

# IntelliJ IDEA 提升效率开发插件必备

工欲善其事，必先利其器，好的工具可以提升我们的开发效率，下面介绍几款个人觉得比较好的编辑器插件，不仅炫酷更重要可以提高你的工作效率。

本文是作者辛苦整理的 16 款插件，每个都是超级实用的，不好不介绍，相信体验过后才知道它的好。

Activate-power-mode

pojo-to-json

SequenceDiagram

Free Mybatis plugin

Key promoter X

CodeGlance

JUnitGenerator V2.0

GenerateAllSetter

ideavim

sonarlint

restfultoolkit

Translation

GenerateSerialVersionUID

Maven Helper

Lombok Plugin

JRebel for IntelliJ

### **Activate-power-mode 插件**

我喜欢称这个插件叫，“会跳舞的神器”，先看下效果

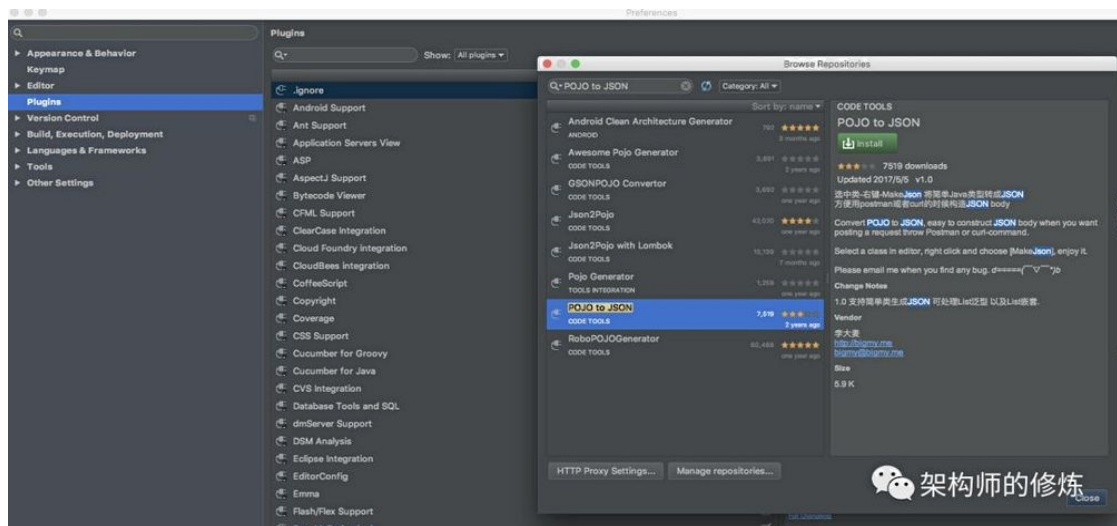
```
15 <div class="container">
16     <header>
17         <h1>Hello World</h1>
18     </header>
19 </div>
```

你的代码在跳舞，跟随我的脚步，免费蹦迪，此插件有玩耍娱乐之嫌，实用性不高，耍酷还可以。下面介绍要说的重点插件，保证让你满意而归。

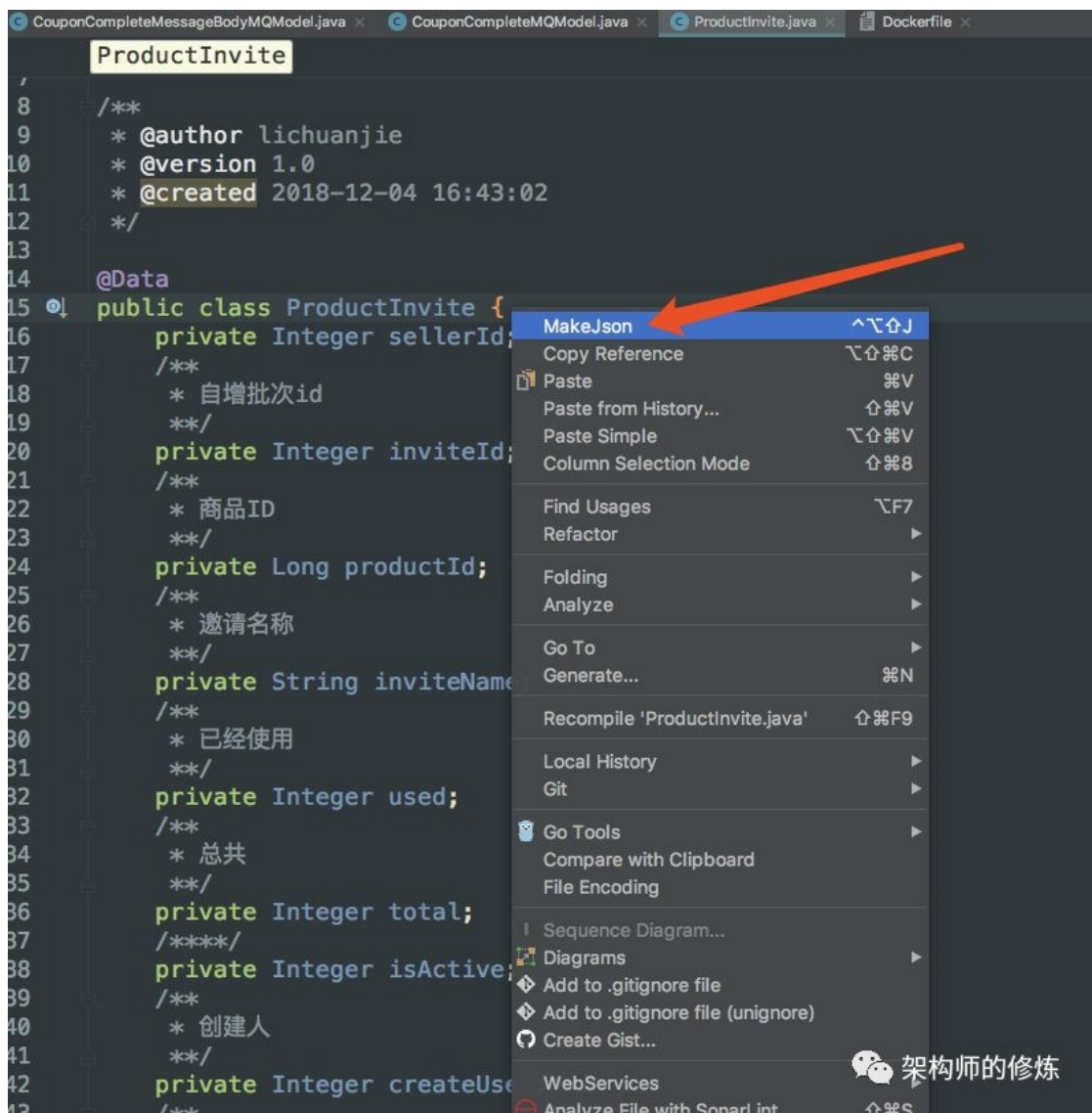
### **pojo-to-json idea 插件**

工作中，为了测试需要，我们需要将简单 Java 领域对象转成 JSON 字符串方便用 postman 或者 curl 模拟数据

官方详细介绍地址



安装插件后重启 Idea, 选择一个类，右键：



会复制生成的 JSON 到粘贴板中：ctr+V 就可以粘贴出来。

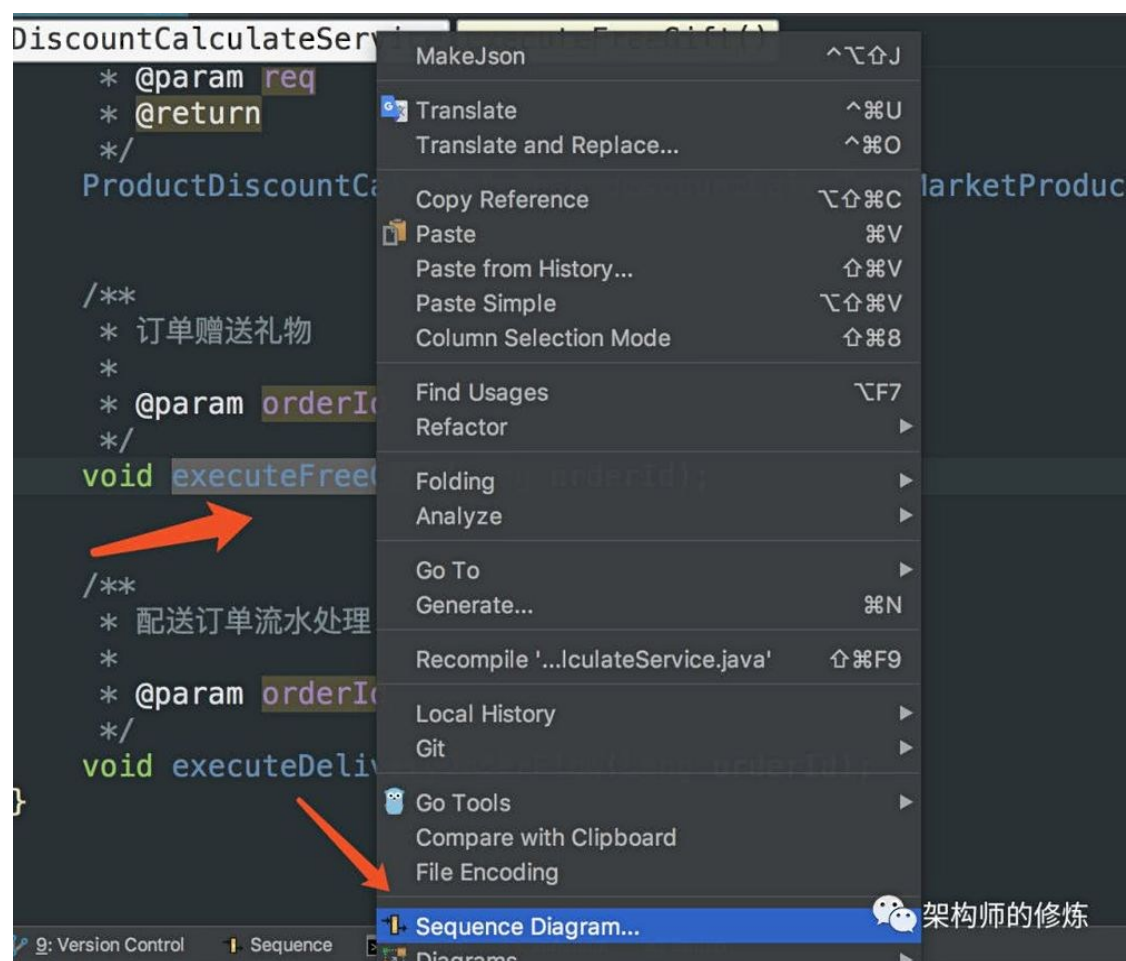
## 时序图生成工具：SequenceDiagram 插件

有的时候，我们需要梳理业务逻辑或者阅读源码。从中，我们需要了解整个调用链路，反向生成 UML 的时序图是强需求。其中，SequenceDiagram 插件是一个非常棒的插件。

官方详细介绍地址

<https://plugins.jetbrains.com/plugin/8286-sequencediagram>

比如：你想看某个方法的执行流程，可以选择右键这个方法接口或者方法实现



立即生成时序图



## Free Mybatis plugin 插件

MyBatis 框架已经应用于各个地方。因此，围绕着 MyBatis 的插件和工具越来越多。Free Mybatis plugin 非常方便进行 Mapper 接口和 XML 文件之间跳转。

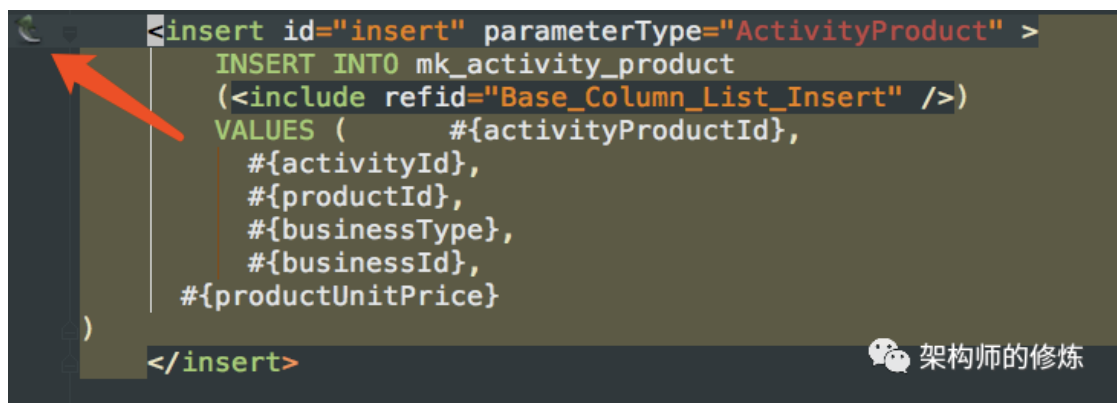
官方详细介绍地址

<https://plugins.jetbrains.com/plugin/8321-free-mybatis-plugin>

安装之后在 Java dao 层会生成一个标记，点击标记立即跳转到 xml 文件指定位置，非常方便：



并且 xml 里面还可以跳转会 java dao 层指定的方法：

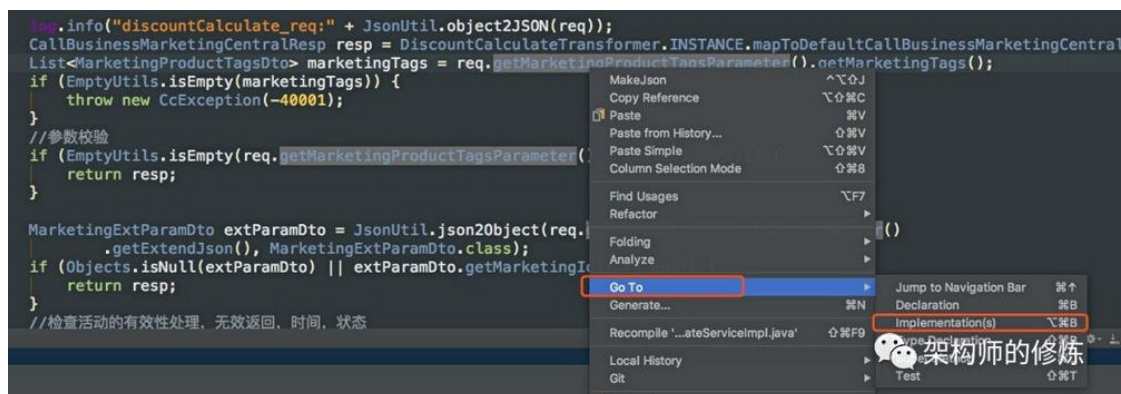


## 快捷键提示工具：Key promoter X

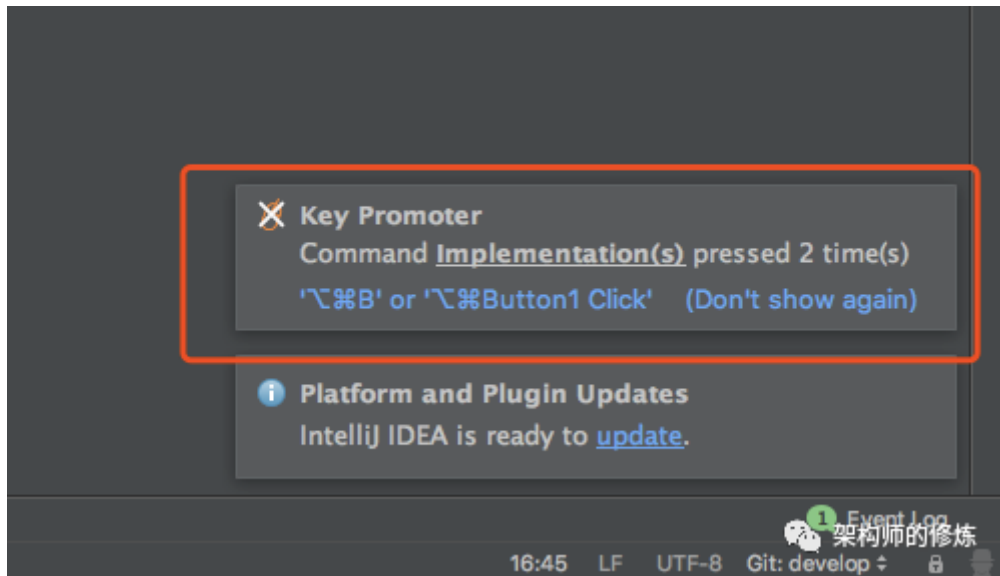
官方详细介绍地址

<https://plugins.jetbrains.com/plugin/9792-key-promoter-x>

这个插件有什么用呢？举个例子，你想看某个方法内部实现的时候，如果你不用快捷键，你会这样鼠标点：



如果你安装了这个插件，你每次这样鼠标点击后，开发工具右下角会提示你，使用快捷键：



安装这个插件后，快捷键用的飞起来。

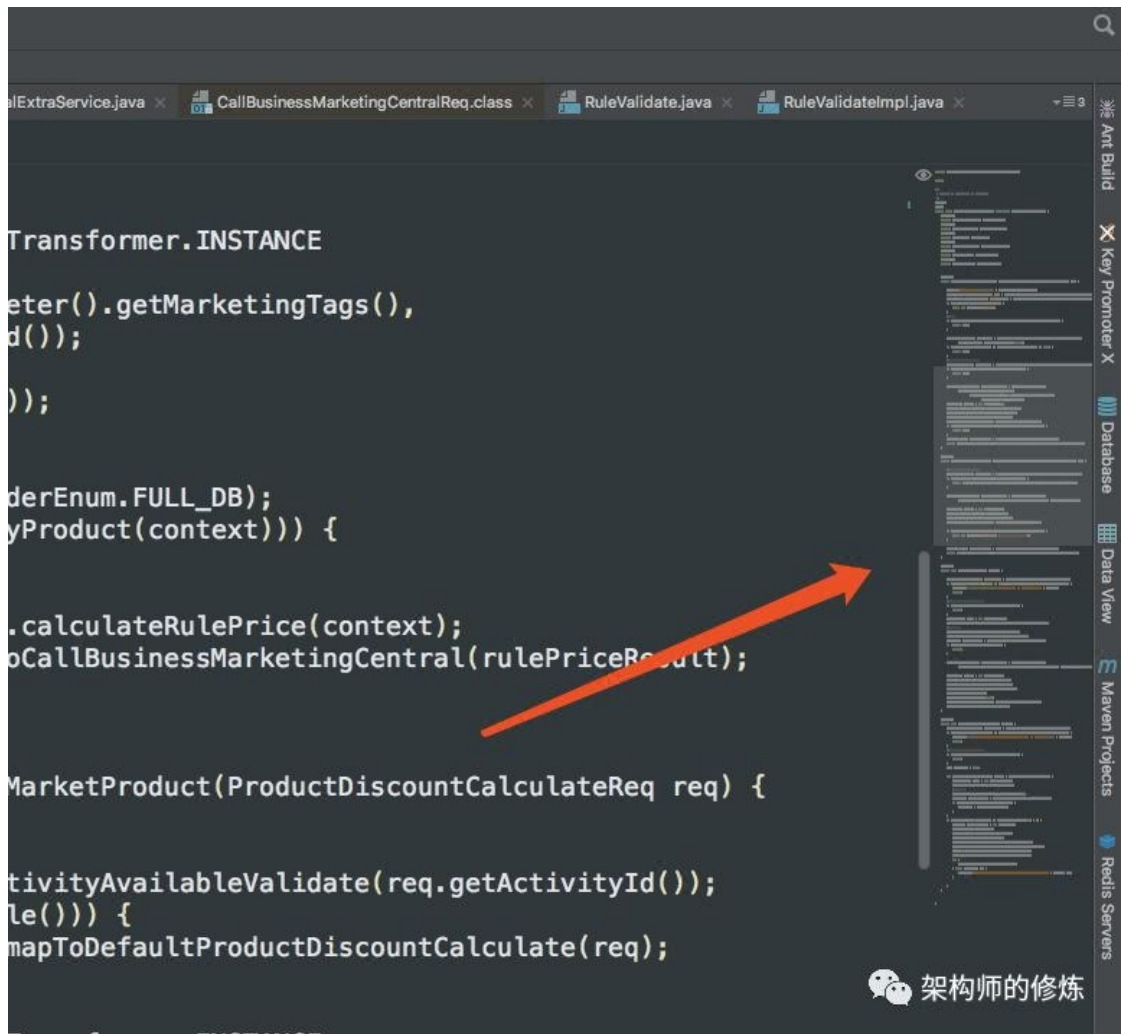
## CodeGlance 插件

当你的类上千行的时候，在你往下拖拽找对象的时候，是不是感觉挺累的，有个这个插件，可以查看缩略图一样，快速切换到自己需要去的地方，飞翔的感觉。

官方详细介绍地址 <https://github.com/Vektah/CodeGlance>

安装后显示效果图如下：



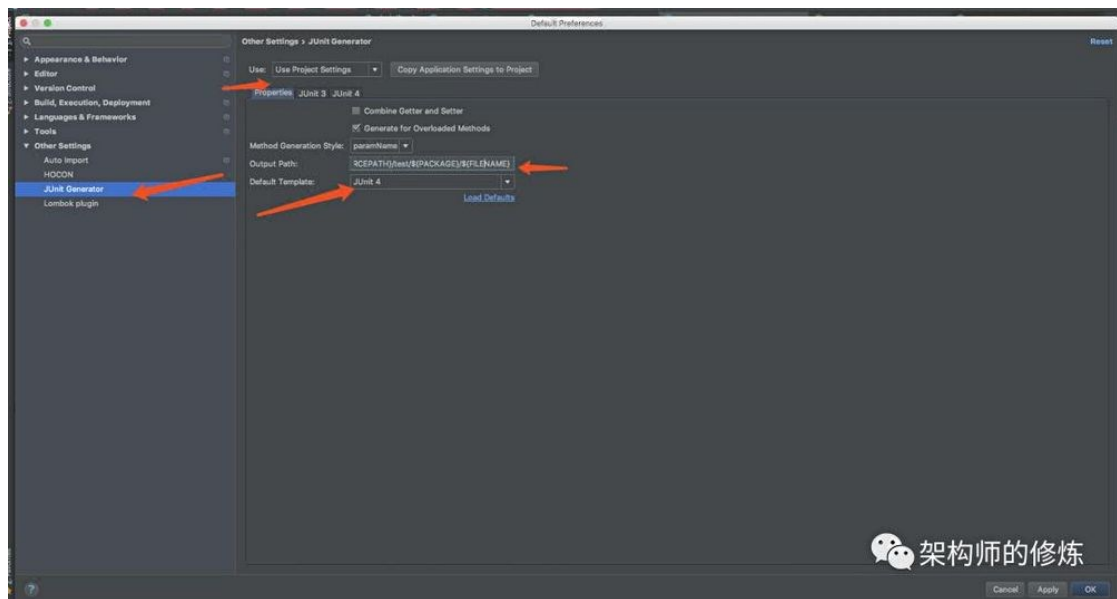


## JUnitGenerator V2.0 自动生成单元测试代码

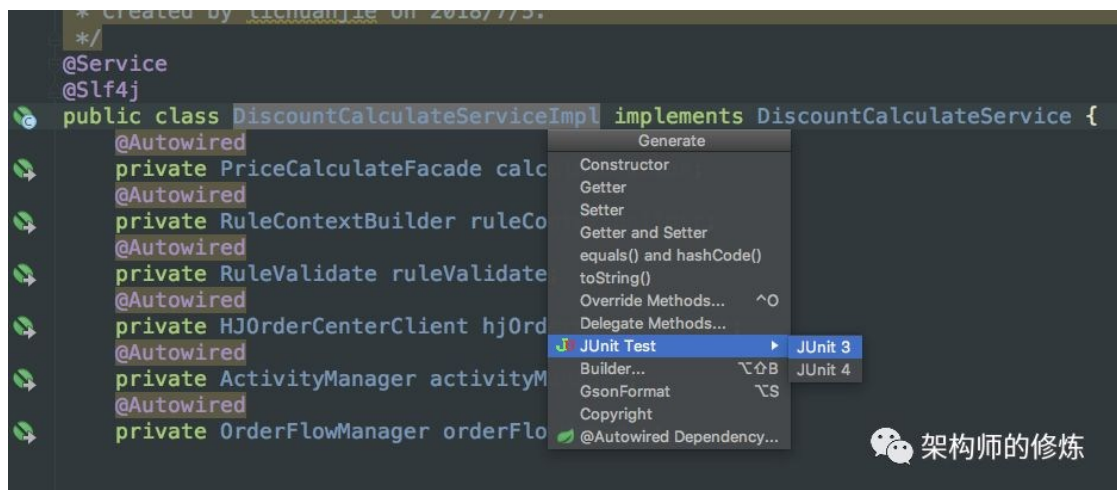
编写单元测试是好的开发者的习惯，但是怎么提高你代码的效率，单元测试代码可否自动生成，这个插件就是帮我们做这件事情。

修改 JUnitGenerator V2.0 的配置，Settings 下的 Other Settings 修改下模版和更新配置



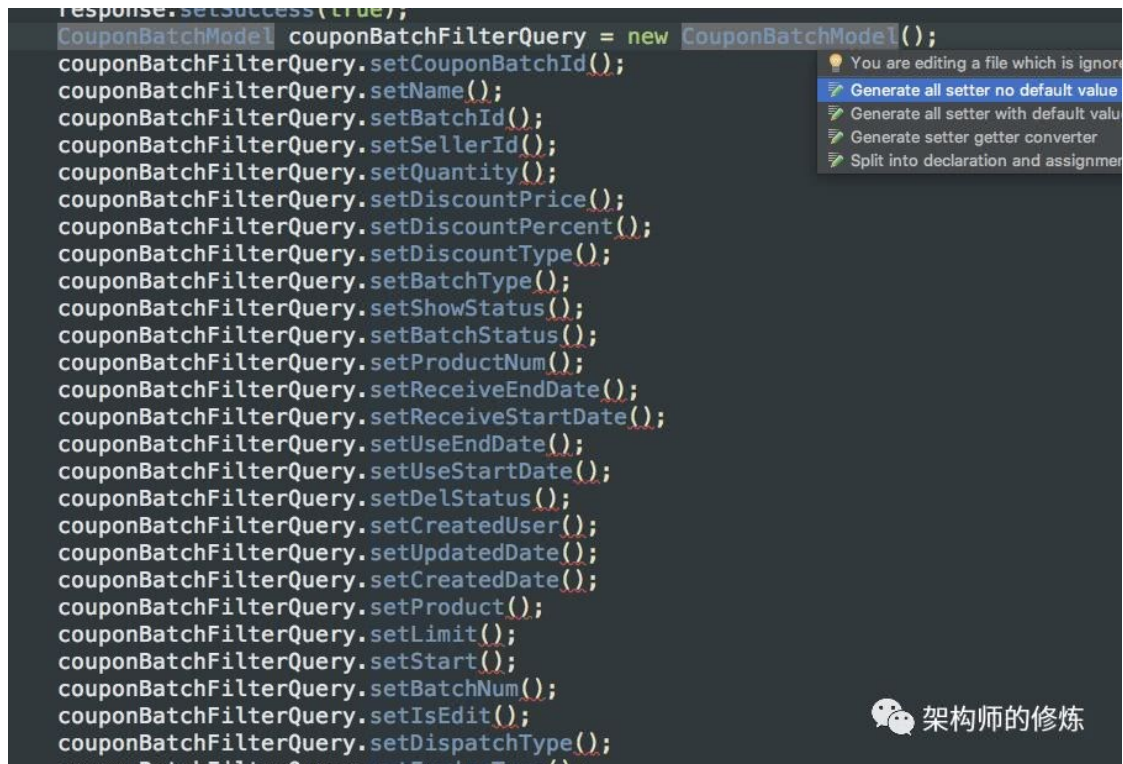


设置成功后，就可以根据方法或者类生成单元测试模版了：



## GenerateAllSetter 插件

当你进行对象之间赋值的时候，有没有觉得好麻烦，能不能有一个更好的办法呢~ 有，只要你选中需要生成 set 方法的对象，按下快捷键 `alt+enter` 界面如下：



有没有感觉很厉害， 官方详细介绍地址

<https://plugins.jetbrains.com/plugin/9360-generateallsetter>

## ideavim 插件

让你的代码如飞：可以让你在编辑器中使用 vim 的命令操作，来回切换，尽量脱离鼠标的限制，提高你的开发效率，这个插件 mac 上面使用效果更好。详细地址：

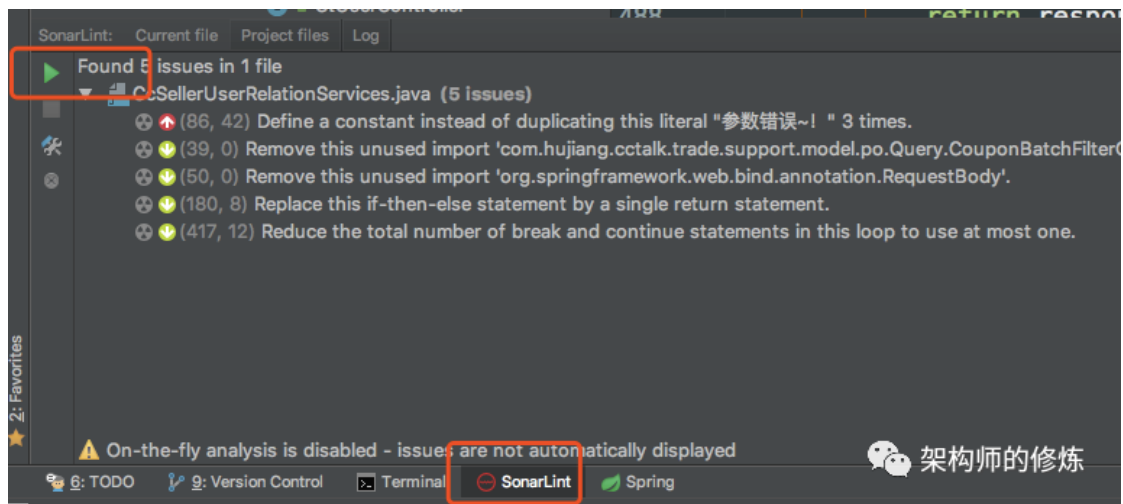
<http://plugins.jetbrains.com/plugin/164-ideavim>

## sonarlint 代码质量管理

找出你的代码隐藏 bug 或者坏味道

官方详细介绍地址 <http://intellij.sonarlint.org>

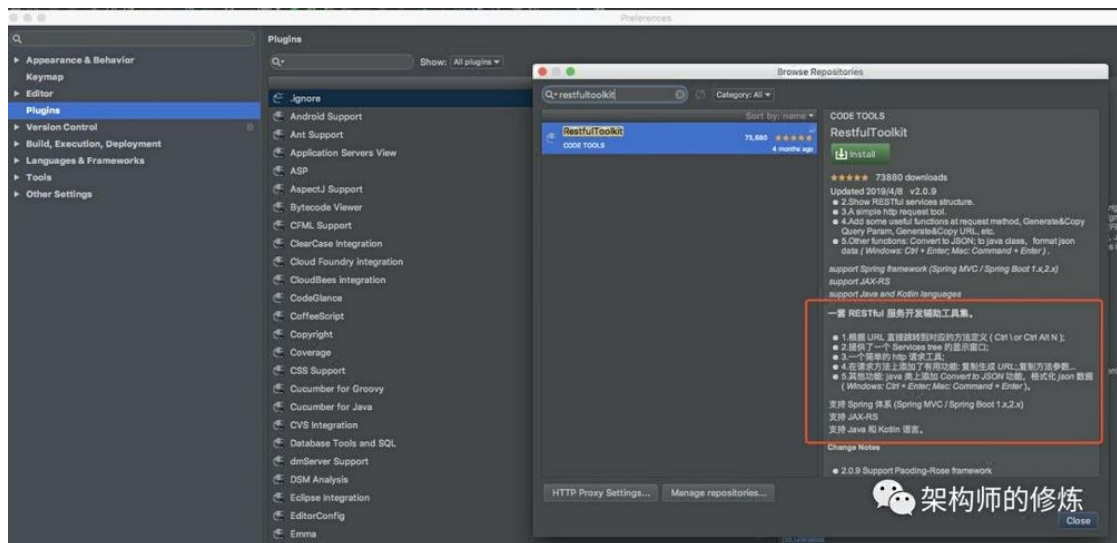
可以选择你要执行的代码类或者项目执行就可以类，下面会列出结果和意见



## restfultoolkit 插件

Java WEB 开发必备，再也不用全局搜索 RequestMapping 了

安装操作：

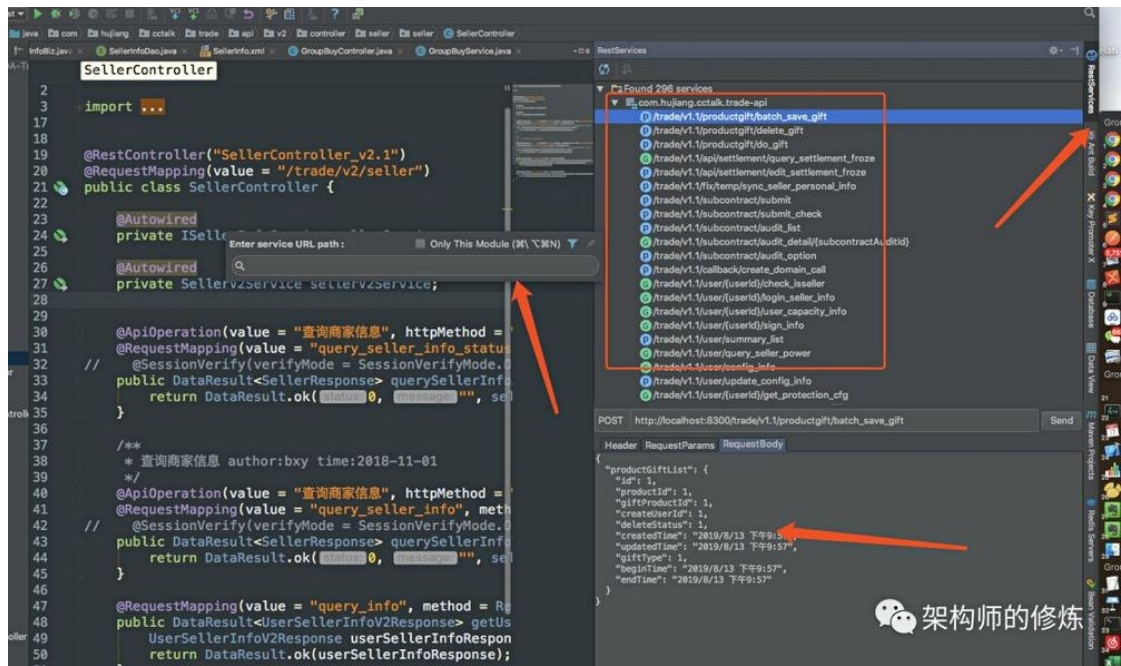


安装完毕之后使用，根据 URL 直接跳转到对应的方法定义

windows ( Ctrl + \ or Ctrl + Alt + N ) 而 mac 机器快捷键：

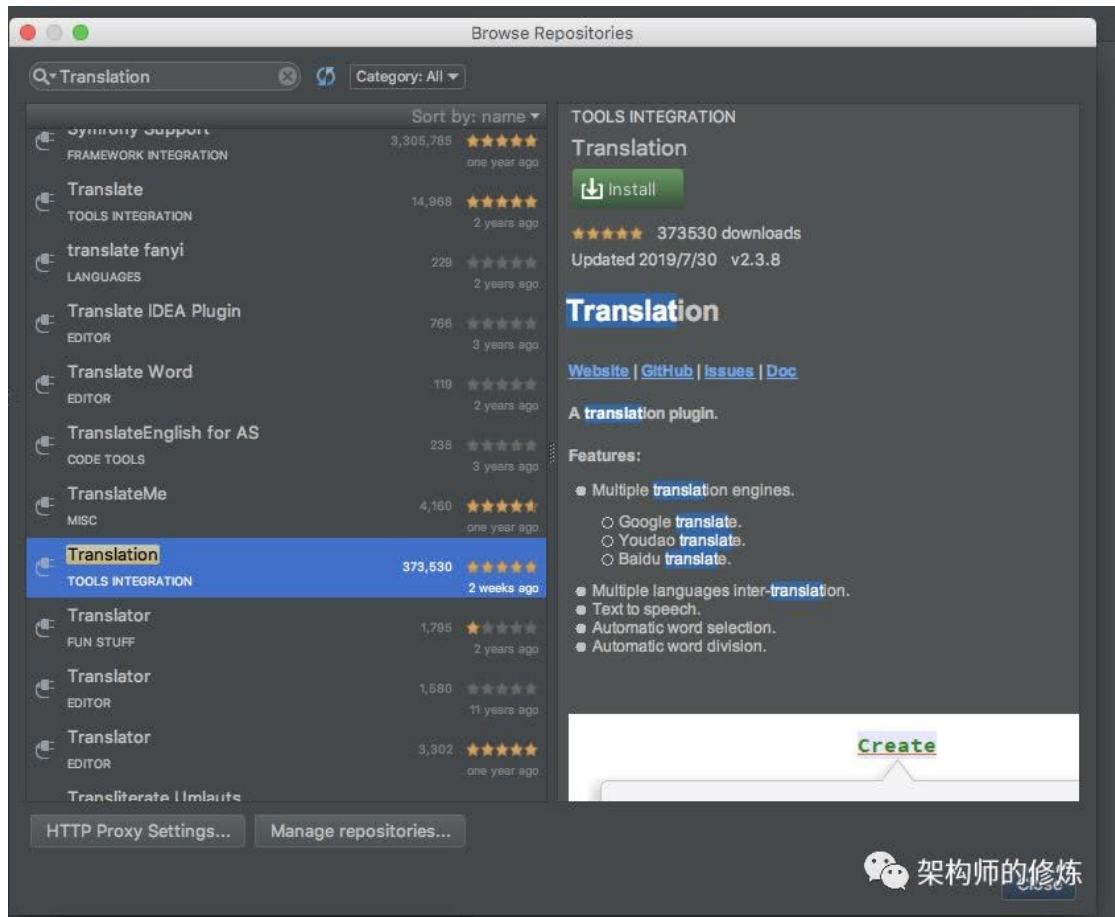
( Command + Option + N or Command + \ )

还可以查看 Services 中所有的接口 url list, 并且模拟请求，非常方便，简化操作



## 中英文翻译工具：Translation

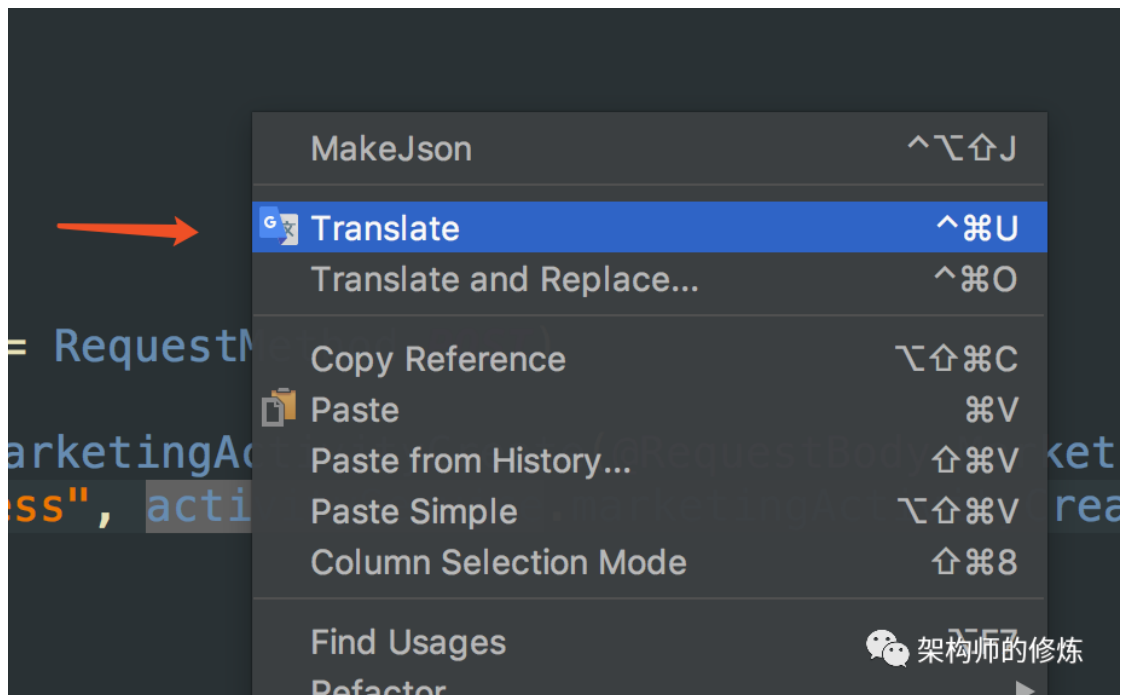
这个最长用就是阅读源码的时候，看别人的注释，英文不好的，可以安装一个。



官方详细介绍地址

<https://plugins.jetbrains.com/plugin/8579-translation>

选择要翻译的单词或者句子，右键或者快捷键，可以翻译操作：



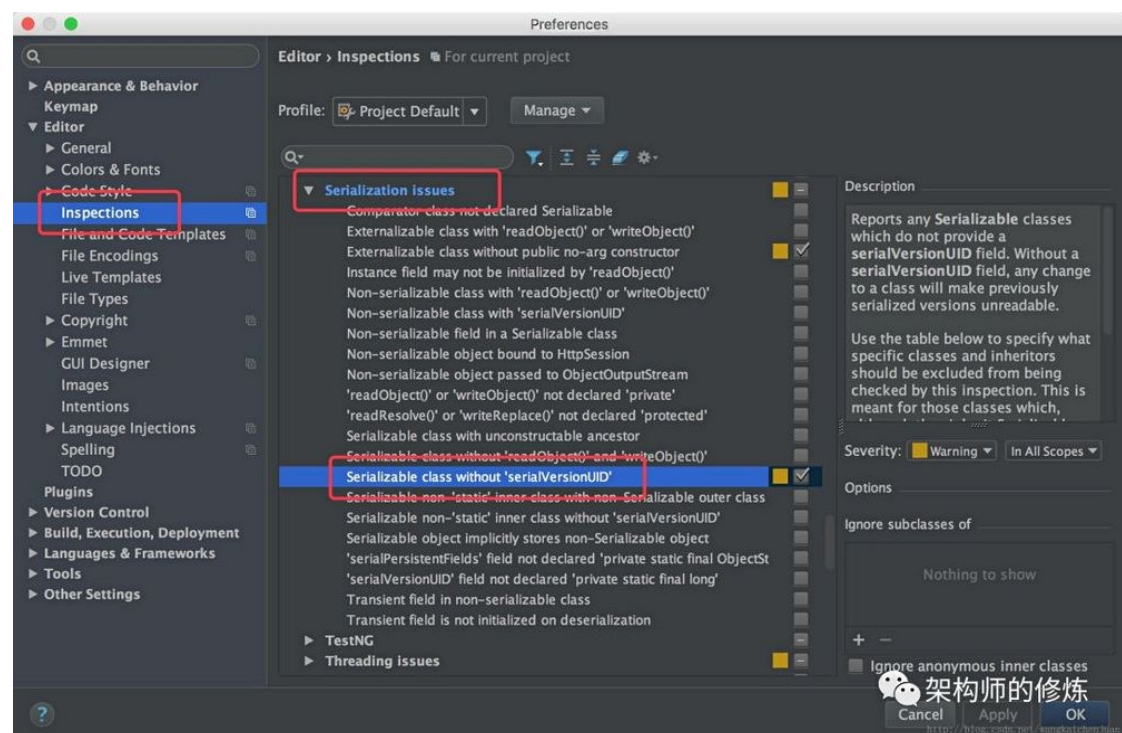


## GenerateSerialVersionUID 插件

Java 的序列化机制是通过在运行时判断类的 `serialVersionUID` 来验证版本一致性的。在反序列化时，JVM 会把传来的字节流中的 `serialVersionUID` 与本地相应实体（类）的 `serialVersionUID` 进行比较，如果相同就认为是一致的，可以进行反序列化，否则就会出现序列化版本不一致的异常。所以生成 `serialVersionUID` 会很频繁，这个插件一个快捷键帮你搞定 `Alt + Insert`

步骤 1：先安装插件，根据上面的名字搜索安装

步骤 2：设置 Inspections 功能



这时候，在 `javabeen` 里面，就可以看到编辑器右边有黄色的提示了，可以点开看看 将光标放到类名上，按 `atl+enter` 键，就会提示生成 `serialVersionUID` 了



```
1 package com.example.demo;
2
3 import java.io.Serializable;
4
5 public class UserBean implements Serializable {
6
7
8
9
10 }
```

(此图片来源于网络)

## Maven Helper 插件

Maven 的依赖机制会导致 Jar 包的冲突。举个例子，现在你的项目中，使用了两个 Jar 包，分别是 A 和 B。现在 A 需要依赖另一个 Jar 包 C，B 也需要依赖 C。但是 A 依赖的 C 的版本是 1.0，B 依赖的 C 的版本是 2.0。这时候，Maven 会将这 1.0 的 C 和 2.0 的 C 都下载到你的项目中，这样你的项目中就存在了不同版本的 C，这时 Maven 会依据依赖路径最短优先原则，来决定使用哪个版本的 Jar 包，而另一个无用的 Jar 包则未被使用，这就是所谓的依赖冲突。

幸运的是什么呢？

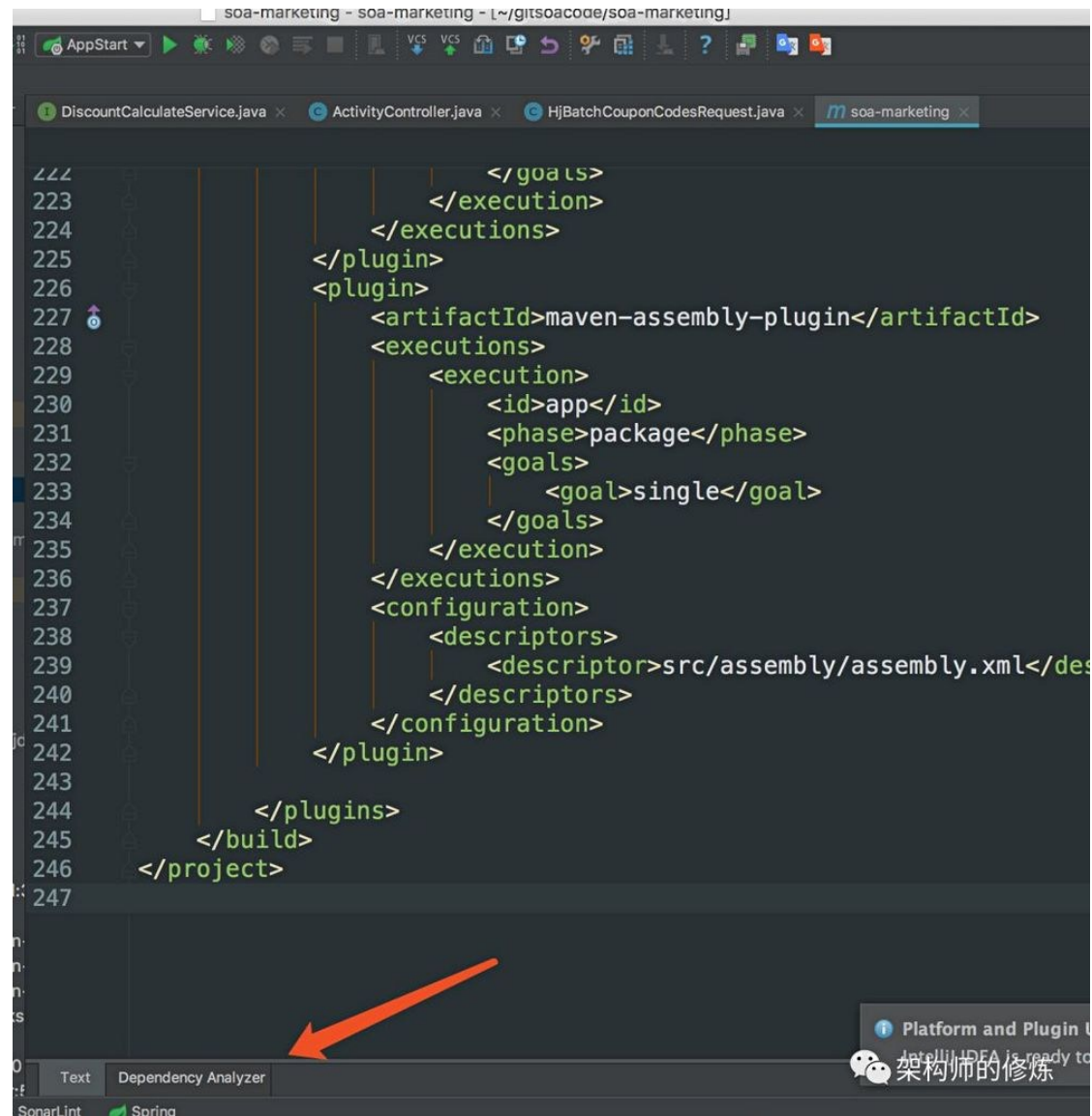
在大多数时候，依赖冲突可能并不会对系统造成什么异常，因为 Maven 始终选择了一个 Jar 包来使用。

但是不幸的却是，不排除在某些特定条件下，会出现类似找不到类的异常，所以，只要存在依赖冲突，在我看来，最好还是解决掉，不要给系统留下隐患。

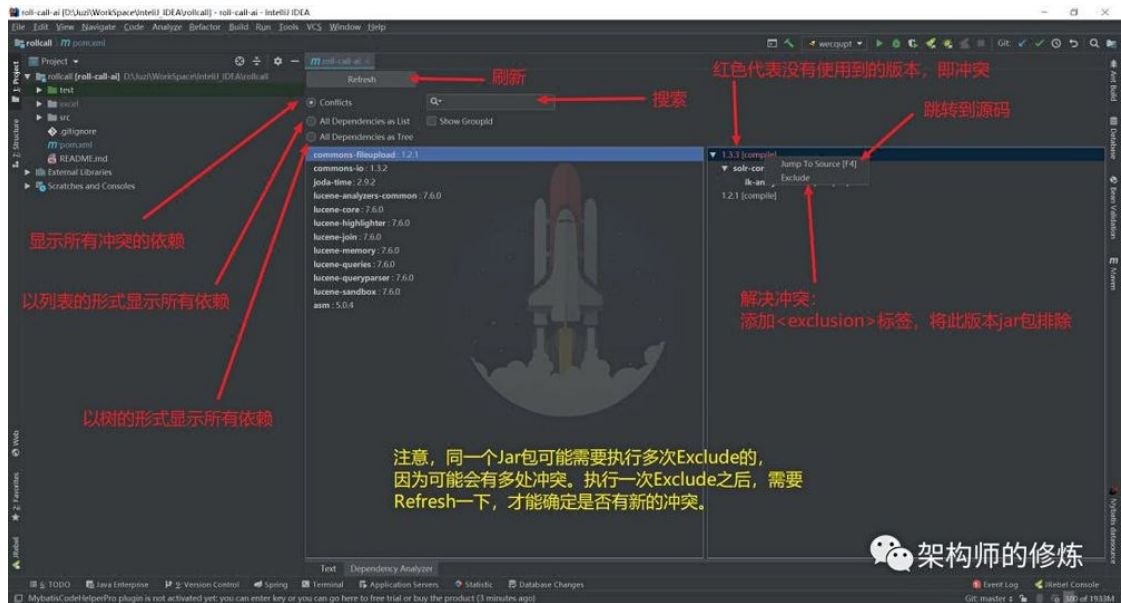
而要介绍的这个 Maven Helper 插件就可以解决这个问题。

怎么用呢？

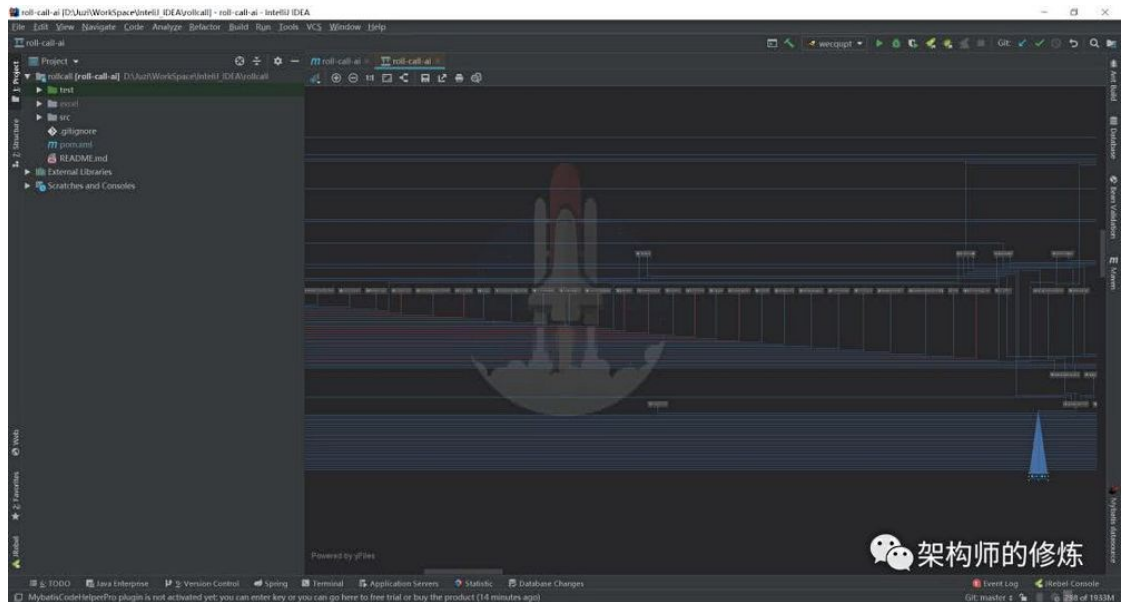
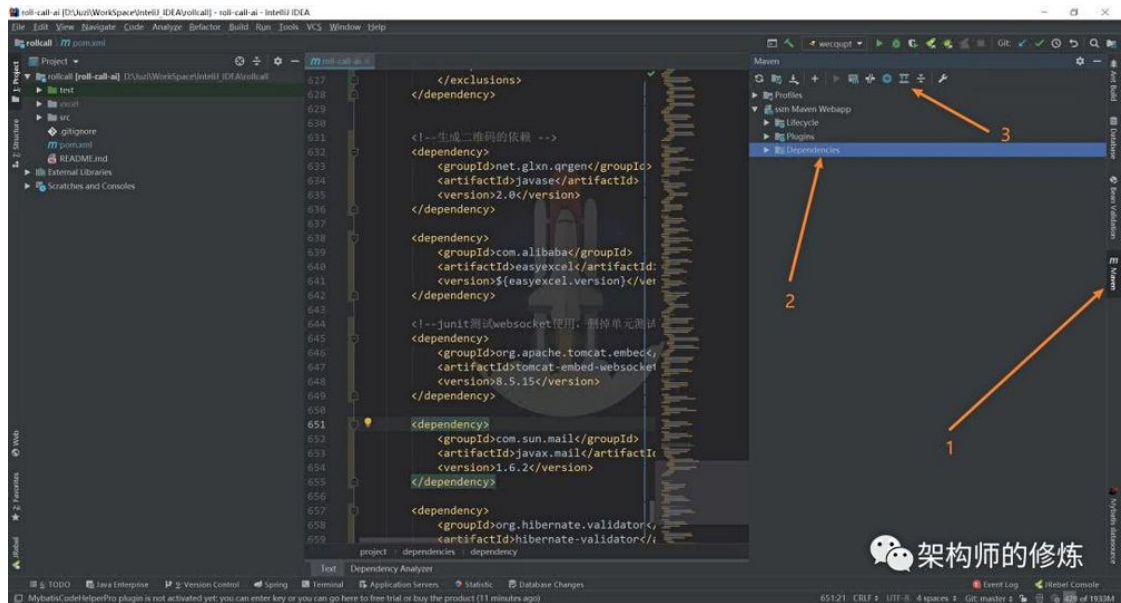
安装之后打开 pom.xml 文件，底部有个 Dependency Analyzer 选项，点击



下面如何解决冲突（此图片来源网络，这哥们标记的非常好，我就不重造轮子了）



还有一个小技巧可看整体项目 Maven 依赖结构图：



记住：图中看到有一些红色的实线，代表依赖冲突，蓝色实线代表正常的依赖

## Lombok Plugin 插件

使用注解自动生成代码，码奴解放者，省去实体类的 set get 注意还要在 Maven 项目的 pom.xml 中添加配置

具体的类上面打上@Data 即可

```
@Data
public class BatchUseProductPlatform {
    private Integer platformId;
}
```

 架构师的修炼

## JRebel for IntelliJ 热部署插件

JRebel 是一个提升生产力的工具，相信不少程序员改完代码后，最头疼的是不能直接看到效果，每次构建，重启以及重新部署的循环操作，时间无形中浪费掉了，有个这个插件，所改即所见，让你编码过程更加流畅。