École Polytechnique de l'Université de Tours
Département Informatique
Cedric Gatay c.gatay@code-troopers.com
Florian Chauveau f.chauveau@code-troopers.com

TP Java EE - 1

Consignes :
Vous êtes libres d'utiliser l'environnement de développement que vous désirez.
Dans la série de TP, vous allez réaliser un livre d'or en partant d'un squelette pré-rempli. L'issue de cet ensemble de TP donnera lieu à une note, donc soyez sérieux lors de la réalisation. Les sujets sont conçus de façon à ce que peu de travail intermédiaire soit nécessaire.
Note sur le code produit :
Cet énoncé ne donne pas de nom aux classes et aux actions de façon volontaire pour que vous soyez libre de faire comme bon vous semble.
Cependant, lorsque vous programmez, pensez à la relecture future de ce que vous produisez, que ce soit pour vous ou pour un autre. Ainsi, aérez votre code, nommez correctement vos variables, méthodes et classes.

Dans ce sujet de TP, le thème abordé est la découverte de l'environnement standard JavaEE de définition de construction de pages HTML : Java Server Faces.

Partie 1 : Récupération des sources

2 options s'offrent à vous pour récuperer le code source initial que vous compléterez au cours des 3 TPs.

- Option 1 : Checkout avec Git
  - Le code source pour le tp est disponible sur https://bitbucket.org/cgatay/tpjavaee2018/src
  - Pour en récupérer une copie en utilisant git vous allez utiliser la commande git clone qui permet d'obtenir une copie de toute la hiérarchie d'un projet
  - La commande à effectuer est donc :
    git clone https://bitbucket.org/cgatay/tpjavaee2018.git
- Option 2 : Téléchargement d'une archive
  - Si vous n'avez pas git ou que vous ne voulez pas l'utiliser vous pouvez simplement télécharger l'archive du code source pour les TPs à l'adresse suivante : https://bitbucket.org/cgatay/tpjavaee2018/downloads/
  - Ensuite vous n'avez qu'à décompresser le résultat dans le répertoire de votre choix.

Partie 2 : Lancement du projet

Le projet nécessite le JDK 8 car il est écrit en Java 8, vous pouvez le récupérer sur le site suivant : http://www.oracle.com/technetwork/java/javase/downloads/index.html

Pour cela, nous allons utiliser Maven :

> *Maven est un outil s'exécutant en ligne de commande, vous aurez donc besoin de lancer le terminal de l'environnement dans lequel vous vous trouvez pour l'utiliser. L'exécutable est nommé "mvn".*
>
> *Pour installer Apache Maven, il vous faudra le télécharger sur https://maven.apache.org/download.cgi (utiliser au moins la versions 3.3.9)*
>
> *Maven est un outil standard dans le monde JavaEE permettant de gérer l'intégralité des phases de construction d'un projet. Cet outil répond à deux problématiques principales :*
> - *gérer les dépendances externes,*
> - *assurer la reproductibilité de la construction*
>
> *Les projets JavaEE sont souvent des projets de plusieurs milliers de lignes de codes reposant sur de multiples projets externes (génération de PDF, manipulation de fichier Office, ...). Maven offre, via une syntaxe simple, un moyen de décrire les dépendances.*
>
> *L'autre point important est d'assurer une reproductibilité des constructions. Aucun projet Java EE n'est assuré par un seul développeur (encore moins sur une seule machine). Il est donc important d'avoir un moyen facile de construire le projet peu importe la configuration et l'OS du développeur.*
> *Maven est complètement indépendant de l'emplacement du répertoire home de l'utilisateur et de l'IDE utilisé. Ce point permet en outre d'outiller la construction en l'automatisant (ainsi que les tests unitaires assurant la qualité du logiciel produit).*

Vous pourrez valider la bonne installation de tous ces éléments en vous assurant qu'ils soient disponibles en ligne de commande avec les commandes suivantes:

```
java -version
mvn -version
git --version
```

Dans un terminal rendez-vous dans le répertoire dans lequel vous avez votre code source.
À la racine du projet doit se trouver un fichier pom.xml c'est le fichier de description de projet qu'utilise maven, vous devez donc vous trouver précisément dans ce répertoire pour exécuter les commandes maven

La commande qui nous intéresse est la suivante : `mvn clean package wildfly:run`
Celle-ci va nous permettre de compiler les sources, les packager les démarrer un conteneur d'application avec le package généré et donc d'avoir l'application démarrée.

Partie 3 : Accès à l'application

Ouvrez un navigateur web et rendez-vous à l'adresse http://localhost:8080/tpjavaee/login.jsf
Voilà ! Votre application fonctionne, maintenant vous pouvez commencer à travailler.

Partie 4 : IDE

Ouvrez votre IDE préféré et importez-y votre projet afin de pouvoir travailler :

- Netbeans
- Eclipse
- IntelliJ IDEA

Partie 5 : Au boulot !
La page de login que vous voyez à l'adresse http://localhost:8080/tpjavaee/login.jsf
est générée a partir du fichier login.xhtml.
Vous devez transformer le formulaire actuel qui est uniquement du html pur en un formulaire qui va utiliser des composants JSF et faire appel aux classes java correspondantes.

Page d'aide utile pour démarrer ce TP : http://www.tutorialspoint.com/jsf/jsf_basic_tags.htm
La classe qui possède la logique métier pour la connexion est **AuthenticationService.class**

Partie 6 : Et ensuite ?
Maintenant que vous avez corrigé le formulaire de login, vous pouvez vous authentifier et vous accédez à une magnifique page qui vous dit que vous l'êtes.
Allez plus loin et redirigez maintenant l'utilisateur après connexion sur la page affichant la liste des commentaires de votre "Livre d'Or".

Partie 7 : Une vraie connexion
Le projet JPA est déjà correctement connecté et utilise une base en mémoire (qui est supprimée à chaque arrêt du container et recréée à chaque démarrage de celui ci).
Vous pouvez d'ailleurs voir dans le fichier persistence.xml certaines options utilisées par l'application pour se connecter à la base en mémoire.

C’est la classe **AuthenticationService** qui gère le contrôle du couple login/mot de passe pour valider une connexion.
Dans cette classe devez utiliser les services fournis par la classe UserDAO qui s’occupe l’accès à la base pour tout ce qui concerne les User.

> Le terme DAO est fréquemment utilisé pour représenter ce genre de classe dans les projets JavaEE, il signifie Data Access Object.

Une fois correctement branché, créez dans UserDAO une méthode isValidLogin() qui interrogera la base avec le couple login / mot de passe. Cette méthode devra retourner un boolean pour savoir si le couple est présent en base ou non.

Aide : Pour consulter le contenue de la base de donnée à un instant t reportez vous au README.md présent à la racine du projet.

Partie 8 : La liste des messages

Sur la page http://localhost:8080/tpjavaee/entries.jsf la liste des messages affichée est pour le moment une liste statique générée dans la classe GoldenBookEntryDAO.
Avec l’aide de JPA vous allez créer la requête qui va permettre de lire et récupérer depuis la base de données la liste des messages.

Pour cela consulter la javadoc de l’EntityManager http://docs.oracle.com/javaee/6/api/javax/persistence/EntityManager.html afin de trouver les méthodes à utiliser.