

Java 9 modulo les modules

Speaker : Jean-Michel Doudoux

Format : Conférence

Date : 6 avril 2017

Slides : <https://fr.slideshare.net/jmdoudoux/java-9-modulo-les-modules-devoxx-fr-2017>

La grosse fonctionnalité de Java 9 est bien entendu Jigsaw, l'introduction d'un système de modules dans le JDK. Jigsaw va avoir beaucoup d'impact et va induire des ruptures. Ce n'est pas le sujet de ce talk qui est l'objet des autres fonctionnalités parmi 89 JEPs.

Changements mineurs dans le langage

JEP 213 reprenant 5 évolutions :

- Etendre l'utilisation de **@SafeVarargs** sur les méthodes privées
- Amélioration du **try with resources** introduit dans Java 7
- L'**opérateur <>** peut désormais être utilisé dans les classes anonymes si le compilateur est capable d'inférer le type
- Ajout de **méthodes privées** dans les **interfaces**. Permet de mutualiser du code de 2 default methods
- L'**identifiant _** est désormais interdit (en vue de Java 10). `Private int _ = 0 ;` génèrera une erreur de compilation

Nouvelles API

- **StackWalking API** : permet de parcourir, filtrer et récupérer des infos sur les stacktraces : nom du fichier, numéro de la ligne ... Nouvelle classe StackWalker thread-safe manipulant un Steam.
- Enrichissement des interactions avec le système hôte. Nouvelles méthodes dans java.awt.desktop (JEP 272)
- **Reactive Stream** : ensemble minimal de 4 interfaces (Flow, Flow.Subscriber, Flow.Subscription, Flow.Processor) qui pourront être utilisés par certains framework pour mettre en œuvre les [Reactive Stream](#) (flow API), mécanisme de type publication/souscription asynchrone. Intérêt : mécanisme de type back pressure (le consommateur pilote)

Evolutions des API

- **API Process** (JEP 102)
 - L'interface ProcessHandle permet d'obtenir le processus courant (méthode current()), terminer un processus, lister tous les processus d'un système ...
 - Permet de déclencher des traitements à la fin du processus
- **Fabrique pour les collections** (JEP 269). Avant Java 9, nous n'avons pas de solution simple pour créer des instances **immtables** (ex : Collections.unmodifiableSet(set)). Dans les interfaces List, Set et Map, Java 9 ajoute des fabriques statiques. 12 surcharges de la méthode of(). Limitations : pas le choix de l'instance de la collection,

on ne peut passer null, pas d'interface dédiée pour savoir qu'une collection immuable.

- La classe **Optional** est enrichie de 3 méthodes : stream, or et ifPresentOrElse
- 4 nouvelles méthodes dans l'API Stream : takeWhile, dropWhile (attention à leur utilisation sur des streams non ordonnés), ofNullable, surcharge de méthode iterate
- **Spin-wait hints**. C'est une première, la JEP 285 n'a pas été proposée par Oracle. Ajout de la méthode onSpinWait() dans la classe Thread. Méthode vide annotée avec @HotSpotIntrinsicCandidate. Permet de réduire la consommation CPU lors de l'attente d'un thread.

Mise à jour des outils du JDK

- **Réorganisation du JDK**. Avant Java 9, on avait un JRE inclus dans le JDK. En Java 9, la structure des répertoires est similaire pour JRE et JDK
- Des mécanismes ont été retirés : mécanisme endorsed, mécanisme d'extension. Les fichiers rt.jar et tools.jar sont retirés
- Les options de la ligne de commande Java 8 et inférieur ne sont pas homogènes : un ou deux -. La JEP 293 permet d'homogénéiser les options en s'inspirant la **syntaxe GNU**. Exemple : --module-path path. Les anciennes options restent pour des raisons de compatibilité.
- Des options GC sont supprimées
- Nouvelle politique pour déterminer quelle version de Java va être supportée par un compilateur. A partir de Java 9, le **compilateur ne supportera que les 3 versions précédentes de Java** : 8, 7 et 6.
- **Multi-release jar files** (JEP 238) : permet d'avoir un seul JAR pour plusieurs versions de Java. Le jar peut contenir plusieurs .class pour chaque version de Java. Version courante à la racine. Autres versions dans META-INF/versions/N.
- **JShell** : interpréteur interactif en ligne de commandes. Nouvel outil issu du projet Kulla.
- Outils retirés de Java 9 : jhat, Java DB,

Améliorations de la JVM

- **Nouvelle numérotation des versions de Java**. Le format de version introduit par Oracle est abandonné. Le nouveau format est inspiré de [SemVer](#) décomposé en 4 parties : \$MAJOR, \$MINEUR, \$SECURITY et \$PATCH. Une API est fournie pour récupérer ces informations depuis la JVM
- Le ramasse miette **G1 devient le GC par défaut**. Ce GC permet de réduire le temps de pause, particulièrement sur les heaps de grande taille.
- **Compact String** : réduction de l'empreinte mémoire. 1^{ière} tentative en Java 6 retirée à cause de bugs. En Java 9, une String est un tableau de byte et un octet supplémentaire indique le format.
- **Indify String concatenation** : réécriture de la concaténation de chaîne : InvokeDynamic à la place de StringBuilder

Support de standards

- Nouveaux charsets
- **JavaDoc : HTML 5**, recherche (implémentée en JavaScript)
- PKCS12 par défaut dans le Keystore
- Support de **SHA-3** dans JCE : SHA3-224 à SHA3-512
- Support du **UTF-8 dans le fichier properties**. Avant Java 9, un fichier properties devait être encodé en ISO-8859-1

Fonctionnalités divers

- Ajout de 2 attributs sur l'annotation `@Deprecated` : `forRemoval()`, `since()`
- Plusieurs `@Deprecated` ont été supprimés dans `java.awt.Component`
- Nouvel outil `jdeprscan` pour rechercher les API dépréciées mais utilisées
- Incubator modules : permet de développer et livrer des modules non standard (ex : module HTTP2)

Fonctionnalités deprecated

- Les applets Java. A remplacer par Java Web Start.
- Démon et exemples

Fonctionnalités reportées

- API JSON
- Microbenchmark basé sur JMH
- ...

Java 9 est prévu pour le 27 juillet 2017.

Ressources tierces :

- <http://www.loicmathieu.fr/wordpress/informatique/les-nouveautes-de-java-9-pour-les-developpeurs/>