

Software Architect Technical Assignment

Description

Implement a service to upload transaction data from files of various formats into database and query transactions by specified criteria.

Use best practices and design patterns, skills in design/architecture, ability to build testable and maintainable software.

Given

You have two possible formats of input files: csv and xml based. All values are mandatory so if one missing then record is invalid. If any record is invalid whole file is treated as invalid and should not be imported. However you should be able to identify all records that was invalid and collect it in some log.

CSV format description:

Index	Data	DataType
1	Transaction Identifier	Text max length 50
2	Amount	Decimal Number
3	Currency Code	Text in ISO4217 format
4	Transaction Date	Format dd/MM/yyyy hh:mm:ss
5	Status	Choice of: <ul style="list-style-type: none">- Approved- Failed- Finished

Example:

"Invoice00000001","1,000.00","USD","20/02/2019 12:33:16","Approved"
"Invoice00000002","300.00","USD","21/02/2019 02:04:59","Failed"

XML Format:

Data	DataType
Transaction Id	Text max length 50
Amount	Decimal Number
Currency Code	Text in ISO4217 format
Transaction Date	Format yyyy-MM-ddThh:mm:ss e.g. 2019-01-23T13:45:10
Status	Choice of: <ul style="list-style-type: none">- Approved- Rejected- Done

Example:

```
<Transactions>
  <Transaction id="Inv00001">
    <TransactionDate>2019-01-23T13:45:10</TransactionDate>
    <PaymentDetails>
      <Amount>200.00</Amount>
      <CurrencyCode>USD</CurrencyCode>
    </PaymentDetails>
    <Status>Done</Status>
  </Transaction>
  <Transaction id="Inv00002">
    <TransactionDate>2019-01-24T16:09:15</TransactionDate>
    <PaymentDetails>
      <Amount>10000.00</Amount>
      <CurrencyCode>EUR</CurrencyCode>
    </PaymentDetails>
    <Status>Rejected</Status>
  </Transaction>
</Transactions>
```

Requirements

Create Web application with ability to:

- 1) Upload file. Create a web-page with standard file uploader. Must support both formats csv and xml. File size is max 1 MB. Save data into database. Feel free to design database structure that is suitable for this.
 - a) If file is in unknown format then return error message "Unknown format".
 - b) If file didn't pass validation return Bad Request with all validation messages.
 - c) If everything is okay then return HTTP Code 200.
- 2) Get all transactions. Create API methods:
 - a) by Currency
 - b) by date range
 - c) by status

Transaction should display these values:

- a) Id
- b) Payment = Amount + CurrencyCode
- c) Status in unified format (see table below)

Transaction status mappings:

CSV	XML	Output Status
Approved	Approved	A
Failed	Rejected	R
Finished	Done	D

Example of output:

```
[{"id": "Inv00001", "payment": "200.00 USD", "Status": "D"},
{"id": "Inv00002", "payment": "10000.00 EUR", "Status": "R"},
{"id": "Invoice0000001", "payment": "1000.00 USD", "Status": "A"},
{"id": "Invoice0000002", "payment": "300.00 USD", "Status": "R"}]
```

Time frame

48 hours

Source Code

Push your code to Github/Bitbucket including SQL Script.

We prefer to see that the candidate can use source control systems, so please make sure you're not pushing everything in a single commit, but having small atomic steps. Commit clean project ignoring binaries and unnecessary resources

Tools and libraries

Candidate is free to use any additional third-party libraries and frameworks.