**Types of API Testing**

**Unit Testing:** For testing the functionality of individual operation.

**Functional Testing:** To make sure that the API endpoints are up and working and doing what exactly they are supposed to do.

**Reliability Testing:** making sure that the API’s works in case of connecting it to various devices and don’t get disconnected any time.

**Load Testing:** When various servers sends request to an API, it is necessary to make sure that the API responds to all of them.

**Stress testing**: When more than set no of requests is received by the API how does it behaves? Does it send some message?

Works as intended. Mandatory to check.

**Security testing:** While giving authentication, it is important to make sure that no security breaches happen in between. No more than required data is shared. Have appropriate authentications, permissions and access controls.

**Integration testing:** All the APIs connected to each other communicate properly and addition of any feature in the API do not cause addition of some bugs in other API modules.

**Usability testing:** The API is functional and top of it, user friendly.

**Error Detection:** For identifying any errors such as exceptions and resource leaks.

**An API is made out of two parts:**

The actual implementation of the API functionality. This is the core of the API – The thing people use the API for.

* If your API translates text between languages, this would be the code to actually do the translation.
* If your API modifies images, this would be the actual program doing the image modifications.

**The interface** – the interface would include the actual endpoints people interface with.

There are few things the interface should take care of:

* Authentication: You want to make sure people sign up to your API, and make requests with a valid key.
* Limiting: You want to limit how many requests each developer (eg: API key). Can make your API.
* Billing: If your API is paid, you want to charge people based on the volume of usage.

**How to do API Testing**

API 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:**

* 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 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)

**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

**API Testing Approach**

Quality Assurance team performs API testing which is a form of Black Box Testing. This testing is conducted post the build is ready.  The source code is not included in this testing.

In this testing, a request is sent to the API with known to analyze the response that includes:

* Accuracy of data
* HTTP status code
* Response time
* Error codes of any errors returned by API
* Authorization checks
* Results of non-functional tests such as performance, security, etc.

**API Testing Guidelines-Best practices**

* Test the API to check what happens consistently and what doesn’t.
* Perform stress testing on the system through a series of API load tests.
* Test the API for failures; until you get the output as failed. Test the API so that it fails consistently.
* Group the test cases by test category.
* Mention the parameters selected in the test case itself.
* Prioritize API function calls to simplify testing for testers so that they can finish testing on time.
* Automate the API documentation creation process and ensure a good level of documentation is there which is easy to understand.
* Plan to perform call sequencing.
* Create test cases for all possible API input combinations to get complete test coverage.
* Reuse test cases and monitor the API in production.
* Depend on manual and automated tests for better API testing outcomes.

<https://docs.optimizory.com/display/vrest/Writing+your+first+test+case>

1. Safe method: GET
2. Unsafe method: POST, PUT, DELETE