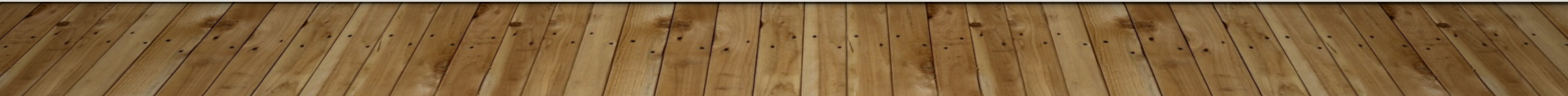


PRÉSENTATION JEE RAPPELS

LICENCE PRO. DAWIN

H. BERGER M.A. TESSIER

2022-2023



TERMES IMPORTANTS

FRAMEWORK

Conteneur d'EJB

JSP

Conteneur Web

SERVLET

JPA

API

ORM

EJB

JDBC

JNDI

MAVEN

POM.XML

FRAMEWORK

- Bibliothèque logicielle (ensemble de classes existantes) utilisées par le développeur pour l'assister dans certaines tâches techniques se retrouvant dans tous les projets (tests unitaires, gestion de la persistance, respect de l'architecture MVC...)
- Un framework n'est **pas seulement une bibliothèque logicielle**: il fournit un **cadre de travail** que le développeur doit respecter afin d'avoir une application correctement structurée.
- Des **annotations** spécifiques à chaque framework sont utilisées dans le code (précédées d'un @ en java, exemples: @Test pour un test unitaire, @OneToMany pour la persistance...)
- Exemples de framework java: **junit** (pour les tests unitaires), **hibernate**(pour la persistance), **spring** (pour développer une application d'entreprise de bout en bout avec respect du modèle MVC)
- Exemples de framework PHP: **Symfony** (développement d'une application web avec respect du modèle MVC), **Laravel** (idem), **Codeigniter** (idem), **Doctrine** (ORM pour la persistance)...

JSP

- Fichier avec l'extension .jsp
- Contient du **code HTML** avec du code **java** au milieu entre les balises `<% %>`
- Chaque fichier jsp est **traduit en un fichier servlet .java** car le serveur d'application web (exemple Tomcat) ne comprend que le code java.
- Correspond à la **couche Vue** du modèle MVC (partie présentation).

SERVLET

- Fichier .java qui hérite de la classe HTTPServlet
- Est capable de recevoir et de traiter une **requête HTTP** (objet HttpServletRequest) et de retourner une **réponse HTTP** (objet HttpServletResponse).
- Joue le rôle de **contrôleur** dans le **modèle MVC**: va **s'appuyer sur le modèle** (en appelant des objets du modèle) pour produire une réponse HTTP et « **router** » vers la bonne vue (.jsp)
- Les Servlets peuvent contenir du code HTML (coder ses vues ainsi engendre une lourdeur d'écriture, c'est pour cela qu'il existe les JSP).
- Chaque fichier .jsp est traduit en un fichier servlet .java

EJB

- Enterprise Java Bean: composant métier développé en java et déployé **sur un serveur distant**.
- Un EJB va fournir un certain nombre de **services**.
- Il est **réutilisable** dans plusieurs projets.
- Il existe **différents types** d'EJB (EJB entité, EJB session et EJB message).
- Un EJB doit être exécuté dans un **conteneur d'EJB**.

CONTENEUR WEB

- Environnement nécessaire pour **exécuter** des **servlets** et des **JSP**.
- Cet environnement fonctionne sur un **serveur d'application web** (exemple: **TOMCAT**).

CONTENEUR D'EJB

- Environnement nécessaire pour **exécuter** des **EJB**.
- Cet environnement fonctionne sur un **serveur d'EJB** (exemple: JBOSS, JOnAS, IBM Websphere, Oracle Weblogic Server, WildFly).

API

-
- **Application Programming Interface**: ensemble de **fonctions** que le **développeur** va pouvoir utiliser en passant par une **interface**, sans avoir à se soucier de la manière dont ces fonctions sont implémentées.
→ c'est une **interface de programmation** destinée au développeur
 - Peut être vue aussi comme un **standard** ou une **norme** à respecter lorsque l'on développe un composant JEE.
 - Un composant est développé dans le but d'être **réutilisé** dans plusieurs projets et pour ce faire, il est donc livré avec une interface permettant aux développeurs de l'utiliser, **en faisant abstraction de son implémentation** (de la manière dont il est codé).
 - Il existe **différents types d'API**: JDBC, JNDI, JPA, JTA, JMS, JAAS...

JPA

- **Java Persistence API**
- API permettant de gérer la **persistance** des objets en base de données.
- Cette API constitue une **norme**, un **standard** pour les **ORM** comme Hibernate.

JDBC

- **Java Data Base Connectivity:** API permettant de gérer la connexion à un SGBD en passant par un **pilote JDBC** (driver JDBC) **spécifique** au SGBD utilisé (exemple: il existe un pilote spécifique pour le SGBD Oracle, un autre pour PostgreSQL, MySQL...).

JNDI

- **Java Naming Directory Interface:** API utilisée pour se **connecter** à un **annuaire** en passant par un **pilote** spécifique à l'annuaire utilisé (exemple: annuaire LDAP).

JTA

- **Java Transaction API:** API permettant de gérer les transactions.

ORM

-
- Object Relationnal Mapper (ou Mapping): outil permettant de gérer la persistance des objets en base de données.
 - Il existe différents ORMs.
 - Exemple d'ORM java: Hibernate, Toplink...
 - Exemple d'ORM PHP: Doctrine.

MAVEN

- **Outil** utilisé pour **construire un projet java EE** (une application d'entreprise JEE).
- Cet outil va utiliser un fichier **pom.xml** contenant toutes les informations nécessaires pour pouvoir **compiler** le projet, le **tester**, **générer** le .war, **installer** et **déployer** le projet sur le serveur d'application.
- Maven utilise **différentes commandes**: mvn compile, mvn test, mvn package, mvn instal, mvn deploy.
- Remarques:
 - Fichier .war: web application archive
 - Fichier .jar: java archive
 - Fichier .ear: enterprise application archive (va emballer un ou plusieurs modules pour les déployer sur un serveur d'application)

POM.XML

- **Project Object Model**: fichier .xml utilisé par **l'outil Maven** et contenant toutes les informations nécessaires pour construire et déployer un projet JEE.
- Ce fichier va :
 - **Décrire** le projet logiciel (nom, numéro de version, noms des contributeurs...)
 - Lister ses **dépendances** (vers d'autres projets éventuellement + bibliothèques logicielles nécessaires)
 - Indiquer les **modules externes** utilisés
 - Indiquer l'ordre à suivre pour la **production** du projet