Contents

1 Where would you start? What would be your first steps?

1.1 Entry Criteria for Testing

- Defined and Approved Requirements
- Approved Test Plan
- The readiness of Test Cases and Test Data
- Testable Code with Appropriate Test Environment
- Executing the primary functional/business flows successfully by leveraging various test inputs and ensuring that they are working fine
- Access to sufficient and desired test data
- Setting up of test environment with all the necessary resources like tools and devices
- Spot checks to ensure all the preconditions are met, and eradicate any defects or tasks that are delaying the process timelines

1.2 Every QA team should fallow the phases of STLC:

- Requirement Analysis
- Test Planning
- Test case development
- Test Environment setup
- Test Execution
- Test Cycle closure

2. Which process would you establish around testing new functionality? How would you want the features to be tested?

API Testing

USER

1. **Presentation Tier** – User Interface (UI)
2. **Logic Tier** – Business logic is written in this tier. It is also called Business Tier. (API)
3. **Data Tier** – Here information and data is stored and retrieved from a Database. (DB)

2.1 Identify types of tests to be performed: - API testing we need to fallow the types of testing like

- **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
- **Gather details about testing priorities and focus.**
- **Prepare Requirement Traceability Matrix (RTM).**
- **Identify test environment details where testing is supposed to be carried out.**
- **Automation feasibility analysis (if required).**

2.2 Entry Criteria for API Testing

1. Understanding the functionality of the API program and clearly define the scope of the program
2. Apply testing techniques such as equivalence classes, boundary value analysis, and error guessing and write test cases for the API
3. Input Parameters for the API need to be planned and defined appropriately
4. Execute the test cases and compare expected and actual results.

2.3 Set-up of API Test environment

- ✓ API Testing required to setup initial environment that invokes API with a required set of parameters and then finally examines the test result.
- ✓ Database and server should be configured as per the application requirements.
- ✓ Once the installation is done, the API Function should be called to check whether that API is working

2.4 Basic Smoke testing criteria for API Testing.

- ✓ Verify if the Test environment is available and ready for use.
- ✓ Verify if test tools installed in the environment are ready for use.
- ✓ Verify if Testable code is available.
- ✓ Verify if Test Data is available and validated for correctness of Data.

2.5 API testers should have in mind to start the testing.

- ✓ Who is your target audience? Who is your API consumer?
- ✓ What environment/s should the API typically be used?
- ✓ What aspects are you testing?
- ✓ What problems are we testing for?
- ✓ What are your priorities to test?
- ✓ What is supposed to happen in normal circumstances?
- ✓ What could potentially happen in abnormal circumstances?
- ✓ What is defined as a Pass or a Fail? What data is the desired output? What is the chain of events?
- ✓ What other APIs could this API interact with?
- ✓ Who on your team is in charge of testing what?

2.6 API Testers should focus on bellow areas

Functionality testing — The API works and does exactly what it's supposed to do.
Reliability testing — The API can be consistently connected to and lead to consistent results
Load testing — The API can handle a large amount of calls
Creativity testing — The API can handle being used in different ways.
Security testing — The API has defined security requirements including authentication, permissions and access controls. See some API security tips for protecting vital data
Negative Testing — checking for every kind of wrong input the user can possibly supply.

2.7 These are common API test examples

- For complete test coverage, create test cases for all possible API input combinations
- Perform well-planned call sequencing
- Checking API return values based on the input condition
- Verifying if the API doesn't return anything at all or the wrong results
- Verifying if the API triggers some other event or calls another API
- Verifying if the API is updating any data structures.
- Verifying loads by throwing as much as you can at it and see how it handles unforeseen problems and verify.
- Test for failure. Make sure you understand how your API will fail. Just make sure the API fails consistently and gracefully
- Make sure your APIs work across devices, browsers, and operating systems

2.8 There are mainly 4 methods involve in Rest API Testing like GET, POST, Delete, and PUT.

- **GET**- The GET method is used to extract information from the given server using a given URI. While using GET request, it should only extract data and should have no other effect on the data.
- **POST**- A POST request is used to create a new entity. It can also be used to send data to the server, for example, customer information, file upload, etc. using HTML forms.
- **PUT**- Create a new entity or update an existing one.
- **DELETE**- Removes all current representations of the target resource given by a URI.

3. Which tools would you suggest using to help your team with a daily work?

we can do it using simple tools like Bellow

3.1 Best tools

JMeter: - JMeter includes functionalities like test an API ,load and Performance testing, JMeter can automatically work with CSV files, which allows your teams to quickly create unique parameter values for your API tests. It also integrates with Jenkins, which means you can include your API tests in your CI pipelines.
Postman: - Postman is an easy-to-use REST client, Postman is also a nice option for exploratory-type API testing
SoapUI: - SoapUI is a fully functional test tool dedicated to API testing
REST-Assured: - When we using JAVA REST-Assured is best tool for API Automation. REST-Assured is a fluent Java library you can use to test HTTP-based REST services. It's designed with testing in mind, and it integrates with any existing Java-based automation framework.

KarateDSL: - It's a new tool for BDD Framework (I don't' have experience on KarateDSL)

4. If you would do a test automation which techniques or best practices would you use the application?

Test driven development :-is a software development practice which describes the pattern of writing tests before implementing software.
Behaviour Driven Development (BDD):- is the process which combines ideas of domain driven design and test-driven development process (TDD).
Rest Testing Framework: -The framework should be able to execute the basic REST operations (GET, POST, PUT, PATCH, DELETE) and perform the validations on the code, message, headers and body of the response.