

Learn API testing in 10 minutes!

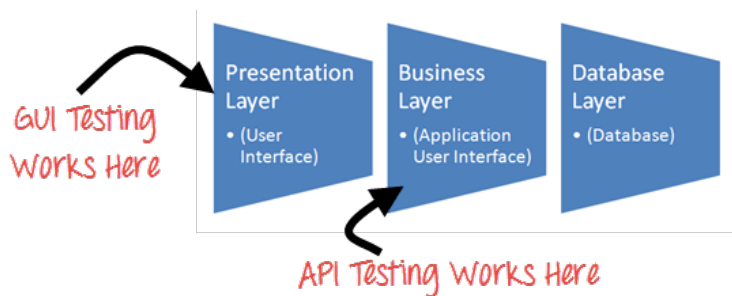
Before going to Api Testing, let's first understand

What is an API?

API is an acronym for Application Programming Interface.

It enables communication and data exchange between two separate software systems. A software system implementing an API contains functions/sub-routines which can be executed by another software system.

What is API testing?



Api Testing is entirely different from [GUI Testing](#) and mainly concentrates on the business logic layer of the software architecture. This **testing won't concentrate on the look and feel of an application.**

Instead of using standard user inputs(keyboard) and outputs, in Api Testing, you use software to send calls to the API, get output, and note down the system's response.

Api Testing requires an application to interact with 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

Set-up of API Test environment

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

Testing Tutorials

[66\) Create Test Strategy Document](#)

[67\) Testing as a Service](#)

[68\) Test Maturity Model](#)

[69\) Defect Severity](#)

[70\) Test Plan V/s Test Strategy](#)

[71\) Static Vs Dynamic](#)

[72\) Retesting Vs Regression Testing](#)

[73\) Quality Assurance Vs Quality Control](#)

[74\) Verification v/s Validation](#)

[75\) Positive Vs Negative](#)

[76\) Test Harness](#)

[77\) Defect Density](#)

[78\) Globalization Vs Localization](#)

[79\) Test Scenario Vs Test Condition](#)

[80\) Unit Test Vs Integration](#)

[81\) Software Testing Type](#)

[82\) API testing](#)

[83\) Black Box Testing](#)

[84\) Alpha Beta](#)

- Once the installation is done, API Function should be called to check whether that API is working.

Testing

85) Functional

Testing

86) UAT

Types of Output of an API

Output of API could be

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

Let's look at an example of each of above Types

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 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 script 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, behavior of API on the system to be checked
- **Trigger some other API/event/interrupt:** If 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:

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
<ul style="list-style-type: none">• Developers perform it	<ul style="list-style-type: none">• Testers perform it
<ul style="list-style-type: none">• Separate functionality is tested	<ul style="list-style-type: none">• End to end functionality is tested
<ul style="list-style-type: none">• Developer can access the source code	<ul style="list-style-type: none">• Testers cannot access the source code
<ul style="list-style-type: none">• UI testing is also involved	<ul style="list-style-type: none">• Only API functions are tested
<ul style="list-style-type: none">• Only basic functionalities are tested	<ul style="list-style-type: none">• All functional issues are tested
<ul style="list-style-type: none">• Limited in scope	<ul style="list-style-type: none">• Broader in scope
<ul style="list-style-type: none">• Usually ran before check-in	<ul style="list-style-type: none">• Ran after build is created

What to test for in API testing

API testing should cover atleast 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:

- Test cases should be grouped by test category
- On top 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 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 caller
- Incorrect handling of valid argument values
- Response Data is not structured correctly (JSON or XML)

Tools for API testing

Since API and [Unit Testing](#) both target source code, similar tools can be used for testing both.

- [SOAPUI](#) - It's an easy-to-use tool with extensive REST and SOAP API testing capabilities – no scripting experience is required.
- Runscope
- Postman with jetpacks
- Postman with Newman
- Curl
- Cfix
- Check
- CTESK
- dotTEST
- Eclipse SDK tool- Automated API testing

Check out top API Testing Tools [List](#)

Challenges of API Testing

Challenges of API testing includes:

- Main challenges in 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 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.

[< Prev](#)[Report a Bug](#)[Next >](#)

YOU MIGHT LIKE:

Software Testing



What is SOAK Testing? Learn with Examples

What is Soak Testing? Soak Testing is a non-functional testing in which, the System under Load...

[Read more](#)

levels

Orthogonal Array Testing (OAT) Tutorial: Tools & Techniques

In the present scenario, delivering a quality software product to the customer has become...

[Read more](#)

Software Testing



Storage Testing in Software Testing

What is Storage Testing? Storage testing is a type of testing which verifies whether the...

[Read more](#)

Software Testing



What is Response Time Testing?

What is Response Time? Response time refer to the time takes for one system node to respond to the...

[Read more](#)

Software Testing



What is Embedded Testing in Software Testing?

In this tutorial, you will learn [What are Embedded systems? What is Embedded Software Testing? ...](#)

[Read more](#)

Software Testing



Configuration Testing Tutorial: Sample Test Cases & Example

What is Configuration Testing? Configuration testing is the method of testing an application with...

[Read more](#)



About

[About US](#)
[Advertise with Us](#)
[Write For Us](#)
[Contact US](#)

Career Suggestion

[SAP Career Suggestion Tool](#)
[Software Testing as a Career](#)
[Certificates](#)

Interesting

[Books to Read!](#)
[Suggest a Tutorial](#)
[Blog](#)
[Quiz](#)
[Review](#)

Execute online

[Execute Java Online](#)
[Execute Javascript](#)
[Execute HTML](#)
[Execute Python](#)

Top Tutorials



Selenium



Testing



SAP



Java



Jmeter



Informatica

© Copyright - Guru99
2018

[Privacy Policy](#)