

- TD 1 - First beans

The purpose of this lab is the development of relatively simple Enterprise Java Beans as well as their deployment and use by different types of clients. To perform all the exercises we will work with the IDE NetBeans and the application server Java EE6 GlassFish. Information about their installation are available on the course page.

We start from the example seen in class (you can get the sources available on the course page) to develop and deploy an application that provides very basic services, and to develop different clients accessing the application. We build a new project called, for example, `SimpleApplication` de type Java Enterprise (Java with Ant) with EJB Module and Web Application Module (the latter is required for web clients).

1. Prepare the web client

First, create a JSF page `index.xhtml` (in the Web Pages of the `SimpleApplication-war` project); you can then « Clean and build » the application and then « Run » the file `index.xhtml` (before running you could « Deploy » the application - you can use the GlassFish console (<http://localhost:4848>) or the « Services » tab in NetBeans to see the applications deployed on server.). You can then replace the content of this latter file with the one provided on the course page and after doing « Clean and build » and « Deploy », reload the page `index.xhtml` - the page should correctly printed but pushing the button would lead to an error. To correct this you should create a new JSF CDI Bean (in the Source Packages of the `SimpleApplication-war` project) of scope Request called `Hello.java` and a new JSF web page `hello.xhtml` (use for both the code provided on the web page). You can now reload the page `index.xhtml` and test it.

You should be aware that with a standard configuration (`web.xml` in Configuration Files) the JSF pages created in Web Pages are accessible at the address `localhost:8080/SimpleApplication-war/faces/*.xhtml` (the address can be changed by creating a Deployment Descriptor `application.xml` for `SimpleApplication` project).

2.The first bean(s)

We add to `SimpleApplication-ejb` a new session bean called `ComplimentBean.java` (Stateless, Local (or no) interface, package `firststeps`). A `ComplimentBeanLocal.java` interface is normally created automatically by NetBeans when the bean `ComplimentBean.java` is created - in this case you should do some refactoring to rename it to `Compliment.java`. We can now write the code for a method that returns the string « *You're the best, James Bond* » (the name could be a parameter), inspired by the example seen in class. We can inject this bean in the CDI Bean `Hello.java` and add a new method which uses the EJB bean `ComplimentBean.java` to compliment; a new JSF web page `greet.xhtml` should be used as target for the `greet` method of `Hello.java`.

You can now create other EJB beans that can be injected and used from `ComplimentBean`. We could for example imagine one which generates a greeting message depending on the language specified as a parameter.

3. Web client(s) for the bean

We will use now the `ComplimentBean` from a servlet called `ControllerServlet` placed in the `Sources Packages` of the project `SimplestWEA-war` (we assume that `Web Application Module` has been selected at the creation of the project). You can use the servlet available on the course website to develop your own. The pages of the application are available at `localhost:8080/SimplestWEA-war` (or the address specified in the `Deployment Descriptor application.xml`) and the entry points are specified in `web.xml`: for example, you can use `http://localhost:8080/SimplestWEA-war/Hello?name=Toto` with the servlet provided to compliment *Toto*. Alternatively, you can use the provided file `index.jsp` to develop web pages (in `Web Pages`) and therefore a more user-friendly and interactive interface (accessible at `localhost:8080/SimpleApplication-war/faces/index.jsp`).