

# Wie man die Spring Boot CLI verwendet

Spring Boot CLI (Command Line Interface) ist ein leistungsfähiges Werkzeug, das es Ihnen ermöglicht, Spring Boot-Anwendungen direkt von der Kommandozeile schnell zu entwickeln, zu testen und zu prototypisieren. Es ist besonders nützlich zum Ausführen von Groovy-Skripten ohne Einrichtung einer vollständigen Projektstruktur, zum Generieren neuer Projekte oder zum Experimentieren mit Spring Boot-Funktionen. Hier ist eine vollständige Anleitung zur Installation und effektiven Nutzung der Spring Boot CLI.

---

## Installation

Bevor Sie die Spring Boot CLI verwenden können, müssen Sie sie installieren. Es gibt zwei Hauptmethoden, abhängig von Ihrem Betriebssystem:

### 1. Verwendung von SDKMAN! (Empfohlen für Unix-basierte Systeme wie Linux oder macOS)

SDKMAN! ist ein Tool zur Verwaltung von Software-Entwicklungskits und erleichtert die Installation der Spring Boot CLI.

- **Schritt 1: Installieren Sie SDKMAN!** Öffnen Sie Ihr Terminal und führen Sie aus:

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

Folgen Sie den Anweisungen zur Initialisierung von SDKMAN! durch Quellen des Skripts:

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

- **Schritt 2: Installieren Sie Spring Boot CLI** Führen Sie den folgenden Befehl aus:

```
sdk install springboot
```

### 2. Manuelle Installation (Für Windows oder manuelle Einrichtung)

Wenn Sie Windows verwenden oder eine manuelle Installation bevorzugen: - Laden Sie die Spring Boot CLI ZIP-Datei von der offiziellen Spring-Website herunter. - Entpacken Sie die ZIP-Datei in ein Verzeichnis Ihrer Wahl. - Fügen Sie das `bin`-Verzeichnis aus dem entpackten Ordner zur PATH-Umgebungsvariable Ihres Systems hinzu.

## Überprüfen der Installation

Um zu bestätigen, dass die Spring Boot CLI korrekt installiert ist, führen Sie diesen Befehl in Ihrem Terminal aus:

```
spring --version
```

Sie sollten die installierte Version der Spring Boot CLI sehen (z.B. Spring CLI v3.3.0). Wenn dies funktioniert, sind Sie bereit, sie zu verwenden!

---

## Wichtige Anwendungsmöglichkeiten der Spring Boot CLI

Spring Boot CLI bietet mehrere Funktionen, die sie ideal für die schnelle Entwicklung und das Prototypisieren machen. Hier sind die Hauptanwendungsmöglichkeiten:

### 1. Ausführen von Groovy-Skripten

Eine der herausragenden Funktionen der Spring Boot CLI ist die Möglichkeit, Groovy-Skripte direkt auszuführen, ohne eine vollständige Projektstruktur zu benötigen. Dies ist perfekt für schnelles Prototypisieren oder Experimentieren mit Spring Boot.

- **Beispiel: Erstellen einer einfachen Webanwendung** Erstellen Sie eine Datei mit dem Namen `hello.groovy` mit folgendem Inhalt:

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

- **Ausführen des Skripts** Navigieren Sie in Ihrem Terminal zum Verzeichnis, das `hello.groovy` enthält, und führen Sie aus:

```
spring run hello.groovy
```

Dies startet einen Webserver auf Port 8080. Öffnen Sie einen Browser und besuchen Sie `http://localhost:8080`, um "Hello, World!" angezeigt zu bekommen.

- **Hinzufügen von Abhängigkeiten** Sie können Abhängigkeiten direkt im Skript mit der `@Grab`-Annotation hinzufügen. Zum Beispiel:

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

Dies fügt Spring Data JPA zu Ihrem Skript hinzu, ohne dass eine Build-Datei benötigt wird.

- **Ausführen mehrerer Skripte** Um alle Groovy-Skripte im aktuellen Verzeichnis auszuführen, verwenden Sie:

```
spring run *.groovy
```

## 2. Erstellen neuer Spring Boot-Projekte

Spring Boot CLI kann eine neue Projektstruktur mit Ihren gewünschten Abhängigkeiten generieren und Ihnen Zeit sparen, wenn Sie eine vollständige Anwendung starten.

- **Beispiel: Generieren eines Projekts** Führen Sie diesen Befehl aus, um ein neues Projekt mit Web- und data-jpa-Abhängigkeiten zu erstellen:

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

Dies erstellt ein Verzeichnis mit dem Namen my-project, das eine Spring Boot-Anwendung enthält, die mit Spring Web und Spring Data JPA konfiguriert ist.

- **Anpassungsoptionen** Sie können zusätzliche Optionen wie Folgendes angeben:

- Build-Werkzeug: --build=maven oder --build=gradle
- Sprache: --language=java, --language=groovy oder --language=kotlin
- Verpackung: --packaging=jar oder --packaging=war

Zum Beispiel:

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

## 3. Verpacken von Anwendungen

Spring Boot CLI ermöglicht es Ihnen, Ihre Skripte in ausführbare JAR- oder WAR-Dateien zur Bereitstellung zu verpacken.

- **Erstellen einer JAR-Datei**

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

Dies verpackt alle Groovy-Skripte im aktuellen Verzeichnis in my-app.jar.

- **Erstellen einer WAR-Datei**

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

Dies erzeugt eine my-app.war-Datei, die für die Bereitstellung in einem Servlet-Container geeignet ist.

## 4. Ausführen von Tests

Wenn Sie Groovy-Testskripte haben, können Sie diese mit ausführen:

```
spring test *.groovy
```

Dies führt alle Testskripte im aktuellen Verzeichnis aus.

## 5. Verwenden der interaktiven Shell

Für eine interaktive Erfahrung starten Sie die Spring Boot CLI-Shell:

```
spring shell
```

Innerhalb der Shell können Sie Befehle wie run, grab oder jar interaktiv ausführen, was großartig für Experimente ist.

---

## Zusammenfassung

Spring Boot CLI ist ein hervorragendes Werkzeug für Entwickler, die schnell und effizient von der Kommandozeile aus mit Spring Boot arbeiten möchten. Hier ist, wie Sie es in Kürze verwenden:

1. **Installieren Sie es** mit SDKMAN! (`sdk install springboot`) oder manuell durch Herunterladen der ZIP-Datei und Aktualisieren Ihres PATH.
2. **Führen Sie Groovy-Skripte** mit `spring run <script.groovy>` für schnelles Prototypisieren aus.
3. **Erstellen Sie neue Projekte** mit `spring init --dependencies=<deps> <project-name>`.
4. **Verpacken Sie Anwendungen** in JAR- oder WAR-Dateien mit `spring jar` oder `spring war`.
5. **Experimentieren Sie interaktiv** mit `spring shell`.

Ob Sie eine Idee prototypisieren, Spring Boot lernen oder Aufgaben automatisieren, Spring Boot CLI bietet eine leichte und flexible Möglichkeit, ohne den Overhead einer vollständigen IDE oder Projektstruktur zu beginnen.