INTRODUCTION

ACCELQ provides comprehensive support for executing and verifying REST API. There isn't much difference in how you setup your action logic for API testing compared to UI testing. And you can also develop an Action that includes both UI and API testing together, making possible a fluid end-to-end verification.

So in this project we are going to create a Qbank Login and Authentication.Firstly we have to add extension to our browser i.e.Accelq view recorder , download accelq agent.

Then, We will write a Scenario to verify login functionality on QBank. You will be introduced to the basic building blocks to develop simple, yet meaningful automation logic. We will also run this first test and review the result.

This Scenario will

- ➤ Log in to the QBank application
- ➤ Create response for both server and rest api
- ➤ Run the test cases

LITERATURE SURVEY

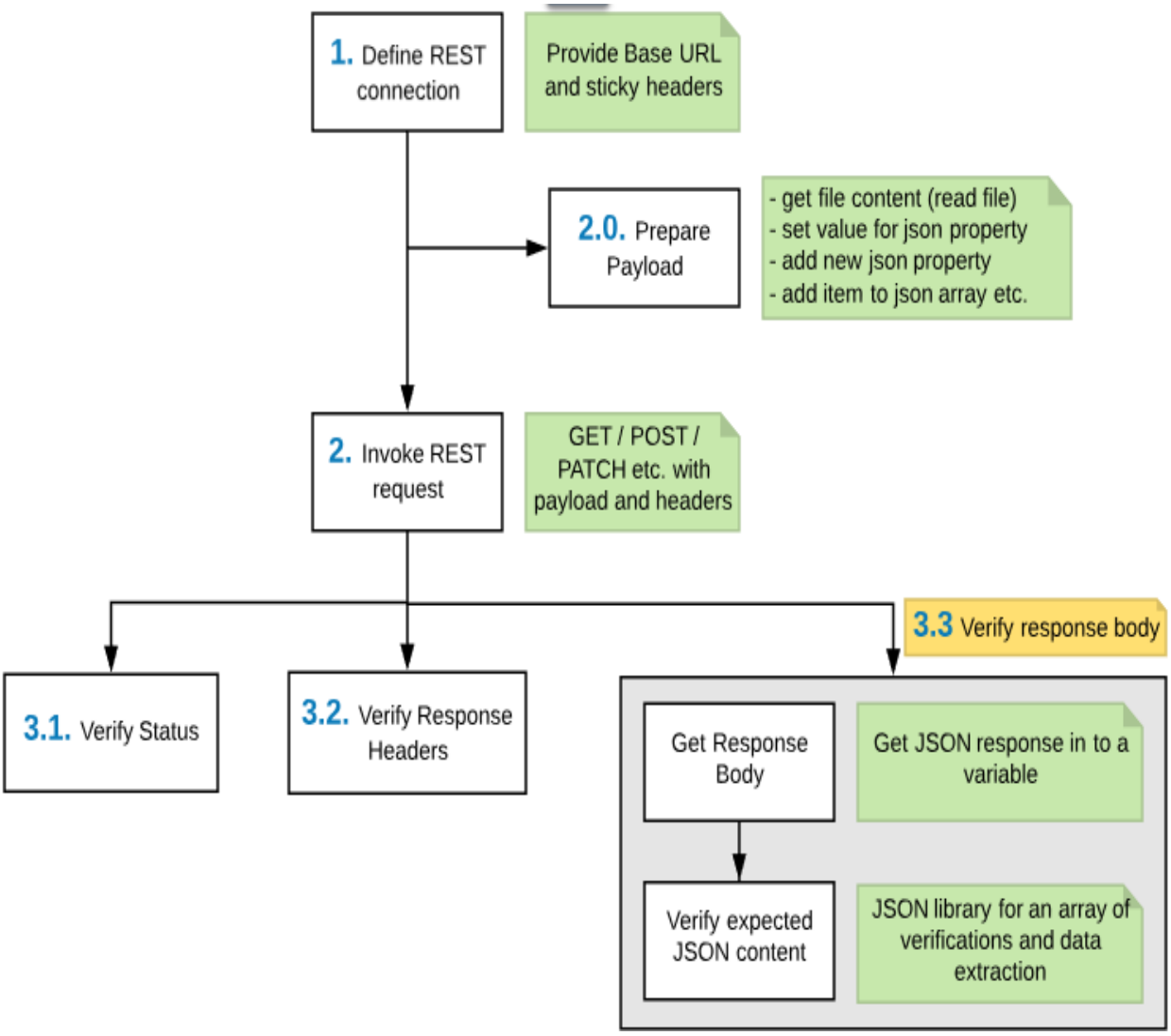
Considering the current scenario, all the tools and processes around API testing are mostly developer-oriented, driven by white-box testing and the automation is limited to just request-response automation.

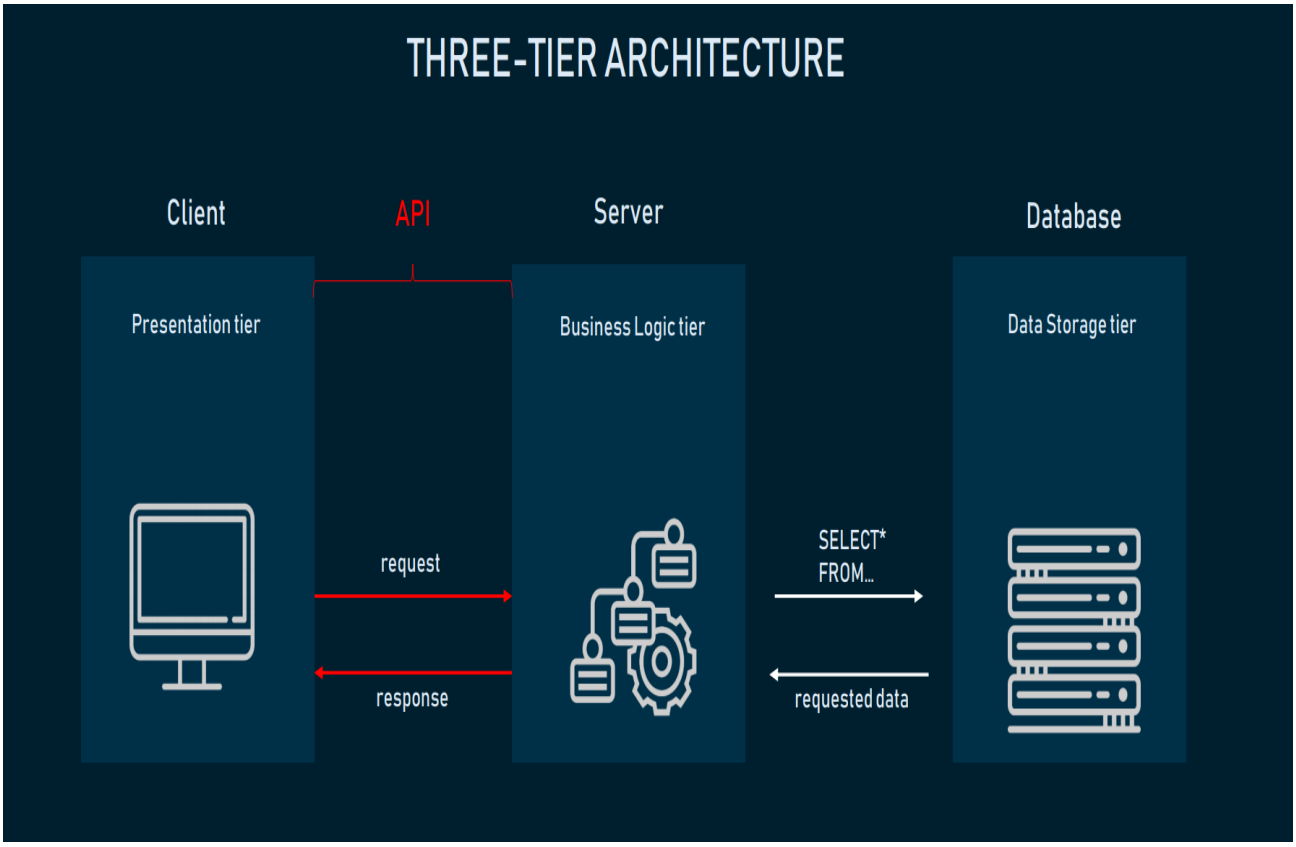
ACCELQ has the ability to design, automate, and execute through the API regression suite from a complete business perspective. With codeless API test automation, it can seamlessly integrate with UI testing.

QBANK API automation is not only about automating request-response verifications. It should bring the same level of maturity as a functional regression. ACCELQ brings the same level of maturity as functional black-box testing with full-blown regressions and integrated end-to-end validations across UIs and APIs.

we aims to remove the technical complexity from the test automation to make it easily accessible to the entire testing community without having to make any compromise in scalability and robustness. It helps businesses in not only implementing test automation that provides an immediate return on investment (ROI) but also to enable continuous delivery.Comunicating through servers may give more efficiency , security to the user.

THEORITICAL ANALYSIS




THREE-TIER ARCHITECTURE
Client
API
Server
Database
Presentation tier
Business Logic tier
Data Storage tier
request
response
SELECT*
FROM...
requested data

Hardware / Software designing

ACCELQ local agent can run a windows desktop, a linux machine or a MacBook. Note that the agent only serves as a machine from where the browser will invoke to run the test. It is not necessary to bring a server class configuration.
Here are some configuration parameters

- ⭐Quad-core processor
- ⭐8 GB RAM
- ⭐250 GB hard-drive
- ⭐JDK 11. Make sure to have Java Development Kit, not just the runtime/JRE.

Windows, Linux or Mac Operating System. For Linux machines, make sure it has GUI or desktop environment enabled.

Make sure to setup local agent with necessary hardware and memory configuration to support execution requirements. Typically, an 8 GB RAM supports execution of up to 2 jobs concurrently.

Depending on the need for additional concurrency, appropriately scale up the RAM size, with about 1.5 GB RM for every additional concurrent job. For example, with 16 GB RAM, you could scale up to 6 concurrent jobs.

EXPERIMENTAL INVESTIGATIONS

The API is intended to be RESTful supports multiple output formats. This means that to make a call to QBank all that is needed is a simple HTTP request. Almost all programming languages and platforms are able to do that and you can even test the calls directly in your browser. The API is designed to allow the consumer to perform all tasks available in QBank with ease. Simple and powerful is the keywords.
Base URL
All URLs referenced in the documentation are relative to the following
http://qbankserver.herokuapp.com/api/login

QBANK REST API

REST has grown to be the de-facto standard of web-accessible APIs and has more or less replaced other standards, eg. The key design principles of REST are:
Uniform interface — Every resource is identified via a unique URL. The resource representation is separate from the implementation, easily understood and manipulated.
Stateless interaction — No client context is stored on the server. All necessary information is included in each request.
Cacheable — A client may cache a response if it is defined as cacheable.
Client-Server — The separation increases the portability of client code and simplifies server implementation.

Authentication

Authentication is done via verification of username and password.
Currently we support the flows client libraries.
To use Authentication Code you must obtain a client identifier
Response formats

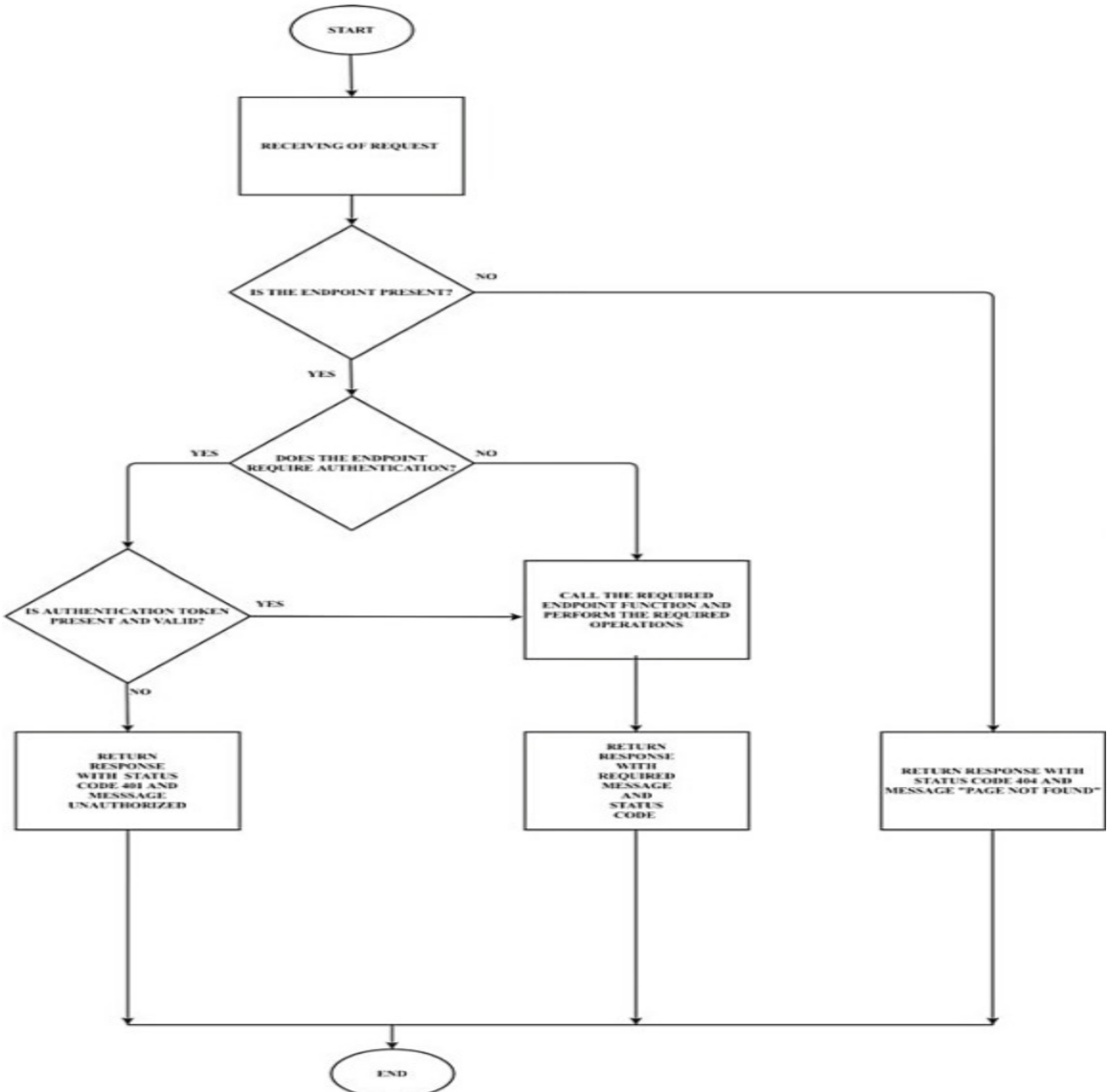
The API supports responses in both JSON and XML interchangeably.

Endpoints

All endpoints are documented and testable at the special endpoint . From there you can browse all the available actions directly in your browser. If you want to, you can even use it live if you authorize it with your QBank account.

http://qbankserver.herokuapp.com/api/login

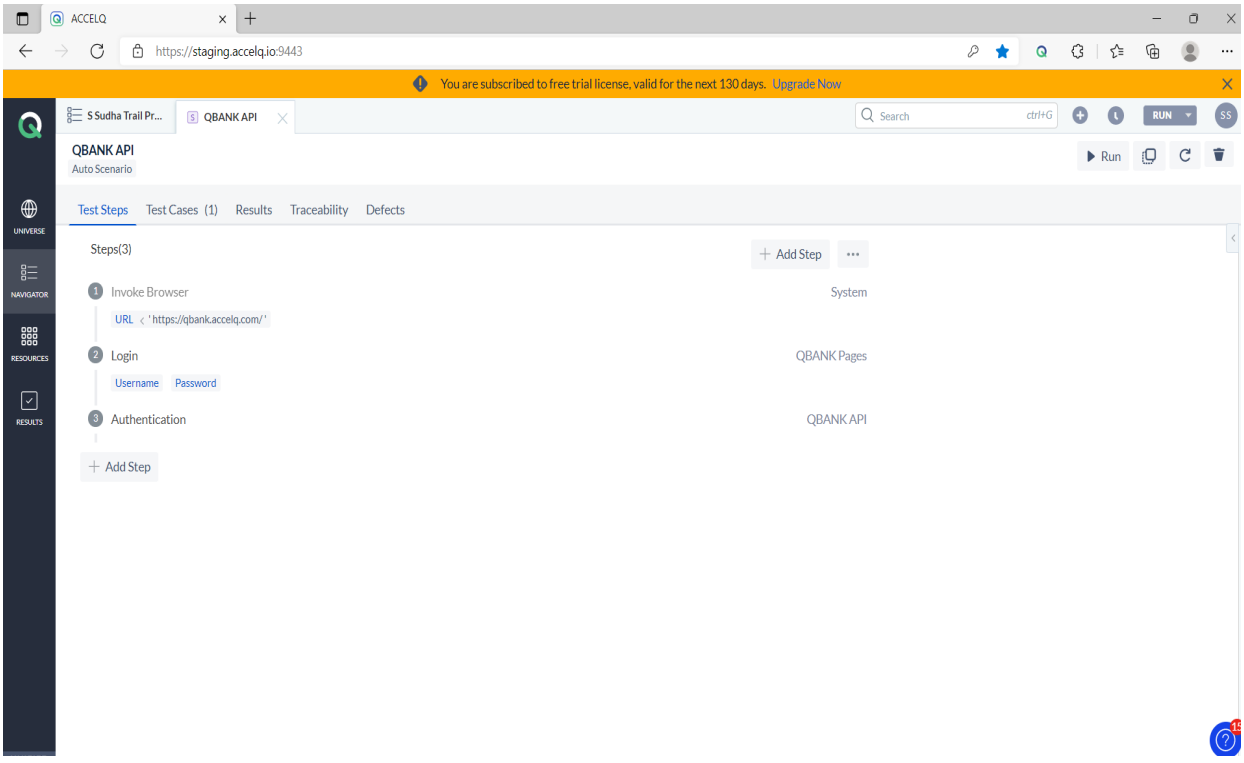
FLOWCHART

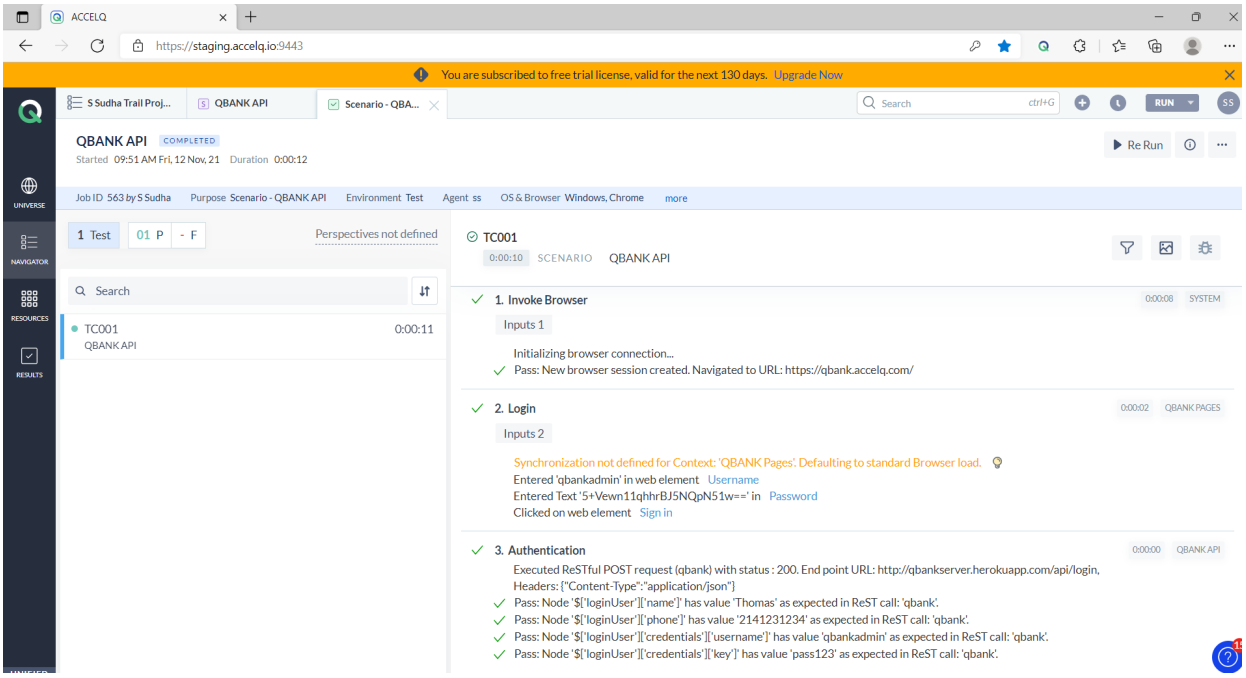


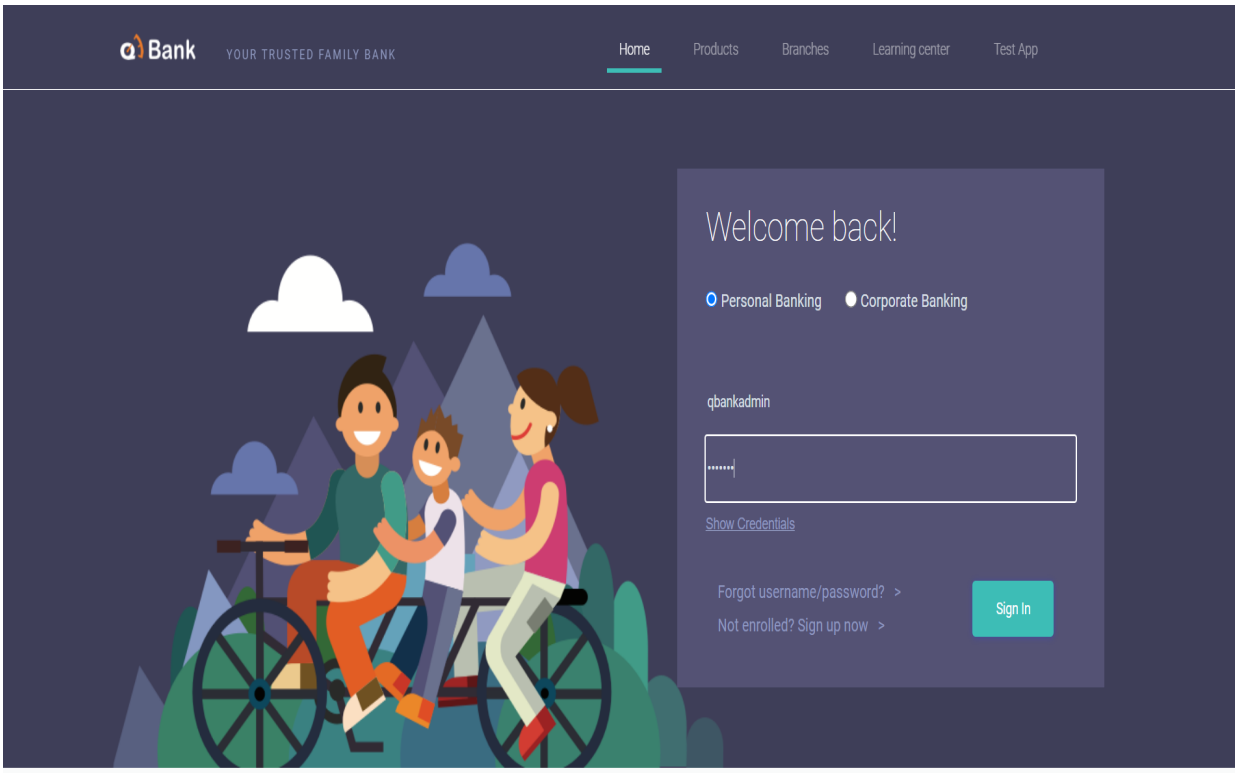
RESULT

- Authentication of QBANK REST API runned successfull
- Username verified successfull
- Password verified successfull
- All the credentials are verified by using API responses
- Test cases passed

screenshots




ACCELQ
https://staging.accelq.io:9443
You are subscribed to free trial license, valid for the next 130 days. Upgrade Now
S Sudha Trail Proj...
QBANK API
Scenario - QBA...
Search
RUN
QBANK API
COMPLETED
Started 09:51 AM Fri, 12 Nov, 21 Duration 0:00:12
Re Run
Job ID 563 by S Sudha
Purpose Scenario - QBANK API
Environment Test
Agent ss
OS & Browser Windows, Chrome
more
1 Test
01 P
- F
Perspectives not defined
Search
TC001
QBANK API
0:00:11
TC001
0:00:10
SCENARIO
QBANK API
1. Invoke Browser
0:00:08
SYSTEM
Inputs 1
Initializing browser connection...
Pass: New browser session created. Navigated to URL: https://qbank.accelq.com/
2. Login
0:00:02
QBANK PAGES
Inputs 2
Synchronization not defined for Context: 'QBANK Pages'. Defaulting to standard Browser load.
Entered 'qbankadmin' in web element Username
Entered Text '5+Vewn11qhhrBJ5NQpN51w==' in Password
Clicked on web element Sign in
3. Authentication
0:00:00
QBANK API
Executed ReSTful POST request (qbank) with status : 200. End point URL: http://qbankserver.herokuapp.com/api/login,
Headers: {"Content-Type":"application/json"}
Pass: Node '$['loginUser']['name']' has value 'Thomas' as expected in ReST call: 'qbank'.
Pass: Node '$['loginUser']['phone']' has value '2141231234' as expected in ReST call: 'qbank'.
Pass: Node '$['loginUser']['credentials']['username']' has value 'qbankadmin' as expected in ReST call: 'qbank'.
Pass: Node '$['loginUser']['credentials']['key']' has value 'pass123' as expected in ReST call: 'qbank'.


Bank
YOUR TRUSTED FAMILY BANK
Home
Products
Branches
Learning center
Test App
Welcome back!
Personal Banking
Corporate Banking
qbankadmin
Show Credentials
Forgot username/password? >
Not enrolled? Sign up now >
Sign In

ADVANTAGES & DISADVANTAGES

Language-independent

Data is exchanged via XML and JSON formats, so any language can be used for test automation. XML and JSON are typically structured data, making the verification fast and stable. There are also built-in libraries to support comparing data using these data formats.

GUI-independent

API testing can be performed in the app prior to GUI testing. Early testing means early feedback and better team productivity. The app's core functionalities can be tested to expose small errors and to evaluate the build's strengths.

Improved test coverage

Most API/web services have specifications, allowing you to create automated tests with high coverage — including functional testing and non-functional testing.

Faster releases

It is common that executing API testing saves up to eight hours compared to UI testing, allowing software development teams to release products faster.

APPLICATIONS

- QBANK REST API works essentially the same way that any website does. A call is made from a client to a server, and data is received back over the HTTP protocol.QBANK API is an easy way to show the similarities between a REST API call and the loading of a webpage.

- REST is a client-service architecture that is based on a request/response design. REST APIs have become increasingly popular as part of a Web Services approach. Developers use RESTful APIs to perform requests and receive responses through HTTP functions.

- Any API uses commands to obtain resources. The state of a resource at any given timestamp is called a resource representation. A RESTful API uses existing HTTP methodologies defined by the RFC 2616 protocol.

- Companies can update the workflow to become more productive.
- It helps you to personalize user experience.
- Organizations can customize the service with ease.
- Content can be embedded from any application or site.

.

CONCLUSION

So to conclude,QBANK REST API is a type of data transfer that is built upon the architecture of the HTTP protocol. It allows you to easily send and retrieve data between two different services using XML or JSON.

When structuring your web applications, it’s usually good practice to build them using REST architecture. This means that collections and resources are easily recognised and can form the basis of building a REST API.

A quick walkthrough of the application with the view recorder resulted in a seamless creation of the structure for the scenario and all its assets. ACCELQ takes a unique approach to produce test assets easy to reuse and manage.

I’ve found the scripting to be very expressive and the command library extensive. And like I had mentioned, reporting was top notch!

In my opinion, ACCELQ is a bold attempt which really sets the tone for the future of Quality Automation Tools. Bravo to the team for keeping it simple and efficient!

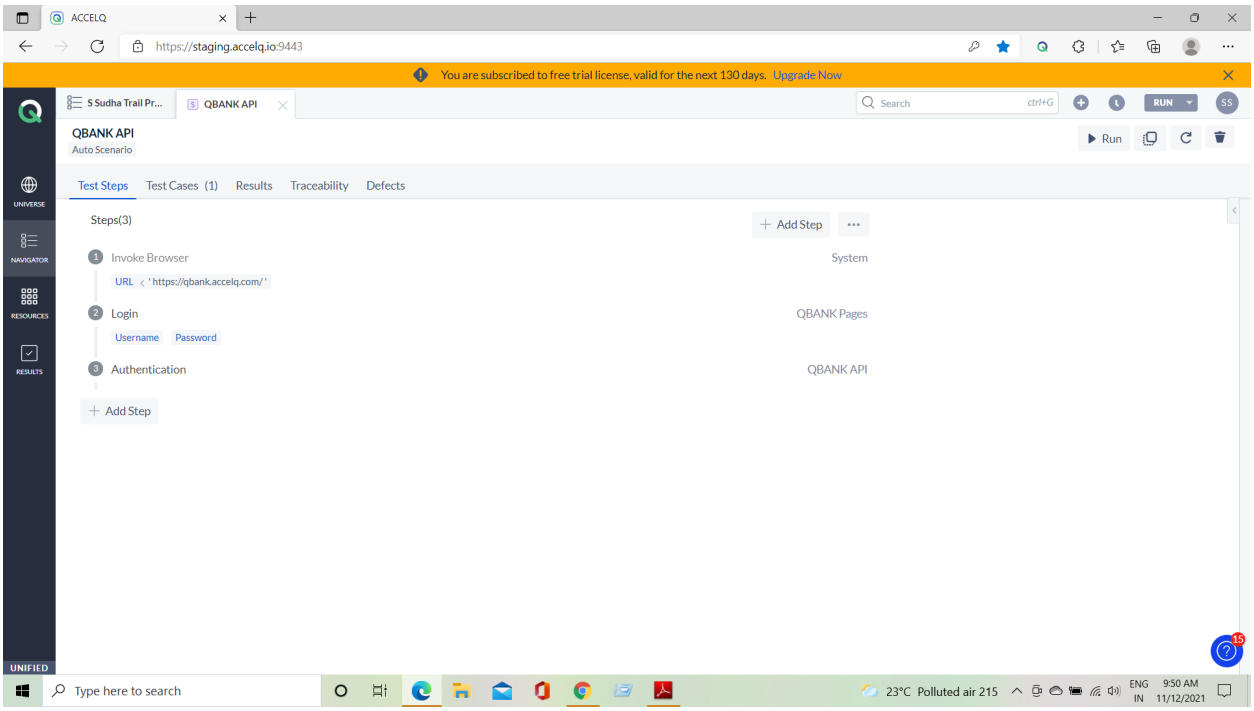
FUTURE SCOPE

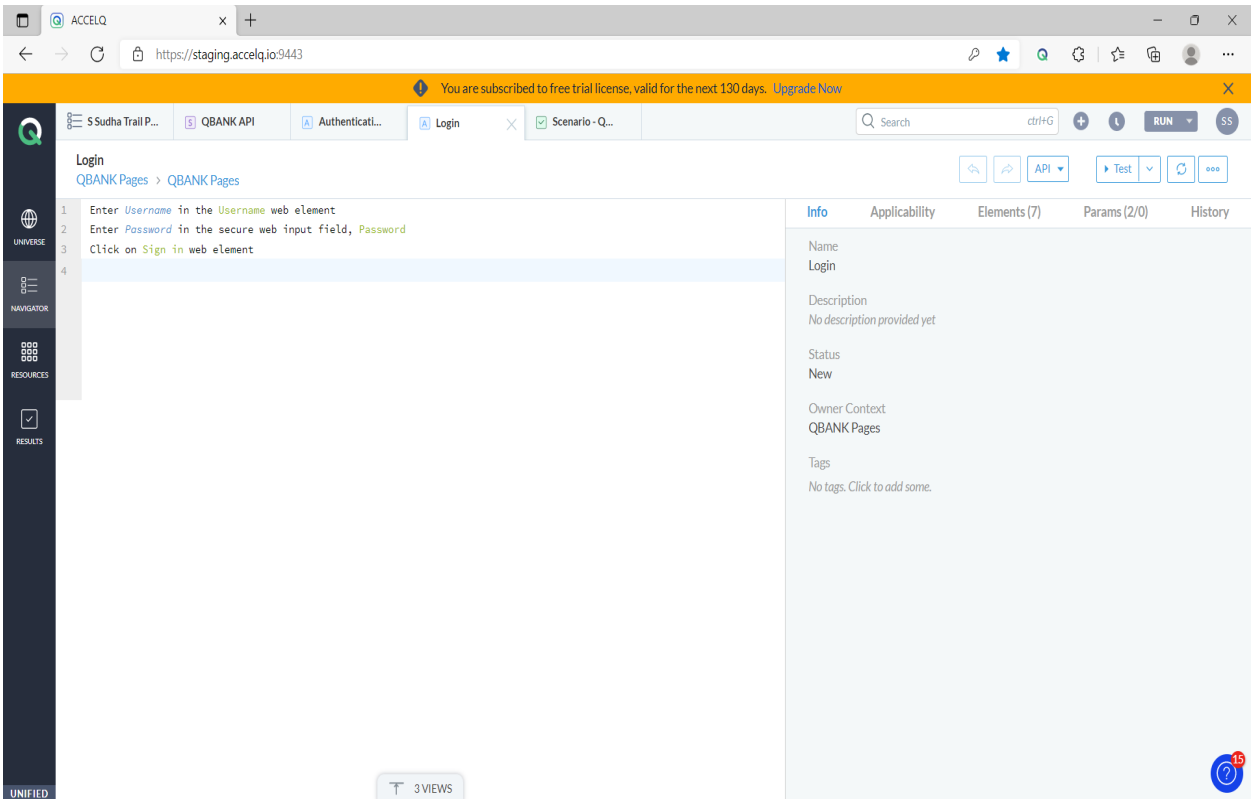
In future , our research shows that the rest api testing might be more successful by managing an investment portfolio specifically tailored to an observable communication among servers and softwares and then testing real time applications using API .

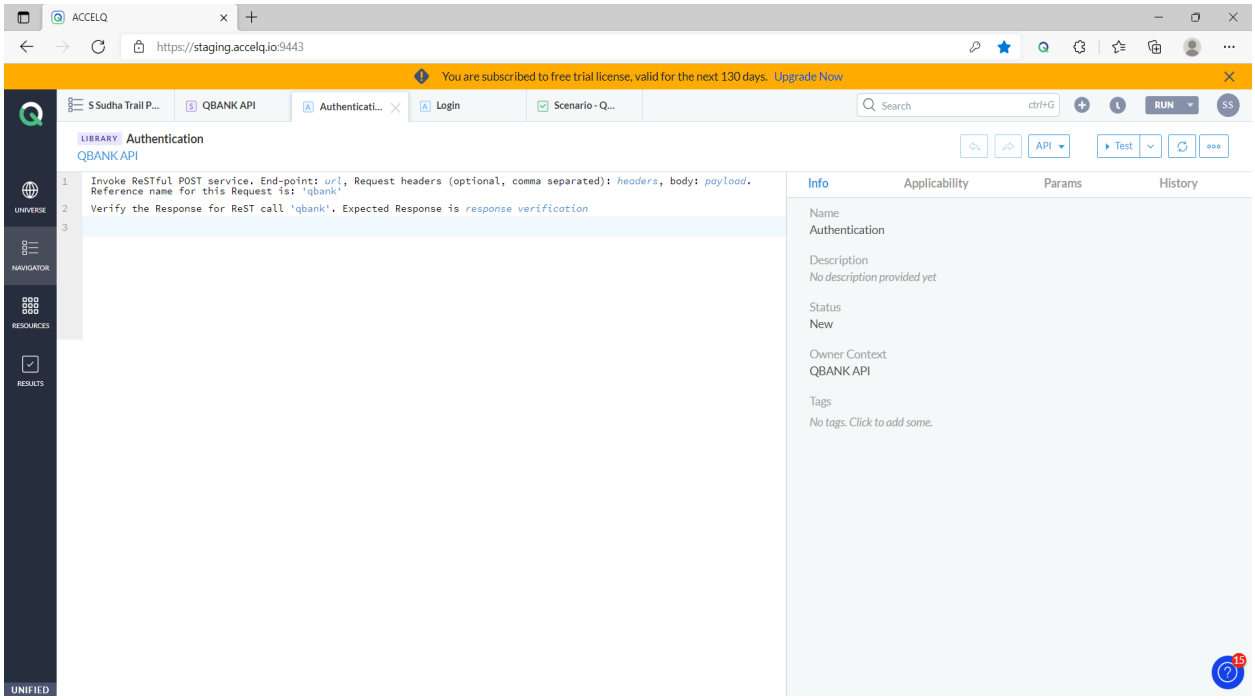
BIBILOGRAPHY

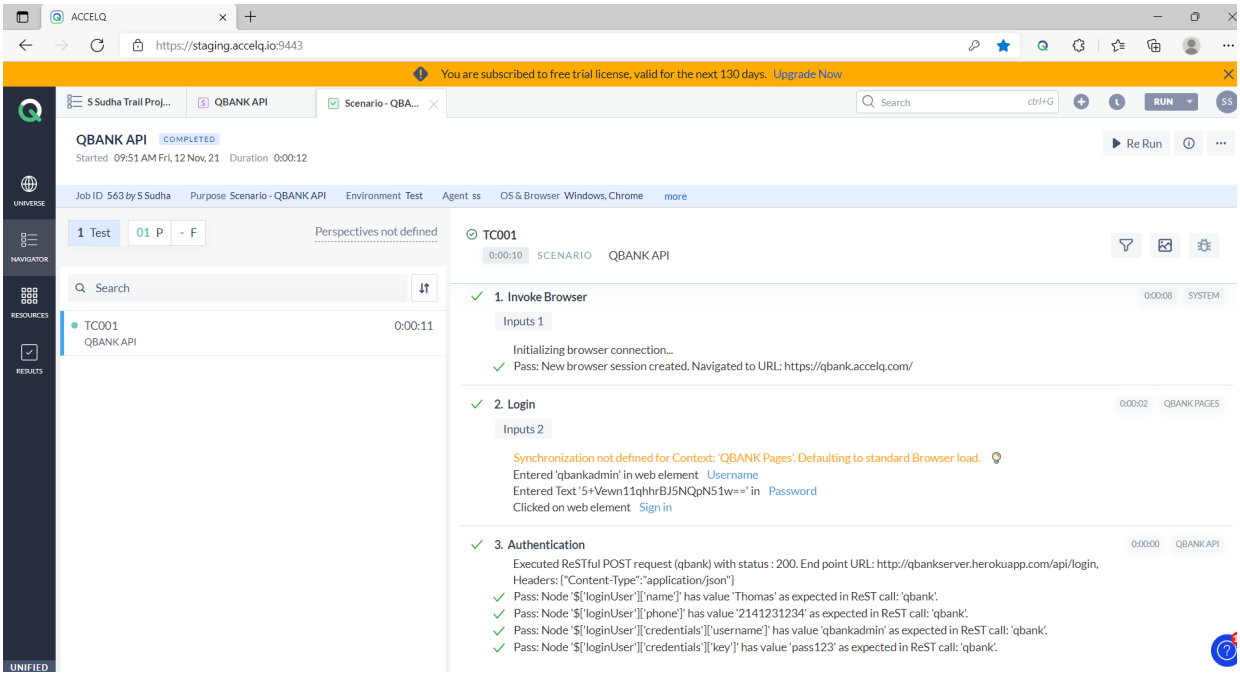
[1] ACCELQ platform
[2] Guidence of Smartinternz
[3] Professional Tutorial to know about REST API
[4] Excellent course provided by smart bridge
[5] ACCELQ platform
[6] References through browser

APPENDIX


ACCELQ
https://staging.accelq.io:9443
You are subscribed to free trial license, valid for the next 130 days. Upgrade Now
S Sudha Trail Pr...
QBANK API
Search
ctrl+G
RUN
QBANK API
Auto Scenario
Run
Test Steps
Test Cases (1)
Results
Traceability
Defects
Steps(3)
Add Step
1 Invoke Browser
System
URL 'https://qbank.accelq.com/'
2 Login
QBANK Pages
Username
Password
3 Authentication
QBANK API
Add Step
UNIVERSE
NAVIGATOR
RESOURCES
RESULTS
UNIFIED
Type here to search
23°C Polluted air 215
ENG
IN
9:50 AM
11/12/2021


ACCELQ
https://staging.accelq.io:9443
You are subscribed to free trial license, valid for the next 130 days. Upgrade Now
S Sudha Trail P...
QBANK API
Authenticati...
Login
Scenario - Q...
Search
ctrl+G
RUN
Login
QBANK Pages › QBANK Pages
API
Test
1 Enter Username in the Username web element
2 Enter Password in the secure web input field, Password
3 Click on Sign in web element
4
Info
Applicability
Elements (7)
Params (2/0)
History
Name
Login
Description
No description provided yet
Status
New
Owner Context
QBANK Pages
Tags
No tags. Click to add some.
3 VIEWS
UNIVERSE
NAVIGATOR
RESOURCES
RESULTS
UNIFIED


ACCELQ
https://staging.accelq.io:9443
You are subscribed to free trial license, valid for the next 130 days. Upgrade Now
S Sudha Trail P...
QBANK API
Authenticati...
Login
Scenario - Q...
Search
RUN
LIBRARY Authentication
QBANK API
API
Test
Invoke ReSTful POST service. End-point: url, Request headers (optional, comma separated): headers, body: payload.
Reference name for this Request is: 'qbank'
Verify the Response for ReST call 'qbank'. Expected Response is response verification
Info
Applicability
Params
History
Name
Authentication
Description
No description provided yet
Status
New
Owner Context
QBANK API
Tags
No tags. Click to add some.
UNIFIED


ACCELQ
https://staging.accelq.io:9443
You are subscribed to free trial license, valid for the next 130 days. Upgrade Now
S Sudha Trail Proj...
QBANK API
Scenario - QBA...
Search
RUN
QBANK API
COMPLETED
Started 09:51 AM Fri, 12 Nov, 21 Duration 0:00:12
Re Run
Job ID 563 by S Sudha
Purpose Scenario - QBANK API
Environment Test
Agent ss
OS & Browser Windows, Chrome
more
1 Test
01 P
- F
Perspectives not defined
Search
TC001
QBANK API
0:00:11
TC001
0:00:10 SCENARIO QBANK API
1. Invoke Browser
0:00:08 SYSTEM
Inputs 1
Initializing browser connection...
Pass: New browser session created. Navigated to URL: https://qbank.accelq.com/
2. Login
0:00:02 QBANK PAGES
Inputs 2
Synchronization not defined for Context: 'QBANK Pages'. Defaulting to standard Browser load.
Entered 'qbankadmin' in web element Username
Entered Text '5+Vewn11qhhrBJ5NQpN51w==' in Password
Clicked on web element Sign in
3. Authentication
0:00:00 QBANK API
Executed ReSTful POST request (qbank) with status : 200. End point URL: http://qbankserver.herokuapp.com/api/login,
Headers: {"Content-Type":"application/json"}
Pass: Node '$['loginUser']['name']' has value 'Thomas' as expected in ReST call: 'qbank'.
Pass: Node '$['loginUser']['phone']' has value '2141231234' as expected in ReST call: 'qbank'.
Pass: Node '$['loginUser']['credentials']['username']' has value 'qbankadmin' as expected in ReST call: 'qbank'.
Pass: Node '$['loginUser']['credentials']['key']' has value 'pass123' as expected in ReST call: 'qbank'.
UNIFIED