

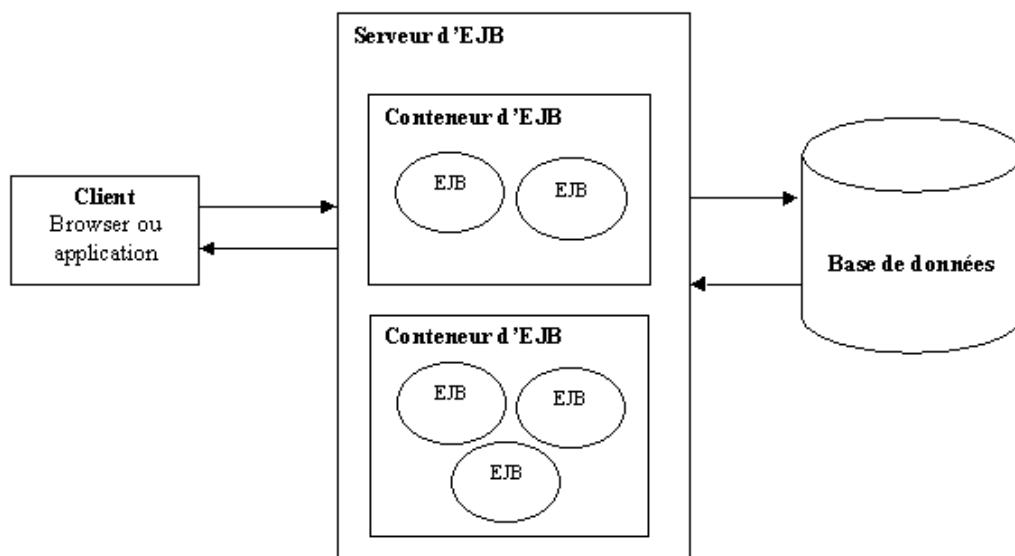
Compte rendu TD2

Avant de commencer

Dans un premier temps, il faut suivre les instructions du *Quick Start* sur <http://tomEE.apache.org/tomee-and-eclipse.html> afin d'installer un serveur TomCat et un conteneur d'applications TomEE. Il est nécessaire de l'installer car il fournit un certain nombre de services de gestion des EJB. Le serveur fonctionne correctement sur ma machine.

Projet Student

Le but du TD est de comprendre le fonctionnement d'une architecture 3-tiers avec un client, un serveur d'EJB et une base de données.



Ici, le serveur d'EJB est TomCat 7.0 et le conteneur d'EJB est TomEE.

Comme spécifié sur le diagramme UML, le *Dynamic Web Project* est constitué d'une interface (StudentEJBRemote) et de trois classes dont une classe Main qui permet de faire des tests.

L'objectif est de comprendre la persistance des données sous JEE. Pour cela, on utilise un fichier de configuration xml *persistence.xml*.

Le projet est disponible sur mon dépôt git : <https://github.com/bastiengot/Student>.

Questions

Question 1

Les beans de session avec état (*stateless* ou *stateful*) sont utilisés lorsque plusieurs échanges avec un même client sont effectués. Les session beans *stateful* conservent leur état entre deux appels au contraire des session beans *stateless*.

Ici, nous n'avons pas besoin d'accéder aux états des précédentes connections entre le client et l'application puisque les données sont stockées dans la base de données et l'historique des actions n'est pas nécessaire. De plus, il n'y a pas d'attributs donc peu importe que ce soit la même instance de la classe ou pas. Le bean *stateless* est donc adapté à la situation.

Question 2

L'interface *remote* nous permet de ne pas oublier de méthodes essentielles telles que *delete*, *add*, *find*, *findbyID* lorsque nous créons le bean entity. Si quelqu'un d'autre veut utiliser cette interface à travers un réseau, l'annotation *@Remote* est de rigueur, sinon l'annotation *@Local* suffit.

Question 3

Les fichiers à télécharger sur le serveur distant dont les fichiers de configuration *web.xml* et *persistence.xml*. Les fichiers relatifs à l'interface utilisateur doivent aussi être uploadés sur ce serveur. Enfin, il faut télécharger l'EJB qui gère la transaction et le stockage des données.

Question 4

L'application web doit par définition pouvoir être utilisée sur plusieurs machines différentes. L'atout principal des EJB est la persistance des données, peu importe la machine sur laquelle tourne l'application, il suffit qu'elle accède au serveur sur lequel elle est installé. Les données sont donc stockées grâce aux EJB et grâce au bean de session *stateless*, le client peut échanger avec le serveur sur plusieurs sessions successives.

Conclusion

Ce TD m'a permis d'apprendre l'organisation d'un *Dynamic Web Project* sur eclipse et de créer ma première application. Malgré beaucoup de bugs compliqués à résoudre – notamment quelques heures passées à essayer de lancer mon serveur (le bug venait d'une ligne inutile dans un de mes fichiers source - j'ai réussi un des objectifs du TD qui était de développer

une application utilisant des EJB. Je pense que je pourrais encore faire évoluer mon projet en développant une UI et en reliant mon projet à une base de données.