**Deployment within a JAX-RS-Unaware Container**

* If Servlet containers are 2.x or older then we have to manually configure web.xml to load your JAX-RS implementation’s proprietary servlet class.
* For Example,

**Jersey**

<servlet>

<servlet-name>mapsRestService</servlet-name>

<servlet-class>

org.glassfish.jersey.servlet.ServletContainer

</servlet-class>

<init-param>

<param-name>

jersey.config.server.provider.packages

</param-name>

<param-value>com.google.googlemaps.rest</param-value>

</init-param>

</servlet>

<servlet-mapping>

<servlet-name>mapsRestService</servlet-name>

<url-pattern>/rest/\*</url-pattern>

</servlet-mapping>

**RESTEasy**

<context-param>

<param-name>resteasy.scan</param-name>

<param-value>true</param-value>

</context-param>

<!-- this need same with resteasy servlet url-pattern -->

<context-param>

<param-name>resteasy.servlet.mapping.prefix</param-name>

<param-value>/rest</param-value>

</context-param>

<listener>

<listener-class>

org.jboss.resteasy.plugins.server.servlet.ResteasyBootstrap

</listener-class>

</listener>

<servlet>

<servlet-name>resteasy-servlet</servlet-name>

<servlet-class>org.jboss.resteasy.plugins.server.servlet.HttpServletDispatcher</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>resteasy-servlet</servlet-name>

<url-pattern>/rest/\*</url-pattern>

</servlet-mapping>

* The <serlvet-mapping> specifies tha base URI path for the JAZ-RS runtime
* The <url-pattern> specifies that all incoming requests should be routed through our JAX-RS implementation. RL pattern should end with \*
* “ServletContainer” in case of Jersey is a yyyConcrete Class which extends HttpServlet. In other words it’s a Servlet, takes all the incoming requests & routes to corresponding resource class.