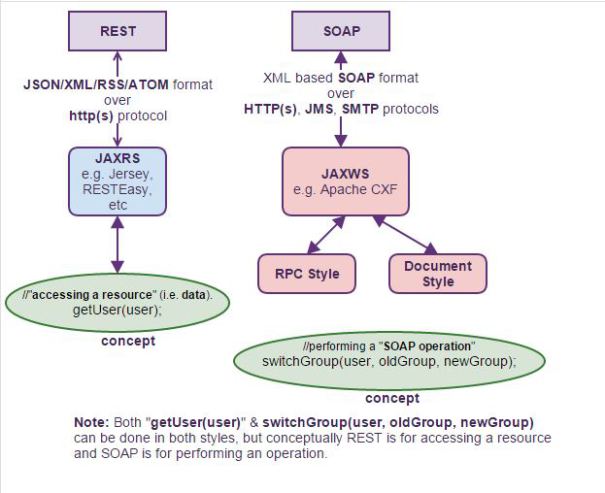
Q1. What are the different styles of Web Services used for application integration? and What are the differences between both SOAP WS and RESTful WS?  
A1. SOAP WS and RESTful Web Service. Web services are very popular and widely used to integrate similar (i.e. Java applications) and disparate systems (i.e. legacy applications and applications written in .Net etc) as they are **language neutral**.

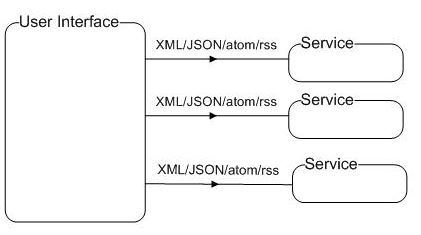
[](http://i2.wp.com/www.java-success.com/wp-content/uploads/2014/08/REST-Vs-SOAP-comparison.jpg)

Java Web Service styles comparison

|  |  |
| --- | --- |
| **SOAP Web service** | **RESTful Web service** |
| SOAP (Simple Object Access Protocol) is a standard**communication protocol** on top of transport protocols such as HTTP, SMTP, Messaging, TCP, UDP, etc. | REST is an **architectural style** by which data can be transmitted over transport protocol such as HTTP(S).  Each unique URL is a some representation of a resource (i.e Object like Account, Customer, etc), and you can get the contents of the resources (i.e Objects) via HTTP verb “GET” and to modify via “DELETE”,”POST”, or “PUT”. |
| [SOAP Layers](http://i1.wp.com/www.java-success.com/wp-content/uploads/2014/08/SOAP-Layers.jpg)  SOAP Layers | [REST Layers](http://i0.wp.com/www.java-success.com/wp-content/uploads/2014/08/RESTful-layers1.jpg)  REST Layers |
| SOAP uses its own protocol and focuses on exposing pieces of **application logic (not data)** as services. SOAP exposes operations. SOAP is focused on accessing named operations, which implement some business logic through different interfaces. | REST is about exposing a public API over the internet to handle CRUD (Create, Read, Update, and Delete) operations on data. REST is focused on accessing named resources through a single consistent interface. |
| SOAP only permits XML data formats.  [SOAP](http://i1.wp.com/www.java-success.com/wp-content/uploads/2014/08/SOAP-paylaod.jpg)  SOAP | REST permits many different data formats like XML, JSON data, text, HTML, atom, RSS, etc. JSON is less verbose than XML and is a better fit for data and parses much faster.  **URL:**http://localhost:8080/myapp/createEmptyCase  [REST content](http://i1.wp.com/www.java-success.com/wp-content/uploads/2014/08/REST-body-content-xml-json-more.jpg)  REST content XML, JSON, RSS, etc |
| SOAP based reads **cannot be cached**. The application that uses SOAP needs to provide cacheing. | REST based **reads can be cached**. Performs and scales better. |
| Supports both **SSL security**and **WS-security**, which adds some enterprise security features. Supports identity through intermediaries, not just point to point SSL.  — WS-Security maintains its encryption right up to the point where the request is being processed.  — WS-Security allows you to secure parts (e.g. only credit card details) of the message that needs to be secured. Given that encryption/decryption is not a cheap operation, this can be a performance boost for larger messages.  — It is also possible with WS-Security to secure different parts of the message using different keys or encryption algorithms. This allows separate parts of the message to be read by different people without exposing other, unneeded information.  — SSL security can only be used with HTTP. WS-Security can be used with other protocols like UDP, SMTP, etc. | Supports **only point-to-point SSL security**.  — The basic mechanism behind SSL is that the client encrypts all of the requests based on a key retrieved from a third party. When the request is received at the destination, it is decrypted and presented to the service. This means the request is only encrypted while it is traveling between the client and the server. Once it hits the server (or a proxy which has a valid certificate), it is decrypted from that moment on.  — The SSL encrypts the whole message, whether all of it is sensitive or not. |
| Has comprehensive support for both **ACID** based transaction management for short-lived transactions and **compensation** based transaction management for long-running transactions. It also supports two-phase commit across distributed resources. | REST supports transactions, but it is **neither ACID compliant nor can provide two phase commit**across distributed transactional resources as it is limited by its HTTP protocol. |
| SOAP has success or retry logic built in and provides end-to-end reliability even through SOAP intermediaries. | REST does not have a standard messaging system, and expects clients invoking the service to deal with communication failures by retrying. |

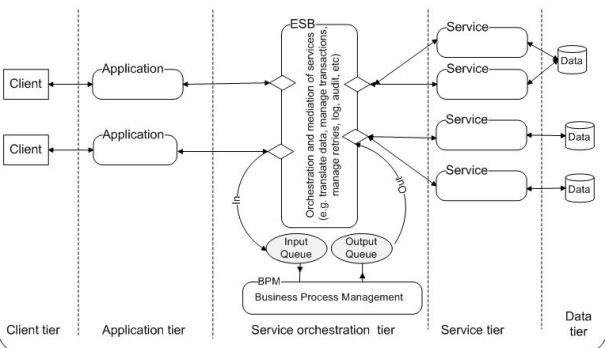
**Which one to favor?** In general, a REST based web service is preferred due to its simplicity, performance, scalability, and support for multiple data formats. SOAP is favored where service requires comprehensive support for security and transactional reliability.

Q2. Differentiate between SOA (Service Oriented Architecture) versus WOA (Web Oriented Architecture)?  
A2. **WOA extends SOA** to be a light-weight architecture using technologies such as REST and POX (Plain Old XML). POX compliments REST. JSON is a variant for data returned by REST Web Services. It consumes less bandwidth and is easily handled by web developers mastering the Javascript language

[](http://i0.wp.com/www.java-success.com/wp-content/uploads/2014/08/JudgingExperience-WS.jpg)

WOA – RESTFul Service Calls via AJAX to populate different sections of a UI

SOA and WOA differ in terms of the layers of abstraction.**SOA is a system-level architectural style** that tries to expose business capabilities so that they can be consumed by many applications. **WOA is an interface-level architectural style** that focuses on the means by which these service capabilities are exposed to consumers. You can start out with a WOA and then grow into SOA.

[](http://i2.wp.com/www.java-success.com/wp-content/uploads/2014/08/Screen-shot-2014-08-28-at-9.25.07-PM.png)

SOA (Service Oriented Architecture)

Q3. How would you decide what style of Web Service to use? SOAP WS or REST?  
A3. In general, a REST based Web service is preferred due to its simplicity, performance, scalability, and support for multiple data formats. SOAP is favored where service requires comprehensive support for security and transactional reliability.

The answer really depends on the functional and non-functional requirements. Asking the questions listed below will help you choose.

**1)** Does the service expose data or business logic? (REST is a better choice for exposing data, SOAP WS might be a better choice for logic).

**2)** Do consumers and the service providers require a formal contract? (SOAP has a formal contract via WSDL)

**3)** Do we need to support multiple data formats?

**4)** Do we need to make AJAX calls? (REST can use the XMLHttpRequest)

**5)** Is the call synchronous or asynchronous?

**6)** Is the call stateful or stateless? (REST is suited for statless CRUD operations)

**7)** What level of security is required? (SOAP WS has better support for security)

**8)** What level of transaction support is required? (SOAP WS has better support for transaction management)

**9)** Do we have limited band width? (SOAP is more verbose)

**10)** What’s best for the developers who will build clients for the service? (REST is easier to implement, test, and maintain)

Q4. What tools do you use to test your Web Services?  
A4. SoapUI tool for SOAP WS & RESTFul web service testing and on the browser the Firefox “**poster**” plugin or Google Chrome “**Postman**” extension for RESTFul services.

Q5. What is the difference between SOA and a Web service?  
A5. **SOA** is a software design principle and an architectural pattern for implementing loosely coupled, reusable and coarse grained services. You can implement SOA using any protocols such as HTTP, HTTPS, JMS, SMTP, RMI, IIOP (i.e. EJB uses IIOP), RPC etc. Messages can be in XML or Data Transfer Objects (DTOs).

**Web service** is an implementation technology and one of the ways to implement SOA. You can build SOA based applications without using Web services – for example by using other traditional technologies like Java RMI, EJB, JMS based messaging, etc. But what Web services offer is the standards based and platform-independent service via HTTP, XML, SOAP, WSDL and UDDI, thus allowing interoperability between heterogeneous technologies such as J2EE and .NET.

Q6. Why not favor traditional style middle-ware such as RPC, CORBA, RMI and DCOM as opposed to Web services?  
A6. The**traditional middle-wares tightly couple** connections to the applications. Tightly coupled applications are hard to maintain and less reusable. Generally do not support heterogeneity. Do not work across Internet and can be more expensive and hard to use.

**Web Services support loosely coupled connections**. The interface of the Web service provides a layer of abstraction between the client and the server. The loosely coupled applications reduce the cost of maintenance and increases re-usability. Web Services present a new form of middle-ware based on XML and Web. Web services are language and platform independent. You can develop a Web service using any language and deploy it on to any platform, from small device to the largest supercomputer. Web service uses language neutral protocols such as HTTP and communicates between disparate applications by passing XML or JSON messages to each other via a Web API. Do work across internet, less expensive and easier to use.

**1) What is the difference between SOAP and REST web services?**  
Ans: These is one of the most frequently asked question in Java interviews. Don’t go to interview without reading some content on how these two kind of web services behave. I shall write in detail about the differences between the two in future but for now bear in mind that they are completely different approaches to achieve the same target. The debate on which one is better has no end but when answering this question, you need to make sure that you keep the balance between the two and mention pros and cons of each type of web services.

While [SOAP based web services](http://www.javaexperience.com/introduction-to-soap-based-java-web-services/) rely heavily on XML standard, REST web services use the HTTP protocol methods PUT, GET, DELETE and POST for operations. If you haven’t written both kind of web services code then let me tell you first hand that it is very easy to write [REST web services](http://www.javaexperience.com/list-of-tools-for-rest-web-services-in-java/) without making mistakes. For both SOAP and REST web services, there are frameworks out there in market which make the task of writing the web services simpler.

Another important difference which you should be mentioning in the answer to this question is that while REST web services can be invoked using simple HTTP clients and browser add-on , the invocation of SOAP based web services require some learning curve and creation of XML messages. The use HTTP helps web developers who may not have got much chance to play with XML.

**2) Since JSP, ASP and PHP pages can be browsed by anyone using a HTTP client and hence platform independence, then what kind of platform independence do SOAP based web services bring?**  
Ans: JSP, ASP and PHP were not designed for platform independence but for writing code which can be sent to the [web browser](http://www.javaexperience.com/java-how-to-detect-browser-proxy/) for rendering. There are no means by which a standard can be set between the developer and consumer of these websites designed using PHP, ASP, JSP. On the other hand, SOAP based web services define a protocol standard to have communication between the producer and consumer of web services. So basically the difference lies in the purpose for which these technologies were founded.

**3) What are the components of WSDL and their significance?**  
Ans: WSDL provides information about the structure of SOAP message to be sent for invocation of different operations exposed by the web service.I have written a tutorial on [WSDL components and their meaning](http://www.javaexperience.com/understanding-wsdl-structure-and-elements/).

**4) Can we modify the header and body parts of SOAP messages so that some common operations like logging can be performed?**  
Ans: This can be achieved by using interceptors which intercept the message before sending and receiving it. An interceptor can be written on the client or server side and perform its job as defined in the code. These interceptors are known as handlers in JAX-WS specification. There are SOAPHandlers and LogicalHandlers which can do different processing on the server side and client side.

**5) What steps do you follow for debugging the issues in your code?**  
Ans: There are a number of debugging techniques available for SOAP based web services. Every developer has its own set of favorite ways to find issues. Some of the commonly used ways to debug web services are:

 Monitoring web services requests and response using some traffic analysis tools like tcpdump or [Eclipse TCP/IP monitor](http://www.javaexperience.com/monitor-web-service-in-eclipse-using-tcpip-monitor/).

 Use some tools like soapUI or [IBM InfoSphere](http://pic.dhe.ibm.com/infocenter/pim/v9r1m0/topic/com.ibm.pim.app.doc/webservices/pim_con_webservdebug.html)

 Use logs or System.out.println() for checking any coding issues.

 Mention any special debugging techniques you have been using in your web services code.  
  
Read more: <http://www.javaexperience.com/jax-ws-webservice-interview-questions/#ixzz3gpWZoQLt>

**19 RESTFul Web Service Interview Questions**

**RESTFul Web Service Interview Questions**

**1. What is differences between RESTful web services and SOAP web services ?**  
**Ans:**Though both RESTful web series and SOAP web service can operate cross platform they are architecturally different to each other, here is some of differences between REST and SOAP:  
1) REST is more simple and easy to use than SOAP. REST language is based on use of nouns and verbs (better readability)  
2) REST uses HTTP protocol for producing or consuming web services while SOAP uses XML.

* The SOAP WS is transport protocol neutral. Supports multiple protocols like HTTP(S),  Messaging, TCP, UDP SMTP, etc.
* The REST is transport protocol specific. Supports only HTTP or HTTPS protocols.

3) REST is lightweight as compared to SOAP and preferred choice in mobile devices and PDA’s. REST does not need XML parsing, no message header (to and from), hence less bandwidth  
4) REST supports different format like text, JSON and XML while SOAP only support XML.

* The SOAP WS permits only XML data format.You define operations, which tunnels through the POST. The focus is on accessing the named operations and exposing the application logic as a service.
* The REST permits multiple data formats like XML, JSON data, text, HTML, etc. Any browser can be used because the REST approach uses the standard GET, PUT, POST, and DELETE Web operations. The focus is on accessing the named resources and exposing the data as a service. REST has AJAX support. It can use the XMLHttpRequest object. Good for stateless CRUD (Create, Read, Update, and Delete) operations.

GET – represent()  
POST – acceptRepresention()  
PUT – storeRepresention()  
DELETE – removeRepresention()  
5) SOAP based reads cannot be cached. REST based reads can be cached. Performs and scales better.  
6) Different error handling:  
REST: requires HTTP error handling  
SOAP: can have user defined error  
7) REST only supports synchronous  message because of its reliance of HTTP and HTTPS  
8) SOAP WS supports both SSL security and WS-security, which adds some enterprise security features like maintaining security right up to the point where it is needed, maintaining identities through intermediaries and not just point to point SSL only, securing different parts of the message with different security algorithms, etc.  
The REST supports only point-to-point SSL security. The SSL encrypts the whole message, whether all of it is sensitive or not.  
9) The SOAP has comprehensive support for both ACID based  transaction management  for short-lived transactions and compensation based transaction management for long-running transactions. It also supports two-phase commit across distributed resources.  
The REST supports transactions, but it  is neither ACID compliant nor can provide two phase commit across distributed transactional resources as it is limited by its HTTP protocol.  
10) The SOAP has success or retry logic built in and provides end-to-end reliability even through SOAP intermediaries. REST does not have a standard messaging system, and expects clients invoking the service to deal with communication failures by retrying.

**2. What is REST and RESTful web services?**  
Ans:This is the first REST interview question on most of interviews as not everybody familiar with REST and also start discussion based on candidates response. Anyway REST stands for REpresentational State Transfer (REST) its a relatively new concept of writing web services which enforces a stateless client server design where web services are treated as resource and can be accessed and identified by there URL unlike SOAP web services which were defined by WSDL.  
Web services written by apply REST Architectural concept are called RESTful web services which focus on System resources and how state of Resource should be transferred over http protocol to a different clients written in different languages. In RESTful web services http methods like GET, PUT, POST and DELETE can can be used to perform CRUD operations.

**3.What is difference between top-down and bottom-up approach of developing web services ?**  
**Ans:**In **top-down approach** first WSDL document is created and than Java classes are developed based on WSDL contract, so if WSDL contract changes you got to change your Java classes while in case of **bottom up approach** of web service development you first create Java code and then use annotations like @WebService to specify contract or interface and WSDL field will be automatically generated from your build.

**4.What happens if RestFull resources are accessed by multiple clients ? do you need to make it thread-safe?**  
**Ans:**Since a new Resource instance is created for every incoming Request there is no need to make it thread-safe or add synchronization. Multiple clients can safely access RestFull resources concurrently.

**5.What will you do when an error code has to be returned to the client  
OR  
How will you handle application error scenarios in RESTful web service?**

**6.What is a web service?**  
**Ans:**A **Web service** is a method of communication between two electronic devices over the Web (Internet). The W3C defines a “Web service” as “a software system designed to support interoperable machine-to-machine interaction over a network”. It has an interface described in a machine-processable format (specifically Web Services Description Language, known by the acronym WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XMLserialization in conjunction with other Web-related standards.”

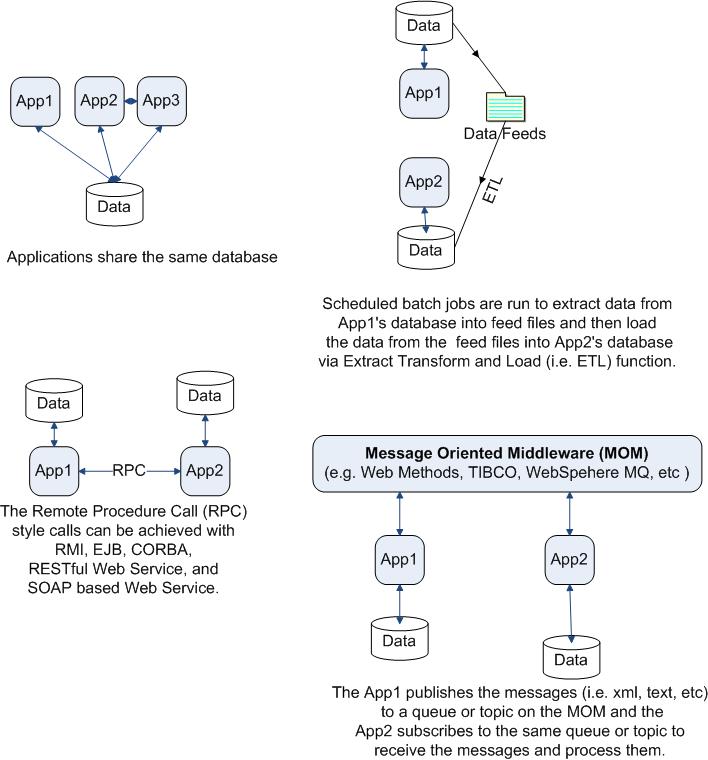
**7.Can I use GET request instead of PUT to create resources?**  
**Ans:**No, you are supposed to use PUT or POST. GET operations should only have view rights.

**8.What all kind of output formats can one generate using RESTful web service?**

**9.What all tools have you used to write RESTful web service?**

**10.What is meant by JAX-WS and JAX-RS?**  
**Ans:**Java API for XML Web Services(**JAX-WS**)  
Java API for RESTful Web Services (**JAX-RS**)

**11.What are the different application integration styles?**  
**Ans.**There are a number of different integration styles like  
1. Shared database  
2. batch file transfer  
3. Invoking remote procedures (RPC)  
4. Exchanging asynchronous messages over a message oriented middle-ware (MOM).



**12.What are the different styles of Web Services used for application integration?**  
**Ans.** **SOAP WS** and **RESTful**Web Service

**13.How would you decide what style of Web Service to use? SOAP WS or REST?**  
**Ans**. In general, a REST based Web service is preferred due to its simplicity, performance, scalability, and support for multiple data formats. SOAP is favored where service requires comprehensive support for security and transactional reliability.  
The answer really depends on the functional and non-functional requirements. Asking the questions listed below will help you choose.

* **Does the service expose data or business logic?**(REST is a better choice for exposing data, SOAP WS might be a better choice for logic).Do the consumers and the service providers require a formal contract? (SOAP has a formal contract via WSDL)
* **Do we need to support multiple data formats?**
* **Do we need to make AJAX calls?**(REST can use the XMLHttpRequest)
* **Is the call synchronous or  asynchronous?**
* **Is the call stateful or stateless?** (REST is suited for stateless CRUD operations)
* **What level of security is required?** (SOAP WS has better support for security)
* **What level of transaction support is required?**(SOAP WS has better support for transaction management)
* **Do we have limited band width?** (SOAP is more verbose)
* **What’s best for the developers who will build clients for the service?**(REST is easier to implement, test, and maintain)

**14. What tools do you use to test your Web Services?**  
**Ans:SoapUI** tool for SOAP WS and the Firefox “**poster**” plugin for RESTFul services.

**15.What is the difference between SOA and a Web service?**  
**Ans:SOA is**a software design principle and an architectural pattern for implementing loosely coupled, reusable and coarse grained services. You can implement SOA using any protocols such as HTTP, HTTPS, JMS, SMTP, RMI, IIOP (i.e. EJB uses IIOP), RPC etc. Messages can be in XML or Data Transfer Objects (DTOs).  
**Web service is** an implementation technology and one of the ways to implement SOA. You can build SOA based applications without using Web services – for example by using other traditional technologies like Java RMI, EJB, JMS based messaging, etc. But what Web services offer is the standards based and platform-independent service via HTTP, XML, SOAP, WSDL and UDDI, thus allowing interoperability between heterogeneous technologies such as J2EE and .NET.

**16.Web services when you can use traditional style middle-ware such as RPC, CORBA, RMI and DCOM?**  
**Ans:**The**traditional middle-wares** tightly couple connections to the applications and it can break if you make any modification to your application. Tightly coupled applications are hard to maintain and less reusable. Generally do not support heterogeneity. Do not work across Internet. Can be more expensive and hard to use.  
**Web Services** support loosely coupled connections. The interface of the Web service provides a layer of abstraction between the client and the server. The loosely coupled applications reduce the cost of maintenance and increases re-usability. Web Services present a new form of middle-ware based on XML and Web. Web services are language and platform independent. You can develop a Web service using any language and deploy it on to any platform, from small device to the largest supercomputer. Web service uses language neutral protocols such as HTTP and communicates between disparate applications by passing XML messages to each other via a Web API. Do work across internet, less expensive and easier to use.

**17.What are the different approaches to developing a SOAP based Web service?**  
**Ans. Following are the two approaches.**

* The **contract-first** approach, where you define the contract first with XSD and WSDL and the generate the Java classes from the contract.
* The **contract-last**approach where you  define the Java classes first and then generate the contract, which is the  WSDL file from the Java classes.

**Note:** The WSDL describes all operations that the service provides, locations of the endpoints (i.e. where the services can be invoked), and simple and complex elements that can be passed in requests and responses.

**18.What are the pros and cons of each approach, and which approach would you prefer?**  
**Ans:**  
**Contract-first Web service**  
**PROS:**

* Clients are decoupled from the server, hence the implementation logic can be revised on the server without affecting the clients.
* Developers can work simultaneously on client and server side based on the contract both agreed on.
* You have full control over how the request and response messages are constructed — for example, should “status” go as an element or as an attribute? The contract clearly defines it. You can change OXM (i.e. Object to XML Mapping) libraries without having to worry if the “status” would be generated as “attribute” instead of an element. Potentially, even Web service frameworks and tool kits can be changed as well from say Apache Axis to Apache CXF, etc

**CONS:**

* More upfront work is involved in setting up the XSDs and WSDLs. There are tools like XML Spy, Oxygen XML, etc to make things easier. The object models need to be written as well.
* Developers need to learn XSDs and WSDLs in addition to just knowing Java.

**Contract-last Web service**  
**PROS:**

* Developers don’t have to learn anything related to XSDs, WSDLs, and SOAP. The services are created quickly by exposing the existing service logic with frameworks/tool sets. For example, via IDE based wizards, etc.
* The learning curve and development time can be smaller compared to the Contract-first Web service.

**CONS:**

* The development time can be shorter to initially develop it, but what about the on going maintenance and extension time if the contract changes or new elements need to be added? In this approach, since the clients and servers are more tightly coupled, the future changes may break the client contract and affect all clients or require the services to be properly versioned and managed.
* In this approach, The XML payloads cannot be controlled. This means changing your OXM libraries could cause something that used to be an element to become an attribute with the change of the OXM.

**19.So, which approach will you choose?**  
**Ans:**The best practice is to use “**contract-first**“, and here is the link that explains this much better with examples –>  contract-first versus contract-last web services In a nutshell, the contract-last is more fragile than the “contract-first”.  You will have to decide what is most appropriate based on your requirements, tool sets you use, etc.

- See more at: <http://www.javaface.com/19-restful-web-service-interview-questions/#sthash.zt1c8ijP.dpuf>

Following are some of the questions on REST with my understanding as answer:

---------------------------------------------------------------------------------------

**REST Interview Question 1:**

What is the Caching mechanism that a RESTful service would provide?

**This is how I would answer this question**

As RESTful service uses HTTP as transport protocol, it can leverage caching

features from HTTP specification as well.

HTTP 1.0 specification has Expires header that can be used to indicate ways

to client side for the appropriate caching intentions of server.

HTTP 1.1 specification however has more caching related features to choose from.

These are in form of various directives used along with header such as Cache-Control.

---------------------------------------------------------------------------------------

**REST interview question 2:**

What are those comma separated directives of Cache-Control header?

**This is how I would answer this question**

The comma separated directives of Cache-Control headers are

private, public, no-cache, no-store, no-transform, max-age, s-maxage.

---------------------------------------------------------------------------------------

**REST interview question 3:**

**What are the differences between no-cache, and no-store directives used along with**

**Cache-Control header?**

**This is how I would answer this question**

no-cache can be set in response in order to inform client/browser that this response

should not be used for caching content and any of the cache data should not be sent to

server without revalidation from server.

While no-store is to inform client/browser as not to store any data in response in local

hard disk of the machine that is used for sending the request.

In case of no-cache, one can use data with revalidation, but in no-store that is no ways

any data can be retrieved locally from the hard disk and data won't be available when machine

if restarted.

---------------------------------------------------------------------------------------

**REST interview question 4:**

**If any intermediary proxy is not used to server any response to client's request,**

**is there any difference between private and public directives of Cache-Control?**

**This is how I would answer this question**

No, as private directive may be used to restrict cahing at proxy/CDN server that could

be some intermediary destinations while serving response.

---------------------------------------------------------------------------------------

**REST interview question 5:**

**What is the difference in usage for the s-maxage and max-age directives?**

**This is how I would answer this question**

s-maxage may be used for the proxy/CDN server to know that this is the directive

for as maximum age for the data that is sent as response.

While max-age is used as a directive to the client as the maximum age for the data

that is sent along with response.

---------------------------------------------------------------------------------------

**REST interview question 6:**

**What are the various annotations available from JAX-RS api specification,**

**for sending data from client-side to service endpoint?**

**This is how I would answer this question**

JAX-RS has provided various annotations for passing data from client-side code

to service are @PathParam, @FormParam, @MatrixParam, @QueryParam, @HeaderParam, @CookieParam.

These annotations are also known as Injection Annotations from JAX-RS API.

---------------------------------------------------------------------------------------

**REST interview question 7:**

**How to inject Web container related values and configurations to the JAX-RS service**

**implementation instance**

**This is how I would answer this question**

There is an annotation @Context provided by JAX-RS specification for the REST service

Implemntation to be able to receive helper and web container specific configuration values.

---------------------------------------------------------------------------------------

**REST interview question 8:**

**Can you write-down an example of MatrixParam expression?**

**This is how I would answer this question**

For example, for the following URI:

/employees/name=Ishtek;age=34

@MatrixParam("age") would return 34 as value, as matrix param precedes with a ';' as

separator.

---------------------------------------------------------------------------------------

**Are you aware of @BeanParam annotation?**

**Auther's View point/Answer to above question:**

Yes, @BeanParam annotation is added in JAX-RS 2.0 version. This annotation can be used

along with a Bean class for using other annotation types such as @FormParam, @HeaderParam etc.

as the field level, for using an application specific bean class as argument in the service

method, rather than using a long list of argument parameters for each of the different type of

attributes used along with a request.

---------------------------------------------------------------------------------------

**Can you elaborate on usage of @BeanParam with an example?**

**Auther's answer to above question:**

Suppose there is a HTML form with ten fields/attributes that is used to receive input from user,

and this data/fields are to be submitted to the service method, then the REST service method would

require to define all the ten attributes as arguments for the service method along with @FormParam

annotation. Instead @BeanParam can be used to declare an application/user defined bean class with

all these ten attributes as fields. This user defined class can be a single argument to the REST

service method argument. This way there could be minimal impact when number of fields changes while

using POST as HTTP method.

---------------------------------------------------------------------------------------

**How to approach for change in attributes to the service method in a Webservice ?**

**Auther's view/answer :**

In order to minimize change/impact on the client side of code, when there is a change in

argument parameters of the service method, one can choose to use user defined bean class

as argument to service method, rather than using all the arguments directly in the service

method definition. In this way, if there is a need for addition or removal of any attribute/argument

from the service method, no change to the service method definition would be needed.

---------------------------------------------------------------------------------------

**What are the major differences you can state while using SOAP or REST, in terms of**

**applicability as concern?**

Some of the differences that may be observed in applicability of SOAP or REST as the

service language/specification :

1. When requirement is to provide a business process as a service, then SOAP may get

little more attention than RESTful services.

2. When we are exposing a server side object as many different type of representations

for the client, such as JSON, TXT, XML, Audio, Video and many more (HTTP content types)

etc. RESTful services can be used/more appropriate than SOAP.

3. In case of contract/interface based service definitions are to be used, then SOAP can be used.

4. In case of exposing a service for any type of devices, be it Desktop/Laptop/Netbook, Tablet,

Mobile phones, Kindle etc., and consumer can be a browser (Thin client) or a native application

(Thick client). In this circumstances we can opt for RESTful services.

5. In case of many different types of transports are to be used for using a service, then SOAP would

be appropriate over RESTful service.

6. For looking for standards-based service declarations and usage, SOAP has many standards to use,

such as WS-\* standards. Whereas RESTful services would be a specification way of exposing and

using any service.

7. Looking at slightly more technical aspects of SOAP, SOAP supports custom objects definitions using

XML Schema and marshalling/unmashalling of various datatypes to communicate across diverse platforms.

---------------------------------------------------------------------------------------

**How can you apply security to RESTful services**

Some of the options available to use for securing a RESTful service, for now, are

1. Basic Authentication

This type of Authentication will require transport level encryption(SSL), as user

credentials are to be sent via wire in plain text.

2. OAuth 1.0a / OAuth 2.0

OAuth 1.0a is using advanced encryption for passing token for authentication purposes.

OAuth 2.0 is using SSL for transport level security.

3. Custom/Third-party security protocol

---------------------------------------------------------------------------------------

**What is the main factor to consider while choosing OAuth version to use, whether to use**

**OAuth 1.0a or OAuth 2.0?**

The main reson is the sensitivity of the data that is exchanged, and transport level

security related considerations. If the application data is less sensitive,

the OAuth 1.0a could be well enough for use, and OAuth 1.0a specification can be applied

without much of encryptions on transport. But OAuth 2.0 would rely on HTTPS transport

level security/encryption for communication.

---------------------------------------------------------------------------------------

**What are the various credential types used along with OAuth 2.0?**

There are three types of credentials available to use along with OAuth 2.0, such as

Bearer Token, MAC token, SAML.

---------------------------------------------------------------------------------------

**What are the HTTP methods corresponding to CRUD operations?**

POST - Create

GET - Read

PUT - Update

DELETE - Delete

are the corresponding HTTP method used for CRUD operations with resource(s).

---------------------------------------------------------------------------------------

**Can you write a very simple code showing resource being exposed as RESTful service?**

@Path("/book")

public class Book {

@GET

@Path("{id}")

public Book getBookInfo(@PathParam("id") String bookId) {

//return Book Instance by using value bookId.

return new Book();

}

}

Acessing this Book resource by using an URI as <<WEB\_APP\_NAME>>/book/b001

---------------------------------------------------------------------------------------

**What are the annotations that can be used for specifying content-type that is supported**

**by any RESTful service?**

@Produces("text/xml") and @Consumes("text/xml") are the annotations that are used for specifying

ways of defining any restrictions that can be defined at method-level for any RESTful service.

---------------------------------------------------------------------------------------

**For using JAXB supported XML to Custom-object and Custom-object to XML mapping/conversion,**

**along with RESTful service, what are the annotations those can be used along with custom object?**

@XmlRootElement, @XmlElement, @XmlAttribute, @XmlAccessorType etc., are the annotations from JAXB

can be used along with custom class for defining class, field level fields.

---------------------------------------------------------------------------------------

**As @FormParam can be used for passing form parameters in request, but in case parameter**

**set is likely to change (parameters can be added or removed), then how to insulate RESTful service**

**method definition from change?**

MultivaluedMap<String, String> type can be used to define argument parameter for the RESTful method

signature for passing key and value pair in request.

---------------------------------------------------------------------------------------

**Is there any ways to provide custom/own JAXBContext for marshalling/unmarshalling XML to Object**

**and vice versa, rather than using default JAXBContext as available with JAX-RS provider?**

Yes, by implementing class file that implements ContextResolver<JAXBContext> and overriddes

public getContext (Class<?>) method for returning custom implementation of JAXBContext.

**Added on this page, as of 03-July-2014:**

**Have you used Jersey framework or any other implementation for JAX-RS specification?**

**Answer :**

Jersey Framework can be used along with web container that is having support for JAX-RS

or not. Where as any web container that has support for JAX-RS specification, can be used

to provide a resource as RESTful web service.

---------------------------------------------------------------------------------------

**Is there any ways to code so as to provide HTTP headers to a method in the main RESOURCE class**

**file?**

**Answer :**

Yes, by using @Context as argument type for the method that is exposed as REST uri.

import javax.ws.rs.core.HttpHeaders;

...

...

@GET

@Path("abc")

public void getValue(@Context HttpHeaders headers) {

...

...

}

---------------------------------------------------------------------------------------

**Have you used Maven to generate Jersey based RESTful services? if yes, how?**

**Answer :**

I have used a archetype from Maven repository, called as 'jersey-quickstart-webapp',

for generating a web application, just to start head's up on using Jersey for creating

RESTful services.

---------------------------------------------------------------------------------------

**Can you be able to provide certain context-level parameters in web.xml, and receiving**

**parameter value in the resource method?**

**Answer :**

One can define context parameter in web.xml file and corresponding value of the parameter

in the resource method, by using @Context annotations with instance level variable of

ServletContext type.

---------------------------------------------------------------------------------------

**What are the headers types from HTTP request, those can be used for mapping same URI**

**but different resource methods?**

**Answer :**

One can use Accept, Accept-Language, Accept-Encoding, Content-Type with appropriate values

for mapping same URI but different methods. Appropriate method would be called that is

receiving corresponding values in form of those headers, in the HTTP REQUEST.

Please keep visiting this page... as more questions will be added here,

as and when available.

---------------------------------------------------------------------------------------

**Is that any ways to approach so that changes(addition/deletion) in the form-level parameters**

**would not have impact on the Resource method signature/arguments**

**Answer :**

One can explore javax.ws.rs.core.MultivaluedMap<Object, Object> and its implementations

for providing ways to pass form-level POST parameters and corresponding values. This way

additions and deletions won't effect method signature to some level.

---------------------------------------------------------------------------------------

**What is the encoding media type associated with a FORM data?**

**application/x-www-form-urlencoded is the media type of the FORM data**

---------------------------------------------------------------------------------------

**As per JAX-RS specification, what are the corresponding annotations used for various HTTP operations,**

**such as GET, PUT, POST, DELETE, HEAD etc.?**

**Answer :**

Various annotations for these operations are

@javax.ws.rs.GET for HTTP GET Operation

@javax.ws.rs.PUT for HTTP PUT Operation

@javax.ws.rs.POST for HTTP POST Operation

@javax.ws.rs.DELETE for HTTP DELETE Operation

@javax.ws.rs.HEAD for HTTP HEAD Operation

---------------------------------------------------------------------------------------

**For an example, if the requirement is to retrieve all the books in a library and the result would be in**

**XML format, how would you write a code for this in simplest form, or just write the Java class that does this mapping ?**

**Answer :**

The main class file would have to have a Path configured, here I shall show this using annotations,

@Path("/library")

public class LibraryDemoService {

//Then I shall provide a method with public as access modifier and with appropriate annotations

//such as GET and Produces

//GET for making this method ready to receive any HTTP GET method request and Produces denotes that

//this method would return a result in Text/XML as return type.

@GET

@Path("books")

@Produces("application/xml")

public String retrieveAllBookInLibrary() {

return "<books><book></book><books/>"; //or any other ways of forming the actual return result

}

}

---------------------------------------------------------------------------------------

**In the code that you have just written, can you add a separate method for retrieving a selected book with a ISBN code?**

**Answer**

Sure, we can add another method with a path for book and a parameter for ISBN code, may be something like as shown below,

@GET

@Path("books/book/{isbnCode}")

@Produces("text/xml")

public String retriveBook(@PathParam("isbnCode") String isbnCode) {

return "<book></book>"; //or any other ways of forming the actual return result

}

---------------------------------------------------------------------------------------