

Concevoir et développer une application Cloud Native avec Spring

Appliquer les bonnes pratiques de TDD et de Devops

DESCRIPTION

Spring est aujourd'hui la plateforme la plus utilisée pour les applications d'entreprises écrites en java. Les derniers modules tels que spring config et spring cloud offrent tous deux les outils pour une application cloud ready.

Tout au long de cette formation, vous découvrirez comment manipuler les concepts de base de la programmation java avec spring : maven, injection de dépendance, contrôleurs, templates jdbc. Le troisième jour abordera les concepts de déploiement cloud ready.

OBJECTIFS PEDAGOGIQUES

- Découvrir Les caractéristiques du framework
- Mettre en œuvre une architecture modulaire et cloud ready
- Tester unitairement et de bout en bout son application

PUBLIC CIBLE

Développeur

PRE-REQUIS

Être à l'aise dans la pratique d'un langage objet

METHODE PEDAGOGIQUE

Formation avec apports théoriques, échanges sur les contextes des participants et retours d'expérience du formateur, complétés de travaux pratiques et de mises en situation.

PROFIL DES INTERVENANTS

MODALITÉS D'ÉVALUATION ET FORMALISATION À L'ISSUE DE LA FORMATION

Stage pratique
Développement

Code :
JAVSP

Durée :
3 jour(s) (21,00 heures)

Exposés : **10 %**
Cas pratiques : **80 %**
Echanges d'expérience : **10 %**

Inter-entreprises :
Prochaines sessions
disponibles [sur notre site web](#).
Tarif : 2 150,00 € HT /
participant

Intra-entreprise :
Tarifs et dates sur demande.

PROGRAMME PEDAGOGIQUE DETAILLE

JOUR 1

INTRODUCTION

- Présentation de l'architecture d'une application web cloud ready
- Architecture logicielle et périmètre d'utilisation du langage java

CHAPITRE 1 : COMPILATION ET PACKAGING EN JAVA

- Le compilateur et les class
- Le classpath
- Développement d'un mini-serveur
- Les fichiers jar, les notions de librairies et frameworks

CHAPITRE 2 : MAVEN

- Le fichier pom.xml
- Les dépendances
- Le plugin Apache Maven Compiler

CHAPITRE 3 : TDD ET INJECTION DE DEPENDANCES

- Les tests unitaires
- La notion d'interface
- Développement, tests unitaires JUnit pour expérimenter l'injection de dépendance

CHAPITRE 4 : SPRING CORE

- La notion d'application context
- La notion de fichier de configuration
- La notion d'annotations
- Refactoring du code précédent de façon à utiliser Spring

JOUR 2

CHAPITRE 5 : SPRING DATA

- Rappel sur les bases de données relationnelles

- Présentation de MySQL et de sa version AWS : RDS
- Le concept de JDBC template
- Développement d'une couche de stockage
- Les tests d'intégration avec la base H2

CHAPITRE 6 : SPRING BOOT

- Spring Boot vs Spring Framework, quelles différences ?
- La classes AutoConfiguration et SpringBootApplication
- De nouveaux outils pour simplifier les tests
- Refactoring du code précédent pour tirer partie de Spring Boot

CHAPITRE 7 : SPRING WEB

- Rappel de conception d'API
- Les notions de contrôleurs et de route
- Les tests d'intégration
- Développement d'une API REST

JOUR 3

CHAPITRE 8 : LE DEPLOIEMENT SUR UN ENVIRONNEMENT D'INTEGRATION

- Gérer l'évolution de son schéma de données avec flyway
- Les notions de Placeholder, de properties, de profile
- La notion d'environnement
- Configuration d'un environnement avec la base H2 et d'un environnement avec la base PostgreSQL

CHAPITRE 9 : UN PREMIER PATTERN CLOUD READY : LE CIRCUIT BREAKER

- Les twelve factor app
- Le pattern circuit breaker
- Mise en place du circuit breaker sur la connexion à la base de données

CHAPITRE 10 : LE DEPLOIEMENT SUR LE CLOUD (OPTIONNEL)

- La plateforme de PaaS Beanstalk
- Déploiement de l'application sur la plateforme beanstalk
- Le concept de load balancing et sa configuration

