

Introduction à Spring et Hibernate

Institut Galilée - Master 2 EDI² et PLS

Jones Magloire

03 Décembre 2018



jawgmaps



takima

- 1 Qu'est-ce que JEE ?
- 2 Pourquoi Hibernate ?
- 3 Pourquoi Spring ?
- 4 Qu'est-ce que ~~Maven~~ Gradle ?
- 5 Reactive APIs?
- 6 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)

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

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

- Requêtes 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)

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 (Goovy vs XML)

Reactive APIs?

- 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, des "workers"
 - Optimiser l'activité du thread plutôt que le nombre de threads

Et pour après ?

Alternatives à JEE

- Vert.x (framework événementiel par la fondation Eclipse)
- Netty (framework non-blocking I/O par Netty Project Community)
- Spring Framework et Spring Boot
- Ktor (framework asynchrone designé pour Kotlin)

Stage takima

- 2 mois de formation
 - 1 semaine en piscine (JEE stack)
 - Projet incrémental
 - Java 11 (Stream API, Lambdas...), Spring, Spring-MVC, Spring-Security, Hibernate ORM, Spring Data JPA, JDBC Template, JPA, Criteria, QueryDSL, Jax-WS, Jax-RS, Jackson, Selenium, Junit, Mockito, Docker, Jenkins, Gatling, Angular.js
 - 1 semaine DevOPS
 - 1 jour sur la performance + tuning
 - Conférences techniques (JVM, Spring, ORMs)
 - Formations annexes: Git, React.js, Android, Gatling, Agilité
 - Certifications Oracle Certified Java Programmer Associate & Professional
- 4 mois sur un projet interne

- Séminaires (Vendredi, Samedi, Dimanche)
- Hackathons
- Tech nights
- 1 Conf offerte / an (Devoxx, Droidcon, Paris Container Day...)
- 1 Billard en interne
- 1 Nintendo Switch