

TD Développement de Web Services (REST)

Création d'un Web Service REST avec jBoss sous Eclipse

Nous allons à présent créer un Web Service REST que nous interrogerons avec un navigateur.

Créer un nouveau projet Web Dynamique. Choisir module dynamique version 2.5 Choisir une configuration personnalisée à laquelle vous ajouterez le Facet JAX-RS (REST Web Service).

Rajouter les jar suivantes sous WEB-INF/lib

- javassist-3.8.0.GA.jar
- jaxrs-api-3.0-5-Final.jar
- resteasy-jaxrs-3.0-5-Final.jar
- scannotation-1.0.3.jar

Remplacer le web.xml par ce qui suit :

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID" version="2.5">
<display-name>RestEasy</display-name>

<listener>
<listener-class>
org.jboss.resteasy.plugins.server.servlet.ResteasyBootstrap
</listener-class>
</listener>
<servlet>
<servlet-name>Resteasy</servlet-name>
<servlet-
class>org.jboss.resteasy.plugins.server.servlet.HttpServletDispatcher</servlet-
class>
</servlet>
<servlet-mapping>
<servlet-name>Resteasy</servlet-name>
<url-pattern>/*</url-pattern>
</servlet-mapping>

<context-param>
<param-name>resteasy.scan</param-name>
<param-value>true</param-value>
</context-param>
</web-app>
```

Considérons ce code d'une classe annotée pour devenir un web service REST :

```
package hds.utc;

import javax.ws.rs.GET;
```

```

import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;

@Path("ConversionService")
public class TestRest {
    @GET
    @Path("/InchToFeet/{i}")
    @Produces(MediaType.TEXT_XML)
    public String convertInchToFeet(@PathParam("i") int i) {

        int inch=i;
        double feet = 0;
        feet =(double) inch/12;

        return "<InchToFeetService>"
            + "<Inch>" + inch + "</Inch>"
            + "<Feet>" + feet + "</Feet>"
            + "</InchToFeetService>";
    }

    @Path("/FeetToInch/{f}")
    @GET
    @Produces(MediaType.TEXT_XML)
    public String convertFeetToInch(@PathParam("f") int f) {
        int inch=0;
        int feet = f;
        inch = 12*feet;

        return "<FeetToInchService>"
            + "<Feet>" + feet + "</Feet>"
            + "<Inch>" + inch + "</Inch>"
            + "</FeetToInchService>";
    }
}

```

Ce service peut être invoqué avec une URL à partir d'un navigateur comme suit :

<http://localhost:8080/RestEasy/ConversionService/FeetToInch/5>

Les plus importantes annotations JAX-RS sont listées dans le tableau suivant:

Annotation	Description
@PATH(your_path)	Sets the path to base URL + /your_path. The base URL is based on your application name, the servlet and the URL pattern from the web.xml" configuration file.

Annotation	Description
@POST	Indicates that the following method will answer to a HTTP POST request
@GET	Indicates that the following method will answer to a HTTP GET request
@PUT	Indicates that the following method will answer to a HTTP PUT request
@DELETE	Indicates that the following method will answer to a HTTP DELETE request
@Produces(MediaType.TEXT_PLAIN [, more-types])	@Produces defines which MIME type is delivered by a method annotated with @GET. In the example text ("text/plain") is produced. Other examples would be "application/xml" or "application/json".
@Consumes(type [, more-types])	@Consumes defines which MIME type is consumed by this method.
@PathParam	Used to inject values from the URL into a method parameter. This way you inject for example the ID of a resource into the method to get the correct object.

En vous inspirant de ce code, annotez le code de la classe `Calcullette.java` développée dans les exercices précédents en vue de la transformer en un Web Service REST

Déployez votre service REST dans Tomcat et testez le en l'interrogeant à partir d'un navigateur.