

Introduction to API

1. *Explain REST and RESTFUL?*

REpresentational State Transfer (REST), is an architectural style for providing standards between computer systems on the web, making it easier for systems to communicate with each other. In other words, REST is a way to access resources which lie in a particular environment. Web services that conform to the REST architecture is called RESTful web services.

2. *Mention what are the HTTP methods supported by REST?*

GET, PUT, POST and DELETE

3. *Explain the architectural style for creating web API?*

4. *Explain the RESTful Web Service?*

Restful Web Service is a lightweight, maintainable, and scalable service that is built on the REST architecture. Restful Web Service, expose API from your application in a secure, uniform, stateless manner to the calling client.

5. *Explain what is a "Resource" in REST?*

REST architecture treats every content as a resource. These resources can be Text Files, Html Pages, Images, Videos or Dynamic Business Data. Each resource is identified by URIs. The most popular representations of resources in REST are XML and JSON.

6. *Which protocol is used by RESTful web services?*

HTTP protocol

7. *What is messaging in RESTful web services?*

A client sends a message in form of a HTTP Request and the server responds in the form of an HTTP Response. This technique is called messaging.

8. *State the core components of an HTTP Request?*

- **Verb** – Indicates the HTTP methods such as GET, POST, DELETE, PUT, etc.
- **URI** – to identify the resource on the server.
- **HTTP Version** – Indicates the HTTP version. For example, HTTP v1.1.
- **Request Header** – Contains metadata for the HTTP Request message as key-value pairs. For example, client (or browser) type, format supported by the client, format of the message body, cache settings, etc.
- **Request Body** – Message content

9. *State the core components of an HTTP response?*

- **Status/Response Code** – Indicates the Server status for the requested resource. For example, 404 means resource not found and 200 means response is ok.
- **HTTP Version** – Same as request component

- **Response Header** – Contains metadata for the HTTP Response message as keyvalue pairs. For example, content length, content type, response date, server type, etc.
- **Response Body** – Response message content

10. *What do you understand about payload in RESTful web service?*

The term *payload* refers to XML or JSON-formatted text that is either posted via an http POST to a web service when a user creates a resource or returned from a web service via an http GET when a user requests a resource.

11. *Explain the caching mechanism?*

Caching refers to storing the server response in the client itself, so that a client need not make a server request for the same resource again and again.

12. *List the main differences between SOAP and REST?*

SOAP has a standardized protocol with predefined rules to follow whereas REST has an architectural style with loose guidelines and is often referred to as the rule breaker. SOAP's message format only supports XML whereas REST supports HTML, XML, JSON and others. Main advantages of SOAP is that it has high security, standardized procedures. On the other hand, REST has better performance, Scalability and flexibility

13. *Enlist advantages and disadvantages of 'Statelessness'.*

Advantages

- Statelessness helps in scaling the APIs to millions of concurrent users by deploying it to multiple servers.
- Being stateless makes REST APIs less complex
- A stateless API is easy to cache.

Disadvantage

- Web services need to get extra information in each request and then interpret to get the client's state.

Object Oriented Programming Fundamentals

1. *What is the main difference between a class and an object?*

A class is used to bind data as well as methods together as a single unit. Object acts like a variable of the class

2. *What is Encapsulation? Explain with a used case*

It describes the idea of bundling data and methods within one unit, for e.g., a class in Java.

3. *What is Polymorphism? Explain with a used case*

Polymorphism is the ability of an object to take on many forms. The most common use of polymorphism in OOP occurs when a parent class reference is used to refer to a child class object.

4. *Explain Overriding & Overloading and its advantages*

Overriding enables a child class to provide different implementation for a method that is already defined and/or implemented in its parent class or one of its parent classes. The overridden method in the child class should have the same name and parameters as the one in its parent class. For e.g.:

We have two classes: A child class Boy and a parent class Father.

The boy class extends Father class. Both the classes have a common method void sleep. Boy class is giving its own implementation to the sleep() method and is overriding the sleep method in parent class. Advantage is that this provides multiple implementation of the same method and can invoke parent class overridden method using super keyword.

Overloading allows different methods to have the same name, but different number of input parameters or type of input parameters or both. E.g.: add (int, int) and add (int, int, char). Overloading is related to compile-time polymorphism. An advantage is that If we have to perform only one operation such as addition, having same name of the methods like 'add' increases the readability of the program.

5. *What is Inheritance and different types of inheritance? Explain with a used case*

Inheritance is a mechanism in which one class acquires the property of another class. For example, a child inherits the traits of his/her parents. With inheritance, we can reuse the fields and methods of the existing class. The different types of inheritance are single inheritance, multiple inheritance, multilevel inheritance, hierarchical inheritance and hybrid inheritance

6. *What is an abstract class?*

A class which is declared as abstract is known as an abstract class. It can have abstract and non-abstract methods. It needs to be extended and its method implemented. It cannot be instantiated.

7. *What is an interface and how multiple inheritance is achieved with this?*

An interface contains variables and methods like a class but the methods in an interface are abstract by default unlike a class. Multiple inheritance by interface occurs if a class implements multiple interfaces or also if an interface itself extends multiple interfaces.

8. *What are the access modifiers?*

Access modifiers are keywords in object-oriented languages that set the accessibility of classes and methods.

9. *What are the various types of constructors?*

Default constructor, No argument constructor, and parameterized constructors

10. *What is 'this' pointer?*

THIS works as a reference to the current object, whose method or constructor is being invoked

11. What is static and dynamic Binding?

Static binding is a binding in which name can be associated with the class during compilation time and Dynamic binding is a binding in which name can be associated with the class during execution time.

12. How many instances can be created for an abstract class and why?

None, because the purpose of an abstract class is to function as a base for subclasses. It acts like a template, or an empty or partially empty structure, you should extend it and build on it before you can use it.

13. Which OOPS concept is used as a reuse mechanism and explain with a use case

Inheritance is the concept that can be used as reuse mechanism. For e.g.: when a child class extends the properties or methods of a parent class using the inheritance feature, the same need not be defined again in the child class

14. Please identify one practical scenario for each pillar of OOPs.

Abstraction: When we are making a call it only shows about the numbers and display that in screen, we really do not know how this connect with other number.

Encapsulation: When we turn on the Bluetooth, we can assess to connect to other phone but not access to call or send sms from that phone

Polymorphism: We can turn on the phone using the lock button, at the same time we can also turn off the phone using the same button.

Inheritance: Features of a phone can be inherited by Samsung and further the features of Samsung can be inherited by Samsung S5