

Spring Boot Basics - Setup & Hello World

1. Was ist Spring Boot?

Spring Boot ist ein Framework zur Vereinfachung der Entwicklung von Java-basierten Anwendungen. Es bietet eine vorgefertigte Konfiguration und nutzt "Convention over Configuration".

Vorteile:

- **Schneller Start:** Vorkonfigurierte Einstellungen.
 - **Standalone:** Eingebettete Webserver wie Tomcat.
 - **Production-Ready:** Features wie Monitoring, Metriken, und Health Checks.
-

2. Setup und Installation

Voraussetzungen:

- **Java:** Version 8 oder höher.
- **Build-Tool:** Maven oder Gradle.

Projekt erstellen:

- Besuche start.spring.io und wähle:
 - **Project:** Maven.
 - **Dependencies:** Spring Web, Spring Data JPA, H2 Database
 - **Packaging:** Jar.
 - **Java Version:** 17 (empfohlen).
 - Alternativ: Nutze die Integration von IntelliJ
-

3. Projektstruktur

```
my-spring-boot-app/  
|-- src/main/java  
|   |-- com.example.demo  
|       |-- DemoApplication.java  
|       |-- controller/
```

```
| -- src/main/resources
|   |-- application.properties
```

- `DemoApplication.java` : Hauptklasse mit `@SpringBootApplication` .
- `controller/` : Enthält REST-Controller.
- `application.properties` : Konfigurationsdateien.

4. Wichtige Annotationen

Annotation	Beschreibung
<code>@SpringBootApplication</code>	Markiert die Hauptklasse und aktiviert Auto-Configuration.
<code>@RestController</code>	Kombiniert <code>@Controller</code> und <code>@ResponseBody</code> für REST-APIs.
<code>@GetMapping</code>	Verarbeitet HTTP GET-Anfragen.

5. Code-Beispiel: Hello World Anwendung

Hauptklasse:

```
package com.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class DemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }
}
```

REST-Controller:

```
package com.example.demo.controller;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;
```

```
@RestController
public class HelloWorldController {

    @GetMapping("/")
    public String helloWorld() {
        return "Hello, World!";
    }
}
```

6. Konfiguration: `application.properties`

Property	Beschreibung
<code>server.port=8080</code>	Definiert den Serverport.

In diesem Beispiel bleibt der Standardport 8080 unverändert.

7. Zusätzliche Ressourcen:

- [Spring Boot Documentation](#)
- [Baeldung Tutorials](#)