API Testing Tutorial: What is API Test Automation? How to Test

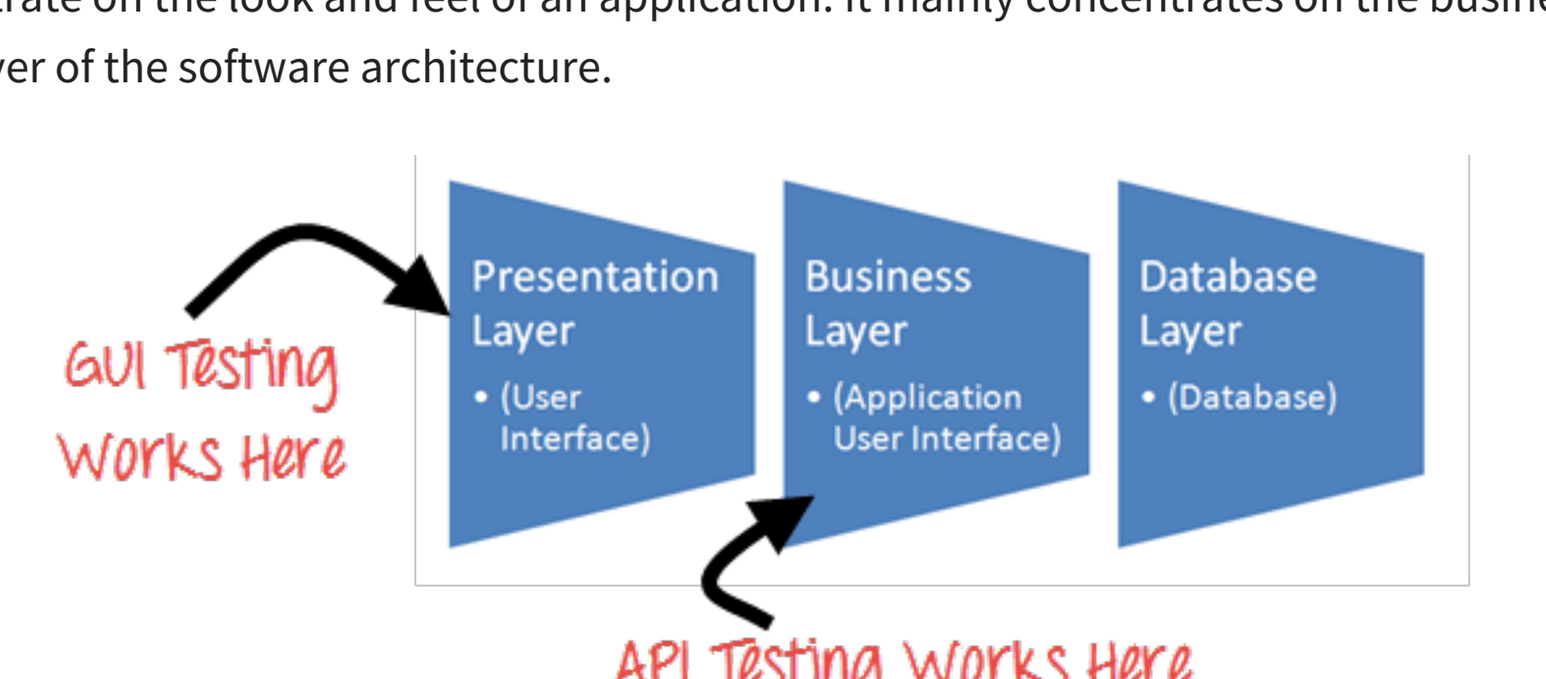
Before going to API Testing tutorial, let's first understand

What is API?

API (Application Programming Interface) is a computing interface which enables communication and data exchange between two separate software systems. Software system that executes an API includes several functions/subroutines that another software system can perform. API defines requests that can be made, how to make requests, data formats that can be used, etc. between two software systems.

What is API Testing?

API TESTING is a software testing type that 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, you use software to send calls to the API, get output, and note down the system's response. API tests are very different from GUI Tests and won't concentrate on the look and feel of an application. It mainly concentrates on the business logic layer of the software architecture.



API automation Testing requires an application that can be interacted via an API. In order to test an API, you will need to

- Use Testing Tool to drive the API
- Write your own code to test the API

In this API testing tutorial, you will learn more about-

Set-up of API Test environment

- API Testing is different than other software testing types as GUI is not available, and yet you are required to setup initial environment that invokes API with a required set of parameters and then finally examines the test result.
- Hence, Setting up a testing environment for API automation testing seems a little complex.
- 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.

Types of Output of an API

An output of API could be

1. Any type of data
2. Status (say Pass or Fail)
3. Call another API function.

Let's look at an example of each of the above Types in this api testing tutorial

Any Type of Data

Example: There is an API function which should add two integer numbers.

```
Long add(int a, int b)
```

The numbers have to be given as input parameters. The output should be a summation of two integer numbers. This output needs to be verified with an expected outcome.

Calling needs to be done such as

```
add (1234, 5656)
```

Exceptions have to be handled if the number is exceeding the integer limit.

Status (say Pass or Fail)

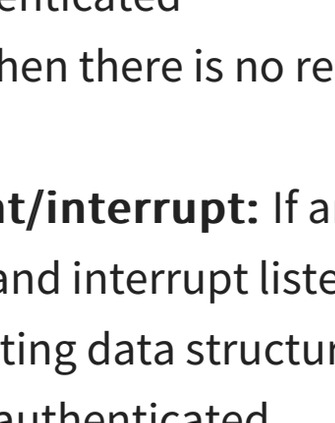
Consider the below API function -

1. Lock()
2. Unlock()
3. Delete()

They return any value such as True (in case of success) or false (In case of error) as an output.

A more accurate Test Case would be, can call the functions in any of the scripts and later check for changes either in the database or the Application GUI.

Calling of another API / Event



In this case, we call one of the API function which in turn will call another function.

For example - First API function can be used for deleting a specified record in the table and this function, in turn, calls another function to REFRESH the database.

Test Cases for API Testing:

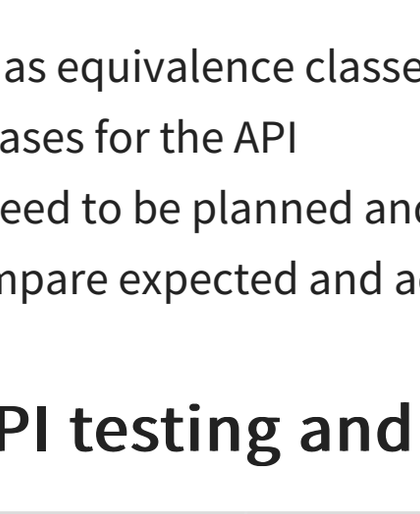
Test cases of API testing are based on

- **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

API Testing Approach

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. The API testing approach helps to better understand the functionalities, testing techniques, input parameters and the execution of test cases.

Following points helps the user to do API Testing approach:



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.

Difference between API testing and Unit testing

Unit testing	API testing
• Developers perform it	• Testers perform it
• Separate functionality is tested	• End to end functionality is tested
• A developer can access the source code	• Testers cannot access the source code
• UI testing is also involved	• Only API functions are tested
• Only basic functionalities are tested	• All functional issues are tested
• Limited in scope	• Broader in scope
• Usually ran before check-in	• Ran after build is created

How to Test API

API automation testing should cover at least following testing methods apart from usual SDLC process

- **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, you 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 your 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

Following tutorials provide a detailed guide to automate API test.

How to test API with REST Assured

How to test API with Postman

How to test API with UFT

Besides there are other tools for API testing. Check them here

Challenges of API Testing

Challenges of API testing includes:

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

Conclusion:

API consists of a set of classes/functions/procedures which represent the business logic layer. If API is not tested properly, it may cause problems not only the API application but also in the calling application. It is an indispensable test in software engineering.

Prev Report a Bug Next

Testing Tutorials