

Springboot热部署实现原理及实例详解

主要介绍了Springboot热部署实现原理及实例详解,文中通过示例代码介绍的非常详细，对大家的学习或者工作具有一定的参考学习价值,需要的朋友可以参考下

1.前言

在实际开发过程中，每次修改代码就得将项目重启，重新部署，对于一些大型应用来说，重启时间需要花费大量的时间成本。对于一个后端开发者来说，重启过程确实很难受啊。在 Java 开发领域，热部署一直是一个难以解决的问题，目前的 Java 虚拟机只能实现方法体的修改热部署，对于整个类的结构修改，仍然需要重启虚拟机，对类重新加载才能完成更新操作。下面我~我们就看看对于简单的类修改的热部署怎么实现。

2. 原理

深层原理是使用了两个ClassLoader，一个ClassLoader加载那些不会改变的类（第三方Jar包），另一个ClassLoader加载会更改的类，称为restart ClassLoader,这样在有代码更改的时候，原来的restart ClassLoader 被丢弃，重新创建一个restart ClassLoader，由于需要加载的类相比较少，所以实现了较快的重启时间。

3. 方式

springboot有3中热部署方式：

- 1.使用springloaded配置pom.xml文件，使用mvn spring-boot:run启动
- 2.使用springloaded本地加载启动，配置jvm参数

-javaagent:<jar包地址> -noverify

- 3.使用devtools工具包，操作简单，但是每次需要重新部署

这里主要讲解一下第三种热部署方式的使用，因为在网上查找资源时，总会发现很多人就是springloaded和devtools都使用了，其实是多余的操作，而且第三种操作简单快捷。

4. 开始配置

4.1 在pom.xml中添加依赖

```
<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
<!--devtools热部署-->
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-devtools</artifactId>
<optional>true</optional>
<scope>true</scope>
</dependency>
</dependencies>

<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
<configuration>
<fork>true</fork>
</configuration>
</plugin>
</plugins>
</build>
```

注：

（1）devtools可以实现页面热部署（即页面修改后会立即生效，这个可以直接在application.properties文件中配置spring.thymeleaf.cache=false来实现），实现类文件热部署（类文件修改后不会立即生效），实现对属性文件的热部署。即devtools会监听classpath下的文件变动，并且会立即重启应用（发生在保存时机），注意：因为其采用的虚拟机机制，该项重启是很快的

（2）配置了true后在修改java文件后也就支持了热启动，不过这种方式是属于项目重启（速度比较快的项目重启），会清空session中的值，也就是如果有用户登陆的话，项目重启后需要重新登陆。

默认情况下，/META-INF/maven，/META-INF/resources，/resources，/static，/templates，/public这些文件夹下的文件修改不会使应用重启，但是会重新加载（devtools内嵌了一个LiveReload server，当资源发生改变时，浏览器刷新）。

4.2 devtools的配置

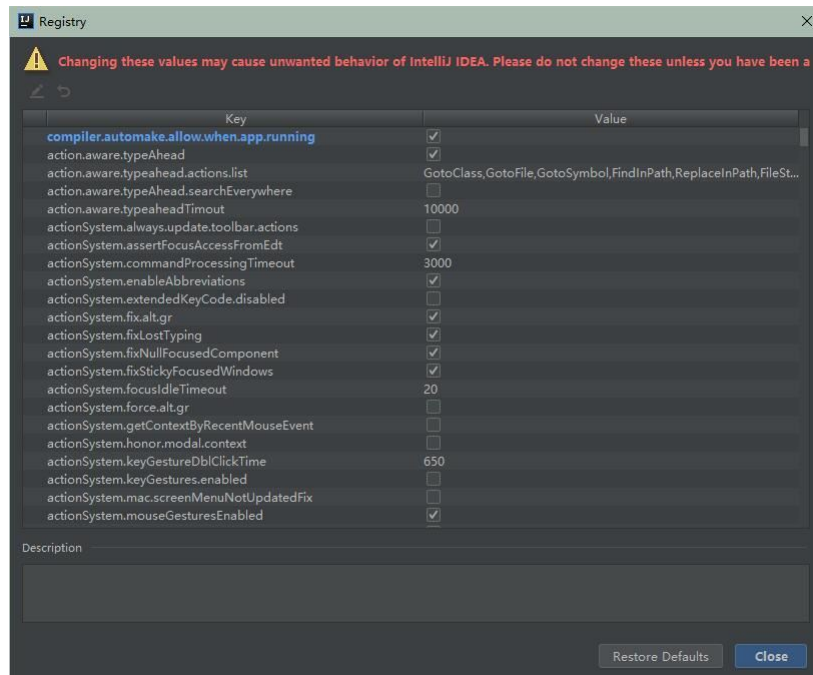
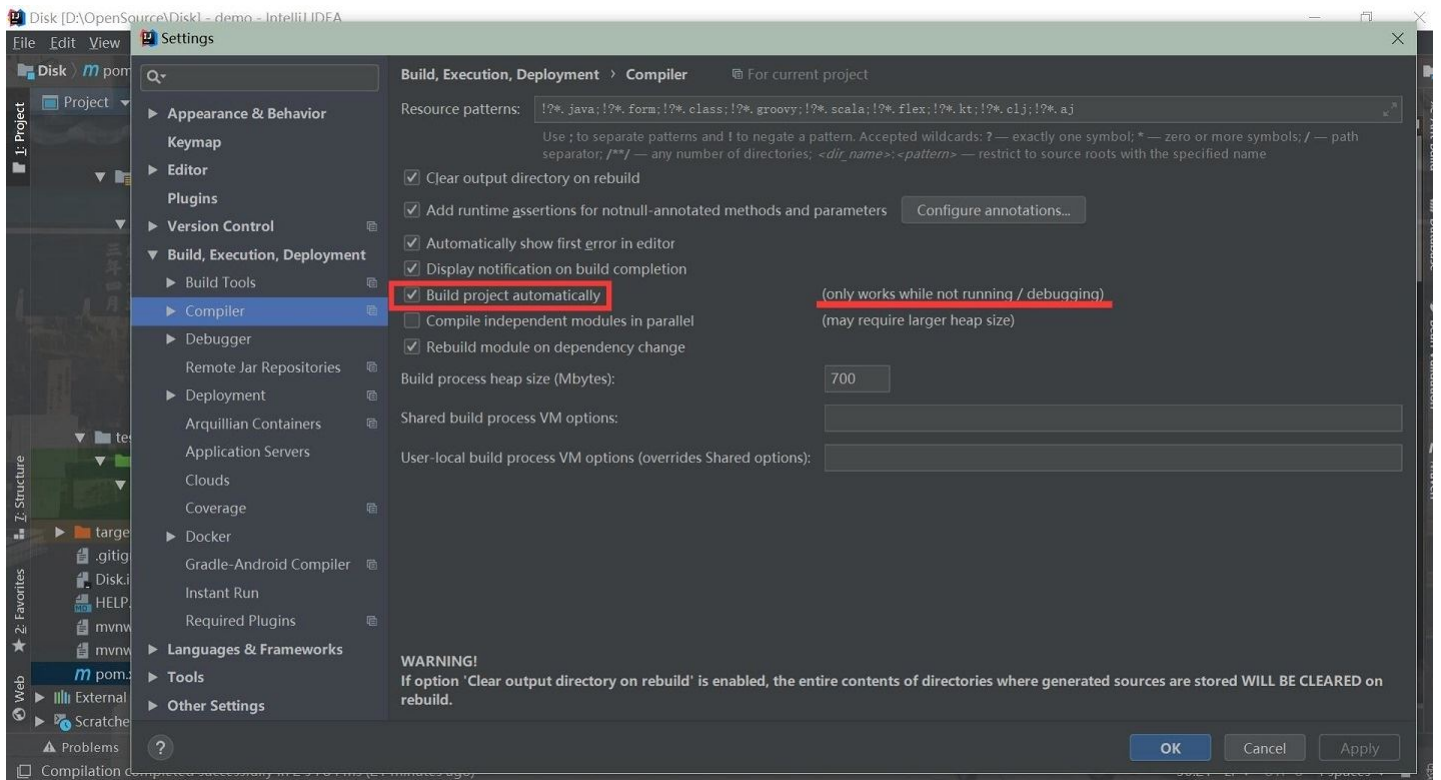
在application.yml中配置一下devtools

```
spring:
devtools:
restart:
enabled: true #设置开启热部署
additional-paths: src/main/java #重启目录
exclude: WEB-INF/**
freemarker:
cache: false #页面不加载缓存，修改即时生效
```

4.3 IDEA中配置

当我们修改了类文件后，idea不会自动编译，得修改idea设置。

- （1）File-Settings-Compiler-Build Project automatically
- （2）ctrl + shift + alt + / ,选择Registry,勾上 Compiler autoMake allow when app running



4.4 测试

修改类→保存：应用会重启

修改配置文件→保存：应用会重启

修改页面→保存：应用不会重启，但会重新加载，页面会刷新

以上就是本文的全部内容，希望对大家的学习有所帮助，也希望大家多多支持我们。