**RABO BANK STATEMENT PROCESSOR**

* This application is developed using the following frameworks/technologies/tools:
  + Spring Boot with MVC
  + Java8
  + Maven
  + IntelliJ Idea as IDE &
  + Postman for testing
* The project is designed with the following structure:
  + StatementsApplciation is the main spring boot application.
  + StatementsController handles all requests and routes to appropriate service.
  + XmlStatementsService processes XML files, validates and identifies failed transactions.
  + CsvStatementsService processes CSV files, validates and identifies failed transactions.
  + ValidationService is used by XmlStatementsService & CsvStatementsService for validating transactions against business rules.
  + CustomExceptionHandler handles exceptions in the application by logging the actual error messages in the logs and responding user with a graceful message
  + Logback is used for application logging.
  + Maven is used for building application.
  + The project can be extended in future by adding new services like JsonStatementsService and injected into the StatementsController.
  + Spring boot tests are done in StatementsApplicationTests. The xmlprocessor and csvprocessor are used to test xml and csv file inputs. There is a negative case as well when input file is not given.
* The jar file under targets/statementsprocessor-0.0.1-SNAPSHOT.jar can be executed with the following command
  + java -jar target/statementsprocessor-0.0.1-SNAPSHOT.jar
* Once the spring boot application starts, the index.html can be hit as:
  + <http://localhost:8080/index.html>
* Choose XML/CSV file upload.
* The file is processed and the failed transactions are returned as a report.
* Optionally individual services can be tested in postman as follows:
  + <http://localhost:8080/customer/statements/xmlprocessor>
  + Method:POST
  + Content-Type : FormData
  + Parameter : file – type:file and value:records.xml
  + <http://localhost:8080/customer/statements/csvprocessor>
  + Method:POST
  + Content-Type : FormData
  + Parameter : file – type:file and value:records.csv