

# JAVA EE COURSE

# SMS REST WEB SERVICE CLIENT WITH JAX-RS



By the expert: Eng. Ubaldo Acosta

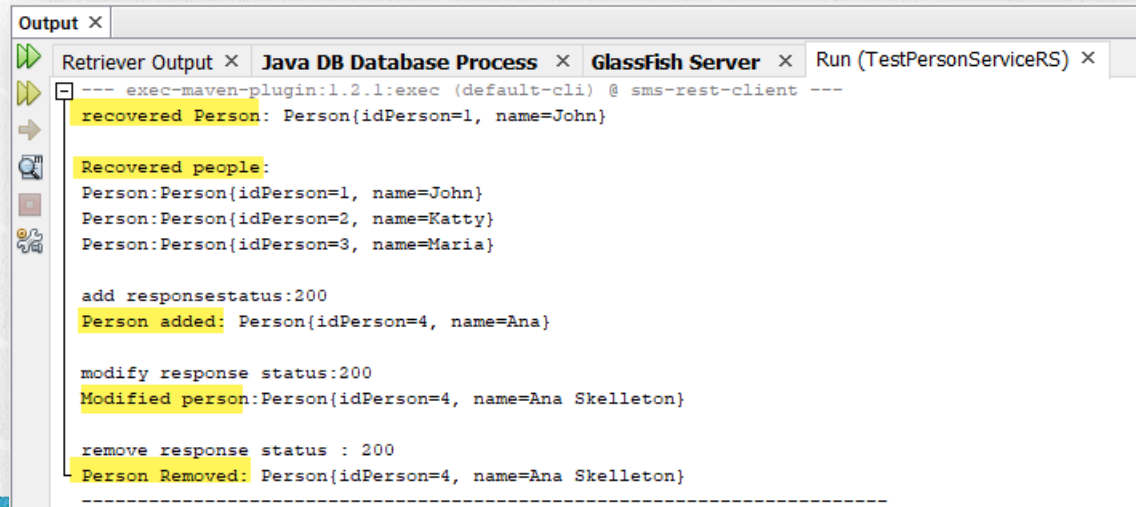


## JAVA EE COURSE

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# EXERCISE OBJECTIVE

The objective of the exercise is to create a client of the sms-jee-web project for the REST Web Service service from the previous exercise. The result is shown below:

A screenshot of an IDE's output window. The window has a title bar with 'Output' and a close button. Below the title bar are several tabs: 'Retriever Output', 'Java DB Database Process', 'GlassFish Server', and 'Run (TestPersonServiceRS)'. The 'Run (TestPersonServiceRS)' tab is active. The output text is as follows:

```
--- exec-maven-plugin:1.2.1:exec (default-cli) @ sms-rest-client ---
recovered Person: Person{idPerson=1, name=John}

Recovered people:
Person:Person{idPerson=1, name=John}
Person:Person{idPerson=2, name=Katty}
Person:Person{idPerson=3, name=Maria}

add responsestatus:200
Person added: Person{idPerson=4, name=Ana}

modify response status:200
Modified person:Person{idPerson=4, name=Ana Skeleton}

remove response status : 200
Person Removed: Person{idPerson=4, name=Ana Skeleton}
-----
```

**JAVA EE COURSE**

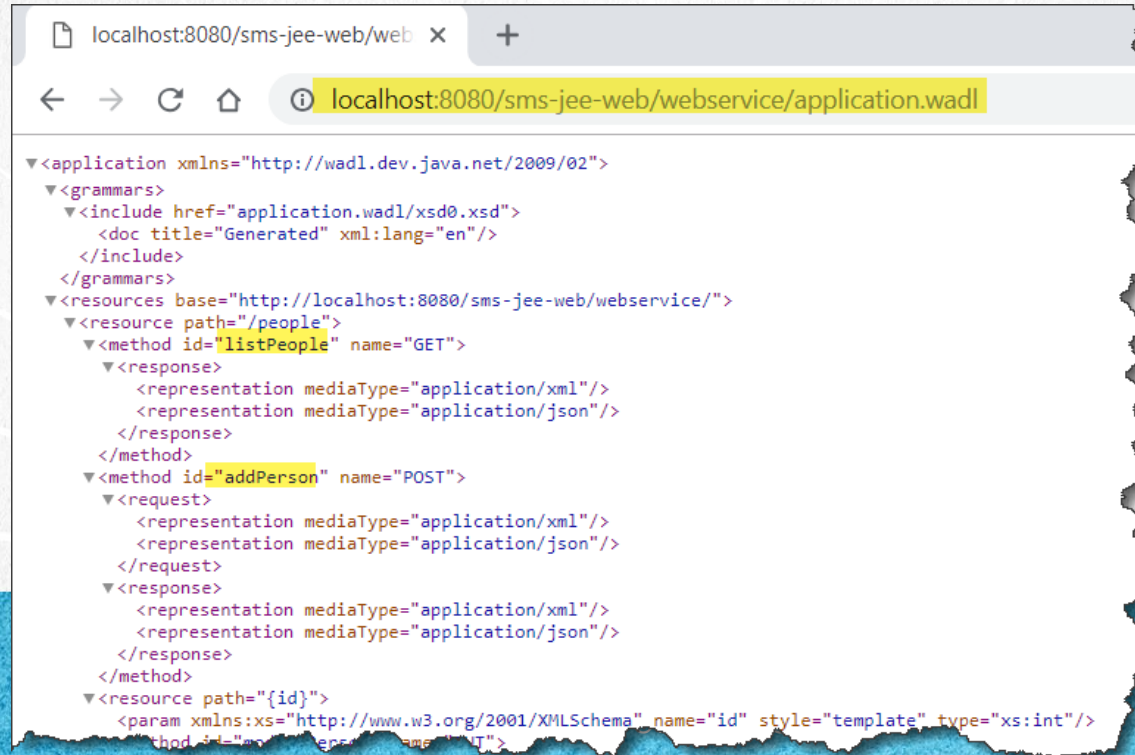
[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)



# STARTED FROM THE PREVIOUS EXERCISE

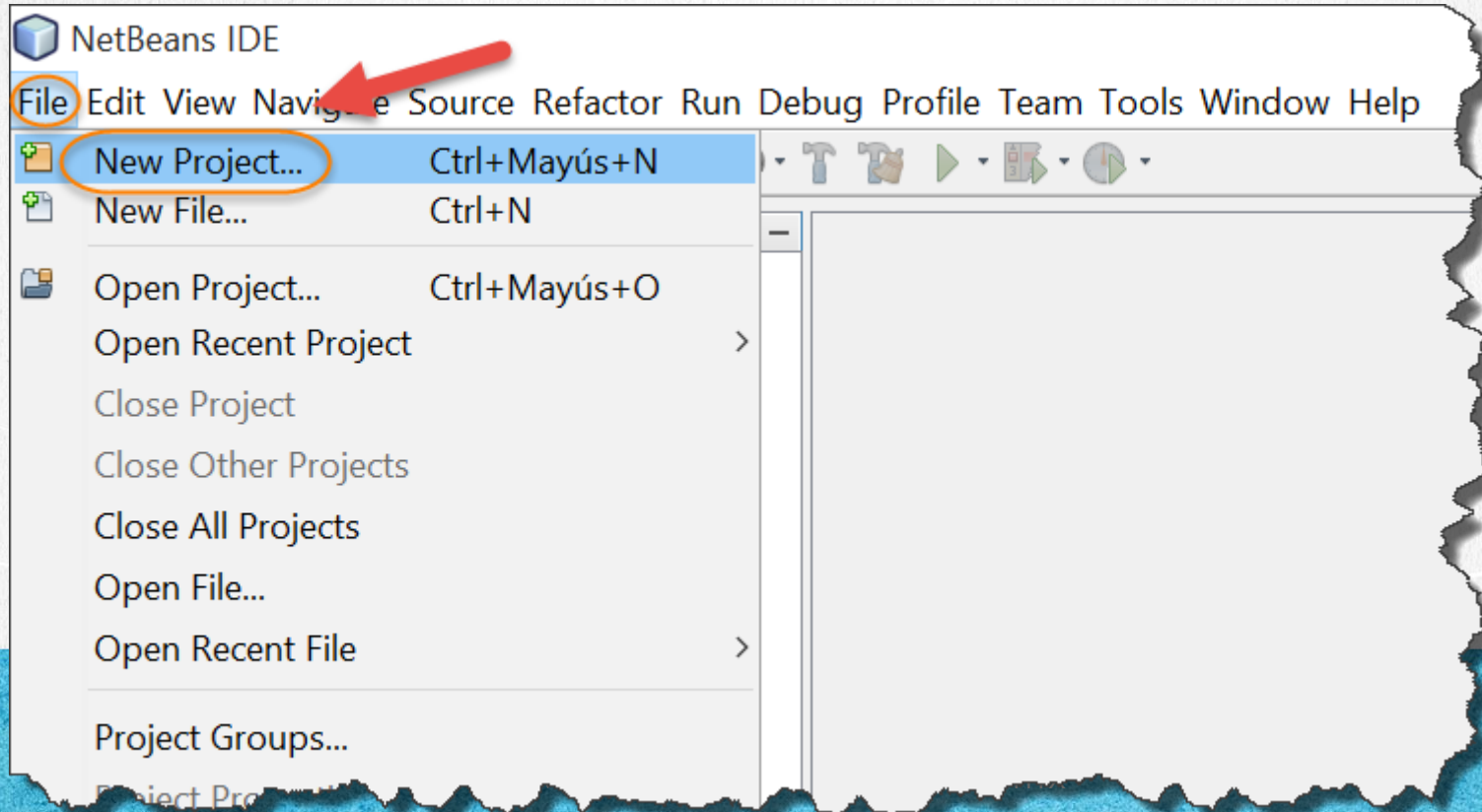
We will start from the previous year. It should already be deployed in Glassfish and working. We must be able to visualize the following url:

<http://localhost:8080/sms-jee-web/webservice/application.wadl>



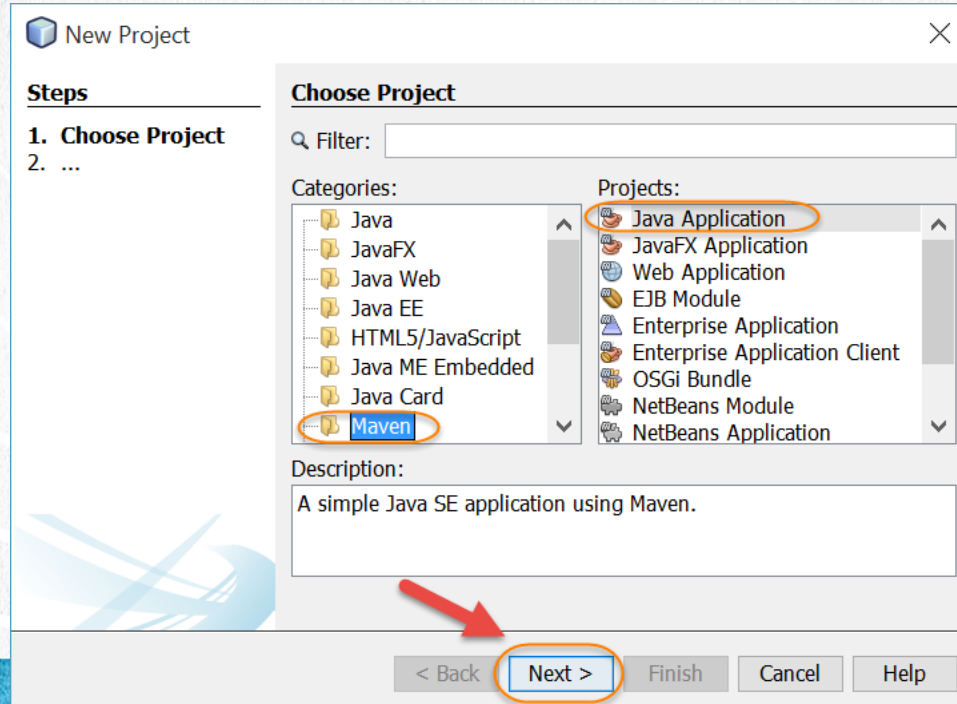
# 1. CREATE A PROJECT

We create the sms-rest-client project:



# 1. CREATE A PROJECT

We create the sms-rest-client project as a maven project:



**JAVA EE COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)



# 1. CREATE A PROJECT

We create the sms-rest-client project as a maven project:

**New Java Application**

**Steps**

1. Choose Project
2. **Name and Location**

**Name and Location**

Project Name: sms-rest-client

Project Location: C:\Courses\JavaEE\Lesson07 Browse...

Project Folder: C:\Courses\JavaEE\Lesson07\sms-rest-client

Artifact Id: sms-rest-client

Group Id: test

Version: 1

Package: (Optional)

< Back Next > **Finish** Cancel Help

## 2. MODIFY THE FILE

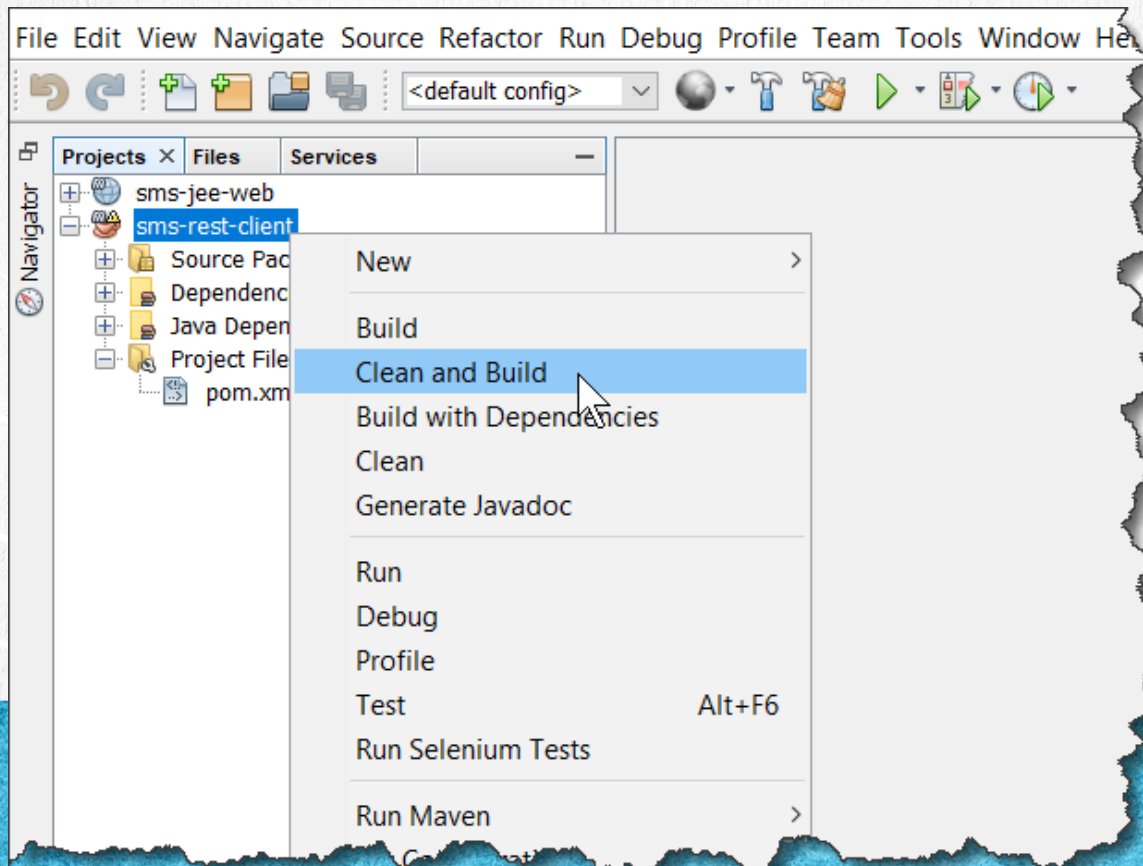
[pom.xml:](#)

Click to download

```
<?xml version="1.0" encoding="UTF-8"?>
<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/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>test</groupId>
  <artifactId>sms-rest-client</artifactId>
  <version>1</version>
  <packaging>jar</packaging>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.glassfish.jersey.core</groupId>
      <artifactId>jersey-client</artifactId>
      <version>2.27</version>
    </dependency>
    <dependency>
      <groupId>org.glassfish.jersey.media</groupId>
      <artifactId>jersey-media-jaxb</artifactId>
      <version>2.27</version>
    </dependency>
    <dependency>
      <groupId>org.glassfish.jersey.inject</groupId>
      <artifactId>jersey-hk2</artifactId>
      <version>2.27</version>
    </dependency>
  </dependencies>
</project>
```

## 2. EXECUTE CLEAN & BUILD

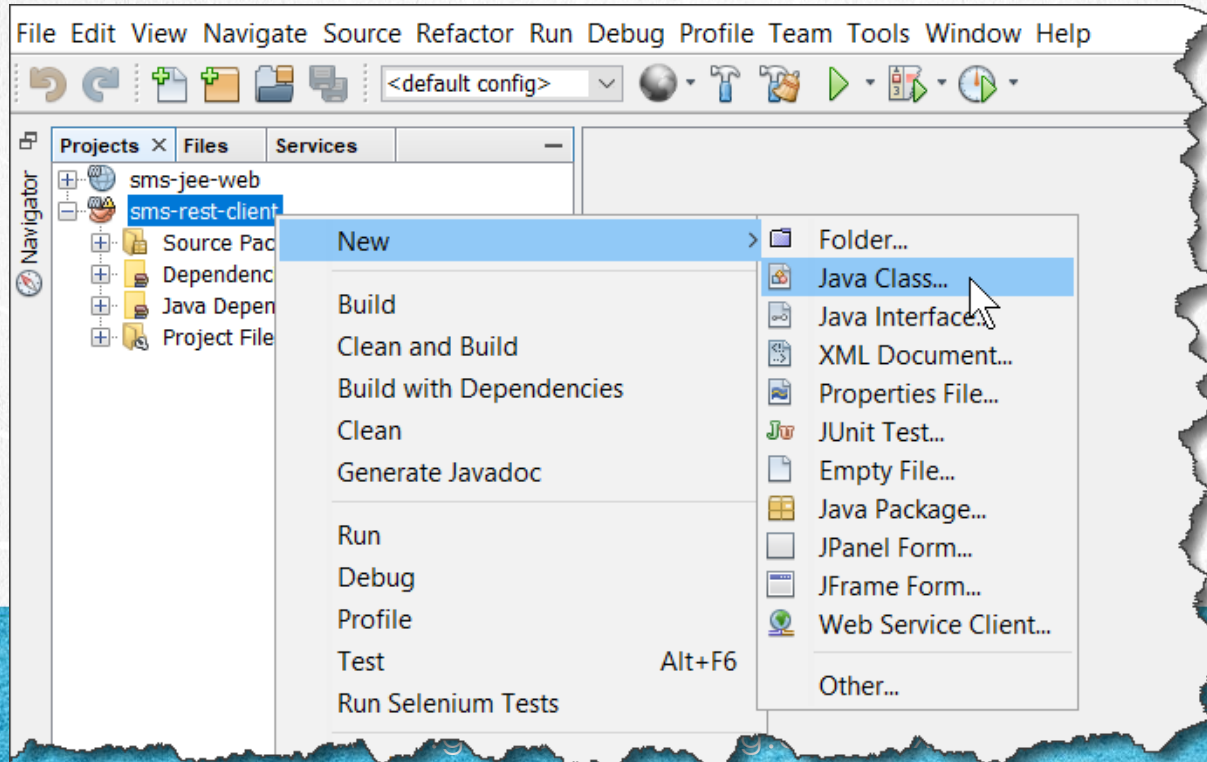
We update the maven libraries by doing clean & build to the project:





### 3. CREATE A JAVA CLASS

Because the Web Service message uses the Person entity, it is necessary to create the Person.java class:



### 3. CREATE A JAVA CLASS

Because the Web Service message uses the Person entity, it is necessary to create the Person.java class:

**New Java Class**

**Steps**

1. Choose File Type
2. **Name and Location**

**Name and Location**

Class Name:

Project:

Location:

Package:

Created File:

< Back   Next >   **Finish**   Cancel   Help

## 4. MODIFY THE FILE

Person.java:

Click to download

```
package domain;

import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement
public class Person {

    private int idPerson;
    private String name;

    public int getIdPerson() {
        return idPerson;
    }

    public void setIdPerson(int idPerson) {
        this.idPerson = idPerson;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```



## 4. MODIFY THE FILE

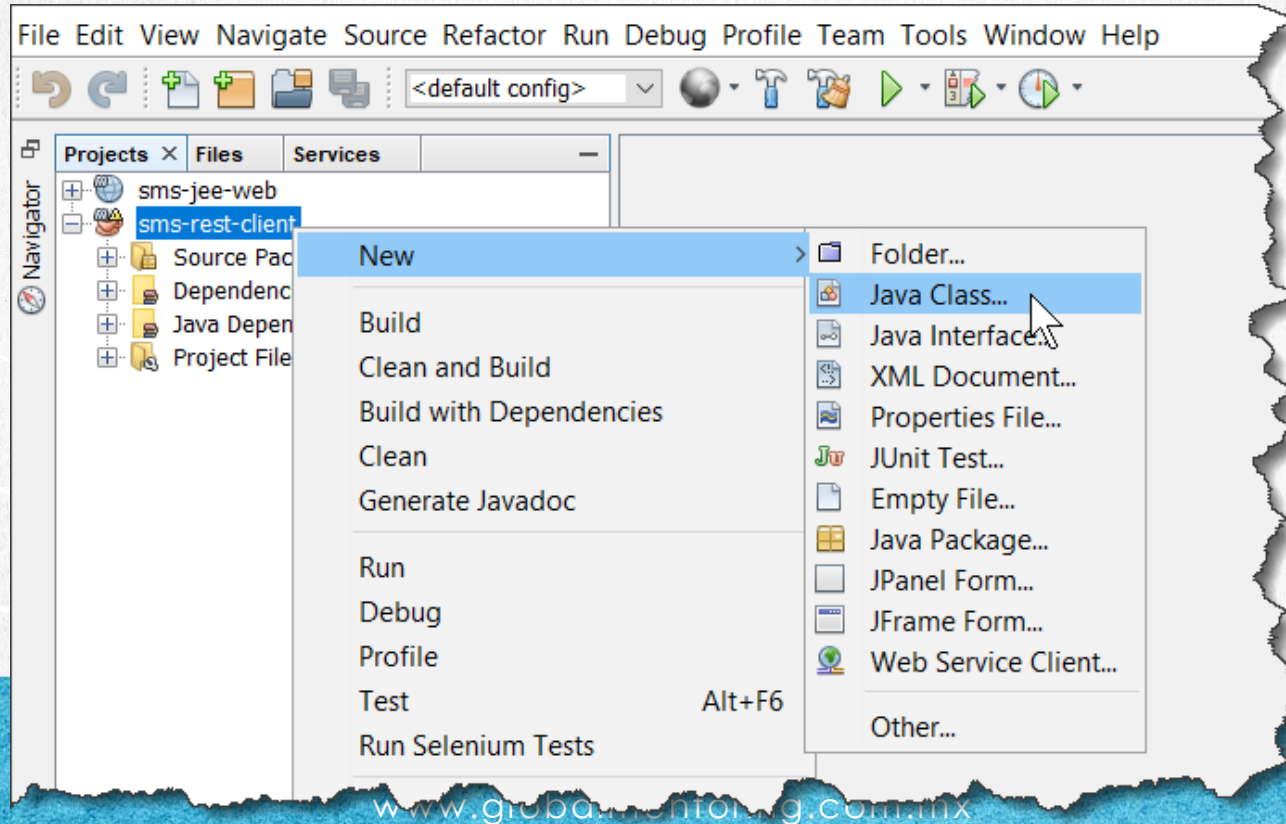
[Person.java:](#)

Click to download

```
@Override
public String toString() {
    return "Person{" + "idPerson=" + idPerson + ", name=" + name + '}';
}
```

# 5. CREATE A JAVA CLASS

We create the class TestPersonServiceRS.java:



## 5. CREATE A JAVA CLASS

We create the class TestPersonServiceRS.java:

**New Java Class**

**Steps**

1. Choose File Type
2. **Name and Location**

**Name and Location**

Class Name: TestPersonServiceRS

Project: sms-rest-client

Location: Source Packages

Package: test

Created File: C:\Courses\JavaEE\Lesson07\sms-rest-client\src\main\java\test\TestPersonServiceRS.java

< Back   Next >   **Finish**   Cancel   Help

**JAVA EE COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)



## 6. MODIFY THE CODE

### TestPersonServiceRS.java:

Click to download

```
package test;

import java.util.List;
import domain.Person;
import javax.ws.rs.client.*;
import javax.ws.rs.core.*;

public class TestPersonServiceRS {

    //Variables that we will use
    private static final String URL_BASE = "http://localhost:8080/sms-jee-web/webservice";
    private static Client client;
    private static WebTarget webTarget;
    private static Person person;
    private static List<Person> people;
    private static Invocation.Builder invocationBuilder;
    private static Response response;
```

**JAVA EE COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

## 6. MODIFY THE CODE

### TestPersonServiceRS.java:

Click to download

```
public static void main(String[] args) {

    client = ClientBuilder.newClient();

    //Read a person (get method)
    webTarget = client.target(URL_BASE).path("/people");
    //We provide a valid idPerson
    person = webTarget.path("/1").request(MediaType.APPLICATION_XML).get(Person.class);
    System.out.println("recovered Person: " + person);

    //Read all people (get method with readEntity of type List <>)
    people = webTarget.request(MediaType.APPLICATION_XML).get(Response.class).readEntity(new
GenericType<List<Person>>() {});
    System.out.println("\nRecovered people:");
    printPeople(people);
}
```

**JAVA EE COURSE**

www.globalmentoring.com.mx

## 6. MODIFY THE CODE

### TestPersonServiceRS.java:

[Click to download](#)

```
//Add a person (post method)
Person newPerson = new Person();
newPerson.setName("Ana");
invocationBuilder = webTarget.request(MediaType.APPLICATION_XML);
response = invocationBuilder.post(Entity.entity(newPerson, MediaType.APPLICATION_XML));

System.out.println("");
System.out.println("add response status:" + response.getStatus());
//We retrieve the newly added person and then modify it and finally eliminate it
Person addedPerson = response.readEntity(Person.class);
System.out.println("Person added: " + addedPerson);

//Modify a person (put method)
//person previously recovered
Person modifiedPerson = addedPerson;
modifiedPerson.setName("Ana Skelletion");
String pathId = "/" + modifiedPerson.getIdPerson();
invocationBuilder = webTarget.path(pathId).request(MediaType.APPLICATION_XML);
response = invocationBuilder.put(Entity.entity(modifiedPerson, MediaType.APPLICATION_XML));

System.out.println("");
System.out.println("modify response status:" + response.getStatus());
System.out.println("Modified person:" + response.readEntity(Person.class));
```



## 6. MODIFY THE CODE

TestPersonServiceRS.java:

Click to download

```
//Delete a person
// person previously recovered
Person deletePerson = addedPerson;
String pathDeleteId = "/" + deletePerson.getIdPerson();
invocationBuilder = webTarget.path(pathDeleteId).request(MediaType.APPLICATION_XML);
response = invocationBuilder.delete();

System.out.println("");
System.out.println("remove response status : " + response.getStatus());
System.out.println("Person Removed: " + deletePerson);
}

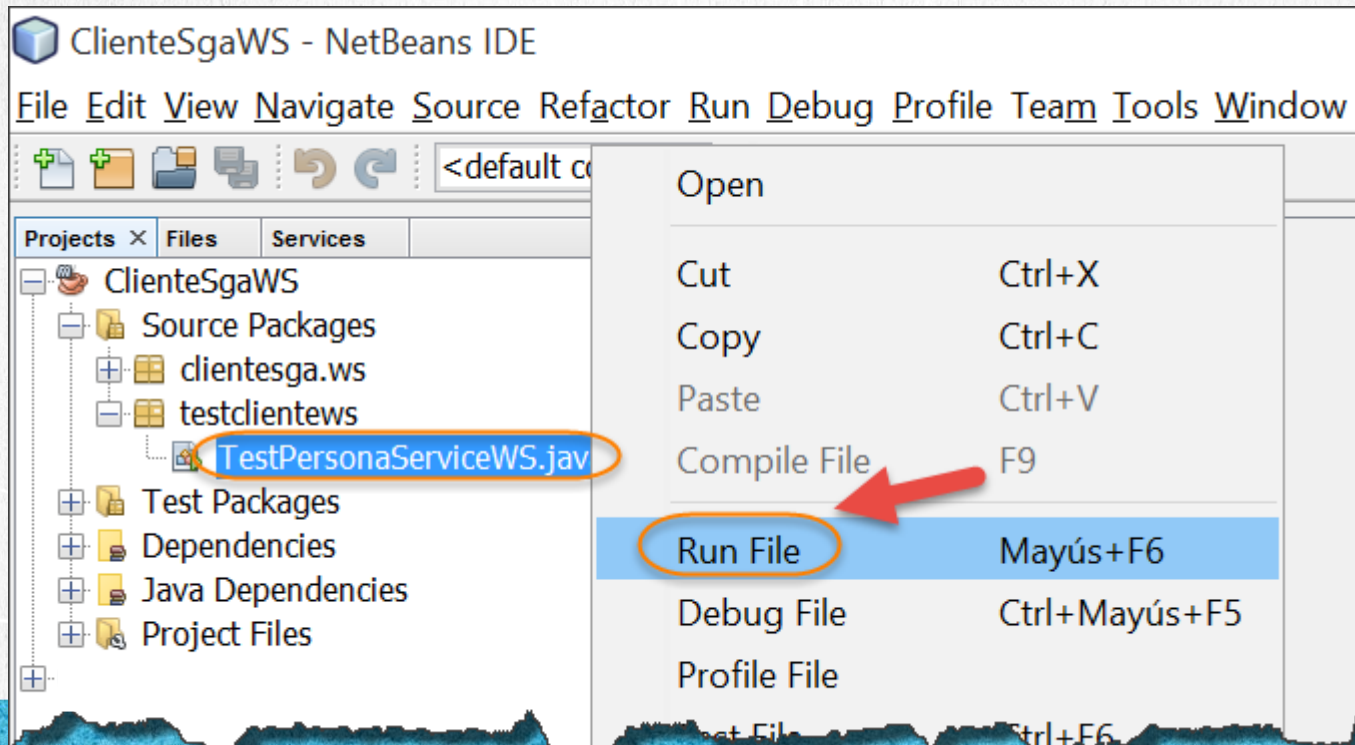
private static void printPeople(List<Person> persons) {
    for (Person person : persons) {
        System.out.println("Person:" + person);
    }
}
}
```

**JAVA EE COURSE**

www.globalmentoring.com.mx

# 7. EXECUTE THE APPLICATION

Execute the class:

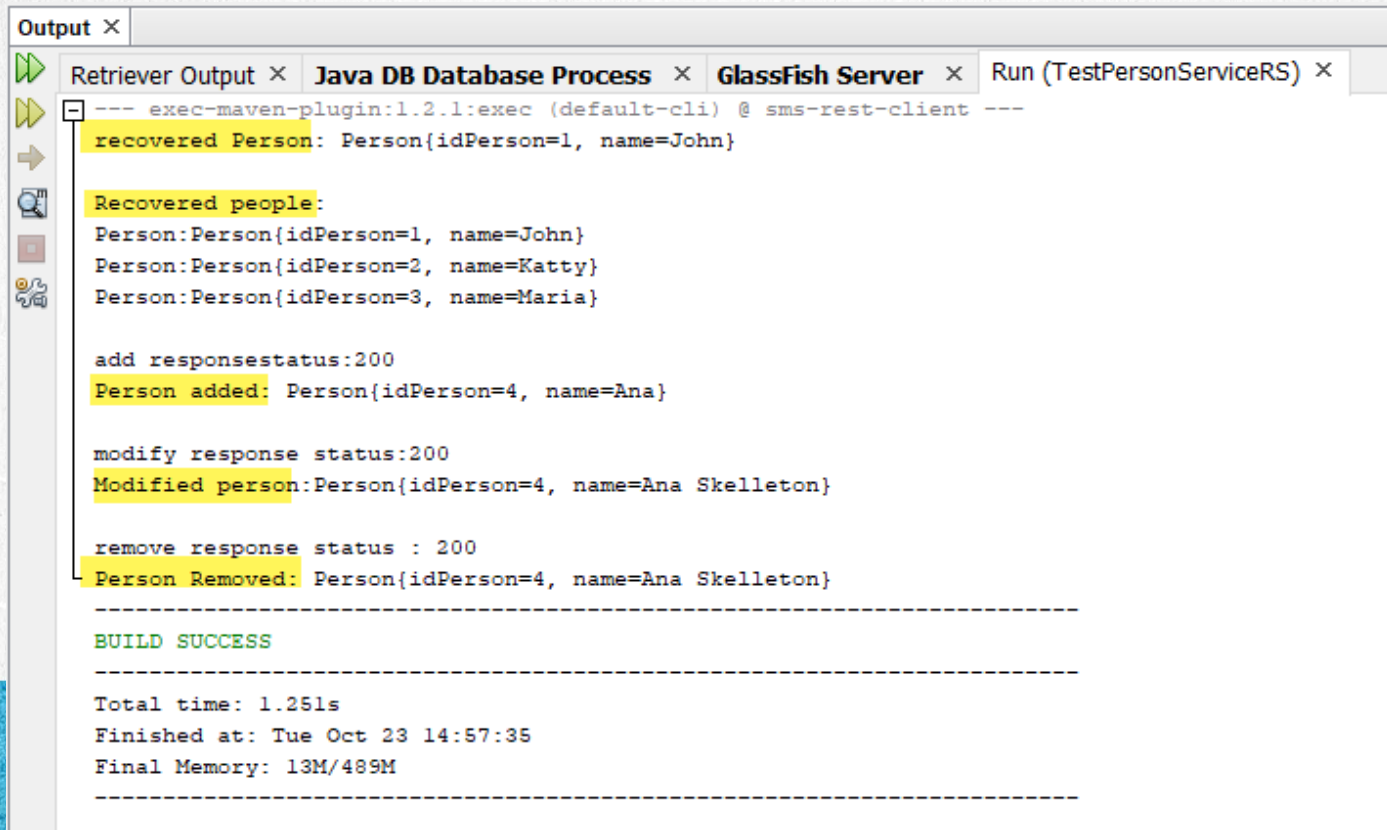


**JAVA EE COURSE**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# 7. EXECUTE THE APPLICATION

Execute the class:



```
Output X
Retriever Output X Java DB Database Process X GlassFish Server X Run (TestPersonServiceRS) X
--- exec-maven-plugin:1.2.1:exec (default-cli) @ sms-rest-client ---
recovered Person: Person{idPerson=1, name=John}

Recovered people:
Person:Person{idPerson=1, name=John}
Person:Person{idPerson=2, name=Katty}
Person:Person{idPerson=3, name=Maria}

add responsestatus:200
Person added: Person{idPerson=4, name=Ana}

modify response status:200
Modified person:Person{idPerson=4, name=Ana Skeleton}

remove response status : 200
Person Removed: Person{idPerson=4, name=Ana Skeleton}

-----
BUILD SUCCESS
-----

Total time: 1.251s
Finished at: Tue Oct 23 14:57:35
Final Memory: 13M/489M
-----
```



# EXERCISE CONCLUSION

- ✔ We observe how to make a test of the Web Service once deployed on the GlassFish server.
- ✔ In addition, we review how to create the Java classes of the Web Service Client. In this case we do not need help from any tool, since the management is easier than in SOAP Web Services.
- ✔ We saw how to check the publication of the RESTful Web Service with the help of the WADL document.
- ✔ With this we have seen the complete process of how to create a REST Web Service and the respective client.
- ✔ Last but not least, we use a 3-layer architecture and we saw how to expose the business logic as a Web Service using REST to any client interested in the system information.

# ONLINE COURSE

# JAVA EE

# JAKARTA EE

---

By: Eng. Ubaldo Acosta



**JAVA EE COURSE**  
[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)