id varchar(20) NOT NULL,
  customer_id varchar(20) NOT NULL,
  current_balance double NOT NULL DEFAULT '0',
  PRIMARY KEY (id)
);

CREATE TABLE IF NOT EXISTS transaction (
  id varchar(20) NOT NULL,
  account_id varchar(20) DEFAULT NULL,
  amount double DEFAULT NULL,
  transaction_type varchar(15) DEFAULT NULL,
  event_time bigint(8) DEFAULT NULL,
  PRIMARY KEY (id)
);
```