**Q1. What is Web Service?**

Web Service is a web component that transfer data between client and servers, client send request and receive response from the server. It can we access over internet, it is mainly using to communicate with the web-based application through xml-messaging concept.

The Web Service is a standard software system used for communication between two devices (client and server) over the network. Web services provide a common platform for various applications written in different languages to communicate with each other over the network.

Component of the web services

SOAP: Simple Object Access Protocol

UDDI: universal description discovery and Integration

WSDL: Web Service Description Language

RDF: resource Description Framework

XML-Extensive Markup Lanugage

Advantage of the web services:

1. Interoperability: with the help of web services, an application can communicate with other application developed in any language.
2. Reusability: We can expose the web service so that other application can use it.
3. Modularity: With the help of web service we can create a service for specific task.
4. A Standard protocol for every application
5. Cheaper cost for communication

**Q2) Explain web service architecture?**

Ans: Web service framework consists of an architecture which consists of three different layers. The roles of these layers are defined as below

Service Provider: As the name denotes, service provider role is to create the web service and makes it accessible to the client applications over the internet for their usage.

Service Requestor: Service requestor is basically any consumer of web service like any client application. Client applications are written in any language contact web service for any type of functionality by sending XML request over the available network connection.

Service Registry: Service registry is the centralized directory which helps locate web services for client applications. Here we can find the existing web services, as well as developers, can also create the new one.

The Service Provider uses the ‘Publish’ interface of Service Registry to make the existing web services available to client applications. With all the information provided by the service registry, service requestor can bind or invoke services.

**Q4. Web service protocol stack and its layers**

It consists of 4 layers

1. Service Transport
2. XML-messaging
3. Service Description
4. Service Discovery

Service Transport: this is the first layer which helps in transporting xml messages between client application, this layer commonly uses the below mentioned protocol

HTTP, SMTP, FTP, BEEP

XML-Messaging: This layer is based on xml model, where messages are encoded in common xml format which is easily understand by others. This layer includes

XML-RPC, SOAP

Service Description: This layer contains description like, location, available functions and data transfer for xml-messaging which describes the public interface to a specific web services.

WSDL

Service Discovery: This layer is responsible for providing a way to publish and find web service over the web. This layer includes

UDDI

**Q4. What are the different types of web services?**

There are two types of web services:

SOAP - It is an XML-based protocol for accessing web services.

RESTful - It is an architectural style, not a protocol.

**Q5. What are the main features of web services?**

It is available over the Internet or private (intranet) networks.

It uses a standardized XML messaging system.

It is not tied to any one operating system or programming language.

It is self-describing via a common XML grammar.

**Q6. What is SOAP?**

The SOAP stands for Simple Object Access Protocol. It is an XML-based protocol for accessing web services. It is platform independent and language independent. By using SOAP, you can interact with other programming language applications.

**Q7. What are the advantages of SOAP web services?**

WS Security - SOAP defines its security known as WS Security.

Language Independent - Its web services can be written in any programming language

Platform Independent - Its web services can be executed on any platform.

**Q8. What are the disadvantages of SOAP web services?**

Slow - It uses XML format that must be parsed to be read and defines many standards that must be followed while developing the SOAP applications. So it is slow and consumes more bandwidth and resource.

WSDL Dependent - It uses WSDL and doesn't have any other mechanism to discover the service.

**Q9. What is RESTful web services?**

The REST stands for Representational State Transfer. It is an architectural style. It is not a protocol like SOAP.

**Q10. What are the advantages of RESTful web services?**

These are some of the important advantages of RESTful web services:

Fast - The Web Services are fast because there is no strict specification of SOAP. It consumes less bandwidth and resource.

Language Independent - The web services can be written in any programming language.

Platform Independent - The web services can be executed on any platform.

Allows different data format - The web service permits different data format such as Plain Text, HTML, XML, and JSON.

**Q11. What is SOA?**

SOA stands for Service Oriented Architecture. It is a design pattern to provide services to other application through protocol.

**Q12. What is the difference between SOAP and REST web services?**

**SOAP** **REST**

Meaning Simple Object Access Protocol Representational State Transfer

Design Standardized protocol with pre-defined rules to follow. Architectural style with loose guidelines and recommendations.

Approach Function-driven (data available as services, e.g.: “getUser”) Data-driven (data available as resources, e.g. “user”).

Statefulness Stateless by default, but it’s possible to make a SOAP API stateful. Stateless (no server-side sessions).

Caching API calls cannot be cached. API calls can be cached.

Security WS-Security with SSL support. Built-in ACID compliance. Supports HTTPS and SSL.

Performance Requires more bandwidth and computing power. Requires fewer resources.

Message format Only XML. Plain text, HTML, XML, JSON, YAML, and others.

Transfer protocol(s) HTTP, SMTP, UDP, and others. Only HTTP

Recommended for Enterprise apps, high-security apps, distributed environment, financial services, payment gateways, telecommunication services. Public APIs for web services, mobile services, social networks.

Advantages High security, standardized, extensibility. Scalability, better performance, browser-friendliness, flexibility.

Disadvantages Poorer performance, more complexity, less flexibility. Less security, not suitable for distributed environments.

**Q13. what is rest URIs?**

REST APIs use Uniform Resource Identifiers (URIs) to address resources.

REST-based Web services are organized into resources. A resource is a chunk of related information, such as a user profile, a collection of updates (activities), or a global user ID (GUID). Each resource is identified by one or more Uniform Resource Identifiers (URIs). To access the resource, an application calls an HTTP operation on one of the resource's URIs.

URI SYNTAX

The URIs for Yahoo REST-based Web services have the following syntax:

https://{service}.yahooapis.com/{version}/{resourcepath}

URI Examples

The following URI defines the profile resource for the user whose ID is 12345.

https://social.yahooapis.com/v1/user/12345/profile

The next URI specifies the connections (friends) for the user whose ID is 6677.

https://social.yahooapis.com/v1/user/6677/connections

The following URI accesses the collection of schools contained in the profile of the user of ID 98765.

https://social.yahooapis.com/v1/user/98765/profile/schools

**Q14. What is the difference between Http Request Headers and Request Parameters?**

Http Request Headers

Holds the browser supplied input values.

Headers names are fixed but the values are browser specified.

Mandatory in every request.

Headers names are unique.

List of http request headers: Accept-Encoding, Cookie, User-agent Authorization etc.

Note: There may be additional headers in the request but depends on the service.

Request Parameters or Query parameter:

Parameters names and values are user-defined. but normally are a Query parameter are key-value pairs (separated by an equal sign), with the sequence separated by either a semicolon or an ampersand. (sno=101 & sname=thakur). The query parameters are sometimes referred to as optional parameters

http://ecomputernotes.com/servlet/intro/httprequestheaders-vs-request-parameters

**Q15. What is the difference in Authentication and Authorization and what are types of Authentication we used in web services?**

Authentication is used to determine who the user of an API is.

Authorization is used to determine what resources the identified user has access to.

Form-based authentication - Web/HTML based authentication that commonly uses HTTP cookies.

Basic/Digest/NTLM authentication - Uses HTTP headers to identify users.

OAuth 1.x/2 - HTTP-based interactions and flows that authorize usage of HTTP resources (API, Web, etc). OAuth indirectly includes a step for authentication but makes no claims on how that authentication should be done.

**Q16. What are the major obstacle users faced when using soap?**

Ans: When using SOAP, users often see the firewall security mechanism as the biggest obstacle. This blocks all the port leaving few http port 80 and the http ports used by SOAP that bypasses the Firewall.

**Q17. What are the primary security issues of web service?**

Ans: To ensure reliable transactions and secure confidential information, web services require very high level of security which can be only achieved through Entrust Secure Transaction Platform. Security issues for web services are broadly divided into three sections as described below

1) Confidentiality: A single web service can have multiple applications and their service path contains a potential weak link at its nodes. Whenever messages or say XML requests are sent by the client along with the service path to the server, they must be encrypted. Thus, maintaining the confidentiality of the communication is a must.

2) Authentication: Authentication is basically performed to verify the identity of the users as well as ensuring that the user using the web service has the right to use or not? Authentication is also done to track user’s activity. There are several options that can be considered for this purpose

Application level authentication

HTTP digest and HTTP basic authentication

Client certificates

**3) Network Security:** This is a serious issue which requires tools to filter web service traffic.

**Q18. Explain the term statelessness in terms of RESTful web services?**

Ans: In REST architecture, there is a restriction where a REST web service is not allowed to keep a client state or session on the server. Such condition is known as ‘Statelessness’. In such situation, the client passes its context to the server and in turn, the server stores the context in order to process client’s further requests.

**Q19. Enlist advantages and disadvantages of statelessness?**

The advantages of statelessness include

Each method request is treated independently.

Application design is simplified as it does not maintain client’s previous interaction.

Works with HTTP protocol as it shares the feature of being statelessness.

The disadvantage of statelessness includes

Every time client interaction takes place, web services are to be provided with extra information about each request so that they can interpret the client’s state.

**Q20. What is WSDL?**

The WSDL stands for Web Services Description Language. It is an XML document containing information about web services such as method name, method parameter. The Client needs a data dictionary which contains information about all the web services with methods names and parameters list to invoke them for the web services. The Web Service Description Language bridge up this gap, by providing all necessary information to the client

**Q21. what are WSDL Document Elements?**

A WSDL document has a definitions element that contains the other five elements, types, message, portType, binding and service.

Types– This define the message datatype which are in the form of xml schema (xsd), used by the webservice.

Message– This define the data element for each operation where messages could be the entire document or an argument that is to be happened.

Port Type–There are multiple services present in WSDL, Port-type defines the collection of operations that can be performed for binding.

Binding– Determine and define the protocol and data format for each port type.

Operations: This define the operation performed for a message to process the message

Service– a collection of related endpoints.

<wsdl:definitions

xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"

xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"

xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"

xmlns:tns="http://Example.org"

xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"

xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"

xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy"

xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:msc="http://schemas.microsoft.com/ws/2005/12/wsdl/contract"

xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"

xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"

xmlns:wsa10="http://www.w3.org/2005/08/addressing"

xmlns:wsx="http://schemas.xmlsoap.org/ws/2004/09/mex" targetNamespace="http://Example.org"

xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">

<wsdl:types>

<xsd:schema targetNamespace="http://Example.org" elementFormDefault="qualified" >

<xsd:element name="Add">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" name="a" type="xsd:int" />

<xsd:element minOccurs="0" name="b" type="xsd:int" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="AddResponse">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" name="result" type="xsd:int" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="Subtract">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" name="a" type="xsd:int" />

<xsd:element minOccurs="0" name="b" type="xsd:int" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

<xsd:element name="SubtractResponse">

<xsd:complexType>

<xsd:sequence>

<xsd:element minOccurs="0" name="result" type="xsd:int" />

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>

</wsdl:types>

<wsdl:message name="ICalculator\_Add\_InputMessage">

<wsdl:part name="parameters" element="tns:Add" />

</wsdl:message>

<wsdl:message name="ICalculator\_Add\_OutputMessage">

<wsdl:part name="parameters" element="tns:AddResponse" />

</wsdl:message>

<wsdl:message name="ICalculator\_Subtract\_InputMessage">

<wsdl:part name="parameters" element="tns:Subtract" />

</wsdl:message>

<wsdl:message name="ICalculator\_Subtract\_OutputMessage">

<wsdl:part name="parameters" element="tns:SubtractResponse" />

</wsdl:message>

<wsdl:portType name="ICalculator">

<wsdl:operation name="Add">

<wsdl:input wsaw:Action="http://Example.org/ICalculator/Add" message="tns:ICalculator\_Add\_InputMessage" />

<wsdl:output wsaw:Action="http://Example.org/ICalculator/AddResponse" message="tns:ICalculator\_Add\_OutputMessage" />

</wsdl:operation>

<wsdl:operation name="Subtract">

<wsdl:input wsaw:Action="http://Example.org/ICalculator/Subtract" message="tns:ICalculator\_Subtract\_InputMessage" />

<wsdl:output wsaw:Action="http://Example.org/ICalculator/SubtractResponse" message="tns:ICalculator\_Subtract\_OutputMessage" />

</wsdl:operation>

</wsdl:portType>

<wsdl:binding name="DefaultBinding\_ICalculator" type="tns:ICalculator">

<soap:binding transport="http://schemas.xmlsoap.org/soap/http" />

<wsdl:operation name="Add">

<soap:operation soapAction="http://Example.org/ICalculator/Add" style="document" />

<wsdl:input>

<soap:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap:body use="literal" />

</wsdl:output>

</wsdl:operation>

<wsdl:operation name="Subtract">

<soap:operation soapAction="http://Example.org/ICalculator/Subtract" style="document" />

<wsdl:input>

<soap:body use="literal" />

</wsdl:input>

<wsdl:output>

<soap:body use="literal" />

</wsdl:output>

</wsdl:operation>

</wsdl:binding>

<wsdl:service name="CalculatorService">

<wsdl:port name="ICalculator" binding="tns:DefaultBinding\_ICalculator">

<soap:address location="http://Example.org/ICalculator" />

</wsdl:port>

</wsdl:service>

</wsdl:definitions>

**Q22. Explain The SOAP and RESTFul Web Service?**

Mostly, there are two kinds of Web Services which are quite popular.

# SOAP (Simple Object Access Protocol) which is an XML-based way to expose web services.

# Web services developed using REST style (Representational State Transfer)are known as RESTful web services.

These web services use HTTP methods to implement the concept of REST architecture.

A RESTful web service usually defines a URI, Uniform Resource Identifier a service, provides resource representation such as JSON and set of HTTP Methods.

HTTP verbs: GET, POST, PUT, and DELETE.

**Q23. when do we use SOAP and Rest service?**

SOPA Used for : Enterprise apps, high-security apps, distributed environment, financial services, payment gateways, telecommunication services.

Rest USed for : Public APIs for web services, mobile services, social networks.

**Q24. what are transfer protocol used in SOAP and Rest**

SOAP : HTTP, SMTP, UDP, and others

Rest: Only HTTP

**Q25. What are the advantage of using SOAP and Rest services?**

SOAP: High security, standardized, extensibility

Rest: Scalability, better performance, browser-friendliness, flexibility.

**Q26. what are the SOAP Document elements.**

Envelope (required) – This is the starting and ending tags of the message.

Header (optional) – It contains the optional attributes of the message. It allows you to extend a SOAP message in a modular and decentralized way.

Body (required) – It contains the XML data that the server transmits to the receiver.

Fault (optional) – It carries information about errors occurring during processing the message.

**Q27. What are the HTTP Status Code:**

CATEGORY DESCRIPTION

1xx: Informational Communicates transfer protocol-level information.

2xx: Success Indicates that the client’s request was accepted successfully.

3xx: Redirection Indicates that the client must take some additional action in order to complete their request.

4xx: Client Error This category of error status codes points the finger at clients.

5xx: Server Error The server takes responsibility for these error status codes.

https://restfulapi.net/http-status-codes/

**Q28. What is an API?**

An API (Application Programming Interface) that enables two applications to communicate with each other.

API examples,

Google Maps API, Amazon Advertising API, Twitter API, YouTube API, etc

**Q29. What are main differences between API and Web Service?**

All Web services are APIs but not all APIs are Web services.

Web services might not contain all the specifications and cannot perform all the tasks that APIs would perform.

A Web service uses only three styles of use: SOAP, REST for communication whereas API may be exposed to in multiple ways.

A Web service always needs a network to operate while APIs don’t need a network for operation.

**Q30. What are principles of an API test design? or what is the process to write the API test case? or What is the procedure to perform API testing?**

The five most important principles of an API test design are:

Setup: Create objects, start services, initialize data, etc

Execution: Steps to apply API or the scenario, including logging

Verification: evaluate the result of the execution

Reporting: Pass, failed or blocked

Clean up: Pre-test state or data created

**Q31. What must be checked when performing API testing?**

Accuracy of data

HTTP status codes

different Data type validation

Authorization checks

Implementation of response timeout

Error codes in case API returns, and

Non-functional testing like performance and security testing

**Q32. What are the testing methods that come under API testing? or What are the common API testing types?**

API testing generally involves following testing methods:

Unit testing and Functional testing

Load testing for testing the performance under load.

Discovery testing for listing, creating and deleting the number of calls that has been documented in API.

Usability testing and Reliability testing for obtaining consistent results.

Security testing and Penetration testing for validating all types of authentication.

Automation testing for creating and executing scripts that require API calls execution regularly.

End to end Integration testing and Web UI testing.

API documentation testing for determining its efficiency and effectiveness.

**Q33. What all challenges are included under API testing?**

The first and foremost challenge is selecting an appropriate parameter and then its combination.

Parameter categorization

Proper sequencing of call is required as this may lead to inadequate coverage in testing.

Output verification and validation

Another important challenge is providing input values, which is very difficult as GUI is not available in this case.

**Q34. What are the types of Issues observed while performing API testing?**

Find below the list of such issues/defects:

Reliability issues with respect to connection with other APIs

Inconsistent or absence of error handling mechanism

Repetition or redundancy of the functionalities

Missing required functionality in some cases

Passing incorrect argument to the input values

Improper messaging

Stress and performance issues

**Q35. How is UI level testing different from API testing?**

The main consideration of the UI (User Interface) level testing is to test the graphical interface part of the application including the features like font, layout, etc.

Whereas, the main consideration of the API testing is establishing communication between different software systems and it mainly resides in business logic layer.

It never concentrates on the look of the application.

**Q36. What are the tools used for API test automation?**

SOAPUI

PARASOFT

JMETER

Postman - Chrome

Poster for Firefox browser.

**Q37. Can You Tell Us Which Java API Helps In Developing A RESTFul Web Service?**

JAX-RS library - is a standard way to develop a REST web service.

Jersey

Apache CFX.

RESTEasy

**Q38. Mention what are the HTTP methods ( or verbs)supported by REST?**

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

**Q39. Mention what is the difference between PUT and POST?**

GET /device-management/devices : Get all devices

POST /device-management/devices : Create a new device

GET /device-management/devices/{id} : Get the device information identified by "id"

PUT /device-management/devices/{id} : Update the device information identified by "id"

DELETE /device-management/devices/{id} : Delete device by "id"

PUT puts a file or resource at a particular URI and exactly at that URI. If the resource already exists, then PUT updates it. If it’s a first-time request, then PUT creates 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.

Note: Generally, in practice, always use PUT for UPDATE operations.Always use POST for CREATE operations.

**Q40. What are API documentation templates that are commonly used?**

Swagger - The ability of APIs to describe their own structure.

**Q41. what are n-Tier Architecture?**

Three-Tier Architecture: The Three-tier architecture is divided into three parts:Client system handles Presentation layer, Application server handles Application layer and Server system handles Database layer.

1. Presentation layer (Client Tier)

2. Application layer (Business Tier)

3. Database layer (Data Tier)

Two-tier architecture: Client system handles both Presentation and Application layers and Server system handles Database layer. It is also known as client server application. The communication takes place between the Client and the Server. Client system sends the request to the Server system and the Server system processes the request and sends back the data to the Client System

1. Client Application (Client Tier)

2. Database (Data Tier)

One Tier Architecture:One Tier application AKA Standalone application

One tier architecture has all the layers such as Presentation, Business, Data Access layers in a single software package. Applications which handles all the three tiers such as MP3 player, MS Office are come under one tier application. The data is stored in the local system or a shared drive.

Note: Another layer is N-Tier application. N-Tier application AKA Distributed application. It is similar to three tier architecture but number of application servers are increased and represented in individual tiers in order to distributed the business logic so that the logic will be distributed.

**Q41. Enlist the operation types response used in WSDL?**

Ans: WSDL basically defines 4 types of Operation type responses. These are enlisted below

One-way: Receives a message but does not return response.

Request-Response: Receives a request and return a response.

Solicit-Response: Sends a request and wait for a response.

Notification: Sends a message but does not wait for a response.

Note: Among these, Request-Response is the most common operation type.