1) Explain what is REST and RESTFUL?

REST represents REpresentational State Transfer; it is a relatively new aspect of writing web API.

RESTFUL is referred for web services written by applying REST architectural concept are called RESTful services, it focuses on system resources and how state of resource should be transported over HTTP protocol to different clients written in different language. In RESTFUL web service HTTP methods like GET, POST, PUT and DELETE can be used to perform CRUD operations.

2) Explain the architectural style for creating web API?

The architectural style for creating web api are

- HTTP for client server communication
- XML/JSON as formatting language
- Simple URI as the address for the services
- Stateless communication

3) Mention what tools are required to test your web API?

SOAPUI tool for SOAP WS and Firefox "poster" plugin for RESTFUL services.

4) Mention what are the HTTP methods supported by REST?

HTTP methods supported by REST are:

- **GET:** It requests a resource at the request URL. It should not contain a request body as it will be discarded. Maybe it can be cached locally or on the server.
- **POST:** It submits information to the service for processing; it should typically return the modified or new resource
- **PUT:** At the request URL it update the resource
- **DELETE:** At the request URL it removes the resource
- **OPTIONS:** It indicates which techniques are supported
- **HEAD:** About the request URL it returns meta information

5) Mention whether you can use GET request instead of PUT to create a resource?

No, you are not supposed to use PUT for GET. GET operations should only have view rights, while PUT resource is used for updating a data.

6) Mention what are resources in a REST architecture?

Resources are identified by logical URLs; it is the key element of a RESTful design. Unlike, SOAP web services in REST, you view the product data as a resource and this resource should contain all the required information.

7) Mention what is the difference between AJAX and REST?

AJAX	REST
• In Ajax, the request are sent to the server by using XMLHttpRequest objects. The response is used by the JavaScript code to dynamically alter the current page • Ajax is a set of technology; it is a technique of dynamically updating parts of UI without having to reload the page • Ajax eliminates the interaction between the customer and server asynchronously	• REST have a URL structure and a request/response pattern the revolve around the use of resources • REST is a type of software architecture and a method for users to request data or information from servers • REST requires the interaction between the customer and server

7) Mention some key characteristics of REST?

Some key characteristics of REST includes

- REST is stateless, therefore the SERVER has no state (or session data)
- With a well-applied REST API, the server could be restarted between two calls as every data is passed to the server
- Web service mostly uses POST method to make operations, whereas REST uses GET to access resources

9) Explain how JAXB related to RESTful web API?

JAXB stands for java arch for XML binding.

10) Mention what is the difference between PUT and POST?

"PUT" puts a file or resource at a particular URI and exactly at that URI. If there is already a file or resource at that URI, PUT changes that file or resource. If there is no resource or file there, PUT makes one

POST sends data to a particular URI and expects the resource at that URI to deal with the request. The web server at this point can decide what to do with the data in the context of specified resource

PUT is idempotent meaning, invoking it any number of times will not have an impact on resources.

However, POST is not idempotent, meaning if you invoke POST multiple times it keeps creating more resources

11) Mention which markup language can be used in restful web api?

JSON and XML are the two markup language that can be used in restful web api

Mention what is the difference between SOAP and REST?

SOAP	REST
• SOAP is a protocol through which two computer communicates by sharing XML document • SOAP permits only XML • SOAP based reads cannot be cached • SOAP is like custom desktop application, closely connected to the server • SOAP is slower than REST • It runs on HTTP but envelopes the message	• Rest is a service architecture and design for network-based software architectures • REST supports many different data formats • REST reads can be cached • A REST client is more like a browser; it knows how to standardized methods and an application has to fit inside it • REST is faster than SOAP • It uses the HTTP headers to hold meta information

What is purpose of a URI in REST based webservices?

URI stands for *Uniform Resource Identifier*. Each resource in REST architecture is identified by its URI. Purpose of an URI is to locate a resource(s) on the server hosting the web service.

A URI is of following format:

<protocol>://<service-name>/<ResourceType>/<ResourceID>

What are advantages of REST web services?

Some of the advantages of REST web services are:

- Learning curve is easy since it works on HTTP protocol
- Supports multiple technologies for data transfer such as text, xml, json, image etc.
- No contract defined between server and client, so loosely coupled implementation.
- REST is a lightweight protocol
- REST methods can be tested easily over browser.

What are disadvantages of REST web services?

Some of the disadvantages of REST are:

- Since there is no contract defined between service and client, it has to be communicated through other means such as documentation or emails.
- Since it works on HTTP, there can't be asynchronous calls.
- Sessions can't be maintained

What are the best practices to create a standard URI for a web service?

Following are important points to be considered while designing a URI:

- **Use Plural Noun** − Use plural noun to define resources. For example, we've used users to identify users as a resource.
- **Avoid using spaces** − Use underscore(_) or hyphen(-) when using a long resource name, for example, use authorized_users instead of authorized%20users.
- **Use lowercase letters** − Although URI is case-insensitive, it is good practice to keep url in lower case letters only.

- **Maintain Backward Compatibility** − As Web Service is a public service, a URI once made public should always be available. In case, URI gets updated, redirect the older URI to new URI using HTTP Status code, 300.
- **Use HTTP Verb** − Always use HTTP Verb like GET, PUT, and DELETE to do the operations on the resource. It is not good to use operations names in URI.

What is the purpose of HTTP Status Code?

HTTP Status code are standard codes and refers to predefined status of task done at server. For example, HTTP Status 404 states that requested resource is not present on server.

Consider following status codes:

- **200** - OK, shows success.
- **201** - CREATED, when a resource is successful created using POST or PUT request. Return link to newly created resource using location header.
- **304** - NOT MODIFIED, used to reduce network bandwidth usage in case of conditional GET requests. Response body should be empty. Headers should have date, location etc.
- **400** - BAD REQUEST, states that invalid input is provided e.g. validation error, missing data.
- **401** - FORBIDDEN, states that user is not having access to method being used for example, delete access without admin rights.
- **404** - NOT FOUND, states that method is not available.
- **409** - CONFLICT, states conflict situation while executing the method for example, adding duplicate entry.
- **500** - INTERNAL SERVER ERROR, states that server has thrown some exception while executing the method