« RESTFUL service 2 »

09/04/2018

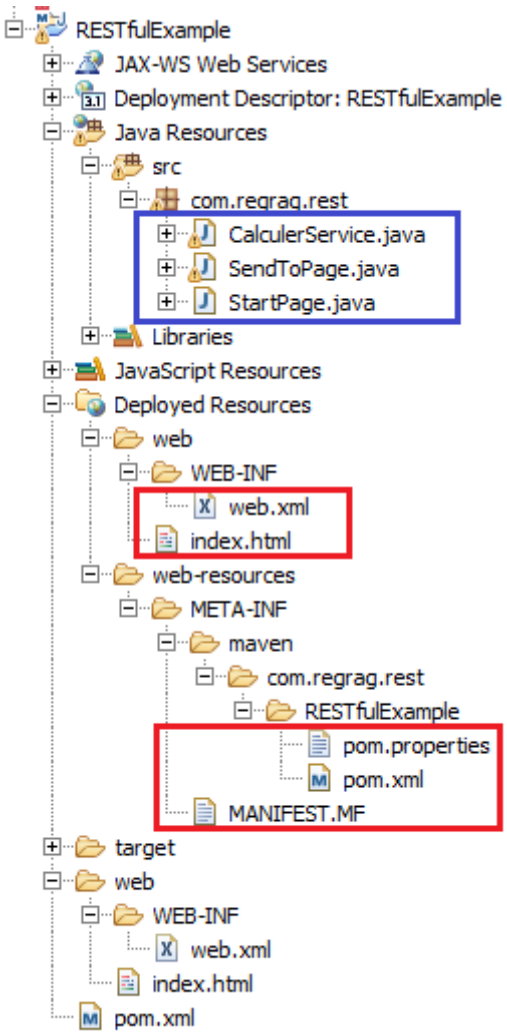
By : REGRAG mouhcine

Table des matières

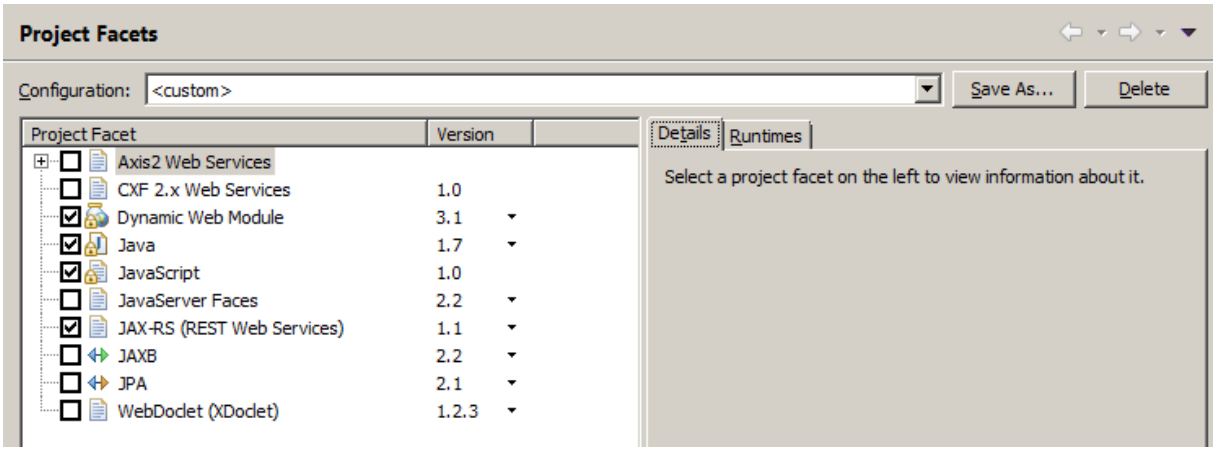
1- Arborescence du projet 3
2- Dépendances en API 4
3- Exécution de l'exemple 5
4- Travail à faire 6
5- Fichiers sources 7

1- Arborescence du projet :

La structure de base d'un projet RESTFUL est comme suit :

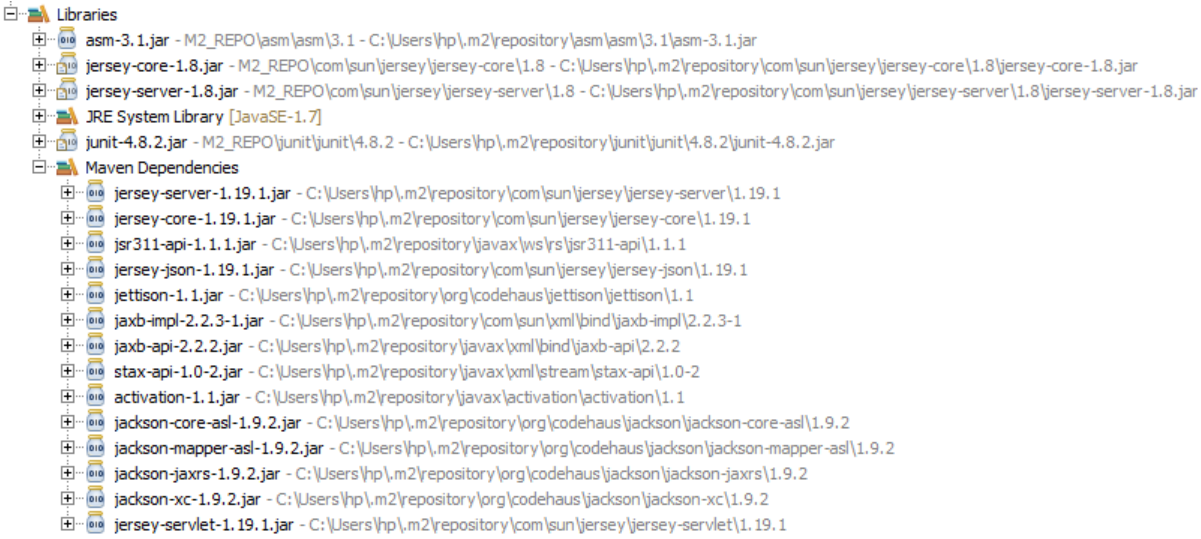


Activez dans les propriétés du projet → projet facets : JAX-RS (REST Web Services) comme indiqué en bas :



2- Dépendances en API :

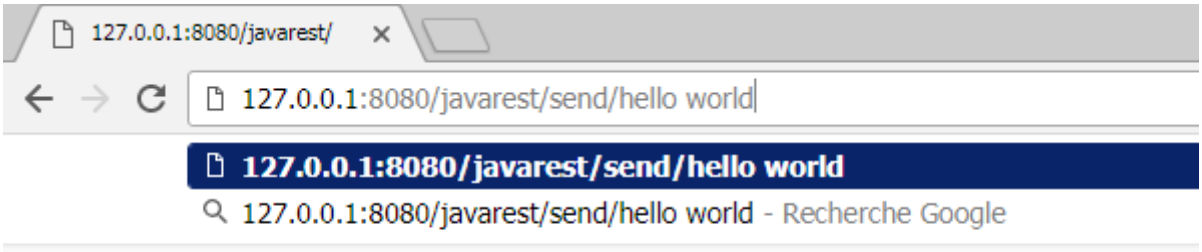
Il faut bien sur ajouter l'ensemble des API comme indiqué en bas :



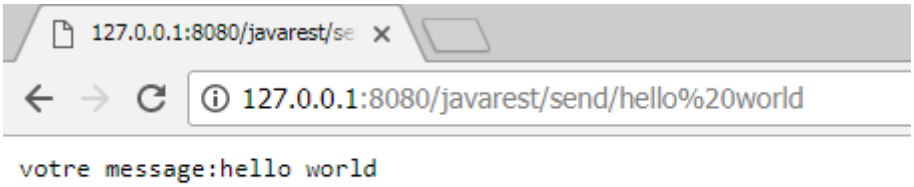
3- Exécution de l'exemple :

L'exemple consiste à réaliser un web service pour pouvoir l'appeler à partir du browser, en lui passant des paramètres.

Appeler le service send en lui passant un message comme suit :

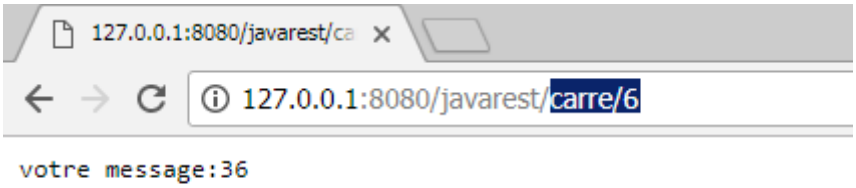


Le résultat est donné par la figure en bas :



4- Travail à faire :

Ajouter le service carre qui permet de calculer pour une entier X la valeur de X*X et afficher le résultat dans le browser comme indiqué en bas.



Pour cela, ajouter la classe CalculerService.java avec le code suivant :

```
package com.regrag.rest;

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.Response;

import javax.ws.rs.core.MediaType;
@Path("/carre")
public class CalculerService {
    @GET
    @Produces(MediaType.TEXT_HTML)
    public String getMessage() {
        return String.format("rien!");
    }
    @Path("{x}")
    @GET
    @Produces(MediaType.TEXT_PLAIN)
    public String afficherinput(@PathParam("x") int x) {
       int y= x * x;
        return String.format("votre message:" + y);

    }

}
```

5- Fichiers sources :

1- Web.xml :

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" version="3.1">
 <servlet>
    <servlet-name>Example API</servlet-name>
    <servlet-
class>com.sun.jersey.spi.container.servlet.ServletContainer</servlet-class>
    <init-param>
      <param-name>com.sun.jersey.config.property.packages</param-name>
      <param-value>com.regrag.rest</param-value>
    </init-param>
    <init-param>
      <param-name>com.sun.jersey.api.json.POJOMappingFeature</param-name>
      <param-value>true</param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name>Example API</servlet-name>
    <url-pattern>/*</url-pattern>
  </servlet-mapping>
  <servlet>
    <description>JAX-RS Tools Generated - Do not modify</description>
    <servlet-name>JAX-RS Servlet</servlet-name>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>JAX-RS Servlet</servlet-name>
    <url-pattern>/jaxrs/*</url-pattern>
  </servlet-mapping>
</web-app>
```

2- StartPage.java :

```
package com.regrag.rest;

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

@Path("/")
public class StartPage {
       @GET
          @Produces(MediaType.TEXT_PLAIN)
          public String getMessage() {
              return String.format("this is the simple page for RestFull!:in URL
add : send/msg or carre/nbr");

          }

    }
```

3- SendToPage.java :

```
package com.regrag.rest;

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

@Path("/send")
public class SendToPage {

    @GET
    @Produces(MediaType.TEXT_PLAIN)
    public String getMessage() {
        return String.format("message vide!");
    }

    @Path("{msg}")
    @GET
    @Produces(MediaType.TEXT_PLAIN)
    public String afficherinput(@PathParam("msg") String msg) {
        return String.format("votre message:" + msg);

    }


    }
```

4- CalculerService.java :

```java
package com.regrag.rest;

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.Response;

import javax.ws.rs.core.MediaType;
@Path("/carre")
public class CalculerService {
    @GET
    @Produces(MediaType.TEXT_HTML)
    public String getMessage() {
        return String.format("rien!");
    }
    @Path("{x}")
    @GET
    @Produces(MediaType.TEXT_PLAIN)
    public String afficherinput(@PathParam("x") int x) {
      int y= x * x;
        return String.format("votre message:" + y);

    }

}
```

5- Pom.properties :

```
#Generated by Maven Integration for Eclipse
#Mon Apr 09 21:10:25 WEST 2018
version=1.0-SNAPSHOT
groupId=com.regrag.rest
m2e.projectName=RESTfulExample
m2e.projectLocation=C\:\\FormationJavaFramworks\\workspace\\wsJavaRestDemos\\RESTf
ulExample
    artifactId=RESTfulExample
```

6- POM.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
      <modelVersion>4.0.0</modelVersion>
      <groupId>com.regrag.rest</groupId>
      <artifactId>RESTfulExample</artifactId>
      <packaging>war</packaging>
      <version>1.0-SNAPSHOT</version>
      <name>RESTfulExample Maven Webapp</name>
            <dependencies>
            <dependency>
                  <groupId>com.sun.jersey</groupId>
                  <artifactId>jersey-server</artifactId>
                  <version>1.19.1</version>
            </dependency>
            <dependency>
                  <groupId>com.sun.jersey</groupId>
                  <artifactId>jersey-core</artifactId>
                  <version>1.19.1</version>
            </dependency>
            <dependency>
                  <groupId>com.sun.jersey</groupId>
                  <artifactId>jersey-json</artifactId>
                  <version>1.19.1</version>
            </dependency>
            <dependency>
                  <groupId>com.sun.jersey</groupId>
                  <artifactId>jersey-servlet</artifactId>
                  <version>1.19.1</version>
            </dependency>
      </dependencies>
      <build>
            <finalName>javarest</finalName>
            <sourceDirectory>src</sourceDirectory>
            <plugins>
                  <plugin>
                        <artifactId>maven-compiler-plugin</artifactId>
                        <version>3.1</version>
                        <configuration>
                              <source>1.7</source>
                              <target>1.7</target>
                        </configuration>
                  </plugin>
                  <plugin>
                        <artifactId>maven-war-plugin</artifactId>
                        <version>2.4</version>
                        <configuration>
                              <warSourceDirectory>web</warSourceDirectory>
                              <failOnMissingWebXml>false</failOnMissingWebXml>
                        </configuration>
                  </plugin>
            </plugins>
      </build>

   </project>
```