

如何使用 Spring Boot CLI

Spring Boot CLI（命令行界面）是一种强大的工具，允许你直接从命令行快速开发、测试和原型设计 Spring Boot 应用程序。它特别适用于运行 Groovy 脚本而不需要设置完整的项目结构，生成新项目或实验 Spring Boot 功能。以下是如何有效安装和使用 Spring Boot CLI 的完整指南。

安装

在使用 Spring Boot CLI 之前，你需要安装它。根据你的操作系统，有两种主要的安装方法：

1. 使用 SDKMAN!（推荐用于基于 Unix 的系统如 Linux 或 macOS）

SDKMAN! 是一种用于管理软件开发工具包的工具，使其成为安装 Spring Boot CLI 的一种简单方法。

- **步骤 1：安装 SDKMAN!** 打开你的终端并运行：

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

按照提示初始化 SDKMAN!，通过源脚本：

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

- **步骤 2：安装 Spring Boot CLI** 运行以下命令：

```
sdk install springboot
```

2. 手动安装（适用于 Windows 或手动设置）

如果你在 Windows 上或更喜欢手动安装：
- 从官方 Spring 网站下载 Spring Boot CLI ZIP 文件。
- 将 ZIP 文件解压到你选择的目录。
- 将解压后文件夹中的 `bin` 目录添加到系统的 PATH 环境变量。

验证安装

要确认 Spring Boot CLI 已正确安装，在终端中运行以下命令：

```
spring --version
```

你应该会看到安装的 Spring Boot CLI 版本（例如 Spring CLI v3.3.0）。如果成功，你就可以开始使用它了！

使用 Spring Boot CLI 的关键方法

Spring Boot CLI 提供了多种功能，使其非常适合快速开发和原型设计。以下是主要的使用方法：

1. 运行 Groovy 脚本

Spring Boot CLI 的一个显著特点是可以直接运行 Groovy 脚本，而不需要完整的项目设置。这对于快速原型设计或实验 Spring Boot 非常理想。

- **示例：创建一个简单的 Web 应用程序** 创建一个名为 `hello.groovy` 的文件，内容如下：

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

- **运行脚本** 在终端中，导航到包含 `hello.groovy` 的目录并运行：

```
spring run hello.groovy
```

这将在端口 8080 启动一个 Web 服务器。打开浏览器并访问 `http://localhost:8080`，你会看到显示“Hello, World!”。

- **添加依赖** 你可以使用 `@Grab` 注解直接在脚本中包含依赖。例如：

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

这将在不需要构建文件的情况下将 Spring Data JPA 添加到你的脚本中。

- **运行多个脚本** 要运行当前目录中的所有 Groovy 脚本，使用：

```
spring run *.groovy
```

2. 创建新的 Spring Boot 项目

Spring Boot CLI 可以生成一个包含你所需依赖的新项目结构，节省了启动完整应用程序的时间。

- **示例：生成项目** 运行以下命令以创建一个带有 web 和 data-jpa 依赖的新项目：

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

这将创建一个名为 `my-project` 的目录，其中包含一个配置了 Spring Web 和 Spring Data JPA 的 Spring Boot 应用程序。

- **自定义选项** 你可以指定额外的选项，例如：

- 构建工具：`--build=maven` 或 `--build=gradle`
- 语言：`--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. 打包应用程序

Spring Boot CLI 允许你将脚本打包成可执行的 JAR 或 WAR 文件以进行部署。

- **创建 JAR 文件**

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

这将当前目录中的所有 Groovy 脚本打包到 `my-app.jar` 中。

- **创建 WAR 文件**

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

这将生成一个适合部署到 Servlet 容器的 `my-app.war` 文件。

4. 运行测试

如果你有 Groovy 测试脚本，可以使用以下命令执行它们：

```
spring test *.groovy
```

这将运行当前目录中的所有测试脚本。

5. 使用交互式 Shell

为了获得交互式体验，启动 Spring Boot CLI Shell：

```
spring shell
```

在 Shell 中，你可以运行命令如 `run`、`grab` 或 `jar`，这对于实验非常方便。

总结

Spring Boot CLI 是一个适合从命令行快速高效地使用 Spring Boot 的工具。以下是如何使用它的简要说明：

1. **安装它** 使用 SDKMAN! (`sdk install springboot`) 或手动下载 ZIP 并更新你的 PATH。
2. **运行 Groovy 脚本** 使用 `spring run <script.groovy>` 进行快速原型设计。
3. **创建新项目** 使用 `spring init --dependencies=<deps> <project-name>`。
4. **打包应用程序** 使用 `spring jar` 或 `spring war` 将应用程序打包成 JAR 或 WAR 文件。
5. **交互式实验** 使用 `spring shell`。

无论你是原型设计一个想法、学习 Spring Boot 还是自动化任务，Spring Boot CLI 都提供了一种轻量级和灵活的方式，无需完整的 IDE 或项目设置。