<http://blog.csdn.net/zknxx/article/details/53428669>

在SpringBoot中启用热部署是非常简单的一件事，因为SpringBoot为我们提供了一个非常方便的工具spring-boot-devtools，我们只需要把这个工具引入到工程里就OK了，下面我就说一下怎么引入spring-boot-devtools。

## 修改pom.xml

我们只需要在pom.xml里，引入相应的jar包就行了。

**[java]** [view plain](http://blog.csdn.net/zknxx/article/details/53428669) [copy](http://blog.csdn.net/zknxx/article/details/53428669)

1. <dependency>
2. <groupId>org.springframework.boot</groupId>
3. <artifactId>spring-boot-devtools</artifactId>
4. <optional>**true**</optional><!-- optional=**true**,依赖不会传递，该项目依赖devtools；之后依赖myboot项目的项目如果想要使用devtools，需要重新引入 -->
5. </dependency>

## 重新编译类文件

当我们修改了一个java类的时候，我们只需要重新编译一下，SpringBoot的就会重启了。因为devtools会监听classpath下的文件变动，所以当java类重新编译的时候，devtools会监听到这个变化，然后就会重新启动SpringBoot。这个重启是非常快的一个过程。因为在SpringBoot中有两个类加载器，一个是加载工程外部资源的，如jar包，还有一个类加载器是用来加载本工程的class的。所以在重启SpringBoot的时候只加载本工程的class文件。

## 监听文件夹的变化

如果你不想重新编译java类的话，还有一种方式用来让SpringBoot重启，那就是让devtools监听文件夹的变化：比如我们想让com.zkn.learnspringboot这个文件夹下的文件改变的时候，重新启动SpringBoot，那么我们只要在application.properties中添加这样一句话就行了：spring.devtools.restart.additional-paths=com\\zkn\\learnspringboot。

## 页面热部署

devtools可以实现页面热部署，即页面修改后会立即生效。你可以在application.properties文件中配置spring.thymeleaf.cache=false来实现这个功能。

版权声明：本文为博主原创文章，未经博主允许不得转载。

* 本文已收录于以下专栏：
* [SpringBoot系列](http://blog.csdn.net/column/details/zkn-springboot.html)