==========================**Assignment 37**==============================

Submitted By TANUJA SHARMA

==

Que1. What is API Testing and why Postman is most popular tool for API Testing?

Ans1= API Testing: API Testing is used to validates Application Programming Interfaces (APIs). The purpose of API Testing is to check the functionality, reliability, performance, and security of the programming interfaces. In API Testing, instead of using standard user inputs(keyboard) and outputs, we use software to send calls to the API, get output, and note down the system's response.

Postman: Postman is a collaboration platform for API development. It is a recognized API client that enables us to organize the creation, division, testing, and documentation of APIs.

Postman is most popular tool for API Testing because,

1. API Client feature lets us send REST, SOAP requests directly quickly within Postman.

2. It ensures safe automation and integration of manual tests that guarantee no code breaks.

3. It enables us to design and mock the responses of an API without having to set up a backend server.

4. Machine-readable documentation.

5. Monitor the API performance and response times at regular intervals.

6. The built-in version control lets us share and collaborate in real-time.

Que2. What is the difference between Rest and Soap based API's?

Ans2= The main difference between Rest and Soap based API's are:

SOAP APIs:

- It relies on SOAP (Simple Object Access Protocol).

- It transports data in standard XML format.

- Because it is XML based and relies on SOAP, it works with WSDL

It works over HTTP, HTTPS.

REST APIs:

- It relies on REST (Representational State Transfer) architecture using HTTP.
- Generally transports data in JSON. It is based on URI. Because REST follows stateless model, REST does not enforces message format as XML or JSON etc.
- It works with GET, POST, PUT, DELETE.

It works over HTTP.

Que3. What are the different http methods for API Testing?

Ans3= The different http methods for API Testing are:

GET: GET method is used to retreive data from a server at the specified resource.

POST: POST requests are used to send data to the API server to create or update a resource.

PUT: PUT requests are used to send data to the API to update or create a resource.

HEAD: HEAD requests are useful for checking what a GET request will return before actually making a GET request.

DELETE: The DELETE method is used to delete the resource at the specified URL.

PATCH: It is similar to POST and PUT. The difference with PATCH is that we only apply partial modifications to the resource.

OPTIONS: An OPTIONS request should return data describing what other methods and operations the server supports at the given URL.

Que4. What are the different Status codes available while doing API Testing?

Ans4= HTTP statuses are standard numeric codes indicating whether a specific HTTP request successfully completed.

200: Ok, The request was successful.

201: Created, The request was successful & data was created.

204: No content. The response is empty.

400: Bad request, The request could not be understood or was missing required parameters.

401: Unauthorized, Authentication failed or user does not have permisiions for the requested operation.

403: fORBIDDEN, ACCESS DENIED.

404: nOT FOUND,dATA WAS NOT FOUND.

405: Method not allowed. Requested method is not supported for the specified esources.

500: Internal server error.

503: Service unavailable. The service is temporary unavailable.

Que5. What are the challenges faced in API Testing?

Ans5= The challenges faced in AI testing are:

- The main challenges in Web API testing are Parameter Combination, Parameter Selection, and Call Sequencing
- There is no GUI available to test the application, which makes it difficult to give input values
- Validating and Verifying the output in a different system is a little difficult for testers
- Parameters selection and categorization are required to be known to the testers
- Exception handling function needs to be tested

Coding knowledge is necessary for testers

===================================**Topics**==============================

- **Types of output of an API**
- **Testcases for API testing**
- **Approach of API testing**
- **Difference b/w API testing & Unit testing**
- **How to test API**
- **Best practices of API testing**
- **Types of Bugs that API testing detects**
- **How to do API testing Automation**
- **Challenges of API testing**

==

Types of Output of an API:

An o/p of an API are,

1. **Any type of data:** Let there is an API function that should add two integer numbers. The output should be a summation of two integer numbers.
2. **Status (say Pass or Fail):**
   Like API functions are Lock(), Unlock(), Delete(). They return any value such as True or false as an output.
3. **Call another API function.** we call one of the API function which in turn will call another function.

==

Test Cases for API Testing:

Test cases of API testing are based on,

Return value based on input condition: it is relatively easy to test, as input can be defined and results can be authenticated

Does not return anything: When there is no return value, a behavior of API on the system to be checked

Trigger some other API/event/interrupt: If an output of an API triggers some event or interrupt, then those events and interrupt listeners should be tracked

Update data structure: Updating data structure will have some outcome or effect on the system, and that should be authenticated

Modify certain resources: If API call modifies some resources then it should be validated by accessing respective resources.

==

Approach of API testing: API Testing Approach is a predefined strategy or a method that the QA team will perform in order to conduct the API testing after the build is ready. This testing does not include the source code.

- Understanding the functionality of the API program and clearly defining the scope of the program
- Apply testing techniques such as equivalence classes, boundary value analysis, and error guessing and write test cases for the API
- Input Parameters for the API need to be planned and defined appropriately

Execute the test cases and compare expected and actual results.

==

Difference between API testing and Unit testing:

API testing:

- Testers perform it.
- End-to-end functionality is tested.
- Testers cannot access the source code.
- Only API functions are tested.
- All functional issues are tested.
- Broader in scope.

Test run after the build is created.

Unit testing:

- Developers perform it.

- •Separate functionality is tested.
- •A developer can access the source code.
- •UI testing is also involved.
- •Only basic functionalities are tested.
- •Limited in scope.

Usually ran before check-in.

==

How to Test API:

API automation testing should cover following testing methods are:

Discovery testing: The test group should manually execute the set of calls documented in the API like verifying that a specific resource exposed by the API can be listed, created and deleted as appropriate.

Usability testing: This testing verifies whether the API is functional and user-friendly. And does API integrates well with another platform as well.

Security testing: This testing includes what type of authentication is required and whether sensitive data is encrypted over HTTP or both.

Automated testing: API testing should culminate in the creation of a set of scripts or a tool that can be used to execute the API regularly.

Documentation: The test team has to make sure that the documentation is adequate and provides enough information to interact with the API. Documentation should be a part of the final deliverable.

==

Best Practices of API Testing:

- •API Test cases should be grouped by test category.
- •On top of each test, we should include the declarations of the APIs being called.
- •Parameters selection should be explicitly mentioned in the test case itself.
- •Prioritize API function calls so that it will be easy for testers to test.
- •Each test case should be as self-contained and independent from dependencies as

possible.

- Avoid “test chaining” in our development.

- Special care must be taken while handling one-time call functions like – Delete, CloseWindow, etc.

- Call sequencing should be performed and well planned.

To ensure complete test coverage, create API test cases for all possible input combinations of the API.

==

Types of Bugs that API testing detects:

- Fails to handle error conditions gracefully

- Unused flags

- Missing or duplicate functionality

- Reliability Issues. Difficulty in connecting and getting a response from API.

- Security Issues

- Multi-threading issues

- Performance Issues. API response time is very high.

- Improper errors/warning to a caller

- Incorrect handling of valid argument values

Response Data is not structured correctly (JSON or XML)

==

How to do API Test Automation:

(By using Postman)

Postman is a collaboration platform for API development. It is a recognized API client that enables us to organize the creation, division, testing, and documentation of APIs.

Features of Postman:

- Postman is most popular tool for API Testing because,

- API Client feature lets us send REST, SOAP requests directly quickly within Postman.
- It ensures safe automation and integration of manual tests that guarantee no code breaks.
- It enables us to design and mock the responses of an API without having to set up a backend server.
- Machine-readable documentation.
- Monitor the API performance and response times at regular intervals.
- The built-in version control lets us share and collaborate in real-time.

(By using ReadyAPI)

ReadyAPI is a leading tool for functional, security, and load testing of RESTful, SOAP and other web services. In one platform we will get three powerful tools: ReadyAPI Test, ReadyAPI Performance, and ReadyAPI Virtualization. With these tools you can perform functional, security, and performance/load testing.

Features of ReadyAPI:

- ReadyAPI can be integrated into any environment.
- It has a Smart Assertion feature that can create bulk assertions against hundreds of endpoints quickly.
- Native support for Git, Docker, Jenkins, Azure, etc.
- Supports Command-line for automated testing.
- Supports parallel execution of functional tests and job queuing.
- Promotes code reuse

Removes dependencies during testing and development.

===

Challenges of API Testing:

- The main challenges in Web API testing are Parameter Combination, Parameter Selection, and Call Sequencing
- There is no GUI available to test the application, which makes it difficult to give input values

- Validating and Verifying the output in a different system is a little difficult for testers

- Parameters selection and categorization are required to be known to the testers

- Exception handling function needs to be tested

Coding knowledge is necessary for testers

===