

« RESTFUL service »

04/04/2018

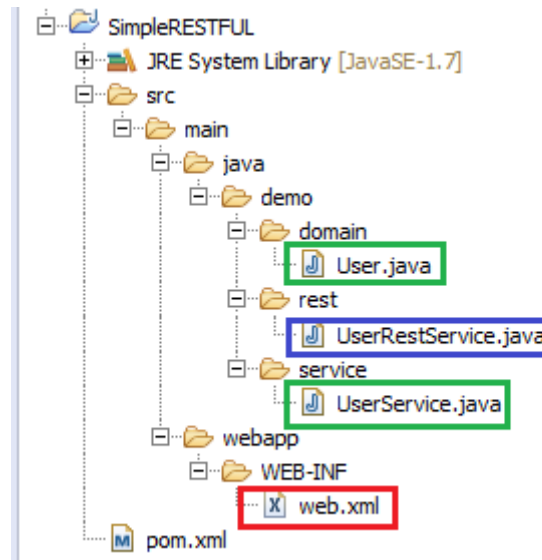
By : REGRAG mouhcine

Table des matières

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

1- Arborescence du projet :

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



Elle contient :

- Un fichier web.xml pour la configuration du service.
- Le Fichier User.java présente le modèle de donnée.
- Le fichier UserService.java pour manipuler vos données.
- Le fichier UserRestService.java qui est la classe qui implémente le service web produit.

2- Dépendances en API :

Plusieurs API sont nécessaires pour faire fonctionner un service web (voir en bas).



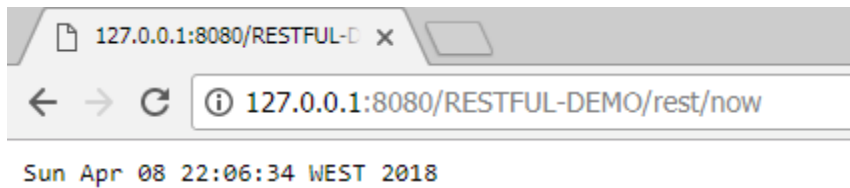
3- Exécution de l'exemple :

L'exemple consiste à pouvoir interroger le service web RESTFUL-DEMO/rest/users, qui permet d'afficher le nom concaténé avec le prénom d'un utilisateur et afficher le résultat sur une page comme suit :



4- Travail à faire :

On veut réaliser un service web appelé TimeRestService, qui permet d'afficher l'heure et la date courant sur une page comme suit.



Procéder comme suit :

a- Créer la classe TimeRestService.java avec le code suivant :

```
package demo.rest;

import java.util.Date;

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

@Path("/now")
public class TimeRestService {

    @GET
    @Produces(MediaType.APPLICATION_JSON)
    public String getDefaultUserInJSON() {
        return new Date().toString();
    }
}
```

5- Fichiers sources :

a- User.java :

```
package demo.domain;

import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement
// from http://www.vogella.com/articles/REST/
// JAX-RS supports an automatic mapping from JAXB annotated class to XML and
JSON
public class User {

    private String firstName;
    private String lastName;

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
}
```

b- UserService.java :

```
package demo.service;

import java.util.Date;

import demo.domain.User;

public class UserService {

    public User getDefaultUser() {

        User user = new User();
        user.setFirstName("REGRAG");
        user.setLastName("mouhcine");
        return user;
    }
}
```

c- UserRestService.java :

```
package demo.rest;

import demo.domain.User;
import demo.service.UserService;

import java.util.Date;

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

@Path("/users")
public class UserRestService {

    @GET
    @Produces(MediaType.APPLICATION_JSON)
    public User getDefaultUserInJSON() {
        UserService userService = new UserService();
        return userService.getDefaultUser();
    }
}
```


d- web.xml :

```
<!DOCTYPE web-app PUBLIC
    "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd" >

<web-app id="WebApp_ID" version="2.4"
    xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
    <display-name>Restful Web Application</display-name>

    <servlet>
        <servlet-name>jersey-serlvet</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>demo.rest</param-value>
        </init-param>

        <init-param>
            <param-name>com.sun.jersey.api.json.POJOMappingFeature</param-
name>
            <param-value>>true</param-value>
        </init-param>

        <load-on-startup>1</load-on-startup>
    </servlet>

    <servlet-mapping>
        <servlet-name>jersey-serlvet</servlet-name>
        <url-pattern>/rest/*</url-pattern>
    </servlet-mapping>

</web-app>
```

e- 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>demo</groupId>
  <artifactId>demo</artifactId>
  <packaging>war</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>ngdemo Maven Webapp</name>
  <url>http://maven.apache.org</url>

  <dependencies>

    <!-- javax: XML binding -->
    <dependency>
      <groupId>javax.xml</groupId>
      <artifactId>jaxb-api</artifactId>
      <version>2.1</version>
    </dependency>

    <!-- RESTful web service: Jersey -->
    <dependency>
      <groupId>com.sun.jersey</groupId>
      <artifactId>jersey-server</artifactId>
      <version>1.17.1</version>
    </dependency>
    <dependency>
      <groupId>com.sun.jersey</groupId>
      <artifactId>jersey-servlet</artifactId>
      <version>1.17.1</version>
    </dependency>
    <dependency>
      <groupId>com.sun.jersey</groupId>
      <artifactId>jersey-json</artifactId>
      <version>1.17.1</version>
    </dependency>

    <!-- Unit testing -->
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>

  </dependencies>
  <build>
    <finalName>RESTFUL-DEMO</finalName>
  </build>
</project>
```