1. **Comparaisons des plateformes**

https://docs.google.com/document/d/1ZX1hzPt95iXcyJN7ufnV3gH0EDS6nQO0diU1fXikEqA/edit

1. **Architecture sur Heroku**
   1. **Via le plugin Eclipse**

* Spring MVC : similaire aux projets STS
* Play :
  + App
    - Controller
    - Model
    - Views
  + Conf
  + Public
    - Image
    - Stylesheet
  + Test
* Jetty Servlet :
  + Src/main/java
  + Src
    - Main
      * Webapp
        + Web-inf

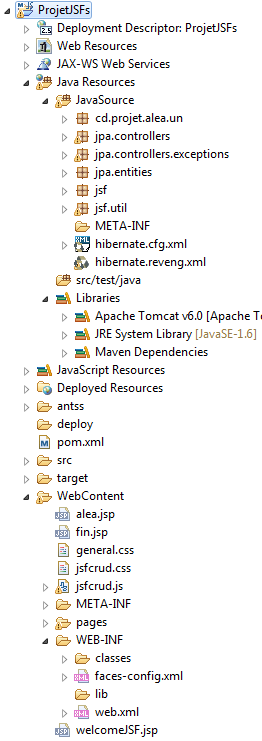
Web.xml

* + - * + Index.html
  + Target
  + Pom.xml
  1. **Via le Dashboard : Dossier vide au départ**

1. **Déploiement sur différentes plateformes PaaS**

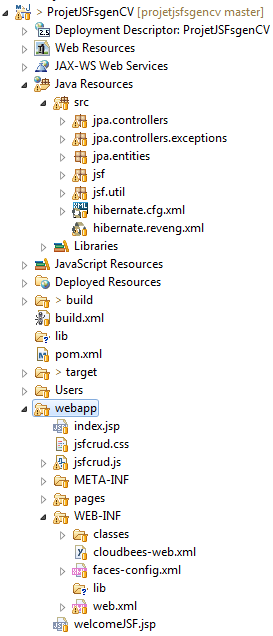
Une application de type JSF / Hibernate a servi dans le but de voir la différence entre les architectures des applications de ce type sur les différentes PaaS.

* 1. **Sur CloudFoundry** : Architecture de l’application



Créer un projet « Dynamic Web Project » **version 2.5 (CloudFoundry ne supporte pas encore la version 3.0 des projets web dynamique)**. Sur cette plateforme, nous avons un serveur Tomcat 6.0.35. Le problème lors du déploiement est la version de l’API « *el* ». En effet, Tomcat 6.0.35 possède déjà dans ses librairies un fichier el-api.jar. Une version différente de cet API lance des erreurs de versions (méthodes introuvable, etc.). Avec cette plateforme, le développeur peut même se passer d’un fichier pom.xml. Il lui suffit juste d’un fichier .war ou de glisser-déposer sur le serveur CloudFoundry. L’accès à la base de données est très simple, qu’il s’agisse d’une connexion distante ou non.

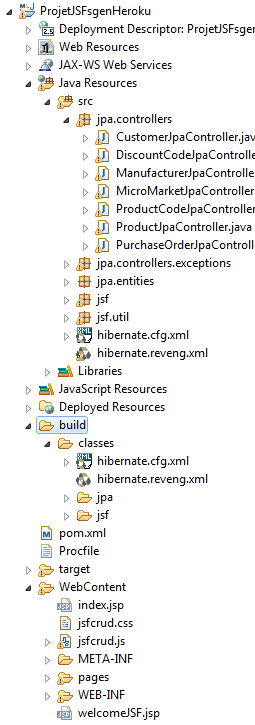
* 1. **Sur CloudBees :** Architecture



Le déploiement de l’application existante créé des problèmes de versions de libraires (**el-api.jar** au début présent dans le répertoire WEB-INF/lib). Il faut donc supprimer ce fichier .jar de la librairie de l’application pour que le déploiement passe, même si le projet nous signale une erreur de classe non trouvée. Après la copie des sources, il a fallu que certaines classes implémentent l’interface **Serialisable** à la demande du serveur d’application (sorties console). Exemple :

java.lang.IllegalArgumentException: setAttribute: Non-serializable attribute purchaseOrderJpa

* 1. Bonne méthode : créer un projet CloudBees avec le plugin Eclipse. Ajouter les natures JSF, etc. Faire attention, car dans le projet généré, le dossier web s’appelle « webapp » et non « webContent ».
  2. J’ai rencontré des problèmes de connexion à la base de données MySQL de CloudBees dû à des erreurs dans la documentation de CloudBees. Dans le fichier hibernate.cfg.xml, la documentation donne le chemin *«*java:com/env/jdbc/mydb » au lieu de « java:**/**com/env/jdbc/mydb», ce qui empêchait de Hibernate de trouver la source de données. Beaucoup de temps perdu pour ce « / » que j’ai pu résoudre via un autre site.
  3. Pour relier la DB à l’application CloudBees, la commande « bees app :bind » était également nécessaire. Sauf que j’utilisai la version 0.35 du SDK qui ne reconnaissait pas cette commande. (Temps perdu également)
  4. **Déploiement sur Heroku** : Architecture du projet.



Avec le plugin Heroku, créer une application de type « *Embedded Jetty-Servlet application »*. Heroku ne possède pas de conteneur d’application. Il se sert d’un fichier Procfile pour lancer les applications. A chaque type d’application correspond une commande avec ses paramètres à exécuter. Dans le cas de notre application, nous devons utiliser Jetty et lui indiquer le port ainsi que .war. Le contenu du fichier Procfile est :

web: java $JAVA\_OPTS -jar target/dependency/jetty-runner.jar --port $PORT target/\*.war

La connexion distante à la base de données sur Heroku nécessite l’utilisation d’une connexion SSL.

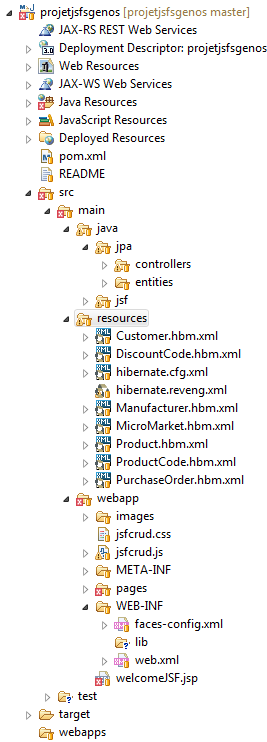
[*https://devcenter.heroku.com/articles/connecting-to-relational-databases-on-heroku-with-java#using-the-in-plain-jdbc*](https://devcenter.heroku.com/articles/connecting-to-relational-databases-on-heroku-with-java#using-the-in-plain-jdbc)

Vérifier dans le fichier pom.xml, que la valeur de Target est bien « 1.6 », sinon erreur de compilation lors du déploiement sur Heroku.

L’interface web *herokupostgres* est très pratique, car il permet d’avoir des informations de connexion pour différents langages en même temps.

**Remarque** : Dans l’interface web *herokupostgres*, le Dashboard affiche « **Tables 0** » alors qu’il y a bien des tables et des données dans la base.

* 1. **Déploiement sur OpenShift : Architecture du projet**



Pour créer le projet, utiliser le plugin OpenShift installé inclut dans JBoss Tool, le type d’application est jbossews-2-0 qui correspond à une application qui sera déployée sur Tomcat 7. Dans le fichier pom.xml, faire attention avec la version javax.el-api.jar utilisée. Par rapport aux applications précédemment déployée sur les autres plateformes, il a fallu changer l’emplacement du fichier hibernate.cfg.xml (ainsi que des autres fichiers). Initialement, le fichier hibernate.cfg.xml se trouvait sous le répertoire « *java* ». A cet instant, nous obtenons une erreur : « /hibernate.cfg.xml not found ». Le placer sous le répertoire *src/main/resources* résout le problème. Un problème persiste à ce jour : l’utilisation des variables d’environnement de connexion à la base de données dans le fichier hibernate.cfg.xml. Le chemin donné par le résultat de la commande rhc app show projetjsfsgenos ne permet pas d’accéder à la base de données. Les données sont donc entrées en dur dans le fichier hibernate.cfg.xml. L’interface PhpMyAdmin permet de visualiser les données.

**NB** : Tourne sur Apache Tomcat.

1. **Prix**

Les offres sont très diversifiées en termes de fonctionnalités, services disponibles, capacité maximal de stockage, etc. En outre, certains fournisseurs définissent leurs propres unités de mesures (Gear, App-cell, Dyno, etc.).

|  |  |  |
| --- | --- | --- |
|  | Mémoire | MySQL |
| OpenShift | $0.05/gear/heure  Gear=512Mo RAM &1Go disque | $1/mois 🡪1Go |
| CloudBees | $0.028/heure/app-cell  App-cell=128Mo | **$25/mois 🡪 5Go** |
| Heroku | $0.05/dyno/heure  Dyno=512Mo RAM & 1xCPU | $9.99/mois 🡪 1Go |
| CloudFoundry  (non extensible) | 2Go free | 128Mo free |
| Azure | €0.015/heure  768Mo & 1GHz RAM & 20Go disque | SQL  €7,44/mois 🡪1Go |