**Generic Test Cases for API Testing**

**While API test cases will differ based on your requirement and application functionality, there are a few general steps that you will probably find in every test case for API testing. Here is a series of generic test cases for you to understand better.**

* **Validate the API keys for minimum and maximum range/length**
* **Verify that these keys are answering the call requests if you are using JSON or XML APIs in your application**
* **Create a separate test case to validate XML and JSON Schema**
* **Verify the Parse response to parse out portions of the response data returned from the server**
* **Check the JSON Schema validation, the Field Type, and the Mandatory Fields**
* **Validate the Response headers and Negative Testcases response**
* **Identify and verify the handling of API error codes**
* **Verify the HTTP response and its code status**
* **Validate Response payload to determine their format and readability factor**
* **Test request chaining to identify the working of multiple APIs together**
* **Verify APIs with input parameters.**
* **Validate end-to-end CRUD (create, read, update, and delete) flow for application API**
* **Check database integrity test cases**
* **Verify file upload test cases**

**Sample API Testing Test Case template https://testsigma.com/blog/test-cases-for-api-testing/**

**Applications and websites use multiple API calls at every stage. One of the prime examples of an API call is performing the login functionality using third-party services, such as Google, Yahoo!, or Facebook. Whenever you choose to sign into an application or a website using your Google details rather than within the app option, you**[**trigger an API request**](https://developers.google.com/apis-explorer)**. This implies the online platform has an existing agreement with Google to access the information already supplied and available to them. Such API calls require proper testing and validation before being released to users.**

**We are taking the same example to write a sample test case for API testing for testing purposes, which is going to test the APIs in the backend.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl.No** | **Test Case Title** | **Priority** | **Severity** | **Steps** | **Expected Result** | **Actual Result** | **Status** |
| 1 | Checking Google login functionality using valid users | High | High | 1. Navigate to the application/website login page 2. Click on the Google icon present on that page for login 3. A prompt will appear. Select ‘Proceed’ or ‘Confirm’ the Google username for signing in 4. The application should allow the process and take you to the dashboard/homepage | Log in process successful | Fill based on observation | Pass/Fail |
| 2 | Checking Google login functionality using invalid users | High | High | 1. Navigate to the application/website login page 2. Click on the Google icon present on that page for login 3. A prompt will appear. Select ‘Add a New Account’ to add another Google account to sign in 4. Add a Google account with the existing username but wrong password 5. A prompt will appear. Select ‘Proceed’ or ‘Confirm’ the Google username for signing in | The application should not allow you to process the request and must ask you to sign up using the same Google account before logging in | Fill based on observation | Pass/Fail |
| 3 | Checking Google login functionality for new users | High | High | 1. Navigate to the application/website login page 2. Click on the Google icon present on that page for login 3. A prompt will appear. Select ‘Add a New Account’ to add another Google account to sign in 4. Add a new Google account 5. Go to the sign up page of the application and sign up using the new Google account details 6. A prompt will appear. Select ‘Proceed’ or ‘Confirm’ the Google username for signing in | The application should allow you to process the request and direct you to the dashboard/homepage | Fill based on observation | Pass/Fail |

**Let’s take another example of validating the Response headers, which is mentioned in the above section. The header part consists of ‘metadata,’ including the Content-Type attribute that makes us understand how to interpret the data of the response body. Thus, if the response body consists of JSON data, the content-type attribute in the header will be application/json. And, if there is XML data in the body is XML, the content-type header will be application/xml. You can run the API URL on an automation tool or a framework to execute the test cases and receive the response.**

**Automating your API Testing**

**Up until now, we were primarily discussing manual API testing, and the template also contains test cases to execute the steps. But there are ways to automate your API testing test cases to save a considerable amount of time and money. Not only that, but automating test cases for API testing will also decrease accuracy issues by a huge percentage and direct your employees’ manual labor toward other productive business elements. One primary aspect to be mindful of is: when to make the transition from manual to automation testing. It is advised to switch when your test cases are frequently executed and take too much of time in manually executing.**

**There are numerous**[**API testing tools**](https://testsigma.com/api-testing-tools)**that support verifying the APIs and collecting relevant output information to extend your QA efforts without giving it much time.**

**Benefits of Automating API Testing**

**There are multiple benefits of automating API testing test cases besides saving time, money, and energy:**

* **Automation offers better test coverage**
* **Allows for faster bug fixes**
* **Offers quick response time**