热部署 更新代码后，不需重启服务器

Spring的热部署

Java热部署、热加载

不重启服务器编译/部署工程

基于java 的类加载器实现

热部署与热加载的区别

部署方式

服务器运行时，重新部署工程，不停服务器，直接替换工程

服务器运行时，热加载重新加载修改的class

热部署 ——重新加载整个应用程序

热加载 运行时重新加载class

热部署，生产环境使用

热加载，开发环境使用

Java类加载过程

初始化jvm》产生启动类加载器》标准扩展类加载器

》系统类加载器》加载class文件

类加载的过程

加载》验证》准备》解析》初始化

加载：找到class字节码加载到jvm，形成运行时数据结构（方法去）

验证：检查字节码是否安全，保证不会对虚拟机产生威胁

准备：成员变量赋值初值——初始值（int类型的为0）

解析：符号引用变为直接引用

初始化：classinit

初始化时机，new static invok static put static

final不会触发初始化。他会把常量放入常量池

反射会初始化，子类调用，父类需要初始化，主类

1、热部署

类的热加载

配置tomcat

Java类加载器特点

1. 由AppClassLoader(系统类加载器)开始加载指定类
2. 类加载器将加载任务給其父类，如果其父类找不到，再由自己加载
3. BootstrapLoader(启动类加载器)是最顶级的类加载器

Java类的热部署

1. 类的热加载
2. 配置Tomcat

继承java.lang.ClassLoader，重写findClass方法

原方法：

**protected** Class<?> findClass(String name) **throws** ClassNotFoundException {

**throw** **new** ClassNotFoundException(name);

}

改写后的方法：

@Override

**protected** Class<?> findClass(String name) **throws** ClassNotFoundException {

System.***out***.println("加载类===" + name);

**byte**[] data = loadClassData(name);

**return** **this**.defineClass(name, data, 0, data.length);

}

通过tomcat配置

1. 直接把项目web文件夹放在webapps里。

启动tomcat，将编译好的工程，copy到webapps就好了

1. 在tomcat\conf\server.xml中的<host></host>内部添加<context/>标签。

<Context debug=”0” docBase=”D:/Imooc/web” path=”/hot” privileged=”true” reloadable=”true”/>

将编译好的工程 meta-inf、web-inf，copy到D:/Imooc/web就好了

1. 在%tomcat\_home%\conf\Catalina\localhost中添加一个xml

xml内容,hot.xml 文件名作为webcontext

<?xml version=”1.0” encoding=”UTF-8”?>

<Context docBase=”D:/Imooc/web” reloadable=”true”/>

Springboot热部署

Springloaded实现热部署

以maven的方式启动

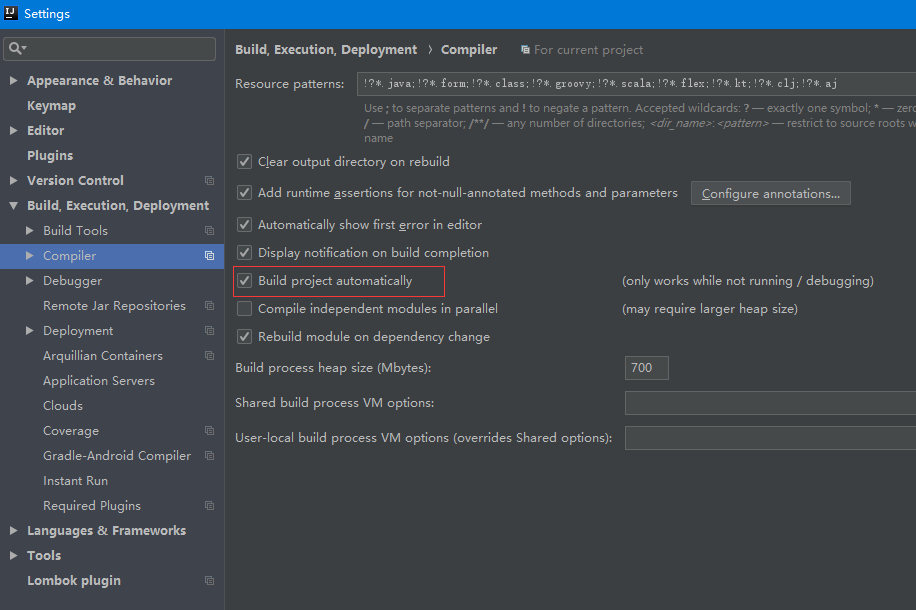
需要依赖的jar

|  |
| --- |
| <!-- https://mvnrepository.com/artifact/org.springframework/springloaded -->  <dependency>  <groupId>org.springframework</groupId>  <artifactId>springloaded</artifactId>  <version>1.2.8.RELEASE</version>  </dependency>  <!-- https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-devtools -->  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-devtools</artifactId>  <version>2.1.2.RELEASE</version>  </dependency> |

pom.xml中配置的位置

|  |
| --- |
| <build>  <plugins>  <plugin>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-maven-plugin</artifactId>  <dependencies>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>springloaded</artifactId>  <version>1.2.8.RELEASE</version>  </dependency>  </dependencies>  </plugin>  </plugins> </build> |

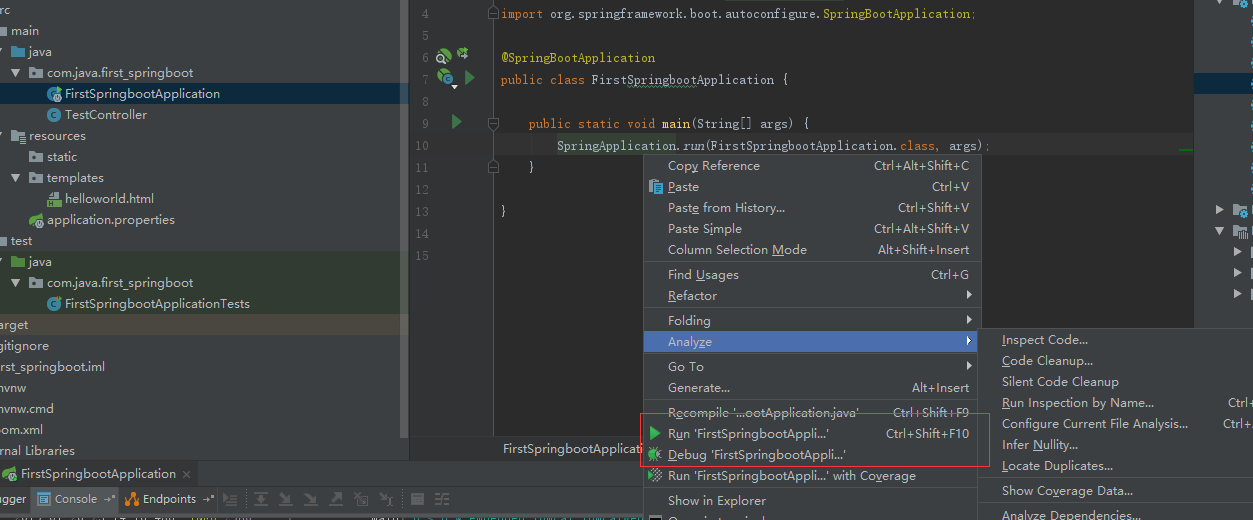
使用idea开发，需要设置



maven命令：mvn spring-boot:run

在eclipse中开发，不需要spring-boot-devtools依赖

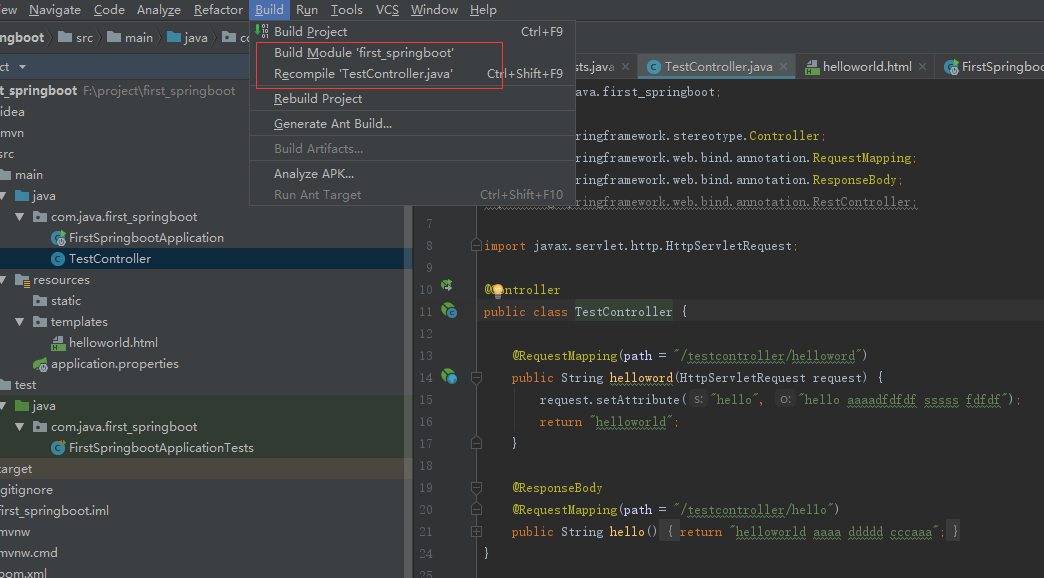
1. 使用run as方法启动spring-boot程序



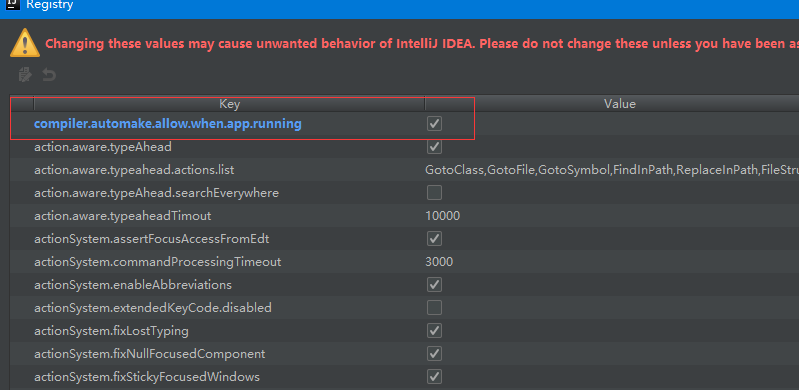
需要配置vm参数：

|  |
| --- |
| -javaagent:F:\maven-repository\org\springframework\springloaded\1.2.8.RELEASE\springloaded-1.2.8.RELEASE.jar -noverify |

修改了工程文件之后，需要手动build project，或者recompile 文件



如果不想手动build和recompile，需要做如下配置：compiler.automake.allow.when.app.running



新版的idea找不到 compiler.automake.allow.when.app.running 。可以在settings中搜索registry，添加一个registry的keymap快捷键。然后用快捷键可以打开上面的界面，就可以找到这个参数，然后设置。

注：不需要再spring-boot-maven插件中添加springloaded依赖

Spring-boot-devtools实现热部署

|  |
| --- |
| <!-- https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-devtools -->  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-devtools</artifactId>  <version>2.1.2.RELEASE</version>  </dependency> |