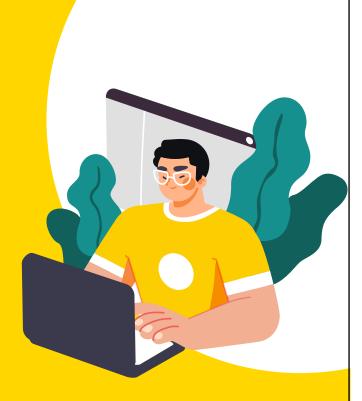
### **ACCELQ**

# **API TESTING**

WHAT, WHY, and HOW?

**Application Programming Interface (API) enables** two or more computer programs to interact with each other and exchange data between them without any user intervention.



### What are the **Benefits** of API Testing?

It helps to validate and qualitatively measure the success of business requirements through a defined test case process.

|  | Early Evaluation |
|--|------------------|
|--|------------------|

- **CI/CD** Compliance
- **Tight Security**
- **Platform Agnosticism**

All these aspects serve as examples of the relevance of API testing, especially in reducing the risk of delivering an application that is not yet ready for the market.

# **Challenges in API** Testing

- Initial Setup of API Testing 0
- Sequencing of API Calls
- Selecting, Testing, and Validating the Parameter Combinations



### **Initial Setup of API Testing**

- It's different from other software testing environments as there's no GUI available
- The database and server must be configured per the application's requirements.
- It's different from other software testing environments as there's no GUI available

### **Sequencing of API Calls**

The API calls may not appear in the sequence expected. This is especially a problem with multi-threaded apps. Haphazard call sequencing could lead to confusion as the tester will not be able to check if the system is working correctly.

### Selecting, Testing, and Validating the **Parameter combinations**

APIs assign data values to parameters. These parameters pass through the data requests to manage communication between the systems. Sometimes the API would assign two different values to the same parameter. The QA team has to test all the possible parameter combinations in API to check for any issues in the specific configurations.

### **Updating the Schema of API Testing**

The QA team needs to ensure that any change in the program that creates an additional parameter for API calls reflects in the schema configuration.

# **How To Approach API Testing?**



6

#### Set Up the Environment and **Integrate Application Data**

Configure the database and server according to the application requirements, and integrate application data with the API tests to test the API against all configurations.

#### **Define the Scope** of Testing

3

5

This includes asking critical questions like what endpoints are available for testing, what should the error message display in the event of an unsuccessful request, what is the workflow of the application, etc.

#### **Apply Various Testing Techniques**

Perform various testing techniques such as functionality testing, load testing, and security testing to ensure that the API works efficiently. Each serves a specific purpose, so test thoroughly before sending to production.

#### **Develop Test Cases and Analyze the Results**

Write and execute test cases and check if the actual results match the expected results.

#### **Choose the Right API Testing Tool**

Discuss the different tools with developers and the QA team, analyze their pros and cons, check the programing language they are based on, and evaluate if they fit the budget.

#### **Document Maintenance**

When the test cases are executed, document the results for future reference. Also, make it a point to update the document regularly in case there are changes in endpoints or the backend impacting the API.

# **Best Practices of API** Testing

- Develop a flowchart to help the QA team visualize the API calls.
- Test for both positive and negative results.
- Perform API testing for all API input combinations to widen the scope of test coverage.
- Use data that would reflect the conditions the API would face in the production phase.



- Use service virtualization to create a simulated environment and test as many test cases, scenarios, and endpoints as possible.
- Perform both manual and automated API tests to produce more trustworthy results.
- Automate the documentation creation process to ensure that all the test cases are documented.

## **Examples of API** Testing

- Testing if a mobile app can pull out the necessary information about the user from Google or Facebook.
- Testing the payment integration in a mobile app.

# Why **ACCELQ** for **API Test Automation?**



- Speed up the continuous delivery cycle by reducing the dependency on manual testing.
- Facilitate end-to-end API validation through chain API tests.
- Empower testing engineers to make more informed decisions on the run.
- Maintain your standards and respond to business requirements rapidly.
- Reduce the cost of development and maintenance of the codebase with a lean testing strategy