

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.

Préface



Histoire de Maven

Maven est né au sein du projet Jakarta Alexandria. Ce projet, aujourd'hui arrêté, fût le terreau non seulement de Maven mais aussi d'autres projets comme Gump et Forrest. Le premier import des sources du prototype eu lieu en août 2001. Maven vécu pendant environ 5 mois au sein d'Alexandria avant de rejoindre sa nouvelle adresse dans le projet Turbine.

Bien que Maven fit ses débuts dans Alexandria, le test en grandeur nature fut le projet Turbine. Turbine tentait de découpler ses couches persistance, service, et présentation web dans des builds séparés et j'étais exaspéré de devoir gérer de multiples scripts de compilation très semblables. Il n'y avait pas de moyen simple à cette époque pour créer des modèles de scripts Ant, chaque build semblait différent. Je trouvais cela incroyablement frustrant et futile : personne n'était intéressé de savoir comment la construction s'effectuait tant qu'elle fonctionnait et qu'elle était facile à utiliser. L'infrastructure d'un projet est incroyablement importante, mais sa valeur réside dans l'application développée. En conséquence le build est souvent négligé et tend à vous lâcher quand vous en avez le plus besoin, par exemple lors de la préparation d'une livraison ou lorsque plusieurs personnes interviennent sur le projet. Dans le projet Jakarta, il y a plusieurs années, il était rare qu'un build Ant fonctionne tel quel.

Les développeurs de Turbine ont souffert lorsque j'ai essayé de faire fonctionner Maven, ce que je regrette, mais j'imagine mal comment un nouveau projet peut démarrer et survivre si personne ne souffre. Je pensais que c'était pour leur propre bien (je suis connu pour avoir une opinion ou deux sur le sujet) et, après quelques grincements de dents, Maven est arrivé à maturité. Cela me rappelle une de mes citations favorite de Ralph Johnson et Don Roberts dans *Patterns for Evolving Frameworks* :

Les gens créent de l'abstraction en généralisant des exemples concrets. Toute tentative de définir l'abstraction correcte sur papier sans développer et exécuter un système réel est condamnée à l'échec. Personne n'est aussi doué. Un framework est une conception réutilisable, donc il se construit en regardant les choses dont il est sensé être le modèle. Plus vous avez d'exemples sous la main, plus le framework pourra être générique.

Je ne savais pas vraiment à quoi le résultat final ressemblerait, mais je savais qu'il devait y avoir une meilleure façon de faire. Pour commencer, je savais ce que je voulais :

- un modèle pour le projet, pour qu'il n'y ait qu'un seul endroit où aller chercher l'information relative au projet ;
- une structure standardisée pour qu'il ne soit pas nécessaire d'aller à la pêche aux bibliothèques, au code source et à la documentation.

La chose suivante que je notais était que tous les JAR dont nous dépendions étaient stockés sous CVS. Nous perdions de la place en conservant plusieurs copies de bibliothèques comme Xerces. À chaque fois qu'une nouvelle version de Xerces apparaissait, je devais mettre à jour chaque projet. Mais plus grave, sans gestion déclarative il n'y avait aucun moyen d'effectuer une analyse. Les gens ont tendance à sous-estimer l'importance d'une gestion déclarative. Ils se disent que c'est si simple de placer les bibliothèques dans le gestionnaire de sources, mais essayez de décomposer votre gros projet poubelle en composants réutilisables et maintenables, ou d'analyser ce qui sera nécessaire à l'exécution entre toutes vos applications avec des dépendances communes dans la chaîne et vous serez bien ennuyé. La vraie puissance de la gestion déclarative ne tient pas à l'économie de quelques octets de disque (quoique cela puisse être significatif si on n'y prend pas garde) mais à la possibilité d'analyse. Une fois un graphe de dépendances en place, tout devient possible. Mais retour à l'histoire : maintenant que la gestion déclarative des dépendances existait, il fallait rendre plus simple le partage des librairies. Juste après avoir créé Maven nous avons créé le référentiel Maven, un référentiel de librairies qui est utilisé aujourd'hui par la plupart des développements Java.

Beaucoup de personnes ont eu des soucis avec Maven 1, mais il fonctionnait généralement bien, et tous les outils dans leur première génération souffrent de divers défauts. La seule façon d'aller au delà est d'en prendre de la graine et de créer quelque chose de mieux pour le coup d'après. Nous avons créé Maven 2.0, et après plusieurs années nous sommes sur le point de publier Maven 3.0. Avec tous les retours que les développeurs ont reçu de la part de l'incroyable communauté des utilisateurs de Maven, je pense que nous sommes arrivés à quelque chose de solide sur lequel nous pouvons itérer. Ne vous inquiétez pas : Maven 3.0 est 100 % compatible avec l'existant en Maven 2.0 :-). Nous avons désormais une compréhension très complète sur comment les organisations construisent leurs applications, depuis le développement en passant par les tests et jusqu'à la mise en production. Ce sont toutes ces connaissances qui ont été utilisées pour créer les bases de Maven 3.0.

À propos de ce livre

Nicolas et Arnaud ont choisi, avec une approche légère et récréative, de proposer un guide aux utilisateurs novices, basé sur l'histoire d'une start-up technologique qui fait le choix d'utiliser Maven. Le livre couvre toutes les phases du projet, de son origine jusqu'à l'étape finale de livraison et de déploiement d'un produit complet. Les lecteurs découvrent progressivement les bonnes pratiques de Maven à travers les utilisations que nos experts en font et bénéficient de techniques puissantes qu'il faudrait sans cela des mois pour apprendre.

Le livre *Apache Maven* n'est pas seulement une introduction pratique à Maven, mais c'est aussi un guide où chaque leçon est basée sur un exemple. Je pense qu'Arnaud et Nicolas ont réalisé un super travail, demandant beaucoup d'efforts. Je recommande sans hésitation cet ouvrage à toute personne s'intéressant à Maven : c'est un ouvrage de référence et de grande valeur pour la communauté Maven.

Jason Van Zyl,

Fondateur du projet Apache Maven

Avant-propos



L'écriture d'un ouvrage technique n'est pas une tâche triviale, car il est facile de perdre le lecteur dans une avalanche de concepts théoriques ou de s'égarer dans des détails non fondamentaux. Décrire un outil comme Maven, ou tout simplement le définir clairement, tout en restant accessible à tous, est encore plus délicat : soit on reste trop vague, et le lecteur n'a plus qu'à attendre le Chapitre 5 pour commencer à apprendre quelque chose de concret, soit on s'embarque dans de longues explications de principes et de concepts et le lecteur n'attendra jamais ce même Chapitre 5.

Pour être honnête, je dois dire que les premières ébauches de cet ouvrage sont immanquablement tombées dans ces travers, ce qui annonçait un livre bien peu pertinent pour les utilisateurs, qu'ils soient novices ou déjà expérimentés. Lorsque j'ai soumis les premiers jets de ce projet à Arnaud, il m'en a rapidement fait la remarque et nous nous sommes accordés sur la forme que nous voulions donner à ce livre.

Mon objectif est de communiquer ma passion autour de ce projet open-source qu'est Maven, lequel réunit des développeurs aux parcours très différents. Les rencontres que j'ai faites dans cette communauté ont forgé mon approche de l'informatique. Avec cette motivation, établir un dictionnaire impersonnel Maven-Français était exclu ; aussi j'ai rapidement choisi, en accord avec Arnaud, de privilégier une approche aussi didactique que possible, bâtie sur des exemples concrets issus de ma propre expérience du terrain.

Il est difficile de sensibiliser les utilisateurs aux enjeux que Maven tente de gérer, alors qu'ils y sont pourtant confrontés en permanence. Situation intéressante où tout le monde rencontre un problème, mais, faute de mettre un nom dessus et d'en évaluer l'importance, celui-ci reste latent tout au long de la vie du projet, amenant parfois à des situations critiques. Nous allons suivre ensemble la vie d'un projet fictif, bien que largement inspiré de situations réelles. Il passera par toutes les phases, du prototype écrit sur un coin de table à l'application stratégique d'entreprise de grande envergure, ce qui nous permettra de couvrir un très large éventail de situations.

Plutôt que de décrire le rôle de Maven sur un projet, ou de vous accabler par un long exposé théorique sur ses concepts, je préfère au travers de cette démonstration un peu

romancée vous montrer les difficultés concrètes auxquelles Maven s'attaque. Sur la base de ces exemples, parfois volontairement excessifs, je souhaite vous démontrer de manière ludique les avantages que Maven peut apporter à vos projets. Malgré les caricatures proposées, de nombreuses situations vous sembleront familières. Derrière la fiction se cachent des cas bien réels, que je n'ai fait qu'amplifier, et beaucoup auront des points communs avec vos propres difficultés. Ce parallèle vous donnera une image réaliste de Maven et des conseils applicables dans les meilleurs délais.

J'espère que vous apprécierez ce choix et que vous tirerez un enseignement pratique du texte qui suit. En particulier, j'aimerais qu'arrivé au bout de votre lecture vous soyez conscient des objectifs visés par Maven, de sa philosophie et des raisons pour lesquelles il devient un élément clé de la boîte à outils du développeur. Enfin, je souhaite réussir à vous transmettre mon enthousiasme pour ce projet libre, auquel vous pouvez participer en rejoignant le forum pour y exposer vos interrogations, apporter de nouvelles idées, proposer des contributions de toutes sortes et participer à l'amélioration générale de cet outil. Arnaud et moi avons commencé de cette façon avant de passer "de l'autre côté du miroir", mais au quotidien nous restons comme vous, avant tout, des utilisateurs de Maven, soucieux de disposer d'un outil pertinent et productif.

Nicolas De loof



Lorsque Nicolas m'a contacté pour écrire un ouvrage sur Maven en français, j'ai commencé par me demander si cela en valait la peine. Certes, la documentation du produit est critiquable. Elle est très dispersée, et il est souvent difficile de trouver l'information utile lorsqu'on ne sait pas où la chercher entre le site web du projet¹, ses nombreux plugins et son wiki². Pourtant, il existe désormais deux ouvrages en anglais disponibles gratuitement sur la Toile pour combler ces manques : *Better Builds with Maven*³, publié en 2006, et *Maven : The Definitive Guide*⁴, publié en 2007 et régulièrement mis à jour. Alors qu'apporter de plus qu'une simple traduction en français de ces ouvrages ?

Après de nombreuses années à utiliser et à préconiser Maven dans des contextes variés, j'avais envie de partager tout ce que j'avais pu emmagasiner comme bonnes pratiques et pointer sur les mauvaises que j'avais pu rencontrer. C'est sur ce principe que nous avons commencé avec Nicolas à bâtir le squelette de cet ouvrage. Fondé sur un projet

-
1. <http://maven.apache.org>.
 2. <http://docs.codehaus.org/display/MAVENUSER>.
 3. MaestroDev (<http://www.maestrodev.com>).
 4. Sonatype, Inc. (<http://www.sonatype.com>).

fictif, il retrace nos expériences ainsi que celles des personnes que nous avons croisées sur notre chemin et permet d'expliquer les enjeux de Maven dans un projet et dans une entreprise. Même si nous n'avons pas recherché l'exhaustivité dans les cas traités, tellement ils peuvent être nombreux, nous avons essayé de faire apparaître les plus fréquents ou les plus épineux que nous ayons eus à résoudre. Nous avons axé nos efforts sur la présentation et la compréhension des concepts plutôt que sur le détail du paramétrage, lequel peut évoluer périodiquement.

J'espère que cet ouvrage saura autant vous divertir que vous former sur cet outil complet afin qu'il ne soit plus jamais complexe à vos yeux.

Arnaud Héritier

Contenu

Cet ouvrage se compose de quatre parties :

- La première, du Chapitre 1 au Chapitre 5, aborde les concepts fondamentaux de Maven et leur mise en œuvre pratique. Nous avons choisi de mettre en scène de manière très explicite et souvent exagérée les problèmes que Maven tente de prendre en charge, afin que cette première partie soit aussi didactique que possible.
- La deuxième, du Chapitre 6 au Chapitre 10, exploite des fonctionnalités plus avancées de Maven pour traiter des besoins orientés "gros projets d'entreprise" mais tout aussi délicats. Cette partie s'adresse typiquement aux développeurs intervenant sur des projets JEE (*Java Enterprise Edition*) en entreprise.
- La troisième regroupe les Chapitres 11 à 15 et couvre des facettes plus spécialisées et moins mises en avant de Maven, mais que nous considérons comme tout aussi essentielles. Vous verrez alors que Maven ne se résume pas comme on le lit souvent à "un outil de compilation".
- Pour terminer cet ouvrage le Chapitre 16 sera l'occasion de résumer les éléments clés présentés, de vous donner nos recommandations, bonnes et mauvaises pratiques à connaître pour tirer le meilleur de Maven. Par ailleurs, nous nous essayerons à l'exercice acrobatique de la boule de cristal en vous présentant l'avenir du projet Maven. Nous indiquerons comment aller au-delà de ce livre en participant à la communauté qui épaula ce projet open-source. Le Chapitre 17 conclura le récit de notre histoire et vous présentera les personnes qui nous ont inspiré les différents protagonistes.

Un dix-huitième chapitre vous propose un lexique qui éclaircit les mots quelques peu abscons utilisés dans cet ouvrage.