1. What is Web-Services?

**Answer**:- Web-Services are XML based information exchange system that uses Internet for direct application-to-application interaction. These systems can include programs, objects, documents, or messages.

2. What is REST?

**Answer**:- REST stands for (Representational State Transfer). It is an architectural style of creating, retrieving, manipulating deleting the resources through web services which uses the advantage of ubiquity (the state of being everywhere at once) of HTTP protocol and uses HTTP methods to define actions.

3. What is RESTFul Web Service?

**Answer**: - There are two popular ways to develop web services using SOAP (Simple Object Access Protocol) which is XML based way to expose web services and second REST based web services which uses HTTP protocol. Web-Services developed using REST style also known as RESTFul web services.

4. What is SOAP?

**Answer: -** SOAP stands for Simple Object Access Protocol. It is a XML based protocol for accessing web Services. SOAP is a standard recommendation from W3C for interaction between two applications. It is platform and language independent. We can say that SOAP is a XML way of defining what information has been sent and how.

5. What is the difference between SOAP and REST?

**Answer**: - The main difference between SOAP and REST is that former provides a standard of communication between client, server and other third parties with restricted set of rules and format. While REST is an architectural style which takes the full advantage of HTTP Protocol, in both client and server in order to allow communication. In short we can say that getting data from RESTFul web services is easier than getting a data from SOAP. Since everybody is familiar with HTTP requests like GET or POST, it is easy to understand and correlate how RESTFul web services are working and which URL of REST web service provides what kind of information. In SOAP we need to understand lengthy WSDL document to find out right methods and right way to call them.

Suppose you want to retrieve todays weather of a particular city from a server which is giving weather related information. Your RESTFul URL will look something like: <http://weatherdata.org/data/weather/uk/london> , which is looking very similar to HTTP request like <http://weatherdata.org/data/weather?q=uk,london> . On the other hand, if you want to get same piece of data from SOAP you have to create XML message with header and body and send it. <http://www.webservicex.net/globalweather.asmx?op=GetWeather> . So RESTFul web services are much simpler, expressive and flexible compared to SOAP.

6. What is the use HTTP Protocol in REST?

**Answer**: - REST takes full advantage of HTTP Protocol, including methods e.g. GET, POST, PUT, DELETE to represent action e.g. from an application which provides books related data, GET can be used to retrieve books, POST can be used to upload the data of new book, and DELETE can be used to remove the book from library. On the other hand, SOAP uses XML messages to communicate with the server.

7. What are the supported formats of REST and SOAP?

**Answer**: - REST web service can return response in multiple formats e.g. JSON, XML and HTML while by using SOAP web services you have to tie your response with XML because your response is bundled inside SOAP message which is always in XML format.

8. What is HTTP Basic Authentication and how it works?

**Answer**: - HTTP Basic Authentication is simple challenge and response mechanism in which a server can request authentication information (like user-id and password) from a client. The client passes the authentication information to the server in an authorization header. The whole authentication process used to be done in four simple steps: -

1. First HTTP client (Web Browser) makes the request to the web server. Request method doesn’t have to be GET it can be any method.
2. If web server sees that the requested resource needs authentication to access web resource, then it sends back 401 unauthorized status code along with WWW-Authenticate header.
3. And then client displays dialog box to take username and password as input. Once the credential enters the client sends it using authorization header.
4. If the credentials are correct then server responds with 200 status code and authorization-info header. If client sends wrong credentials in the authorization request, then server again respond with 401 status code. The client is allowed to try again and again.

9. Can you tell me which API can be used to develop RESTFul web services in Java?

**Answer**: - There are many framework and API out there which helps to develop REST web services in java including JAX-RS which is standard way to develop REST web services. Jersey is one of the important implementation of JAX-RS which offers more than specification recommends. Then you also have RESTEasy, RESTlet and Apache CFX. If you like Scala, then you can also use Play framework to develop REST web services.

10. How do you configure RESTFul web service?

**Answer**: - Jersey is the open source implementation of java JAX-RS specification. It provides Java library using which we can easily create RESTFul web services in java platform. JAX-RS/Jersey supports JAXB based XML binding. JAXB provides API to access and process XML documents. Following are the configuration steps for RESTFul web service: -

1. Create New Dynamic Web Project with web.xml deployment descriptor.
2. Add JAX-RS/Jersey dependent Jar Files in the lib folder.
3. Create source file in src folder.
4. Define jersey servlet dispatcher. We need to register jersey as the servlet dispatcher for REST requests in web.xml
5. Run your web service.

11. How do you apply security in RESTFul web services?

**Answer**: - There are multiple ways to secure RESTFul API in Java. Some of these are following: -

1. **BASIC Authentication**: - It’s simplest of all techniques and probably most used as well. You use login/password form it’s a simple authentication, you input your username and password and submit the form to server, and application identify you as a user either you are allowed to use the system else you will get error.

The main problem with this security implementation is that your credentials are propagating in a plain way from the client to server. Credentials are encoded with Base64 transit but not encrypted in any way. This way any sniffer could read the packages over the network.

1. **DIGEST Authentication**: - This authentication method makes use of a hashing algorithm to encrypt password (called password hash) entered by the user before sending it to the server. This, obviously makes it much safer than basic authentication where password travels in plain text which can be easily read by whoever intercept it. There are many such hashing algorithm in java also like MD5, SHA, BCrypt, SCrypt etc.

But, once this password hash is generated and stored in a database, you cannot convert it back to the original one. Each time user login into application you have to regenerate the password hash again, and match with password hash stored in a database.

1. **CLIENT CERT Authentication**: - This is a mechanism in which a trust agreement is establish between client and server through certificates. They must be signed by an agency established to ensure that the certificate provided for authentication is legitimate which is known as CA.

Using this technique, when the client attempts to access a protected resource, instead of providing username and password, it presents the certificate to the server. The certificate contains the user information for authentication including security credentials, besides a unique private public key pair. The server then determines if the user is legitimate through CA. Additionally it must verify whether the user has access to that resource.

1. **Using API Keys**: - If you have ever developed applications which interact with other applications over the cloud e.g. Facebook integration then you have already used this. They require you to provide API key and API secret to rightly identify you. These API key and secret are some random encoded string which is impossible to guess.

12. How you maintain session in RESTFul services?

**Answer**: - As per REST architecture, a RESTFul web service should not keep a client state on server. This restriction is called statelessness. It is the responsibility of client to pass its context to server and then server can store this context to process client’s further request. For example, session maintained by server is identified by session identifier passed by the client. RESTFul web service should stick to this restriction Web service method are not storing any information from the client they are invoked from. So we typically don’t want session management in REST because our interface is no longer stateless.

13. What is WADL in RESTFul services?

**Answer**: - Jersey contains support for WADL (Web Application Description Language). WADL is a XML description of a deployed RESTFul web application. It contains model of the deployed resources, their structure, supported media types, HTTP methods and so on. In other words, we can say that WADL is similar to WSDL (Web Service Description Language) which describes SOAP web services. WADL is specifically designed to support RESTFul Web resources.

14. What do you understand by payload in RESTFul?

**Answer**: - Payload means data which passed inside request body also payload is not request parameters. So only you can do payload in POST and not in GET and DELETE method. The term ‘payload ‘is used to distinguish between the ‘interesting’ information from chunk of data and the overhead to support it. For e.g. A tanker truck can carry 20 tons of oil but the fully loaded vehicle weighs much more than that- there’s the vehicle itself, the driver, the fuel, the tank etc. It costs money to move all these but the customer only cares about (and pays for) the oil, hence ‘pay-load’.

15. How much maximum payload you could do in POST method?

**Answer**: - Theoretically we can pass unlimited data as a payload to POST method but we need to take practical things into account e.g. sending POST with large payload will consume more bandwidth, take more time and present performance challenge to your server.

16. What is the difference between SOAP and RESTFul web services?

**Answer**: - There are many difference between these two styles of web services e.g. SOAP takes more bandwidth because of heavy weight XML based protocol but REST takes less bandwidth because of popular use of JSON as a message protocol and leveraging HTTP method to define actions. This is also means that REST is faster than SOAP based web services.

17. If you have to develop web services which one you will choose SOAP or RESTFul and why?

**Answer**: - RESTFul.

18. What framework you had used to develop RESTFul web services?

**Answer**: - Jersey.