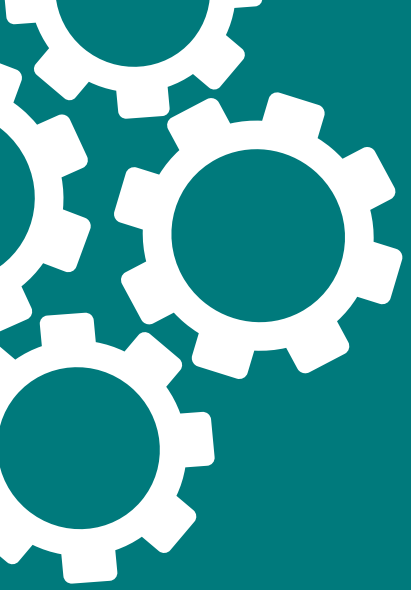


Introduction to API Testing

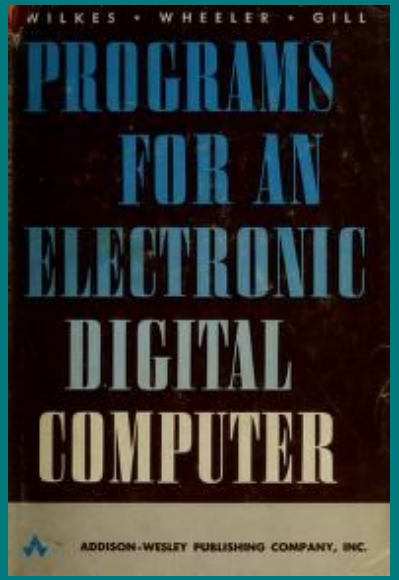
01

What is API?

What is API?

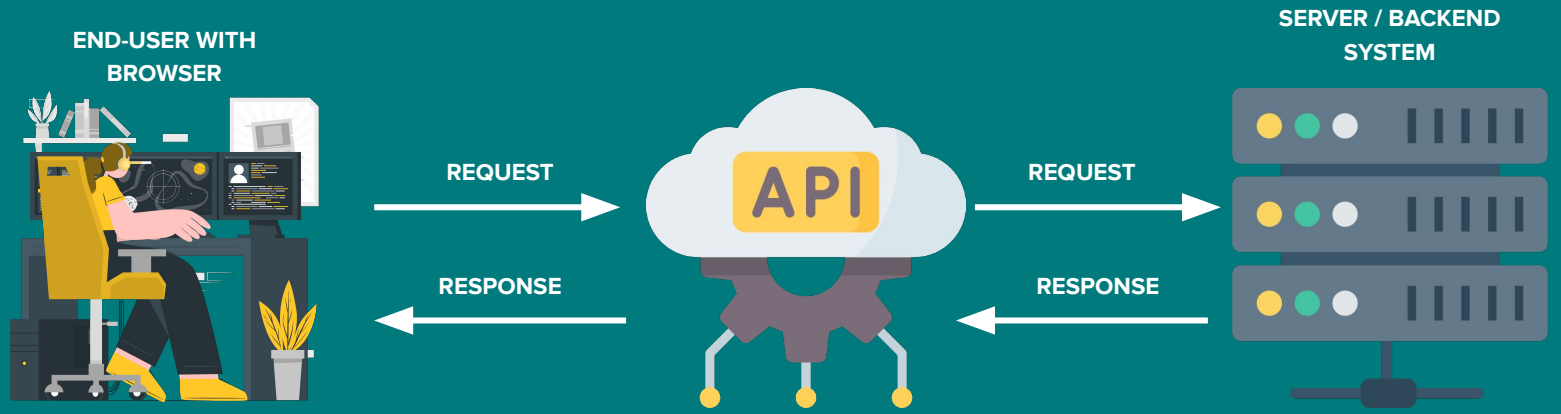
API = Application Programming Interface

API is a set of routines, protocols and tools for building Software Application.
The working principle of an API is commonly expressed through the request-response communication between a client and a server. The client is any front-end application that a user interacts with. The server is in charge of backend logic and database operations. In this scenario, an API works as a middle layer between the client and the server, making it possible to send data requests and responses.



The term of API was first mentioned in a 1951

What is API?



What is API?

1. Search
2. Create a new cart
3. Add a product to cart
4. Place the order

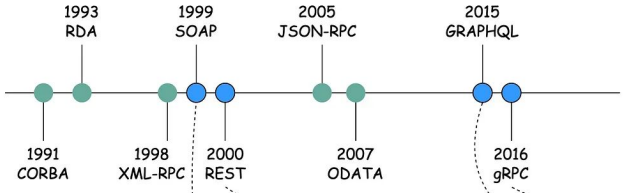
02

LET'S TAKE A LOOK AT API IN MORE DETAIL

TYPES OF API PROTOCOLS

API Architectural Styles Comparison

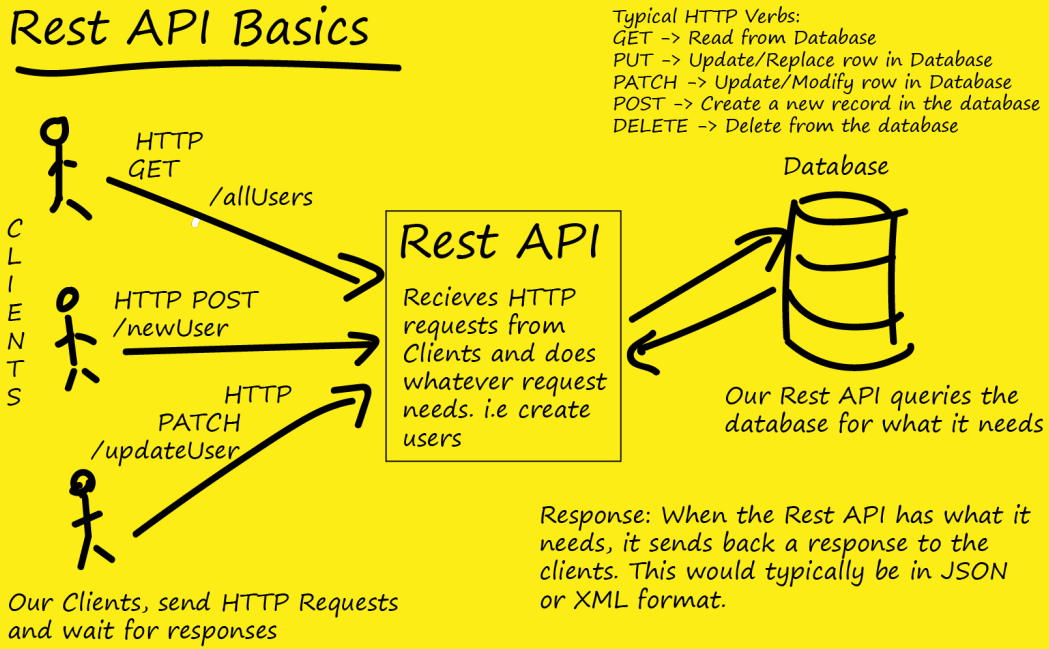
Source: altexsoft



	SOAP (Simple Object Access Protocol)	REST (REpresentational State Transfer)	GraphQL	RPC (Remote Procedure Call)
Organized in terms of	enveloped message structure	compliance with six architectural constraints	schema & type system	local procedure call
Format	XML only	XML, JSON, HTML, plain text	JSON	JSON, XML, Protobuf, Thrift, FlatBuffers
Learning curve	Difficult	Easy	Medium	Easy
Community	Small	Large	Growing	Large
Use cases	- payment gateways - identity management - CRM solutions - financial and telecommunication services - legacy system support	- public APIs - simple resource-driven apps	- mobile APIs - complex systems - micro-services	- command and action-oriented APIs - high performance communication in massive micro-services systems

Source: https://blog.bytebytego.com/p/soap-vs-rest-vs-graphql-vs-rpc

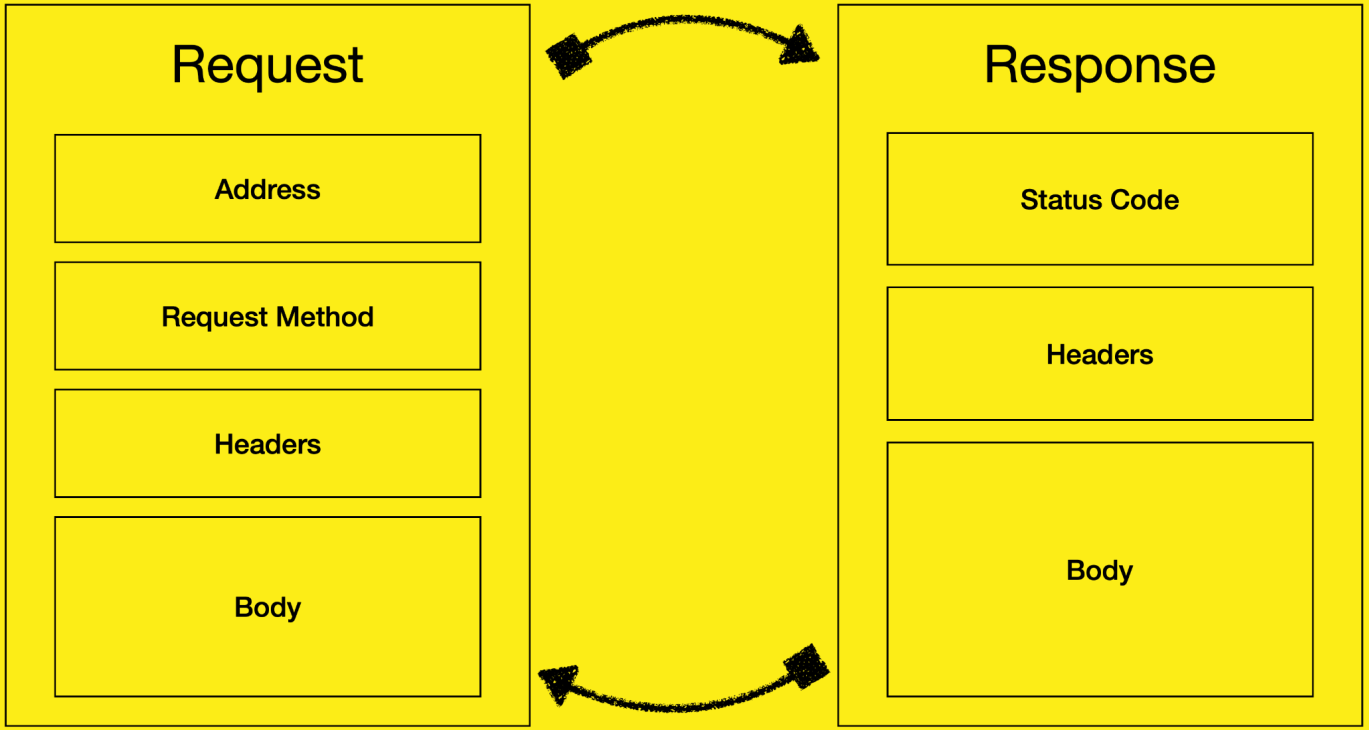
REST API



These are the general steps for any REST API call:

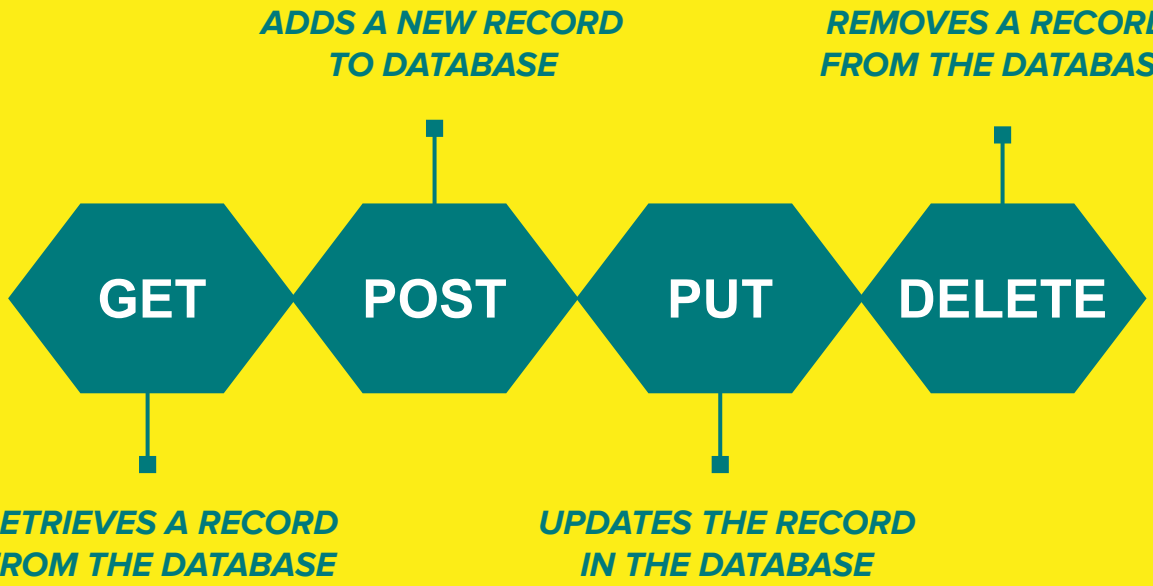
1. The client sends a request to the server. The client follows the API documentation to format the request in a way that the server understands.
2. The server authenticates the client and confirms that the client has the right to make that request.
3. The server receives the request and processes it internally.
4. The server returns a response to the client. The response contains information that tells the client whether the request was successful. The response also includes any information that the client requested.

REQUEST VS. RESPONSE



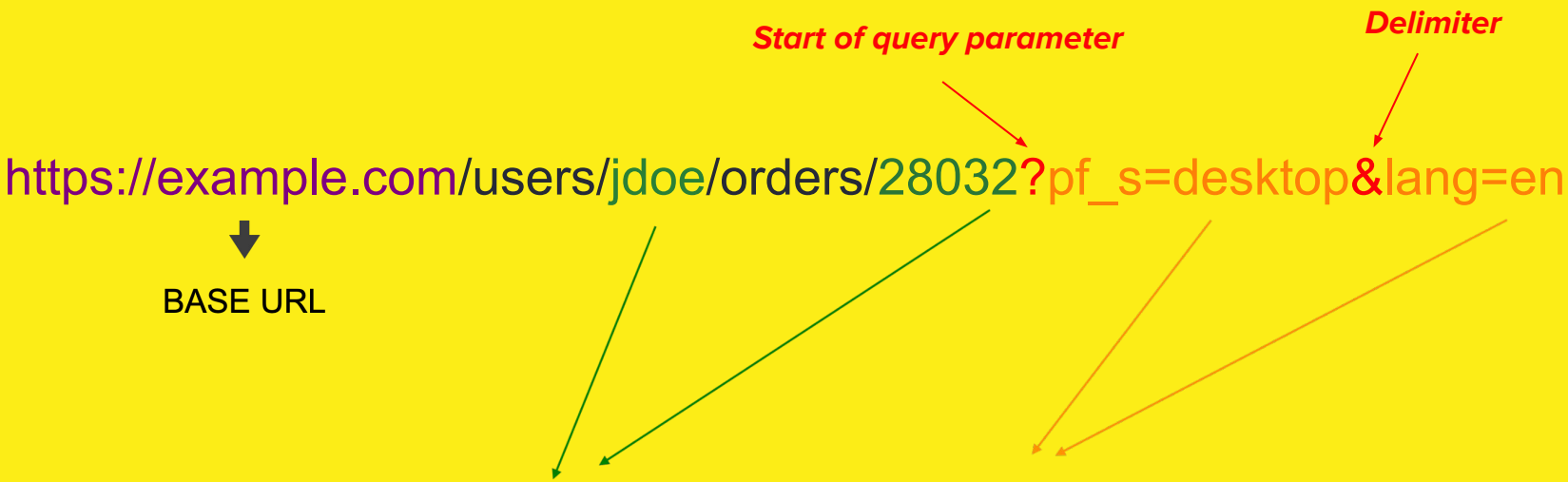
HTTP METHODS

Request methods are the actions that the client wants to perform on the server resource



WHAT DOES URL CONSIST OF?

WHAT DOES URL CONSIST OF?



Path variables	Query parameters
Only value	Key=value pair
Mandatory	Mandatory or optional
Part of the endpoint / path	Start after the question mark

WHAT ARE THE HEADERS?

WHAT ARE THE HEADERS?

Headers are used to provide additional information for a server to process an API request.

Headers are logically grouped into three categories:

- ***request headers,***
- ***response headers***
- ***general headers***

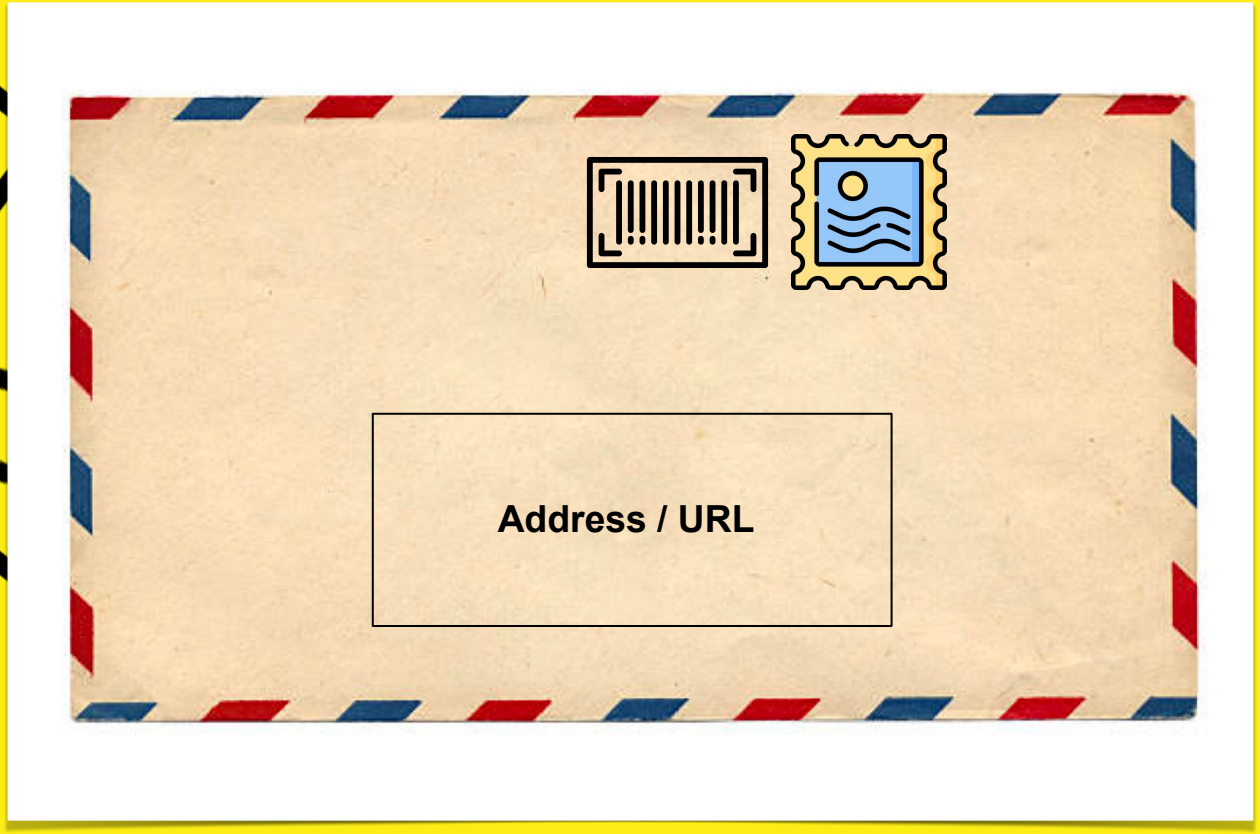
Some commonly used headers in REST API calls include:

- **Accept**: Specifies the content type that the client expects to receive in response to the request.
- **Content-Type**: Specifies the media type of the request body.
- **Authorization**: Specifies the authentication token or credentials for the request.
- **User-Agent**: Specifies the software client making the request.
- **Cache-Control**: Specifies caching behavior for the request and response.

WHAT ARE THE HEADERS?

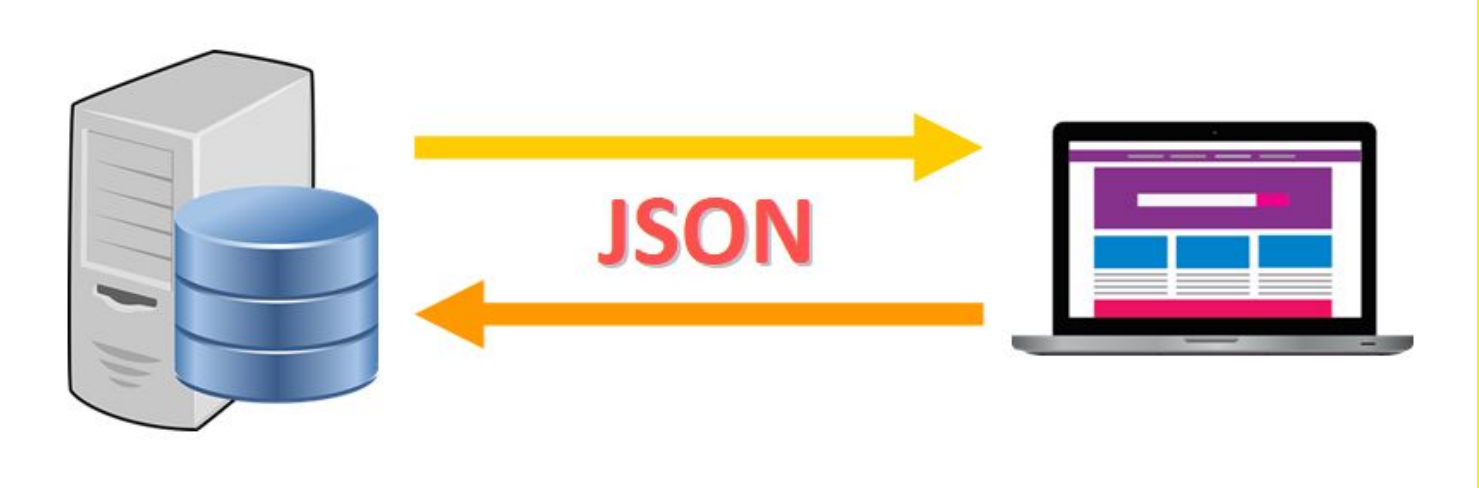
```
POST /api/orders HTTP/1.1
Host: api.amaxon.com
Authorization: Bearer YOUR_ACCESS_TOKEN
Content-Type: application/json
Accept: application/json
X-Client-Version: 2.0.1
User-Agent: AmaxonApp/1.01
```

WHAT IS THE REQUEST BODY?



WHAT IS THE REQUEST BODY?

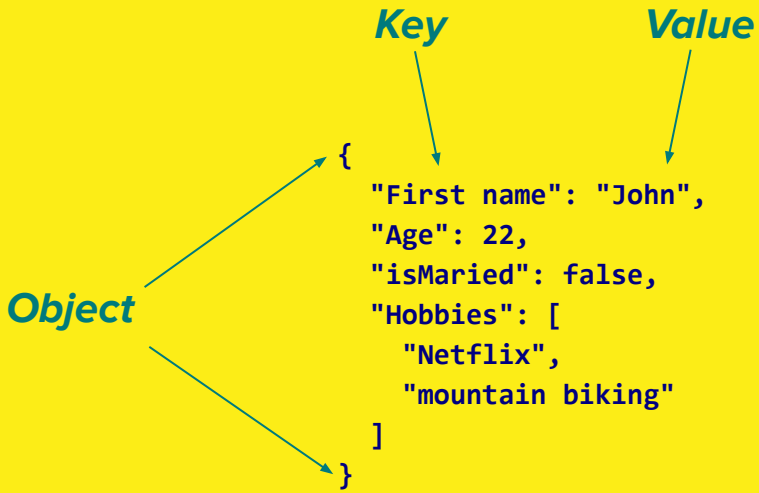
As part of request, a data payload could be send to the server in the body of the request. The body content can be any valid JSON object



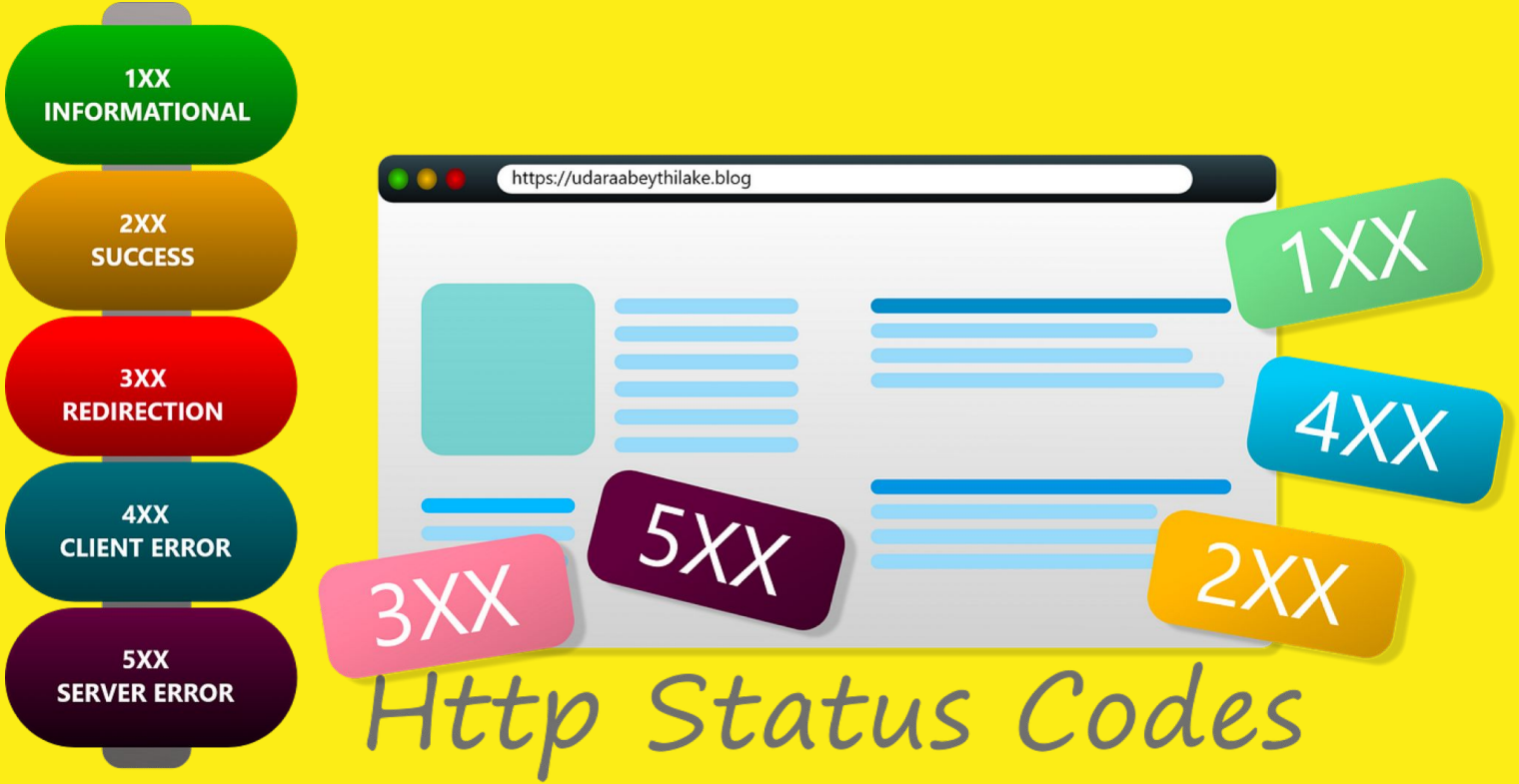
We use JSON to transfer data

JSON FORMAT

JSON has a simple key-value format.



RESPONSE CODES



RESPONSE CODES

1xx: Informational - Request received, continuing process
2xx: Success - The action was successfully received, understood, and accepted
200 OK - everything went as expected
201 Created - a new resource has been created as the result of request
202 Accepted - request was accepted but is not complete yet
204 No Content - request was processed successfully and no data returned
3xx: Redirection - Further action must be taken in order to complete the request
301 Moved Permanently - this response should include the new URI in the header so that the client will know where to point the request next time
4xx: Client Error - The request contains bad syntax or cannot be fulfilled
401 Unauthorised - client doesn't have the appropriate authorisation to make the request, such as a JWT or token
403 Forbidden - the client has the appropriate authentication to make the request but doesn't have the permission to view the resource
404 Not Found - the specific resource is not found
409 Conflict - the request puts data resources in conflict with one another
5xx: Server Error - The server failed to fulfill an apparently valid request
503 Service Unavailable - the responding server is temporarily down for some reason.

03

HOW TO TEST API?

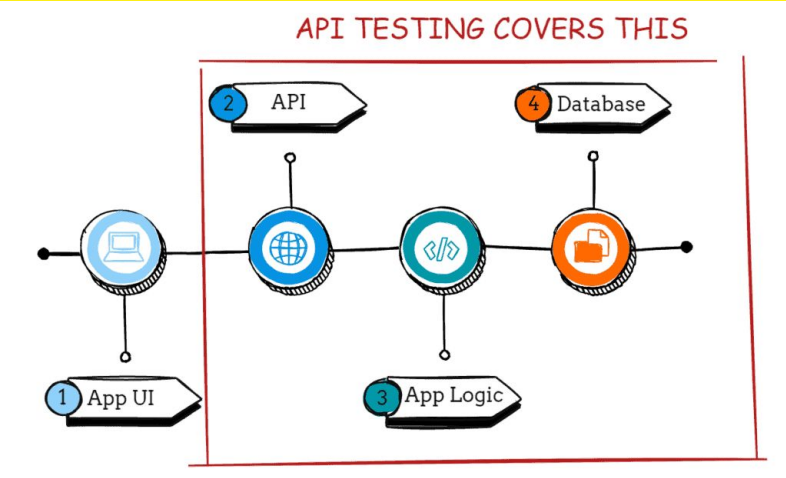
Why should you want to test API?

(a) You can find bugs earlier
Testing REST requests means you can find bugs earlier in the development process, sometimes even before UI has been created!

(b) You can find flaws before they are exploited
Malicious users know how to make REST request and can use them to exploit security flaws in your application by making requests the UI doesn't allow; you'll want to find and fix these flaws before they are exploited

(c) It is easy to automate

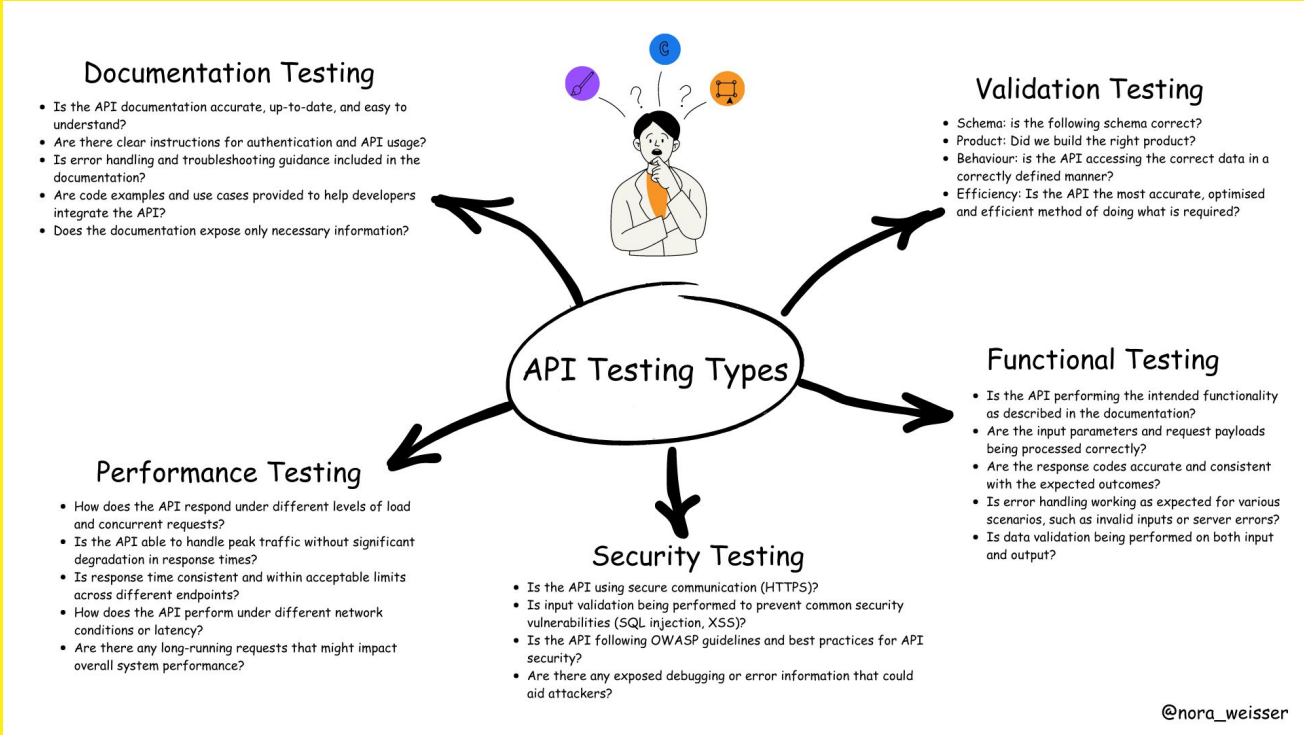
(d) Automation scripts run much faster than UI Automation



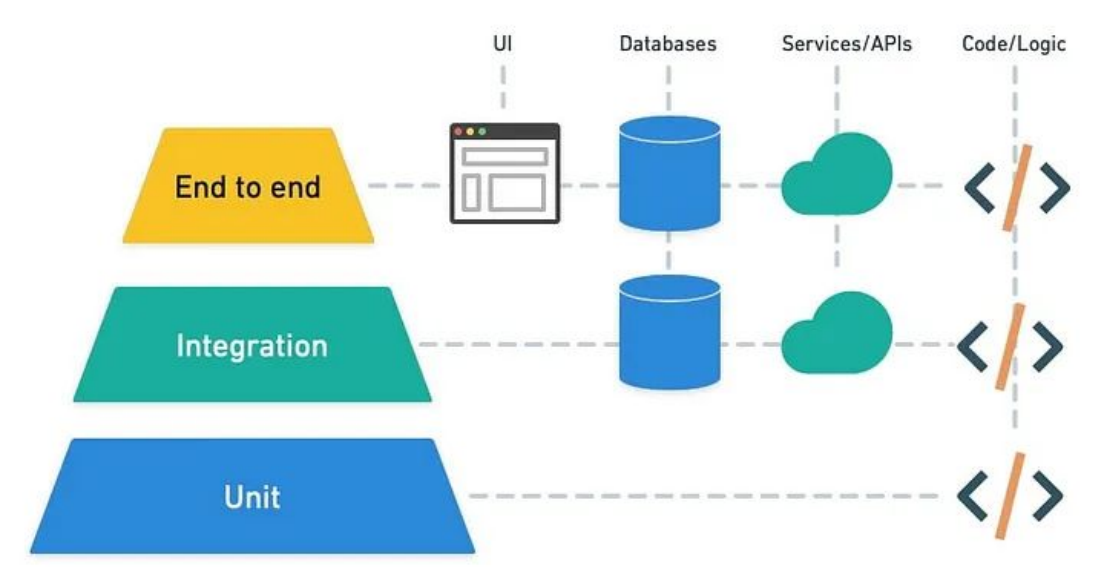
Source: https://methodpoet.com/api-testing-vs-ui-testing/

API TESTING TYPES

- VALIDATION
- FUNCTIONAL
- SECURITY
- LOAD
- INTEGRATION
- DOCUMENTATION



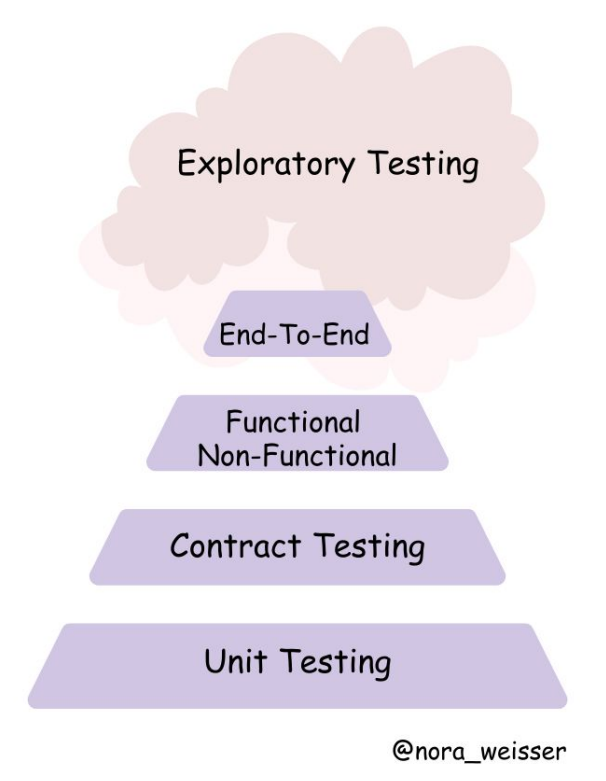
How to approach API Testing?



Introduced by Mike Cohn in his book Succeeding with Agile (2009), the pyramid is a metaphor for thinking about testing in software.

The testing pyramid is a concept in software testing that represents the ideal distribution of different types of tests in a software development process.

Testing Pyramid Adjusted

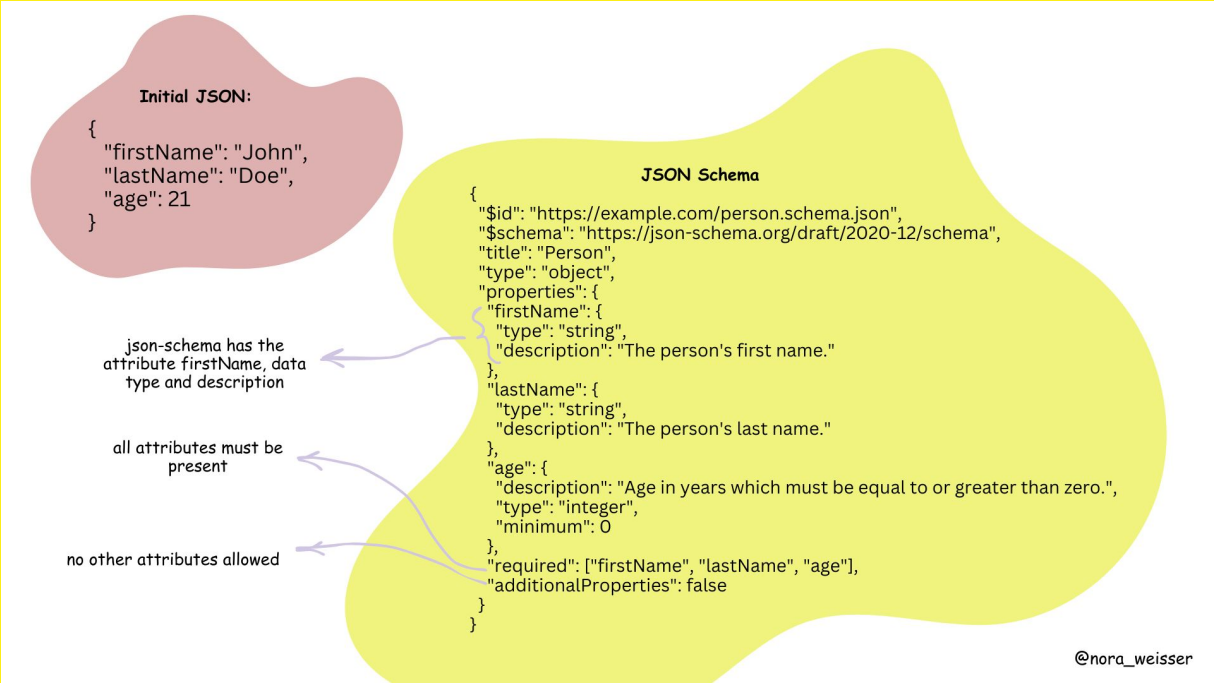


Unit Testing. By writing unit tests as early as possible we should be able to identify any problems with the current components of APIs as soon as possible.

Contract Testing assert that the specs have not changed. This type of testing is used to test the contracts or agreements established between various software modules, components, or services that communicate with each other via APIs. These contracts specify the expected inputs, outputs, data formats, error handling, and behaviours of the APIs.

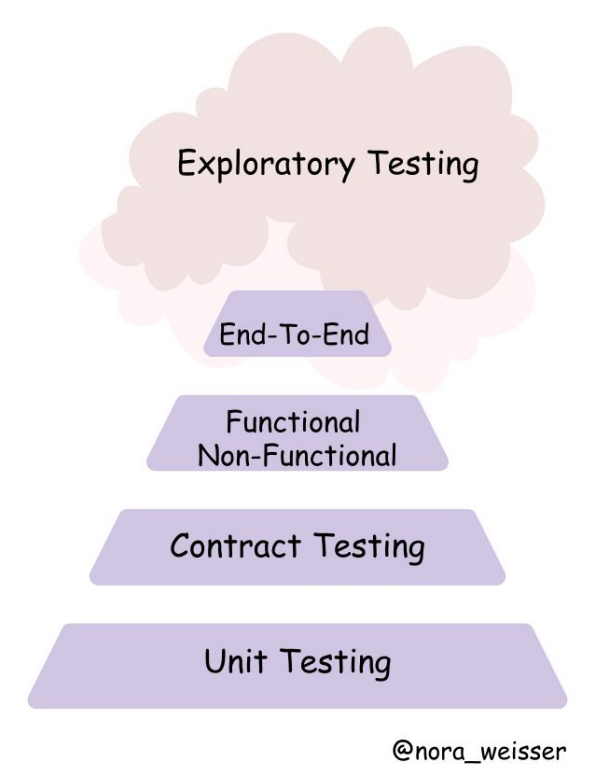
Testing Pyramid Adjusted (2)

JSON-schema is a contract that defines the expected data, types and formats of each field in the response and is used to verify the response.



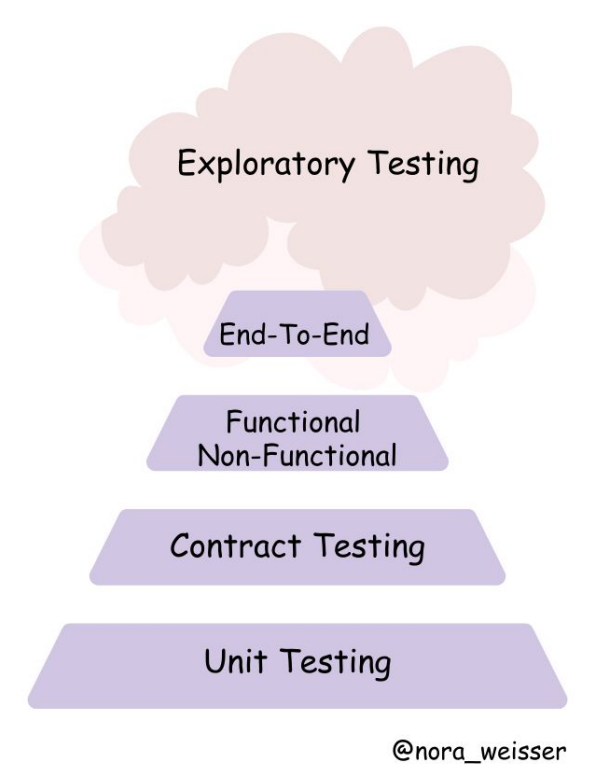
Official Documentation:
https://json-schema.org/

Testing Pyramid Adjusted (3)



Functional Testing. The purpose of functional testing is to ensure that you can send a request and get back the anticipated response along with status. That includes positive and negative testing.

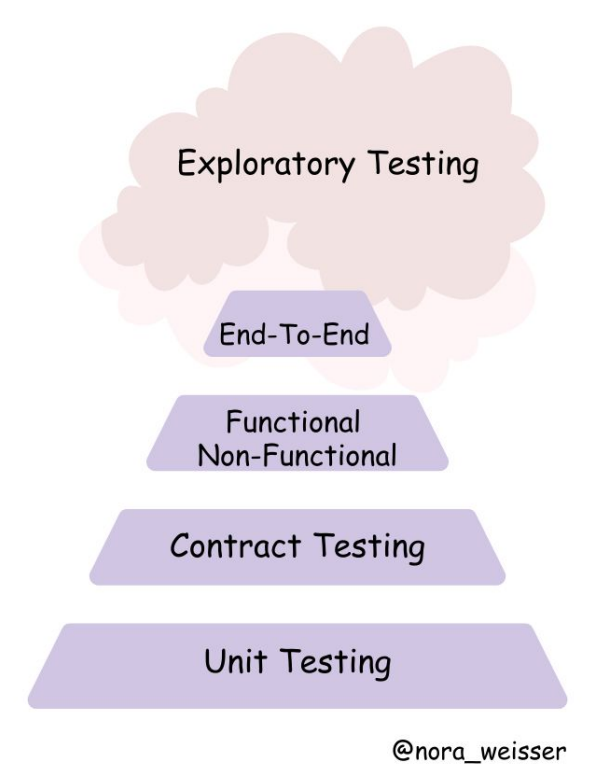
Testing Pyramid Adjusted (4)



How to start with Functional Testing?

1. Read API Documentation
2. Based on the functionality you are going to test, outline positive and negative test scenarios
3. Setup the test environment
4. Select the appropriate tool, framework, technologies, programming languages
5. Plan test data
6. Write automation scripts
7. Test error-handling mechanisms
8. Document test results
9. Collaborate with developers
10. Continuous improvement
11. Feedback loop

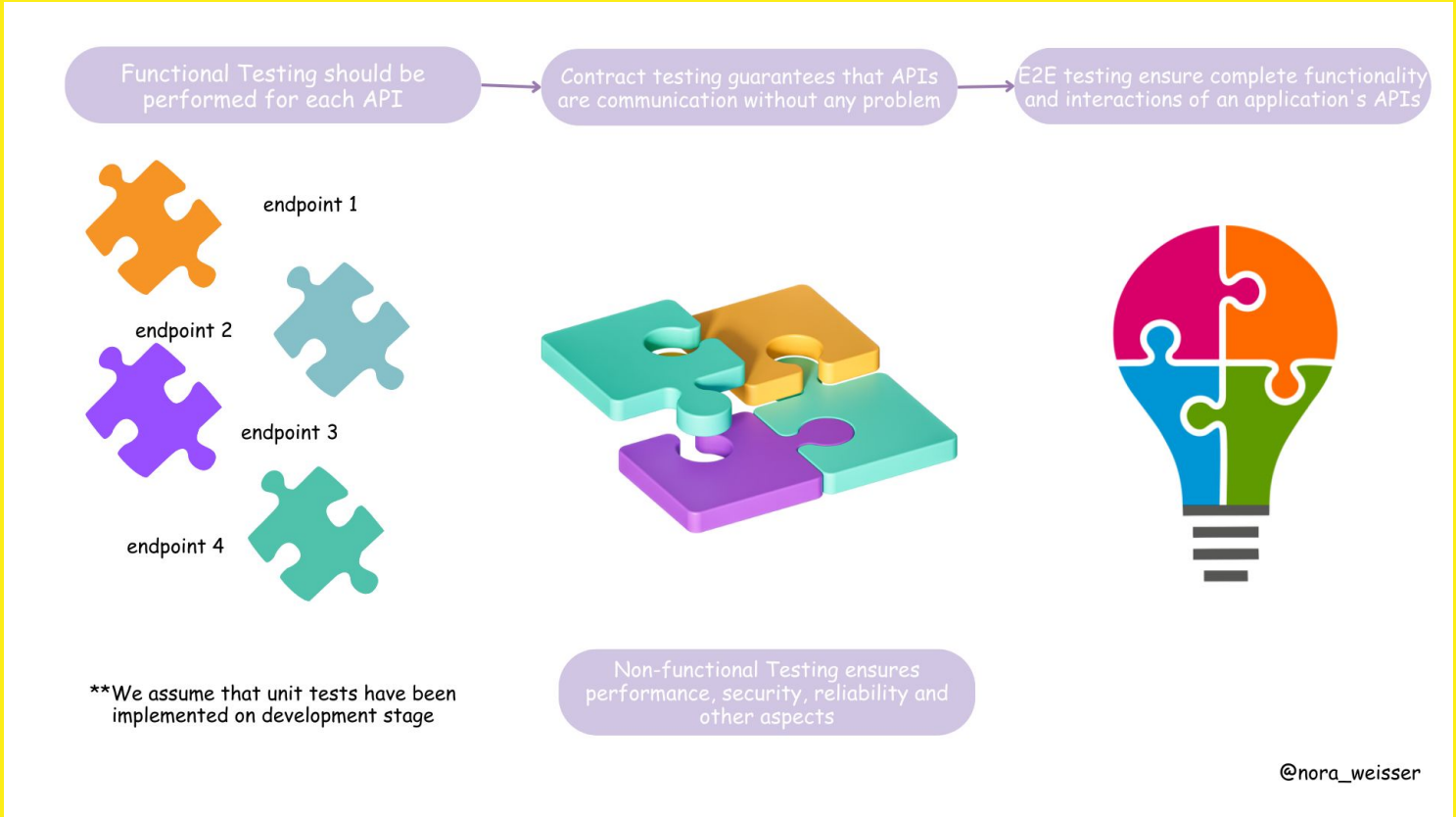
Testing Pyramid Adjusted (5)



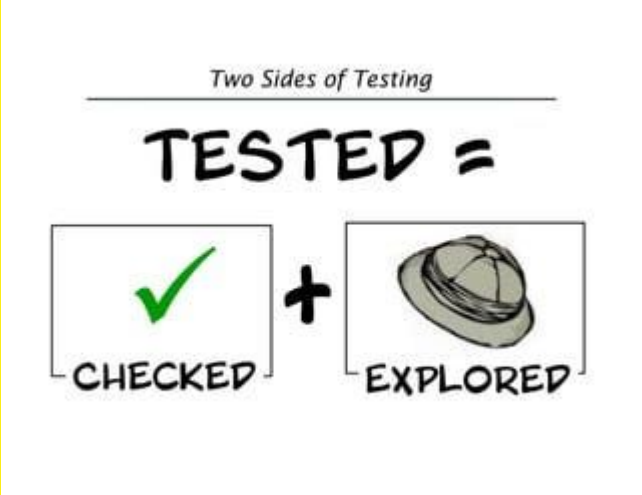
Non-Functional
Non-functional API testing is where the testers check the non-functional aspects of an application, like its performance, security, usability, and reliability.

End-to-end testing
In general, end-to-end testing is the process of testing a piece of software from start to finish. We are checking it by mimicking user actions. If it comes to API, it is crucial to check if APIs can communicate properly by making call like a real client.

Testing Pyramid Adjusted (6)



Testing Pyramid Adjusted (7)



> *You're not done testing until you've checked that the software meets expectations and you've explored whether there are additional risks. A comprehensive test strategy incorporates both approaches.*
>
> *Elisabeth Hendrickson, book "Explore It!"*

API TESTING TOOLS

Examples of APIs to practice testing

1. PetStore Swagger - REST API with OpenAPI Docs
2. BoredAPI - GET Endpoints
3. ExpandTesting
4. Simple Grocery Store API
5. Restful Booker
6. The Restful Pokemon API

04

Let's take a look at some examples

Thank you

Follow us!
https://linktr.ee/wwcode_london