

Version numérique

OFFERTE !

www.editions-eni.fr

Apache Maven

Maîtrisez l'infrastructure
d'un projet **Java EE**

Préface de Arnaud HÉRITIER

*Membre du « Project Management Committee »
du projet Apache Maven*

→ Informatique technique

Téléchargement
www.editions-eni.fr



ε
Collection

epsilon

Maxime GRÉAU

Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>. Saisissez la référence ENI
de l'ouvrage **EP3APAM** dans la zone de recherche et validez.
Cliquez sur le titre du livre puis sur le lien de téléchargement.

Chapitre 1

Présentation d'Apache Maven

1. Qu'est-ce qu'Apache Maven ?	17
1.1 Définition	17
1.2 Historique	18
1.3 Apache Maven sur mon projet ?	21
2. La communauté Maven	22
2.1 Les acteurs d'Apache Maven	22
2.1.1 Des rôles bien définis	23
2.1.2 Comment contribuer au projet ?	24
2.1.3 L'activité de la communauté	25
2.2 Les ressources de références	26
3. Installation d'Apache Maven	28
3.1 Pré-requis : la plate-forme Java	28
3.1.1 Présentation de Java SE (Standard Edition)	28
3.1.2 Installer le JDK (Java Development Kit) sous Windows	29
3.1.3 Installer le JDK sous GNU/Linux	35
3.2 Installer Apache Maven sous Windows	38
3.2.1 Où le trouver ?	38
3.2.2 Procédure d'installation	38
3.3 Installer Apache Maven sous Linux	40
3.3.1 Où le trouver ?	40
3.3.2 Procédure d'installation	40

4.	Le POM (Project Object Model)	41
4.1	Présentation	41
4.1.1	Format du fichier pour le POM	41
4.1.2	Exemple de POM minimal	42
4.1.3	Le modèle de données 4.0.0	44
4.2	Le Super POM	46
5.	Configurer l'installation d'Apache Maven	48
5.1	Qu'est-ce qu'un référentiel (repository) ?	48
5.2	Le fichier settings.xml	49
5.2.1	Où le trouver ?	49
5.2.2	Configuration de base	50
5.2.3	Configurer un proxy	51
5.3	Les options de configuration	51
5.3.1	La mémoire allouée à la JVM	51
5.3.2	La résolution des bibliothèques logicielles en parallèle	52
6.	Premières commandes Maven	53
6.1	Valider le POM	53
6.2	Afficher le POM complet	53
6.3	Gérer les erreurs	55
7.	Maven au cœur de l'infrastructure projet	57
7.1	Présentation du projet mvnbook-entry-plugins	57
7.2	Spécifications du projet mvnbook-entry-plugins	58
7.2.1	Les spécifications fonctionnelles	58
7.2.2	Les spécifications techniques	62
7.3	Objectifs du projet envers Apache Maven	63
7.3.1	Les objectifs fonctionnels	63
7.3.2	Les objectifs techniques	64

Chapitre 2

Définition et cycle de vie d'un projet

1. Qu'est-ce qu'un projet Apache Maven ?	65
1.1 Description d'un projet Apache Maven	65
1.1.1 Structure d'un projet simple	66
1.1.2 Identifier chaque projet	67
1.1.3 Produire un artefact par projet	68
1.2 Le modèle métier de mvnbook-entry-plugins	68
1.2.1 Présentation du modèle métier	68
1.2.2 Présentation de l'API Java Bean Validation	70
1.3 Créer un projet Maven	72
1.3.1 Créer un projet pour une archive Java	72
1.3.2 Exécuter les tests	78
1.4 Construire l'archive du projet	79
1.4.1 Créer une archive Java (JAR)	79
2. Le cycle de vie selon Apache Maven	80
2.1 Qu'est-ce que le cycle de vie d'un projet ?	80
2.2 Le cycle de vie pour le nettoyage du projet	81
2.3 Le cycle de vie par défaut	82
2.3.1 Définition	82
2.3.2 Le cycle de vie par défaut pour le JAR	84
2.4 Exécuter un traitement par un goal ou une phase ?	85
3. Le mode commande de Maven	87
3.1 Comment ça marche ?	87
3.2 La syntaxe	88
3.3 Les options principales en détail	89
3.3.1 Les options liées à la configuration Apache Maven	89
3.3.2 Les options liées à la sécurité	90
3.3.3 Les options liées aux référentiels	90
3.3.4 Les options pour les interactions avec l'exécuteur	91
3.3.5 Les options liées à la construction des projets	92
3.4 Les goals et les phases	92

4. Maven et les plugins.	93
4.1 Introduction aux plugins.	93
4.2 La résolution des plugins.	95
4.2.1 Le fichier components.xml	95
4.2.2 Les éléments du POM.	96
4.3 Les référentiels pour les plugins.	98
4.4 L'appel aux plugins	99
4.5 Configuration des plugins	99
4.5.1 maven-compiler-plugin	100
4.5.2 maven-jar-plugin.	100
4.6 Associer le traitement d'un plugin à une phase	102
4.6.1 Configuration par défaut.	102
4.6.2 Choisir une phase du cycle	105
5. Gérer l'encodage des fichiers d'un projet	106
5.1 Définition de l'encodage	106
5.1.1 ASCII	106
5.1.2 ISO-8859-1	107
5.1.3 CP1252	107
5.1.4 Unicode.	108
5.1.5 UTF-8	108
5.1.6 Bilan	109
5.2 Cas concret de problèmes liés à l'encodage.	109
5.2.1 Mise en place de l'exemple	109
5.2.2 Contexte de l'exemple.	110
5.2.3 Mise en œuvre de l'exemple.	112
5.2.4 L'utilité de la configuration Maven	117
5.3 Conclusion	118
6. En résumé.	120

Chapitre 3

Organisation et dépendances des projets

1. La gestion des dépendances.	121
1.1 Définition	121
1.1.1 Les éléments du POM concernés	122
1.1.2 Illustration des dépendances	123
1.2 Les champs d'application (scope) des dépendances.	125
1.2.1 Définition	125
1.2.2 Le scope compile	125
1.2.3 Le scope provided	127
1.2.4 Le scope runtime.	128
1.2.5 Le scope test	129
1.2.6 Le scope system.	130
1.2.7 Le scope import.	131
1.3 La persistance du projet mvnbook-persistence	131
1.3.1 Présentation de l'API JPA 2.0	131
1.3.2 Implémentation de JPA 2.0	132
1.3.3 Définition de l'injection de dépendances en Java	135
1.3.4 Apache Maven et l'injection de dépendances	135
1.3.5 Implémentation de l'injection de dépendance avec Guice.	136
2. La résolution des dépendances	137
2.1 Comment sont identifiées les dépendances ?	137
2.1.1 La transitivité	137
2.1.2 L'influence du scope sur la transitivité des dépendances	139
2.1.3 Les coordonnées complètes d'une dépendance	141
2.2 Gérer les conflits de dépendances	142
2.2.1 Dépendances directes	142
2.2.2 Dépendances transitives	143
2.2.3 Cas du projet mvnbook-persistence.	144
2.3 À quel moment sont résolues les dépendances ?	149

3. L'héritage et les projets multi-modules	151
3.1 Présentation des projets mvnbook-service et mvnbook-webapp	151
3.2 Héritage d'un POM commun	153
3.2.1 Le type POM	153
3.2.2 Mise en place du Corporate POM	154
3.3 Organisation des projets en modules	156
3.3.1 Le besoin	156
3.3.2 Définition du reactor	157
3.3.3 Mise en place sur le projet mvnbook-entry-plugins ...	158
3.3.4 Le mode commande adapté au reactor	161
3.4 Choisir une organisation de projets	166
3.5 Organiser les dépendances	167
3.5.1 Factoriser les dépendances	167
3.5.2 Redéfinir une dépendance	168
3.5.3 Centraliser les informations pour les dépendances ...	168
3.5.4 Centraliser les informations pour les plugins	172
4. Les propriétés du POM	174
4.1 Définition	174
4.1.1 Les propriétés disponibles	174
4.1.2 Créer une propriété	175
4.1.3 Les propriétés des plugins	176
4.2 Héritage et interactions des propriétés	177
4.2.1 Propriétés pour les versions	177
4.3 Filtrer les fichiers ressources	179
4.3.1 Gérer la version et la date de construction du projet ..	180
5. En résumé	182

Chapitre 4

Les artefacts pour Java EE et les profils Maven

1. La plate-forme Java EE	183
1.1 Introduction à Java EE	183
1.1.1 Java EE 5	184
1.1.2 Java EE 6	184
1.2 Conteneurs Java EE	185
1.2.1 Java EE et projets Apache Maven	185
1.2.2 Apache Tomcat	186
1.2.3 JBoss AS (Application Server)	190
1.3 Présentation de JAX-WS et JAXB	192
1.3.1 Présentation de JAXB	192
1.3.2 Mise en œuvre de JAXB	193
1.3.3 Présentation de JAX-WS	194
1.3.4 Mise en œuvre de JAX-WS	195
2. Les profils Apache Maven	199
2.1 Présentation générale	199
2.1.1 Pourquoi utiliser les profils ?	199
2.2 Les éléments du POM et du fichier settings.xml	200
2.2.1 Éléments du profil dans le POM	200
2.2.2 Éléments du profil dans le settings.xml	201
2.3 Intégrer les profils dans un processus Maven	202
2.3.1 Activer les profils par leurs identifiants	202
2.3.2 Les autres modes d'activation des profils	202
2.3.3 Faut-il activer un profil par défaut ?	206
3. Les types d'artefacts pour les projets Apache Maven	209
3.1 Les types d'artefacts par défaut	209
3.2 Les archives pour les applications Web (WAR)	210
3.2.1 Présentation	210
3.2.2 Apache Maven et le format WAR	210
3.2.3 Les projets WAR	212
3.2.4 Déployer un WAR dans Apache Tomcat	215

3.3	Les archives pour les applications d'entreprise (EAR)	220
3.3.1	Présentation	220
3.3.2	Apache Maven et le format EAR	222
3.3.3	Les projets EAR	223
3.3.4	Déployer un EAR pour JBoss AS	226
3.3.5	Isoler les EAR dans JBoss AS	232
3.4	Les autres formats d'artefacts	235
3.4.1	Créer un JAR encapsulant ses dépendances	235
3.4.2	Créer une archive ZIP	240
4.	En résumé.	245

Chapitre 5

Apache Maven dans un contexte professionnel

1.	Introduction	247
1.1	Le contexte	247
1.2	Qu'est-ce qu'un SNAPSHOT ?	249
1.3	Qu'est-ce qu'une Release ?	249
2.	Apache Maven et les SCM	250
2.1	Qu'est-ce qu'un SCM ?	250
2.2	Configuration des SCM pour Apache Maven	251
2.2.1	Le plugin maven-scm-plugin	252
2.3	Configuration pour Apache Subversion (SVN)	253
2.3.1	Présentation de Subversion	253
2.3.2	Configuration du projet pour SVN	254
2.4	Maven et le SCM dans un contexte professionnel	257
3.	Les environnements de développement	257
3.1	Les IDE et Apache Maven	257
3.2	L'environnement de développement Eclipse	258
3.2.1	Présentation	258
3.2.2	Installer Eclipse	258
3.2.3	Configurer Eclipse	259

3.2.4	Installer un plugin Eclipse pour SVN.	263
3.3	Le plugin Maven pour Eclipse : m2eclipse	266
3.3.1	Présentation	266
3.3.2	Installation	266
3.3.3	Configuration	268
3.3.4	Utilisation.	270
4.	Apache Maven et les tests	275
4.1	Introduction aux tests.	275
4.2	Les tests unitaires	276
4.2.1	Exécuter un seul test d'une classe de tests.	276
4.3	Les tests d'intégration	278
4.3.1	Partager des classes de tests.	280
4.3.2	Exécuter les tests d'intégration avec un container Tomcat.	282
4.4	Configurer l'exécution des tests.	286
5.	Les référentiels Apache Maven	286
5.1	Présentation	286
5.1.1	Le référentiel local.	287
5.1.2	Les référentiels distants.	287
5.2	Cycle de vie et accès aux référentiels.	290
5.3	Installation dans un référentiel local	291
5.4	Déploiement dans un référentiel distant	292
5.4.1	Les protocoles pour le déploiement	292
5.4.2	Déploiement dans les référentiels avec WebDAV.	296
5.4.3	Sécuriser les déploiements	303
5.4.4	Bilan	305
5.5	Gestion des artefacts dans le référentiel.	306
5.5.1	L'organisation des artefacts en cours de développement	306
5.5.2	L'organisation des artefacts en version Release.	311
5.5.3	Les artefacts déployés	312
5.5.4	Purger un référentiel	315
5.5.5	Travailler en mode déconnecté	315

5.6	Les problèmes d'accès réseau	315
5.7	Bilan	316
6.	Les gestionnaires de référentiels Maven	317
6.1	Présentation	317
6.2	Les avantages	317
6.3	L'infrastructure du projet avec un gestionnaire	318
6.4	Le gestionnaire de référentiels Nexus.	319
6.4.1	Présentation	319
6.4.2	Installation	319
6.4.3	Configuration	322
6.4.4	Validation	329
6.4.5	Réaliser des traitements automatisés sur les référentiels	330
7.	L'intégration continue.	331
7.1	Introduction	331
7.2	Les avantages de l'intégration continue	332
7.3	Jenkins CI (Continuous Integration)	333
7.3.1	Présentation	333
7.3.2	Installation	334
7.3.3	Configuration	337
7.3.4	Déployer des artefacts dans Nexus	339
7.3.5	Installer un plugin.	340
8.	La phase de livraison du projet : la Release	340
8.1	Introduction	340
8.2	Gestion des numéros de version	341
8.2.1	Définition générale	341
8.2.2	La gestion des numéros de version avec Maven	341
8.2.3	Plugin pour les numéros de version	342
8.3	Processus de Release avec Apache Maven	343
8.3.1	Optimiser la réussite de la Release.	345
8.4	Étapes pré-requises au processus de Release.	345
8.4.1	Vérifier les dépendances	345

8.4.2	Tester les versions SNAPSHOT du projet	347
8.4.3	Valider les paramètres techniques	347
8.4.4	Tester le processus de Release	348
8.5	Release à partir de l'intégration continue	348
8.5.1	Configurer Jenkins	349
8.5.2	Lancer la Release	349
8.5.3	Vérifier le déploiement	351
8.5.4	Bilan	352
9.	En résumé.	352

Chapitre 6

Rapports Maven et mesure de la qualité

1.	Présentation des rapports avec Maven 3	355
1.1	Introduction	355
1.2	Les éléments du POM	356
1.3	Configurer l'infrastructure de déploiement des sites	357
1.3.1	Configuration du serveur Web.	358
1.3.2	Configurer le Corporate POM et le settings.xml	359
1.3.3	Adapter la configuration à l'infrastructure.	360
1.4	Le format des rapports	362
2.	Générer le site Web du projet	362
2.1	Le cycle de vie pour le site du projet	362
2.2	Le plugin de base pour la génération du site Web	363
3.	Générer les rapports Apache Maven	366
3.1	Rapports liés au code source	366
3.1.1	Introduction et configuration	366
3.1.2	Code source en HTML	369
3.1.3	Documentation du code Java (javadoc)	371
3.1.4	Analyse avec PMD	372
3.1.5	Analyse avec Checkstyle	374

3.2	Rapports sur les tests du projet	375
3.2.1	Rapport d'exécution de tests	375
3.2.2	Rapport de couverture de tests	376
4.	La mesure de la qualité des projets	378
4.1	Qu'est-ce que la mesure de qualité d'un projet ?	378
4.2	La mesure de la qualité des projets Maven avec Sonar	379
4.2.1	Présentation de Sonar	379
4.2.2	Installation	380
4.2.3	Configuration de Sonar	382
4.2.4	Configuration de Maven	383
4.2.5	Exemples avec le projet Apache Maven 3	385
5.	En résumé.	387
	Index	389