

Partagez simplement vos Java CLI Apps

Pierre-Yves Fourmond

Oracle Dev Days

Journée Java - 22 Mai 2025



Qui suis-je ?



Pierre-Yves Fourmond

Développeur Back

Tech Lead / Consultant Senior

 Tribu « Java / Kotlin »



 Membre de Paris JUG

 pyfourmond.bsky.social

 [grumpyf0x48](https://github.com/grumpyf0x48)

#Java #Linux #Bash #Scripting #CLI

Historique du talk

-  Gagnant de la **Paris JUG** Academy 2024
-  Présenté à **Devoxx France** en 2024
- Présenté à **Human Talks** en 2025

Pourquoi ce talk ?

Une application de validation :

- Utilisée par les profils fonctionnels
- Qui traite de gros fichiers
- Déployée sur les environnements de test
- **Est-ce de cela qu'on a besoin ?**

Comment aurait-on pu faire ?

 Et si on avait utilisé du scripting ?

- Il n'y a pas de compilation
- Le code **source** est **interprété**


Il est donc :

- **Livré**
- **Modifiable**

On va vraiment faire du scripting en Java ?



Avec quel outil ?

-  JBang!
- Installer JBang sur la machine cible 🤖
- 🖱️ **Utilisons le JDK**

1. Java: Revenons aux bases

Utilisons Java 8 :

- Comment exécuter un programme Java ?

1. Java: Revenons aux bases

- Java est un langage **compilé**
- La compilation produit du **bytecode**
- Le bytecode est **interprété** par la **JVM**
- Il ne dépend pas de la plateforme d'exécution: « **Write once, run everywhere** »





2. Un nouveau mode de lancement

😏 Pour l'instant, ce qu'on a fait n'est **pas du scripting** !

Passons à Java 11 pour :

- Lancer un programme **plus facilement** avec la [JEP 330: Launch Single-File Source-Code Programs](#)
 - Enhance the java launcher to run a program supplied as a single file of Java source code,
 - including usage from within a script by means of "shebang" files and related techniques.





2. Un nouveau mode de lancement

-  On peut lancer un programme **sans le compiler** de deux façons :
 - i. En appelant la commande `java` sur un fichier `.java`
 - ii. En utilisant un fichier sans extension mais ayant un Shebang `#!`
-  Les deux façons de faire sont **exclusives**
-  Le Shebang ne compile pas en `java`
-  Tout le code source doit être dans le même fichier

3. Scripting: Revenons aux bases

😊 Cela commence à **ressembler à du scripting**, mais ...

3. Scripting: Revenons aux bases

-  On a maintenant un programme Java :
 -  Qui s'exécute dans le terminal
 -  Compris par les IDE
-  Il se comporte **comme un script**




4. Une première application

Construisons une application qui va :

- Extraire des informations depuis un export CSV provenant de Kibana

4. Une première application




On vient de construire une application Java en ligne de commandes :

-  Qui s'exécute comme un script
 - Le code source est modifiable
-  Le packaging est simplifié
-  On n'a besoin que du JDK

5. Une deuxième version de l'application

Que peut-on améliorer ?

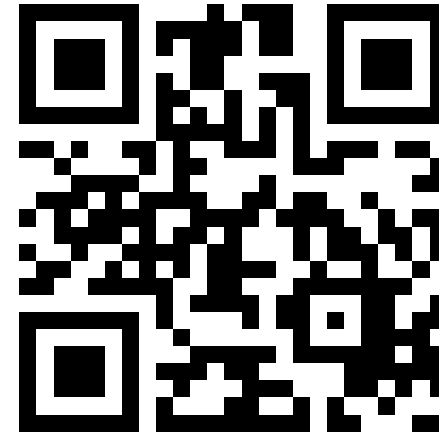
5. Une deuxième version de l'application

-  Les dépendances sont maintenant gérées par Gradle
 - Les dépendances transitives aussi
-  On a documenté l'application
-  Les évolutions du langage Java simplifient le code
 - JEP 477: Implicitly Declared Classes and Instance Main Methods (Third Preview) - Java 23
 - **JEP 512: Compact Source Files and Instance Main Methods - Java 25**
 - JEP 482: Flexible Constructor Bodies (Second Preview)
 - JEP 443: Unnamed Patterns and Variables (Preview)

Envie d'essayer ?

L'organisation  [java-cli-apps](#) contient :

- Des templates d'applications pour débiter
 - [basic-java-23-quickstart](#)
 - [java-23-quickstart](#)
- Les slides



Liens et ressources

JDK, Doc

[dev.java: The Destination for Java Developers](#)

[jdk.java.net: Production and Early-Access OpenJDK Builds, from Oracle](#)

[docs.oracle.com: The java Command](#)

[docs.oracle.com: The Java Virtual Machine Instruction Set](#)

Sur la JEP 330

[JEP 330: Launch Single-File Source-Code Programs](#)

[JEP 458: Launch Multi-File Source-Code Programs](#)

[Inside Java Newscast #63: Does Java 22 Kill Build Tools ?](#)

[JDK Source Code: Handling Shebang](#)

[The Go Cookbook: Running a file via a shebang line](#)

Liens et ressources

Simplification de Java

[JEP 495: Simple Source Files and Instance Main Methods \(Fourth Preview\)](#)

[JEP 494: Module Import Declarations \(Second Preview\)](#)

Divers

[RecordArgs: A simple command-line argument parser for Java applications](#)




[SDKMAN: The Software Development Kit Manager](#)

[JShell: an interactive tool for learning the Java programming language](#)

[direnv: unclutter your .profile](#)

[The Java Version Almanac](#)

Autres approches possibles


Utiliser ce qu'on connaît déjà    :

- Une **application compilée** avec Maven ou Gradle
- Une application et son JRE dédié avec `jlink`
- Un **binaire natif** avec GraalVM et `native-image`
- Un **paquet système** (rpm, deb, pkg, ...) avec `jpackage`

Avez-vous des questions ?

Ressources

 <https://github.com/java-cli-apps>

- Slides
- Vidéos des précédentes présentations
- Templates  pour démarrer

Contact

 [pyfourmond.bsky.social](https://bsky.social/pyfourmond)

 <https://linkedin.com/in/pyfourmond>

 pyfourmond@gmail.com