



链滴

热部署的那些事

作者: [sirwsl](#)

原文链接: <https://ld246.com/article/1596266115501>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



一、起因

马上就大四了，准备写个能够拿得出手的项目面试时候用，恰好现在也在实训，然后准备开始写项目然后想起来，之前用eclipse的时候eclipse可以热部署，虽然用idea好长时间了，但是发现我以前用idea没有在意过这个问题，每次都是重启项目，哪怕做个很小的更改也是在不断的重启项目。现在一想自己的好多时间都用在了重启项目上去了，然后就有了这篇文章。

二、基础知识：

使用场景

本地调试、线上发布、频繁的部署、启动耗时长的无法停止的服务应用。

eg：我们在开发比较大的系统的时候，很多时候在debug的时候修改了代码然后就重启服务，但是有大的服务每次启动都会耗时很长，这时候本地调试的热部署就出现了。

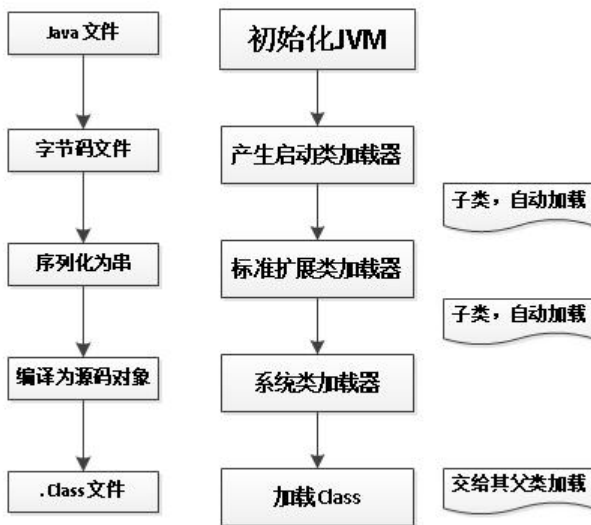
eg：有时候想过这样的一个问题吗？例如英航系统需要全年不间断的运行，但是银行系统真的就不需升级或者优化吗？在升级期间下线所有服务吗？这时候热部署就出现了。

优点

- 无论本地还是线上都适用，无需重启服务
- 能够提高开发、调试的效率
- 能够提高发布、运维的效率，降低运维的成本

热部署与热加载

类加载过程:



类加载器特点:

- 由AppClass Loader（系统类加载器）开始加载指定的类
- 类加载器加载任务交给其父，如果其父找不到，再自己去加载。
- Bootstrap Loader(启动类加载器)是最顶级的类加载器

热加载五个阶段:

- **加载**：找到类的静态存储结构后加载到虚拟机中，然后转化方法区为运行时的数据结构，生产Class象。用户可以自定义类加载器参与进来
- **验证**：确保字节码是安全的，以保证不会对虚拟机造成危害，可以通过java虚拟机的一些参数来限一些验证。但有可能可能会对虚拟机安全有所损害。
- **准备**：确定内存布局，初始化类遍历，只是赋初始值，并不是会执行程序自己的赋值操作。
- **解析**：将符号应用变为直接引用
- **初始化**：调用程序自定义代码，eg;程序真实赋值进行初始化。

共同点： 不重启服务器就能够部署相符，都是基于java类加载器实现的

部署方式:

热部署在服务器运行时候重启加载项目

热加载在运行时候重新加载class

实现原理:

热部署直接加载整个应用，需要耗费更多的时间

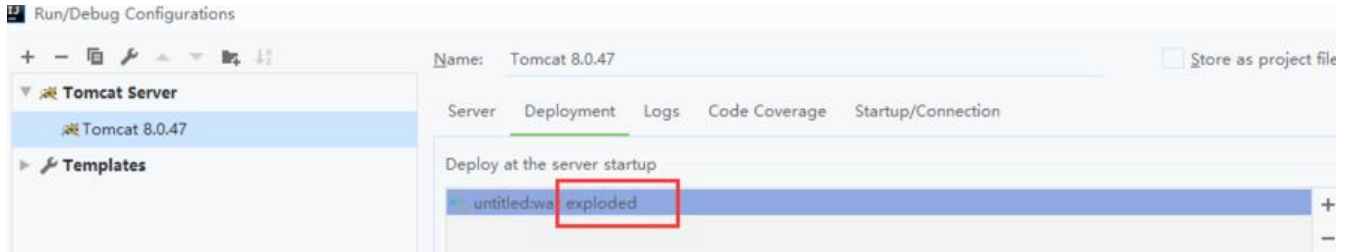
热加载依赖java类加载机制，在容器运行时候通过重新加载来改变类的信息

安全性:

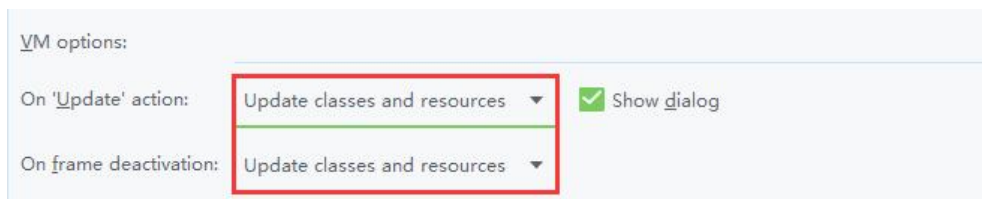
热加载直接修改java虚拟机中的字节码的信息，造成难以监控，无法记录代码逻辑的变化，越注重安的应用风险越高。热部署更多的是在生产环境中使用，热加载则更多的是在开发环境中使用。

三、idea自带热部署

其实idea也有自带的部署，我们在部署的时候讲发布模式改为exploded而不是war模式



设置VM option:



接下来简单说一下war与exploded区别:

war模式下，每次重启都会讲工程打包为war包。然后讲war包推送给tomcat运行。在发布后更改项中的东西，由于没有重启重新打包，所以资源不会更新。

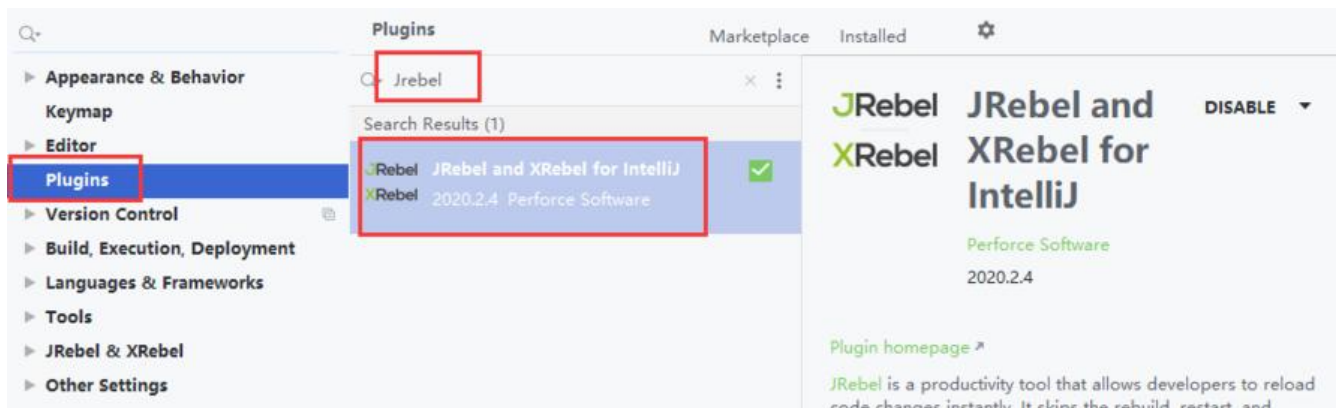
exploded模式下获取的是target所在路径，之后跟新target下的class文件后就会起到更新的作用。

四、idea插件jrebel热部署

jrebel是我最推荐的插件，贼好用，比起其他的来，简直是方便快捷。接下来说一下**下载与破解**（PS不得用于商用，仅做个人学习使用，如果有钱还是必须要支持一下正版的，毕竟作为程序员都理解）

安装:

在IDEA>Setting>Plugins下搜索Jrebel



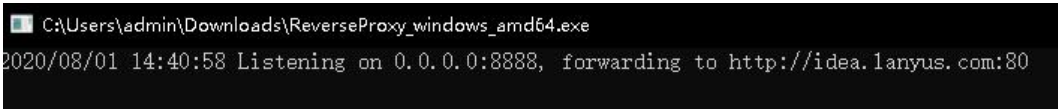
然后install，好了以后重启，这就安装结束了，接下来说一下**破解**（再次声明：只作为个人学习使用有钱的请支持正版）

破解之前需要两个东西：

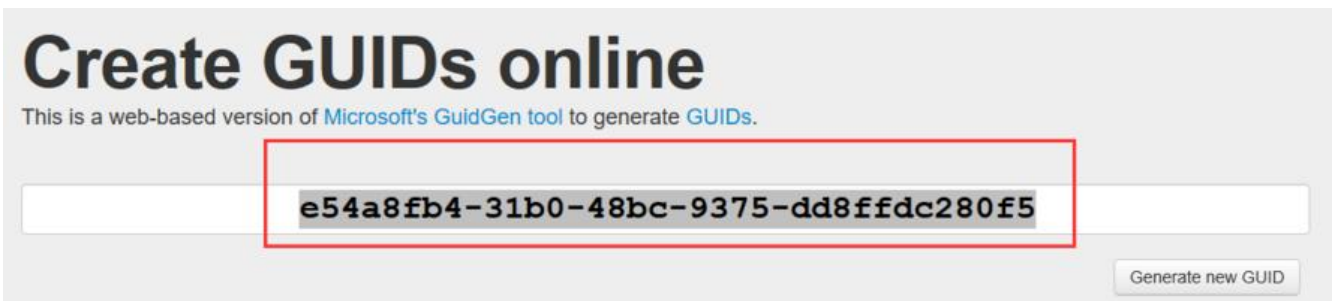
- 反代理软件：去GitHub上下载：<https://github.com/ilanyu/ReverseProxy/releases/tag/v1.4>
- GUID（一会用）：在线生成网址：<https://www.guidgen.com/>



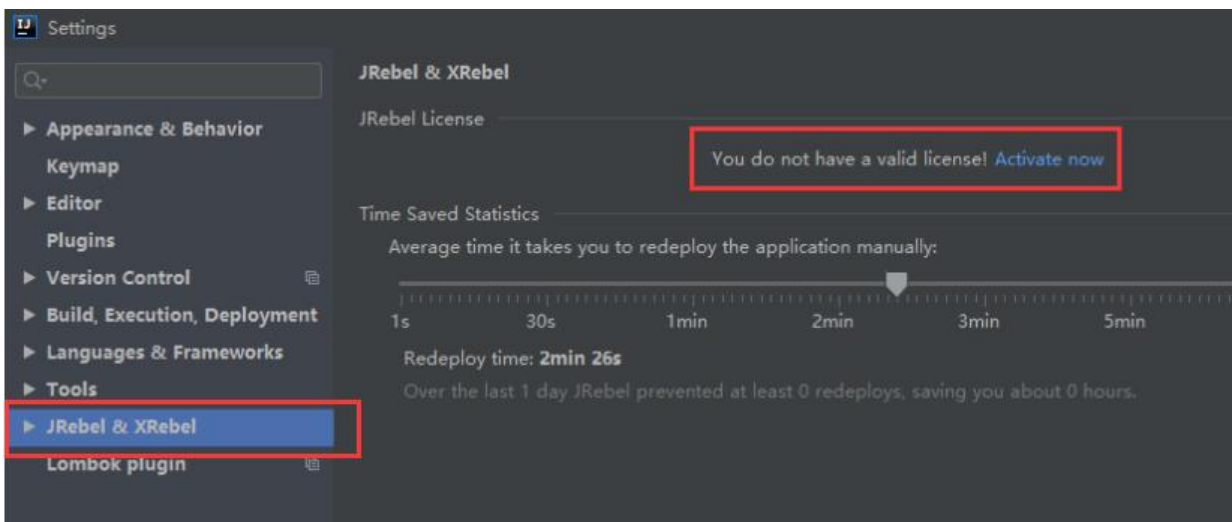
这是我下载的，下载之后直接打开可以看见：



然后不要关闭。接着打开GUID的网页，复制GUID



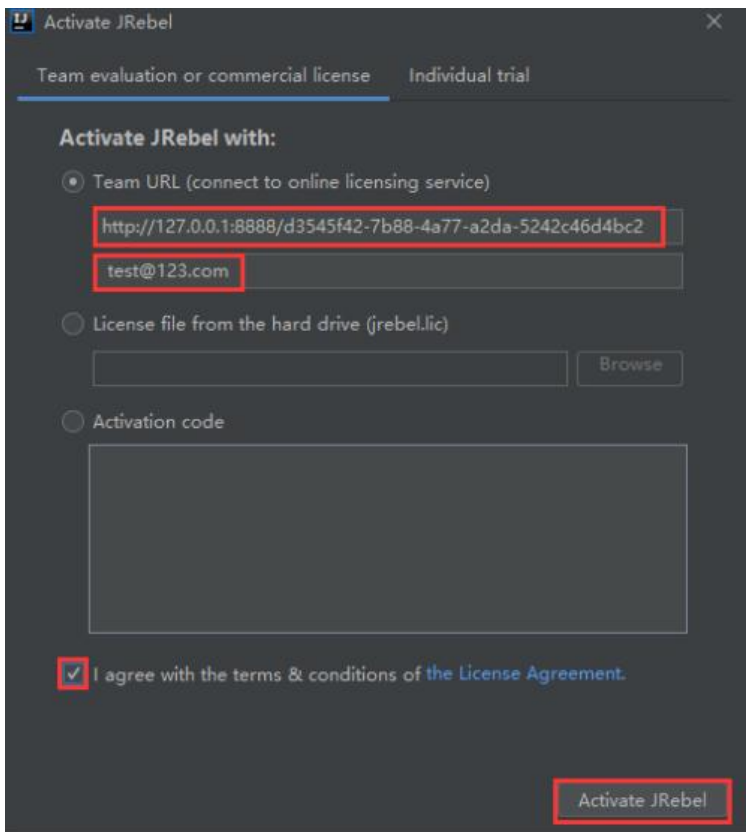
接下来在idea中File>Setting>JRebel&XRebel



选择Team URL

第一行填写：<http://127.0.0.1:8888/>生成的GUID

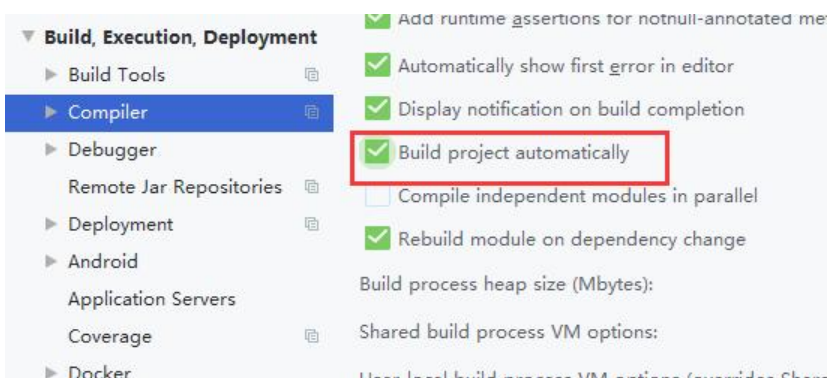
第二行填写邮箱：test@123.com



之后可以看见有限期180天



设置自动编译功能: File>Setting>Builder>Compiler



保存退出后按shift+Ctrl+Alt+/. 选择Registry. 之后勾选compiler.automake.allow.when.app.runnng



接下来你就可以看见原来的地方多了几个东西



然后第一个是run，第二个是debug。你点第一个运行。热部署就实现了

五、springboot热部署

在springboot中由springloaded与spring-boot-devtools两种热部署方式

springloaded

在build>plugins>plugin>dependencies>下面引入依赖

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>springloaded</artifactId>
  <version>1.2.6.RELEASE</version>
</dependency>
```

之后通过配置运行环境或者mvn spring-boot:run运行（个人不喜欢，所以不写具体步骤了不喜欢原如下）

存在问题：配置略显麻烦，每次使用时候，只能在原有代码中修改（eg：修改返回值之类）如果新加方法或者其他的就会无效需要重新运行。

spring-boot-devtools

直接在maven中添加依赖：

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

如果是springboot项目，然后没有插件，这个方式是最为推荐的，**最大的优点简单。直接设置option I为true就OK。**

end

今天就到此为止