**Introduction to API**

1. Explain REST and RESTFUL? REST is a style of software architecture whilst RESTful refers to web services implementing such an architecture.
2. Mention what are the HTTP methods supported by REST? GET, POST, PUT, DELETE, HEAD
3. Explain the architectural style for creating web API? Unsure about the question
4. Explain the RESTFul Web Service? data and functionality are accessed using Uniform Resource Identifiers (URIs). Clients and servers exchange representations of resources by using a standardized interface and protoco
5. Explain what is a “Resource” in REST? An object with a type, associated data, relationships to other resources, and a set of methods that operate on it.
6. Which protocol is used by RESTful web services? HTTP
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.
8. State the core components of an HTTP Request? Verb, Uniform Resource Identifier , version, header, body
9. State the core components of an HTTP response? Version, header, body
10. What do you understand about payload in RESTFul web service? the part of that response that is communicating directly to you (like JSON data)
11. Explain the caching mechanism? By caching you are storing information in the client so that the client doesn’t need to request it from the server anymore.
12. List the main differences between SOAP and REST? REST has no official standard at all because it is an architectural style. SOAP API, on the other hand, has an official standard because it is a protocol. REST APIs uses multiple standards like HTTP, JSON, URL, and XML while SOAP APIs is largely based on HTTP and XML.
13. Enlist advantages and disadvantages of ‘Statelessness’:   
    pros : Web services can treat each request independently. Web services don’t need to maintain the client's previous interactions. It simplifies the design of the application. As HTTP is itself a statelessness protocol, RESTful Web Services work seamlessly with the HTTP protocols.  
    cons: Web services need to get extra information in each request and then interpret to get the client's state in case the client interactions are to be taken care of.