JAVA EE COURSE

SECURING SMS SYSTEM WITH JAVA EE

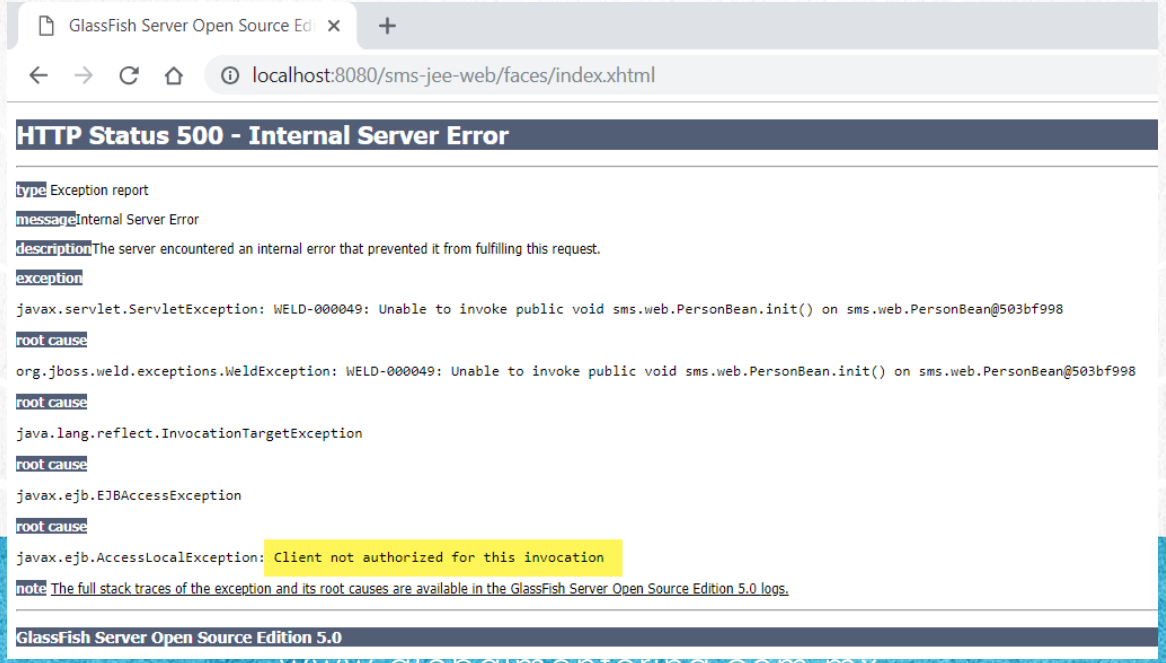
By the expert: Eng. Ubaldo Acosta

Global Mentoring

JAVA ONLINE COURSES UNIVERSITY

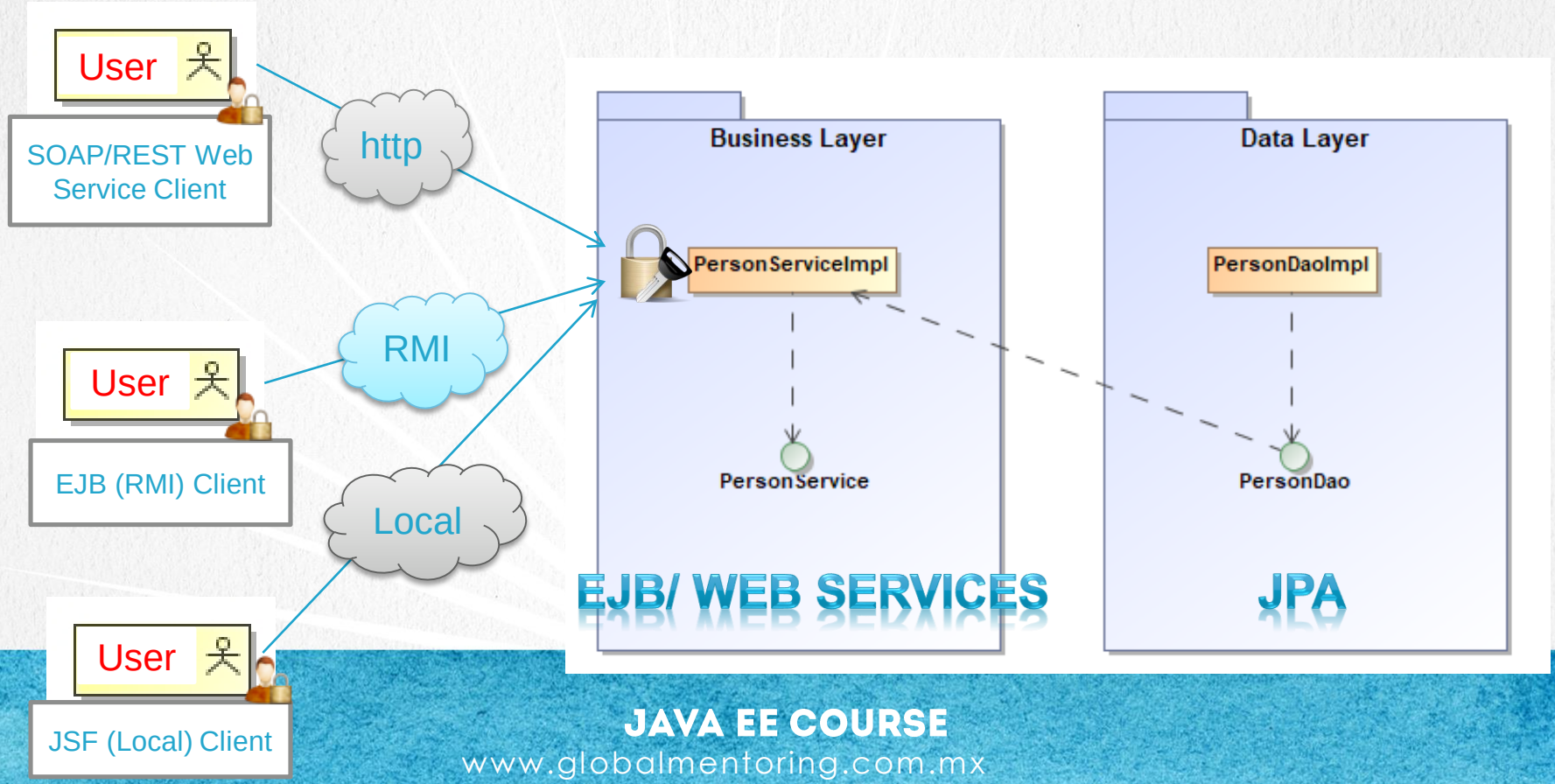
EXERCISE OBJECTIVE

•The objective of the exercise is the security and the business methods of the Person Service, and the ways of doing the login for each created client: Web Client, Web Service SOAP Client and Web Service REST Client and Client EJB. The final result is that each client must execute correctly when sending the respective credentials (user and password) to the SMS system. This figure is the result of the client's website (JSF):



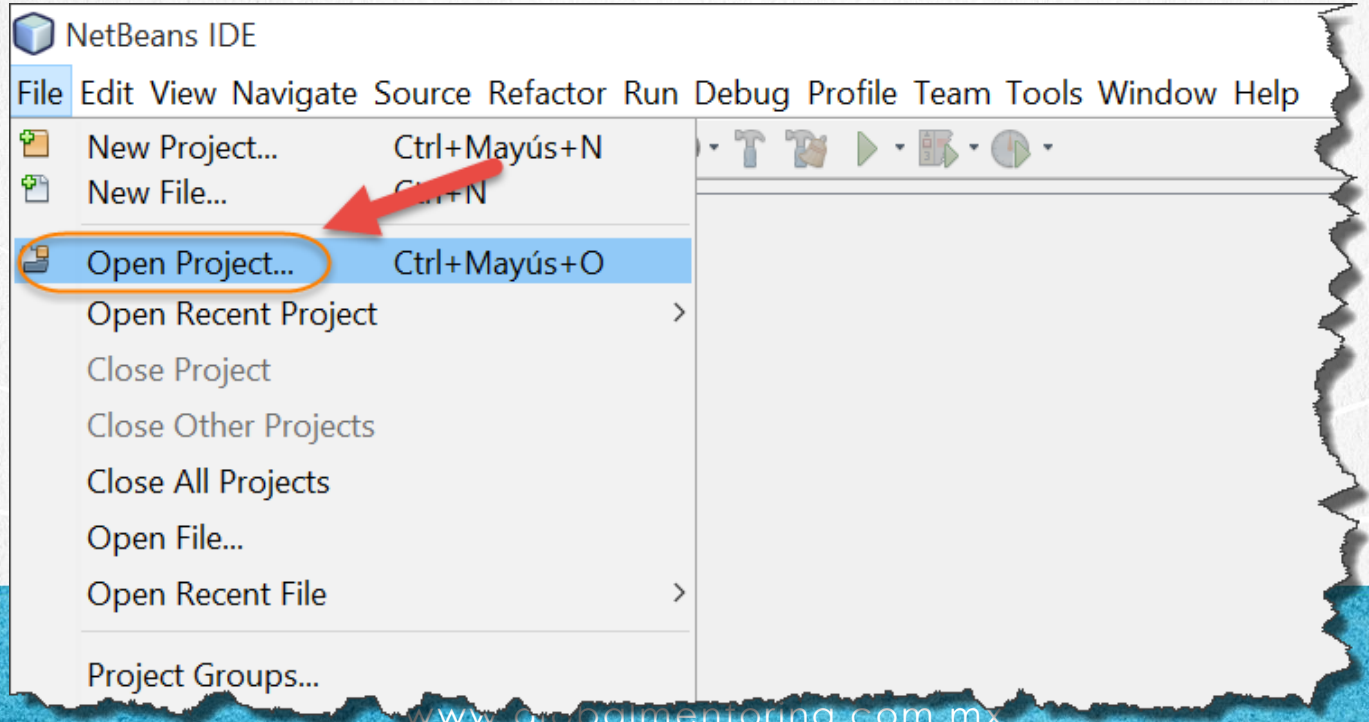
ARCHITECTURE WITH SECURITY JAVA EE

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



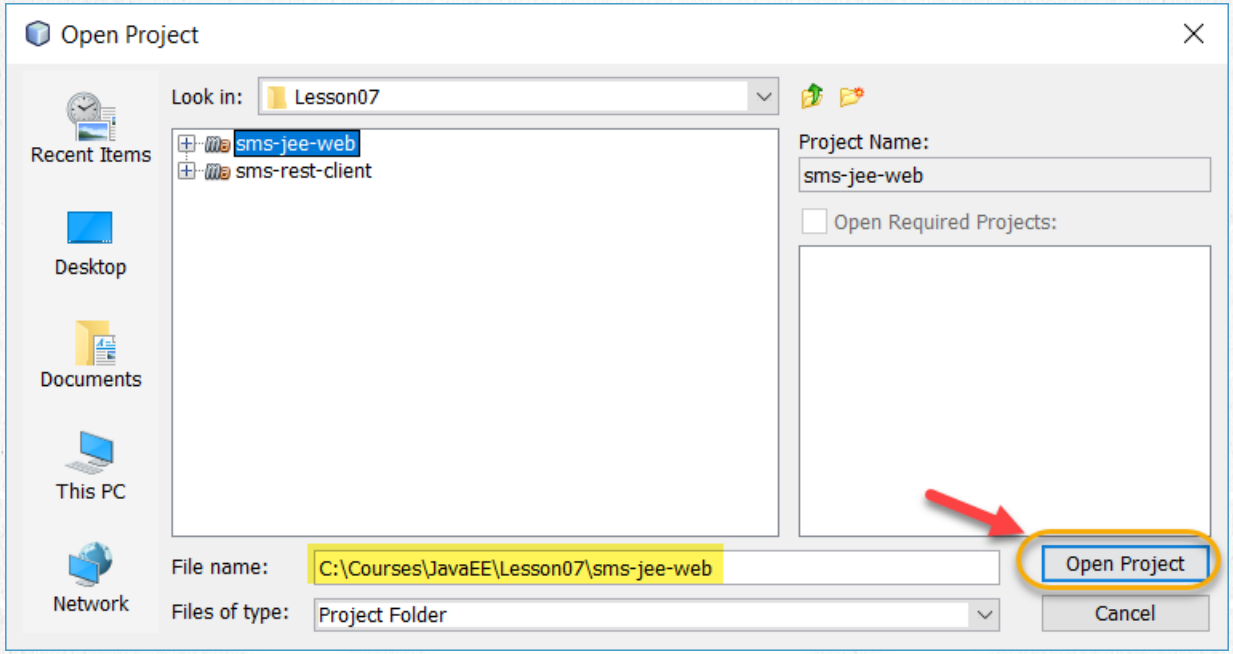
1. OPEN THE PROJECT

In case we do not have open the sms-jee-web project we open it:



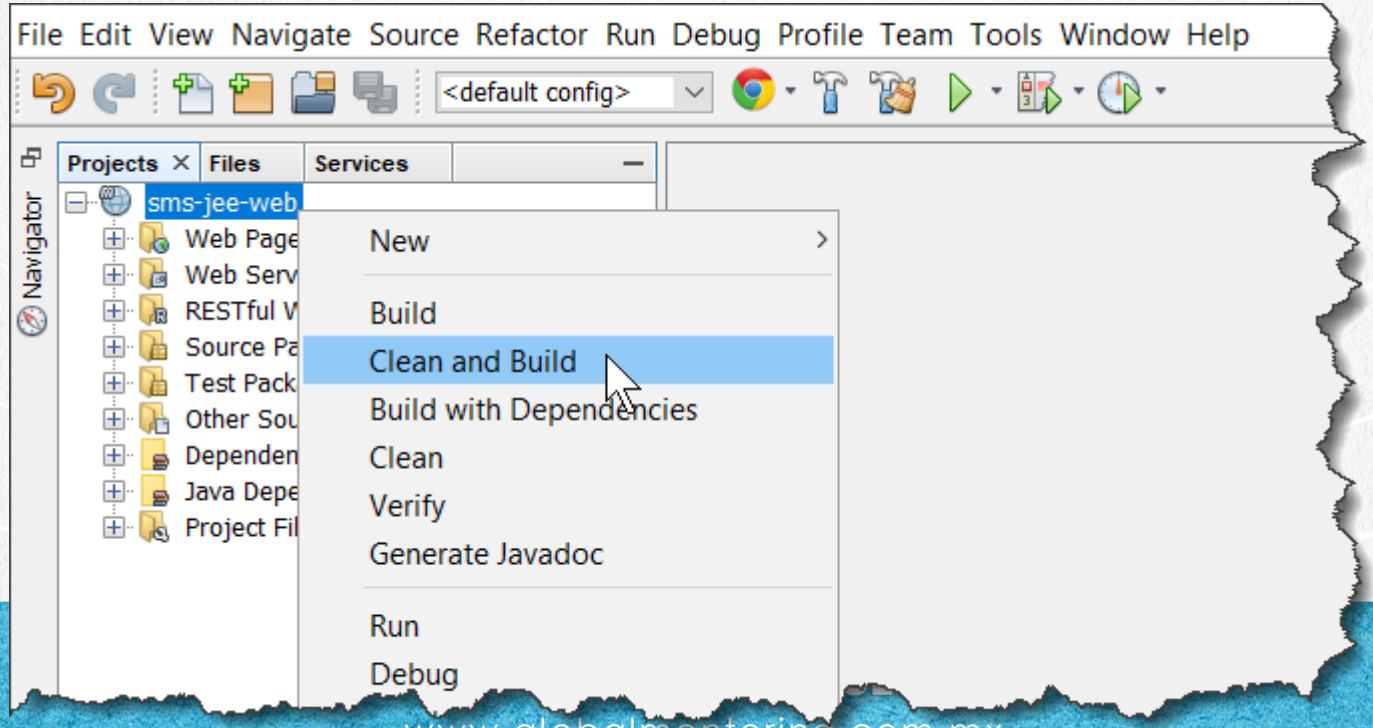
1. OPEN THE PROJECT

In case we do not have open the sms-jee-web project we open it:



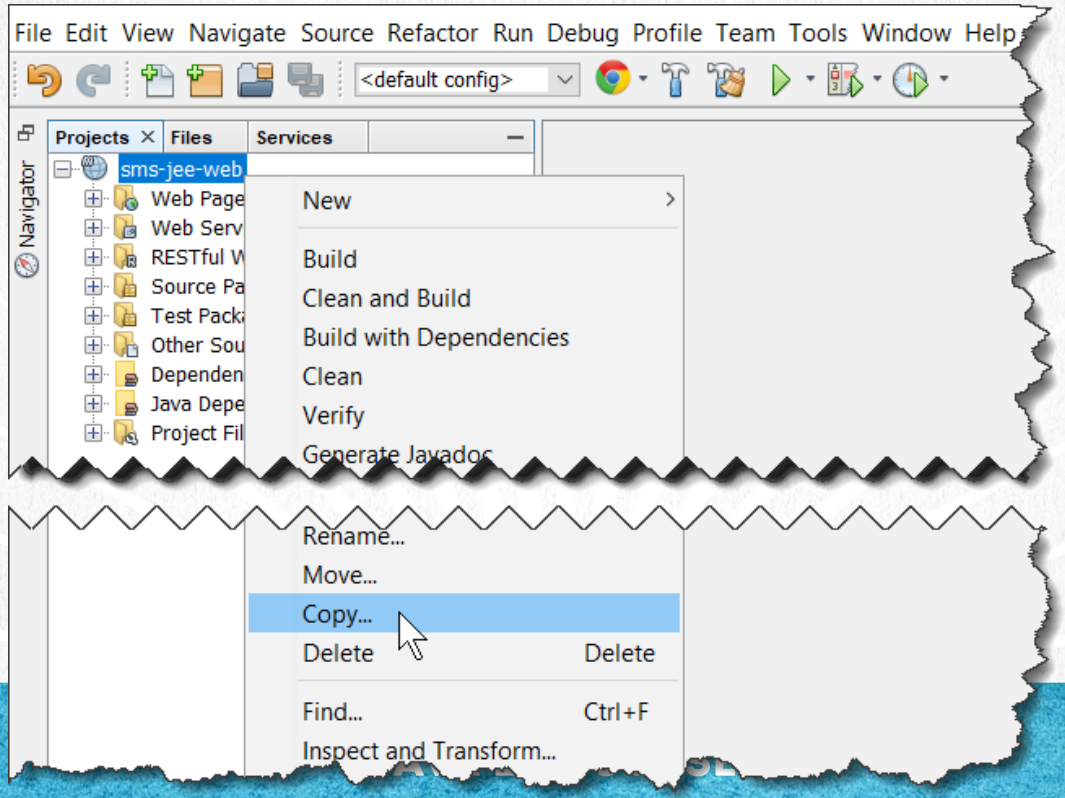
1. OPEN THE PROJECT

We wait for you to fully load the project. In case the project shows errors, we make a Clean & Build so that all the files are shown, this step is optional:



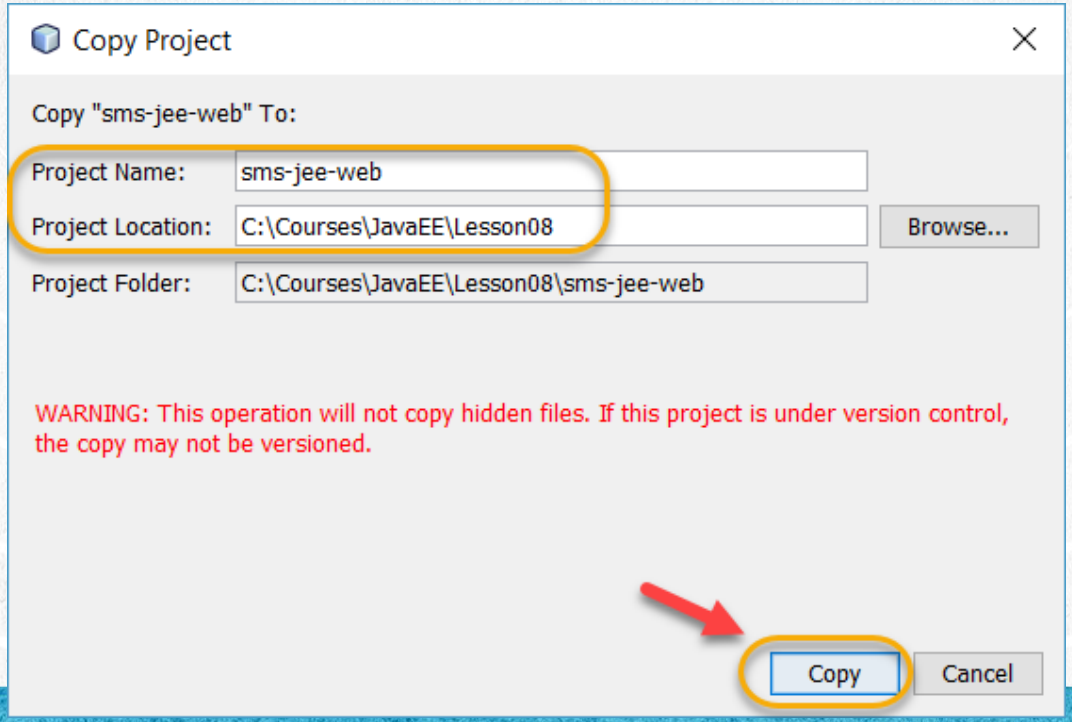
2. COPY THE PROJECT

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



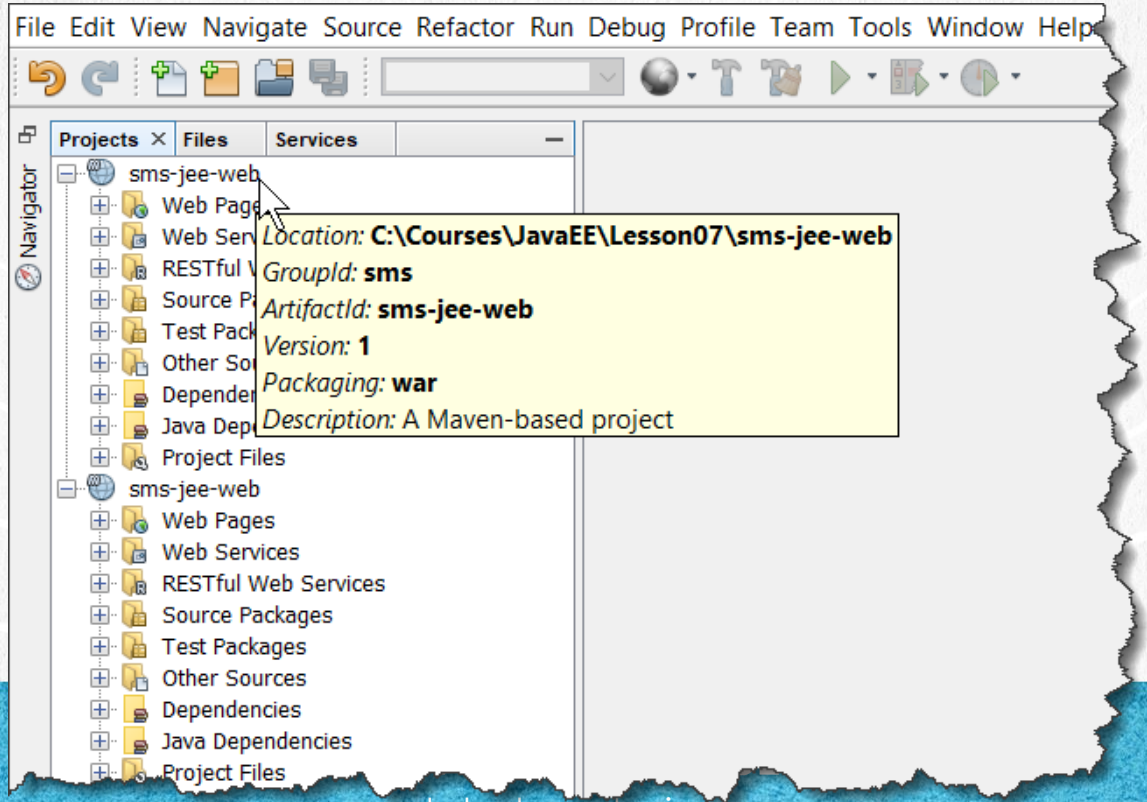
2. COPY THE PROJECT

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



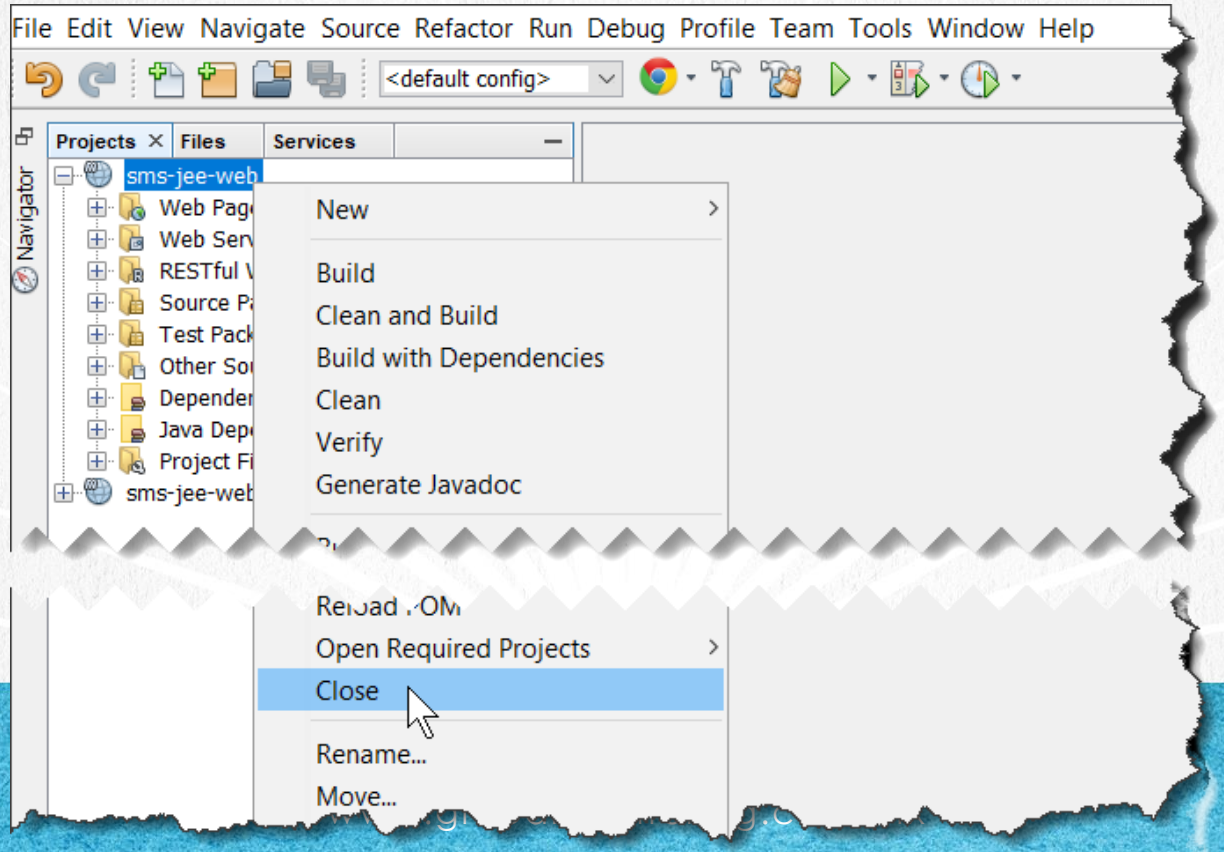
3. CLOSE THE PREVIOUS PROJECT

We closed the previous project, we identified it by positioning ourselves on the project:



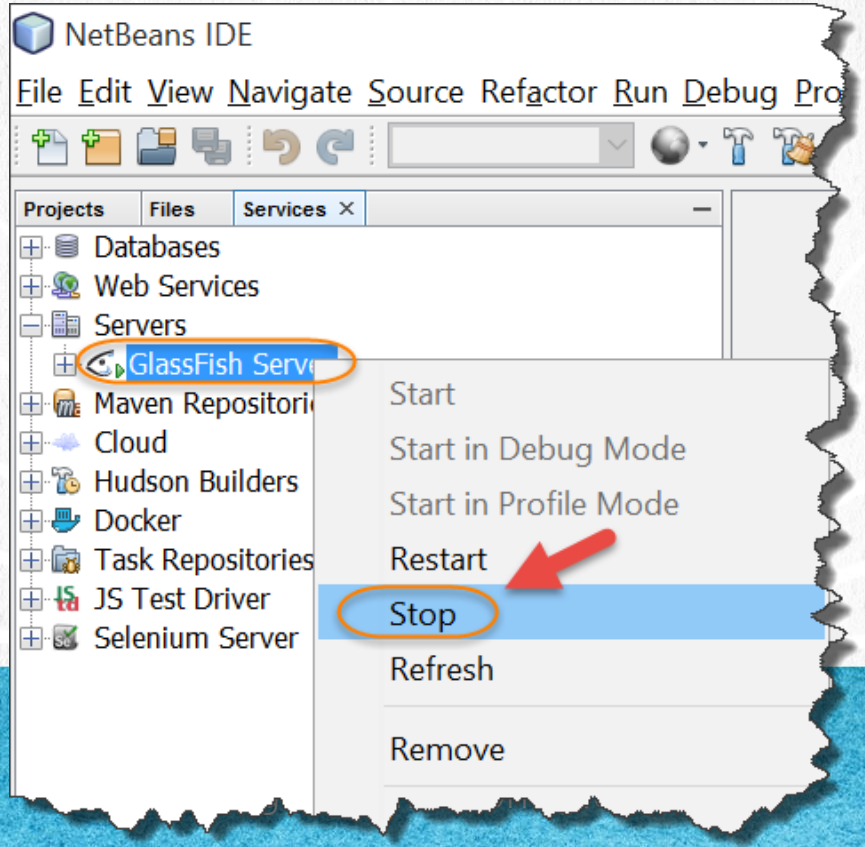
3. CLOSE THE PROJECT

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



4. STOP GLASSFISH

We stop the Glassfish server:



5. MODIFY A JAVA CLASS

We add the following annotations to the EJB of PersonServiceImpl.java in the declaration of the class:

```
@DeclareRoles({ "ROLE_ADMIN", "ROLE_USER" })
@RolesAllowed({ "ROLE_ADMIN", "ROLE_USER" })
```

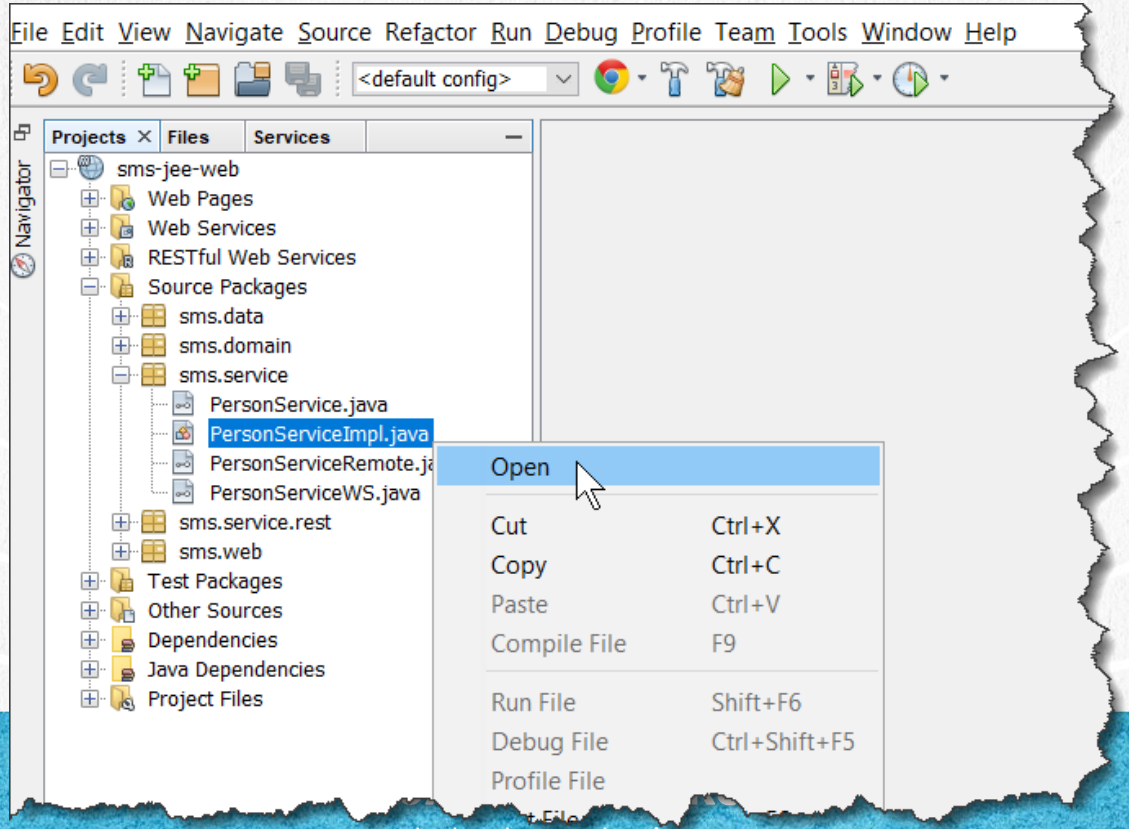
In addition, to the deletePerson method, we add the annotation :

```
@RolesAllowed("ROLE_ADMIN")
```

Let's see how our class is already modified:

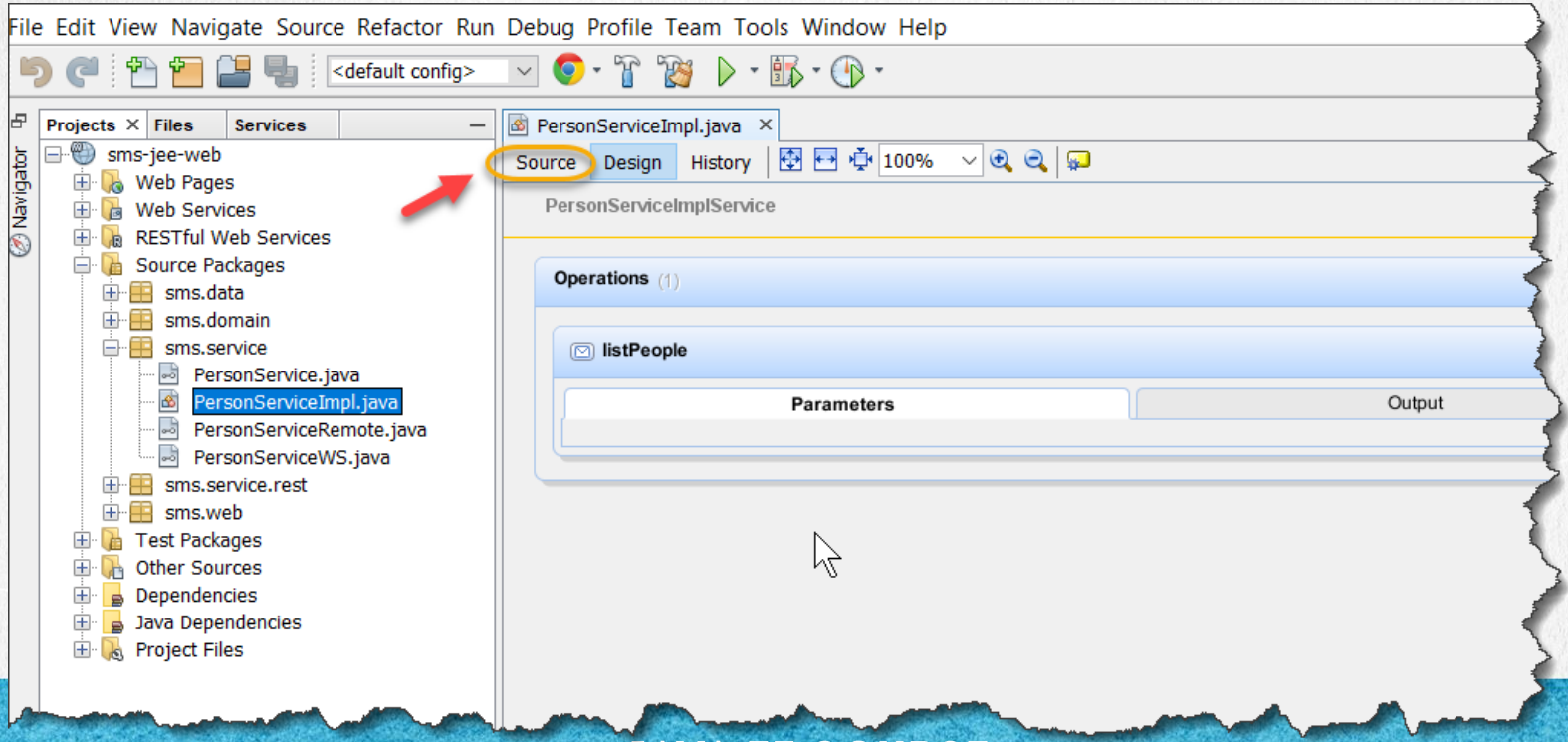
5. MODIFY A JAVA CLASS

Modify the PersonServiceImpl.java class :



5. MODIFY A JAVA CLASS

Modify the PersonServiceImpl.java class :



5. MODIFY THE FILE

PersonServiceImpl.java:

Click to download

```
package sms.service;

import java.util.List;
import javax.annotation.Resource;
import javax.annotation.security.DeclareRoles;
import javax.annotation.security.RolesAllowed;
import javax.ejb.SessionContext;
import javax.ejb.Stateless;
import javax.inject.Inject;
import javax.jws.WebService;
import sms.data.PersonDao;
import sms.domain.Person;

@Stateless
@WebService(endpointInterface = "sms.service.PersonServiceWS")
@DeclareRoles({ "ROLE_ADMIN", "ROLE_USER" })
@RolesAllowed({ "ROLE_ADMIN", "ROLE_USER" })
public class PersonServiceImpl implements PersonServiceRemote, PersonService, PersonServiceWS {

    @Resource
    private SessionContext context;

    @Inject
    private PersonDao personDao;

    @Override
    public List<Person> listPeople() {
        return personDao.findAllPeople();
    }
```

5. MODIFY THE FILE

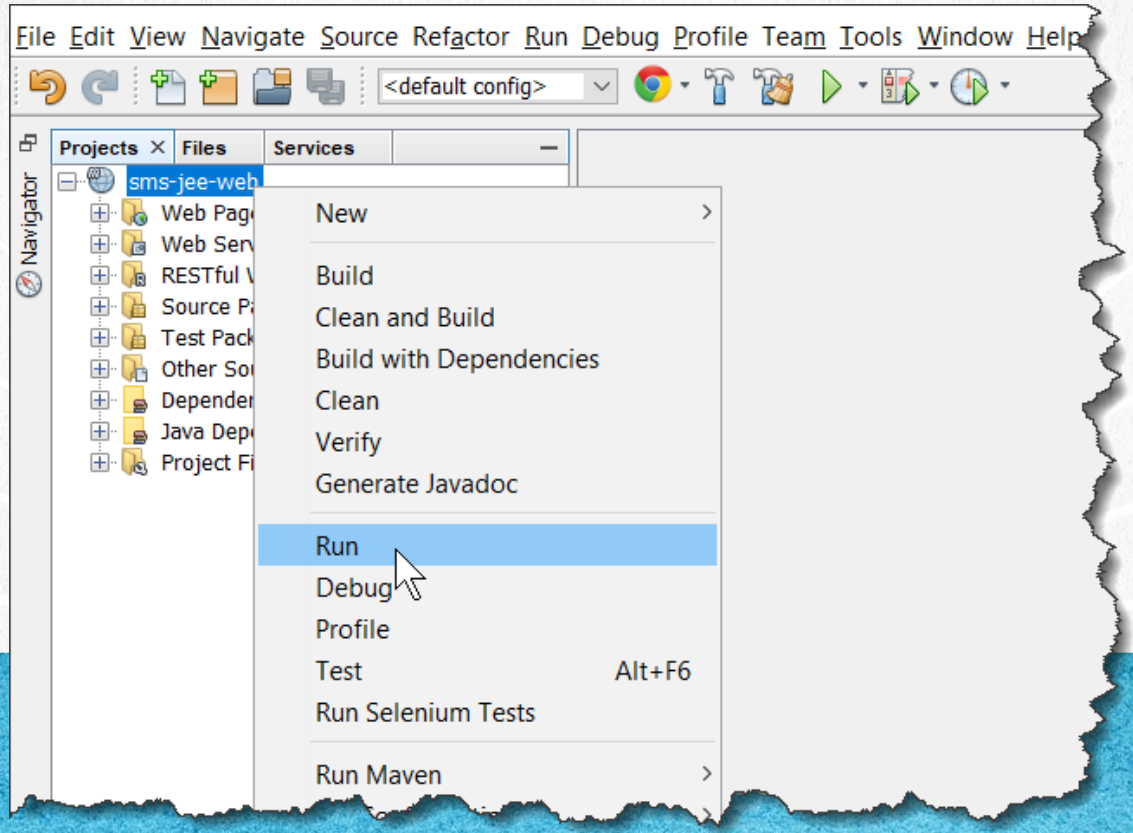
PersonServiceImpl.java:

Click to download

```
    @Override
    public Person findPerson(Person person) {
        return personDao.findPerson(person);
    }

    @Override
    public void addPerson(Person person) {
        personDao.insertPerson(person);
    }

    @Override
    public void modifyPerson(Person person) {
        try {
            personDao.updatePerson(person);
        } catch (Throwable t) {
            context.setRollbackOnly();
            t.printStackTrace(System.out);
        }
    }

    @Override
    @RolesAllowed("ROLE_ADMIN")
    public void deletePerson(Person person) {
        personDao.deletePerson(person);
    }
}
```

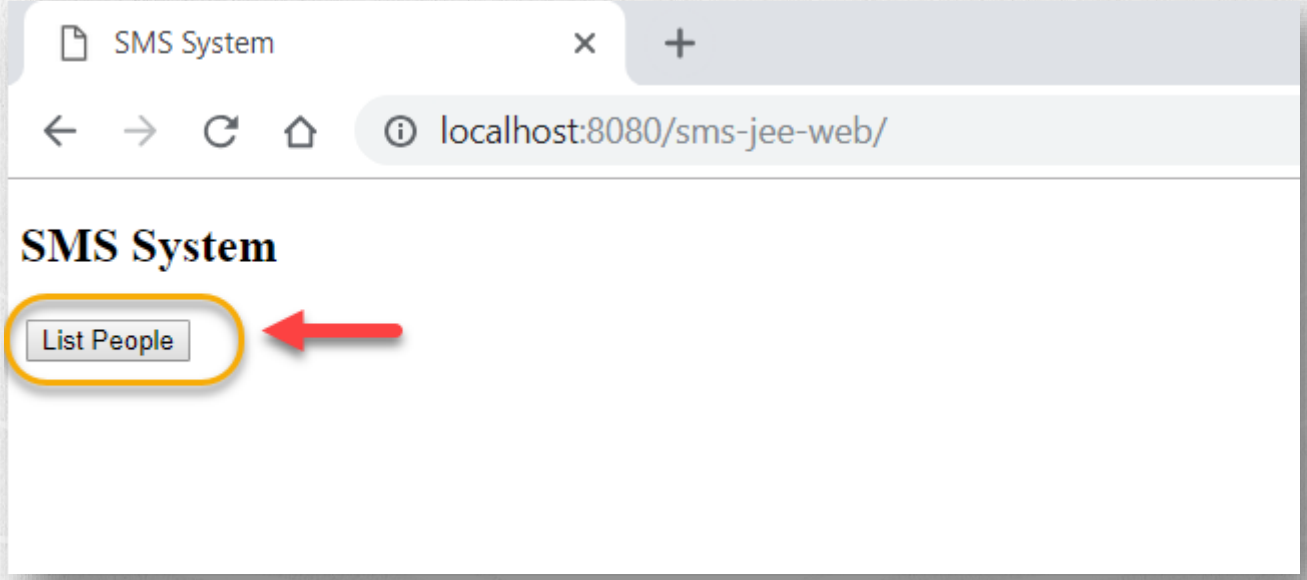
6. VERIFY THAT CLIENTS FAIL

Once the annotation is added and check that we no longer have access to the EJB methods.
Eg This is the result from the Web client:



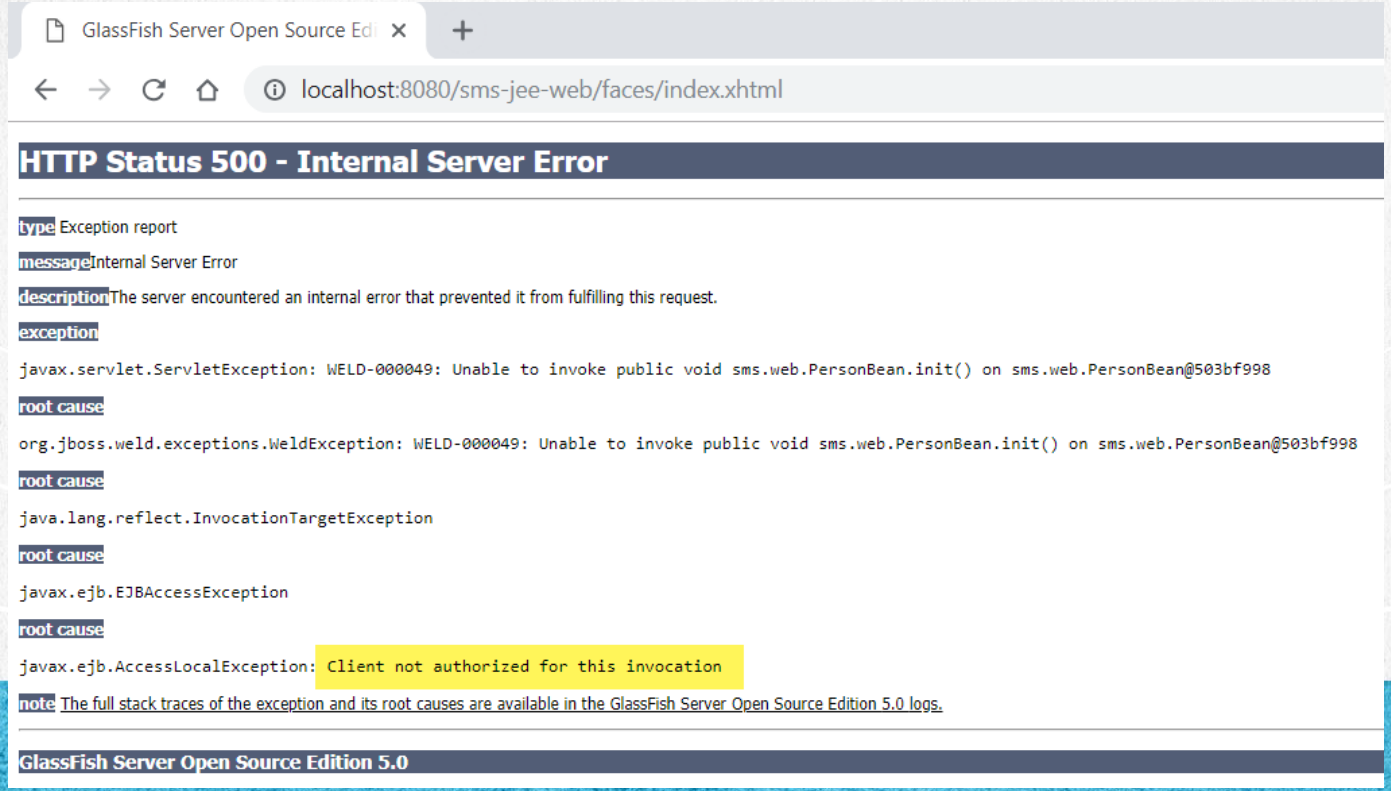
6. VERIFY THAT CLIENTS FAIL

The Web client and the other clients fail to access the listPeople method or any other method of the EJB:



6. VERIFY THAT CLIENTS FAIL

The Web client and the other clients fail to access the listPeople method or any other method of the EJB:



EXERCISE CONCLUSION

With this exercise we have added security to our service layer, which is precisely the one that exposes the information of our Java Enterprise system to Web clients, and Web Services, or any other.

Let's see in the following exercises how to modify the clients to be able to access the services offered by the EJB but now adding the security concept (authorization and authentication).

ONLINE COURSE

JAVA EE
JAKARTA EE

By: Eng. Ubaldo Acosta