JAVA EE COURSE

SECURING JAX-RS CLIENT WITH JAVA EE

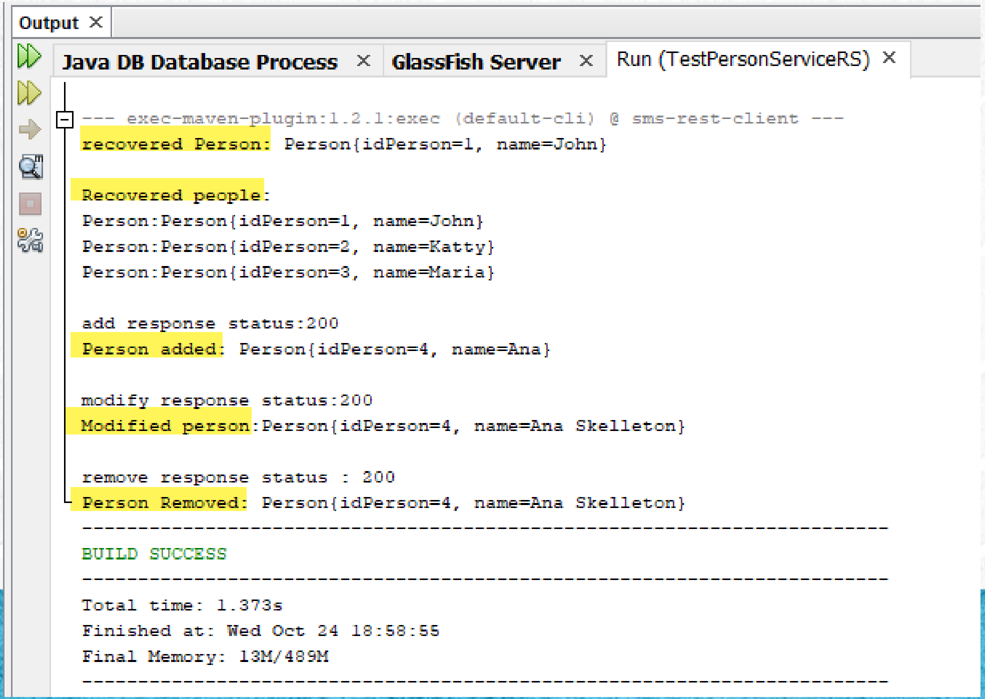
By the expert: Eng. Ubaldo Acosta

JAVA EE COURSE

www.globalmentoring.com.mx

EXERCISE OBJECTIVE

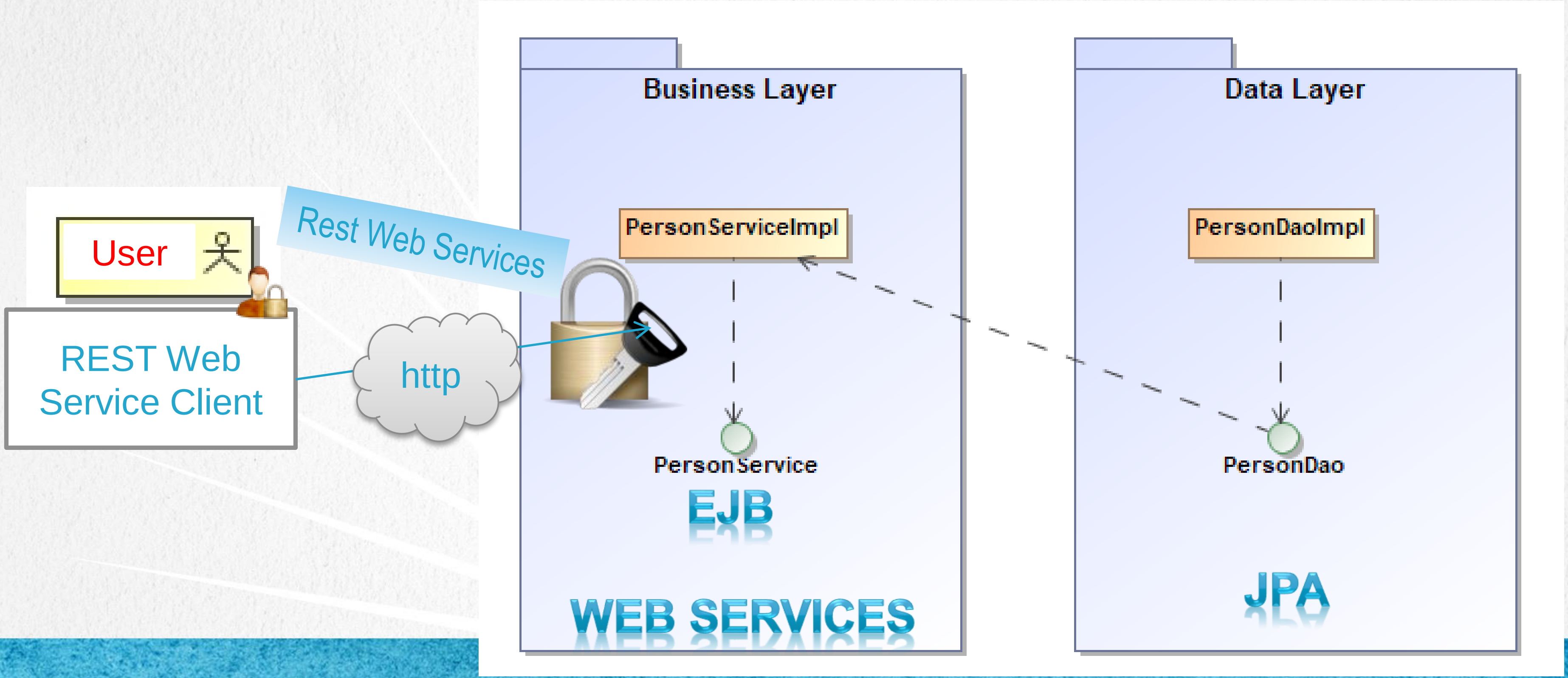
•The objective of the exercise is to modify the JAX-RS client in order to correctly access the Web Service of the People list, providing the respective credentials (user and password) to the SMS system. This figure is the result is similar to the following:



```
--- 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 response status:200
Person added: Person{idPerson=4, name=Ana}

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

remove response status : 200
Person Removed: Person{idPerson=4, name=Ana Skelleton}
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time: 1.373s
Finished at: Wed Oct 24 18:58:55
Final Memory: 13M/489M
------------------------------------------------------------------------
```

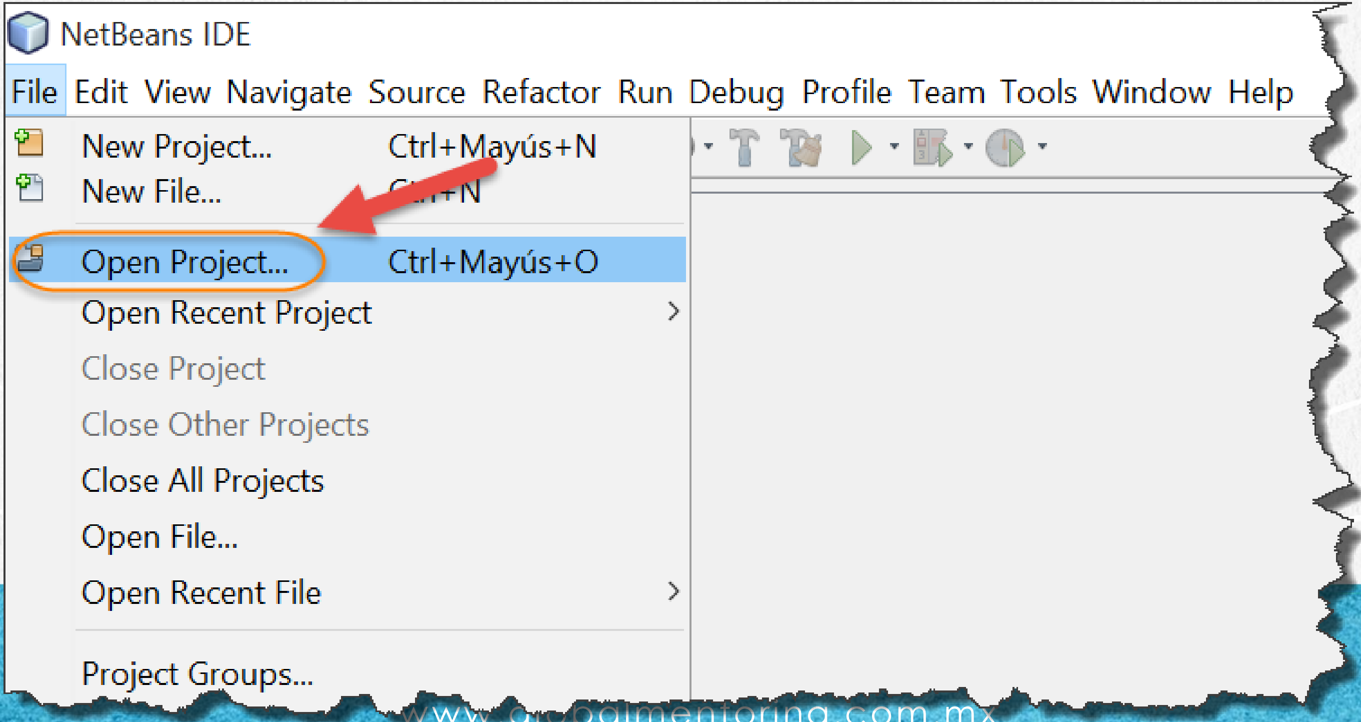
SMS ARCHITECTURE WITH WEB SERVICES

This is the Exercise Class Diagram, where you can see the Architecture of our System:



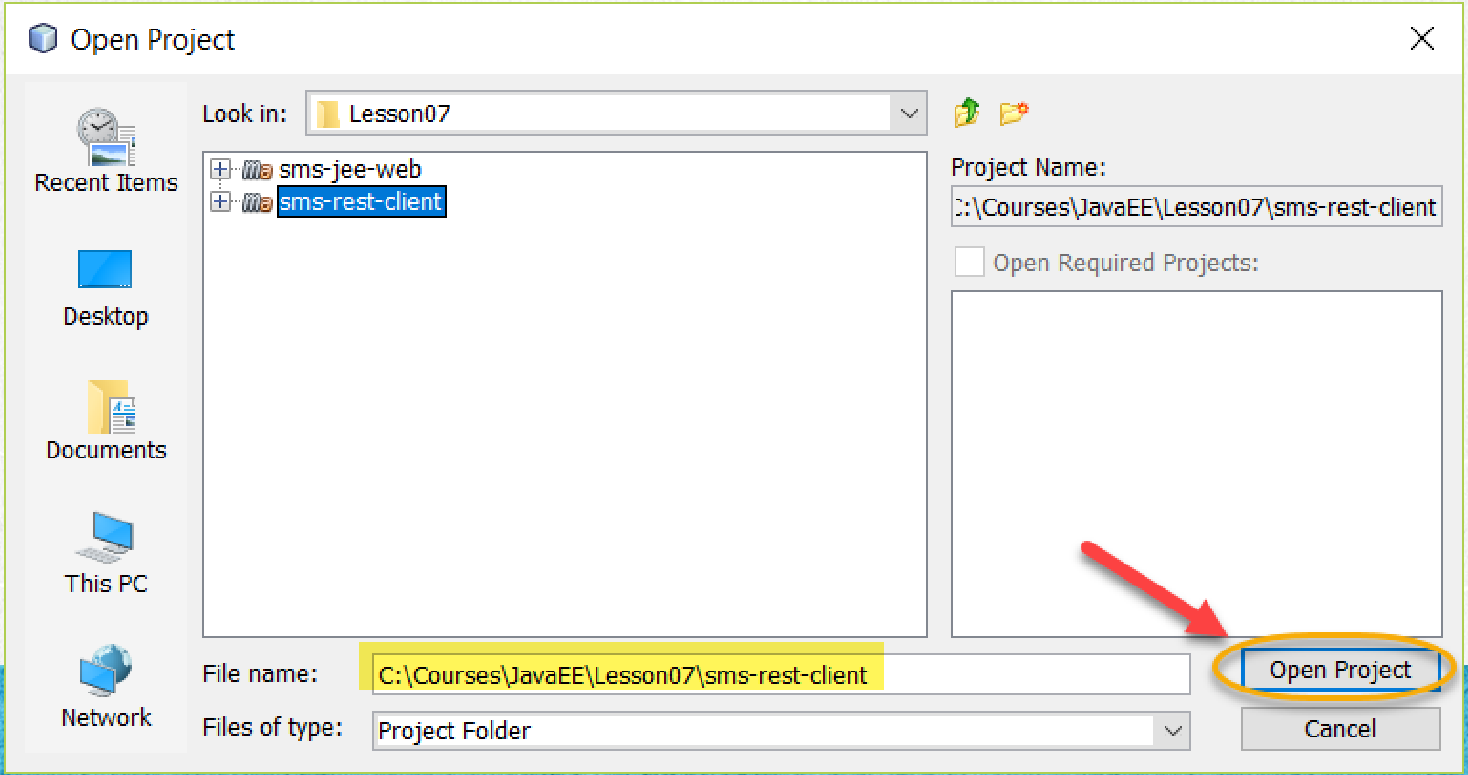
1. OPEN THE PROJECT

In case we do not have the sms-rest-client project open, we open it:



1. OPEN THE PROJECT

In case we do not have the sms-rest-client project open, we open it:

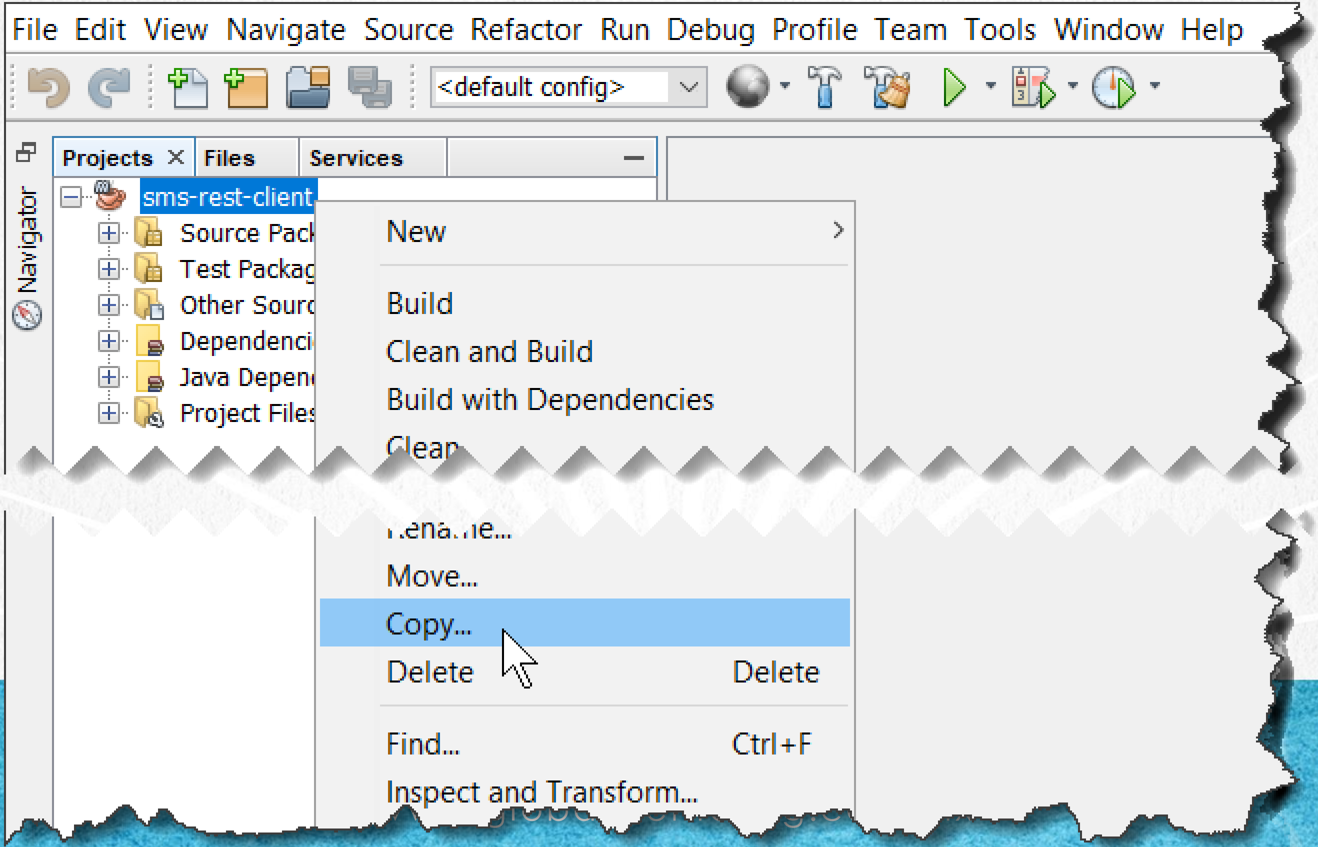


File name: C:\Courses\JavaEE\Lesson07\sms-rest-client

Files of type: Project Folder

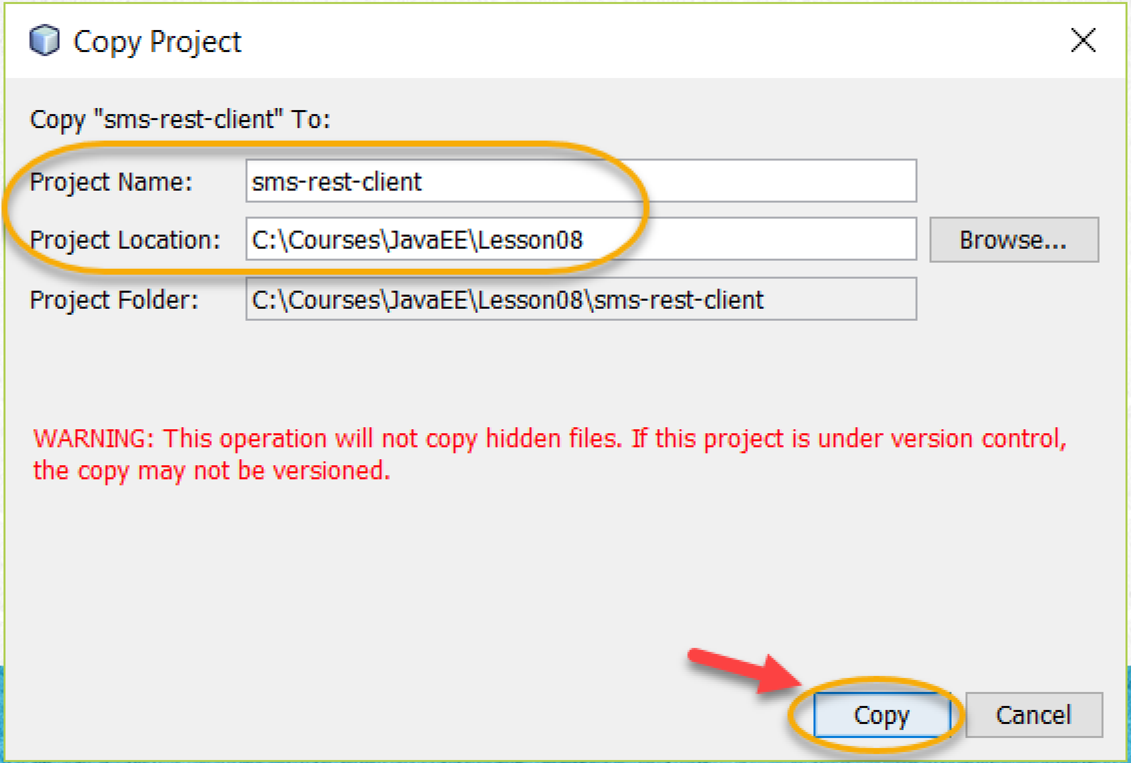
2. COPY THE PROJECT

We copy the project to put it in the new path:



2. COPY THE PROJECT

We copy the project to put it in the new path:



Copy Project

Copy "sms-rest-client" To:

Project Name: sms-rest-client

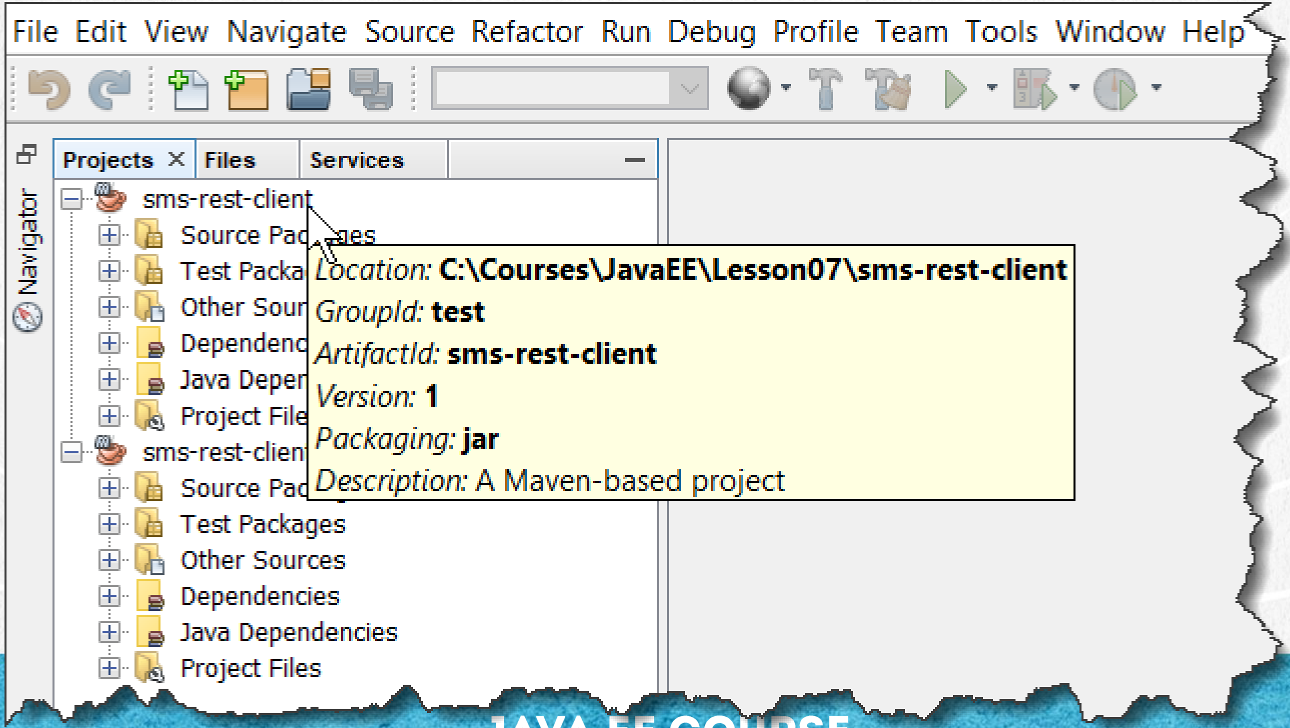
Project Location: C:\Courses\JavaEE\Lesson08

Project Folder: C:\Courses\JavaEE\Lesson08\sms-rest-client

WARNING: This operation will not copy hidden files. If this project is under version control, the copy may not be versioned.

3. CLOSE THE PREVIOUS PROJECT

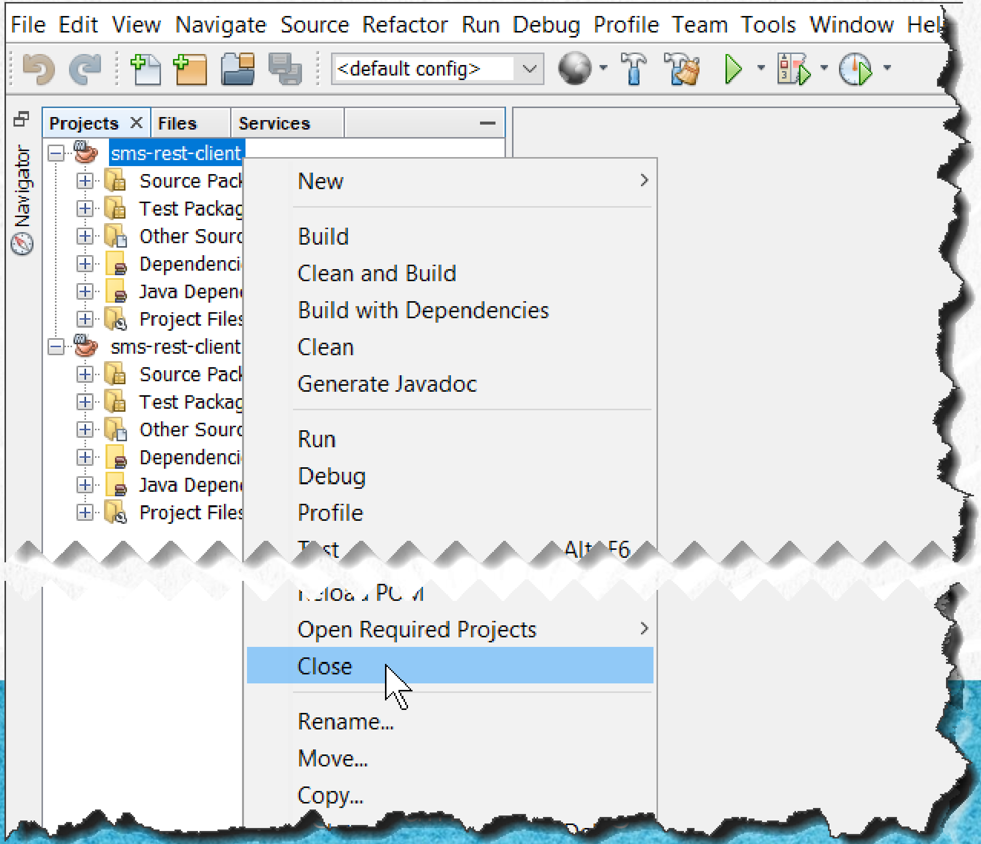
We closed the previous project and left only the new one:



Location: **C:\Courses\JavaEE\Lesson07\sms-rest-client**
GroupId: **test**
ArtifactId: **sms-rest-client**
Version: **1**
Packaging: **jar**
Description: A Maven-based project

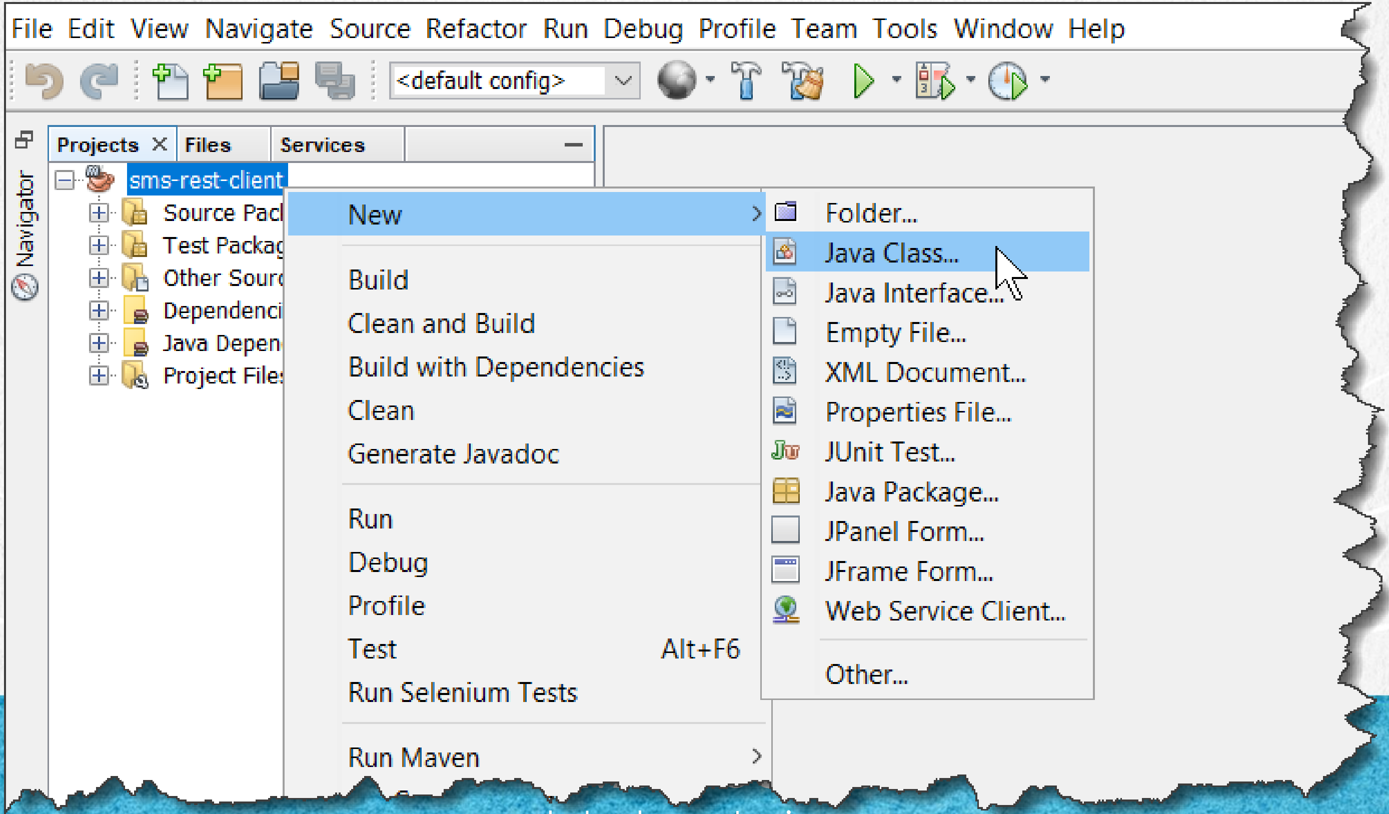
3. CLOSE THE PREVIOUS PROJECT

We closed the previous project and left only the new one:



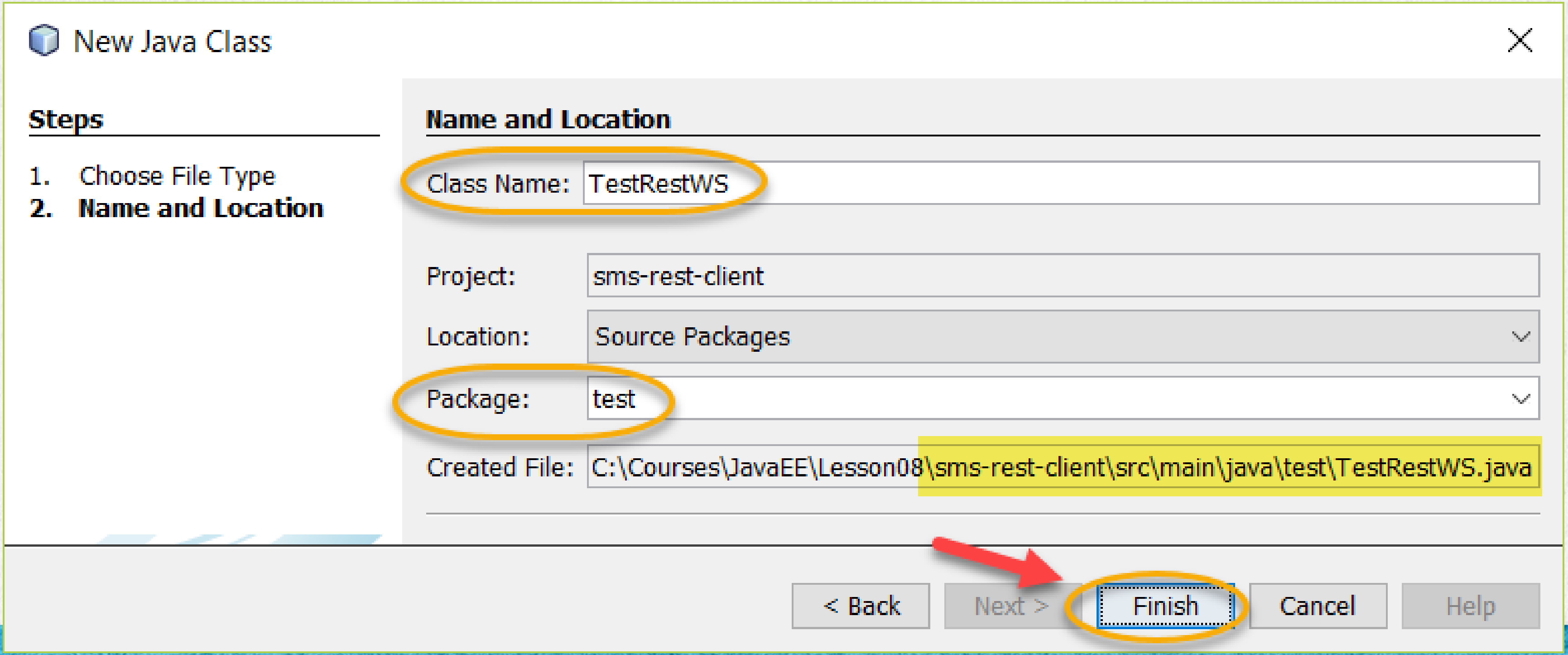
4. CREATE A JAVA FILE

We add the TestRestWS.java class:



4. CREATE A JAVA FILE

We add the TestRestWS.java class:



New Java Class

Steps

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

Name and Location

Class Name: TestRestWS

Project: sms-rest-client

Location: Source Packages

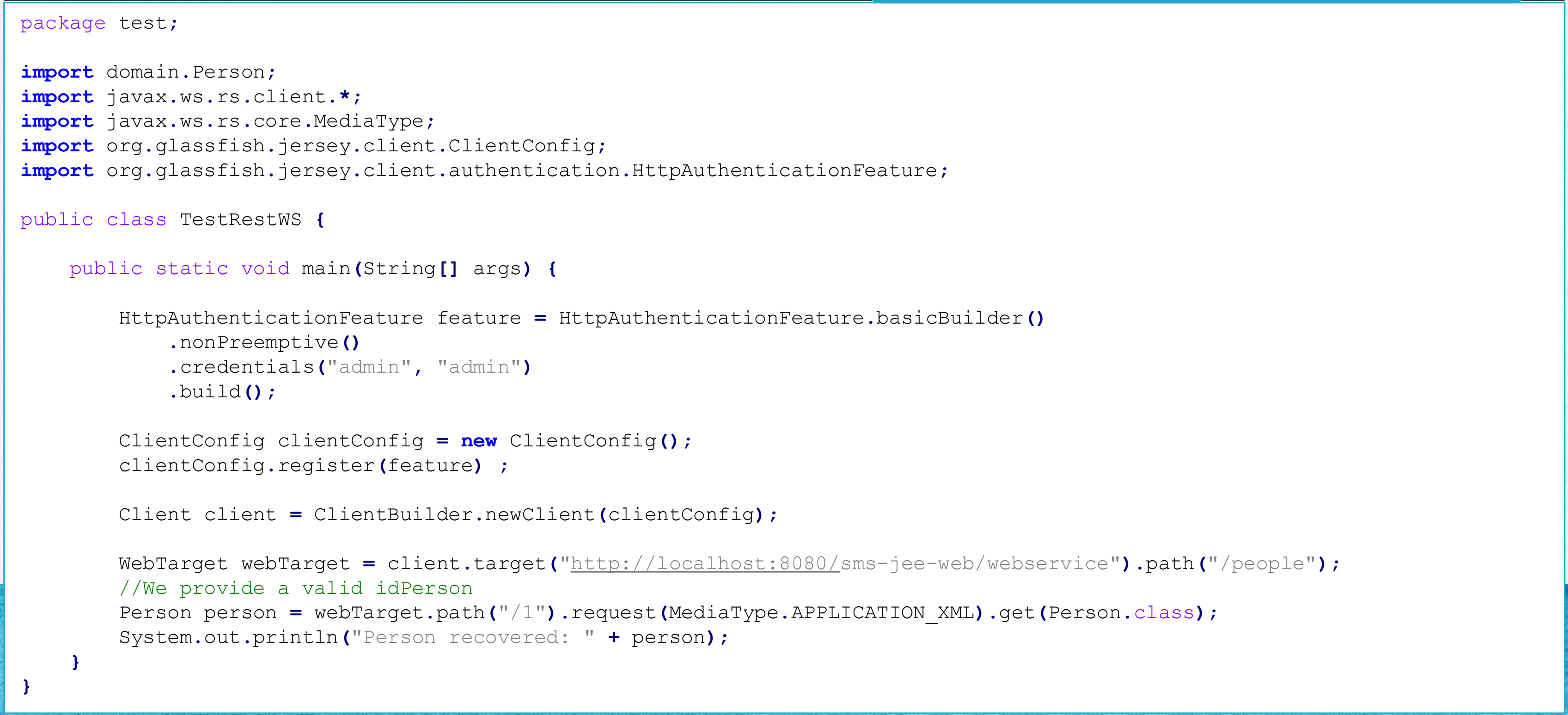
Package: test

Created File: C:\Courses\JavaEE\Lesson08\sms-rest-client\src\main\java\test\TestRestWS.java

5. MODIFY THE FILE

TestRestWS.java:

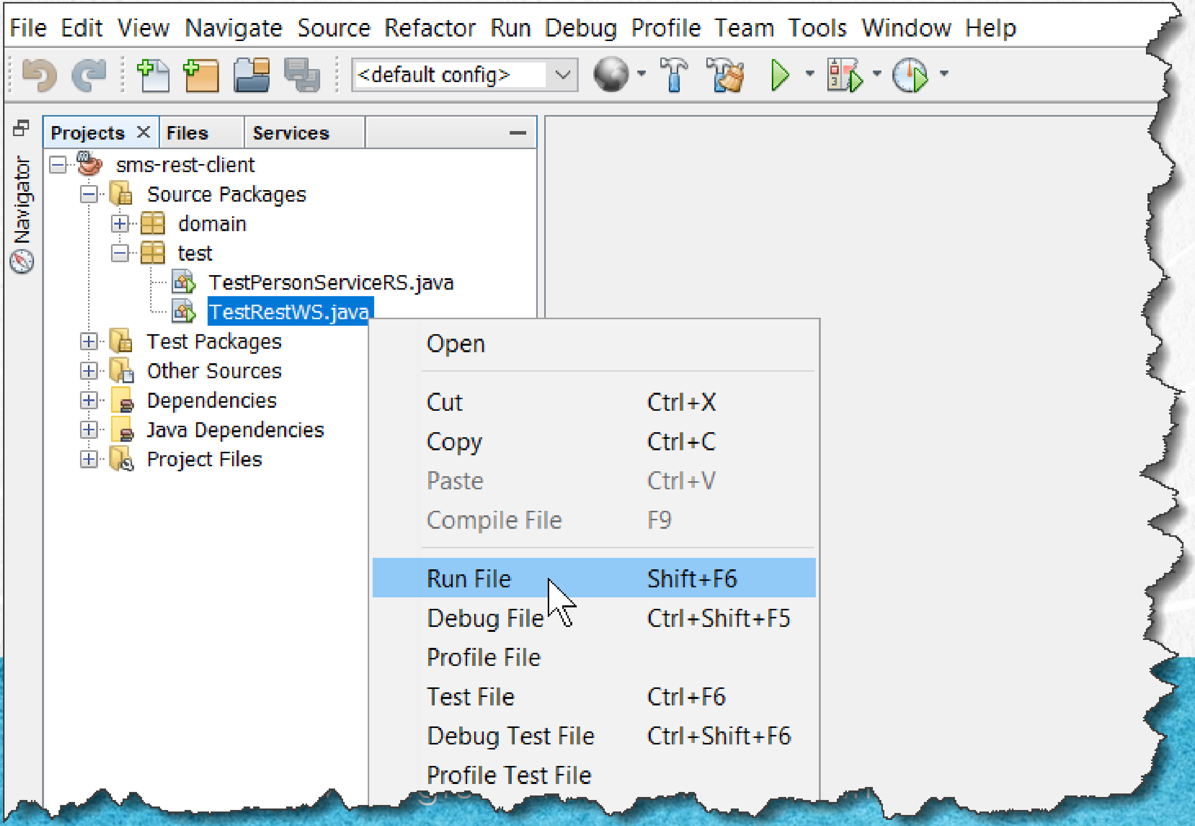
Click to download



```
package test;

import domain.Person;
import javax.ws.rs.client.*;
import javax.ws.rs.core.MediaType;
import org.glassfish.jersey.client.ClientConfig;
import org.glassfish.jersey.client.authentication.HttpAuthenticationFeature;

public class TestRestWS {

    public static void main(String[] args) {

        HttpAuthenticationFeature feature = HttpAuthenticationFeature.basicBuilder()
            .nonPreemptive()
            .credentials("admin", "admin")
            .build();

        ClientConfig clientConfig = new ClientConfig();
        clientConfig.register(feature) ;

        Client client = ClientBuilder.newClient(clientConfig);

        WebTarget webTarget = client.target("http://localhost:8080/sms-jee-web/webservice").path("/people");
        //We provide a valid idPerson
        Person person = webTarget.path("/1").request(MediaType.APPLICATION_XML).get(Person.class);
        System.out.println("Person recovered: " + person);
    }
}
```

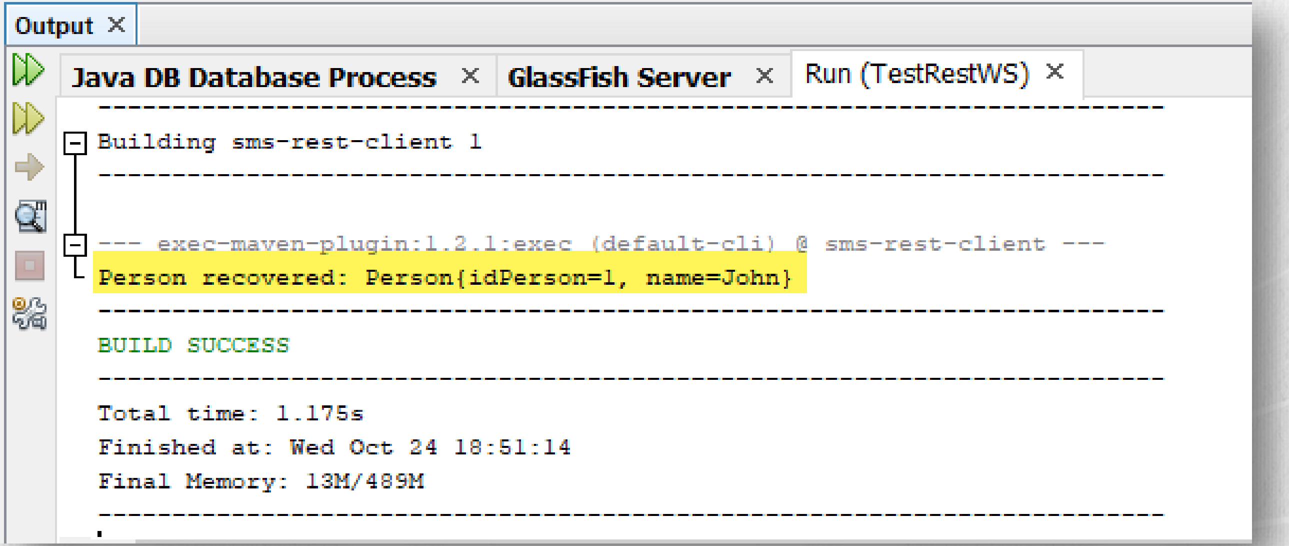
6. EXECUTE THE PROJECT

We run the TestRestWS.java class. Note: The sms-jee-web application and the Glassfish server must already be started for the test to work:



6. EXECUTE THE APPLICATION

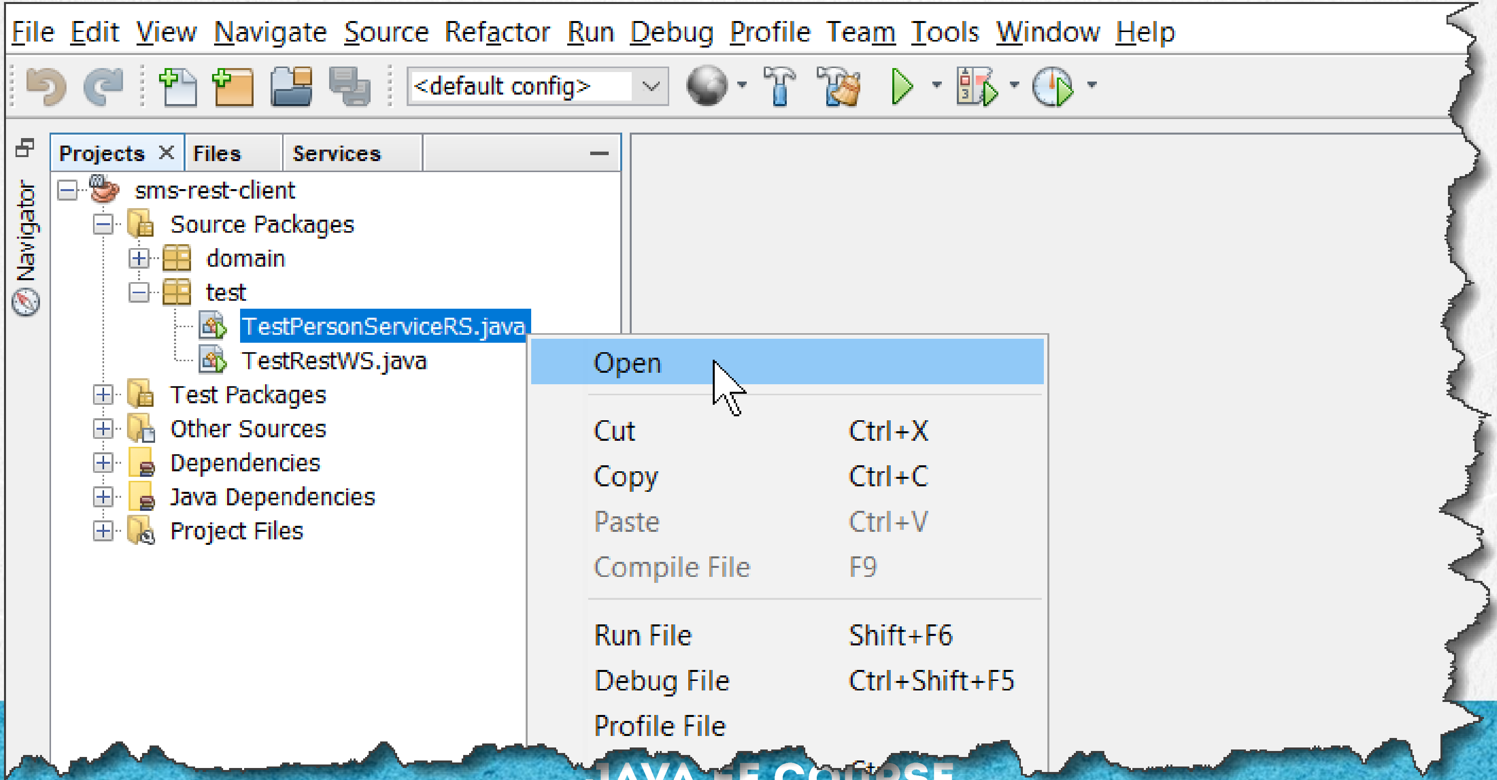
We run the TestRestWS.java class. The values can vary according to the idPerson used for the test:



```
------------------------------------------------------------------------
Building sms-rest-client 1
------------------------------------------------------------------------

--- exec-maven-plugin:1.2.1:exec (default-cli) @ sms-rest-client ---
Person recovered: Person{idPerson=1, name=John}
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time: 1.175s
Finished at: Wed Oct 24 18:51:14
Final Memory: 13M/489M
------------------------------------------------------------------------
```

7. MODIFY A JAVA CLASS

Modify the test to support security:



7. MODIFY THE FILE

TestPersonServiceRS.java:

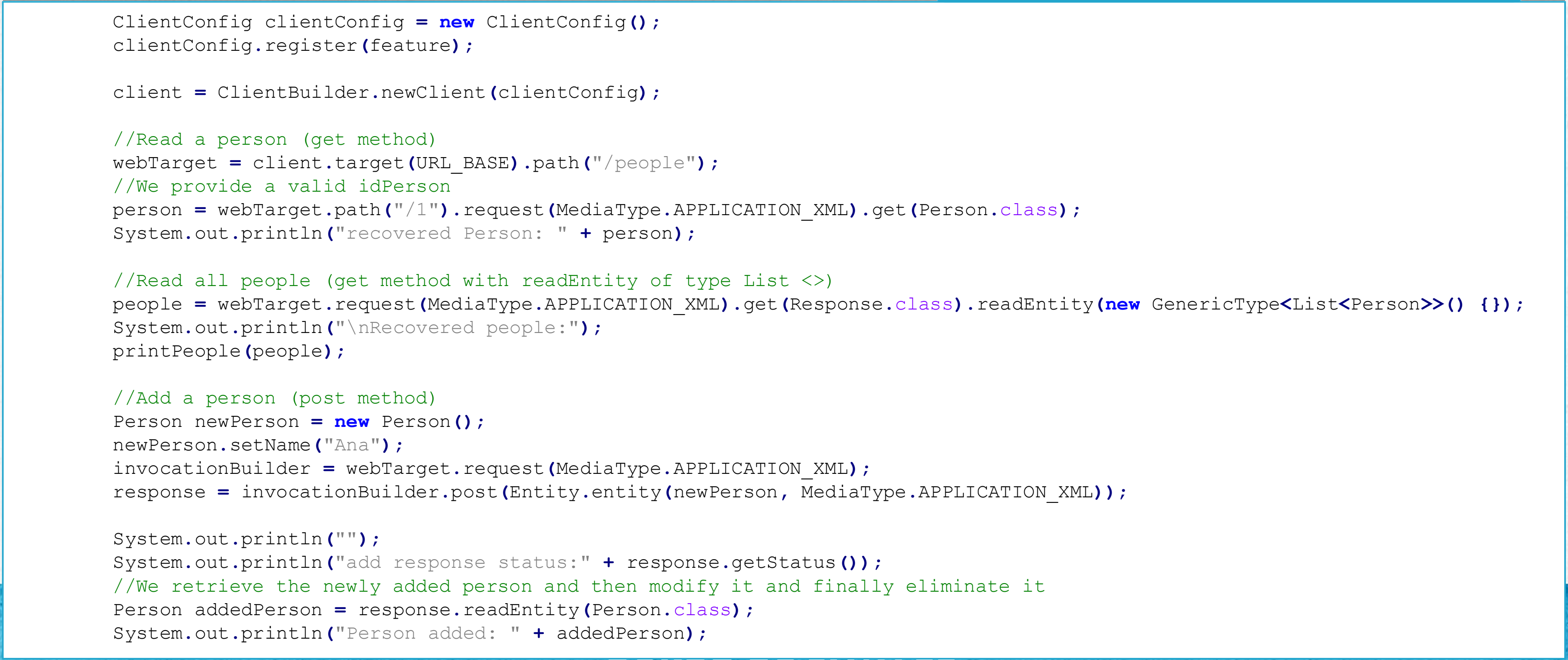
click to download

```
package test;

import java.util.List;
import domain.Person;
import javax.ws.rs.client.*;
import javax.ws.rs.core.*;
import org.glassfish.jersey.client.ClientConfig;
import org.glassfish.jersey.client.authentication.HttpAuthenticationFeature;

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;

    public static void main(String[] args) {

         HttpAuthenticationFeature feature = HttpAuthenticationFeature.basicBuilder()
                .nonPreemptive()
                .credentials("admin", "admin")
                .build();
```

7. MODIFY THE FILE

TestPersonServiceRS.java:

click to download

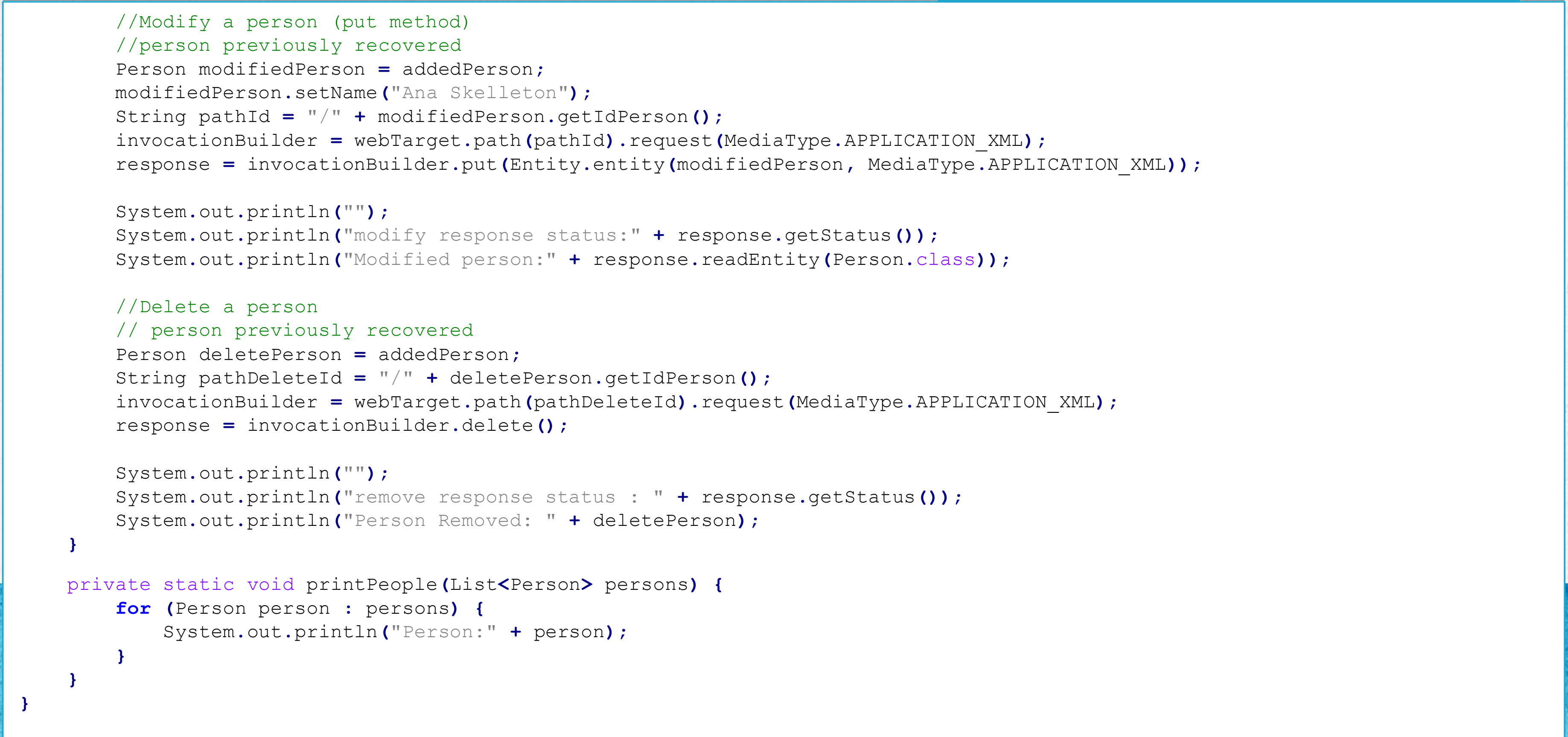


```
        ClientConfig clientConfig = new ClientConfig();
        clientConfig.register(feature);

        client = ClientBuilder.newClient(clientConfig);

        //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);

        //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);
```

7. MODIFY THE FILE

TestPersonServiceRS.java:

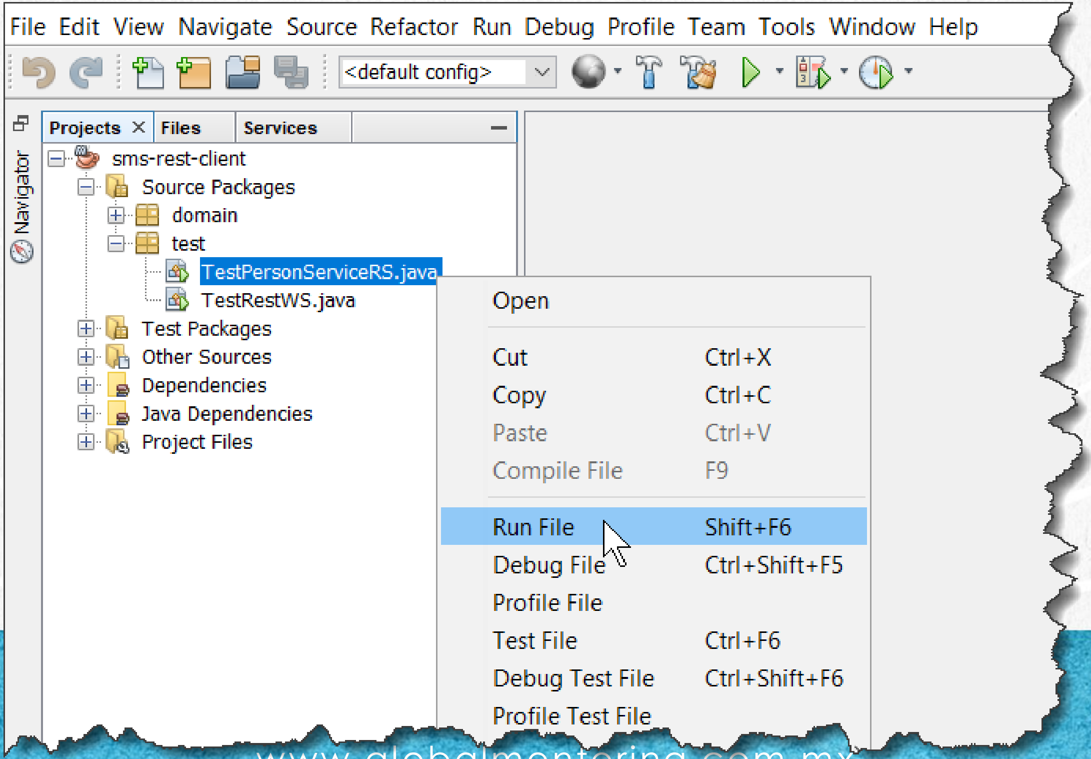
click to download



```java
        //Modify a person (put method)
        //person previously recovered
        Person modifiedPerson = addedPerson;
        modifiedPerson.setName("Ana Skelleton");
        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));

        //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);
        }
    }
}
```

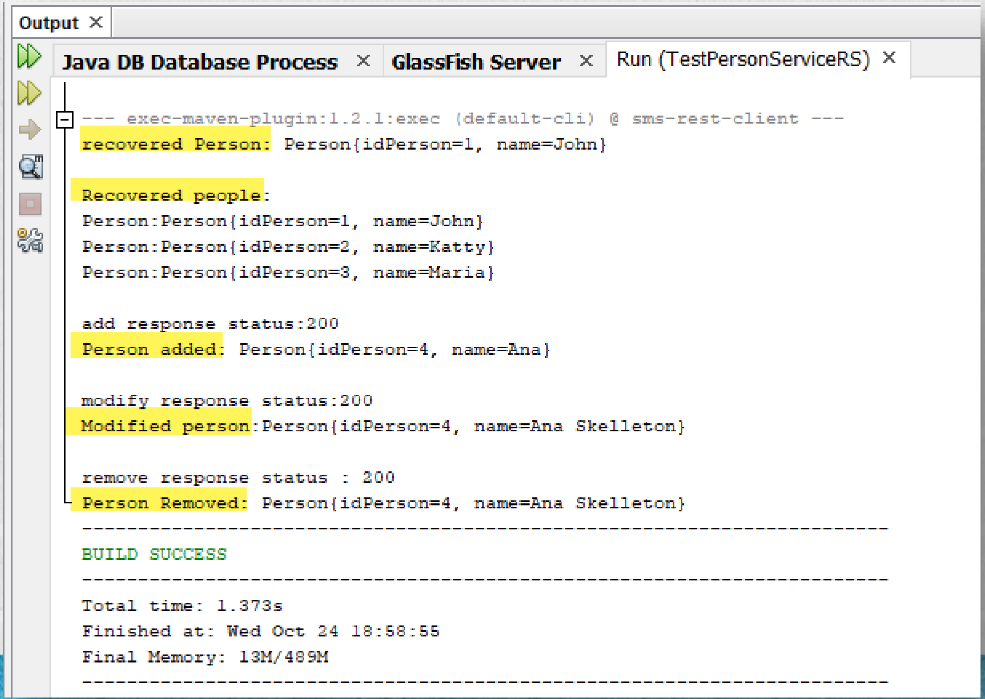
8. EXECUTE THE PROJECT

We execute the class TestPersonServiceRS.java. Note: The sms-jee-web application and the Glassfish server must already be started for the test to work:



8. EXECUTE THE APPLICATION

We execute the class TestPersonaServiceRS.java. The values may vary according to the data used for the test:



```
--- 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 response status:200
Person added: Person{idPerson=4, name=Ana}

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

remove response status : 200
Person Removed: Person{idPerson=4, name=Ana Skelleton}
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time: 1.373s
Finished at: Wed Oct 24 18:58:55
Final Memory: 13M/489M
------------------------------------------------------------------------
```

IN CASE OF ERROR

- Verify that you have the latest version of the projects by doing Clean & build on them.
- The Glassfish server must already be started
- The idPerson must be valid and exist in the database.
- If none of this works, try loading the completed project, and rerun the test.

EXERCISE CONCLUSION

With this exercise we have modified the REST Web Service client to access the People Web Service of the sms-jee-web project.

We created a test class separately and we also modified the test that we already had created in order to verify that we have been able to correctly access the Web Service, which in turn is also an EJB of the Stateless type.

ONLINE COURSE

JAVA EE
JAKARTA EE

By: Eng. Ubaldo Acosta

Global
Mentoring

JAVA
ONLINE COURSES
UNIVERSITY

JAVA EE COURSE

www.globalmentoring.com.mx