**Relative account balance from CSV transaction records**

-Ravi Kumar Enukonda

1. **Overview and design:**

The application calculates the ***relative account balance*** for the list of transaction records from input CSV file. And also finds ***number of transactions***.

I have created a Maven project and developed it in intellij IDE. I have completely used **JAVA 8 concepts like streams and lambda expressions**. And used efficient ways like using AtomicInteger wherever it is applicable. Written number of unit test cases to check the functionality of the application and various methods.

Used *OpenCSV* library to read transactions from CSV file in to the collection of objects. Used java 8 streams and lambda expression to process the date from collections. And used *Lombok* to keep the code clean.

I have created a *enum* TransactionType*,* to differentiate PAYMENT and REVERSAL transactions while processing the data. And date parsing is handled in the utility class DateParse.java.

As part of the functionality implementation, business logic is implemented in the class *AccountSummary.java*. Class *GetUserData.java* is used to take user inputs, and to parse them. Created model classes and utility classes as per the need and domain. Created different test set CSV files under resources directory for unit testing.

Based on the assumptions given in the assignment pdf file, it is considered that the records in input CSV file are in valid format and in order.



1. **Build the code, generate executable Jar file and run it:**

**Option 1:**

1. Extract the zip file AccountSummary\_Ravi.zip

Or checkout the Git repository “<https://github.com/raviassignment/mebank.git>”

1. Navigate to the directory “…\accountsummary”
2. Open the command prompt and execute the commands below

**mvn install**

**mvn clean compile assembly:single**

The above command will generate the executable jar file “*account-summary-1.0.jar*” in the directory “…\accountsummary\target”

**Note:** I have used *maven-assembly-plugin*, so that we can generate executable JAR file with all the dependencies.

1. Run the Jar file.

Navigate to the directory “…\accountsummary\target”

**java -jar -Xmx512m account-summary-1.0.jar <Path\_To\_CSV\_File> <AccountID> <fromDate> <toDate>**

**Note:** -Xmx512moptionspecifies *the* maximum memory for java process. It can be increased or omitted based on need.

Example Run:

Ravis-MBP:target ravikumarenukonda$ java -jar -Xmx512m account-summary-1.0.jar "/Volumes/RaviData/accountsummary/src/main/resources/transactions\_TestSet1.csv" "ACC334455" "20/10/2018 12:00:00" "20/10/2018 19:00:00"

**Output:**

*Relative balance for the period is: -25.0*

*Number of transactions included is: 1*

**Option 2:**

Import the source code in to intellij, do maven install, compile from IDE and run it by passing the required arguments.