

# Introduction à Spring et Hibernate

Institut Galilée - Master 2 PLS

---

Jones Magloire

21 Octobre 2019



JawgMaps



takima

- ① Qu'est-ce que JEE ?
- ② Pourquoi Hibernate ?
- ③ Pourquoi Spring ?
- ④ Spring vs Spring Boot ?
- ⑤ Qu'est-ce que Maven Gradle ?
- ⑥ Reactive Programming ?
- ⑦ Et pour après ?

## Qu'est-ce que JEE ?

---



# Qu'est-ce que JEE ?

- Version actuelle Jakarta EE 8
- Ensemble de Java Specification Requests ou JSR
  - JTA (Java Transaction API)
  - JDBC (Java DataBase Connection)
  - JPA (Java Persistence API)
  - Servlet
  - JSP (JavaServer Pages)
  - EJB (Enterprise Java Bean)
- Cédé à Eclipse Foundation en septembre 2017

# Qu'est-ce que JEE ?

## Plusieurs type de serveur

- Serveur d'application (Glassfish, WildFly...)
- Conteneur de servlets (Tomcat, Jetty, WebSphere...)

## Implémentation des spécifications

- Spring Framework et ses modules
  - Spring JDBC (implémente JDBC)
  - Spring ORM (binding hibernate et JPA)
  - Spring Web et WebMVC (implémentent servlet)
- Hibernate ORM (JPA)

JEE

Démo

## Pourquoi Hibernate ?

---



# Que fait Hibernate ?

## Object-Relational Mapping (ORM)

- Classe Java  $\Leftrightarrow$  Table SQL (Entity)
- Transactions (JTA)
- Création des tables au démarrage
- Lazy loading
- Gestion des sessions à la DB
- Cache



Hibernate

Démo

# Avantages et Inconvénients

## Avantages

- Mapping des classes automatique
- Rapidité d'implémentation
- Génère les requêtes SQL
- Changement facile de SGBD

## Inconvénients

# Avantages et Inconvénients

## Avantages

- Mapping des classes automatique
- Rapidité d'implémentation
- Génère les requêtes SQL
- Changement facile de SGBD

## Inconvénients

- Requetes complexes pas toujours optimisées
- ⚠ Les relations One/Many To Many en mode lazy

# Pourquoi Spring ?

---



# Que fait Spring ?

## L'inversion de contrôle (IOC)

- Résolution des dépendances pour la création des beans (Singleton/Prototype)
- Injection des dépendances

## Programmation orientée aspect (AOP)

- Transactions
- Exception handler
- Intercepteurs (ex: sécurité)

Spring Framework

Démo

# Spring vs Spring Boot ?

---



# Spring vs Spring Boot ?

## Spring Boot

- Basé sur Spring Framework
- Configuration automatique de l'application
- Pas de déploiement de war
- Beaucoup, beaucoup, beaucoup moins de code
- Produit un fat jar (plus simple à lancer)
- Bien pour faire des microservices scalable



Spring Boot

Démo

# Qu'est-ce que ~~Maven~~ Gradle ?

---



# Qu'est-ce que Maven Gradle ?

- Gestionnaire de build et de dépendances évolué
- Utilisation du multi-module
- Lancement des tests
- Plugins en tout genre
- Sans installation grâce au wrapper
- Moins verbeux que Maven (Groovy/Kotlin vs XML)

Gradle

Démo

# Reactive Programming ?

---

## Modèle classique

- 1 requête = 1 thread
- 10 requêtes = 10 threads
- 100 requêtes = 100 threads
- Mais combien d'opérations peuvent réellement être exécutées en même temps?

## Modèle réactif

- Des requêtes sur des “workers”
- Optimiser l'activité du thread plutôt que le nombre de threads
- Opérations non bloquantes

## Démo

**Et pour après ?**

---



# Autres frameworks ?

## Alternatives à JEE

- Vert.x (framework événementiel par la fondation Eclipse)
- Netty (framework non-blocking I/O par Netty Project Community)
- Quarkus + GraalVM
- Ktor (framework asynchrone designé pour Kotlin)

# Démo

Questions ?

- 2 mois de formation
  - 1 semaine en piscine (JEE stack)
  - Projet incrémental
    - Java 11 (LTS), Spring, Spring-MVC, Spring-Security, Hibernate ORM, Spring Data JPA, JDBC Template, JPA, Jackson, Selenium, Junit, Mockito, Docker, Jenkins, Gatling, Angular
    - 1 semaine DevOPS
    - 1 jour sur la performance + tuning
  - Conférences et formations JVM, Spring, Hibernate, Git, React.js, Gatling, Agilité, REST
  - Certifications Oracle Certified Java Programmer Associate & Professional
- 4 mois sur un projet interne

- Séminaires (Samedi, Dimanche, Lundi)
- Hackathons
- Meetups interne
- Billard en interne
- Nintendo Switch