

API INTERVIEW QUESTION AND ANSWER

1.What is Api ?

Answer: API stands for Application Programming Interface. It is a set of rules and protocols that allow different software applications to communicate with each other. APIs define the methods and data formats that developers can use to request and exchange information between systems, services, or software components.

- APIs provide a way for different software components to interact and share data. This interaction can happen over internet or within the same application.
- APIs can be found in various forms, including web APIs, libraries, operating system APIs, and more. Web APIs, for example, allow web applications to interact with remote servers or services over HTTP.
- APIs often include mechanisms for authentication and authorization to ensure that only authorized users or applications can access their resources.
- APIs serve as intermediaries that enable different software components to communicate and share data, allowing developers to build more powerful and integrated applications by leveraging existing services and functionalities.

2. What is a Web API and why is it used?

A Web API (Application Programming Interface) is a set of rules and protocols that allow different software applications to with each other over the internet or a network. It provides a standard way for applications to request and exchange data, typically using HTTP (Hypertext Transfer Protocol) requests and responses. Web APIs is commonly used for enabling communication between a client (such as a web or mobile application) and a server or between various parts of a distributed system.

Some Key reason to use Web API:

- Web APIs allows different software systems to integrate with each other seamlessly. For example, a web application can use a payment gateway's API to process payments, or a mobile app can integrate with social media platforms to fetch user data.
- Developers can leverage third-party services and resources through APIs, rather than reinventing the functionality. For example, mapping and geolocation services, payment gateways, and weather data can all be accessed through APIs.
- Web APIs can include authentication and authorization mechanisms to ensure that only authorized users or applications can access certain resources. This adds a layer of security to data and services.

3.What are four types of Web API?

Web APIs can be categorized into several types based on their functionality, usage, and how they are accessed. Here are four common types of Web APIs:

1. RESTful APIs (Representational State Transfer):

- RESTful APIs are designed around a set of architectural principles, often referred to as REST.
- They use standard HTTP methods (GET, POST, PUT, DELETE, etc.) for CRUD (Create, Read, Update, Delete) operations on resources.
- Resources are represented using URLs, and data is typically exchanged in formats like JSON or XML.
- RESTful APIs are known for their simplicity, scalability, and statelessness.

2. SOAP APIs (Simple Object Access Protocol):

- SOAP is a protocol for exchanging structured information in the implementation of Web services.
- SOAP APIs use XML as their message format and typically rely on HTTP, SMTP, or other transport protocols.
- They have a strict and standardized structure and often include features like security and transactions.

3. GraphQL APIs:

- GraphQL is a query language for APIs that allows clients to request exactly the data they need.
- Unlike REST, where the server defines the structure of responses, GraphQL clients specify their data requirements in the query.
- GraphQL APIs provide more flexibility and efficiency for clients, as they can retrieve only the specific data fields they require in a single request.
- This type of API is commonly used in applications where fine-grained control over data fetching is essential.

4. WebSocket APIs:

- WebSocket is a communication protocol that provides full-duplex, bidirectional communication channels over a single TCP connection.
- WebSocket APIs enable real-time, interactive communication between a client and a server.
- They are often used for applications requiring instant updates or live data streaming, such as chat applications, online gaming, and financial trading platforms.

4. What is RESTful API?

REST, which stands for Representational State Transfer, is an architectural style and set of constraints for designing networked applications. RESTful APIs are widely used in web development because of their simplicity, scalability, and ease of use.

REST models the system's resources as URLs (Uniform Resource Locators). Resources can represent data objects, such as user profiles, articles, or product listings. Each resource is identified by a unique URL, and clients interact with these resources using standard HTTP methods (GET, POST, PUT, DELETE) to perform CRUD (Create, Read, Update, Delete) operations.

5. What are the benefits of RESTful Web services?

RESTful web services offer several benefits, making them a popular choice for building APIs and web applications. Here are some of the key advantages of RESTful web services:

- REST follows a simple and intuitive architectural style based on standard HTTP methods (GET, POST, PUT, DELETE) and URLs. This simplicity makes it easy for developers to understand, use, and implement RESTful APIs.
- REST is stateless and relies on standard HTTP, it is highly scalable. Servers can handle many concurrent clients, and load balancing is straightforward since each request is independent.
- RESTful APIs can be secured using standard HTTP security mechanisms, such as HTTPS for data encryption, authentication, and authorization through various methods like API keys, tokens, or OAuth.
- RESTful services leverage HTTP caching mechanisms, which can improve performance by reducing the need for repeated data retrieval.
- REST leverages the existing HTTP infrastructure, including status codes, headers, and methods. This integration simplifies error handling and response interpretation.

5. What are the main differences between API and Web Service?

API	Web Service
A web service is a specific type of API that is designed to enable communication and data exchange over the internet using standard web protocols such as HTTP.	A web service is a specific type of API that is designed to enable communication and data exchange over the internet using standard web protocols such as HTTP.
APIs can be implemented using a variety of protocols and standards, depending on the application and use case.	Web services follow specific protocols and standards, such as SOAP or REST, which define the structure, communication methods, and behaviours of the service.
APIs can work over different transport mechanisms, including HTTP, message queues, sockets, and more. They are not limited to web-based communication.	Web services are specifically designed for web-based communication, primarily using HTTP/HTTPS for transport.

6. What are the limits of API?

There are some basic Limitation of the API are :

1. Many APIs impose rate limits on the number of requests a client can make within a specific time period. These limits are often in place to prevent abuse and ensure fair usage.
2. APIs are dependent on the availability and reliability of the underlying services or systems they connect to. If the API's backend experiences downtime or outages, it can impact the functionality of the client applications.
3. Some APIs, especially those offered by third-party providers, may have associated costs based on usage. Developers need to be mindful of potential costs and budget accordingly.
4. As an application grows, the demands on the API can increase. Ensuring that the API can scale to handle growing traffic and usage is a concern that needs to be addressed.
5. Large volumes of data transferred through APIs can lead to bandwidth and performance issues. Minimizing unnecessary data transfer and optimizing data payloads are important considerations.

7. What are the different types of API testing?

1. Functional Testing:
 - Purpose: Functional testing verifies that the API functions according to its specifications and behaves as expected in various scenarios.
 - Scope: Test cases cover the API's functionality, including input validation, error handling, and the correctness of responses.
 - Tools: Tools like Postman, SoapUI, and Rest Assured are commonly used for functional API testing.
2. Integration Testing:
 - Purpose: Integration testing checks how well different components of the application or system work together when interacting through APIs.
 - Scope: It focuses on testing the interactions and data flow between APIs, ensuring that they integrate correctly and produce the expected outcomes.
 - Tools: Integration testing can be done using test frameworks, and it often involves running end-to-end tests or using API-specific tools.
3. Load Testing:

- Purpose: Load testing assesses the API's performance under various levels of concurrent user traffic or data loads.
- Scope: It measures the API's response times, throughput, and resource utilization to identify performance bottlenecks and scalability issues.
- Tools: Tools like Apache JMeter, LoadRunner, and Gatling are commonly used for load testing APIs.

4. End-to-End Testing:

- Purpose: End-to-end testing evaluates the entire workflow of an application, including the interactions between various components and APIs, to ensure that the entire system works correctly.
- Scope: It simulates real-world scenarios to verify that APIs and their integration with other components function seamlessly.
- Tools: End-to-end testing may involve a combination of API testing tools and application testing frameworks.

8. What are the most used tools for API testing?

1. Postman: Comprehensive API testing and development tool. Easy-to-use interface for creating, testing, and documenting APIs.
2. SoapUI: Specialized tool for testing SOAP and REST APIs. Robust testing capabilities for both SOAP and RESTful web services. Supports data-driven testing, security testing, and mock services.
3. JUnit and TestNG: Java testing frameworks. While not API-specific tools, JUnit and TestNG are commonly used to write and execute API tests in Java.
4. Apache JMeter: Performance and load testing tool. Although primarily designed for load testing, JMeter can also be used for functional API testing. It supports various protocols, including HTTP, SOAP, and REST.
5. Cypress: End-to-end testing framework with API testing capabilities. Cypress is primarily designed for web application testing, but it can also make API requests and assertions.