

Spring Boot CLI の使い方

Spring Boot CLI（コマンドラインインターフェース）は、コマンドラインから直接 Spring Boot アプリケーションを迅速に開発、テスト、プロトタイプ化するための強力なツールです。特に、完全なプロジェクト構造を設定せずに Groovy スクリプトを実行する、新しいプロジェクトを生成する、または Spring Boot の機能を試験するのに特に便利です。以下は、Spring Boot CLI を効果的にインストールして使用するための完全なガイドです。

インストール

Spring Boot CLI を使用する前に、インストールする必要があります。操作環境によって 2 つの主要な方法があります。

1. SDKMAN! を使用する (Unix 系 OS 向け、推奨)

SDKMAN! は、ソフトウェア開発キットを管理するツールで、Spring Boot CLI を簡単にインストールする方法です。

- **ステップ 1: SDKMAN! をインストールする** ターミナルを開き、以下のコマンドを実行します：

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

プロンプトに従って SDKMAN! を初期化します：

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

- **ステップ 2: Spring Boot CLI をインストールする** 以下のコマンドを実行します：

```
sdk install springboot
```

2. 手動インストール (Windows または手動設定向け)

Windows を使用している場合や手動インストールを希望する場合：- 公式 Spring ウェブサイトから Spring Boot CLI の ZIP ファイルをダウンロードします。- ZIP ファイルを任意のディレクトリに展開します。- 展開したフォルダから bin ディレクトリをシステムの PATH 環境変数に追加します。

インストールの確認

Spring Boot CLI が正しくインストールされているか確認するには、ターミナルで以下のコマンドを実行します：

```
spring --version
```

インストールされた Spring Boot CLI のバージョン（例：Spring CLI v3.3.0）が表示されます。これで使用を開始できます！

Spring Boot CLI の主要な使用方法

Spring Boot CLI は、迅速な開発とプロトタイピングに適したいくつかの機能を提供します。以下に主要な使用方法を示します。

1. Groovy スクリプトの実行

Spring Boot CLI の特徴の一つは、完全なプロジェクト設定なしで Groovy スクリプトを直接実行できる点です。これは、迅速なプロトタイピングや Spring Boot の試験に最適です。

- **例：シンプルな Web アプリケーションの作成** `hello.groovy` という名前のファイルを作成し、以下の内容を追加します：

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

- **スクリプトの実行** ターミナルで `hello.groovy` が含まれるディレクトリに移動し、以下のコマンドを実行します：

```
spring run hello.groovy
```

これにより、ポート 8080 で Web サーバーが起動します。ブラウザを開き、`http://localhost:8080` にアクセスして「Hello, World!」が表示されることを確認します。

- **依存関係の追加** `@Grab` アノテーションを使用して、スクリプトに依存関係を直接含めることができます。例えば：

```
@Grab('org.springframework.boot:spring-boot-starter-data-jpa')
@RestController
class HelloController {
    @RequestMapping("/")
    String home() {
        "Hello, World!"
    }
}
```

```
    }  
}
```

これにより、ビルドファイルなしで Spring Data JPA がスクリプトに追加されます。

- **複数のスクリプトの実行** 現在のディレクトリ内のすべての Groovy スクリプトを実行するには、以下のコマンドを使用します：

```
spring run *.groovy
```

2. 新しい Spring Boot プロジェクトの作成

Spring Boot CLI は、希望する依存関係を含む新しいプロジェクト構造を生成し、完全なアプリケーションを開始する際に時間を節約できます。

- **例：プロジェクトの生成** Web と data-jpa の依存関係を含む新しいプロジェクトを作成するには、以下のコマンドを実行します：

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

これにより、my-project という名前のディレクトリが作成され、Spring Web と Spring Data JPA で構成された Spring Boot アプリケーションが含まれます。

- **カスタマイズオプション** 追加のオプションを指定することもできます：

- ビルドツール：--build=maven または--build=gradle
- 言語：--language=java、--language=groovy または--language=kotlin
- パッケージング：--packaging=jar または--packaging=war

例えば：

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

3. アプリケーションのパッケージ化

Spring Boot CLI を使用して、スクリプトを実行可能な JAR または WAR ファイルにパッケージ化し、デプロイすることができます。

- **JAR ファイルの作成**

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

これにより、現在のディレクトリ内のすべての Groovy スクリプトが my-app.jar にパッケージ化されます。

- **WAR ファイルの作成**

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

これにより、`my-app.war` という名前のファイルが生成され、サーブレットコンテナにデプロイするのに適しています。

4. テストの実行

Groovy のテストスクリプトがある場合、以下のコマンドで実行できます：

```
spring test *.groovy
```

これにより、現在のディレクトリ内のすべてのテストスクリプトが実行されます。

5. インタラクティブシェルの使用

インタラクティブな体験を得るには、Spring Boot CLI シェルを起動します：

```
spring shell
```

シェル内で `run`、`grab`、`jar` などのコマンドをインタラクティブに実行でき、これは実験に最適です。

まとめ

Spring Boot CLI は、コマンドラインから迅速かつ効率的に Spring Boot を使用したい開発者にとって優れたツールです。以下のように使用します：

1. **インストール**：SDKMAN! を使用してインストールします (`sdk install springboot`) または、ZIP をダウンロードして PATH を更新します。
2. **Groovy スクリプトの実行**：`spring run <script.groovy>` を使用して迅速なプロトタイピングを行います。
3. **新しいプロジェクトの作成**：`spring init --dependencies=<deps> <project-name>` を使用して新しいプロジェクトを作成します。
4. **アプリケーションのパッケージ化**：`spring jar` または `spring war` を使用して JAR または WAR ファイルにパッケージ化します。
5. **インタラクティブな実験**：`spring shell` を使用してインタラクティブに実験します。

アイデアのプロトタイピング、Spring Boot の学習、またはタスクの自動化を行う場合、Spring Boot CLI は完全な IDE やプロジェクト設定なしで軽量で柔軟な方法を提供します。