

## Servicio Rest – Emi Kohmann

1. **File → New → Dynamic Web Project** (PruebaRest)
2. **Configuration → Modify**
3. Activamos **JAX-RS (REST Web Services) → Ok**
4. **Next → Next → Activamos generate web.xml → Next**
5. Acá si tenemos bien configurado nos tiene que salir nuestra librería **JAX-RS 2.23.2**
6. Tenemos que verificar que por defecto tenemos exactamente esto, luego **Finish**.



**Servlet Information**

JAX-RS servlet name:

JAX-RS servlet class name:

URL mapping patterns:  Add... Remove

7. Creamos un paquete **ar.edu.ubp.das.resources**
8. Creamos una clase **EjemploResource.java** en el paquete.

Podemos copiar el siguiente código como ejemplo de una clase simple con una función simple sin parámetros (**se podrían agregar con @FormParam**) que devuelve una cadena. Igual, en Rest, siempre se devuelve un Response:

```
package ar.edu.ubp.das.resources;

import javax.ws.rs.POST; import javax.ws.rs.Path; import javax.ws.rs.Produces; import
javax.ws.rs.core.MediaType; import javax.ws.rs.core.Response;

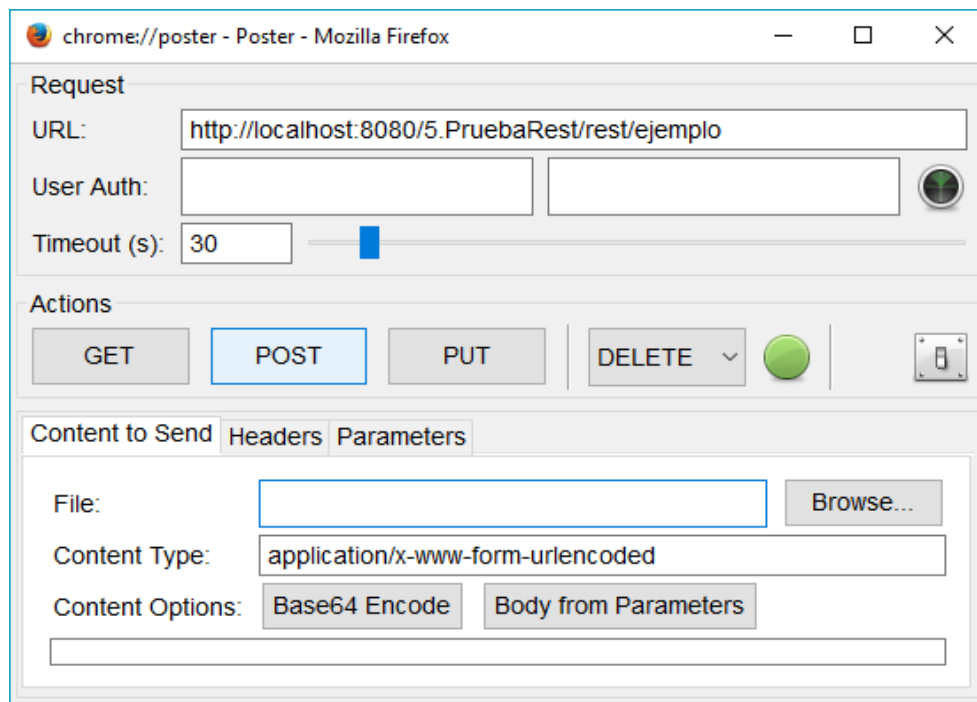
@Path("/ejemplo") // Esto es lo que se agrega al path del REST configurado
@Produces(MediaType.APPLICATION_JSON)
public class EjemploResource {
    @POST
    public Response funcionEjemplo( ) {
        String ret = "hola rest";
        return Response.status( Response.Status.OK ).entity( ret ).build();
    }
}
```

9. Muy importante: Tenemos que agregar el siguiente código en web.xml sí o sí (se puede copiar y pegar).

```
<servlet>
  <description>JAX-RS Tools Generated - Do not modify</description>
  <servlet-name>JAXRSServlet</servlet-name>
  <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
  <init-param>
    <param-name>jersey.config.server.provider.packages</param-name>
    <param-value>ar.edu.ubp.das.resources,com.fasterxml.jackson.jaxrs</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>JAXRSServlet</servlet-name>
  <url-pattern>/rest/*</url-pattern>
</servlet-mapping>
```

Debe quedar antes de que cierre **</web-app>**  
Es decir entre **</welcome-file-list>** y **</web-app>**

10. Si hacemos **click derecho >> Run as >> Run on server**, veremos que nos da 404. Porque esta url no es visible, tenemos que usar poster en este caso para consultar el **Rest**. Tenemos que añadir el **body from parameters**.



11. Luego, hacemos POST y en una ventana visualizamos la respuesta del Rest funcionando.

