

Introduction à Spring et Hibernate

Institut Galilée - Master 2 P2S

Jones Magloire

27 Septembre 2024



Qui suis-je ?

CTO / DevOPS / Carto & Geocoding Expert

- Diplômé Master PLS 2015 (Mention Très Bien)
- Stage de 6 Mois chez takima
- Création du projet Jawg Maps 2015 (juillet)
- Création de la start-up Jawg Maps 2017
- Developpeur backend (Kotlin, NodeJS, Rust, Java)



Qui suis-je ?

Passionné de Développement et Photographie

- Site web: <https://joxit.dev>
- Contributions journalières sur Github @Joxit
 - Pelias Geocoder (github.com/pelias)
 - Docker Registry UI (github.com/Joxit/docker-registry-ui)
 - Vert.x (github.com/eclipse-vertx)
- Partage photos sur instagram @jox.it



- 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 ?

Histoire de JEE

- Création en 1999 sous Java 1.2
- Version actuelle Jakarta EE 10
- 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 ?

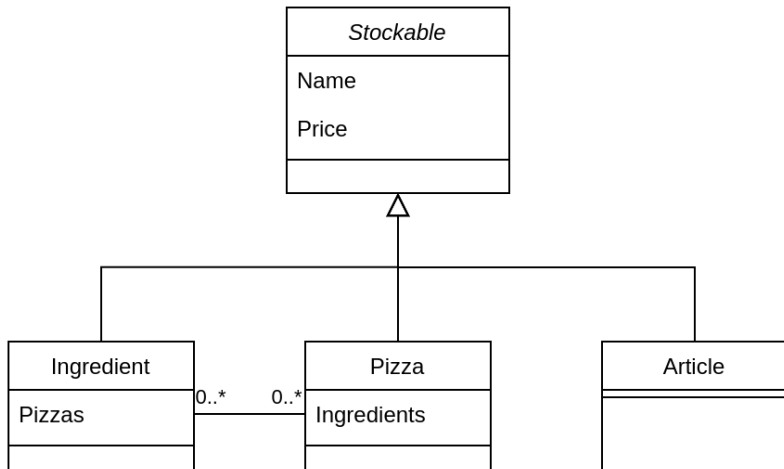
Plusieurs type de serveur

- 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)



Démo

Pourquoi Hibernate ?

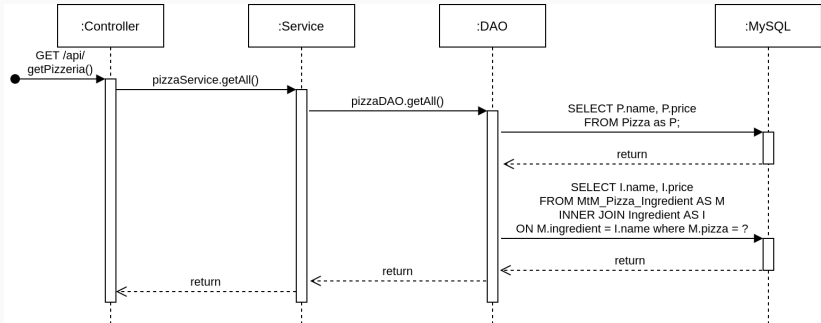


Que fait Hibernate ?

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 de requêtes

Hibernate (Démo)



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

- Requêtes complexes pas toujours optimisées
- ⚠ Les relations One/Many To Many en mode lazy sue certaines versions

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)

Démo

Spring vs Spring Boot ?



Spring Boot Autoconfigure

Autoconfiguration

- Chargement de configuration par défaut
- Chaque module Spring apporte sa configuration

Charger une configuration

1. Configuration ligne de commande ex: `--server.port=9000`
2. Configuration venant de `SPRING_APPLICATION_JSON` (inline JSON) ex: `{"server":{"port": 9000}}`
3. Java System Properties ex `java -Dserver.port=9000 ...`
4. Variable d'environnement OS ex: `SERVER_PORT=9000`
5. Fichiers `application.properties` ou `application.yaml` dans le dossier `/config` ou à côté du jar ou dans le classpath
6. Les valeurs par défaut de configuration (dans le code)

Spring Framework vs Spring Boot ?

Spring Boot

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

Spring Framework

- Configuration via XML
- Déploiement d'un war (serveur lourd)

Démo

Qu'est-ce que Maven Gradle ?



Qu'est-ce que Maven Gradle ?

Gradle

- Gestionnaire de build et de dépendances évolué
- Utilisation du multi-module
- Lancement des tests
- Plugins en tout genre
- Sans installation grâce au wrapper (également disponible dans Maven 3.7+)
- 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

Reactive Programming Démo

Spring Framework + Servlet

VS

Spring Boot Web MVC

VS

Spring Boot Webflux (Réactive)

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)

Spring Boot Weblux

vs

Vert.x

Questions ?

Pour 6 mois de stages

- 3 mois et demi de formation
- 2 mois et demi sur un projet interne
- Back: Java, Kotlin, Spring, Hibernate, JEE...
- Front: React et Angular
- Ops: Docker, Gitlab CI...



Les vie d'entreprise

- Invité aux séminaires (vendredi, samedi, dimanche)
- Soirés jeux de sociétés
- Meetups et BBL en interne
- Billard + Nintendo Switch...

