**javax.ws.rs.core.Response**

* Many times we may need to send “Additional Information OR Metadata Information” along with the response like Cookies , Last Modified Data, Language, Caching Info Location URI(For newly created Resource), etc.
* In such cases, resource methods can return instances of “javax.ws.rs.core.Response”
* “javax.ws.rs.core.Response” is an Abstract Class and “javax.w.rs.core.Response.ResponseBuilder” is an inner abstract class of Response.
* Both are part of JAX-RS API and both classes have couple of abstract methods
* The Classes which extends above abstract classes are in Jersey Framework

1. org.glassfish.jersey.message.internal.OutboundJaxrsResponse extends Response
2. org.glassfish.jersey.message.internal.OutboundJaxrsResponse$Builder extends ResponseBuilder

* Hence implementation of abstract methods present in Response & ResponseBuilder classes are provided in Jersey Framework.
* Response object cannot be created directly; instead,

1. First we need to get the object of javax.ws.rs.core.Response.ResponseBuilder” by invoking “ONLY ONE“ of the static methods (depending on out need) of “javax.ws.rs.core.Response” (This Step is Mandatory)
2. ResponseBuilder object has many non-static methods which would help us to set the Metadata info to Response, Hence we Need to invoke “Zero/One/More non-static methods” (depending on our need) on ResponseBuilder (This Step is Optional)
3. Finally we “MUST” invoke build() method to get the Response objects (This Step is Mandatory)

**Psuedo-Code**

**ResponseBuilder builder = Response.one-of-the Static\_Methods();**

**builder.one-or-more-Non\_Static\_Methods();**

**Response resp = builder.build();**

* The ResponseBuilder class is a factory that helps to create one individual Response object.
* For example, below is the code which returns 201 (Created) status code and a Location header whose value is the URI to the newly created resource

@POST

@Consumes(“application/xml”)

public Response createdResource(String content){

//Code to Create the Resource using content

URI createdUri = …

ResponseBuilder builder = Response.created(createUri);

Response resp = builder.build();

return resp;

//The above lines can also be written as below

//return Response.created(createdUri).build();

}