esprit Se former autrement	Année Universitaire : 2021-2022 **Ateliers Services web** **UP_WEB**

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:8081/Hello_JAX-RS/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 of 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.glassfish.jersey.core</groupId>
		<artifactId>jersey-client</artifactId>
		<version>2.25.1</version>
	</dependency>
  </dependencies>
```

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

5. Exécutez la classe java, **Run As-> Java Application**, 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:8081/Hello_JAX-RS/rest/greetings/**foulen/benfoulen***

- Pour consommer la troisième ressource REST accessible via l'URI: *http://localhost:8081/Hello_JAX-RS/rest/greetings**?FirstName=foulen&LastName=benfoule***

  Ajoutez le code suivant :

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

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

```
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();
```