

Avant-propos

1. Public concerné	13
2. Prérequis	14
3. Objectifs	14
4. Organisation du livre	15
5. Sur le Web	16

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 ?	23

2.1.3 L'activité de la communauté	24
2.2 Les ressources de références	25
3. Installation d'Apache Maven	27
3.1 Prérequis : la plate-forme Java	27
3.1.1 Présentation de Java SE	27
3.1.2 Installer le JDK sous Windows	29
3.1.3 Installer le JDK sous GNU/Linux	33
3.2 Installer Apache Maven sous Windows	35
3.2.1 Où le trouver ?	35
3.2.2 Procédure d'installation	36
3.3 Installer Apache Maven sous Linux	37
3.3.1 Où le trouver ?	37
3.3.2 Procédure d'installation	37
4. Le POM (Project Object Model)	39
4.1 Présentation	39
4.1.1 Format du fichier pour le POM	39
4.1.2 Exemple de POM minimal	39
4.1.3 Le modèle de données 4.0.0	41
4.2 Le Super POM	44

5. Configurer l'installation d'Apache Maven	46
5.1 Qu'est-ce qu'un référentiel (repository) ?	46
5.2 Le fichier settings.xml	47
5.2.1 Où le trouver ?	47
5.2.2 Configuration de base	48
5.2.3 Configurer un proxy	48
5.3 Les options de configuration	49
5.3.1 La mémoire allouée à la JVM	49
5.3.2 La résolution des bibliothèques logicielles en parallèle	50
6. Premières commandes Maven	51
6.1 Valider le POM	51
6.2 Afficher le POM complet	52
6.3 Gérer les erreurs	54
7. Maven au cœur de l'infrastructure projet	55
7.1 Présentation du projet "banque"	55
7.2 Spécifications du projet "banque"	57
7.2.1 Les spécifications fonctionnelles	57
7.2.2 Les spécifications techniques	57
7.3 Objectifs du projet envers Apache Maven	59

Définition et cycle de vie d'un projet

1. Qu'est-ce qu'un projet Apache Maven ?	61
1.1 Description d'un projet Apache Maven	61
1.1.1 Structure d'un projet simple	62
1.1.2 Identifier chaque projet	63
1.1.3 Produire un artefact par projet	64
1.2 Le modèle métier du projet "banque"	64
1.2.1 Présentation du modèle métier	64
1.2.2 Présentation de l'API Java Bean Validation	65
1.3 Créer un projet Maven	67
1.3.1 Créer un projet pour une archive Java	67
1.3.2 Exécuter les tests	73
1.4 Créer l'archive Java (JAR) du projet	73
2. Le cycle de vie selon Apache Maven	75
2.1 Qu'est-ce que le cycle de vie d'un projet ?	75
2.2 Le cycle de vie pour le nettoyage du projet	76
2.3 Le cycle de vie par défaut	77
2.3.1 Définition	77

2.3.2 Le cycle de vie par défaut pour le JAR	79
2.4 Exécuter un traitement par un goal ou une phase ?	81
3. Le mode commande de Maven	83
3.1 Comment ça marche ?	83
3.2 La syntaxe	84
3.3 Les options principales en détail	84
3.3.1 Les options liées à la configuration Apache Maven	85
3.3.2 Les options liées à la sécurité	86
3.3.3 Les options liées aux référentiels	86
3.3.4 Les options pour les interactions avec l'exécuteur	86
3.3.5 Les options liées à la construction des projets	87
3.4 Les goals et les phases	88
4. Maven et les plugins	88
4.1 Introduction aux plugins	88
4.2 La résolution des plugins	90
4.2.1 Le fichier components.xml	90
4.2.2 Les éléments du POM	91
4.3 Les référentiels pour les plugins	93
4.4 L'appel aux plugins	94
4.5 Configuration des plugins	94

4.5.1 maven-compiler-plugin	95
4.5.2 maven-jar-plugin	95
4.6 Associer le traitement d'un plugin à une phase	96
4.6.1 Configuration par défaut	97
4.6.2 Choisir une phase du cycle	97
5. Gérer l'encodage des fichiers d'un projet	100
5.1 Définition de l'encodage	102
5.1.1 ASCII	102
5.1.2 ISO-8859-1	103
5.1.3 CP1252	103
5.1.4 Unicode	103
5.1.5 UTF-8	104
5.1.6 Bilan	105
5.2 Cas concret de problèmes liés à l'encodage	105
5.2.1 Mise en place de l'exemple	105
5.2.2 Contexte de l'exemple	106
5.2.3 Mise en œuvre de l'exemple	108
5.2.4 L'utilité de la configuration Maven	113
5.3 Conclusion	114

6. En résumé	115
Organisation et dépendances des projets	
1. La gestion des dépendances	117
1.1 Définition et problématiques de la gestion des dépendances	117
1.1.1 Les éléments du POM concernés	118
1.1.2 Illustration des dépendances	119
1.2 Les champs d'application (scope) des dépendances	121
1.2.1 Définition	121
1.2.2 Le scope compile	122
1.2.3 Le scope provided	123
1.2.4 Le scope runtime	124
1.2.5 Le scope test	125
1.2.6 Le scope system	125
1.2.7 Le scope import	126
1.3 La persistance du projet banque-persistance	126
1.3.1 Présentation de l'API JPA 2	126
1.3.2 Implémentation de JPA 2	127
1.3.3 Définition de l'injection de dépendances en Java	129
1.3.4 Apache Maven et l'injection de dépendances	

1.3.5 Implémentation de l'injection de dépendances avec Spring Framework	130
	130
2. La résolution des dépendances	133
2.1 Comment sont identifiées les dépendances ?	133
2.1.1 La transitivité	133
2.1.2 L'influence du scope sur la transitivité des dépendances	136
2.1.3 Les coordonnées complètes d'une dépendance	138
2.2 Gérer les conflits de dépendances	138
2.2.1 Dépendances directes	139
2.2.2 Dépendances transitives	139
2.2.3 Cas concret d'un conflit de dépendances	140
2.3 À quel moment sont résolues les dépendances ?	145
3. L'héritage et les projets multimodules	146
3.1 Présentation des projets banque-metier et banque-web	146
3.2 Héritage d'un POM commun	147
3.2.1 Le type POM	147
3.2.2 Mise en place du Corporate POM	148
3.3 Organisation des projets en modules	150
3.3.1 Le besoin	150
3.3.2 Définition du reactor	152

3.3.3 Mise en place sur le projet banque	152
3.3.4 Le mode commande adapté au reactor	155
3.4 Choisir une organisation de projets	160
3.5 Organiser les dépendances	161
3.5.1 Factoriser les dépendances	161
3.5.2 Redéfinir une dépendance	162
3.5.3 Centraliser les informations pour les dépendances	162
3.5.4 Centraliser les informations pour les plugins	166
4. Les propriétés du POM	168
4.1 Définition	168
4.1.1 Les propriétés disponibles	169
4.1.2 Créer une propriété	169
4.1.3 Les propriétés des plugins	170
4.2 Héritage et interactions des propriétés	171
4.2.1 Propriétés pour les versions	172
4.3 Filtrer les fichiers ressources	174
4.3.1 Gérer la version et la date de construction du projet	174
5. En résumé	176

Les artefacts pour Java EE et les profils Maven

1. La plateforme Java EE	177
1.1 Introduction à Java EE	177
1.1.1 Historique de Java EE	178
1.1.2 Java EE aujourd'hui	179
1.2 Conteneurs Java EE	179
1.2.1 Java EE et projets Apache Maven	179
1.2.2 Apache Tomcat	180
1.2.3 WildFly	184
1.3 Présentation de JAX-WS et JAXB	187
1.3.1 Présentation de JAXB	187
1.3.2 Mise en œuvre de JAXB	187
1.3.3 Présentation de JAX-WS	188
1.3.4 Mise en œuvre de JAX-WS	188
1.4 Le projet de service web et son projet client	189
1.4.1 Le projet de service web : banque-webservice	189
1.4.2 Implémentation du projet de service web : banque-webservice	192
1.4.3 Implémentation du projet de client de service web : banque-client-webservice	194
2. Les profils Apache Maven	196
2.1 Présentation générale	196

2.1.1 Pourquoi utiliser les profils ?	196
2.2 Les éléments du POM et du fichier settings.xml	197
2.2.1 Éléments du profil dans le POM	197
2.2.2 Éléments du profil dans le fichier settings.xml	198
2.3 Intégrer les profils dans un processus Maven	199
2.3.1 Activer les profils par leurs identifiants	199
2.3.2 Les autres modes d'activation des profils	200
2.3.3 Faut-il activer un profil par défaut ?	203
3. Les types d'artefacts pour les projets Apache Maven	207
3.1 Les types d'artefacts par défaut	207
3.2 Les archives pour les applications web (WAR)	208
3.2.1 Présentation	208
3.2.2 Apache Maven et le format WAR	208
3.2.3 Les projets WAR	210
3.2.4 Déployer un WAR dans Apache Tomcat	213
3.2.5 Configurer le POM	214
3.3 Les archives pour les applications d'entreprise (EAR)	219
3.3.1 Présentation	219
3.3.2 Apache Maven et le format EAR	220
3.3.3 Les projets EAR	221

3.3.4 Déployer un EAR pour WildFly	225
3.3.5 Configurer les sous-systèmes de WildFly	230
3.4 Les autres formats d'artefacts	232
3.4.1 Créer un JAR encapsulant ses dépendances	233
3.4.2 Créer une archive ZIP	237
4. En résumé	243
Apache Maven dans un contexte professionnel	
1. Introduction	245
1.1 Le contexte	245
1.2 Qu'est-ce qu'un SNAPSHOT ?	247
1.3 Qu'est-ce qu'une Release ?	247
2. Apache Maven et les SCM	248
2.1 Qu'est-ce qu'un SCM ?	248
2.2 Configuration des SCM pour Apache Maven	249
2.2.1 Le plugin maven-scm-plugin	250
2.3 Configuration pour Apache Subversion (SVN)	251
2.3.1 Présentation de Subversion	251
2.3.2 Configuration du projet pour SVN	251

2.4 Configuration pour Git	252
2.4.1 Présentation de Git	255
2.4.2 Configuration du projet pour Git	255
2.5 Maven et le SCM dans un contexte professionnel	259
3. Les environnements de développement	259
3.1 Les IDE et Apache Maven	259
3.2 L'environnement de développement Eclipse	260
3.2.1 Présentation	260
3.2.2 Installer Eclipse	260
3.2.3 Configurer Eclipse	261
3.2.4 Installer un plugin Eclipse pour SVN	265
3.2.5 Eclipse et Git	268
3.3 Le support de Maven dans Eclipse : m2eclipse	270
3.3.1 Présentation	270
3.3.2 Configuration	270
3.3.3 Utilisation	273
4. Apache Maven et les tests	277
4.1 Introduction aux tests	277
4.2 Les tests unitaires	278

4.2.1 Exécuter un seul test d'une classe de tests	278
4.3 Les tests d'intégration	280
4.3.1 Partager des classes de tests	282
4.3.2 Exécuter les tests d'intégration avec un container Tomcat	284
4.4 Configurer l'exécution des tests	288
5. Les référentiels Apache Maven	288
5.1 Présentation	288
5.1.1 Le référentiel local	289
5.1.2 Les référentiels distants	289
5.2 Cycle de vie et accès aux référentiels	292
5.3 Installation dans un référentiel local	294
5.4 Déploiement dans un référentiel distant	294
5.4.1 Les protocoles pour le déploiement	295
5.4.2 Déploiement dans les référentiels avec WebDAV	298
5.4.3 Sécuriser les déploiements	306
5.4.4 Bilan	308
5.5 Gestion des artefacts dans le référentiel	309
5.5.1 L'organisation des artefacts en cours de développement	309
5.5.2 L'organisation des artefacts en version Release	313
5.5.3 Les artefacts déployés	314

5.5.4 Purger un référentiel	316
5.5.5 Travailler en mode déconnecté	317
5.6 Les problèmes d'accès réseau	317
5.7 Bilan	318
6. Les gestionnaires de référentiels Maven	318
6.1 Présentation	318
6.2 Les avantages	319
6.3 L'infrastructure du projet avec un gestionnaire	320
6.4 Le gestionnaire de référentiels Nexus Repository	320
6.4.1 Présentation	320
6.4.2 Installation	321
6.4.3 Configuration	324
6.4.4 Validation	328
6.4.5 Réaliser des traitements automatisés sur les référentiels	328
7. L'intégration continue	329
7.1 Introduction	329
7.2 Les avantages de l'intégration continue	330
7.3 Jenkins CI (Continuous Integration)	331
7.3.1 Présentation	331
7.3.2 Installation	331

7.3.3 Configuration	332
7.3.4 Déployer des artefacts dans Nexus Repository	335
7.3.5 Installer un plugin	337
8. La phase de livraison du projet : la Release	338
8.1 Introduction	338
8.2 Gestion des numéros de version	339
8.2.1 Définition générale	339
8.2.2 La gestion des numéros de version avec Maven	339
8.2.3 Plugin pour les numéros de version	340
8.3 Processus de Release avec Apache Maven	341
8.3.1 Optimiser la réussite de la Release	342
8.4 Étapes prérequis au processus de Release	343
8.4.1 Vérifier les dépendances	343
8.4.2 Tester les versions SNAPSHOT du projet	344
8.4.3 Valider les paramètres techniques	345
8.4.4 Tester le processus de Release	346
8.5 Release à partir de l'intégration continue	346
8.5.1 Configurer Jenkins	346
8.5.2 Lancer la Release	347
8.5.3 Vérifier le déploiement	347

8.5.4 Bilan	347
	348
9. En résumé	348
	348
Rapports Maven et mesure de la qualité	
1. Présentation des rapports avec Maven 3	351
1.1 Introduction	351
1.2 Les éléments du POM	352
1.3 Configurer l'infrastructure de déploiement des sites	353
1.3.1 Configuration du serveur web	353
1.3.2 Configurer le Corporate POM et le fichier settings.xml	355
1.3.3 Adapter la configuration à l'infrastructure	356
1.4 Le format des rapports	358
2. Générer le site web du projet	358
2.1 Le cycle de vie pour le site du projet	358
2.2 Le plugin de base pour la génération du site web	359
3. Générer les rapports Apache Maven	363
3.1 Rapports liés au code source	363

3.1.1 Introduction et configuration	363
3.1.2 Code source en HTML	366
3.1.3 Documentation du code Java (javadoc)	368
3.1.4 Analyse avec PMD	370
3.1.5 Analyse avec Checkstyle	372
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 avec Cobertura	377
3.2.3 Rapport de couverture de tests avec JaCoCo	379
4. La mesure de la qualité des projets	382
4.1 Qu'est-ce que la mesure de qualité d'un projet ?	382
4.2 La mesure de la qualité des projets Maven avec SonarQube	383
4.2.1 Présentation de SonarQube	383
4.2.2 Installation	384
4.2.3 Configuration de SonarQube	386
4.2.4 Configuration de Maven	386
5. En résumé	389

Index

391