JAX-RS 1.1. (Jersey 1.5)

**REST web services**

Representational State Transfer

REST is neither a standard nor a protocol. It is just an architectural style like say for example client-server architecture (client-server is neither a standard nor a protocol).

Web services following this architectural style are said to be RESTful Web services.

**@GET,@PUt , @POST, @DELETE**

**@PathParam** : Inject the value from URL in to our REST method @PathParam ("name") String name)

**@Consume**: Defines MIME type of the request body.

**@Produce**: Define MIME type of the response Body

Steps to create REST service

1. Rest service WEB module

2. REST service Client

1. It is the web appplication based with 1 Jersey Servlet on it

<servlet>   
 <servlet-name>jersey</servlet-name>  
 <servlet-class>com.sun.jersey.spi.container.servlet.ServletContainer</servlet-class>  
 <load-on-startup>1</load-on-startup>   
</servlet>   
<servlet-mapping>  
 <servlet-name>jersey</servlet-name>  
 <url-pattern>/rest/\*</url-pattern>   
</servlet-mapping>

2. Init context root in **weblogic.xml** of Rest module

<wls:weblogic-version>12.1.3</wls:weblogic-version>  
 <wls:context-root>isisRest</wls:context-root>

3. create abstract class to have a method called getInitialContext() in order to access remote instances EJB API

4. create RestService.java :   
 Class Level :   
 @Path("name1") --> in class level comes after after root-context

Method Level

**@Path**("name2") comes after class level Path

**@Produces**(MediaType.Application\_XML)

**@GET,@PUT,@POST,@DELETE** --> delete does not work with spring

We initialize

context root: in weblogic.xml of Rest module

shared librray: in weblogic-application.xml

Difference between ERST and SOAP :

SOAP is XML based messages. REST can supports more MIME TYPE   
 REST is totally Stateless and SOAP can support State full also  
 REST API can be cached by JAX-RS with Http 1.1 --> cache Control specifies age and behavior

public Response getBook(@PathParam("id") long id){

Book myBook = getBookFromDB(id);  
 CacheControl cc = new CacheControl();  
 cc.setMaxAge(86400);  
 cc.setPrivate(true);  
 ResponseBuilder builder = Response.ok(myBook);  
 builder.cacheControl(cc);  
 return builder.build();

}

REST is Lighter, SOAP requires an XML wrapper around every request and response(SOAP Headers, XML tags etc).  
 REST has no XML message, ideal choice for Mobile   
 REST supports few more MIME type but SOAP supports only XML   
 REST development is much easier than SOAP

Browsers can handle REST easily as compared to SOAP as REST is based on HTTP where SOAP is another wrapper over HTTP.

REST client :   
They are the same we can use Spring template or JErsey. Its common in both to define REST URL, and request Parameters, and request Headers.

REST Security :

There are two basic approaches here. The first is, use HTTP AUTH. adding credential user name and password with base64 endocing to request header and then in Server side using interceptor or filter authenticate them.

The second solution would be having dedicated login service that is called and returns a token. then token goes to URL like a parameter and passed to REST API