

18th Feb Assignment

February 27, 2023

1 Assignment 17

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

Ans.API stands for Application Programming Interface, which is a set of protocols, routines, and tools that allow different software applications to communicate with each other. An API acts as an intermediary that enables two applications to talk to each other and exchange data in a standardized and structured manner.

For example, consider a weather forecasting application that provides current weather conditions and forecasts for a particular location. This application can use an API provided by a third-party weather service to access up-to-date weather data for that location. The API will provide the weather data in a standardized format that the application can easily understand and use to display the weather information to its users.

Q2. Give advantages and disadvantages of using API.

Ans.Advantages of using API:

- Improved efficiency: APIs provide a standardized way for applications to communicate with each other, which can improve efficiency by reducing the need for manual data entry or human intervention.
- Increased flexibility: APIs can allow developers to create new applications or add functionality to existing applications more quickly and easily by leveraging the capabilities of other software services.
- Enhanced user experience: APIs can provide access to data and functionality that may not be available in the application itself, which can lead to a better user experience.

Disadvantages of using API:

- Security risks: APIs can create potential security risks if they are not properly secured, as they provide a pathway for external entities to access sensitive data and functionality.
- Complexity: APIs can be complex to develop and maintain, especially when dealing with large volumes of data or complex integrations.
- Reliance on third-party services: When using third-party APIs, businesses may become reliant on the services provided by those third parties and may face disruptions if those services experience downtime or other issues.

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

Ans.A Web API is a type of API that is accessed over the internet through HTTP(S) protocols and is typically designed to be used by web applications. Web APIs provide a way for web applications to interact with external services and retrieve data or perform actions by making HTTP requests to a web server. Web APIs are often used to expose business logic or data that can be used by external clients, such as mobile applications or other web applications.

The main difference between API and Web API is that APIs can be designed to be used by any type of application, not just web applications, while Web APIs are specifically designed for use by web applications. APIs can be accessed using a variety of protocols, including TCP/IP, whereas Web APIs are typically accessed using HTTP(S) protocols. Additionally, while APIs can be used to interact with any type of software or service, Web APIs are typically used to interact with web services or web applications.

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

Ans.REST (Representational State Transfer) is an architectural style that uses HTTP(S) protocols to expose web services. RESTful web services are built around resources that can be identified by a URI, and data is exchanged between the client and the server using HTTP methods such as GET, POST, PUT, and DELETE. REST is a popular choice for building web APIs because it is lightweight, scalable, and flexible.

SOAP (Simple Object Access Protocol) is an architectural style that uses XML-based messaging to exchange data between clients and servers. SOAP is a more rigid and heavyweight approach compared to REST, as it uses a strict set of rules and protocols for exchanging messages. SOAP can support more complex functionality, such as distributed transactions and security protocols, but it is also more complex to implement and can be slower than REST.

Shortcomings of SOAP:

- Complexity: SOAP can be complex to implement and maintain, requiring a significant amount of development effort and expertise.
- Performance: SOAP can be slower than REST due to the overhead involved in processing XML-based messages.
- Limited browser support: SOAP is not supported by most browsers, making it difficult to use in web applications.

Q5. Differentiate between REST and SOAP.

Ans.The difference between REST and SOAP are:

- Protocol: REST uses HTTP(S) protocols to expose web services, while SOAP uses XML-based messaging protocols.
- Messaging format: REST typically uses lightweight formats such as JSON (JavaScript Object Notation) or XML for data exchange, while SOAP uses XML for message exchange.

- Complexity: REST is generally simpler and easier to implement compared to SOAP, which can be more complex and requires more development effort.
- Performance: REST can be faster and more efficient than SOAP due to its lightweight messaging format and lack of a strict message exchange protocol.
- Statelessness: REST is stateless, meaning each request from the client to the server contains all the necessary information to process the request, while SOAP supports stateful interactions.
- Flexibility: REST is more flexible than SOAP, as it allows for multiple data formats and can be used with different programming languages and technologies. SOAP, on the other hand, has a strict set of rules and protocols that can limit flexibility in some cases.