

qbank login & authentication using rest api

NAME:-B.muralikrishna
SBID:- SB2021014932

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

In this we are Verifying Qbank login & Authentication using REST API, which provides us access to the data in Qbank Website. The flexibility and scalability of REST API make it an excellent choice for integrating Qbank into the applications and for performing complex operations on a large scale.

Project Flow

1. Create the Context
2. Create the Actions
3. Create the Scenario
4. Run the test case
5. Submit the project

1.2 PURPOSE

To use REST API tools to create, manipulate, and search and access data in Qbank by sending HTTP requests to endpoints in Qbank. Depending on where you send requests, you access and operate on different pieces of information, called resources. Resources

include records, query results, metadata, and more

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

Manual testing approaches are carried out. Some of the testings are:

1. Unit testing
2. Integration testing
3. Functional testing
4. System testing
5. Stress testing
6. Performance testing
7. Usability testing
8. Acceptance testing
9. Regression testing
10. Beta testing

With the Qbank UAT login, testers can carry out test cases designed to examine the business-friendliness and usefulness of Qbank before launching it to all end users.

2.2 PROPOSED SOLUTION

Testing the Qbank login and Authentication using Rest API in ACCELQ- Test Automation tool.

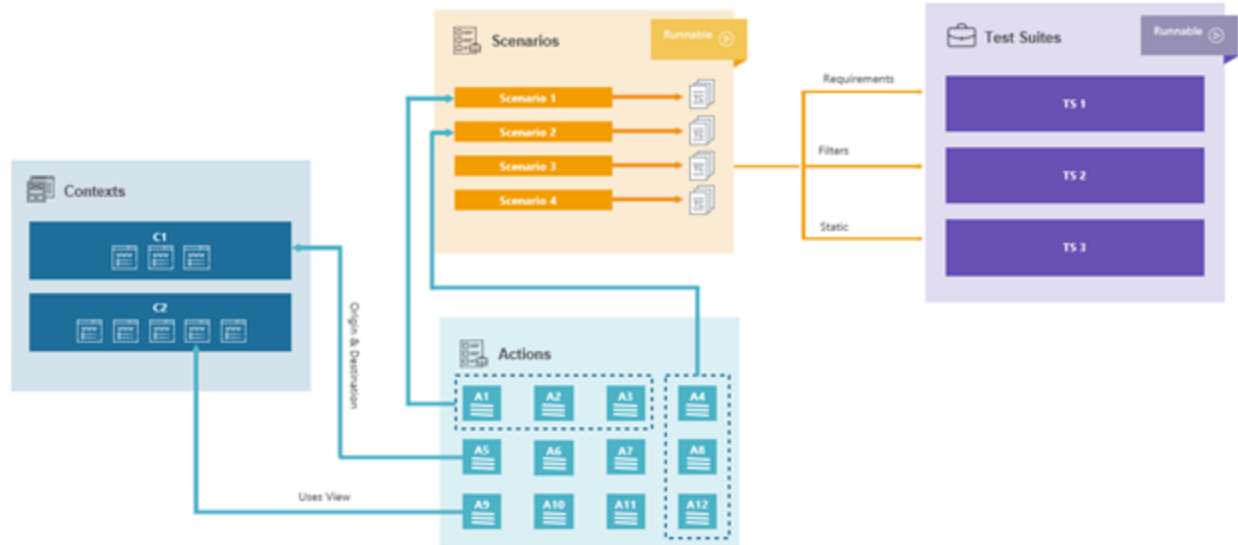
The need for speed and agility in today's consumer-oriented world is paramount to an organizations success, as is the delivery of quality products. This makes it crucial for organizations to adopt test strategies that help them deliver rapidly at the highest quality.

Test automation methodology and framework is well integrated with agile principles and DevOps strategies, which ensures a consistent ROI during the entire software life cycle. The test automation framework is built with a focus on long term utilization and provides great benefits.

CHAPTER 3

THEORITICAL ANALYSIS

3.1 BLOCK DIAGRAM



3.2 HARDWARE / SOFTWARE DESIGNING

Hardware Requirements:

1. 4GB RAM
2. 100 MB Storage

Software Requirements:

1. AccelQ Browser Extension
2. AccelQ Local Agent Installed in Desktop

CHAPTER 4

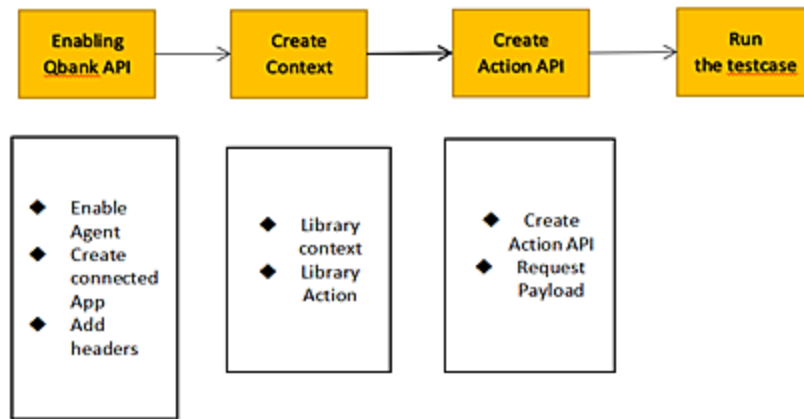
EXPERIMENTAL INVESTIGATIONS

Software quality assurance is in dire need of substantial progress. Software testing is resource-hungry, time-consuming, labor-intensive, and prone to human omission and error. Despite massive investments in quality assurance, serious code defects are routinely discovered after software has been released , and fixing them at so late a stage carries substantial cost . Thorough testing of large, complex software involves great effort, and the software industry still employs relatively primitive testing techniques.

We need a “disruptive technology” to substantially improve software quality. Various studies have found the average bug density in production-ready software to have stayed relatively constant over time, while average code volume of software has increased along an exponential curve, with the net effect that the number of bugs per product is increasing. It is therefore necessary to quickly find a way of reducing bug density by at least an order of magnitude.

CHAPTER 5

FLOWCHART



Creating context

The screenshot displays the ACCELQ web interface. The browser tabs include 'Welcome to ACCELQ - muralicse', 'ACCELQ', 'Student Dashboard', and 'New Tab'. The address bar shows 'staging.accelq.io:9443'. A notification banner at the top states: 'You are subscribed to free trial license, valid for the next 129 days. Upgrade Now'.

The main content area is titled 'verification of qbank using rest api Context'. It features a 'Capture your first View' section with a sub-header 'View is an interactable and actionable screenshot of the page, representing a Context. A given UI Context may have one or more Views depending on variations in visual presentation. You can capture view using accelQ's View Analyzer Plugin.' Below this is a 'Record New View' button and a 'Learn more' link.

The right sidebar contains a table with columns: Info, Elements, Actions (0), and History. The 'Info' tab is active, showing details for the context:

- Name:** verification of qbank using rest api
- Description:** No description provided yet
- Status:** New
- Tags:** No tags. Click to add some.
- SYNCHRONIZATION:** Click to add Sync Element for this Context

The bottom of the interface shows a Windows taskbar with a search bar, application icons, and system information: 28°C, Light rain, 12:47 AM, 11/13/2021.

Creating action

The screenshot displays the ACCELQ web interface. The top navigation bar includes links for 'Welcome to ACCELQ - murali...', 'ACCELQ', 'Student Dashboard', and 'https://qbank.accelq.com'. The main content area shows a 'Response Information' dialog box with tabs for 'Status', 'Headers', and 'Response'. The 'Response' tab is active, displaying a table of response data.

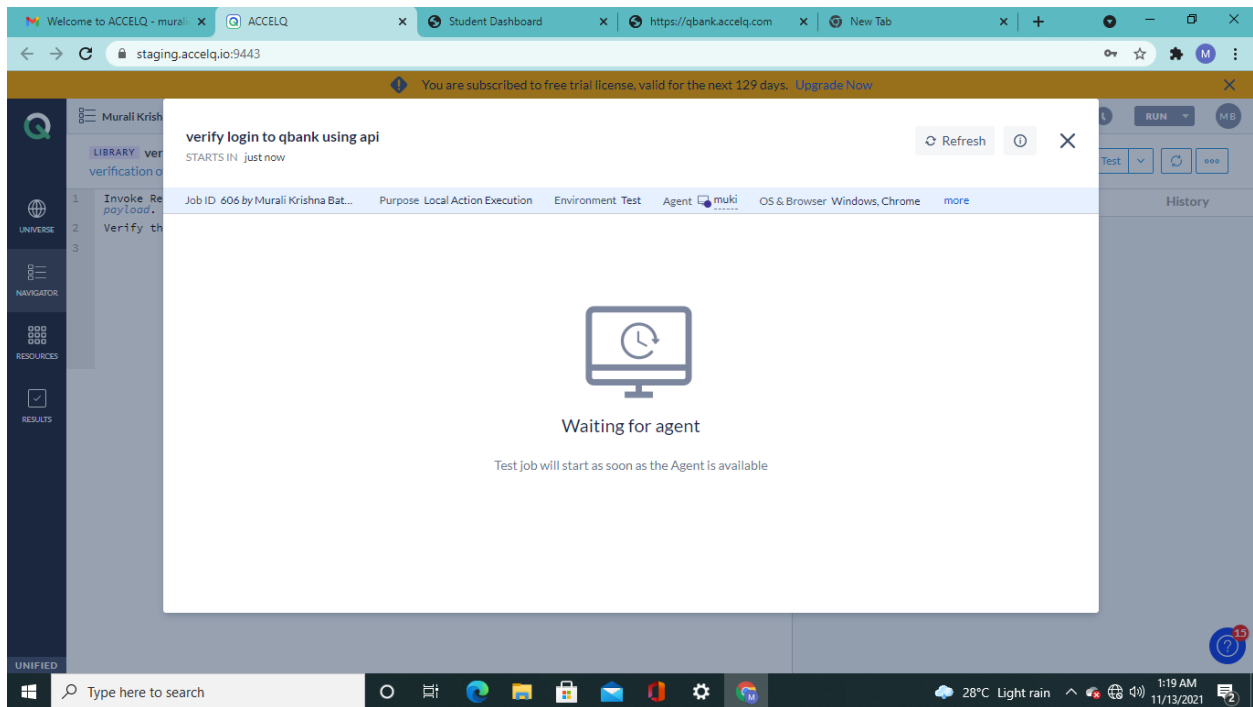
NODE (99)	VALUE
Object	
loginUser	
id	0
name	Thomas
email	thomas@qbank.com
phone	2141231234
country	United States
age	21
credentials	

Below the table, there is a 'Click on a row for JSON path' instruction and a 'CANCEL' button. To the right of the table, there is an 'Add Verification' button. The dialog also includes 'Save Response' and 'Load New' buttons at the top.

The background interface shows a sidebar with 'UNIVERSE', 'NAVIGATOR', 'RESOURCES', and 'RESULTS' sections. The main workspace displays a test case titled 'qbank login using api' with a RESTful API definition. The 'Info' tab is active, showing details about the test case, including its name, description, status, and owner context.

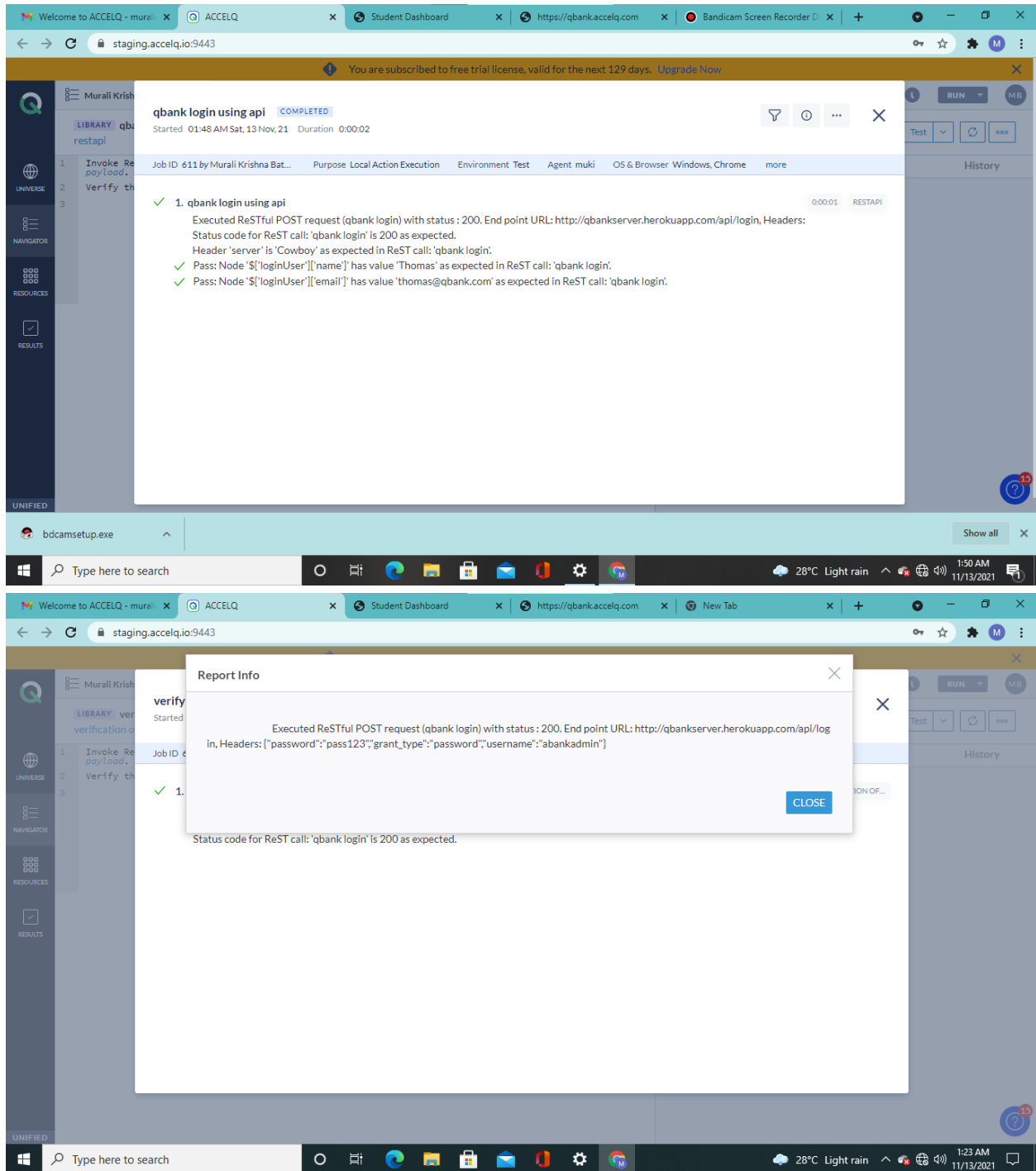
The bottom of the screen shows a Windows taskbar with the search bar, task view button, and several open applications including 'bdcamsetup.exe'.

Run the test cases



CHAPTER 6

RESULT



CHAPTER 7

ADVANTAGES & DISADVANTAGES

ADVANTAGES:

1. Automated testing improves the coverage of testing as automated execution of test cases is faster than manual execution.
2. Automated testing reduces the dependability of testing on the availability of the test engineers.
3. Automated testing provides round the clock coverage as automated tests can be run all time in 24*7 environment.
4. Automated testing takes far less resources in execution as compared to manual testing.
5. It helps to train the test engineers to increase their knowledge by producing a repository of different tests.
6. It helps in testing which is not possible without automation such as reliability testing, stress testing, load and performance testing.
7. It includes all other activities like selecting the right product build, generating the right test data and analyzing the results.
8. It acts as test data generator and produces maximum test data to cover a large number of input and expected output for result comparison.

DISADVANTAGES:

1. Automated testing is very much expensive than the manual testing.
2. It also becomes inconvenient and burdensome as to decide who would automate and who would train.
3. It has limited to some organizations as many organizations not prefer test automation.

4. Automated testing would also require additionally trained and skilled people.
5. Automated testing only removes the mechanical execution of testing process, but creation of test cases still required testing professionals.

CHAPTER 8

APPLICATIONS

1. Automates testing saves time and money.
2. Vastly increases your test coverage.
3. Testing improves accuracy.
4. Automation does what manual testing cannot.
5. It helps developers and testers.
6. QA and Dev team morale improves.

CONCLUSION

Successful automation **mandates a testing process**. Just as a developer needs a system development process, testers need a testing process to successfully use test tools. The testing process provides the steps, guidelines and techniques that will ensure practical, successful automation.

FUTURE SCOPE

Ever since technology is progressing at a speedy pace, the demand for getting projects done quicker has increased more than ever. To get projects done fast, the complete procedures

followed during a software life cycle needs to become accelerated as well.

In the area of software testing, automation can be implemented to save cost and time but only when used in time-taking projects. When it comes to performing regression testing, large scale testing, automation testing is the way to go. It can be a good choice. There are a number of necessary advantages from test automation like Increases the software quality, lessens manual software testing operations and eradicate redundant testing efforts, create extra systematic repeatable software tests, Minimizing repetitive work and generate more consistent testing outcomes, higher consistency.

Execute more **software tests** and accomplish better testing coverage in a very limited schedule. Improves productivity and so on. It is an effective means of eliminating/lessening manual effort during regression as well as functional test case execution. Plus, the chances of defect escape will be decreased considerably, since mistakes will not occur once the script is highly developed. Well, in upcoming days, the automation will be considered more economical considering the fact of quality and time.

BIBLIOGRAPHY

Andrews-06 Mike Andrews, James A. Whitaker, How to Break Web Software: Functional and Security Testing of Web Applications and Web Services, Addison Wesley, New York, 2006, ISBN 0321369440.

Ash-03 Lydia Ash, The Web Testing Companion: Insider's Guide to Efficient and Effective Tests, Wiley, 2003, ISBN 0471430218.

Barr-04 Adam Barr, Find the Bug: A Book of Incorrect Programs, Addison-Wesley Professional, 2004, ISBN 0321223918.

Beck-03 Beck, Kent, Test-Driven Development: By Example, Addison-Wesley, Boston, MA, 2003, ISBN 0-321-146553-0.