

Web Services Interview Questions – SOAP, RESTful

APRIL 2, 2018 BY [PANKAJ](#) — [31 COMMENTS](#)



Welcome to the Web Services Interview Questions with detailed answers. Recently I have written a lot about web services. How we can create SOAP and RESTful web services in Java.

Web Services Interview Questions

1. [What is a Web Service?](#)
2. [What are the advantages of Web Services?](#)
3. [What are different types of Web Services?](#)
4. [What is SOAP?](#)
5. [What are advantages of SOAP Web Services?](#)
6. [What are disadvantages of SOAP Web Services?](#)
7. [What is WSDL?](#)
8. [What are different components of WSDL?](#)
9. [What is UDDI?](#)
10. [What is difference between Top Down and Bottom Up approach in SOAP Web Services?](#)
11. [What is REST Web Services?](#)
12. [What are advantages of REST web services?](#)
13. [What are disadvantages of REST web services?](#)
14. [What is a Resource in Restful web services?](#)
15. [What are different HTTP Methods supported in Restful Web Services?](#)
16. [Compare SOAP and REST web services?](#)
17. [What are different ways to test web services?](#)
18. [Can we maintain user session in web services?](#)
19. [What is difference between SOA and Web Services?](#)
20. [What is the use of Accept and Content-Type Headers in HTTP Request?](#)
21. [How would you choose between SOAP and REST web services?](#)
22. [What is JAX-WS API?](#)
23. [Name some frameworks in Java to implement SOAP web services?](#)
24. [Name important annotations used in JAX-WS API?](#)
25. [What is use of `javax.xml.ws.Endpoint` class?](#)
26. [What is the difference between RPC Style and Document Style SOAP web Services?](#)

27. How to get WSDL file of a SOAP web service?
28. What is sun-jaxws.xml file?
29. What is JAX-RS API?
30. Name some implementations of JAX-RS API?
31. What is wsimport utility?
32. Name important annotations used in JAX-RS API?
33. What is the use of @XmlRootElement annotation?
34. How to set different status code in HTTP response?

1. What is a Web Service?

Web Services work on client-server model where client applications can access web services over the network. Web services provide endpoint URLs and expose methods that can be accessed over network through client programs written in java, shell script or any other different technologies. Web services are stateless and doesn't maintain user session like web applications.

2. What are the advantages of Web Services?

Some of the advantages of web services are:

- Interoperability: Web services are accessible over network and runs on HTTP/SOAP protocol and uses XML/JSON to transport data, hence it can be developed in any programming language. Web service can be written in java programming and client can be PHP and vice versa.
- Reusability: One web service can be used by many client applications at the same time.
- Loose Coupling: Web services client code is totally independent with server code, so we have achieved loose coupling in our application.
- Easy to deploy and integrate, just like web applications.
- Multiple service versions can be running at same time.

3. What are different types of Web Services?

There are two types of web services:

1. SOAP Web Services: Runs on SOAP protocol and uses XML technology for sending data.

2. Restful Web Services: It's an architectural style and runs on HTTP/HTTPS protocol almost all the time. REST is a stateless client-server architecture where web services are resources and can be identified by their URIs. Client applications can use HTTP GET/POST methods to invoke Restful web services.

4. **What is SOAP?**

SOAP stands for Simple Object Access Protocol. SOAP is an XML based industry standard protocol for designing and developing web services. Since it's XML based, it's platform and language independent. So our server can be based on JAVA and client can be on .NET, PHP etc. and vice versa.

5. **What are advantages of SOAP Web Services?**

SOAP web services have all the advantages that web services has, some of the additional advantages are:

- WSDL document provides contract and technical details of the web services for client applications without exposing the underlying implementation technologies.
- SOAP uses XML data for payload as well as contract, so it can be easily read by any technology.
- SOAP protocol is universally accepted, so it's an industry standard approach with many easily available open source implementations.

6. **What are disadvantages of SOAP Web Services?**

Some of the disadvantages of SOAP protocol are:

- Only XML can be used, JSON and other lightweight formats are not supported.
- SOAP is based on the contract, so there is a tight coupling between client and server applications.
- SOAP is slow because payload is large for a simple string message, since it uses XML format.
- Anytime there is change in the server side contract, client stub classes need to be generated again.
- Can't be tested easily in browser

7. **What is WSDL?**

WSDL stands for Web Service Description Language. WSDL is an XML based document that provides technical details about the web service. Some of the useful information in WSDL document are: method name, port types, service end point, binding, method parameters etc.

8. **What are different components of WSDL?**

Some of the different tags in WSDL xml are:

- xsd:import namespace and schemaLocation: provides WSDL URL and unique namespace for web service.
- message: for method arguments
- part: for method argument name and type
- portType: service name, there can be multiple services in a wsdl document.
- operation: contains method name
- soap:address for endpoint URL.

9. **What is UDDI?**

UDDI is acronym for Universal Description, Discovery and Integration. UDDI is a directory of web services where client applications can lookup for web services. Web Services can register to the UDDI server and make them available to client applications.

10. **What is difference between Top Down and Bottom Up approach in SOAP Web Services?**

In Top Down approach first WSDL document is created to establish the contract between web service and client and then code is written, it's also termed as contract first approach. This is hard to implement because classes need to be written to confirm the contract established in WSDL. Benefit of this approach is that both client and server code can be written in parallel.

In Bottom Up approach, first web service code is written and then WSDL is generated. It's also termed as contract last approach. This approach is easy to implement because WSDL is generated based on code. In this approach client code have to wait for WSDL from server side to start their work.

11. **What is REST Web Services?**

REST is the acronym for REpresentational State Transfer. REST is an architectural style for developing applications that can be accessed over the network. REST architectural style was brought in light by Roy Fielding in his doctoral thesis in 2000.

REST is a stateless client-server architecture where web services are resources and can be identified by their URIs. Client applications can use HTTP GET/POST methods to invoke Restful web services. REST doesn't specify any specific protocol to use, but in almost all cases it's used over HTTP/HTTPS. When compared to SOAP web services, these are lightweight and doesn't follow any standard. We can use XML, JSON, text or any other type of data for request and response.

12. **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.

13. **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.

14. **What is a Resource in Restful web services?**

Resource is the fundamental concept of Restful architecture. A resource is an object with a type, relationship with other resources and methods that operate on it. Resources are identified with their URI, HTTP methods they support and request/response data type and format of data.

15. **What are different HTTP Methods supported in Restful Web Services?**

Restful web services supported HTTP methods are – GET, POST, PUT, DELETE and HEAD.

16. **Compare SOAP and REST web services?**

SOAP	REST
SOAP is a standard protocol for creating web services.	REST is an architectural style to create web services.
SOAP is acronym for Simple Object Access Protocol.	REST is acronym for REpresentational State Transfer.
SOAP uses WSDL to expose supported methods and technical details.	REST exposes methods through URIs, there are no technical details.
SOAP web services and client programs are bind with WSDL contract	REST doesn't have any contract defined between server and client
SOAP web services and client are tightly coupled with contract.	REST web services are loosely coupled.
SOAP learning curve is hard, requires us to learn about WSDL generation, client stubs creation etc.	REST learning curve is simple, POJO classes can be generated easily and works on simple HTTP methods.
SOAP supports XML data format only	REST supports any data type such as XML, JSON, image etc.

SOAP web services are hard to maintain, any change in WSDL contract requires us to create client stubs again and then make changes to client code.	REST web services are easy to maintain when compared to SOAP, a new method can be added without any change at client side for existing resources.
SOAP web services can be tested through programs or software such as Soap UI.	REST can be easily tested through CURL command, Browsers and extensions such as Chrome Postman.

17. What are different ways to test web services?

SOAP web services can be tested programmatically by generating client stubs from WSDL or through software such as Soap UI.

REST web services can be tested easily with program, curl commands and through browser extensions. Resources supporting GET method can be tested with browser itself, without any program.

18. Can we maintain user session in web services?

Web services are stateless so we can't maintain user sessions in web services.

19. What is difference between SOA and Web Services?

Service Oriented Architecture (SOA) is an architectural pattern where applications are designed in terms of services that can be accessed through communication protocol over network. SOA is a **design pattern** and doesn't go into implementation.

Web Services can be thought of as Services in SOAP architecture and providing means to implement SOA pattern.

20. What is the use of Accept and Content-Type Headers in HTTP Request?

These are important headers in Restful web services. Accept headers tells web service what kind of response client is accepting, so if a web service is capable of sending response in XML and JSON format and client sends Accept header as "application/xml" then XML response will be sent. For Accept header "application/json", server will send the JSON response.

Content-Type header is used to tell server what is the format of data being sent in the request. If Content-Type header is "application/xml" then server will try to parse it as XML data. This header is useful in HTTP Post and Put requests.

21. How would you choose between SOAP and REST web services?

Web Services work on client-server model and when it comes to choose between SOAP and REST, it all depends on project requirements. Let's look at some of the conditions affecting our choice:

- Do you know your web service clients beforehand? If Yes, then you can define a contract before implementation and SOAP seems better choice. But if you don't then REST seems better choice because you can provide sample request/response and test cases easily for client applications to use later on.
- How much time you have? For quick implementation REST is the best choice. You can create web service easily, test it through browser/curl and get ready for your clients.
- What kind of data format are supported? If only XML then you can go with SOAP but if you think about supporting JSON also in future then go with REST.

22. What is JAX-WS API?

JAX-WS stands for Java API for XML Web Services. JAX-WS is XML based Java API to build web services server and client application. It's part of standard Java API, so we don't need to include anything else which working with it. Refer to [JAX-WS Tutorial](#) for a complete example.

23. Name some frameworks in Java to implement SOAP web services?

We can create SOAP web services using JAX-WS API, however some of the other frameworks that can be used are Apache Axis and Apache CXF. Note that they are not implementations of JAX-WS API, they are totally different framework that work on Servlet model to expose your business logic classes as SOAP web services. Read more at [Java SOAP Web Service Eclipse](#) example.

24. Name important annotations used in JAX-WS API?

Some of the important annotations used in JAX-WS API are:

- @WebService
- @SOAPBinding
- @WebMethod

25. What is use of `javax.xml.ws.Endpoint` class?

Endpoint class provides useful methods to create endpoint and publish existing implementation as web service. This comes handy in testing web services before making further changes to deploy it on actual server.

26. What is the difference between RPC Style and Document Style SOAP web Services?

RPC style generate WSDL document based on the method name and it's parameters. No type definitions are present in WSDL document.

Document style contains type and can be validated against predefined schema. Let's look at these with a simple program. Below is a simple test program where I am using Endpoint to publish my simple SOAP web service.

TestService.java

```
package com.journaldev.jaxws.service;

import javax.jws.WebMethod;
import javax.jws.WebService;
import javax.jws.soap.SOAPBinding;
import javax.xml.ws.Endpoint;

@WebService
@SOAPBinding(style = SOAPBinding.Style.RPC)
public class TestService {

    @WebMethod
    public String sayHello(String msg){
        return "Hello "+msg;
    }

    public static void main(String[] args){
        Endpoint.publish("http://localhost:8888/testWS", new
TestService());
    }
}
```

When I run above program and then access the WSDL, it gives me below XML.

rpc.xml

```
<?xml version='1.0' encoding='UTF-8'?>
<!-- Published by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI
2.2.10 svn-revision#919b322c92f13ad085a933e8dd6dd35d4947364b. --><!-- Generated
by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.10 svn-
revision#919b322c92f13ad085a933e8dd6dd35d4947364b. -->
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
```



```

xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://service.jaxws.journaldev.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://service.jaxws.journaldev.com/"
name="TestServiceService">
<types/>
<message name="sayHello">
<part name="arg0" type="xsd:string"/>
</message>
<message name="sayHelloResponse">
<part name="return" type="xsd:string"/>
</message>

```

Notice that **types** element is empty and we can't validate it against any schema. Now just change the SOAPBinding.Style.RPC to SOAPBinding.Style.DOCUMENT and you will get below WSDL.

document.xml

```

<?xml version='1.0' encoding='UTF-8'?>
<!-- Published by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI
2.2.10 svn-revision#919b322c92f13ad085a933e8dd6dd35d4947364b. --><!-- Generated
by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.10 svn-
revision#919b322c92f13ad085a933e8dd6dd35d4947364b. -->
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://service.jaxws.journaldev.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://service.jaxws.journaldev.com/"
name="TestServiceService">
<types>
<xsd:schema>
<xsd:import namespace="http://service.jaxws.journaldev.com/"
schemaLocation="http://localhost:8888/testWS?xsd=1"/>
</xsd:schema>
</types>
<message name="sayHello">

```

Open schemaLocation URL in browser and you will get below XML.

schemaLocation.xml

```
<?xml version='1.0' encoding='UTF-8'?>
<!-- Published by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI
2.2.10 svn-revision#919b322c92f13ad085a933e8dd6dd35d4947364b. -->
<xs:schema xmlns:tns="http://service.jaxws.journaldev.com/"
xmlns:xs="http://www.w3.org/2001/XMLSchema" version="1.0"
targetNamespace="http://service.jaxws.journaldev.com/">

<xs:element name="sayHello" type="tns:sayHello"/>

<xs:element name="sayHelloResponse" type="tns:sayHelloResponse"/>

<xs:complexType name="sayHello">
<xs:sequence>
<xs:element name="arg0" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>

<xs:complexType name="sayHelloResponse">
<xs:sequence>
<xs:element name="return" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
```

So here WSDL document can be validated against the schema definition.

27. How to get WSDL file of a SOAP web service?

WSDL document can be accessed by appending ?wsdl to the SOAP endpoint URL. In above example, we can access it at <http://localhost:8888/testWS?wsdl> location.

28. What is sun-jaxws.xml file?

This file is used to provide endpoints details when JAX-WS web services are deployed in servlet container such as Tomcat. This file is present in WEB-INF directory and contains endpoint name, implementation class and URL pattern. For example;

sun-jaxws.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<endpoints xmlns="http://java.sun.com/xml/ns/jax-ws/ri/runtime" version="2.0">
  <endpoint
```

```

        name="PersonServiceImpl"
        implementation="com.journaldev.jaxws.service.PersonServiceImpl"
        url-pattern="/personWS"/>
</endpoints>

```

29. What is JAX-RS API?

Java API for RESTful Web Services (JAX-RS) is the Java API for creating REST web services. JAX-RS uses annotations to simplify the development and deployment of web services. JAX-RS is part of JDK, so you don't need to include anything to use its annotations.

30. Name some implementations of JAX-RS API?

There are two major implementations of JAX-RS API.

1. Jersey: Jersey is the reference implementation provided by Sun. For using Jersey as our JAX-RS implementation, all we need to configure its servlet in web.xml and add required dependencies. Note that JAX-RS API is part of JDK not Jersey, so we have to add its dependency jars in our application.
2. RESTEasy: RESTEasy is the JBoss project that provides JAX-RS implementation.

31. What is wsimport utility?

We can use wsimport utility to generate the client stubs. This utility comes with standard installation of JDK. Below image shows an example execution of this utility for one of JAX-WS project.

```

pankaj:temp pankaj$
pankaj:temp pankaj$ wsimport -s . http://localhost:8888/ws/person?wsdl
parsing WSDL...

Generating code...

Compiling code...

pankaj:temp pankaj$ ls -ltr com/journaldev/jaxws/service/
total 96
-rw-r--r-- 1 pankaj staff 127 Oct 3 01:38 package-info.java
-rw-r--r-- 1 pankaj staff 270 Oct 3 01:38 package-info.class
-rw-r--r-- 1 pankaj staff 3255 Oct 3 01:38 PersonServiceImplService.java
-rw-r--r-- 1 pankaj staff 2433 Oct 3 01:38 PersonServiceImplService.class
-rw-r--r-- 1 pankaj staff 2224 Oct 3 01:38 PersonService.java
-rw-r--r-- 1 pankaj staff 1783 Oct 3 01:38 PersonService.class
-rw-r--r-- 1 pankaj staff 1807 Oct 3 01:38 PersonArray.java
-rw-r--r-- 1 pankaj staff 885 Oct 3 01:38 PersonArray.class
-rw-r--r-- 1 pankaj staff 2831 Oct 3 01:38 Person.java
-rw-r--r-- 1 pankaj staff 934 Oct 3 01:38 Person.class
-rw-r--r-- 1 pankaj staff 1208 Oct 3 01:38 ObjectFactory.java
-rw-r--r-- 1 pankaj staff 628 Oct 3 01:38 ObjectFactory.class
pankaj:temp pankaj$

```

32. Name important annotations used in JAX-RS API?

Some of the important JAX-RS annotations are:

- `@Path`: used to specify the relative path of class and methods. We can get the URI of a webservice by scanning the Path annotation value.

- `@GET`, `@PUT`, `@POST`, `@DELETE` and `@HEAD`: used to specify the HTTP request type for a method.
- `@Produces`, `@Consumes`: used to specify the request and response types.
- `@PathParam`: used to bind the method parameter to path value by parsing it.

33. What is the use of `@XmlRootElement` annotation?

`XmlRootElement` annotation is used by JAXB to transform java object to XML and vice versa. So we have to annotate model classes with this annotation.

34. How to set different status code in HTTP response?

For setting HTTP status code other than 200, we have to use `javax.ws.rs.core.Response` class for response. Below are some of the sample return statements showing it's usage.

```
return Response.status(422).entity(exception).build();  
return Response.ok(response).build(); //200
```

For a complete example, please read [RESTEasy Tutorial](#).

That's all for **web services interview questions**, let me know what you think about it through comments. If you think that I have missed some important questions, please let me know through comments and I will be happy to answer them and add to the list.



**About Pankaj**

If you have come this far, it means that you liked what you are reading. Why not reach little more and connect with me directly on [Google Plus](#), [Facebook](#) or [Twitter](#). I would love to hear your thoughts and opinions on my articles directly.

Recently I started creating video tutorials too, so do check out my videos on [Youtube](#).

FILED UNDER: [INTERVIEW QUESTIONS](#), [JAVA EE](#)

Comments

Shailesh says

SEPTEMBER 16, 2018 AT 3:55 AM

JAX-RS is not part of JDK. It is part of Java EE specification.

[Reply](#)

kamalakar reddy perla says

AUGUST 27, 2018 AT 4:10 AM

If u have mule soft interview questions & answers. Please share

[Reply](#)

Suraj Kumar says

AUGUST 9, 2018 AT 8:14 PM

Hi Pankaj,

You have stated, "Since it works on HTTP, there can't be asynchronous calls." However, use of REST web service call in all our applications is asynchronous in nature. Call of one web service is independent of other than how come it can't be asynchronous.

[Reply](#)

SHYAM says

SEPTEMBER 12, 2018 AT 11:29 PM

You shouldn't confuse concurrent web-service calls with Asynchronous behavior. Asynchronous means non blocking function call. Even though you can make multiple concurrent web service requests from multiple clients to a single web-service at the same time, each concurrent web-service request will block each incoming client request and make the client wait for a HTTP response from the web-service to complete processing the HTTP request. Such behavior is clearly synchronous behavior. Its not Asynchronous. Hope you understand what I have explained....

[Reply](#)

Motilal Daravatu says

JULY 8, 2018 AT 8:39 PM

Its really nice information for beginner. What i felt was it will be really good if you can extend this tutorial by adding security and exception handling in both SOAP and REST web services

[Reply](#)

Sandeesh says

JUNE 20, 2018 AT 6:35 PM

Thanks Pankaj.

[Reply](#)

Dnyaneshwar says

APRIL 15, 2018 AT 1:18 PM

Really helpful

[Reply](#)

mohamed says

APRIL 5, 2018 AT 10:47 AM

really helpful for beginner...

[Reply](#)

jp says

FEBRUARY 26, 2018 AT 3:31 AM

very useful many thanks.

[Reply](#)

Rajnish says

NOVEMBER 7, 2017 AT 12:42 AM

Please add questions on how to handle exceptions in RESTful web services and how to implement Authentications & policies in REST.

Regards

Rajnish

[Reply](#)

Binh Thanh Nguyen says

AUGUST 16, 2017 AT 3:10 AM

Thanks, nice share!

[Reply](#)

Nagalaxmi says

JULY 23, 2017 AT 10:35 AM

Hi pankaj, can u please send me brief description about Salesforce.

[Reply](#)

giri babu says

JUNE 28, 2017 AT 7:33 PM

Nice collection and very much useful. Thank you

[Reply](#)

satya says

JUNE 8, 2017 AT 7:08 AM

Useful stuff for beginners

[Reply](#)

Harold Mamakoko says

MAY 31, 2017 AT 5:49 AM

Best Tutorial, this really helped me get started with SOAP Web Services.

[Reply](#)

KL says

APRIL 24, 2017 AT 9:58 PM

Superb !! Very helpful. It is easy to quickly revise the concepts before going to interview. Also while preparation one can understand the concepts he has missed

[Reply](#)

Manas says

FEBRUARY 28, 2017 AT 11:24 AM

Very useful

[Reply](#)

Karishma says

FEBRUARY 17, 2017 AT 5:14 AM

Thank you so much!!!!

very useful

[Reply](#)

skadam says

FEBRUARY 10, 2017 AT 3:49 AM

Hi Pankaj Kumar,

Very good post.

Please add some questions/details about rest/soap web services security. That would be very helpful. Because this is the tope most interview question "How you implemented the Rest/Soap web services security in project."

[Reply](#)

satendra says

DECEMBER 14, 2016 AT 8:44 AM

good help

[Reply](#)

Deepak Rathi says

NOVEMBER 4, 2016 AT 8:27 PM

Hi Pankaj,

According your explanation-

Web service advantage is loosely coupled and disadvantage is tightly coupled how is it possible?

If it is loosely coupled how it can be tight????????

[Reply](#)

Patil says

DECEMBER 27, 2016 AT 5:37 AM

May be he meant that WSDL contract change(Methods signature) results in change for the consumer.

[Reply](#)

Dinesh says

SEPTEMBER 21, 2016 AT 9:16 AM

Good One for Interview preparation

[Reply](#)

bappa says

SEPTEMBER 21, 2016 AT 12:17 AM

Hello

Thanks your good and helpful work

If possible please dd some questions about web services security

[Reply](#)

Raja Rajagopal says

AUGUST 27, 2016 AT 6:58 AM

Hi Pankaj,

Good Post. However,if could add some questions about web services security. That would be helpful. Because I have been asked in one interview about web services security.

Thanks,

Raja Rajagopal

[Reply](#)

Deepak Reddy S says

AUGUST 27, 2016 AT 12:39 AM

Answer for 23 is wrong. As per the <http://cxf.apache.org/docs/why-cxf.html>, Apache CXF is an implementation of JAX-WS api

[Reply](#)

Pankaj says

AUGUST 27, 2016 AT 10:55 AM

Apache XSF supports SOAP web services too. Check their home page <http://cxf.apache.org/>

[Reply](#)

ravi says

MAY 19, 2016 AT 4:57 AM

really helpful for beginner...

Thanks

[Reply](#)

namrata dhumal says

FEBRUARY 17, 2016 AT 11:04 PM

Hi Pankaj,

Really good post.

Namrata

[Reply](#)

Veranga says

DECEMBER 13, 2015 AT 9:44 PM

Hi Pankaj,

Does SOAP is a protocol. If SOAP is a protocol , SOAP should be a defined standards which used to communicate data. But SOAP uses http or https.

Please advice.

Thanks

Veranga Sooriyabandara

vsooriyabandara@gmail.com

[Reply](#)

Piyush Bisht says

NOVEMBER 16, 2015 AT 8:49 AM

Just to add, Restful web services also support PATCH, OPTIONS methods

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

POST COMMENT