

如何使用 Spring Boot CLI

Spring Boot CLI（命令行界面）是一個強大的工具，允許你直接從命令行快速開發、測試和原型設計 Spring Boot 應用程序。它特別適合用於運行 Groovy 腳本而不需要設置完整的項目結構、生成新項目或實驗 Spring Boot 功能。以下是如何有效安裝和使用 Spring Boot CLI 的完整指南。

安裝

在使用 Spring Boot CLI 之前，你需要安裝它。根據你的操作系統，有兩種主要方法：

1. 使用 SDKMAN! (推薦給基於 Unix 的系統如 Linux 或 macOS)

SDKMAN! 是一個用於管理軟件開發工具包的工具，使其成為安裝 Spring Boot CLI 的簡單方法。

- **步驟 1：安裝 SDKMAN!** 打開你的終端並運行：

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

按照提示初始化 SDKMAN!，通過源文件：

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

- **步驟 2：安裝 Spring Boot CLI** 運行以下命令：

```
sdk install springboot
```

2. 手動安裝 (適用於 Windows 或手動設置)

如果你在 Windows 上或偏好手動安裝：
- 從官方 Spring 網站下載 Spring Boot CLI ZIP 文件。
- 將 ZIP 文件解壓縮到你選擇的目錄。
- 將解壓縮文件夾中的 bin 目錄添加到系統的 PATH 環境變量。

驗證安裝

要確認 Spring Boot CLI 已正確安裝，在終端中運行以下命令：

```
spring --version
```

你應該會看到已安裝的 Spring Boot CLI 版本（例如 Spring CLI v3.3.0）。如果這樣做，你就可以開始使用它了！

使用 Spring Boot CLI 的主要方法

Spring Boot CLI 提供了幾個功能，使其非常適合快速開發和原型設計。以下是主要的使用方法：

1. 運行 Groovy 腳本

Spring Boot CLI 的突出特點之一是能夠直接運行 Groovy 腳本而不需要完整的項目設置。這對於快速原型設計或實驗 Spring Boot 非常理想。

- **示例：創建簡單的 Web 應用程序** 創建一個名為 `hello.groovy` 的文件，內容如下：

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

- **運行腳本** 在終端中，導航到包含 `hello.groovy` 的目錄並運行：

```
spring run hello.groovy
```

這將在 8080 端口啟動一個 Web 伺服器。打開瀏覽器並訪問 `http://localhost:8080`，你應該會看到顯示“Hello, World!”。

- **添加依賴** 你可以使用 `@Grab` 註釋直接在腳本中包含依賴。例如：

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

這將在不需要構建文件的情況下將 Spring Data JPA 添加到你的腳本中。

- **運行多個腳本** 要運行當前目錄中的所有 Groovy 腳本，請使用：

```
spring run *.groovy
```

2. 創建新的 Spring Boot 項目

Spring Boot CLI 可以生成具有所需依賴的新項目結構，當開始完整應用程序時節省時間。

- **示例：生成項目** 運行以下命令以創建具有 Web 和 data-jpa 依賴的新項目：

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

這將創建一個名為 `my-project` 的目錄，其中包含配置了 Spring Web 和 Spring Data JPA 的 Spring Boot 應用程序。

- **自定義選項** 你可以指定其他選項，例如：

- 构建工具：`--build=maven` 或 `--build=gradle`
- 語言：`--language=java`、`--language=groovy` 或 `--language=kotlin`
- 打包：`--packaging=jar` 或 `--packaging=war`

例如：

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

3. 打包應用程序

Spring Boot CLI 允許你將腳本打包成可執行的 JAR 或 WAR 文件以進行部署。

- **創建 JAR 文件**

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

這將當前目錄中的所有 Groovy 腳本打包到 `my-app.jar` 中。

- **創建 WAR 文件**

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

這將生成一個適合部署到 Servlet 容器的 `my-app.war` 文件。

4. 運行測試

如果你有 Groovy 測試腳本，可以使用以下命令執行它們：

```
spring test *.groovy
```

這將運行當前目錄中的所有測試腳本。

5. 使用交互式 Shell

對於交互式體驗，啟動 Spring Boot CLI Shell：

```
spring shell
```

在 Shell 中，你可以交互式地運行命令，如 `run`、`grab` 或 `jar`，這對於實驗非常有用。

摘要

Spring Boot CLI 是一個適合從命令行快速高效地使用 Spring Boot 的優秀工具。以下是如何使用它的簡要說明：

1. **安裝它**使用 SDKMAN! (`sdk install springboot`) 或手動下載 ZIP 文件並更新你的 PATH。
2. **運行 Groovy 腳本**使用 `spring run <script.groovy>` 進行快速原型設計。
3. **創建新項目**使用 `spring init --dependencies=<deps> <project-name>`。
4. **打包應用程序到 JAR 或 WAR 文件**使用 `spring jar` 或 `spring war`。
5. **交互式實驗**使用 `spring shell`。

無論你是原型設計一個想法、學習 Spring Boot 還是自動化任務，Spring Boot CLI 都提供了一種輕量級和靈活的方式來開始，而不需要完整的 IDE 或項目設置。