**File Parser Service**

**Introduction:**

This File Parser Service is used to validate Transaction Reference and End Balance. This service will validate if the Reference is duplicate and also it will validate if the End Balance is negative. If any validation fails then the service will send the json data which will have both the transaction reference and description of each failed records.

This service has been developed using below technologies.

* JAVA 8
* Spring Boot
* Maven
* Junit

**Strategy Design Pattern:**

Strategy Pattern is used when there is a family of interchangeable algorithms of which one algorithm is to be used depending on the program execution context.

com.rabo.parser package contains below classes

1. BaseFileParser
2. CSVFileParser
3. XMLFileParser
4. ParserClient

* BaseFileParser class is an abstract base class which represents family of FileParser instances in the design. It has a single abstract method parseFile() which needs to be invoked to parse a file. This method will be overridden by individual parsers.
* XMLFileParser & CSVFileParser both implement BaseFileParser. Both also override parseFile() method for processing information in XML and CSV formats.

**Logging:**

Logger is enabled and rabo.log is available in application root. Logging level can be changed in application properties based on the requirement

**Environment Properties:**

DUPLICATE\_ERROR and ENDBALANCE\_NEGATIVE\_ERROR properties has been set in application properties to avoid hot coding.

**How to use this service**:

Build the jar using **Maven** :

*mvn clean package*

Deploy the jar in Tomcat server which is java8 supported.

Place the records.xml or records.csv under resources path.

Start the Tomcat Server

URL to run:

[http://hostname:port/parser/{typeoffile}/records.{typeoffile}](http://hostname:port/parser/%7btypeoffile%7d/records.%7btypeoffile%7d)

**Example:**

http://hostname:port/parser/csv/records.csv

http://hostname:port/parser/xml/records.xml