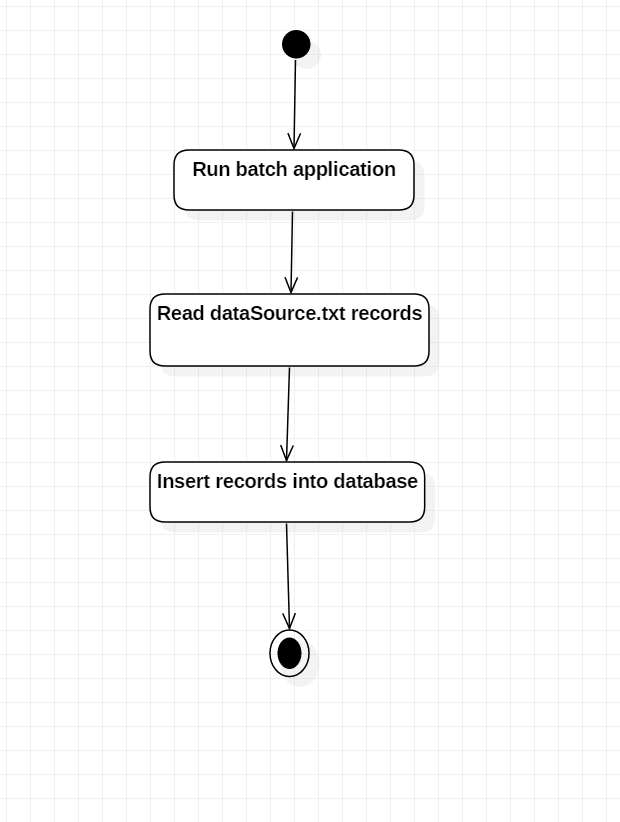
**Solution documentation**

Application developed using Springboot using Spring framework with MyBatis as ORM framework and H2 as database. Sql script for table and test data creation is at data.sql.

To run the batch to process dataSource.txt, execute BatchStarter class. Record is inserted into Transactions table. Database is reset every time application is restarted.

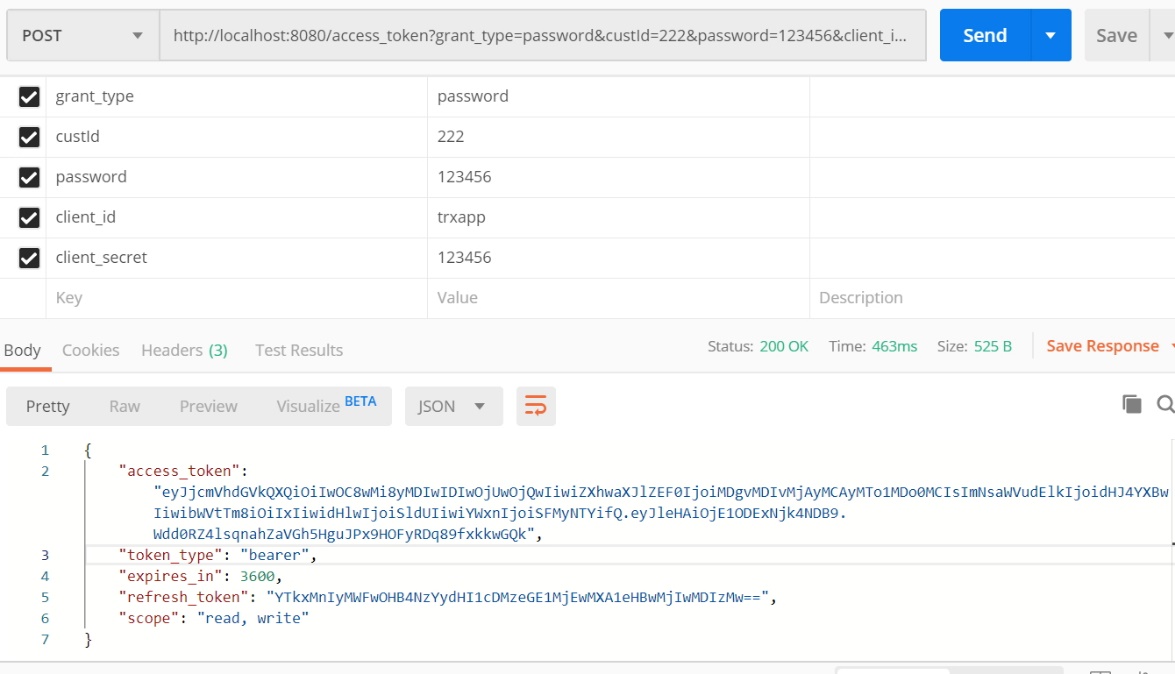
Activity diagram for batch process.



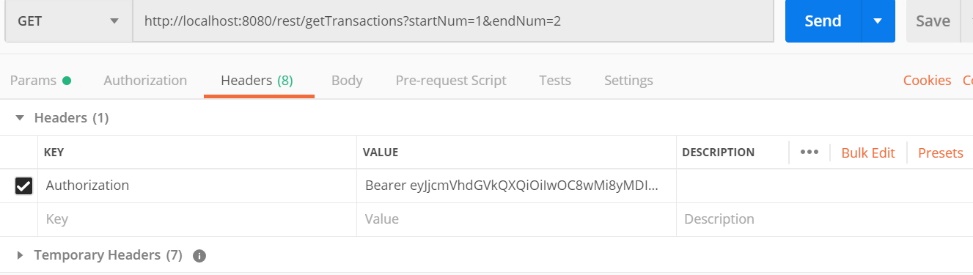
To run rest api server, execute ApplicationStarter class. Database is reset every time application is restarted.

To access the available api’s, access token needs to be obtained by accessing access\_token api. Default value for client\_id is trxapp and client\_secret is 123456. Password for customer Id 222 is 123456, customer Id 333 password 654321.

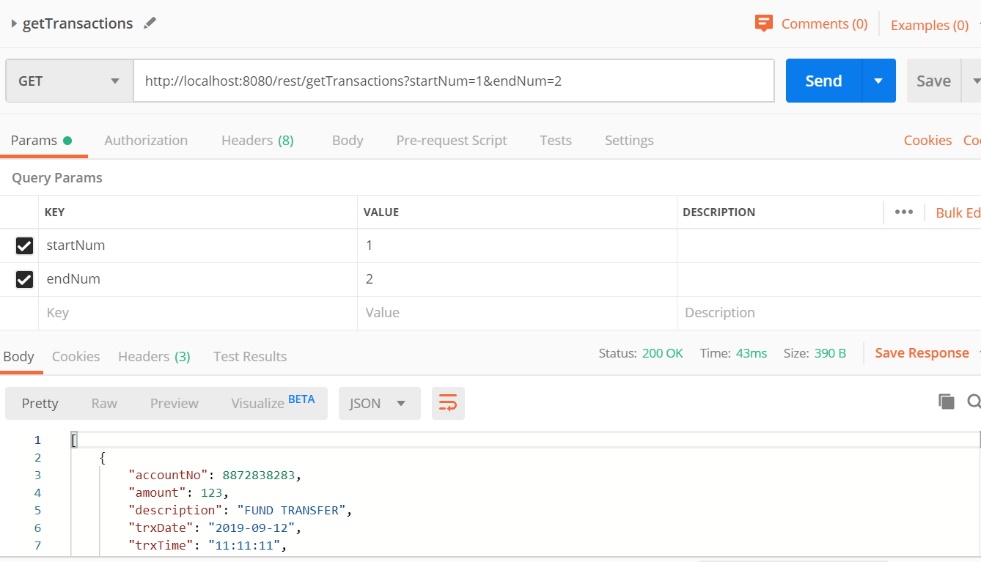
The tokens are stored in app\_api\_auth\_info table to keep track of the tokens generated based on OAuth2 standard.



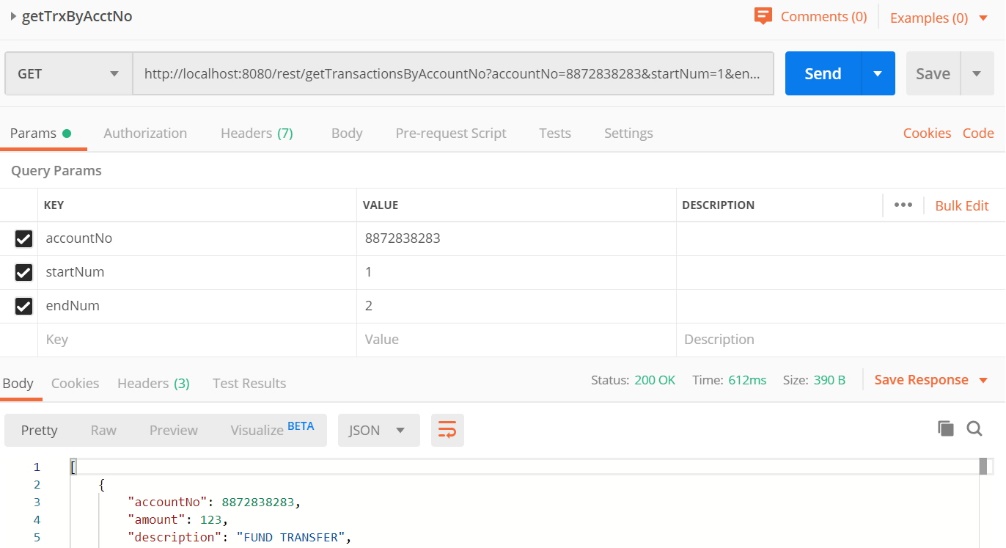
The access token is needed in the Header as value to access the apis.



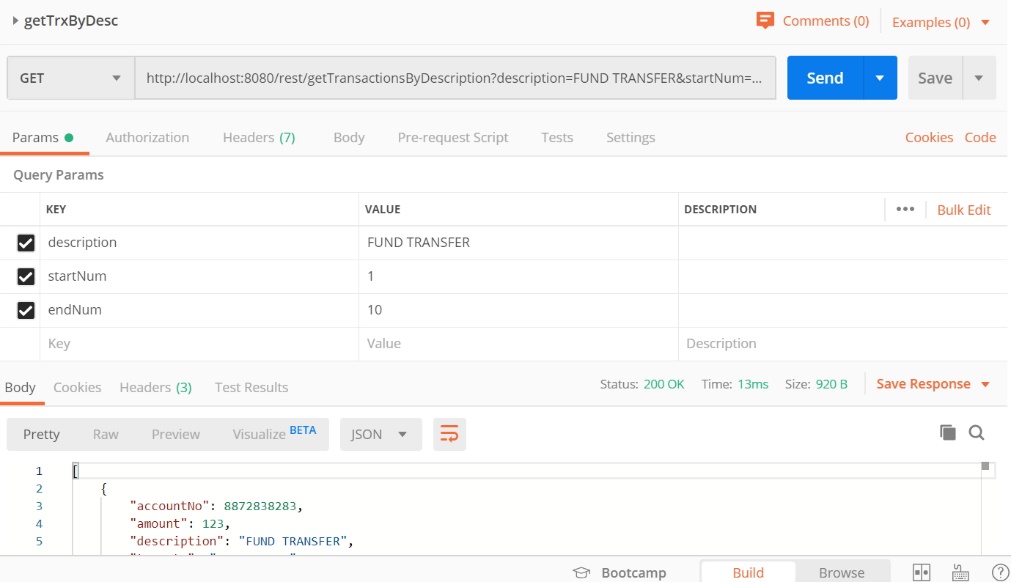
Api getTransactions is to retrieve all transactions. Parameters startNum and endNum for pagination.



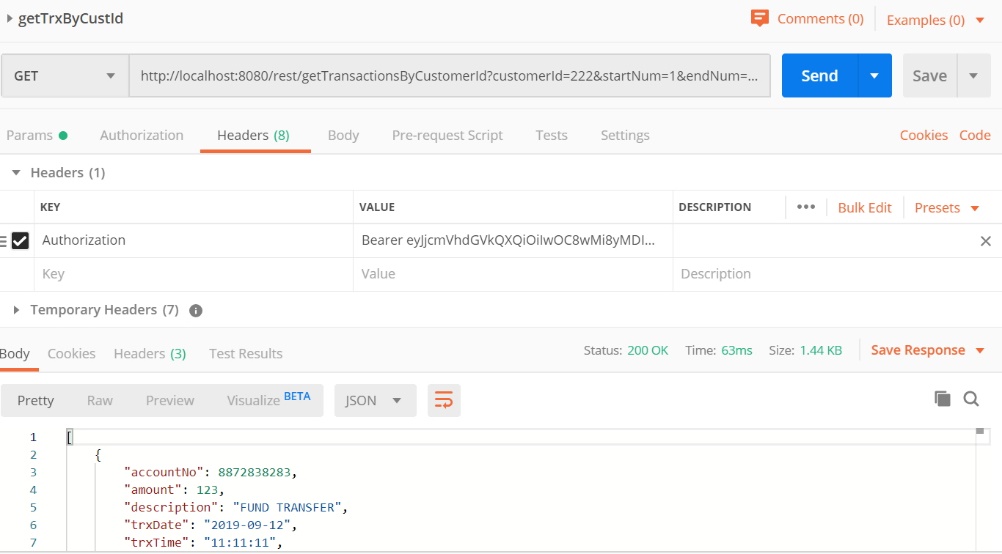
Api getTransactionsByAccountNo is to retrieve transactions by account number. Input parameters accountNo with startNum and endNum for pagination.



Api getTransactionsByDescription is to retrieve transactions by description. Input parameters description with startNum and endNum for pagination.



Api getTransactionsByCustomerId is to retrieve transactions by customer id. Input parameters customerId with startNum and endNum for pagination.

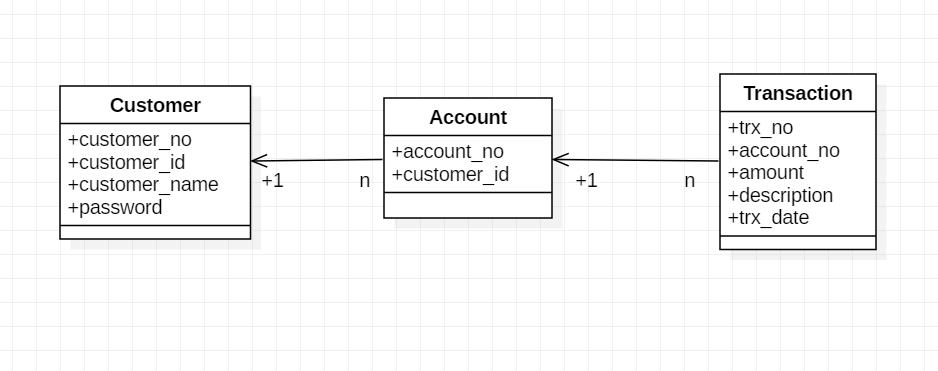


Patterns in project

Patterns used are

* Service layer pattern to integrate controller with the data access layer
* Data mapper pattern used in MyBatis when getting data from database
* Interceptor pattern used when intercepting calls to api endpoints to check whether access token is valid.

Class diagram



Activity diagram for rest api flow

