10 raisons pour adopter Springboot

Table of Contents

Spring Framework

Spring est un framework Java très populaire pour la construction d'applications Web et d'entreprise. Contrairement à de nombreux autres cadres, qui se concentrentsur un seul domaine, Spring Framework offre une grande variété de fonctionnalités répondant aux besoins commerciaux modernes grâce à ses projets de portefeuilleLe framework Spring offre une flexibilité pour configurer les beans de plusieurs manières telles que XML , Annotations et JavaConfig . Avec le nombre defonctionnalités augmentées, la complexité augmente également et la configuration des applications Spring devient fastidieuse et susceptible d'erreursL'équipe Spring a créé SpringBoot pour répondre à la complexité de la configuration.Mais avant de plonger dans SpringBoot, nous allons jeter un coup d'œil sur le cadre du printemps et voir quel type de problèmes SpringBoot essaie de résoudre.Dans cet article, nous couvrirons:Vue d'ensemble du cadre de printempsUne application Web utilisant Spring MVC et JPA (Hibernate)Un exemple rapide de SpringBootVue d'ensemble de Spring FrameworkSi vous êtes un développeur Java, il y a de fortes chances que vous ayez entendu parler de Spring Framework et l'avez probablement utilisé dans vos projets. Lecadre de printemps a été créé principalement comme un conteneur d'injection de dépendance, mais c'est beaucoup plus que cela. Spring est très populaire en raison de plusieurs raisons:

- L'approche d'injection de dépendance de Spring encourage l'écriture du code vérifiable
- Fonctionnalités de gestion des transactions de base de données faciles à utiliser mais puissantes
- Spring simplifie l'intégration avec d'autres frameworks Java comme JPA / Hibernate ORM, Struts / JSF / etc. Cadres web
- Cadre Web MVC de pointe pour la construction d'applications Web

Parallèlement à Spring Framework, il existe de nombreux autres projets Spring Brothers qui aident à créer des applications répondant aux besoins des entreprises modernes:

- Spring Data: simplifie l'accès aux données des magasins de données relationnels et NoSQL.
- Spring Batch: offre un puissant cadre de traitement par lots.
- Spring Security: cadre de sécurité robuste pour sécuriser les applications.
- Spring Social: prend en charge l'intégration avec des sites de réseaux sociaux comme Facebook, Twitter, LinkedIn, GitHub, etc.
- Spring Integration: une implémentation de Patterns d'intégration d'entreprise pour faciliter l'intégration avec d'autres applications d'entreprise utilisant des messagerie légère et des adaptateurs déclaratifs.

Il existe de nombreux autres projets intéressants abordant divers autres besoins de développement d'applications modernes. Pour plus d'informations, consultez http://spring.io/projects. Dans les premiers jours, Spring Framework fournit une approche basée sur XML pour configurer les beans. Plus tard Spring a introduit des DSL basées sur XML,desannotations et des approches basées sur JavaConfig pour configurer des beans Permettez-nous de regarder rapidement comment ressemble chacun de ces styles deconfiguration.

Pourquoi Springboot?

Spring Boot a été conçu pour rendre la vie du développeur plus simple et lui permettre de se concentrer sur le coeur de l'application et non pas sur les aspects annexes : configuration, tests, sécurité, déploiement…

Les 10 raisons

RestController

Kiss:application.yml

Plus de script de gestion d'environnement (@Component, @Configuration et possibilité de déclarer dans application.yml ou application.properties). @EnableAutoConfiguration. Cette annotation va alors s'appuyer sur l'ensemble des dépendances de l'application (MVC, Tomcat, …) pour configurer l'application.

Plus de serveur d'application Deploiement facile (tomcat embarqué ou jety)

Simplicité de déploiement (le seul prérequis est Java). Cloud-ready.

Securité

Boite à outils pour intégration

Bases relationnelles : JDBC, JPA, JdbcTemplate Bases NoSQL : Redis, MongoDB, ElasticSearch … Messaging : JMS

Springboot et Docker

```
FROM java:8u45
MAINTAINER Gregory Le Bonniec "gregory.le.bonniec@ellixo.com"
ADD springboot-1.0-SNAPSHOT.jar app.jar
ENTRYPOINT [ "java", "-Dspring.profiles.active=test", "-jar", "/app.jar" ]
```

```
$ docker build -t ellixo/springboot .
$ docker run -d -p=9292:9292 ellixo/springboot
```

Environnemet Prod:

Pour aller plus loin, il est possible d'installer le module Actuator qui fournit de nombreuses fonctionnalités d'administation système (via notamment une API Rest): health : fournit des données permettant de vérifier l'état de l'application (UP/DOWN, état disque, état systèmes externes ...) metrics : fournit des métriques processus (threads, CPU, mémoire ...) trace : fournit les informations des dernières connexions HTTP applicatives ... Libre à vous ensuite de connecter ce module à l'outil de monitoring du système d'information (Graphite, Promotheus ...)

```
Exemple : API Health
----
$ curl http://user:password@localhost:9292/health
{"status":"UP","diskSpace":{"status":"UP","free":169718296576,"threshold":10485760},"m
ongo":{"status":"UP","version":"3.0.2"}}
----
```

Cloud

Encore une fois, le fait qu'une application Spring Boot embarque son propre containeur (Tomcat ou Jetty par défaut donc) simplifie un déploiement cloud. Pour démontrer le rapidité du processus, j'ai décidé d'exposer le déploiement sous la plateforme Cloud Foundry de Pivotal (à tout seigneur, tout honneur) : Une fois votre compte Pivotal Web Services créé et le client associé installé, la seule commande à exécuter sur votre environnement est : $ cf push springboot-demo -p springboot-1.0-SNAPSHOT.jar ... Uploading app files from: springboot-1.0-SNAPSHOT.jar Uploading 623.8K, 96 files Done uploading OK

Par défaut, Cloud Foundry prend en compte le profil "cloud" ; pour autant, il est possible d'activer un autre profil en positionnant la variable d'environnement JAVA_OPTS (exemple : -Dspring.profiles.active=test). L'application est alors disponible via l'URL nom_app.cfapps.io (http://springboot-demo.cfapps.io ici)