API

▼ What is an API?

API stands for "Application Programming Interface".

It is a system that enables communication between different software.

You can think of it as two people speaking different languages communicating through an interpreter. API acts as an **interpreter** facilitating understanding between two different software and enabling data exchange.

Benefits of API:

- **Security:** Establishes a secure connection between two different servers.
- **Speed:** Speeds up data exchange.
- **Convenience:** Facilitates software development.
- **Saving:** Saves time and money.
- **Traffic:** Increases the traffic of your website or application.
- **Visibility:** Increases the visibility of your website or application.

Types of API:

- **Internal API:** APIs used by specific individuals.
- **Open API:** APIs open to everyone's use.
- **Partner API:** APIs used between two companies.
- **Composite API:** APIs that combine multiple APIs.

API Architectures:

- **REST API:** The most commonly used API architecture.
- **SOAP API:** A more secure API architecture.

Examples of APIs:

- Google Ads API
- Facebook API
- YouTube API
- WhatsApp Business API

Examples:

- Automatic synchronization of a website and a mobile application.
- Integration of an e-commerce site with payment infrastructure.
- Connection of a social media platform with other platforms.

▼ Request

The areas we need to check when preparing an API Request:

(*) Mandatory fields.

1. HTTP Request Types (Get, Post, Put - Patch, Delete)*
2. Base URL*
3. Endpoint*
4. Request Headers (Location for Additional Information)

5. Params
    a. Path
    b. Query
6. Request Body (Mandatory for Post)
7. Authorization, Authentication (Token)

▼ Response

1. Status Code
    a. **1xx:** The server acknowledges receiving your request and starts processing it.
    b. **2xx:** The server indicates that it has successfully received, understood, and accepted your request.
        i. (200→ Ok, 201→ Created, 202→ Accepted, 204→ No Content)
    c. **3xx**: Indicates that additional steps are required to complete your request.
    d. **4xx**: The server cannot process your request because you may have made it incorrectly.
        i. (400-Bad Request, 401-Unauthorized, 403-Forbidden, 404-Not Found, 405-Method not Allowed)
    e. **5xx:** Indicates that the server cannot process your request due to a server error.
2. Response Headers
3. Response Body (Json) // There are 6 ways to verify the Body.

▼ Validation Response Body

1. response.asString();
2. response.path("GPATH SYNTAX")
3. Jsonpath jsonpath = response.jsonpath();
   Jsonpath.getString("GPATH SYNTAX")
4. HamCrestMatchers
   RestAssured.
   .given()
   .when()
   .get("BASEURL + ENDPOINT")

   .then()
   ...
5. Json to Java with as() method → DE-SERIALIZATION
6. POJO (PLAİN OLD JAVA OBJECT)

▼ Authentication - Authorization

▼ Authentication

- Who is this?
- **401:** Invalid credentials
- **401: Unauthorized**
    - The API doesn't know who you are.

▼ Authorization

- Give permissions
- **403:** You don't have sufficient privileges to perform the operation.
- **403: Forbidden**
    - The API allows entry but with limited privileges."

▼ API Architectures

API architectures are a set of principles and rules that determine how APIs are designed and developed. Different API architectures have different advantages and disadvantages. The most commonly used API architectures are as follows:

1. REST API (Representational State Transfer):

- It is the most widely used API architecture.
- It is simple and easy to use.
- It utilizes HTTP methods (GET, POST, PUT, DELETE).
- It uses data formats such as JSON or XML.
- It is scalable and flexible.

2. SOAP API (Simple Object Access Protocol):

- It is a more secure API architecture.
- It is XML-based.
- It uses standards such as WSDL (Web Service Definition Language).
- It is more complex and harder to use.
- **REST:** Preferred when speed, flexibility, and simplicity are important.
- **SOAP:** Preferred in situations where security and error management are critical.

▼ API vs Web Services?

Both APIs (Application Programming Interfaces) and Web Services are ways to communicate between applications. However, they have some important differences:

API (Application Programming Interface)

- **Scope:** APIs are more general concepts. They are software interfaces used for any communication protocol and data exchange.
- **Protocols:** They can use various protocols including REST, SOAP, GraphQL, and more.
- **Formats:** APIs support JSON, XML, and other data formats.
- **Flexibility:** APIs are more flexible architecturally and provide a broader range of interaction between applications on different platforms.

Web Services

- **Nature:** Web services are a specific subset of APIs.
- **Protocols:** They are inherently tied to specific web technologies. They use protocols like SOAP (usually transmitting data in XML format) and less commonly XML-RPC, UDDI, etc.
- **Standards:** Web services have stricter standards and protocols.
- **Compatibility:** When it comes to cross-platform compatibility with systems built on different technologies, web services may be more restrictive compared to APIs.

In Summary:

- **All Web Services are APIs, but not all APIs are Web Services.**
- **Web Services are more rigid and use mandatory protocols like SOAP and XML.** SOAP APIs typically support an XML document called WSDL (Web Service Definition Language) that defines the functionality of the API.

- **APIs allow for more flexibility with architectures like REST, GraphQL, and offer more options for data formats like JSON.**

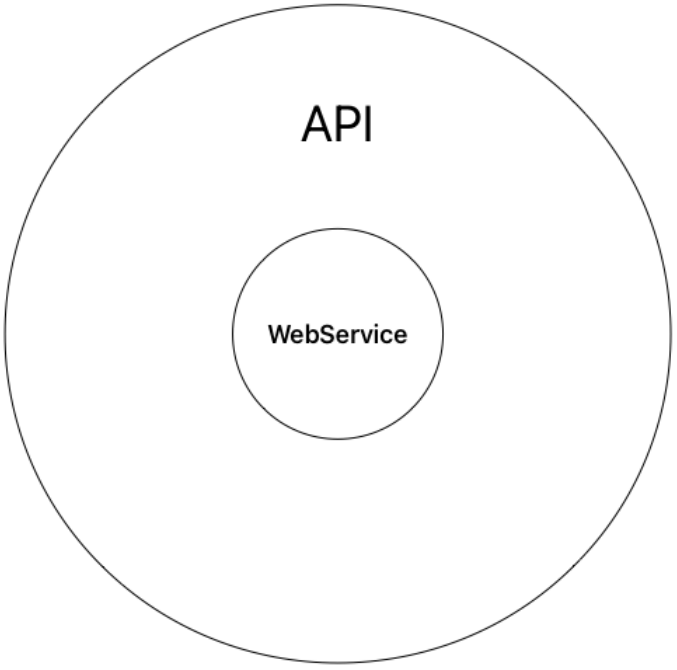
Which One to Choose?

What you choose will depend on your objectives:

- **Integrating different platforms:** If you need to work with legacy systems using SOAP and require more robust standards, web services might be appropriate.
- **High flexibility and different protocols:** If you need a more flexible architecture or prefer lighter data formats like JSON, APIs are a better choice.

Additional Notes:

- In modern usage, the term "API" often refers to web-based APIs like REST APIs. However, APIs do not require network connections (operating system libraries can also be considered APIs).
- Web Services are now considered an older technology, and API types like REST have become more popular.



▼ INTERVIEW Questions and Answers

▼ Group A

▼ What is an API and what is its purpose?

- *Answer:* An API (Application Programming Interface) is an interface used to communicate between software applications. It is used to share specific functions or data of one software with others.

▼ What is the importance of API testing and why should it be done?

- *Answer:* API tests are performed to verify the functionality, reliability, and performance of software. These tests are done to check if the software produces the expected results, to detect and fix errors, and to ensure that the application is stable and reliable.

▼ Do you prefer to perform API tests manually or automatically? Why?

- *Answer:*I generally prefer a combination of both; meaning, I perform some critical scenarios manually alongside automated tests.

    - Automated tests provide advantages in terms of repeatability, comprehensiveness, and speed.
    - Manual tests, on the other hand, are more effective in checking more complex scenarios using human intelligence and intuitive understanding.

▼ What do you pay attention to in API testing and how do you evaluate based on which criteria?

- *Answer:* Important aspects in API testing include the accuracy of inputs, whether outputs produce expected results, handling of error conditions, security checks, and performance. Additionally, it's important for the API to be well-documented, allowing users to easily understand and use it.

▼ Which HTTP methods (GET, POST, PUT, DELETE, etc.) do you use and when are they used?

- *Answer:*
    - GET is used to retrieve a specific resource or data,
        - Viewing the content of a page.
    - POST is used to create a new resource or add data.
        - Saving a new user.
    - PUT is used to update the current state of a resource.
        - Updating the price of a product.
    - PATCH can be used to partially update a resource.
        - Updating certain information in a user's profile.
    - DELETE is used to delete a resource.
        - Deleting a product.

▼ How is authentication done in API testing and why is it important?

- *Answer:* Access to the API is typically provided via API keys, OAuth tokens, or other authentication mechanisms. Authentication helps prevent unauthorized access and ensures that users perform only authorized operations.

▼ How are error conditions handled and tested in API testing?

- *Answer:* Error conditions should be tested to determine how the API behaves in unexpected situations. These conditions often include incorrect inputs, lack of authorization, network interruptions, etc. It's important for error conditions to align with documented error codes and descriptions of the API.

▼ What types of security tests are performed in API testing and why are they important?

- *Answer:* Security tests in API testing cover areas such as authentication and authorization controls, data integrity, data accuracy, data privacy, and resilience against attacks. These tests are important to ensure the security of the API and the protection of sensitive data.

▼ B Group

▼ What is API testing?

Answer: API testing is the process of testing the functionality, performance, and security of an API.

▼ What types of tests should be performed on APIs?

Answer: Various types of tests should be performed on APIs including unit tests, integration tests, functional tests, performance tests, security tests, and user interface tests.

▼ How do you perform API testing manually?

Answer: To perform API testing manually, a testing tool or a tool like **Postman** can be used to send requests to the API and verify the responses.

▼ What tools can you use to automate API testing?

Answer: There are many tools available to automate API testing such as RestAssured, Karate DSL, Postman, and SoapUI.

▼ What are the most common errors in API testing?

Answer: The most common errors in API testing include 404 errors, 401 errors, 500 errors, and incorrect response formats.

▼ What tests do you perform to test the functionality of an API?

Answer: To test the functionality of an API, you perform tests to cover all endpoints of the API (e2e) and send appropriate requests to get expected responses.

▼ What tests do you perform to test the performance of an API?

Answer: To test the performance of an API, you perform load testing and performance testing to measure response times and how the API behaves under load.

▼ What tests do you perform to test the security of an API?

Answer: To test the security of an API, you perform security tests to check for attacks such as SQL injection, authentication bypass, and authorization escalation.

▼ How important is automation in API testing?

Answer: Automation is very important in API testing because it saves time, increases the consistency and accuracy of tests, and facilitates regression testing.

▼ What are the best steps in API testing?

Answer: The best steps in API testing include creating a test plan, writing test scenarios, generating test data, using automation tools, and reporting test results.

▼ Group C

▼ What is checked first after receiving the request?

- Answer:
  1. Status code
  2. Headers (Optional)
  3. Response body (Optional)

▼ What is Serialization and Deserialization?

- Answer:

  Serialization: Serialization is the process of converting the state or structure of an object or data structure in memory into a format suitable for storage. This process is typically done by converting an object or data structure into a format such as JSON, XML, or binary. Serialization allows an object's state to be transferred from memory to disk or over a network while preserving its state.

  Summary:

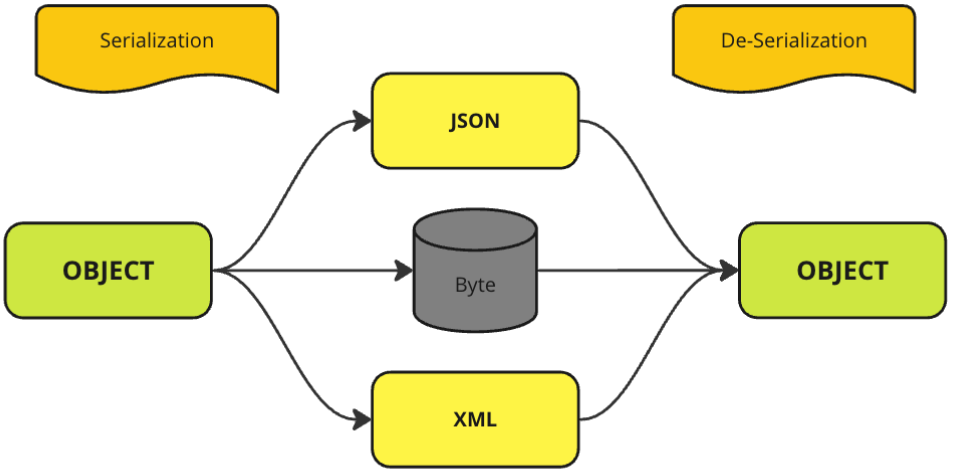
  It is the process of converting an object or data structure into a specific format for storage or transmission purposes.

  Deserialization: Deserialization is the opposite of serialization. It involves converting data retrieved from a format (such as JSON or XML) back into its original object or data structure. This process allows data stored or transmitted on disk or over a network to be reconstructed by the program for use in memory.

  Summary:

  It is the process of converting a serialized object or data structure back into its original form.

Serialization and deserialization are important parts of APIs and allow for data exchange by converting different data types into each other.



Why is it used?

- **Data Storage:** Objects or data structures need to be serialized for storage on disk, in a database, or over a network.
- **Data Transmission:** Objects or data structures need to be serialized to be transferred between different systems or applications.
- **Interoperability:** A common format is used for data exchange between different programming languages and applications.

Example:

- A Java application can serialize a user object to JSON format and send this JSON data to a web API.
- The web API can deserialize the JSON data back to a user object using deserialization and store this object in a database.

▼ What is an Endpoint?

- Answer:

  It is a specific address that provides access to a service or resource.

  - An Endpoint is the entry point for a request made to an API.
  - Endpoints retrieve specific data or trigger operations that modify data on the server.
  - Endpoints specify the type of request using HTTP methods.

    Example:

    - **House Analogy:** Think of the API as a house. Endpoints are like doors leading to different rooms in this house. The bedroom door leads to the bedroom, the kitchen door leads to the kitchen, and so on.
    - **Restaurant Analogy:** Imagine the API as a restaurant. Endpoints can be thought of as the dishes served in this restaurant. Each dish has its own URL (/pizza, /hamburger, /salad, etc.).

▼ What are XML and JSON concepts?

- **XML (Extensible Markup Language):** It is a markup language used to define and store data.

- **JSON (JavaScript Object Notation):** It is a lightweight data interchange format used to represent data.

  Data is stored in a "Key" and "Value" format.

Differences Between XML and JSON:

Feature	XML	JSON
Complexity	More complex.	Simpler.
Readability	Less readable.	More readable.
Flexibility	Less flexible.	More flexible.
Performance	Slower.	Faster.
Data Types	Supports more data types.	Supports fewer data types.
Use Cases	Used in areas such as web services, configuration files, data transmission.	Used in areas such as web APIs, NoSQL databases, data interchange with JavaScript.

▼ Gson

- Answer:

  Gson is a Java library developed by Google for converting Java objects to JSON (serialization) and JSON to Java objects (deserialization).

  - JSON is a data interchange format, while Gson is a Java serialization/deserialization library.
  - JSON is simple and works across different platforms. Gson is specifically designed for Java and offers more flexibility.

▼ Swagger

- Answer:

  Swagger is an open-source software used for **designing**, **documenting**, and **testing** APIs. With features like easy design and creation, standard format, and live testing, it makes your API more usable and developer-friendly.

▼ How to Test an API?

- Answer:

  As a tester we send a API request and verify the status code, response body and checking the endpoints of the api URL is working as expected

  Pozitive - I send valid requests, headers, parameters, and JSON bodies, and verify that the response is 200/201.

  Negative - I send invalid requests, headers, parameters, and bodies, expecting the response not to be 200.

▼ How to Test a REST API?

- Answer:
  - **API Validation:** Making sure each REST API endpoint works as expected.
  - **Postman:** A popular API platform for manual testing.
  - **Rest Assured:** A library for automating API tests with Java.

  Methods:

  - **HTTP Requests:** Sending requests to API endpoints using various HTTP methods like POST, PUT, GET, DELETE.
  - **Response Verification:** Checking if the API returns the correct status codes (200, 400, 401, 500, etc.) and if the response content is as expected. Headers can also be verified.
  - **Positive and Negative Testing:**

- **Positive Tests:** Testing with valid request parameters, headers, and JSON bodies to verify that the API works as expected in successful scenarios (200 status code and correct JSON response).
- **Negative Tests:** Testing with invalid request parameters, headers, and JSON bodies to verify that the API handles error scenarios (non-200 status codes and error messages) correctly.

Summary:

- Comprehensive testing using different HTTP methods.
- Preparation of positive and negative test scenarios.
- Verification of HTTP response codes, response bodies, and headers.
- Use of appropriate tools for manual testing (Postman) and automated testing (Rest Assured).

▼ RestAssured

- Answer:

  RestAssured is an easy-to-use, flexible, and comprehensive open-source library for testing REST APIs in Java.

  With RestAssured, you can test the functionality of API endpoints, verify expected responses and error codes, and create automated test scenarios.

▼ JsonPath

One of the ways to verify the response body.

```
Jsonpath jsonpath = response.jsonpath();
```