

API Testing

Topics:

1. What is an API
2. What is API Testing
3. API Testing Types
4. Common tests on APIs
5. Advantages of API Testing
6. What exactly needs to be verified in API Testing
7. Tools used for API Testing
8. Challenges in API testing
9. API Testing Best Practices

What is an API?

API is an acronym and it stands for **A**pplication **P**rogramming **I**nterface. API is a set of routines, protocols, and tools for building Software Applications. APIs specify how one software program should interact with other software programs.

Routine: a program that performs a particular task. Routine is also known as procedure, function, or subroutine.

Protocols: A format for transmitting data between two systems.

In simple words, API stands for **A**pplication **P**rogramming **I**nterface. API acts as an interface between two software applications and allows the two software applications to communicate with each other. API is a collection of software functions that can be executed by another software program.

Let's see some examples of an API in a more approachable way.

Assume an **API** as a **Waiter** at a **Restaurant**.

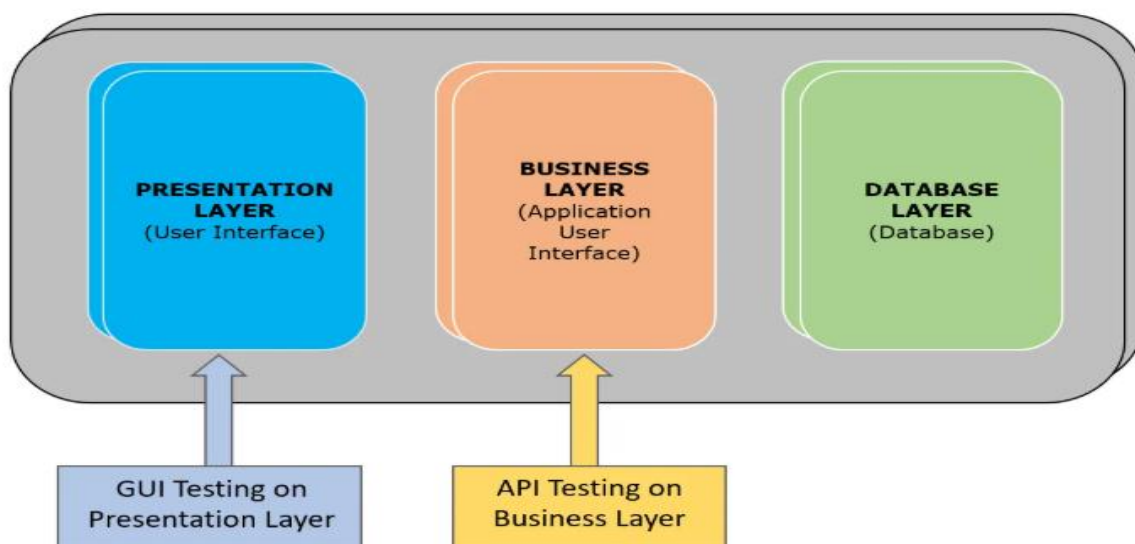
At a restaurant, you give an order based on the items available on the menu. A waiter in the restaurant writes down your order and delivers it to the kitchen who prepares your meal. Once the meal is ready, the waiter picks up your food from the kitchen and serves it to you at your table.

In this scenario, the waiter's role is similar to an API. As a waiter, the API takes a request from a source, takes that request to the database, fetches the requested data from the database, and returns a response to the source.

Now let's see another example.

If you are using a flight service engine say Expedia, where you search for flights on a specific date. Once you pass the data such as Source, Destination, Onward Date, and Return Date and click on search. Expedia sends a request to airlines through an API as per your search details. The API then takes the airline's response to your request and delivers it right back to Expedia.

What is API Testing?



API testing is a type of software testing that involves testing APIs directly and also as a part of integration testing to check whether the API meets expectations in terms of functionality, reliability, performance, and security of an application. In API Testing our main focus will be on a Business logic layer of the software architecture. API testing can be performed on any software system which contains multiple APIs. API testing won't concentrate on the look and feel of the application. API testing is entirely different from GUI Testing. Let's see how is UI testing is not similar to API testing?

UI (User Interface) testing is to test the graphical interface part of the application. Its main focus is to test the look and feel of an application. On the other hand,

API testing enables the communication between two different software systems. Its main focus is on the business layer of the application.

API Testing Types?

API testing typically involves the following practices:

- **Unit testing:** To test the functionality of individual operation
- **Functional testing:** To test the functionality of broader scenarios by using a block of unit test results tested together
- **Load testing:** To test the functionality and performance under load
- **Runtime/Error Detection:** To monitor an application to identify problems such as exceptions and resource leaks
- **Security testing:** To ensure that the implementation of the API is secure from external threats
- **UI testing:** It is performed as part of end-to-end integration tests to make sure every aspect of the user interface functions as expected
- **Penetration testing:** To find vulnerabilities of an application from attackers
- **Fuzz testing:** To test the API by forcibly input into the system in order to attempt a forced crash

Common tests on APIs:

Some of the common tests we perform on APIs are as follows.

- To verify whether the return value is based on the input condition. The response of the APIs should be verified based on the request.
- To verify whether the system is authenticating the outcome when the API is updating any data structure
- To verify whether the API triggers some other event or request another API
- To verify the behavior of the API when there is no return value

Advantages of API Testing:

- API Testing is time effective when compared to GUI Testing. API test automation requires less code so it can provide faster and better test coverage.
- API Testing helps us to reduce the testing cost. With API Testing we can find minor bugs before the GUI Testing. These minor bugs will become bigger during GUI Testing. So, finding those bugs in the API Testing will be cost-effective to the Company.
- API Testing is language independent.

- API Testing is quite helpful in testing Core Functionality. We can test the APIs without a user interface. In GUI Testing, we need to wait until the application is available to test the core functionalities.
- API Testing helps us to reduce the risks.

What exactly needs to be verified in API Testing?

Basically, on API Testing, we send a request to the API with the known data and we analyse the response.

- Data accuracy
- HTTP status codes
- Response time
- Error codes in case API return any errors
- Authorization checks
- Non-functional testing such as performance testing, security testing

Tools used for API Testing:

Some of the tools used for API Testing are as follows:

- Postman
- Katalon Studio
- SoapUI
- Assertible
- Tricentis Tosca
- Apigee
- JMeter
- Rest-Assured
- Karate DSL
- API Fortress
- Parasoft
- HP QTP(UFT)
- vREST
- Airborne
- API Science
- APlary Inspector
- Citrus Framework
- Hippie-Swagger
- HttpMaster Express
- Mockbin
- Ping API

- Pyresttest
- Rest Console
- RoboHydra Server
- SOAP Sonar

Challenges in API testing:

Some of the challenges we face while doing API testing are as follows

- Selecting proper parameters and its combinations
- Categorizing the parameters properly
- Proper call sequencing is required as this may lead to inadequate coverage in testing
- Verifying and validating the output
- Due to absence of GUI it is quite difficult to provide input values

Types of bugs we face when performing API testing:

Issues observed when performing API testing are

- Stress, performance, and security issues
- Duplicate or missing functionality
- Reliability issues
- Improper messaging
- Incompatible error handling mechanism
- Multi-threaded issues
- Improper errors

API Testing Best Practices:

- Test for the expected results
- Add stress to the system by sending series of API load tests
- Group API test cases by test category
- Create test cases with all possible input's combinations for complete test coverage
- Prioritize API function calls to make it easy to test
- Create tests to handle unforeseen problems
- Automate API testing wherever it is possible