

## Atelier REST n°2

# Consommation d'un service web RESTFul avec JAX-RS 2.0

### Objectifs

Le but de cet atelier est le développement d'un client JAX-RS consommateur d'un service web RESTFul existant.

### Création d'un client du service RESTFul

JAX-RS 2.0 introduit une API client standardisée grâce à laquelle vous pouvez effectuer des demandes HTTP vers vos services Web RESTFul.

1. Créez un nouveau projet **maven** de packaging **war** que vous le nommez « HelloRESTClient ».
2. Créez une classe nommée « HelloResourceClient » encapsulant la méthode « main ».
3. Ajoutez le code suivant permettant la consommation de la première ressource créée dans l'atelier1 « Hello JAX-RS ».

```
public static void main(String[] args) {  
    // Create new JAX-RS client  
    Client client = ClientBuilder.newClient();  
    //The base URL of the service  
    WebTarget target = client.target("http://localhost:8383/Hello_JAX-RS-0.0.1-SNAPSHOT/rest/greetings");  
    //consuming sayHello method  
    // Get the response from the target URL:  
    Response response = target.request().get();  
    // Read the result as a String:  
    String result = response.readEntity(String.class);  
    // Print the result to the standard output:  
    System.out.println(result);  
    response.close();  
}
```

4. Ajoutez le code suivant dans votre fichier pom.xml:

```
<properties>
  <failOnMissingWebXml>false</failOnMissingWebXml>
  <project.buil.sourceEncoding>utf-8</project.buil.sourceEncoding>
  <maven.compiler.target>1.8</maven.compiler.target>
  <maven.compiler.source>1.8</maven.compiler.source>
</properties>
<dependencies>
  <dependency>
    <groupId>org.jboss.resteasy</groupId>
    <artifactId>resteasy-jaxrs</artifactId>
    <version>3.0.19.Final</version>
    <scope>provided</scope>
  </dependency>
  <dependency>
    <groupId>org.jboss.resteasy</groupId>
    <artifactId>resteasy-client</artifactId>
    <version>3.0.19.Final</version>
  </dependency>
</dependencies>
```

5. Exécutez la classe java pour tester la consommation du service web RESTful.

## Consommation d'un service web RESTful paramétré

- Ajoutez le code suivant dans la méthode main pour consommer la deuxième ressource REST créée dans l'atelier1 «Hello JAX-RS» et identifiée par l'URI:

*http://localhost:8383/Hello\_JAX-RS-0.0.1-SNAPSHOT/rest/greetings/foulen/benfoulen*

```
WebTarget helloTo1 =target.path("foulen").path("benfoulen");
Response response1 = helloTo1.request().get();
String result1 = response1.readEntity(String.class);
System.out.println(result1);
response1.close();
```

- Pour consommer la troisième ressource REST accessible via l'URI:

*http://localhost:8383/Hello\_JAX-RS-0.0.1-*

*SNAPSHOT/rest/greetings?FirstName=foulen&LastName=benfoulen* ,ajoutez le code suivant :

```
WebTarget helloTo2 =target.queryParam("FirstName", "foulen").queryParam("LastName", "benfoulen");
Response response2 = helloTo2.request().get();
String result2 = response2.readEntity(String.class);
System.out.println(result2);
response2.close();
```

- Après avoir terminé la consommation des ressources REST, il faut fermer la connexion établie par le client :

```
client.close();
```