

Apache Maven

**Nicolas De loof
et Arnaud Héritier**

Avec la contribution des membres francophones de la communauté Maven,
en particulier Stéphane Nicoll, Vincent Siveston, Raphaël Piéroni, Hervé Boutemy,
Jérôme Van Der Linden, Antonio Goncalves et François Le Droff

PEARSON

The Pearson logo consists of the word "PEARSON" in a white, uppercase, sans-serif font, centered within a dark rectangular background. Below the text is a thin, white, curved line that arches under the letters.

Pearson Education France a apporté le plus grand soin à la réalisation de ce livre afin de vous fournir une information complète et fiable. Cependant, Pearson Education France n'assume de responsabilités, ni pour son utilisation, ni pour les contrefaçons de brevets ou atteintes aux droits de tierces personnes qui pourraient résulter de cette utilisation.

Les exemples ou les programmes présents dans cet ouvrage sont fournis pour illustrer les descriptions théoriques. Ils ne sont en aucun cas destinés à une utilisation commerciale ou professionnelle.

Pearson Education France ne pourra en aucun cas être tenu pour responsable des préjudices ou dommages de quelque nature que ce soit pouvant résulter de l'utilisation de ces exemples ou programmes.

Tous les noms de produits ou marques cités dans ce livre sont des marques déposées par leurs propriétaires respectifs.

Apache, Apache Maven, Maven, and the Apache Maven logo are trademarks of The Apache Software Foundation. Used with permission. No endorsement by The Apache Software Foundation is implied by the use of these marks.

Publié par Pearson Education France
47 bis, rue des Vinaigriers
75010 PARIS
Tél. : 01 72 74 90 00
www.pearson.fr

Mise en pages : TyPAO

ISBN : 978-2-7440-2337-8
Copyright © 2009 Pearson Education France
Tous droits réservés

Aucune représentation ou reproduction, même partielle, autre que celles prévues à l'article L. 122-5 2° et 3° a) du code de la propriété intellectuelle ne peut être faite sans l'autorisation expresse de Pearson Education France ou, le cas échéant, sans le respect des modalités prévues à l'article L. 122-10 dudit code.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Table des matières

Table des listings	XI
Préface	XIII
Avant-propos	XVII
Contenu	XIX

Partie I

Premiers pas avec Maven

1 Introduction	3
Prologue	3
Partageons !	4
Les fourmis à la rescousse	6
Et Maven dans tout ça ?	7
Que fait Maven ?	8
La clé du mystère	10
Convention plutôt que configuration	11
Décrire plutôt que programmer	12
POM	12
Pourquoi adopter ces conventions ?	15
La force de Maven	16
2 Au-delà de <code>java.lang</code>	17
Des JAR sous CVS	17
Quand le répertoire <code>lib</code> explose	17
Identification univoque	20
Dépôt de bibliothèques	21
Avis aux amateurs de casse-tête	22
L'ami de mon ami...	23
Testons un peu	26
Les "scopes"	27

Une arme à double tranchant	27
L'analyse des dépendances	30
Conclusion	32
3 Un peu plus que compiler	33
Êtes-vous prêt pour Java 7 ?	33
Plugins	35
Propriétés	36
Quand Java ne suffit plus	38
Où placer les sources	39
Ajouter un plugin	40
Plugin et tâches	41
Compiler... en JavaScript	42
Invoquer un plugin	45
Cycle de vie	45
Générer du code	48
Produire autre chose qu'un JAR	50
Des plugins pour tout ?	52
Conclusion	53
4 Mettre en place des tests unitaires	55
Tester ? Pour quoi faire ?	55
Automatisons !	56
Utiliser un framework de test	58
Les tests sous Maven	60
Le scope "test"	61
Le développement piloté par les tests	62
Pas de JAR sans tests réussis	65
Réutiliser notre outillage de test	65
L'intégration continue	67
Continuum	68
Hudson	69
Lequel choisir ?	70
Conclusion	71
5 Mettre en place des tests d'intégration	73
Des tests unitaires... de moins en moins unitaires	73
Qu'est-ce qu'un test "unitaire"	74
Les profils	75
S'adapter à l'environnement	76
Désactiver à la demande	78
Tester l'accès à une base de données	79

Utiliser des tests fonctionnels	82
Tester la charge et les performances	84
Intégration continue	85
Conclusion	86

Partie II

Maven en entreprise

6 Gestion avancée des dépendances	91
Oracle, quand tu nous tiens	91
Un moteur de recherche pour Maven	92
Pourquoi publier un POM sans JAR ?	93
Installer le fichier manquant	94
Les dépendances "System"	94
Créer son propre dépôt	96
Contrôle d'identité, vos papiers s'il vous plaît !	96
Rebelote : mais où est javax.jms ?	97
Gérer son dépôt privé	98
Métadonnées	100
Passer à un "véritable" gestionnaire de dépôt	100
Un miroir de <i>central</i>	101
Un gestionnaire d'artefacts	101
Conclusion	106
7 Quand le projet devient trop lourd	107
Un projet – un artefact	107
Héritage	108
Packaging d'un POM parent	109
Parent "naturel"	110
Mutualiser	111
Gestion des dépendances	111
Gestion des plugins	112
Diviser pour régner	114
Héritage "naturel"	115
Et l'intégration continue ?	115
Un gros projet vs plein de modules	116
Les modules au service de l'architecture	117
Conclusion	117

8 Maven et JEE	119
Java Entreprise Edition	119
Construire une archive web WAR	120
Construire un EJB	123
Construire une archive d'entreprise EAR	126
Tester	128
Selenium	128
Cargo	132
Soyons pragmatiques, soyons productifs	134
Une application web "sur place"	134
Une application web sans assemblage	135
Ne plus sortir de l'IDE	135
Tester les EJB	137
JEE6	137
Conclusion	139
9 Maven et les IDE	141
Un plugin Maven pour Eclipse	141
Maven vu depuis l'IDE	142
Eclipse	143
Installation	143
Import d'un projet Maven	143
Prise en main	144
Gestion du POM	145
Intégration des plugins Maven	147
Et la cerise... ..	148
Interrogations	149
Intellij Idea	149
Import d'un projet Maven	149
Gestion du POM	150
Intégration des plugins Maven	152
Bonus	152
NetBeans	153
Prise en main	153
Import d'un projet Maven	153
Gestion du POM	154
Intégration des plugins Maven	154
Bonus	154
Délibération du jury	155
Conclusion	156

10 Le jour J : la livraison	157
Stratégie de livraison	157
Première livraison	157
Deuxième livraison	158
Troisième livraison	158
Documentation	159
Le plugin release	160
Étape 1 : préparation	160
Étape 2 : livraison	161
Et si ça foire ?	163
Notre prochaine version	163
Tester des candidats	163
Urgence !	165
Au-delà de l'intégration continue	166
Conclusion	168

Partie 3

Encore plus loin avec Maven

11 Utiliser un outil non supporté	171
Un outil maison	171
Réutiliser l'existant	172
Retour dans un monde de scripts ?	174
Créer un plugin	174
Pas de panique !	174
Des paramètres pour le plugin	176
Un modèle dynamique	177
Plexus	178
Des classes et des royaumes...	180
Au-delà de Java	183
Tester notre plugin	185
Plugin testing harness	186
Plugin invokeer	187
Conclusion	189
12 L'assurance qualité	191
Audit de code	191
Analyse statique	192
Analyse dynamique	195

Les rapports Maven	198
Autre chose que du HTML	200
Exploiter notre gestion documentaire	201
68 %, qui dit mieux ?	202
L'entropie augmente	202
Maîtrise de S	203
Sonar	204
Conclusion	207
13 Respecter un format de distribution	209
D'où vient ce JAR ?	209
Numéro de construction	210
Numéro de révision	211
Utiliser le MANIFEST	212
La confiance règne...	214
L'EAR ne suffit pas	216
Assemblage du livrable	217
L'intégration continue produit notre livrable	219
L'œuf ou la poule ?	220
OSGi ?	221
Conclusion	222
14 Un nouveau projet démarre.....	223
Mutualiser	223
Qui paye ?	224
Partager un POM parent	224
Copier-coller	225
Copier et mutualiser !	226
Un plugin qui crée des projets	226
Un archétype ?	228
Construire ses propres archétypes	228
Gérer un projet de référence	229
Donner le meilleur de nous-mêmes	230
Démarrer sur les bons rails	230
Un support pour expérimenter	231
Un support de démonstration	232
Conclusion	233
15 Avons-nous fait le bon choix ?	235
Les limites	236
Points faibles	236
Les plugins	237

Le support	238
Le coût de Maven	240
La concurrence	240
Maven bon partout ?	240
Ant et Ivy	242
EasyAnt.....	242
Gradle.....	242
Maven 1	244
Buildr	244
Un outil reconnu	244
La communauté	245
L'équipe de développement	247
L'adoption en entreprise	247
L'avenir de Maven	248
Maven 2.x	249
Maven 3.x	251
À qui appartient Maven ?	255
La fondation Apache	256
Sonatype	256
Maven + OSGi = Tycho	257
Non, Sonatype n'est pas seul !	258
La garantie par l'open-source	259
Conclusion	260
16 Nos recommandations	261
Les bonnes bases	261
Commandement n° 1 : Les conventions de Maven tu suivras.	262
Commandement n° 2 : Simplicité tu choisiras.	262
Commandement n° 3 : Au fur et à mesure de tes besoins, les outils nécessaires tu mettras en place.	264
Commandement n° 4 : De la sur-conception point tu ne feras.	265
Commandement n° 5 : Tes outils et ton build à jour tu maintiendras.	266
Commandement n° 6 : Dans un projet, la même version tous les modules auront.	267
Commandement n° 7 : La gestion des versions tu centraliseras.	268
Commandement n° 8 : Comme la peste les dépendances optionnelles tu éviteras.	268
Commandement n° 9 : Les SNAPSHOT tu utiliseras.	269
Commandement n° 10 : L'IDE toujours tu privilégieras.	269
Conclusion	271

17	Épilogue	273
	Récapitulons	274
	Sortez de l'amateurisme	274
	Le mot de la fin	275
	Qui est qui ?	275
	Les membres francophones de l'équipe Maven	275
	Les membres de la communauté Java	281
	Post-scriptum	283
18	Lexique	285
	Le petit monde open-source	285
	Les concepts Maven	287
	Ceux qui font tourner Maven	291
	Et tout ce qui tourne autour... ..	293
	Liens utiles	294
	Index	295