

Q1. What is an API? Give an example, where an API is used in real life.

Answer.

An application programming interface is a way for two or more computer programs to communicate with each other. It is a type of software interface, offering a service to other pieces of software. A document or standard that describes how to build or use such a connection or interface is called an API specification.

APIs are used in web applications to connect user-facing front ends with critical back-end functionality and data. APIs are used to deliver material by streaming services such as Spotify and Netflix. APIs are used by automakers such as Tesla to provide software upgrades.

Q2. Give advantages and disadvantages of using API.

Answer: Advantages:

- *Increased Interoperability:** API solutions enable different software systems to communicate with each other, regardless of their programming language or platform. This enhances the interoperability between systems, allowing for the exchange of data and functionality.
- *Improved Efficiency:** API driven infrastructures enable developers to reuse code and functionality from existing software systems, saving time and reducing the amount of code that needs to be written from scratch.
- *Better User Experience:** APIs can enable third-party developers to create applications that integrate with existing software systems, providing a better user experience and increased functionality.
- *Increased Flexibility:** APIs provide developers with greater flexibility, enabling them to choose the best programming languages and frameworks for their needs.
- *New Revenue Streams:** APIs can enable companies to create new revenue streams by providing third-party developers with access to their software systems.

Disadvantages:

- *Security Concerns:** API infrastructures can be vulnerable to security breaches, such as SQL injection attacks and cross-site scripting (XSS) attacks, making it crucial to implement security measures to protect against these threats.
- *Complexity:** APIs can be complex, requiring a significant amount of development effort to implement and maintain.
- *Compatibility Issues:** APIs can have compatibility issues between different versions of the same software system or between different systems, which can lead to errors and system failures.
- *Dependency on Third-Party Developers:** APIs rely on third-party developers to create applications that integrate with the software system. This can create dependencies that are outside of the control of the company that owns the software system.
- *Governance and Management:** APIs require proper governance and management to ensure they are being used correctly and consistently across different applications and systems.

Q3. What is a Web API? Differentiate between API and Web API.

Answer: Web API is an API over the web which can be accessed using HTTP protocol. It is a concept and not a technology. We can build Web API using different technologies such as Java, .NET etc. For example, Twitter's REST APIs provide programmatic access to read and write data using which we can integrate twitter's capabilities into our own application.

As far as API's are concerned, API is a set of subroutine definitions, protocols, and tools for building software and applications. API is a kind of interface which has a set of functions that allow programmers to access specific features or data of an application, operating system or other services.

Q4. Explain REST and SOAP Architecture. Mention shortcomings of SOAP.

Answer: A REST API is an application programming interface (API) that uses a representational state transfer (REST) architectural style. The REST architectural style uses HTTP to request access and use data.

SOAP is a protocol for the exchange of information in a distributed environment. SOAP messages are encoded as XML documents and can be exchanged using various underlying protocols.

Shortcomings of SOAP:

***API Calls are not Cached**

An API call is a process of sending a request when an API has been set up with the necessary endpoints. The information is transferred, processed, and feedback is supplied as a result of the process. It is not feasible to cache SOAP API requests.

***Extremely Complicated**

SOAP is substantially more sophisticated than REST. It is also less adaptable. For developers who aren't very experienced, this could be a problem. Performance may be slowed as a result of the substantial processing required.

***No Choice of Data Formats**

SOAP's support for data formats is likewise severely constrained. HTML, JSON, YAML, XML, and more formats are supported by REST. SOAP, on the other hand, only supports XML.

***Requires More Bandwidth**

SOAP is often slower than REST, and it also consumes more bandwidth due to its complexity. It's also a limiting issue in the technology's efficacy for some applications.

Q5. Differentiate between REST and SOAP.

Answer:

SOAP API	REST API
Relies on SOAP (Simple Object Access Protocol)	Relies on REST (Representational State Transfer) architecture using HTTP.
Transports data in standard XML format.	Generally transports data in JSON. It is based on URI. Because REST follows stateless model, REST does not enforces message format as XML or JSON etc.
Because it is XML based and relies on SOAP, it works with WSDL	It works with GET, POST, PUT, DELETE
Works over HTTP, HTTPS, SMTP, XMPP	Works over HTTP and HTTPS
Highly structured/typed	Less structured -> less bulky data
Designed with large enterprise applications in mind	Designed with mobile devices in mind