

API Testing 101: Best Practices With Practical Examples

API Testing 101: Best Practices With Practical Examples

API testing involves verifying that an API complies with specified requirements. This objective poses a considerable challenge, with various approaches available for testers and developers to execute it proficiently.

To streamline the process, we've outlined ten best practices for API testing. However, let's briefly outline a pragmatic approach to API testing without going nuts about it:

👉 Automate Wherever Possible

By automating repetitive tasks, such as data management and verification processes, teams can execute tests more rapidly and consistently. This not only accelerates the testing lifecycle but also enhances the reliability of results.

Example: Overall Test Planning Automation

- *Without Automation:*

```
# Manual test planning
execute_test_case("Scenario 1")
execute_test_case("Scenario 2")
```

- *With Automation:*

```
# Automated test planning
execute_all_test_cases()
```

👉 Run your tests in parallel for speed

Executing API tests in parallel is a powerful strategy to enhance testing speed and efficiency. Instead of running tests sequentially, parallel execution allows multiple tests to run simultaneously.

Example: Consider an API with multiple endpoints. Running tests in parallel can involve executing tests for different endpoints concurrently:

```
$ pytest test_endpoint_1.py & pytest test_endpoint_2.py & pytest test_endpoint_3.py
```

This approach significantly reduces the overall testing time, making it a **valuable strategy for teams working on continuous integration and deployment pipelines**.

👉 Keep your testing as simple as possible

Maintaining simplicity in API testing is a fundamental principle that can significantly enhance the efficiency and effectiveness of the testing process.

Guidelines for Simplifying API Testing:

Feature	Complex Approach	Simple Approach
Focused Test Objectives	Testing multiple functionalities in a single test case	Break down test cases to focus on specific functionalities
Clear Verification Points	Elaborate verification steps with unnecessary details	Clearly defined and focus on essential verification points
Minimal Dependencies	Intertwining tests with dependencies on external factors	Minimize dependencies to ensure tests remain independent and reproducible

👉 Use the Right API Testing Tool

The choice of tools can significantly impact the efficiency and effectiveness of the testing process. Factors like API type compatibility, ease of integration, and support for advanced testing features should guide the selection process.

While selecting an API automation tool, these must be the focus points:

- It should include no or very little manual coding effort.
- The tool should focus on end-to-end scenario testing, and should be self-sufficient.
- It should save some time by automatically taking care of all the documentation and schemas.
- It should be compatible with testing all major types of APIs i.e., GraphQL, gRPC, SOAP and REST.

Based on all these key points, we've already created a list of top 10 best performing API testing tools, check that out [here](#).

That concludes the brief overview. For a comprehensive understanding of the best practices for API testing, you can access the complete set of guidelines [here](#).