

TD 1

Premiers beans

Le but de ce TD est d'expérimenter le développement des Entreprise Java Beans relativement simples ainsi que leur déploiement et leur utilisation par différents types de clients. Pour réaliser l'ensemble des exercices nous allons travailler avec l'IDE NetBeans et le serveur d'applications GlassFish. Des informations concernant leur installation sont disponibles sur la page du cours.

1. L'application

On partira de l'exemple vu en cours - vous pouvez récupérer les sources disponibles sur la page du cours - pour développer et déployer une application fournissant des services très simples et pour développer différents clients accédant à l'application. On construit un nouveau projet appelé, par exemple, [SimplestWEA](#) (de type JavaEE-EnterpriseApplication) avec EJB Module et Web Application Module (ce dernier est nécessaire pour les clients web).

2. Le premier bean

On ajoute dans le module [SimplestWEA-ejb](#) un nouveau Session Bean appelé [ComplimentBean](#) (Stateless, interface Local, package [firststeps](#)). Une interface [ComplimentBeanLocal](#) est normalement créée automatiquement par NetBeans à la création du bean [ComplimentBean](#) - dans ce cas il faut faire du refactoring pour la renommer en [Compliment](#). On peut maintenant écrire le code d'une méthode qui retourne la chaîne de caractères "*You're the best, James Bond*" (le nom pourrait être un paramètre) en s'inspirant de l'exemple vu en cours. On déploie l'application et on peut utiliser la console GlassFish (<http://localhost:4848>) ou l'onglet «Services» (item Servers) dans NetBeans pour consulter les applications déployées sur le serveur.

3. Client JSF pour le bean

On utilisera les beans développés dans le framework JSF. Vous pouvez utiliser dans le projet [SimplestWEA-war](#) les pages JSF ([index.xhtml](#)) et le managed bean ([SayBestBean.java](#)) fournis sur la page du cours comme point de départ pour proposer des pages interactives. Attention, avec une configuration standard ([web.xml](#) dans Configuration Files) les pages JSF créées dans Web Pages sont accessibles à l'adresse localhost:8080/SimplestWEA-war/faces/*.xhtml (l'adresse peut être modifiée en créant un Deployment Descriptor [application.xml](#) pour le projet [SimplestWEA](#)).

4. Clients Servlet et JSP pour le bean (optionnel)

On utilisera maintenant le [ComplimentBean](#) depuis une servlet appelée [ControllerServlet](#) dans les Sources Packages du projet [SimplestWEA-war](#) (on suppose que Web Application Module a été sélectionné à la création du projet). Vous pouvez vous inspirer de la servlet ([DispatchServlet](#)) disponible sur la page du cours pour développer la votre. Les pages de l'application sont accessibles à l'adresse localhost:8080/SimplestWEA-war (ou l'adresse spécifiée dans le Deployment Descriptor [application.xml](#)): par exemple, vous pouvez utiliser l'adresse <http://localhost:8080/SimplestWEA-war/Salut?name=Toto> avec la servlet fournie pour complimenter *Toto*. Alternativement, vous pouvez partir du fichier [index.jsp](#) fourni pour développer des pages web (dans Web Pages) et donc une interface plus conviviale et interactive.

5. Client Java (optionnel)

On veut développer maintenant un client (projet JavaEE-EnterpriseApplicationClient) appelé [SimplestWEA-Client](#) pour notre application. On développera dans un premier temps un nouveau Session Bean appelé [HelloWorldBean](#) (Stateless, package [firststeps](#)) avec une interface Remote pour un accès depuis des clients externes. Ce bean permettra de dire *Bonjour* au lieu de *You're the best*. On utilisera pour cette version une architecture qui sépare l'implantation et la spécification des beans - pour cela on peut créer une librairie [SimplestWEA-library](#) (nouveau projet Java-Java Class Library) avec un package [firststeps](#) et définir à cet endroit l'interface correspondante [HelloWorld](#) (une interface [HelloWorldBeanLocal](#) est normalement créée automatiquement par NetBeans à la création du bean [HelloWorldBean](#) - dans ce cas il faut faire du refactoring pour la renommer en [HelloWorld](#) et éventuellement la déplacer dans la librairie).

On peut écrire le code en s'inspirant encore une fois de l'exemple du cours - du refactoring pourrait être nécessaire et la librairie [SimplestWEA-library](#) (situé dans le répertoire `dist`) doit être ajoutée dans les propriétés du projet. Il faudra exécuter le client (en s'assurant que la `Mainclass` du projet est correcte) et vérifier que les messages correspondants sont affichés dans les consoles du serveur et du client. Il faudra également proposer et exécuter un client qui utilise un lookup explicite pour trouver le bean [HelloWorld](#) (et non de l'injection). On peut de nouveau utiliser la console `GlassFish` pour consulter les applications déployées sur les serveur.

Vous pouvez ensuite compléter l'application en y ajoutant d'autres méthodes (par exemple, [String greetingsMessage\(String langue, String name\)](#) qui retourne une chaîne de caractères représentant un message en [langue](#) pour [name](#)) dans l'interface [HelloWorld](#) et éventuellement un autre bean [DireBonjourBean](#) (local) appelé par le bean [HelloWorldBean](#).