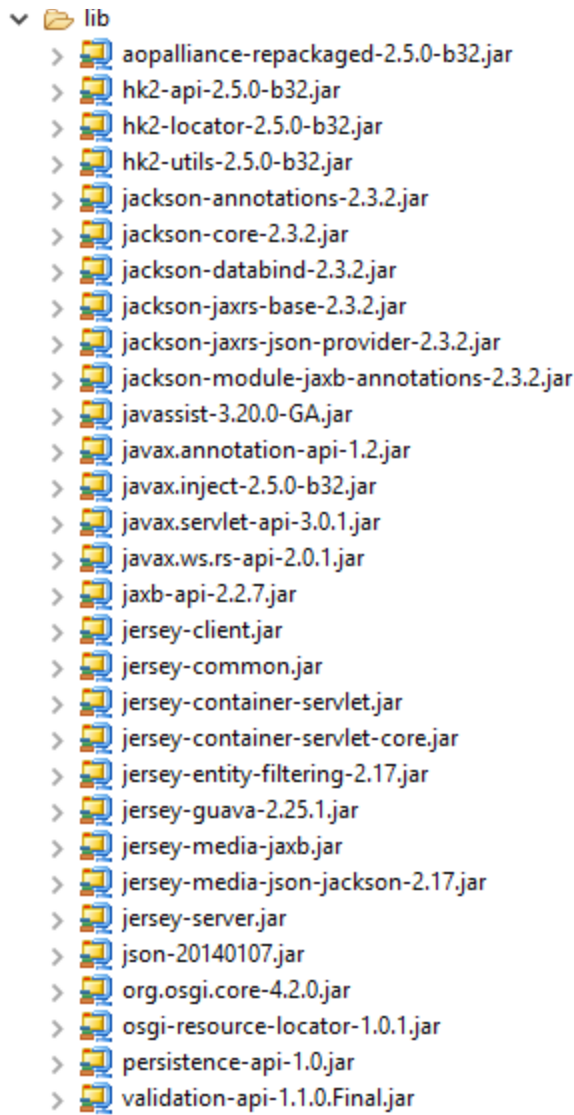


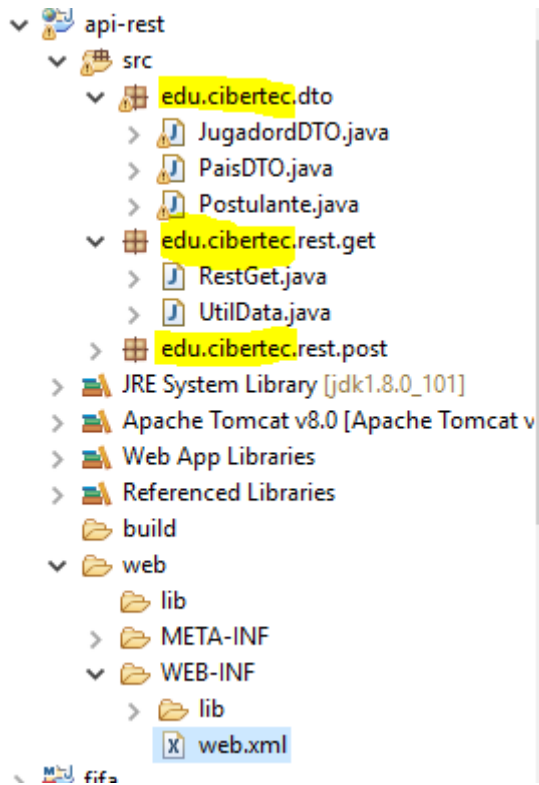
GUIA DE LABORATORIO – FIFA API REST

SERVICIOS REST- Tipo GET– Parte I

Para trabajar servicios REST trabajaremos con JAX-RS con Jersey.
Creamos un proyecto Dynamic Web Project. Y agregamos las librerías necesarias.



Ahora procedemos a crear nuestros servicios REST. En este caso le he puesto de nombre al proyecto api-rest con la siguiente estructura de paquetes



Desarrollamos la clase donde se encuentran los Servicios REST

```
import edu.cibertec.dto.JugadordDTO;
import edu.cibertec.dto.PaisDTO;

@Path("/get")
public class RestGet {

    //http://localhost:puerto/proyecto/path_clase/path_servicio
    //http://localhost:8080/api-rest/get

    @GET
    @Produces(MediaType.TEXT_PLAIN)
    public String holaMundo() {
        System.out.println("inicio servicio holaMundo()");
        return "Hola Mundo";
    }

    //http://localhost:8080/api-rest/get/saludo/Pedro/29

    @GET
    @Path("/saludo/{p_nombre}/{p_edad}")
    @Produces(MediaType.TEXT_PLAIN + ";charset=utf-8")
    public String saludo(@PathParam("p_nombre") String nombre, @PathParam("p_edad") int edad) {
        System.out.println("inicio servicio saludo()");
        return "Hola "+nombre+", tienes "+edad+" años.";
    }
}
```

```
//http://localhost:8080/api-rest/get/obtenerCapitan/PERU
```

```
@GET
@Path("/obtenerCapitan/{p_pais}")
@Produces(MediaType.TEXT_PLAIN + ";charset=utf-8")
public String obtenerCapitan(@PathParam("p_pais") String nombrePais) {
    System.out.println("inicio servicio obtenerEntrenador()");
    UtilData datos = new UtilData();
    datos.cargarData();

    PaisDTO pais = datos.getPais(nombrePais);

    if(pais!=null) {
        return pais.getCapitan();
    }else {
        return "PAIS NO EXISTE EN NUESTRA BASE DE DATOS";
    }
}
```

```
//http://localhost:8080/api-rest/get/obtenerConvocadosxml/PERU
@GET
@Path("/obtenerConvocadosxml/{p_pais}")
@Produces(MediaType.APPLICATION_XML)
public List<JugadorDTO> obtenerListaConvocados(@PathParam("p_pais") String nombrePais) {
    System.out.println("inicio servicio obtenerListaConvocados()");

    UtilData datos = new UtilData();
    datos.cargarData();
    PaisDTO pais = datos.getPais(nombrePais);

    if(pais!=null) {
        return pais.getLstJugadores();
    }else {
        return new ArrayList<JugadorDTO>();
    }
}
```

```
//http://localhost:8080/api-rest/get/obtenerConvocadosjson/PERU
@GET
@Path("/obtenerConvocadosjson/{p_pais}")
@Produces(MediaType.APPLICATION_JSON)
public List<JugadorDTO> obtenerListaConvocadosJson(@PathParam("p_pais") String nombrePais) {
    System.out.println("inicio servicio obtenerListaConvocadosJson()");

    UtilData datos = new UtilData();
    datos.cargarData();
    PaisDTO pais = datos.getPais(nombrePais);

    if(pais!=null) {
        return pais.getLstJugadores();
    }else {
        return new ArrayList<JugadorDTO>();
    }
}
```

Configuramos el web.xml que va escanear los paquetes del API.

```
<servlet>
  <servlet-name>Jersey Servicio REST Basico</servlet-name>
  <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
  <init-param>
    <param-name>jersey.config.server.provider.packages</param-name>
    <param-value>edu.cibertec</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>Jersey Servicio REST Basico</servlet-name>
  <url-pattern>*</url-pattern>
</servlet-mapping>
```

Al ser métodos GET podemos probarlo levantando el API en el TOMCAT y apuntando a los servicios desde el BROWSER.