

Comment utiliser Spring Boot CLI

Spring Boot CLI (Interface en ligne de commande) est un outil puissant qui vous permet de développer, tester et prototyper rapidement des applications Spring Boot directement depuis la ligne de commande. Il est particulièrement utile pour exécuter des scripts Groovy sans configurer une structure de projet complète, générer de nouveaux projets ou expérimenter avec les fonctionnalités de Spring Boot. Voici un guide complet sur la façon d'installer et d'utiliser efficacement Spring Boot CLI.

Installation

Avant de pouvoir utiliser Spring Boot CLI, vous devez l'installer. Il existe deux méthodes principales en fonction de votre système d'exploitation :

1. Utilisation de SDKMAN! (Recommandé pour les systèmes Unix comme Linux ou macOS)

SDKMAN! est un outil pour gérer les kits de développement logiciel, ce qui en fait un moyen facile d'installer Spring Boot CLI.

- **Étape 1 : Installer SDKMAN!** Ouvrez votre terminal et exécutez :

```
curl -s "https://get.sdkman.io" | bash
```

Suivez les instructions pour initialiser SDKMAN! en sourceant le script :

```
source "$HOME/.sdkman/bin/sdkman-init.sh"
```

- **Étape 2 : Installer Spring Boot CLI** Exécutez la commande suivante :

```
 sdk install springboot
```

2. Installation manuelle (Pour Windows ou configuration manuelle)

Si vous êtes sous Windows ou préférez une installation manuelle : - Téléchargez le fichier ZIP de Spring Boot CLI depuis le site officiel de Spring. - Extrayez le fichier ZIP dans un répertoire de votre choix. - Ajoutez le répertoire `bin` du dossier extrait à la variable d'environnement PATH de votre système.

Vérification de l'installation

Pour confirmer que Spring Boot CLI est installé correctement, exécutez cette commande dans votre terminal :

```
spring --version
```

Vous devriez voir la version installée de Spring Boot CLI (par exemple, Spring CLI v3.3.0). Si cela fonctionne, vous êtes prêt à commencer à l'utiliser !

Principales façons d'utiliser Spring Boot CLI

Spring Boot CLI propose plusieurs fonctionnalités qui en font un outil idéal pour le développement rapide et le prototypage. Voici les principales façons de l'utiliser :

1. Exécution de scripts Groovy

L'une des fonctionnalités les plus remarquables de Spring Boot CLI est sa capacité à exécuter des scripts Groovy directement sans nécessiter une configuration de projet complète. Cela est parfait pour le prototypage rapide ou l'expérimentation avec Spring Boot.

- **Exemple : Création d'une application web simple** Créez un fichier nommé `hello.groovy` avec le contenu suivant :

```
@RestController
class HelloController {
    @RequestMapping("/")
    String home() {
        "Hello, World!"
    }
}
```

- **Exécuter le script** Dans votre terminal, naviguez jusqu'au répertoire contenant `hello.groovy` et exéutez :

```
spring run hello.groovy
```

Cela démarre un serveur web sur le port 8080. Ouvrez un navigateur et visitez `http://localhost:8080` pour voir “Hello, World!” affiché.

- **Ajout de dépendances** Vous pouvez inclure des dépendances directement dans le script en utilisant l'annotation `@Grab`. Par exemple :

```
@Grab('org.springframework.boot:spring-boot-starter-data-jpa')
@RestController
class HelloController {
```

```

@RequestMapping("/")
String home() {
    "Hello, World!"
}

```

Cela ajoute Spring Data JPA à votre script sans nécessiter de fichier de build.

- **Exécution de plusieurs scripts** Pour exécuter tous les scripts Groovy dans le répertoire actuel, utilisez :

```
spring run *.groovy
```

2. Création de nouveaux projets Spring Boot

Spring Boot CLI peut générer une nouvelle structure de projet avec les dépendances souhaitées, vous faisant gagner du temps lors du démarrage d'une application complète.

- **Exemple : Générer un projet** Exécutez cette commande pour créer un nouveau projet avec les dépendances web et data-jpa :

```
spring init --dependencies=web,data-jpa my-project
```

Cela crée un répertoire appelé `my-project` contenant une application Spring Boot configurée avec Spring Web et Spring Data JPA.

- **Options de personnalisation** Vous pouvez spécifier des options supplémentaires comme :

- Outil de build : `--build=maven` OU `--build=gradle`
- Langage : `--language=java`, `--language=groovy`, OU `--language=kotlin`
- Emballage : `--packaging=jar` OU `--packaging=war`

Par exemple :

```
spring init --dependencies=web --build=gradle --language=kotlin my-kotlin-project
```

3. Emballage des applications

Spring Boot CLI permet d'emballer vos scripts en fichiers JAR ou WAR exécutables pour le déploiement.

- **Créer un fichier JAR**

```
spring jar my-app.jar *.groovy
```

Cela emballé tous les scripts Groovy dans le répertoire actuel dans `my-app.jar`.

- **Créer un fichier WAR**

```
spring war my-app.war *.groovy
```

Cela génère un fichier `my-app.war` adapté au déploiement dans un conteneur de servlets.

4. Exécution des tests

Si vous avez des scripts de test Groovy, vous pouvez les exécuter avec :

```
spring test *.groovy
```

Cela exécute tous les scripts de test dans le répertoire actuel.

5. Utilisation de la coquille interactive

Pour une expérience interactive, lancez la coquille Spring Boot CLI :

```
spring shell
```

À l'intérieur de la coquille, vous pouvez exécuter des commandes comme `run`, `grab`, ou `jar` de manière interactive, ce qui est idéal pour l'expérimentation.

Résumé

Spring Boot CLI est un excellent outil pour les développeurs qui souhaitent travailler avec Spring Boot rapidement et efficacement depuis la ligne de commande. Voici comment l'utiliser en résumé :

1. **Installez-le** en utilisant SDKMAN! (`sdk install springboot`) ou manuellement en téléchargeant le ZIP et en mettant à jour votre PATH.
2. **Exédez des scripts Groovy** avec `spring run <script.groovy>` pour le prototypage rapide.
3. **Créez de nouveaux projets** avec `spring init --dependencies=<deps> <nom-du-projet>`.
4. **Emballez les applications** en fichiers JAR ou WAR en utilisant `spring jar` ou `spring war`.
5. **Expérez de manière interactive** avec `spring shell`.

Que vous soyez en train de prototyper une idée, d'apprendre Spring Boot ou d'automatiser des tâches, Spring Boot CLI offre une manière légère et flexible de commencer sans la surcharge d'un IDE complet ou d'une configuration de projet.