**Using Postman we can make complex API requests(like GET & POST request methods)Postman is a powerful HTTP client for testing web services.**

**==>Web Services: Exchange of the data happens over the web (Http).**

**==>Protocol: When a client makes a request, they are usually messages that are sent from one machine to another.**

**These messages need to be in a format that both the client and server can understand.**

**This message format is refered to as a Protocol.**

**This Protocol is sometimes standardized in some Web Services. For example, a SOAP web service uses standard protocol for communication ==> SOAP**

**==>SOAP(Simple Access Object Protocol): SOAP webservices use SOAP protocol which is a specific format of messages. The messages are in an XML format,**

**and there are specific rules which detail how the XML should be, so the client and the server needs to talk each other using SOAP protocol only.**

**==>Whereas in REST there is no specific Protocol**

**\*It can send messages using XML, JSON or Text format etc**

**==>RESTful Web Services: The Request and the Response is happening over the Web. They exchange over HTTP. In REST messages can be exchanged in**

**any Http method(GET, POST, PUT, etc.)**

**1) Explain what is REST and RESTFUL?**

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

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

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

|  |  |  |
| --- | --- | --- |
| **No.** | **SOAP** | **REST** |
| 1) | SOAP is a **protocol**. | REST is an **architectural style**. |
| 3) | SOAP **can't use REST** because it is a protocol. | REST **can use SOAP** web services because it is a concept and can use any protocol like HTTP, SOAP. |
| 4) | SOAP **uses services interfaces to expose the business logic**. | REST **uses URI to expose business logic**. |
| 6) | SOAP **permits XML** data format only. | REST **permits different** data format such as Plain text, HTML, XML, JSON etc. |

**2) Explain the architectural style for creating web api?**

The architectural style for creating web api are

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

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

[SOAPUI](https://goo.gl/XczVPL) tool for SOAP WS and Firefox “poster” plugin for RESTFUL services.

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

HTTP methods supported by REST are:

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

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

No, you are not supposed to use POST or GET.  GET operations should only have view rights

**6) Mention what are resources in a REST architecture?**

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

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

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

**7) Mention some key characteristics of REST?**

Some key characteristics of REST includes

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

**8) Mention what are the different application integration styles?**

The different integration styles includes

* Shared database
* Batch file transfer
* Invoking remote procedure (RPC)
* Swapping asynchronous messages over a message oriented middle-ware (MOM)

**9) Explain how JAXB related to RESTful web api?**

JAXB stands for java arch for xml binding.

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

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

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

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

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

**12) Mention what is the difference between RPC or document style web services? How you determine to which one to choose?**

In document style web services, we can transport an XML message as part of SOAP request which is not possible in RPC style web service.  Document style web service is most appropriate in some application where XML message behaves as document and content of that document can alter and intention of web service does not rely on the content of XML message.

**13) Mention what is JAX-WS and JAX-RS?**

Both JAX-WS and JAX-RS are libraries (APIs) for doing communication in various ways in Java.  JAX-WS is a library that can be used to do SOAP communication in JAVA, and JAX-RS lets you do the REST communication in JAVA.

**14) List out the tools or API for developing or testing web api?**

Testing tools for web services for REST APIs includes

* Spring REST web service using MVC
* Jersey API
* CFX
* Axis
* Restlet,

**15) Mention what is the difference between SOAP and REST?**

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

**Q1  What are web services ?**  
  
According to [oracle docs](http://docs.oracle.com/javaee/6/tutorial/doc/gijvh.html), web services can be defined as

*Web services are client and server applications that communicate over the World Wide Web’s (WWW) HyperText Transfer Protocol (HTTP). Web services provide a standard means of inter operating between software applications running on a variety of platforms and frameworks.*

Main characteristics of the Web Services  are :  
  
1. Interoperability   
2. Extensibility  
3. Machine processable descriptions.  
  
for example in simple words , when we call somebody so the person dialing and calling is the client  application , while person receiving the call is server applicationand "hello" word is the protocol as similar to HTTP request .  
  
  
**Q2** **What is the difference between SOA and a web service?**

SOA (Service-Oriented Architecture) is an architectural pattern that makes possible for

services to interact with one another independently.

Web Services is a realization of SOA concept, that leverages XML, JSON, etc. and common Internet protocols such as HTTP(S), SMTP, etc.

SOA is a system-level architectural style that tries to expose business. WOA is an interface-level architectural style that focuses on the means by which these service capabilities are exposed to consumers.  
 **Q3 What is SOAP?**  
*SOAP* *(*Simple Object Access Protocol*)*is a transport protocol for sending and receiving requests and responses on XML format, which can be used on top of transport protocols such as HTTP, SMTP, UDP, etc.

**Q4** **What is REST?**

REST (REpresentational State Transfer) is an architectural style by which data can be transmitted over transport protocol such as HTTP(S).  
 **Q5  What is the difference between a REST web service and a SOAP web service?**

Below are the main differences between REST and SOAP web service

* REST supports different formats like text, JSON and XML; SOAP only supports XML;
* REST works only over HTTP(S) on a transport layer; SOAP can be used different protocols on a transport layer;
* REST works with resources, each unique URL is some representation of a resource; SOAP works with operations, which implement some business logic through different interfaces;
* SOAP based reads can’t be cached, for SOAP need to provide caching; REST based reads can be cached;
* SOAP supports SSL security and WS-security(Web Service-security); REST only supports SSL security;
* SOAP supports ACID (Atomicity, Consistency, Isolation, Durability); REST supports transactions, but it is neither ACID compliant nor can provide two phase commit.

**Q6 How  to decide which one of web service to use REST or SOAP?**  
  
“REST vs SOAP” we can rephrased to "Simplicity vs Standard". Of course, "Simplicity" with REST at most cases wins, it wins in performance, scalability and support for multiple data formats, but SOAP is favored where service requires comprehensive support for security (WS-security) and transactional safety (ACID).

**“SOAP”**  
 **Q7  What is WSDL?**

WSDL (Web Services Description Language) is an XML format for describing web services and how to access them.

**Q8  What is JAX-WS?**

JAX-WS (Java API for XML Web Services) is a set of APIs for creating web services in XML format.

**Q9 What is JAXB?**

JAXB (Java Architecture for XML Binding) is a Java standard that defines how Java objects are converted from and to XML. It makes reading and writing of XML via Java relatively easy.

**Q10 Can we send soap messages with attachments?**

Yes, we can send different formats such as PDF document, image or other binary file with soap messages as an attachment. Messages send using the binary data. SOAP messages is attached with MIME extensions that come in multipart/related.

An example:

MIME-Version: 1.0

Content-Type: Multipart/Related; boundary=MIME\_boundary; type=text/xml;

        start="<claim061400a.xml@ javahungry.com>"

Content-Description: This is the optional message description.

<?xml version='1.0' ?>

<SOAP-ENV:Envelope

xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Body>

..

<theSignedForm href="cid:claim061400a.tiff@javahungry.com"/>

..

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

--MIME\_boundary

Content-Type: image/tiff

Content-Transfer-Encoding: binary

Content-ID: <claim061400a.tiff@javahungry.com>

...binary TIFF image...

--MIME\_boundary—  
 **Q11 What is MTOM?**

MTOM (Message Transmission Optimization Mechanism) is a mechanism for transmitting large binary attachments with SOAP messages as raw bytes, allowing for smaller messages.

**Q12 What is XOP?**

XOP (XML-binary Optimized Packaging) is a mechanism defined for the serializationof XML Information Sets that contain binary data, as well as deserialization back into the XML Information Set.

**Q13 What is a SOAP envelope element?**

SOAP envelop element is the root element of a SOAP message which defines the XML document as a SOAP message.

An example:

<?xml version="1.0"?>  
<soap:Envelope  
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"  
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">  
  ...  
  Message information  
  ...  
</soap:Envelope>  
 **Q14 What does a SOAP namespace defines?**

SOAP *namespace*defines the Envelope as a *SOAP* Envelope.

An example:

xmlns:soap=http://www.w3.org/2001/12/soap-envelope

**Q15 What is the SOAP encoding?**

SOAP encoding is a method for structuring the request which is suggested within the SOAP specification, known as the SOAP serialization.

**Q16 What does SOAP encodingStyle attribute defines?**

SOAP encodingStyle defines the serialization rules used in a SOAP message. This attribute may appear on any element, and is scoped to that element's contents and all child elements not themselves containing such an attribute. There is no default encoding defined for a SOAP message.

An example:

SOAP-ENV:encodingStyle="http://www.w3.org/2001/12/soap-encoding"

**Q17 What are 2 styles web service’s endpoint by using JAX-WS?**

* RPC (remote procedure call) style web service in JAX-WS;
* document style web service in JAX-WS.

**Q18 What is encoding rules for header entries?**

* a header entry is identified by its fully qualified element name, which consists of the namespace URI and the local name. All immediate child elements of the SOAP Header element must be namespace-qualified.
* the SOAP encodingStyle attribute may be used to indicate the encoding style used for the header entries.
* the SOAP mustUnderstand attribute and SOAP actor attribute may be used to indicate how to process the entry and by whom.

**Q19 What is the wsimport tool?** 

The wsimport tool is used to parse an existing Web Services Description Language (WSDL) file and generate required files (JAX-WS portable artifacts) for web service client to access the published web services: https://docs.oracle.com/javase/6/docs/technotes/tools/share/wsimport.html

**Q20 What is the wsgen tool?**

The wsgen tool is used to parse an existing web service implementation class and generates required files (JAX-WS portable artifacts) for web service deployment: http://docs.oracle.com/javase/6/docs/technotes/tools/share/wsgen.html

* What the tool are required to test SOAP services?

      SOAPUI tool for SOAP WS: http://www.soapui.org/

**Q21 What is the difference between SOAP and other remote access techniques?**

* SOAP is simple to use and it is non - symmetrical unlike DCOM or CORBA is highly popular and usually have complexity in them.
* SOAP provides greater platform independent with the language independence unlike DCOM or CORBA doesn't provide any of these.
* SOAP uses HTTP as its transport protocol and the data are being saved in XML format that can be ready by human, whereas DCOM or CORBA have their own binary formats that are used to transport the data in complicated manner.   
  SOAP identify the object other than URL endpoint. SOAP objects are stateless and it is hard to maintain that. Whereas, it is not hard to maintain in case of other remote access techniques.

**“REST”**

**Q22 What is a resource in a REST?**

      A resource is a unique URL with representation of an object which we can get contents via GET and modify via PUT, POST, DELETE.

**Q23 What are HTTP methods supported by REST?**

* GET;
* POST;
* PUT;
* DELETE;
* OPTIONS;
* HEAD.

**Q24 Whether can use GET request instead of POST to create a resource?**

It is not possibly, because **GET can’t change a resource.**

**Q25 What is the difference between PUT and POST?**

Need to use PUT when can update a resource completely through a specific resource. For example, if know that an article resides at http://javahungry.blogspot.com/article/123, can PUT a new resource representation of this article through a PUT on this URL. If do not know the actual resource location for instance, when add a new article, can use POST.

PUT is idempotent, while POST is not. It means if use PUT an object twice, it has no effect.

**Q26 What is WADL?**

WADL (Web Application Description Language) is a XML description of a deployed RESTful web application.

**Q27 What are frameworks available to implement REST web services?**

Jersey, Restlet, EasyRest, etc.

**Q28 What is the Restlet framework?**

Restlet is a lightweight, comprehensive, open source RESTful web API framework for the Java platform.

It has advantages such as

* websocket and server-sent events support;
* HTTP/2 support;
* transparent HTTP PATCH support;
* client cache service;
* fluent APIs.

http://restlet.com/

**Q29 What is the Jersey framework?**

Jersey is open source framework for developing RESTful Web Services in Java that provides support for JAX-RS APIs and serves as a JAX-RS (JSR 311 & JSR 339) Reference Implementation. It has advantages such as

* contains support for Web Application Description Language (WADL);
* contains Jersey Test Framework which lets run and test Jersey REST services inside JUnit;
* supports for the REST MVC pattern, which would allow to return a View from Jersey services rather than just data.

https://jersey.java.net/

**Q30 What is the RESTeasy framework?**

RESTeasy is a JBoss project, which implements of the JAX-RS specification. It has benefits such as

* fully certified JAX-RS implementation; supports HTTP 1.1 caching semantics including cache revalidation;
* JAXB marshalling into XML, JSON, Jackson, Fastinfoset, and Atom as well as wrappers for maps, arrays, lists, and sets of JAXB Objects;
* OAuth2 and Distributed SSO with JBoss AS7;
* rich set of providers for: XML, JSON, YAML, Fastinfoset, Multipart, XOP, Atom, etc.

http://resteasy.jboss.org/

**Q31 What is the difference between AJAX and REST?**

* in Ajax, the request are sent to the server by using XMLHttpRequest objects; REST have a URL structure and a request/response pattern the revolve around the use of resources;
* Ajax eliminates the interaction between the customer and server asynchronously; REST requires the interaction between the customer and server;
* Ajax is a set of technology; REST is a type of software architecture and a method for users to request data or information from servers.

**Q32 What tool are required to test REST services?**

Firefox “poster” plugin for RESTFUL services. https://addons.mozilla.org/en-us/firefox/addon/poster/

[](http://1.bp.blogspot.com/-l_4Sb1tEk_c/VbSfpcirsbI/AAAAAAAAAo4/zqLjasVLwWw/s1600/java+web+services+interview+questions+and+answers.jpg)

**Q33 What does a @Path annotation do?**

     @Path annotation binds URI pattern to a Java method.

**import** **javax.ws.rs.GET**;

**import** **javax.ws.rs.Path**;

**import** **javax.ws.rs.core.Response**;

**@Path**("/persons")

public **class** **PersonRestService** {

**@GET**

public Response getPerson() {

**return** Response.status(**200**).entity("getPerson is called").build();

}

**@GET**

**@Path**("/vip")

public Response getPersonVIP() {

**return** Response.status(**200**).entity("getPersonVIP is called").build();

}

}

On calling URI: “/persons” result:  getPerson is called

On calling URI: “/persons/vip” result:  getPersonVIP is called

**Q34  What does a @PathParam do?**

     @PathParam annotation injects the value of URI parameter that defined in @Path expression.

**import** **javax.ws.rs.GET**;

**import** **javax.ws.rs.Path**;

**import** **javax.ws.rs.PathParam**;

**import** **javax.ws.rs.core.Response**;

**@Path**("/persons")

public **class** **PersonRestService** {

**@GET**

**@Path**("{id}")

public Response getPersonById(**@PathParam**("id") String id) {

**return** Response.status(**200**).entity("getPersonById is called, id : " + id).build();

}

}

On calling URI: “/persons/1” result:  getPersonById is called, id : 1

**Q35  What does a @QueryParam do?**

    @QueryParam annotation injects URI query parameter into Java method.

**import** **java.util.List**;

**import** **javax.ws.rs.GET**;

**import** **javax.ws.rs.Path**;

**import** **javax.ws.rs.QueryParam**;

**import** **javax.ws.rs.core.Response**;

**@Path**("/persons")

public **class** **PersonService** {

**@GET**

**@Path**("/query")

public Response getPersons(

**@QueryParam**("from") int from,

**@QueryParam**("to") int to,

**@QueryParam**("orderBy") List&lt;String&gt; orderBy) {

**return** Response

.status(**200**)

.entity("getPersons is called, from : " + **from** + ", to : " + to

+ ", orderBy" + orderBy.toString()).build();

}

}

On calling URI: “/persons/query?from=10&to=20&orderBy=age&orderBy=name” result: getPersons is called, from : 10, to : 20, orderBy**[**age, name**]**

**Q36  What does a @MatrixParam do?**

@MatrixParam are a set of **“name=value”** in URI path.

**import** **javax.ws.rs.GET**;

**import** **javax.ws.rs.MatrixParam**;

**import** **javax.ws.rs.Path**;

**import** **javax.ws.rs.PathParam**;

**import** **javax.ws.rs.core.Response**;

**@Path**("/books")

public **class** **BookService** {

**@GET**

**@Path**("{year}")

public Response getBooks(**@PathParam**("year") String year,

**@MatrixParam**("author") String author,

**@MatrixParam**("country") String country) {

**return** Response

.status(**200**)

.entity("getBooks is called, year : " + year

+ ", author : " + author + ", country : " + country)

.build();

}

}

On calling URI: “/books/2015” result: getBooks is called, year : 2015, author : null, country : null

On calling URI: “/books/2015;author= doyle;country=scotland” result: getBooks is called, year : 2015, author : doyle, country : scotland

**Q37  What does a @FormParam do?**

@FormParam bind HTML form parameters value to a Java method.

**import** **javax.ws.rs.FormParam**;

**import** **javax.ws.rs.POST**;

**import** **javax.ws.rs.Path**;

**import** **javax.ws.rs.core.Response**;

**@Path**("/persons")

public **class** **PersonService** {

**@POST**

**@Path**("/add")

public Response addPerson(

**@FormParam**("name") String name,

**@FormParam**("age") int age) {

**return** Response.status(**200**)

.entity("addPerson is called, name : " + name + ", age : " + age)

.build();

}

}

HTML form:

<html>

<body>

<form action="/persons/add" method="post">

<p>

Name : <input type="text" name="name" />

</p>

<p>

Age : <input type="text" name="age" />

</p>

<input type="submit" value="Add Person" />

</form>

</body>

</html>

**Q39  How to get HTTP request header in JAX-RS (2 ways)?**

* inject directly with @HeaderParam;

**import** **javax.ws.rs.GET**;

**import** **javax.ws.rs.Path**;

**import** **javax.ws.rs.HeaderParam**;

**import** **javax.ws.rs.core.Response**;

**@Path**("/persons")

public **class** **PersonService** {

**@GET**

**@Path**("/get")

public Response getPerson(

**@HeaderParam**("person-agent") String personAgent) {

**return** Response.status(**200**)

.entity("getPerson is called, personAgent : " + personAgent)

.build();

}

}

On calling URI: “/persons/get” result: getPerson is called, personAgent : Mozilla**/**5.0 **(**Windows NT 6.1; rv:5.0**)** Gecko**/**20100101 Firefox**/**5.0

* pragmatically via @Context.

**import** **javax.ws.rs.GET**;

**import** **javax.ws.rs.Path**;

**import** **javax.ws.rs.core.Context**;

**import** **javax.ws.rs.core.HttpHeaders**;

**import** **javax.ws.rs.core.Response**;

**@Path**("/persons")

public **class** **PersonService** {

**@GET**

**@Path**("/get")

public Response getPerson(**@Context** HttpHeaders headers) {

String personAgent = headers.getRequestHeader("person-agent").get(**0**);

**return** Response.status(**200**)

.entity("getPerson is called, personAgent : " + personAgent)

.build();

}

}

On calling URI: “/persons/get” result: getPerson is called, personAgent : Mozilla**/**5.0 **(**Windows NT 6.1; rv:5.0**)** Gecko**/**20100101 Firefox**/**5.0

**Q40  How to download file in JAX-RS?**

* put @Produces(“?”) on service method, with a Response return type. Instead “?” write a type text/plain, image/png, etc.
* set “Content-Disposition” in Response header to tell browser pop up a download box for user to download.

**import** **java.io.File**;

**import** **javax.ws.rs.GET**;

**import** **javax.ws.rs.Path**;

**import** **javax.ws.rs.Produces**;

**import** **javax.ws.rs.core.Response**;

**import** **javax.ws.rs.core.Response.ResponseBuilder**;

**@Path**("/image")

public **class** **ImageService** {

private static final String FILE\_PATH = "c:**\\**my.png";

**@GET**

**@Path**("/get")

**@Produces**("image/png")

public Response getFile() {

File file = new File(FILE\_PATH);

ResponseBuilder response = Response.ok((Object) file);

response.header("Content-Disposition",

"attachment; filename=image\_from\_server.png");

**return** response.build();

}

}