JavaEE
Développement d'application web

Aix-Marseille Université

Gaël Guibon, Jean-Luc Massat, Omar Boucelma

2018-2019

Sommaire

Chapitre 1

Notions de bases et environnement

1.1 Notions de bases

1.1.1 Java et JavaEE

Vérification du niveau des étudiants en java standard
Topo sur l'intéret du JEE par rapport au java standard + distinction serveur / navigateur
Quelques rappels :

Java	**JavaEE**
Socle de base	Extension de Java
Programmes locaux	Programmes connectés
Entrée par méthode main()	Entrées par chaque servlet
HTML	**JavaEE**
Site web statique	Site web dynamique
Visible	Opaque et sécurisé

1.1.2 Le modèle MVC

Le principe Modèle Vue Controlleur (MVC) est une bonne pratique de programmation.

Modèle	**Vue**	**Controlleur**
Données	Affichage IHM	Actions / logique du code
Accès BDD	Adaptation de l'UI	Lien entre les éléments
Messages/contacts	Visualisation du message/contacts	Boutons "répondre à tous", envoyer emoji, ...

Exemple d'une répartition MVC :

De nombreux frameworks en différents langages sont désormais MVC :

- Java : Spring [1], Struts [2], Tapestry [3]
- Python : Django, Flask, Pyramid
- Javascript : Angular, ReactJS, EmberJS

1. http://spring.io/
2. http://struts.apache.org/
3. http://tapestry.apache.org/

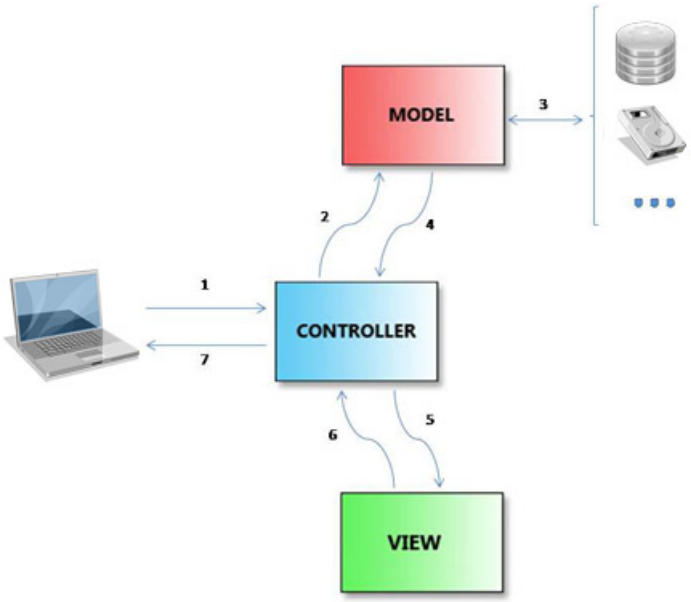

FIGURE 1.1 – Exemple : Messagerie SMS

1.2 Environnement

1.2.1 Installer Java8

Ouvrir un terminal et lancer les commandes suivantes :

1. sudo add-apt-repository ppa :webupd8team/java
2. sudo apt-get update -y
3. sudo apt-get install oracle-java8-installer
4. java -version

La dernière commande devrait vous afficher la version 1.8.x de Java.

1.2.2 Installer Maven

1. cd /opt/
2. wget http ://www-eu.apache.org/dist/maven/maven-3/3.5.0/binaries/apache-maven-3.5.0-bin.tar.gz
3. sudo tar -xvzf apache-maven-3.3.9-bin.tar.gz
4. sudo mv apache-maven-3.3.9 maven
5. sudo nano /etc/profile.d/mavenenv.sh
6. Dans ce fichier ajouter les lignes suivantes :

```
export M2_HOME=/opt/maven
export PATH=${M2_HOME}/bin:${PATH}
```

7. sudo chmod +x /etc/profile.d/mavenenv.sh
8. sudo source /etc/profile.d/mavenenv.sh
9. mvn –version

1.2.3 Installer l'IDE : Eclipse Oxygen

1. Télécharger Eclipse : https://www.eclipse.org/downloads/download.php?file=/oomph/epp/oxygen/R/eclipse-inst-linux64.tar.gz
2. Lancer l'installeur
3. Choisir Eclipse for JavaEE
4. Suivre l'installation avec les configurations par défaut

1.3 Comprendre Maven

1.3.1 Qu'est-ce que c'est ? ! !

Un outil d'automatisation :

— Très répandu en entreprise
— Pour la compilation
— Pour le déploiement
— Facilite le partage de code

Format d'usage :

```
usage: mvn [options] [<goal(s)>] [<phase(s)>]
```

1.3.2 Les éléments de base

```
<project>
  <modelVersion/>     VERSION DE MON PROGRAMME
  <groupId/>  ID DU GROUPE DE MON PROGRAMME : cnrs.amu.lsis ou com.usa.google ou autre
  <artifactId/> ID DE MON PROGRAMME : monprojet ou minecraft ou autre
  <packaging/> TYPE DE PAQUET EN SORTIE : jar ou war ou autre
  <version/>   VERSION DE MON PROGRAMME : 0.0.9-beta ou 2.1.56
  <name/>     NOM EN INTERNE
  <url/>       URL DU PROGRAMME : par defaut mettre : http://maven.apache.org
  <dependencies/>     DEPENDANCES DU PROGRAMME : jsp, guava, et autres librairies externes
  <build>     EXECUTION LORS DE LA COMPILATION DU PROGRAMME
    <plugins/> PLUGINS DIVERS
  </build>
  </project>
```

1.3.3 Architecture d'un projet maven

Générer un projet maven basique (java SE) :

```
mvn archetype:generate -DgroupId=test.project -DartifactId=superTest -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=
    false
```

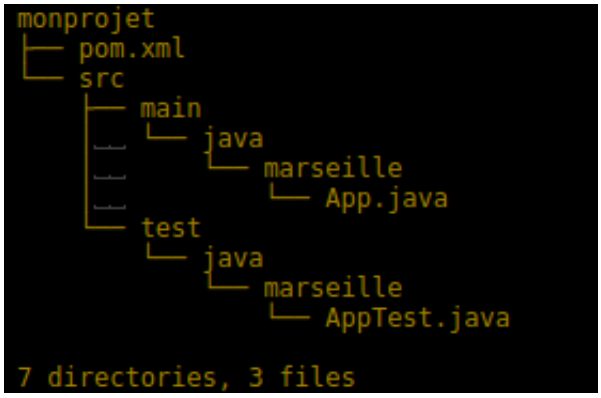

FIGURE 1.2 – Arborescence de base de maven

L'explication officielle de l'arborescence est présente ici : http://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html

1.3.4 Le fichier POM.xml

C'est le fichier central ! Celui qui permet de guider maven dans la compilation.
Un guide officiel du POM est présent à cette adresse : http://maven.apache.org/guides/introduction/introduction-to-the-pom.html

Les dépendances

```
<dependencies>
        <dependency>
                <groupId>org.apache.tomcat.embed</groupId>
                <artifactId>tomcat-embed-core</artifactId>
                <version>${tomcat.version}</version>
        </dependency>
        .
        .
        .
</dependencies>
```

Les plugins

```
<plugins>
    <plugin>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.3</version>
            <configuration>
                    <source>1.8</source>
                    <target>1.8</target>
            </configuration>
    </plugin>
    .
    .
    .
</plugins>
```

1.3.5 Accéder aux dépendances et plugins

Aller chercher sur maven central : https://search.maven.org/

1.4 Hello World

1. Cloner le repository : git clone https ://github.com/gguibon/CoursesJavaEE.git
2. Aller dans le projet de base : cd basic_hello_world
3. Terminal : mvn package
4. Terminal : java -jar target/basicHelloWorld-jar-with-dependencies.jar OU double clic sur le jar
5. Ouvrir un navigateur à l'adresse localhost :8080

Astuce : La commande htop vous indique les processus en cours (sudo apt install htop)

1.5 Sources utiles pour compléter

Documentation officielle de maven : http://maven.apache.org/guides/index.html
Tutoriel officiel : https://docs.oracle.com/javaee/7/tutorial/
Documentations et API officielle : http://docs.oracle.com/javaee/7/index.html
Tutoriel Site du Zero : https://openclassrooms.com/courses/creez-votre-application-web-avec-java-ee
Mémento et exemples d'annotations Servlet 3.0 : http://www.codejava.net/java-ee/servlet/servlet-annotations-

Chapitre 2

Le modèle - beans

2.1 Qu'est-ce que c'est ?

Une manière de stocker les données. Un est :
- Réutilisable ! (objet java)
- Persistant ! (serialization)
- Paramétrable ! (propriétés)

Il respecte donc plusieurs règles :
- C'est une classe publique.
- Implémente Serializable pour pouvoir être sauvegardé.
- AUCUN champ public ! (getter et setters)
- Possède un constructeur par défaut public, sans paramètres ?

2.2 Exo : Agenda

2.2.1 Créer le bean Rendezvous

Créer un bean (un objet java) Rendezvous.java.

1. Ajouter les propriétés "duree"(int), "personnes"(liste de noms), "lieu"(string) et "type"(string)
2. Ajouter des getter et setter pour chaque propriété
3. Ajouter une méthode qui retourne le nombre de personnes

2.2.2 Créer le service RendezvousService

Un service est une classe qui va ouvoir être instancier et gérer un ou plusieurs beans, ainsi que leurs relations.

1. Créer un service RendezvousService.java
2. Ajouter une liste de rendezvous
3. Ajouter un constructeur qui initialise des rendez-vous par défauts
4. Ajouter une méthode d'ajout de rendez-vous
5. Ajouter une méthode qui récupère le nombre de rendez-vous
6. Ajouter une méthode qui récupère les rendez-vous en fonction de plusieurs types de rendez-vous différents

2.2.3 Utiliser le bean

1. Afficher un tableau auto généré pour lequel chaque ligne est un rendez-vous, et chaque colonne une propriété (dans terminal/log)
2. Tester l'ajout de chaque focntionnalité en visualisant au fur et à mesure (manuellement ou non)

Chapitre 3

Le stockage de données - DAO et JDBC Template

3.1 Exo : Stockage de l'agenda

3.1.1 Créer l'interface RendezvousDAO.java

1. Spécifier une méthode de sauvegarde d'un nouveau rendez-vous (ajout)
2. Spécifier une méthode de mise à jour d'un rendez-vous
3. Spécifier une méthode de suppression d'un rendez-vous
4. Spécifier une méthode de requête des rendez-vous
5. Spécifier une méthode de requête des rendez-vous en fonction de leurs types

3.1.2 Créer l'implémentation RendezvousDAOImpl.java

1. Créer une méthode de sauvegarde d'un nouveau rendez-vous (ajout) avec requête SQL et paramètres nommés
2. Créer une méthode de mise à jour d'un rendez-vous avec requête SQL et paramètres nommés
3. Créer une méthode de suppression d'un rendez-vous avec requête SQL et paramètres nommés
4. Créer une méthode de requête des rendez-vous avec requête SQL et paramètres nommés
5. Créer une méthode de requête des rendez-vous en fonction de leurs types avec requête SQL et paramètres nommés

3.1.3 Modifier le service Rendez-vous

— Adapter RendezvousService pour qu'il utilise le DAO
— Tester les fonctionnalités et visualiser les résultats et changements par terminal/logs

3.1.4 *Optionnel* Ajouter d'autres données

Un agenda a plus que de simples rendez-vous :
— Ajouter des rappels (anniversaires, jours fériés, etc.)
— Ajouter des dates butoir ayant une méthode pour une booléenne proprieté *done*