Ex Banking Services

API Integration and Testing

Overview

The purpose of this document is to explain how API integration and API testing have been conducted on the Ex-Banking services. This document is containing one or more of the following.

- Used tools and languages
- Installation instructions
- Identified nonfunctional and functional test cases
- Configuration of Mock service
- Process of nonfunctional test case automation

Used Tools and Languages

Used 'Postman' as the tool to integrate and automate the API test assert. And for the mocking, the APIs used 'Postman' as the tool. When asserting the APIs use JavaScript as the language.
Since 'Postman' is not directly supported to automate the nonfunctional test case that I have chosen, decided to use the 'K6' tool to test out the performance of the created APIs.
To write down the identified functional and nonfunctional test cases used 'Google sheet'.

Installation Instructions

1. Install Postman
    a. https://www.postman.com/downloads/
2. Install K6
    a. https://k6.io/docs/get-started/installation/

Identified Nonfunctional and Functional Test Cases

Based on the given instruction and the APIs identified and designed the functional and non-functional test cases in the google sheet. (Test case document has attached along with this.)

Selected Functional Test Cases for the Automation

Create User

1. Send create user request with valid first name, last name, date of birth, nationality, username, password, national identity number and email
2. Send create user request with invalid email
3. Send create user request without username

Deposit

1. Send deposit request with valid first name, last name, account number, amount and national identity number
2. Send deposit request with invalid account number
3. Send deposit request with amount which is greater than to the balanced amount

Withdraw

1. Send withdraw request with valid first name, last name, account number, amount and identification number
2. Send withdraw request without account number
3. Send withdraw request with amount as a floating number

Get Balance

1. Send get_balance request with valid first name, last name, account number and identification number
2. Send get_balance request with invalid account number

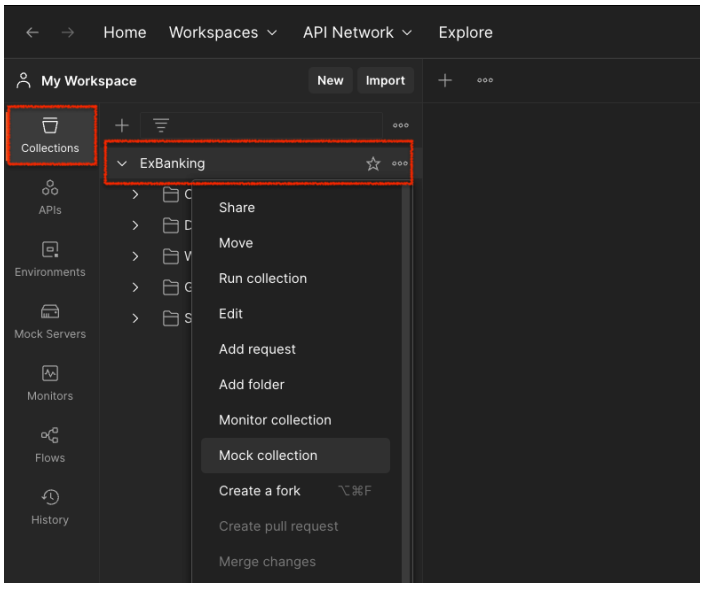
Send

1. Send request with valid first name, last name, remarks, receiving account number, sending account, identification number, amount
2. Send request with using same account number for sending and receiving account numbers
3. Send request with minus amount

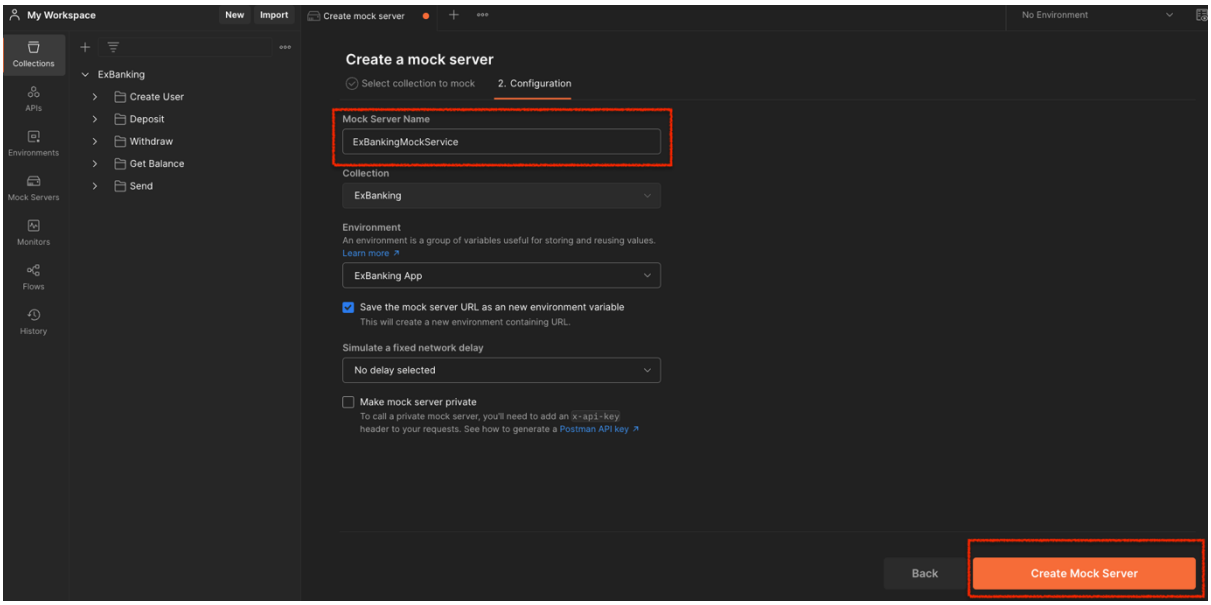
Configuration of Mock service

- Import shared postman collection and environment variables file.
- The shared collection contains a mock service endpoint.
- Follow the below steps to create a new mock service endpoint.

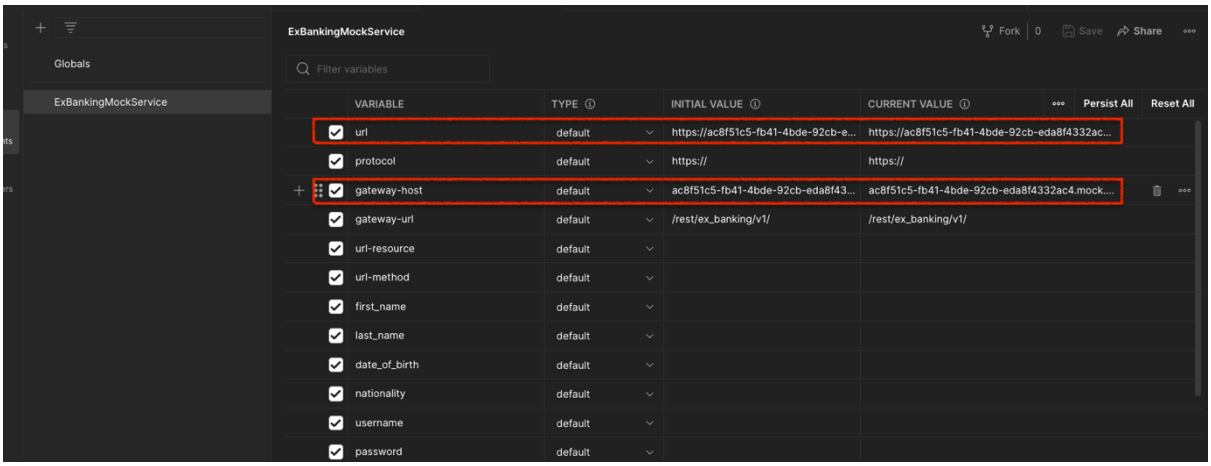
1. Right-click on the collection from the top right corner “more” icon and select “Mock collection”.



2. Insert “Mock Server Name” and select the collection. Create Mock Server button.



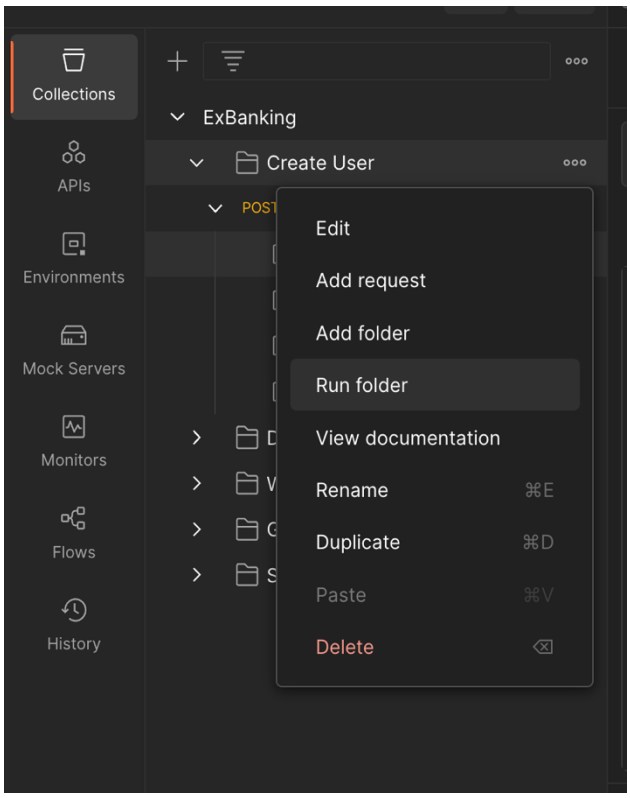
3. Copy and paste the values to environment variables,
   - url = full URL value
   - gateway-host = only the domain name



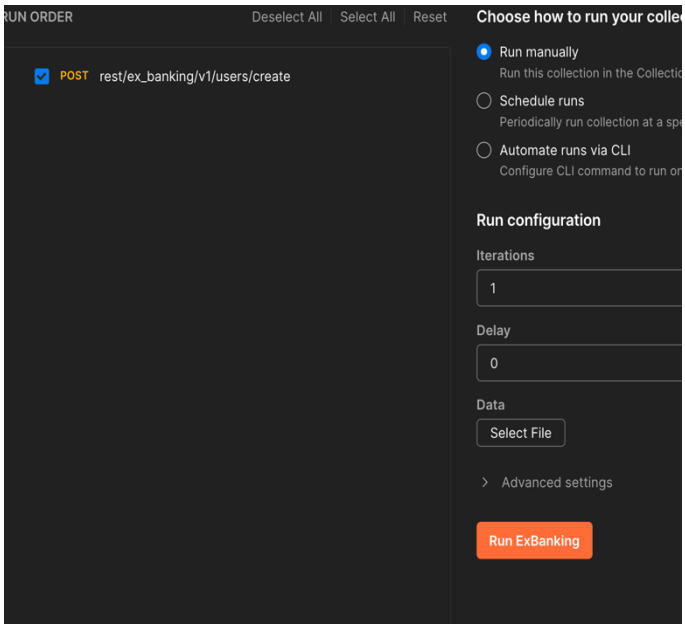
Execute the Mock Service

To execute the mock service use file from the test data folder.

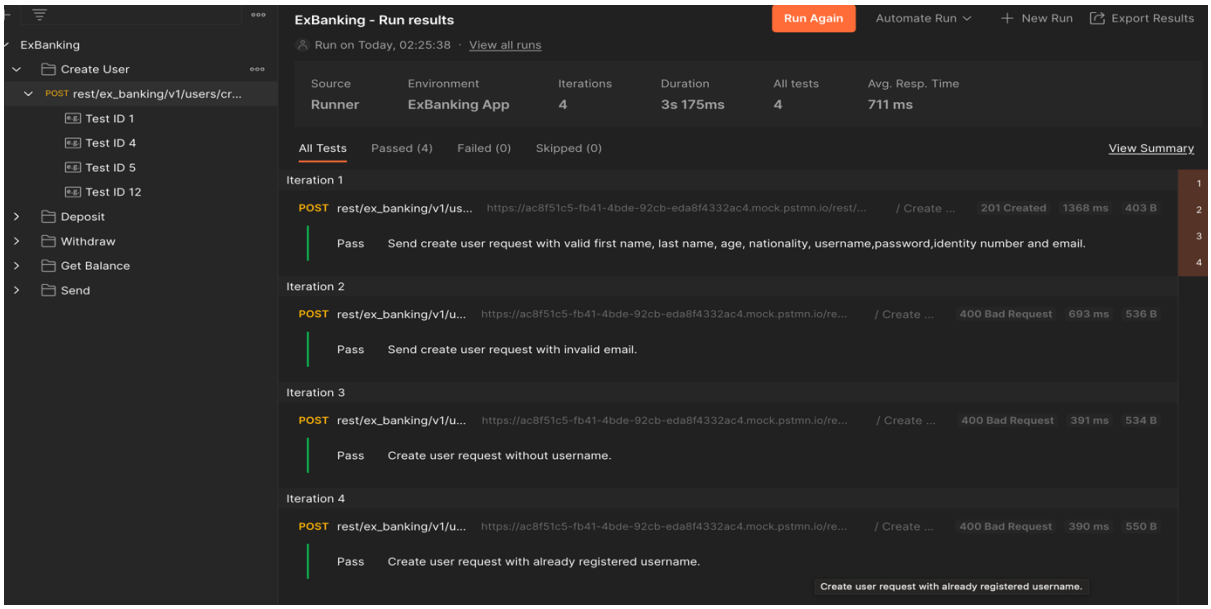
1. Right-click on the folder inside the collection from the top right corner “more” icon and select “Run Folder”.



2. Select the correct JSON file from the folder and run the tests.



3. Following results can be seen in the postman; similarly, you can execute for other test files.



Process of Non-functional Test Case Automation

As explained in the Used Tools and Languages, for the load test use the 'K6' performance tool. Exported Postman API collection has been integrated with the K6 and runs the performance testing script. Here attached the screenshot of the results after executing the performance test.

Selected Test Case as the Non-functional Test case

1. Send request with valid first name, last name, remarks, reciveing account number, sending account, identification number, amount