



Nous allons vous faire aimer Java

Formation Maven (3j)

Notre formation maven (maven 3.0.4) complète nos tutoriaux Maven. Après la formation maven objjis, vous aurez les réflexes et compétences clés vous permettant d'intégrer efficacement tout projet impliquant l'outil de construction et de gestion de projet maven, moins connu que l'outil ant, mais qui s'impose comme une brique incontournable pour le développement modulaire, la gestion de dépendances et le pilotage d'applications modernes.

Couplé avec les frameworks Spring, struts et hibernate, il constitue aujourd'hui le socle de nombreux projet à forte valeur ajoutée, réalisés par des équipes de développement réparties.

Objectifs : fournir les réflexes et le savoir-faire permettant d'intégrer facilement un projet impliquant MAVEN 3

Audience : développeurs , administrateurs, intégrateurs

Prérequis : Aucun. Une connaissance de Java serait un plus.

Moyens pédagogiques : 1 ordinateur/stagiaire. Supports cours. Travaux pratiques. Vidéoprojecteur. Tests

Durée : 3 jours

Vous allez apprendre à :

- ▶ Comprendre les notions de modèle POM, phases, plugin, coordonnées de maven
- ▶ Créer, lire et comprendre rapidement tout fichier pom.xml de maven
- ▶ Créer un squelette de projet à partir des modèles et archetypes maven
- ▶ Ajouter une dépendance externe à votre projet
- ▶ Remplacer Ant par Maven dans vos projets
- ▶ Créer des applications multi modules maven
- ▶ Balises dependencyManagement, pluginManagement
- ▶ Automatiser des tâches récurrentes (compil., test, déploiement)
- ▶ Maîtriser les concepts d'héritage et de composition d'application
- ▶ Factoriser les propriétés des versions et dépendances
- ▶ Mettre oeuvre la technique de filtrage de ressources de maven
- ▶ Sécuriser vos mots de passe Maven
- ▶ Développer en équipe avec Maven, Subversion, eclipse
- ▶ Générer le site d'un projet
- ▶ Générer les rapports clés pour un chef de projet
- ▶ Intégrer Spring, Hibernate, CXF, JSF dans un projet Maven
- ▶ Mettre en oeuvre un Repository Archiva pour vos livraisons.
- ▶ Bonnes pratiques Repository : releases, snapshots
- ▶ Créer un plugin Maven

Programme détaillé

Introduction

- ▶ Définition de maven pour un développeur
- ▶ Définition de maven pour un chef de projet
- ▶ Règle du « Convention over configuration » dans maven
- ▶ Valeur ajoutée de maven par rapport à ANT

Le POM (Project Object Model)

- ▶ Présentation du modèle POM maven et notion de coordonnées
- ▶ Détails et sections du fichier pom.xml de maven
- ▶ Le « super POM » et les mécanismes d'héritage de maven

- ▶ Exploration de la structure des projets Maven
- ▶ Les 200 types de projets Maven
- ▶ Notion de propriétés et de filtre des ressources dans maven

Maven 3

- ▶ Les limites de maven 2
- ▶ Conteneurs plexus, google guice
- ▶ Verbose pom.xml
- ▶ complexité configuration
- ▶ Multilangage : yaml, groovy
- ▶ refonte Noyau
- ▶ compatibilité maven 2 / maven 3

Repository Maven et coordinations

- ▶ Repository local de maven : .m2/repository
- ▶ Mécanisme de localisation d'une librairie dans maven
- ▶ Notion de repositories distants
- ▶ Outillage pour la gestion du cache et de la sécurité avec Archiva
- ▶ Mise en place de Maven

Modèle de gestion des dépendances

- ▶ Dépendance transitive dans maven
- ▶ Optimisation des dépendances dans maven
- ▶ Le cycle de construction d'un projet maven
- ▶ Les différents cycles de construction
- ▶ Comprendre les phases / goals /cycles Maven

Exécution de plugins

- ▶ Maven : une plateforme d'exécution de plugins
- ▶ Configuration plugins et cycle de construction Maven
- ▶ Connaître les tâches d'un plugin maven
- ▶ Mise en œuvre plugins maven

Les profils

- ▶ Définition et intérêt des profils maven
- ▶ Configuration et mise en œuvre de profils maven
- ▶ Technique du filtrage de ressources
- ▶ Présentation des bonnes pratiques maven

Développement de plugins

- ▶ Création d'un plugin Maven en Java
- ▶ Extension pour la création d'un type de projet
- ▶ Debugging de l'exécution d'un plugin dans un IDE

Archetype et Assemblies

- ▶ Définition et spectre des archetypes existants
- ▶ Génération de livrable à la demande
- ▶ La gestion de configuration dans maven
- ▶ Communication avec un gestionnaire de sources
- ▶ Fabrication d'un livrable et détails des pré-requis

L'exécution des tests.

- ▶ Organisation Maven pour les tests d'intégration et fonctionnels
- ▶ Mise en œuvre du debugging des tests dans Eclipse
- ▶ Tests d'intégration application web démarré par Maven

Les rapports

- ▶ Génération site du projet Maven et déploiement
- ▶ Rapport des différents modules du projet maven
- ▶ Rapport couverture de test dans maven
- ▶ Rapport couverture de code dans maven
- ▶ Rapport détection de bugs dans maven

- ▶ Rapport respect normes développement (Sun, IBM...) dans maven.
- ▶ Injection des rapports dans Sonar

Les tests

- ▶ plugin surefire : déclaration et configuration
- ▶ Tests unitaires avec Junit4 et TestNG
- ▶ Tests unitaires : phase 'test'
- ▶ Test d'intégration : phase 'integration-test'
- ▶ Phases pre-integration et post-integration
- ▶ Patterns de fichiers
- ▶ Profil pour test intégration
- ▶ Plugins clover, emma, cobertura
- ▶ Plugin Selenium et intégration client selenium-rc
- ▶ Déploiement multiples avec Cargo

Maven dans les équipes de développement

- ▶ Intégrer un projet maven dans IDE Eclipse
- ▶ Les différents modes d'utilisation de Maven avec les IDE Eclipse
- ▶ Cas d'utilisation du plugin eclipse m2eclipse
- ▶ Graphe de dépendances
- ▶ Indexation et recherche rapide coordonnées

Intégration continue

- ▶ Problématiques et enjeux
- ▶ Le rôle de maven dans l'intégration continue
- ▶ balises ciManagement et distributionManagement
- ▶ Outillages et mise en pratique avec Hudson
- ▶ Concurrents d'hudson

Guide de dépannage

- ▶ Que faire en cas d'erreur ?
- ▶ Méthode de résolution de pb.
- ▶ options -q, -e, iX
- ▶ plugin help
- ▶ Connaître le pom effectif
- ▶ Connaître le profil effectif