



## EFFECTIVE API TEST AUTOMATION

### Abstract

“API Testing Market size was valued at USD 937.51 Million in 2021 and is projected to reach USD 5,328 Million by 2030, growing at a CAGR (Compound Annual Growth Rate) of 20.5% from 2022 to 2030.”

The growth corresponds to increased organizations adopting cloud applications and platforms that require API's (Application Programming Interfaces). The rapid adoption of modern technologies makes the systems more complex and the need to ensure the APIs (application programming interfaces) work as expected is even more relevant. Various aspects like functional, performance, and security need to be considered to ensure complete API testing which is where automation comes into play. This paper explains the types of API testing, challenges, need for automation and choosing appropriate tools. It also explains the areas and use cases where API test automations can come in handy to reduce effort and cost with a complete shift-left approach as in case of 5G network and IOT (Internet of Things) world.

## Introduction

APIs are everywhere today from mobiles, websites to IOT devices. Many APIs are used in all spaces including retail markets, healthcare, food industry, gaming consoles etc. Choosing the right API testing strategy for automation is even more relevant today.



## What is API testing?

API testing involves sending requests and validating the response thereby testing the business logic. API allows the application to communicate with other applications. API testing allows the tester to validate the functionality even before the actual UI (User Interface) is ready. This is key in the shift-left approach where issues can be identified early in application development lifecycle.

## Types of API testing

Type of testing	Description
Functional testing	This would validate the response for a particular API.
Performance testing	This type of testing would be done to see how the API handles large volume of requests
Security testing	This type of testing is done to validate if the API is secure enough with proper authentication mechanisms and how it handles sensitive data
Fuzz testing	This is a type of random testing by sending incorrect inputs and validating whether the API is responding with correct error messages
Validation testing	This type of testing is done at a later stage to see if the API meets the business requirements

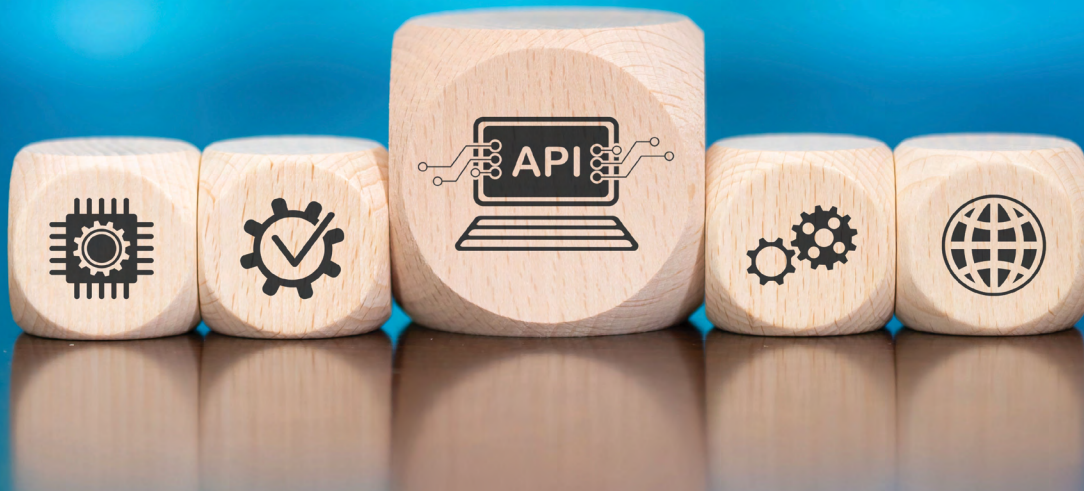
## Challenges in API testing and need for automation



- **No UI involved** – Since the API testing does not involve any UI, it makes it even more difficult to test the APIs. We need to go through the API specification to understand the functionality, required input parameters and the expected response.

- **Complexity in API input data** – Usually the API input data is overly complex, which involves many variables. The input format can be JSON/XML depending on the type of API. Identifying the static and variable fields is a challenge.
- **Test Coverage** – Ensuring that the test cases cover all scenarios including positive and negative cases becomes a challenge. It is important to validate how the API handles invalid input data and error scenarios.
- **Right sequencing of APIs** – Testing the individual API behavior would not be enough to validate the application functionality. Calling the APIs in the right sequence is the key for end-to-end testing.
- **System Integration** – Testing individual APIs do not give a complete picture. API dependencies also needs to be considered during the design phase itself to ensure the entire system operates reliably.
- **Lack of standardization** – Since there are no standards defined for system integration, the APIs could be built on different technologies which makes it even more complex to test.

All the above challenges stress the need for automation in API testing. Creating test cases ensuring maximum coverage and all the required input data is often challenging. Manually validating the API response is time consuming. Maintenance of the test scripts can take away a lot of time from other important work.



## Preparations for API testing

### 1. Understand the API specification

Usually, the API specification would describe the endpoints, method type, query and path variables, input parameters, expected output and authentication mechanisms. OpenAPI is the most common specification for REST APIs. The other formats include YAML and JSON. Below is the OpenAPI specification for sample Pet store application.

Figure 1. Sample OpenAPI specification

```

openapi: 3.0.3
info:
  title: Swagger Petstore - OpenAPI 3.0
  description: |-
  termsOfService: http://swagger.io/terms/
  contact:
    email: apiteam@swagger.io
  license:
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
  version: 1.0.11
externalDocs:
  description: Find out more about Swagger
  url: http://swagger.io
servers:
  - url: https://petstore3.swagger.io/api/v3
tags:
paths:
  /pet:
    put:
      tags:
        - pet
      summary: Update an existing pet
      description: Update an existing pet by Id
      operationId: updatePet
      requestBody:
        description: Update an existent pet in the store
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/Pet'
          application/xml:

```

Source: <https://editor.swagger.io/>

### 2. Study the input and output parameters

Proper understanding of input and output parameters is necessary to create test cases covering all positive and negative scenarios. In the case of Rest API, the inputs can be headers, path or query variables and request body. Below is the sample request body for pet store application.

Figure 1. Sample OpenAPI specification

POST /pet Add a new pet to the store

**Parameters**

Name	Description
<b>body</b> <sup>* required</sup> object (body)	Pet object that needs to be added to the store  Example Value   Model <div style="background-color: #333; color: #eee; padding: 10px; margin-top: 10px;"> <pre> {   "id": 0,   "category": {     "id": 0,     "name": "string"   },   "name": "doggie",   "photoUrls": [     "string"   ],   "tags": [     {       "id": 0,       "name": "string"     }   ],   "status": "available" } </pre> </div>

Parameter content type

Source: <https://petstore.swagger.io/>

Understanding the API's expected output is also important in creating the test cases. Test cases should also test how the API handles error conditions with invalid or missing inputs.

### 3. Scope the testing requirements

Understanding the application and its related workflows is important in deciding the scope of API testing. Equally important is studying the feature/functionality of the API. The interaction between APIs should also be considered. Below are some of the points to look for while scoping the API testing.

- o Response time – The acceptable time for the API to respond
- o Response code – API should respond with the correct status code
- o Output data – Validating whether the API returns the desired output data
- o Definition of pass/fail for test cases – Clearly defining when a test case should be considered as passed.

### 4. Choose the right API test automation tool

An API test automation tool can help to simplify the entire testing process. Each tool will have its pros and cons, the following factors should be considered while choosing the right automation tool

- Kind of APIs under test
- Ease of Use
- Pricing
- CI/CD integrations

Some of the commonly available market tools for API testing are: -

**Postman** –With its ease of use and integrations it is one of the most popular tools used for API testing. Maintaining the API collections and working as a team becomes difficult since Postman runs on local machines.

**ReadyAPI** – This tool offers rich UI and integrations with CI/CD tools but comes with a licensing cost.

**REST Assured** – This is a java library for testing REST based services. It requires strong coding skills to effectively use this library for API testing.

### Infosys Offering for API testing

**Infosys API Test Automation (IATA)** enables organizations to exercise a broader range of simplified and script-less test conditions and scenarios for API testing. The tool offers intuitive UI for creating test cases and validations. Mock APIs can also be created for early testing.



## Benefits of API testing

- **Quick Releases** – API tests take lesser time when compared to UI tests. API testing can be started early since it does not require the UI to be ready which results in quicker releases.
- **Reduced Costs** – Once the API tests are automated, the executions are faster which helps to identify any issues earlier. Early identification and fixing helps to reduce costs.
- **Improved Test Coverage** – Automated API tests helps to cover maximum application area by ensuring all the systems are working as intended including servers, databases.
- **Greater stability and low maintenance** – API test cases are easy to maintain since changes in the API layer are rare. Also, with the availability of detailed API specification any changes can be incorporated into the test cases easily as and when required.

## API Testing Best Practices

- API should be well defined.
- Clearly understand the APIs, define the scope and keep API tests simple and structured.
- Include tests for performance and security aspects of API in addition to functional.
- Consider API virtualization in case the API is not available due to various reasons.
- Keep API tests up to date based on the API specification
- Keep the test data dynamic as much as possible. Test the API for all input values including valid and invalid data
- Use a standard well-defined framework for API testing.
- Usage of automation tool ensures maximum coverage and reduced testing time.

## API testing Use cases

Below are some of the use cases for API testing in emerging areas

Use case	Description	Sample Scenarios
Application APIs	Any application whose business logic is exposed as APIs	End to End application workflow testing Validating individual functionalities
RESTful APIs of main network functions in the 3GPP 5G core network	The core network relies on a service-based architecture. New APIs can be added to the centralized repository and exposed	Testing and validating the network function APIs
APIs in IoT (Internet of Things)	APIs play a significant role in IoT space. APIs do the data collection, analysis and respond to the device appropriately	Validating the APIs interacting with devices in healthcare space, monitoring devices in automotive space

## Conclusion

With increased services being used and added in all the spaces including mobile apps, retail stores, healthcare devices and 5G, the need to ensure the APIs are working as expected is unavoidable. To release products faster, API testing is an important phase and by automating the same we can ensure the APIs are tested regularly and working as expected. Understanding the APIs and choosing the right tool is an important strategic decision.

## References

1. <https://blog.hubspot.com/website/api-testing>
2. <https://www.verifiedmarketresearch.com/product/global-api-testing-market-size-and-forecast-to-2025/>
3. <https://www.altexsoft.com/blog/api-testing/>

## Authors



**Chitra Subramanian**  
Technology Lead



**Jitty Joseph**  
Senior Technical Manager



## Mentor



**Vijendra Singh Rajput**  
Principal Technology Architect



For more information, contact [askus@infosys.com](mailto:askus@infosys.com)



© 2023 Infosys Limited, Bengaluru, India. All Rights Reserved. Infosys believes the information in this document is accurate as of its publication date; such information is subject to change without notice. Infosys acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. Except as expressly permitted, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording or otherwise, without the prior permission of Infosys Limited and/ or any named intellectual property rights holders under this document.