

Préface

Dans les grandes lignes, il y a une différence très nette entre les versions de J2EE qui sont apparues jusqu'à J2EE 1.4 et celles qui ont suivi cette version : à partir de Java EE 5, le cœur de la plate-forme a essayé non seulement de s'enrichir de nouvelles fonctionnalités, mais aussi de proposer un modèle de programmation complet, intégré et facile à apprendre et à utiliser pour le développement d'applications d'entreprise. En réalité, l'introduction des annotations et l'intégration de plus en plus étroite entre les différentes APIs ont engendré des changements irréversibles dans le travail quotidien des développeurs Java EE.

Alors que les nombreux changements proposés par Java EE 5 étaient toujours frais dans nos esprits, Java EE 6 est sorti et nous pouvons nous demander à quel titre Java EE 6 pourrait être plus qu'une version intermédiaire. Après tout, elle ne propose pas de nouvelle fonctionnalité tirant parti de nouveautés du langage Java autour desquelles nous pourrions concevoir nos développements et elle ne provoque pas non plus de révolution dans notre façon de bâtir nos applications. À ce sujet, je voudrais affirmer que les nouvelles fonctionnalités de Java EE 6 sont bien aussi importantes et d'une portée aussi grande que n'importe quelle nouveauté de Java EE 5. Simplement, de manière plus subtile.

Il est vrai qu'une grande partie des nouvelles fonctionnalités de Java EE 6 suit un modèle qui nous est désormais familier : des annotations pour les Servlets et des filtres ; un descripteur de déploiement facultatif `web.xml` ; une API basée sur les annotations pour les services Web RESTful (JAX-RS) ; une facilité d'injection de dépendances dépourvue de XML (ou presque). Et pourtant, en regardant de plus près, nous nous apercevons que Java EE 6 a ramené le niveau d'intégration entre APIs à un nouveau sommet ; nous remarquons que la validation de *beans* (Bean Validation) fonctionne sans procédure de mise en œuvre avec JSF et JPA, CDI (Contexts and Dependency Injection) offre un couplage JSF et EJB, et parvient à supprimer les différences entre les *managed beans* légers et les composants EJB lourds. Ce niveau d'intégration demande une coordination considérable qui aura nécessité que les différents groupes d'experts travaillent de concert plutôt que de travailler sur une fonctionnalité utilisateur supplémentaire livrée de façon isolée. De toute évidence, ce mode de développement facilité et efficace s'est généralisé et concerne désormais chaque élément de la plate-forme.

L'autre aspect révolutionnaire de Java EE 6 concerne l'introduction de plusieurs points d'extensibilité, notamment dans le conteneur de Servlet et dans CDI. Ce n'est pas moins qu'une reconnaissance explicite de l'importance d'un écosystème autour de Java EE avec des frameworks, des bibliothèques et des extensions qui participent au succès de la plate-forme à long terme. En mettant à disposition un mécanisme d'extension puissant, qui sera très probablement étendu dans des versions futures, les concepteurs de la plate-forme ont approuvé un modèle dans lequel des bibliothèques non standards peuvent participer au modèle de programmation, pour le bienfait de tous les développeurs.

Malgré tous les efforts pour réduire l'impact de Java EE 6 en passant par des actions d'élagage et la définition des profils, Java EE 6 reste tout de même une plate-forme conséquente. Vous ne trouverez pas un meilleur guide à ses secrets que celui de Jérôme Lafosse. Dans ce livre, il mène le lecteur d'un domaine à un autre de façon experte, avec adresse et sans jamais perdre de vue les besoins réels des développeurs à toutes les étapes. Loin d'être un traité académique ou une liste stérile d'APIs, ce livre utilise comme guide une application du monde réel afin de proposer une vue vivante et bien présentée des composants principaux de la plate-forme Java EE 6.

L'aspect le plus remarquable de ce livre est la quantité stupéfiante d'informations pratiques qu'il propose, ce qui vous fera gagner un temps colossal à toutes les étapes du développement de logiciels, à partir de la mise en œuvre de votre environnement et du serveur d'applications GlassFish, jusqu'au choix des outils, en passant par les paramètres de configuration les plus étonnantes. Vous trouverez la même attention aux détails et la même clarté d'organisation de l'architecture concernant la description de chaque tiers et de chaque API de la plate-forme.

Je suis reconnaissant à Jérôme d'avoir écrit ce livre et d'avoir partagé avec la communauté des développeurs ses vastes connaissances de la plate-forme Java EE. Je considère ce livre comme un outil d'apprentissage de valeur inestimable et j'espère que vous, lecteur, serez du même avis.

Roberto Chinnici
Java EE Platform Lead
Oracle Corporation
San Francisco, CA