

第一章：SpringBoot 入门

1.1 本章大纲

- SpringBoot 介绍
- 构建 SpringBoot 项目及启动器
- SpringBoot 入门案例

1.2 SpringBoot 介绍

Spring 诞生时是 Java 企业版（Java Enterprise Edition，JEE，也称 J2EE）的轻量级框架。

Spring 为企业级 Java 开发提供了一种相对简单的方法，通过依赖注入和面向切面编程，用简单的 Java 对象（Plain Old Java Object，POJO）实现了 EJB 的功能。虽然 Spring 的组件代码是轻量级的，但它的配置却是重量级的。

第一阶段：xml 配置

在 Spring 1.x 时代，使用 Spring 开发满眼都是 xml 配置的 Bean，随着项目的扩大，我们需要把 xml 配置文件放到不同的配置文件里，那时需要频繁的在开发的类和配置文件之间进行切换

第二阶段：注解配置

在 Spring 2.x 时代，随着 JDK1.5 带来的注解支持，Spring 提供了声明 Bean 的注解（例如@Component、@Service），大大减少了配置量。主要使用的方式是应用的基本配置（如数据库配置）用 xml，业务配置用注解

第三阶段：java 配置

Spring 3.0 引入了基于 Java 的配置能力，这是一种类型安全的可重构配置方式，可以代替 XML。我们目前刚好处于这个时代，Spring4.x 和 Spring Boot 都推荐使用 Java 配置。

Spring Boot 简化了基于 Spring 的应用开发，只需要“run”就能创建一个独立的、生产级别的 Spring 应用。Spring Boot 为 Spring 平台及第三方库提供开箱即用的设置（提供默认设置），这样我们就可以简单的开始。多数 Spring Boot 应用只需要很少的 Spring 配置。

Spring Boot 主要目标是：

- 为所有 Spring 的开发提供一个从根本上更快的入门体验
- 开箱即用，但通过自己设置参数，即可快速摆脱这种方式。
- 提供了一些大型项目中常见的非功能性特性，如内嵌服务器、安全、指标，健康检测、外部化配置等
- 绝对没有代码生成，也无需 XML 配置。

1.3 构建 SpringBoot 项目及启动器

1.3.1 环境准备

- Spring Tool Suite4
- Maven 3.5+
- Jdk 8+

1.3.2 使用 STS 构建 SpringBoot 项目

The image shows two screenshots from the Spring Tool Suite (STS) IDE. The left screenshot is the 'New Spring Starter Project' dialog. It contains the following fields and values: Service URL: https://start.spring.io; Name: springboot02; Location: F:\code\sts\springboot02; Type: Maven; Packaging: War; Java Version: 8; Language: Java; Group: com.kazu; Artifact: springboot02; Version: 0.0.1-SNAPSHOT; Description: Demo project for Spring Boot; Package: com.kazu. The right screenshot is the 'New Spring Starter Project Dependencies' dialog. It shows 'Spring Boot Version' as 2.1.3. Under the 'Available' section, 'web' is listed. Under the 'Selected' section, 'Web' is checked. The 'Web' category is expanded, showing options like Validation, Reactive Web, Web Services, Jersey, Websocket, Vaadin, Apache CXF, and Mobile. The 'Web' option is selected.

1.3.3 修改 pom 文件，更新 JDK 版本

```
<!-- JDK版本号 -->
<properties>
    <java.version>1.8</java.version>
</properties>
```

1.3.4 注入 SpringBoot 启动坐标

```
<!-- 注入启动器 -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

1.3.5 Spring 启动器

所谓的 springBoot 启动器其实就是一些 jar 包的集合。SpringBoot 一共提供 44 启动器。

➤ spring-boot-starter-web

支持全栈式的 web 开发，包括了 tomcat 和 springMVC 等 jar

➤ spring-boot-starter-jdbc

支持 spring 以 jdbc 方式操作数据库的 jar 包的集合

➤ spring-boot-starter-redis

支持 redis 键值存储的数据库操作

1.4 SpringBoot 入门案例

1.4.1 编写 HelloWorld 的 Controller

```
@Controller
@RequestMapping("/home")
public class HelloController {

    @RequestMapping("/hello")
    @ResponseBody
    public String index() {
        return "hello,springboot";
    }
}
```

1.4.2 启动 SpringBoot

```
@SpringBootApplication
public class Springboot01Application {

    public static void main(String[] args) {
        SpringApplication.run(Springboot01Application.class, args);
    }

}
```

1.4.3 启动器需要注意的问题

启动器存放的位置。启动器可以和 controller 位于同一个包下，或者位于 controller 的上一级包中，但是不能放到 controller 的平级以及子包下。