

Formation Gradle

Automatiser le build d'une application avec Gradle

Référence : GRADLE-02

Durée : 3 jour(s)

Présentation

Rejoignez-nous pour trois jours d'enseignement intensif de Gradle.

La formation commence avec les fondamentaux de Groovy, puis rapidement une couverture complète des sujets Gradle comme des tâches, standard plugins Gradle, manipulation de fichiers, en charge la gestion des log avec Gradle, les IDE intégration, multiprojet construit, l'intégration de Ant, l'intégration de Maven, des tâches personnalisées, et même des plugins Gradle sont vus.

La moitié du temps sera consacrer à coder avec Gradle et l'autre moitié sera passer avec le formateur afin d'échanger et d'apprendre à travers des exercices extraits de son livre.

Cette formation s'adresse aux personnes ne connaissant pas Gradle comme à ceux qui sont familier avec la pratique de Gradle. Ne ratez pas ce cours!

Objectifs

- Apprendre les éléments fondamentaux du Gradle Domain Specific Language (DSL)
- Apprendre à utiliser le système de plugins de Gradle
- Maîtriser la personnalisation de le chaîne de build
- Appréhender la puissance de la maîtrise des dépendances avec Gradle

Répartition: 25% Théorie, 75% Pratique

Public: Architecte, Leader Technique, Développeur

Pré-requis: Connaissance de Java et notions de Groovy

Programme

Les bases de Groovy

- Utilisation de la console Groovy
- Les bases de la syntaxe
- Le typage dynamique
- Strings
- Closures
- Collections and iterators
- Properties
- File I/O

- SLF4J
- Rediriger System.out.println()
- Les niveaux de Log

L'intégration dans les IDE

- Le plugin Gradle de Eclipse
- Utilisation de SpringSource Tool Suite avec Gradle
- Le plugin Eclipse de IntelliJ
- Utiliser IntelliJ avec Gradle

Introduction à Gradle

- Les étapes de build impératives
- Les styles de build impératif
- Convention over configuration
- Modélisation par convention
- Le graphe de dépendances

Les builds multi-projets

- Le modèle de build Gradle et l'environnement multi-projet
- Déclarer un build multi-projet
- L'API multi-projet

Tasks

- Déclarer une tâche Gradle ("task")
- Les actions d'une tâche
- La configuration d'une tâche
- Les phases du cycle de vie Gradle
- Les propriétés des tâches
- Les tâches typées
- Déclarer une tâche

L'intégration avec Ant

- Comparaison de paradigmes
- Comparaison de vocabularies
- L'objet AntBuilder de Gradle
- Importer un build Ant
- Etendre les cibles ("target") Ant à travers des tâches Gradle
- Modifier les propriétés Ant dans un build Gradle

Les plugins

- Le modèle d'un plugin Gradle
- Revu des plugins du noyau

L'intégration avec Maven

- Introspecter le plugin Maven de Gradle
- Importer le plugin GradleM2Metadata
- Conversion avec Maven2Gradle

Le plugin Java

- Une simple construction Java
- Les tâches du plugin Java
- Les tests Java
- Introspection de l'objet SourceSets
- Personnaliser un build Java
- Utilisation du JavaExec
- La création dynamique de tâches

Les tâches personnalisées

- Extraire le code impératif du build
- Annoter les actions de tâches
- Annoter les entrées et sorties de fichiers
- Build file, buildSrc, localisation de JAR extérieur

La gestion de dépendances

- Spécifier des dépendances
- Configurations
- Résolution avec le repository Maven central
- Résolution avec des répertoires

Les plugins personnalisés




- La philosophie des plugins: étendre la DSL
- L'API des plugins
- Les conventions des plugins
- Créer des tâches personnalisées
- Créer des objets métier
- Packager des plugins

Manipulation de fichier

- file(), files(), et fileTree()
- L'interface FileCollection
- Accéder aux fichiers Zip
- La tâche Copy
- CopySpecs
- Les filtres et la modification de fichiers

Logging

OFFERT EN INTER-ENTREPRISE

-  Le petit déjeuner croissants, jus d'orange, café)
-  Le déjeuner
-  Une qualification téléphonique si nécessaire avec l'un de nos consultants

Tel: +33(0)1 45 26 19 15
Fax : +33(0)1 75 43 49 92
Email : training@zenika.com

Auteur du cours



Grégory Boissinot est un spécialiste des outils de build et des problématiques d'intégration continue qu'il a mis en place à grande échelle. Grégory possède une expérience industrielle grand compte de ces problématiques et contribue activement à la communauté **Open Source** dans ce domaine en étant l'un des principaux commiteurs Hudson/Jenkins mondiaux.



zenika
ARCHITECTURE INFORMATIQUE