

Spring Boot快速上手

教学内容

- 第一节 SpringBoot介绍
- 第二节 快速创建SpringBoot应用
- 第三节 开发环境热部署
- 第四节 系统配置



SpringBoot介绍

- Spring Boot是由Pivotal团队提供的基于Spring的全新框架，旨在简化Spring应用的初始搭建和开发过程。
- Spring Boot是所有基于Spring开发项目的起点。
- Spring Boot就是尽可能地简化应用开发的门槛，让应用开发、测试、部署变得更加简单。



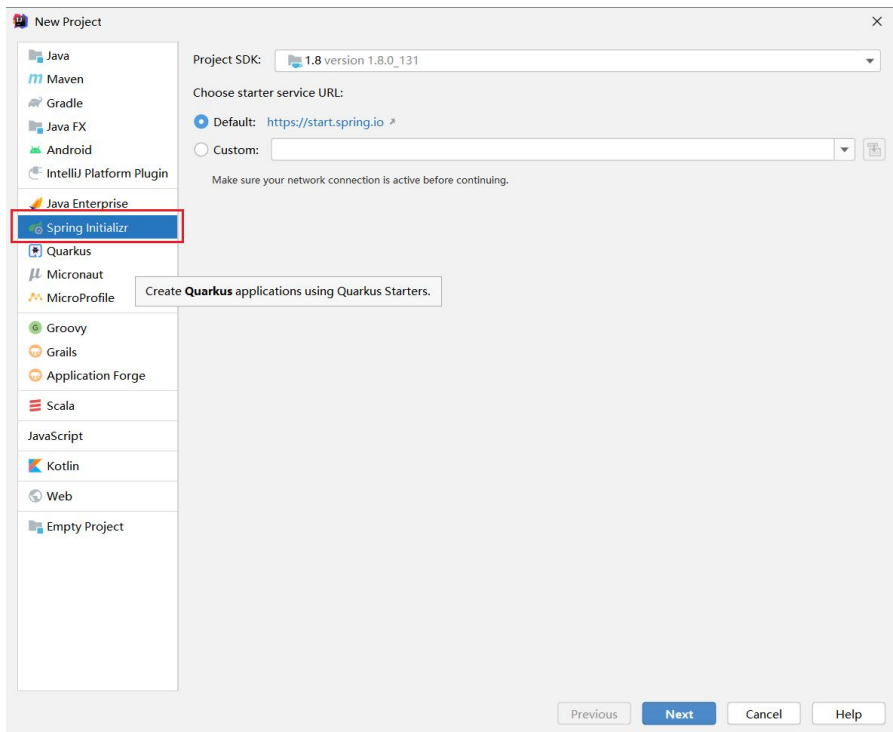
SpringBoot特点

- 遵循“约定优于配置”的原则，只需要很少的配置或使用默认的配置。
- 能够使用内嵌的Tomcat、Jetty服务器，不需要部署war文件。
- 提供定制化的启动器Starters，简化Maven配置，开箱即用。
- 纯Java配置，没有代码生成，也不需要XML配置。
- 提供了生产级的服务监控方案，如安全监控、应用监控、健康检测等。



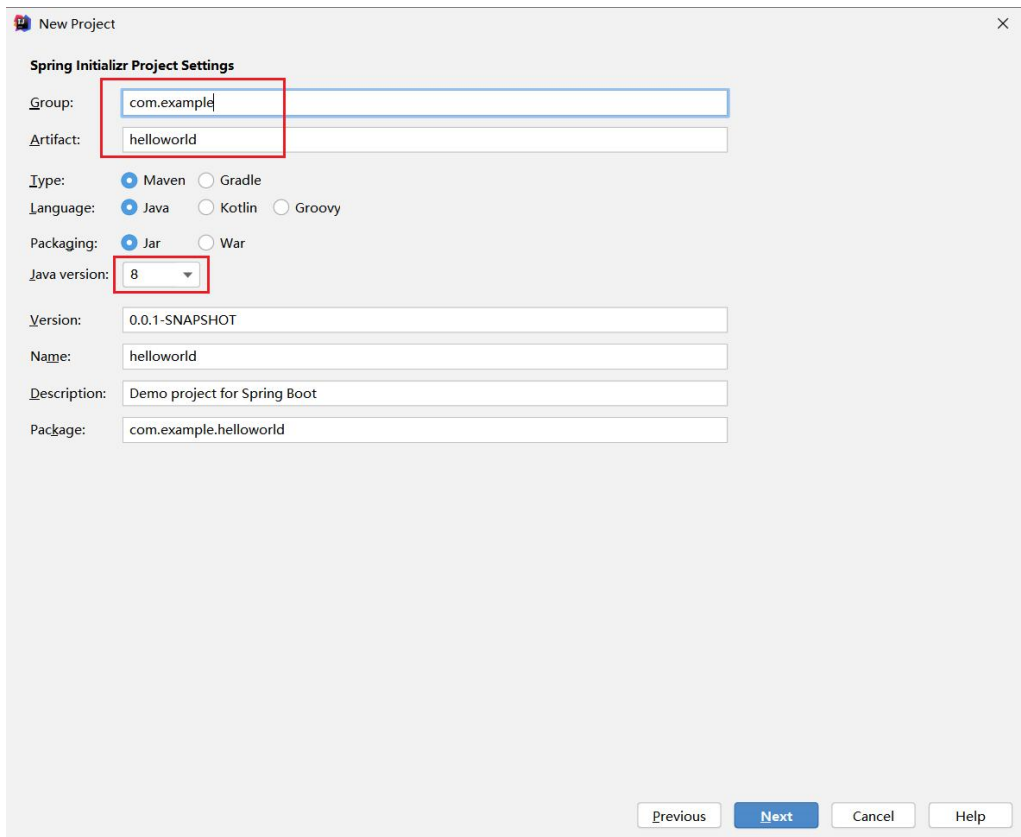
快速创建SpringBoot应用

利用IDEA提供的Spring Initializr创建SpringBoot应用



填写项目信息

- Group: 一般输入公司域名。
- Artifact: 项目名称。



The image shows the 'New Project' dialog box in an IDE, specifically the 'Spring Initializr Project Settings' tab. The dialog is titled 'New Project' and has a close button (X) in the top right corner. It contains several fields and options for configuring a new project. The 'Group' field is set to 'com.example' and the 'Artifact' field is set to 'helloworld'. The 'Type' section has 'Maven' selected. The 'Language' section has 'Java' selected. The 'Packaging' section has 'Jar' selected. The 'Java version' dropdown is set to '8'. The 'Version' field is set to '0.0.1-SNAPSHOT'. The 'Name' field is set to 'helloworld'. The 'Description' field is set to 'Demo project for Spring Boot'. The 'Package' field is set to 'com.example.helloworld'. At the bottom, there are four buttons: 'Previous', 'Next', 'Cancel', and 'Help'.

Spring Initializr Project Settings

Group: com.example

Artifact: helloworld

Type: ☒ Maven ☐ Gradle

Language: ☒ Java ☐ Kotlin ☐ Groovy

Packaging: ☒ Jar ☐ War

Java version: 8

Version: 0.0.1-SNAPSHOT

Name: helloworld

Description: Demo project for Spring Boot

Package: com.example.helloworld

Previous Next Cancel Help

第一个helloworld程序

- 创建子目录controller
- 在目录controller中, 创建HelloController.java文件
- 启动项目, 在浏览器窗口中输入 "http://localhost:8080/hello"

```
@RestController
public class HelloController {

    @RequestMapping(value="/hello")
    public String index(){
        return "Hello world!";
    }
}
```



开发环境热部署

- 在实际的项目开发调试过程中会频繁地修改后台类文件，导致需要重新编译、重新启动，整个过程非常麻烦，影响开发效率。
- Spring Boot提供了spring-boot-devtools组件，使得无须手动重启Spring Boot应用即可重新编译、启动项目，大大缩短编译启动的时间。
- devtools会监听classpath下的文件变动，触发Restart类加载器重新加载该类，从而实现类文件和属性文件的热部署。
- 并不是所有的更改都需要重启应用（如静态资源、视图模板），可以通过设置spring.devtools.restart.exclude属性来指定一些文件或目录的修改不用重启应用



开发环境热部署

- 在pom.xml配置文件中添加dev-tools依赖。
- 使用optional=true表示依赖不会传递，即该项目依赖devtools；其他项目如果引入此项目生成的JAR包，则不会包含devtools

```
<dependency>  
    <groupId>org.springframework.boot</groupId>  
    <artifactId>spring-boot-devtools</artifactId>  
    <optional>true</optional>  
</dependency>
```



开发环境热部署

- 在application.properties中配置devtools。

热部署生效

`spring.devtools.restart.enabled=true`

设置重启目录

`spring.devtools.restart.additional-paths=src/main/java`

设置classpath目录下的WEB-INF文件夹内容修改不重启

`spring.devtools.restart.exclude=static/**`



开发环境热部署

- 如果使用了Eclipse，那么在修改完代码并保存之后，项目将自动编译并触发重启，而如果使用了IntelliJ IDEA，还需要配置项目自动编译。
- 打开Settings页面，在左边的菜单栏依次找到Build,Execution,Deployment→Compile，勾选Build project automatically
- 按Ctrl+Shift+Alt+/快捷键调出Maintenance页面，单击Registry，勾选compiler.automake.allow.when.app.running复选框。
- 做完这两步配置之后，若开发者再次在IntelliJ IDEA中修改代码，则项目会自动重启。



系统配置

- 项目创建成功后会默认在resources目录下生成application.properties文件。该文件包含Spring Boot项目的全局配置。
- 配置格式如下：

```
# 服务器端口配置  
server.port=8081
```

