RESTful JAVA Web Services Interview Questions

Question 1. In Which Scenarios Restful Web Services Are Preferred Over Soap Ones?

Answer : RESTful Web Services are preferred when:

a) The Web Services are completely stateless.

b) When data can be cached to improve performance.

c) Schema of data can be agreed between service consumer and service provider.

d) Bandwidth is limited.

Question 2. How Do You Refer Or Identify Resources In Rest Design Idiom?

Answer : Resources are identified by their unique URLs.

Question 3. Is Rest Stateless Or Stateful Client-server Architecture?

Answer : REST is stateless client-server architecture.

Question 4. Name The Java Api That Is Used For Building And Deploying Restful Web Services?

Answer :
- JAVA API for XML Web Services (JAX-WS) and
- JAVA API for XML RESTful Web Services (JAX-RS).

Question 5. How Web Service Interface For Restful Web Services Are Described?

Answer : There is no formal way of describing the Web Service interfaces. To overcome this, service producer and service consumer agree on schemas that describe the data being exchanged.

Question 6. Name Few Of The Jax-rs Implementations?

Answer : Jersey, RESTEasy, Restlet are to name a few.

Question 7. If I Plan To Use The Jersey Distribution For Restful Web Services, Do I Need To Download It?

Answer : Yes. But if you are using NetBeans IDE 6.0, downloading is not required.

Question 8. How Are Resources Addressed?

Answer : Each resource is uniquely addressable using a universal syntax for use in hypermedia links.

Question 9. In Restful Architecture, What Represents The Application State And Functionality?

Answer : Uniquely Addressable Resources represent the application state and functionality.

Question 10. What Is Shared Between Restful Resource That Is Used For Transfer Of State Between Client And Resource?

Answer : A uniform interface is shared between resources that comprises of a set of well-defined operations, content types and optionally supporting code-on-demand.

Question 11. How Do You Annotate A Method To Respond To Http Get Requests?

Answer : javax.ws.rs.Get annotation specifies that the annotated method will serve HTTP Get request.

Question 12. Name The Most Commonly Http Methods Used In Designing Restful Services?

Answer : PUT, GET, POST, DELETE.

Question 13. Define One-to-one Mapping Between Http Methods And Crud Operations?

Answer : PUT maps to Create, GET maps to Retrieve, POST maps to Update, DELETE maps to Delete.

Question 14. What Are Idempotent Methods?

Answer : A method is said to be idempotent if repeated calls to it does not cause duplicates.

Question 15. In Order To Answer Http Put Requests, How Will You Annotate Your Resource Methods?

Answer : The @PUT annotation is used to answer HTTP PUT requests.

Question 16. Why Put Methods Is Idempotent?

Answer : PUT is used to create a resource and repeating the operation should never create another copy of the resource.

Question 17. What Is The Purpose Of Javax.ws.rs.httpmethod?

Answer : HttpMethod annotation is used to associate the name of a HTTP method with an annotation.

Question 18. List Resource Method Designator Annotations?

Answer : @GET, @PUT, @POST, @DELETE

Question 19. Define Root Resource Class?

Answer : Root resource class is basically a POJO (Plain Old JAVA Objects) that is annotated with @Path and has at least one method annotated with @Path or have at least a single resource method.

Question 20. Define Resource Method?

Answer : Method of a resource class that is annotated with resource method designator annotation is called Resource Method.

Question 21. How To You Specify Relative Path Of A Resource In Restful Web Service?

Answer : The @Path annotation is used to specify relative URI path.

e.g.

@Path("/test")

Question 22. How Can You Embed Variables In The Uri?

Answer : Variables can be embedded within the URI syntax (called URI path template). The variables are substituted at runtime.

@Path("/test/{username}")

Question 23. How Embedded Variables Are Represented In The Uri Syntax?

Answer : Embedded variables are denoted by curly braces.

Question 24. How Can You Obtain The Value Of A Method Parameter In Restful Resource Method?

Answer : @PathParam can be used on method parameter of a resource method to obtain the value for a method parameter.

e.g.

```
@GET
@Produces("text/plain")
public String
getWelcomeMsg(@PathParam("username") String
userName) {
...
}
```

Question 25. How Can You Define A Regular Expression For A Variable Say Username?

Answer : @Path("users/{username: [a-zA-Z][a-zA-Z_0-9]*}")

Question 26. What Happens If Regular Expression Of A Uri Embedded Variable Is Not Matched?

Answer :   A 404 (Not Found) response will occur.

Question 27. Will The Following Direct To The Same Resource? @path("/test/") @path("/test")

Answer :   Yes. By default, both should point to the same resource, although Jersey has arection mechanism that is disabled by default.

Question 28. Explain How To Specify Mime Media Types Of A Resource That Is Sent Back To The Client?

Answer :   @Produces annotation is used for this purpose..

 e.g.

@Produces("text/plain")

Question 29. Is It True That @produces Can Only Be Applied At Method Level?

Answer :   False. It can be applied at both the class and method levels.

Question 30. Can @produces Annotation Be Overridden In A Restless Resource? Give Example?

Answer :   Yes. In the following example, the default MIME type is set as text/plain but doGetAsHtml method overrides it to text/html.

```
@Path("/testResource")
@Produces("text/plain")
public class TestResource {
@GET
public String doGetAsPlainText() {
...
}

@GET
```

```
@Produces("text/html")
public String doGetAsHtml() {
...
}
}
```

Question 31. How Can You Declare More Than One Media Type In The Same @produces Declaration?

Answer : @Produces({"application/plain","application/xml", "application/json"})

Question 32. Assume That There Is A Resource Class Capable Of Producing More Than One Mime Media Type For Get. How Will Be The Response Method Chosen?

Answer : Resource method is chosen depending on the most acceptable media type as declared by the client in the request header.

e.g.

Accept: text/plain.

Question 33. How Will You Specify The Mime Media Types Of Representations A Restful Resource Can Use?

Answer : @Consumes annotation is to be used.

Question 34. Write A Method That Is Triggered On Post Request And Takes Text/plain As Input?

Answer : @POST
```
@Consumes("text/plain")
public void postMessage(String message) {
...
}
```

Question 35. At Which Levels, @consumes Can Be Applied?

Answer : @Consumes can be applied at both the class and method levels.

Question 36. Give An Example Of Declaring More Than One Media Type In @consumes Declaration?

Answer :   @POST
@Consumes({"application/plain","application/xml"})
public void postMessage(String message) {
...
}

Question 37. Can Restful Web Services Support Multiple Types Of Response (mime) Formats?

Answer :   RESTful Web Services, by confirming to HTTP, does support multiple types of response (MIME) formats e.g. XML, JSON, PLAIN etc.