**Techie Community . neT**

[**Spring 3**](http://www.techiecommunity.net/Spring3) [**Hibernate 3**](http://www.techiecommunity.net/Hibernate3) [**Android**](http://www.techiecommunity.net/Android) [**Home**](http://www.techiecommunity.net/)

**REST Interview Questions Answer   
Author : Ishtek**

**Author: Ishtek**

**Date: 25-March-2014**

Disclaimer:

While this page Author strive to ensure 100% accuracy with content, this page Author and/or

this web site can not be held responsible for any damages caused by the information presented

in this page. We request all readers to report any incorrect/wrong question and/or answer,

by sending an email to: **reaching @ techiecommunity.net** . Appropriate action will be

taken after thorough reviews.

|  |  |
| --- | --- |
|  |  |

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

}

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

|  |  |
| --- | --- |
|  |  |

Any of these questions and/or answer you think those needs changing/ requires correction,

please share your point of view, by using following form.

|  |  |  |
| --- | --- | --- |
|  | | |
| **If interested in commenting related to REST Interview Questions with Answer** | **Click Here** |  |
| Top of Form  Name (To be shown on this Page:\*  Email (Not to be shown):\*  Website (To be shown on this Page)  Your Comments here (To be shown on this Page):\*  (You can use <a> Tag in your comments)  Please enter same number as shown here **844**      Bottom of Form | | |

|  |  |
| --- | --- |
| Name: **Yogesh** | Dated: 2014-04-02 17:26:47 |
| Thanks for nice set of questions put together here.  I am looking for certain questions from usability of REST with a comparison with SOAP-based web services. | |
|  | |  |  | | --- | --- | | Name: **Santosh** | Dated: 2014-05-13 20:08:27 | | Good set of questions and good approach to answer. | |  |  |  | | --- | --- | | Name: **Nilesh** | Dated: 2014-09-10 18:23:56 | | Very GOOD Ishtek. I like the set. Please add more question-Answer set in this. | |  |  |  | | --- | --- | | Name: **SubbaRao Boddu** | Dated: 2014-12-04 16:15:39 | | Very Good questions...Keep it Up | |  |  |  | | --- | --- | | Name: **BX** | Dated: 2014-12-19 07:13:34 | | great collection man, appreciate your effort. | | |

|  |  |
| --- | --- |
|  | |
| Reply | **Click Here** |
| Top of Form  Name (To be shown on Page):\*  Email (Optional): (never shown)  Website (To be shown on this Page)  Your Comments here (To be shown on Page):\*  Please enter same number as shown here **140**    Bottom of Form | |

|  |  |
| --- | --- |
| Name: **Chandu** | Dated: 2014-09-09 09:51:17 |
| Very nice pick....very useful  Thanks a lot... | |
|  | |  |  | | --- | --- | | Name: **Jeetu** | Dated: 2014-12-12 10:11:14 | | Good explanation. | |  |  |  | | --- | --- | | Name: **Sandip** | Dated: 2014-12-26 23:56:07 | | Vedry informative article on REST. Thank you very much. | | |

|  |  |
| --- | --- |
|  | |
| Reply | **Click Here** |
| Top of Form  Name (To be shown on Page):\*  Email (Optional): (never shown)  Website (To be shown on this Page)  Your Comments here (To be shown on Page):\*  Please enter same number as shown here **140**    Bottom of Form | |

[Android Thread Example](http://www.techiecommunity.net/Android/Android-Thread-Example)

[REST Interview Questions](http://www.techiecommunity.net/Webservices/REST-Interview-Questions-Answer)

[JAX-WS Webservice Example](http://www.techiecommunity.net/Webservices/JAX-WS-Webservice-example)

[Android Overlap SurfaceView Example](http://www.techiecommunity.net/Android/Android-Example-Overlap-SurfaceView)

[Spring3 Integration of File Endpoint](http://www.techiecommunity.net/Spring3/Spring3-Integration-File-Example)

[JAXWS Webservice Example](http://www.techiecommunity.net/Webservices/JAXWS-Webservice-Example)

[Android WebView Example](http://www.techiecommunity.net/Android/Android-WebView-Example)

[Android Context Menu Example](http://www.techiecommunity.net/Android/Android-Context-Menu-Example)

[Setting up Eclipse for Android App Development](http://www.techiecommunity.net/Android/Setup-Eclipse-Android-ADT-Plugin-Environment)

[Android SensorEvent Orientation Example](http://www.techiecommunity.net/Android/Android-SensorEvent-Orientation-Example)

[Android ImageSwitcher Example](http://www.techiecommunity.net/Android/Android-ImageSwitcher-Example)

[REST API Design for Android App](http://www.techiecommunity.net/Android/Android-REST-Api-Design)

[Android ExpandableListView Example](http://www.techiecommunity.net/Android/Android-ExpandableListView-Layout-Example)

[Singleton Serializable on Android](http://www.techiecommunity.net/Android/Android-Singleton-Serializable)

[Android GPS with Google MAP Example](http://www.techiecommunity.net/Android/Android-Project-GPS-Google-Map-Places)

[Android Project With Example](http://www.techiecommunity.net/Android/Android-Project-With-Example)

[Android Project Manageing Files](http://www.techiecommunity.net/Android/Android-Projects-Manage-Files-Extension)

[Project Ideas for App on Android Platform](http://www.techiecommunity.net/Android/Android-students-projects-ideas)

[Android EditText with Background](http://www.techiecommunity.net/Android/Android-EditText-Background-Example)

[Android SQLite Database Example](http://www.techiecommunity.net/Android/Android-SQLiteDatabase-Example)

[Android AutoCompleteTextView Example](http://www.techiecommunity.net/Android/Android-AutoCompleteTextView-Example)

[Android Tester](http://www.techiecommunity.net/Android/Android-Tester)

[Android Interview Questions](http://www.techiecommunity.net/Android/Android-Interview-Questions-Answer)

[Android ListView Example](http://www.techiecommunity.net/Android/android-listview-example-article)

[Android Intent Notification Example](http://www.techiecommunity.net/Android/Android-Receiving-Intent-Notification-example)

[Android Intent Broadcast Example](http://www.techiecommunity.net/Android/Android-Intent-Broadcast-Example)

[Android TextView Link Example](http://www.techiecommunity.net/Android/Android-TextView-Link-Example)

[Android Sensor List](http://www.techiecommunity.net/Android/android-sensors-list)

[Android Sensor Orientation Example](http://www.techiecommunity.net/Android/Android-Sensor-Orientation-Example)

[Android Intent Broadcast Receiver](http://www.techiecommunity.net/Android/Android-Intent-Broadcast-Receiver-Example)

[Android Gallery SurfaceViews Example](http://www.techiecommunity.net/Android/android-Gallery-surfaceviews-spinner)

[Android Location Example](http://www.techiecommunity.net/Android/Android-Location-Example)

[Android Text to Speech Example](http://www.techiecommunity.net/Android/android-example-text-2-speech)

[Android Intent Example](http://www.techiecommunity.net/Android/Android-Intent-Example)

[Android Layout Example](http://www.techiecommunity.net/Android/android-layout-example)

[Android Download File Example](http://www.techiecommunity.net/Android/Android-example-download-any-file-sourcecode)

[Android Expandable ListView Example](http://www.techiecommunity.net/Android/android-expandable-list-example)

[Android Tab Example](http://www.techiecommunity.net/Android/android-tab-example)

[Android RelativeLayout Example](http://www.techiecommunity.net/Android/Android-RelativeLayout-Example)

[Android Intent Example](http://www.techiecommunity.net/Android/Android-Intent-Example)

[Android Text to Speech Example](http://www.techiecommunity.net/Android/android-example-text-2-speech)

[Android SQLite Example](http://www.techiecommunity.net/Android/Android-SQLite-Example)

[Android CustomView Example](http://www.techiecommunity.net/Android/Android-CustomView-Example)

[Android Canvas Example](http://www.techiecommunity.net/Android/Android-Canvas-Example)

[Android SharedPreferences Example](http://www.techiecommunity.net/Android/Android-SharedPreferences-Example)

[Android Shape Background Example](http://www.techiecommunity.net/Android/Android-Shape-Background-Example)

[Android ShapeButton Example](http://www.techiecommunity.net/Android/Android-ShapeButton-Example)

[Android ViewFlipper Example](http://www.techiecommunity.net/Android/Android-ViewFlipper-Example)

[Android Gallery Example](http://www.techiecommunity.net/Android/android-Gallery-example)

[Android TimePickerDialog Example](http://www.techiecommunity.net/Android/android-TimePickerDialog-example)

The content provided in this page is NOT warranted and/or NOT guaranteed by techiecommunity.net .

techiecommunity.net is not liable for any negative consequences that may result/arise from

implementing directly/indirectly any information covered in these pages/articles/tutorials.

For any of the content, if you would like to bring it to notice for removal from this web site,

please write to this web site administrator @ [EMAIL-ID](mailto:reaching%20@%20techiecommunity.net?Subject=remove%20/Webservices.php/REST-Interview-Questions-Answer) (please remove space before and after @),

with appropriate concern and supporting proof(s). After thorough review and if found genuine concern,

we would take appropriate action and remove disputed content from this web site within 24 hours

starting from the time it has brought to techiecommunity.net Administrator notice.

**While using this web site, you agree to have read and accepted techiecommunity.net**