

Rest cliente – Emi Kohmann

1. **File >> New >> Dynamic Web Project >> PruebaRestCliente >> Finish.**
2. El Rest Cliente no necesita tener el facet de JAX-RS. Es decir, creamos un proyecto normal sin incluir nada extra.
3. Creamos un paquete **ar.edu.ubp.das.servlets** y dentro un Servlet **EjemploServlet**
4. Le agregamos la notation **index.jsp** porque usamos **DWM 3.1**
5. Consumimos directamente, la forma más fácil es la siguiente:

```
URI uri = URI.create("http://localhost:8080/5.PruebaRest/rest/ejemplo");

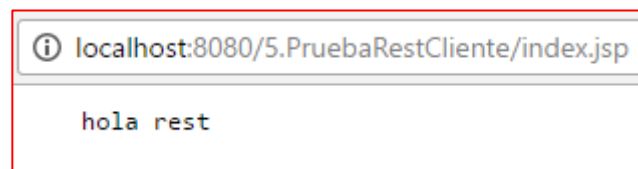
HttpPost req = new HttpPost(); req.setURI(uri);
HttpClient client = HttpClientBuilder.create().build();

HttpResponse resp = client.execute(req);

HttpEntity responseEntity = resp.getEntity();
String restResp= EntityUtils.toString(responseEntity);

response.getWriter().println( restResp );
```

6. Esto directamente nos consume el Rest en el mismo momento, si hacemos Run >> Run on Server, podremos ver el resultado.



7. Hay que recordar que existen métodos para agregar parámetros, controlar errores, etc. Por ejemplo, para agregar parámetros:

```
List <NameValuePair> nvps= new ArrayList <NameValuePair>();
nvps.add(new BasicNameValuePair("nro_categoria", request.getParameter("nroCat")));
nvps.add(new BasicNameValuePair("string_busqueda", request.getParameter("strBusq")));
```

Y luego en el objeto HttpPost req invocamos a:

```
req.setEntity(new UrlEncodedFormEntity(nvps));
```

Para controlar errores:

```
StatusLine responseStatus = resp.getStatusLine();

if (responseStatus.getStatusCode() != 200)
{
    throw new RuntimeException(restResp);
}
```

8. Es decir, se hace todo en el mismo Servlet de manera mucho más directa, ya existen clases preparadas para cada tarea. En este punto se puede usar la librería **GSON** para generar objetos a partir del String que queda guardado en **restResp**. Con una sola función mapeamos el String a un conjunto de objetos que debemos crear previamente, utilizando por ejemplo algún **Bean**.

Ejemplo:

```
Gson gson = new Gson();
```

```
LinkedList< VideoBean > videos = gson.fromJson( restResp, new
    TypeToken<LinkedList<VideoBean>>(){}.getType() );
```