**http://www.cnblogs.com/yjmyzz/p/intellij-idea-13-getting-started.html**

[**IntelliJ IDEA 13试用手记(附详细截图)**](http://www.cnblogs.com/yjmyzz/p/intellij-idea-13-getting-started.html)

从去年开始转java以来，一直在寻找一款趁手的兵器，eclipse虽然是很多java程序员的首选，但是我发现一旦安装了一些插件，workspace中的项目达到数10个以后，经常崩溃，实在影响编程的心情．

今天试用了近年获得众多好评的IntelliJ IDEA 13，感觉焕然一新，记录如下：

**一、下载安装**

<http://www.jetbrains.com/idea/download/> 这是官网的下载地址，支持mac/windows/linux三大主流平台，我今天试用的是mac版本

**说明：**IntelliJ IDEA是商业软件，有二个版本，

Ultimate Edition是商业版本，功能完善，可以免费试用30天，建议大家下载该版本学习；

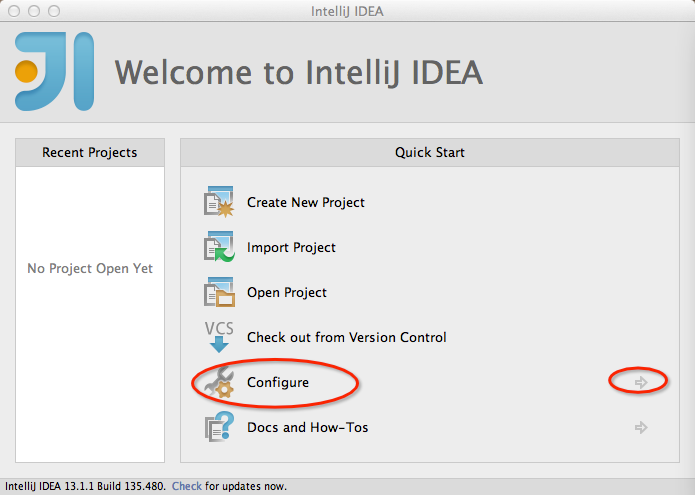
Community Edition　是社区免费版本，功能有所精减，可用于google android开发

**另：**mac上的IntelliJ IDEA 13不支持jdk1.7，如果您在mac上安装了oracle jdk1.7，请先恢复到apple公司自带的jdk 1.6([恢复方法可参考这里](http://www.cnblogs.com/yjmyzz/p/3647516.html))

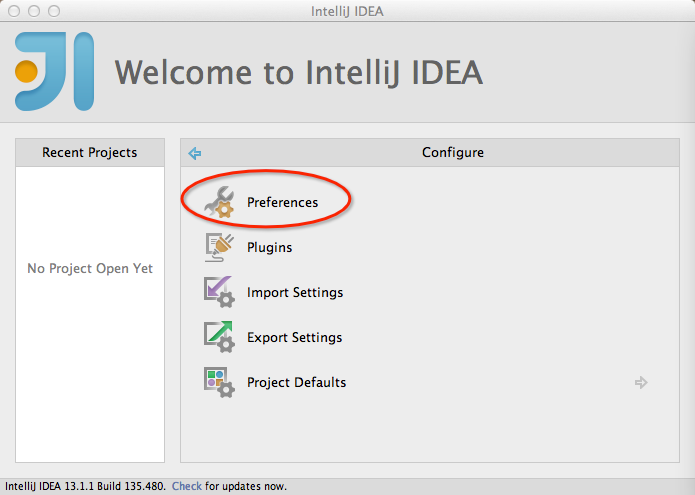
**二 、偏好设置**

2.1 皮肤/字体设置

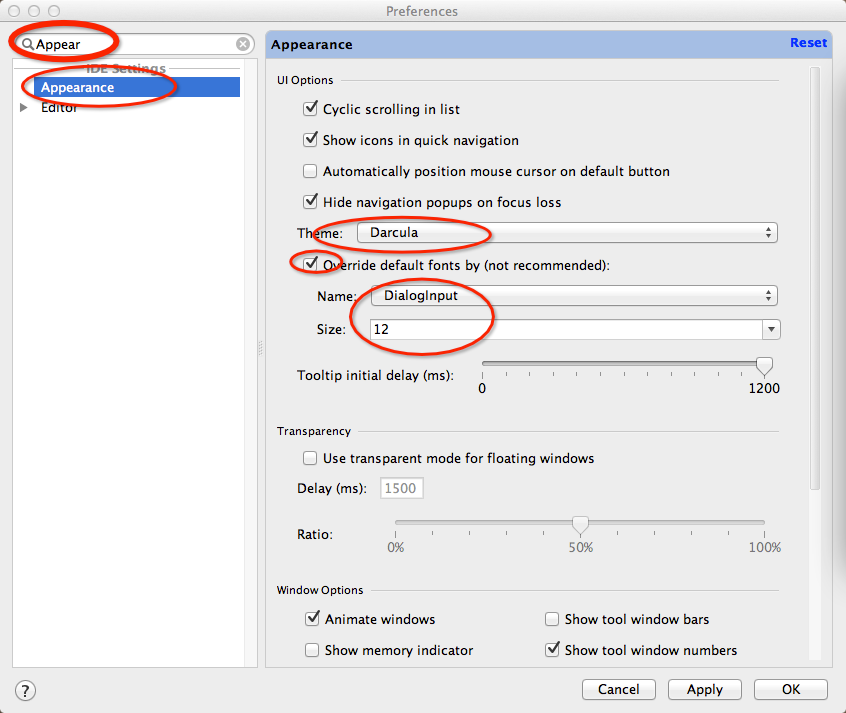
现在越来越多的IDE喜欢黑色风格的配色了，IntelliJ IDEA也赶上了这一潮流，首次启用IDEA后，会看到下面的界面：



点击Configure旁边的"向右"箭头，选择＂Preferences＂(参考下图)



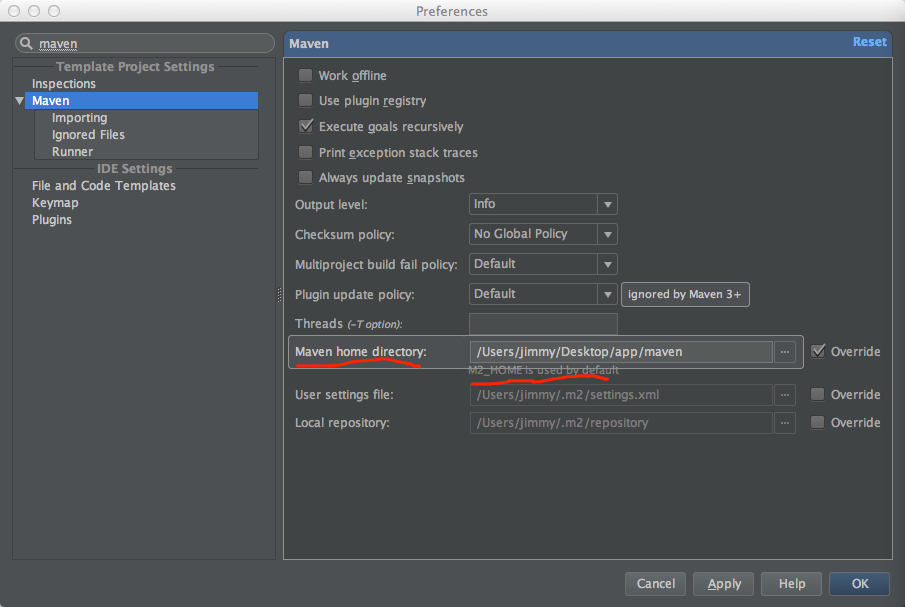
在最左上角的搜索框里，输入＂Appear＂(IntelliJ IDEA很智能，马上会匹配出Appearance)，参考下图：



将Theme切换成＂Darcula＂，同时为了避免中文乱码，参考上图，把默认字体调整为＂DialogInput ＂，12号大小，点击ＯＫ结束

2.2 maven设置

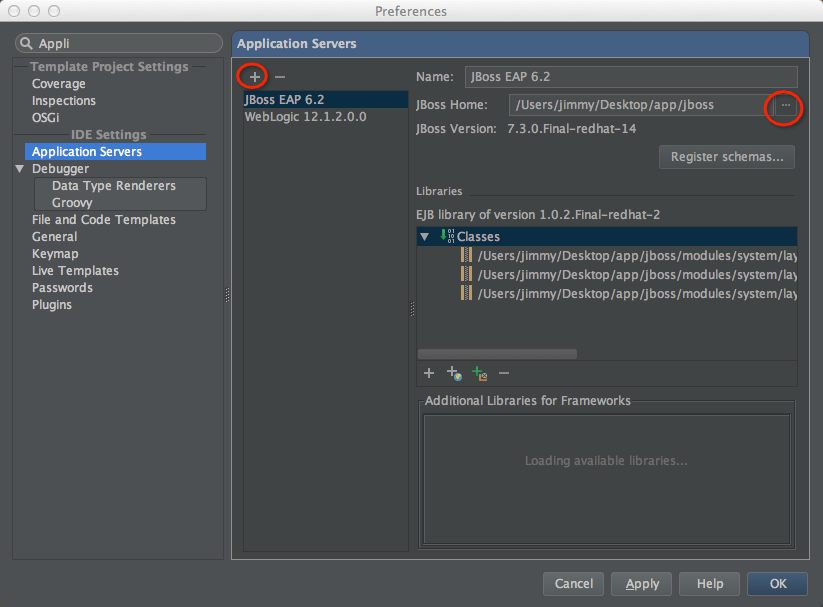
设置黑色风格皮肤后，IntelliJ IDEA会重启，同样还是在偏好设置里，找到maven的设置，参考下图，设置Maven home directory

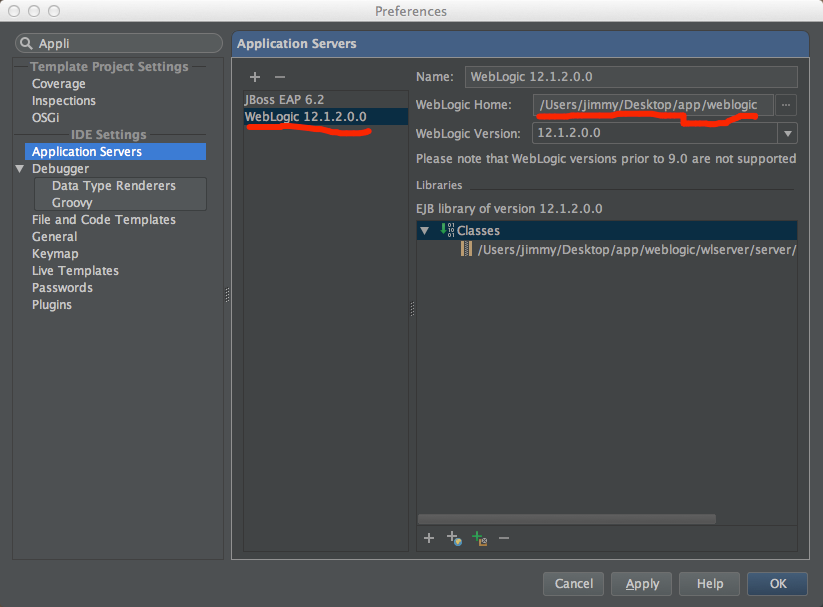


**注：**如果mac上设置了M2\_HOME的环境变量，IntelliJ IDEA默认会以M2\_HOME环境变量的值作为Maven的主目录，如果未设置Ｍ2\_HOME环境变量，就需要在上图的界面中，勾选Override，选择本机Maven的路径

2.3 application server设置

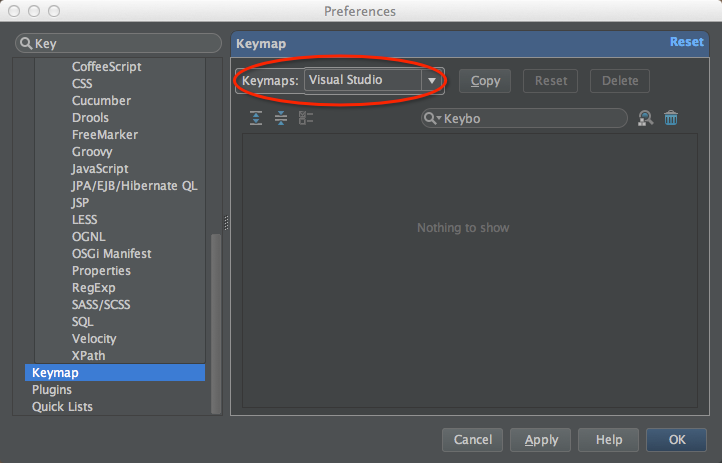
IntelliJ IDEA 已经内置了当下几乎所有主流Application Server的支持插件，下图是添加Jboss EAP 6.2**、**Weblogic 12.1.2.0.0后的界面





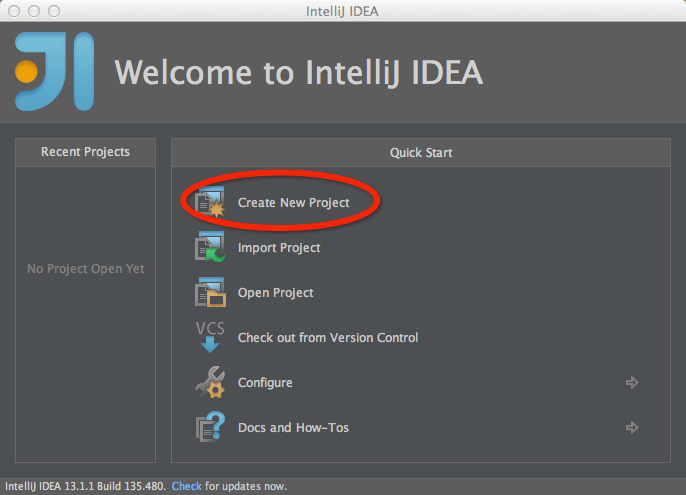
2.4 Keymap(快捷键)设置

在键盘方案中，居然发现有Visual Studio的方案选择，.NET背景的程序员有福了:)

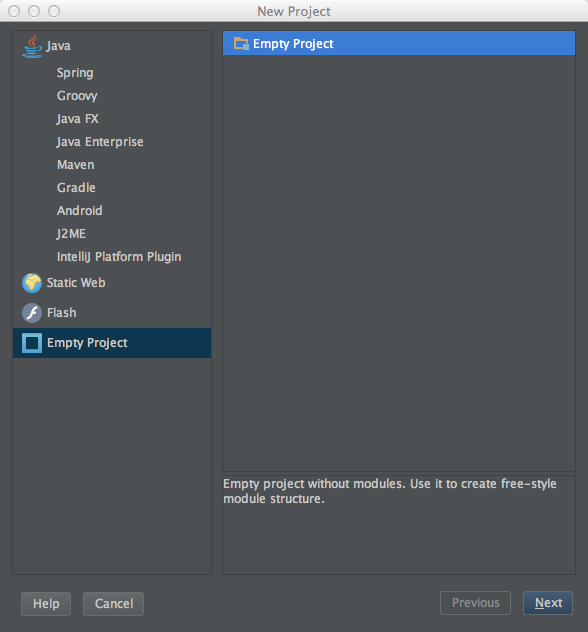


**三、创建Project**

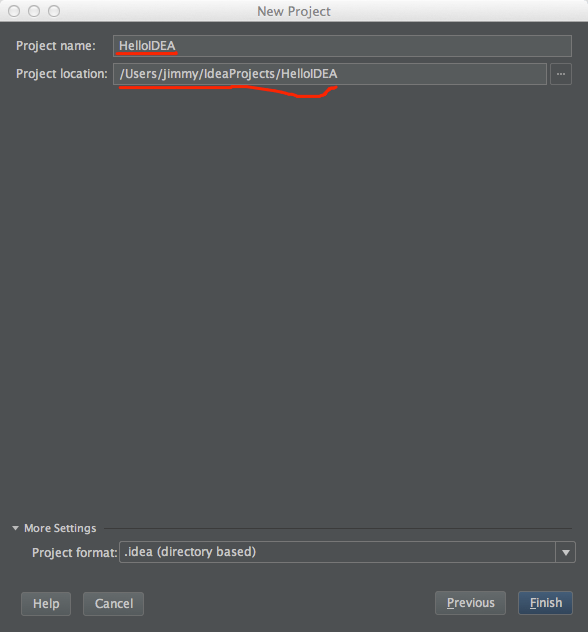
先参考下图，创建一个新的Project



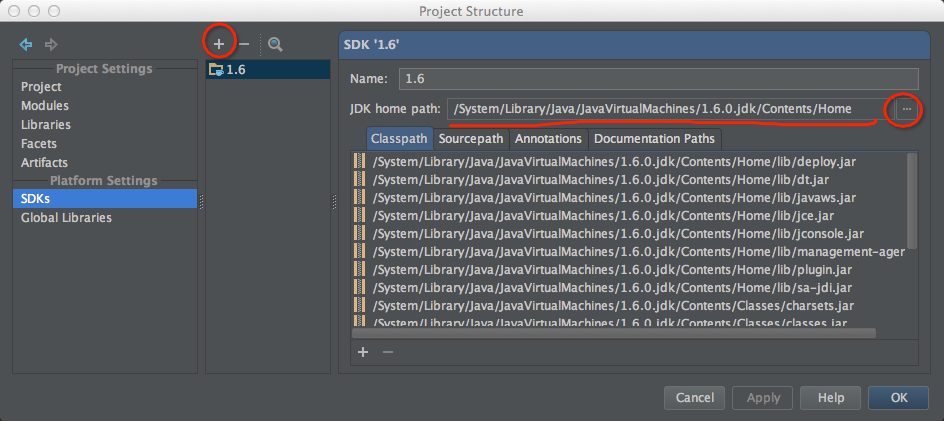
**注：**IntelliJ IDEA中的Project，并不是真正的project，它其实跟eclipse中的workspace**、**visual studio中的solution是一个概念



这里我们选择一个Empty Project，即：相当于eclipse中创建了一个空工作区 或 visual studio中创建一个空白解决方案，然后在下面的界面中输入Project Name



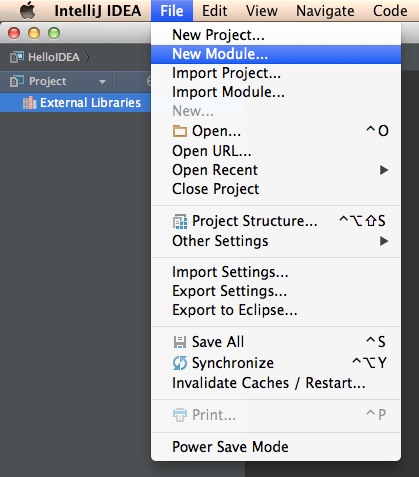
点击Finish后，因为现在只是一个空壳，里面没有任何东西，所以会弹出一个对话框，让我们添加Module，这里我们先不忙添加模块，先来设置一下Project的JDK，参考下图我们设置整个Project的JDK版本为1.6



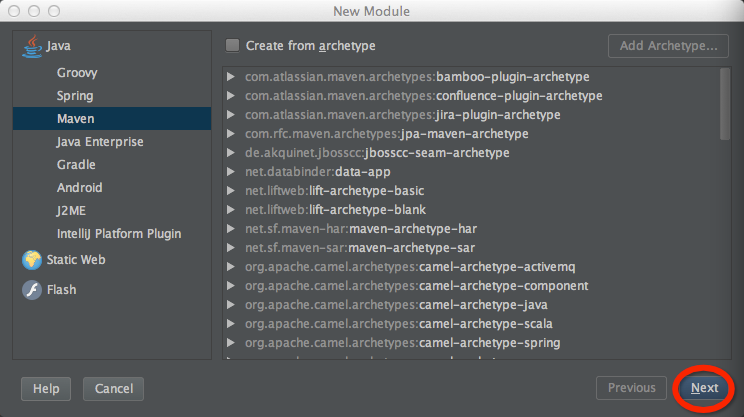
到此，完成Project的创建

**四、添加Module**

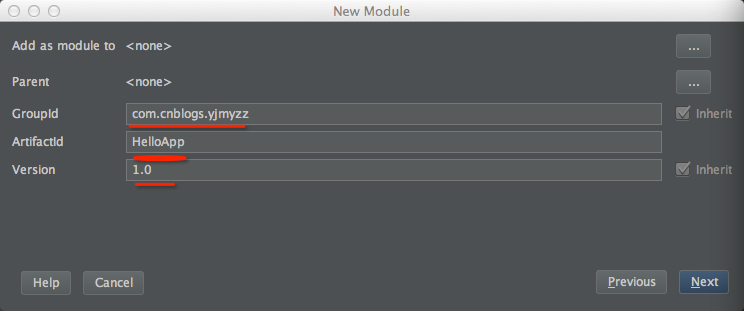
Module才是eclipse/visual studio中真正包含代码的"项目"， File -> New Module添加一个模块

****

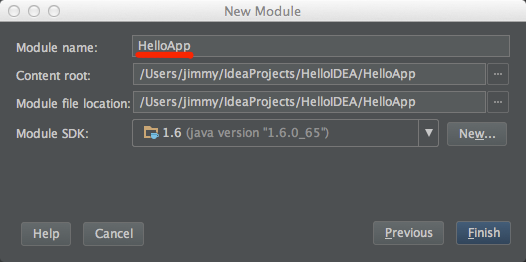
 我们选择Maven模块，参考下图，直接next，即不使用任何archetype，仅创建一个空的maven项目



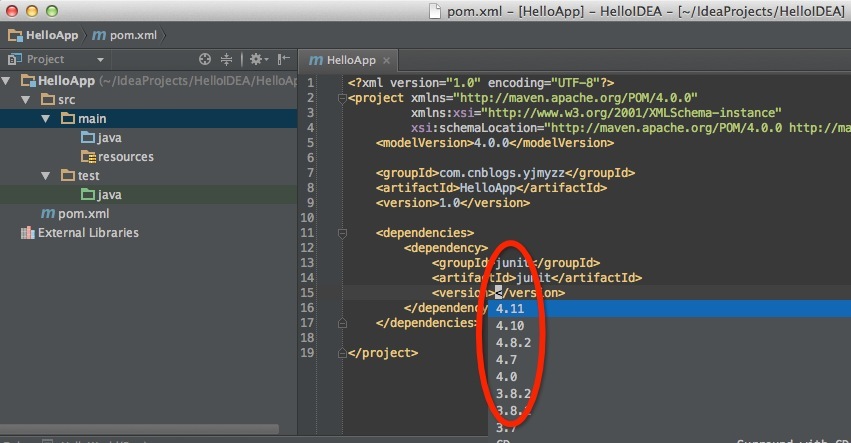
接下来，输入maven坐标3要素(GroupId，ArtifactId，Version)，继续Next



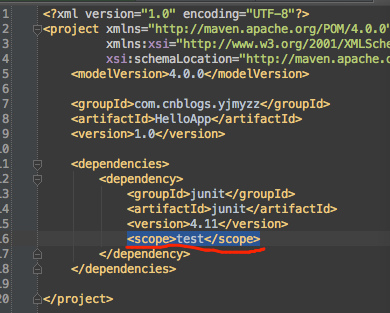
接着输入Module name，我个人觉得这里需要改进，最好Module name默认就是刚才输入的ArtifactId，这样可以减少用户输入，最后点击Finish



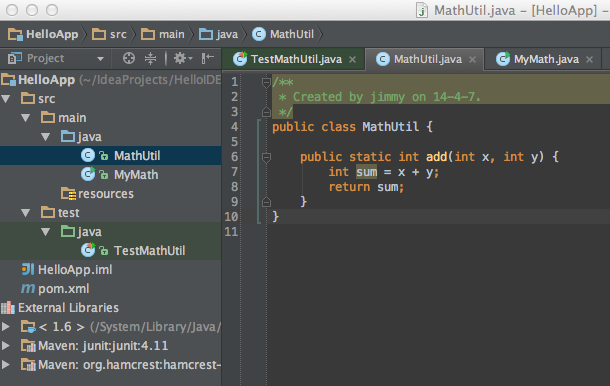
参看下图，可以发现左侧的Project目录树里，已经显示了文件夹的结构，双击pom.xml打开，我们添加一个junit的依赖项，intelliJ IDEA的智能提示功能非常强，连依赖项的版本号，都能智能感知出来



注:如果是添加Junit的依赖项，记得要加<scope>test</scope>，即:

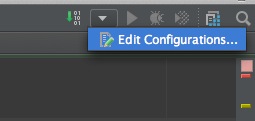


随便写几个类试试

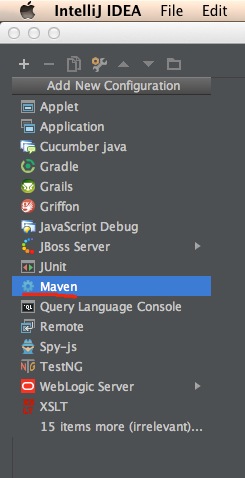


**五 、运行及调试**

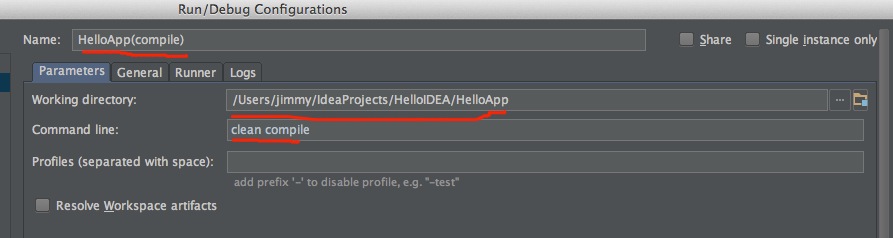
参考下图， 注意左上角有一个向下的小三角，如下图，点击它，选择"Edit Configurations..."



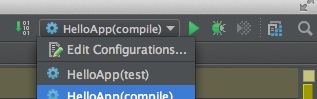
参考下图，点击右上角的+号，添加一个maven配置



接下来的界面跟eclipse就很相似了，添加maven执行的命令行，参考下图，并在name栏里填入"比较友好"的名称

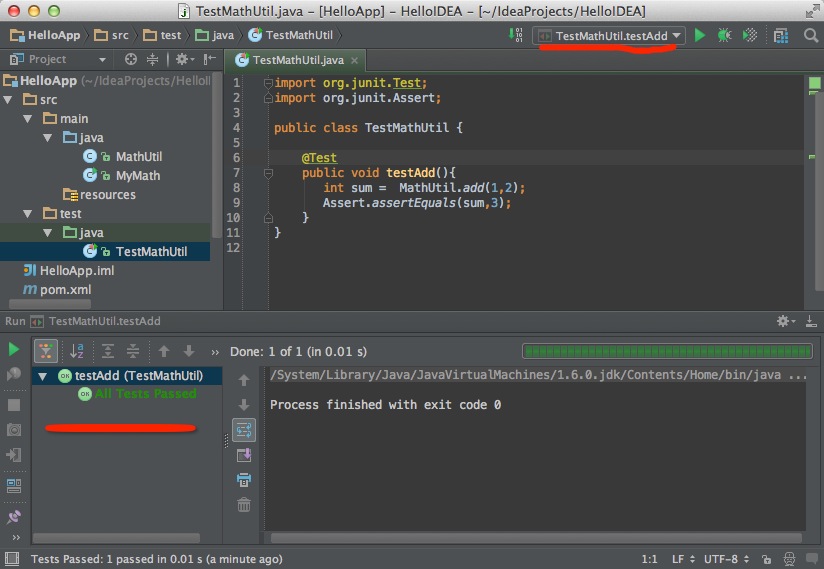


当然，你可以添加多个配置，比如HelloApp(compile)表示编译，HelloApp(test)表示测试， 保存配置后，回到idea主界面，发现下拉框里，就有东西了



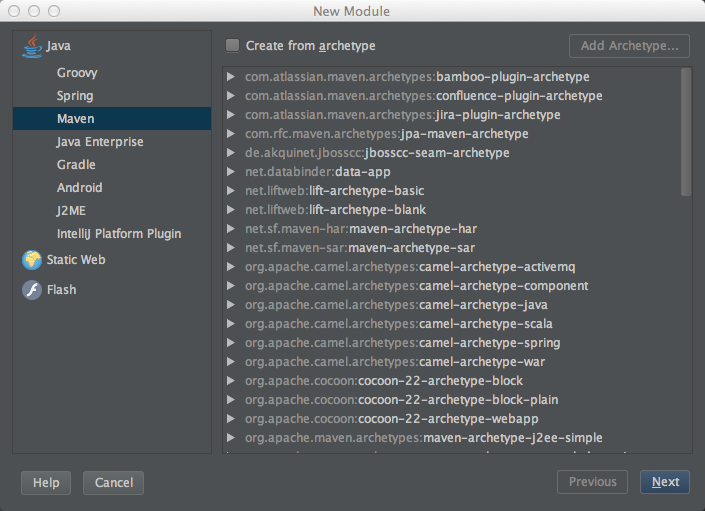
下拉框右侧的三角按钮，表示运行(Run)，而三角右侧的"发光的太阳"按钮，表示调试(Debug)，点击它们，就能"运行"或"调试"

另外:对于单元测试，除了通过配置执行 mvn clean test外，还可以有专门的junit配置，再刚才点击+添加配置时，可以添加Junit配置，这个大家试试就知道了，不过，我不建议这么弄，因为有一个更简单的方法，如下图，先打开有@Test的类文件，然后把光标停在要测试的方法，比如testAdd上，直接按shift+control+F10，就能让IDE自动帮我们生成junit的配置，并马上运行该配置

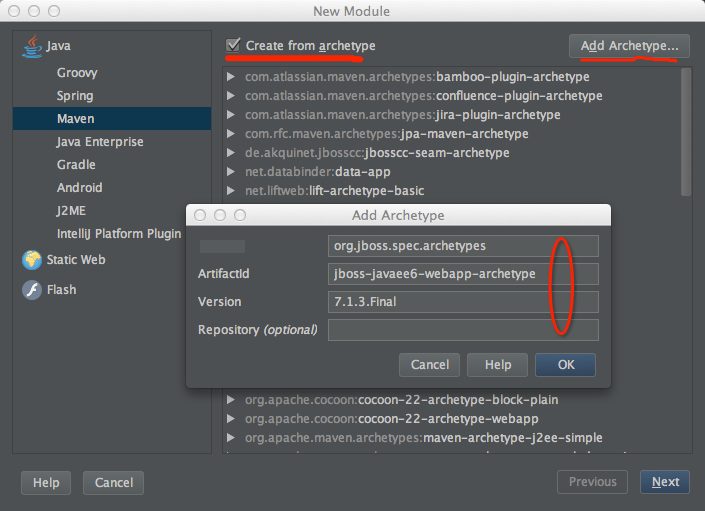


**六 、 添加Web模块**

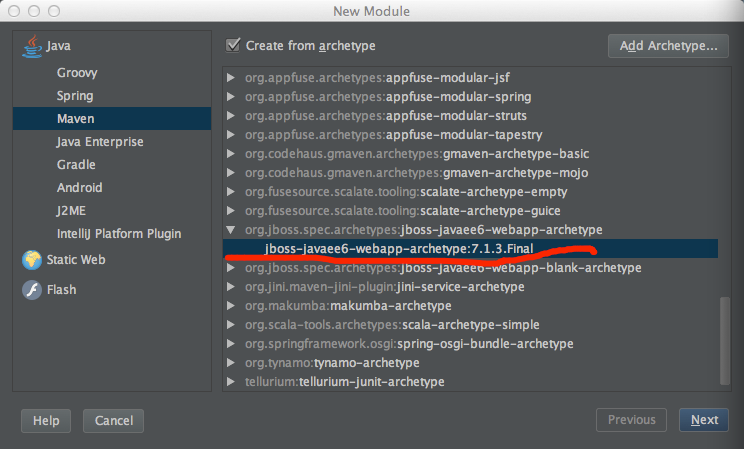
仍然还是在下面这个添加Module的界面，IDEA默认自带了一些Archetype，但是数量有限，比如jboss提供的Archetype里面一个都没有



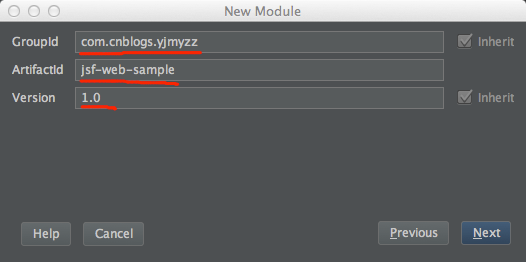
勾选Create from archetype，然后点击Add Archetype，手动添加一个jboss的示例webapp archetype



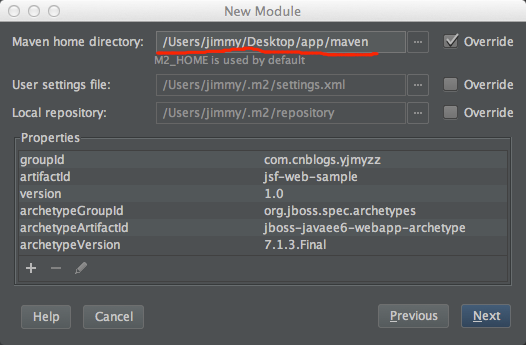
点击ok后，就有了，选择它



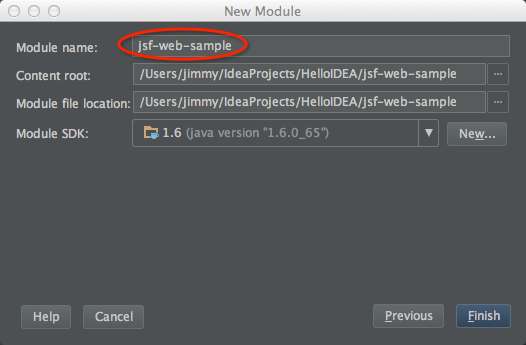
输入maven项目的坐标3要素



在接下来的界面中，设置maven路径(一般会自动识别，无需个性)，然后next

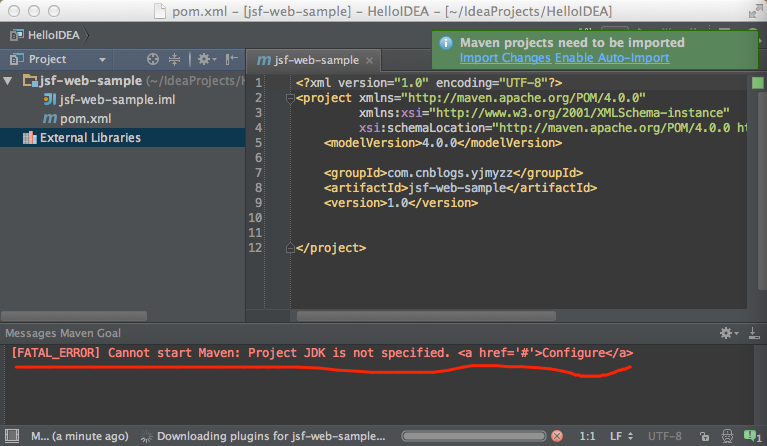


接下来输入module的名称，然后点击Finish

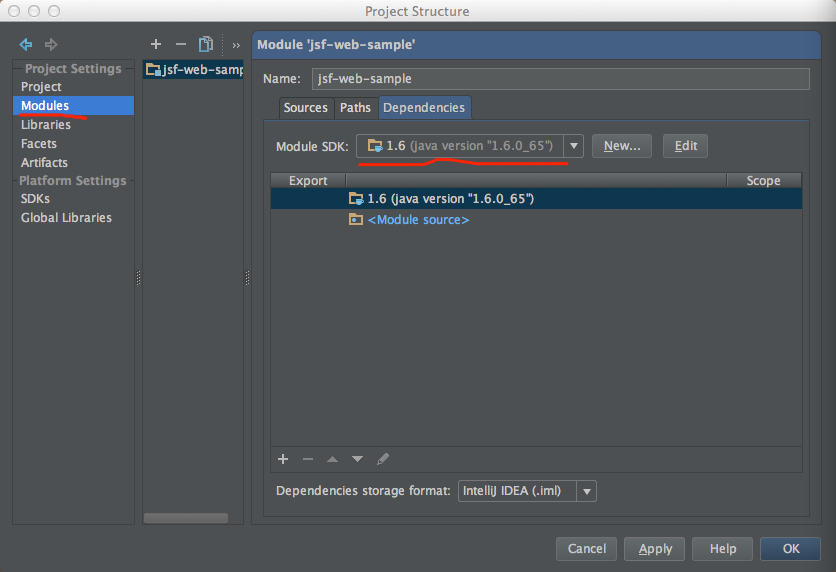


点击完成，在接下面的界面中，会发现一个大大的错误，说明IDEA并没有识别出刚才我们设置的JDK(不知道算不算IDEA的一个bug)

注: 经实际多次测试发现，如果在刚才上一步的界面中，Module SDK这里，哪怕系统已经默认选中了JDK 1.6，人工点击旁边的小三角，再手动选择一下，最后点击Finish，就不会有这个错误了.

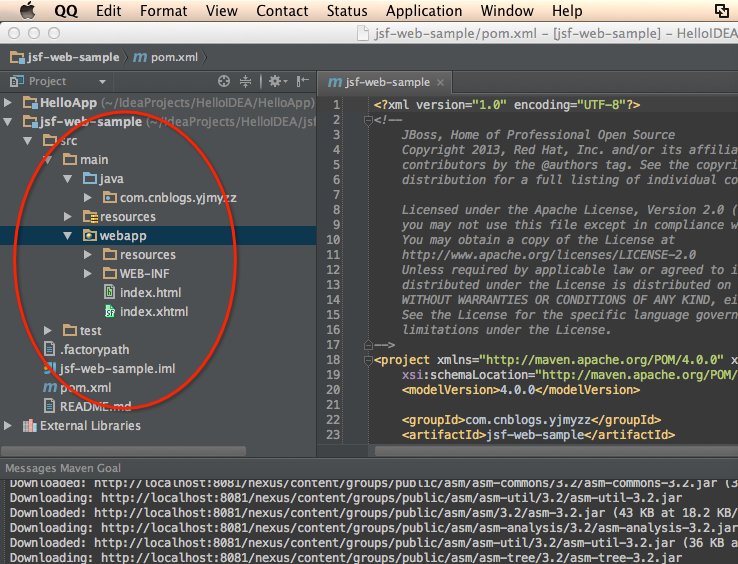


在jsf-web-sample这个model上右击 -> Open Module Setting，打开module的设置，在Module SDK这里再手动选择一下JDK版本

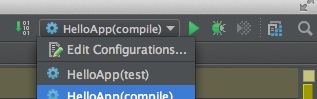


点击OK后，刚才的错误就消除了，但是这时idea不会自动创建maven的目录结构，所以建议还是把module给delete掉，重新再来一次，注意这次一定记得要手动选择一下module sdk，再点击finish!

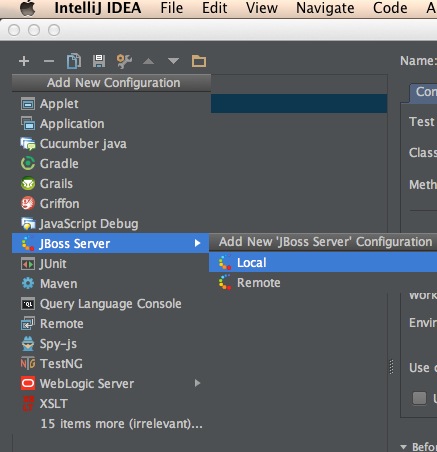
正常的话，idea会自动帮我们创建好目录结果，以及一些测试文件



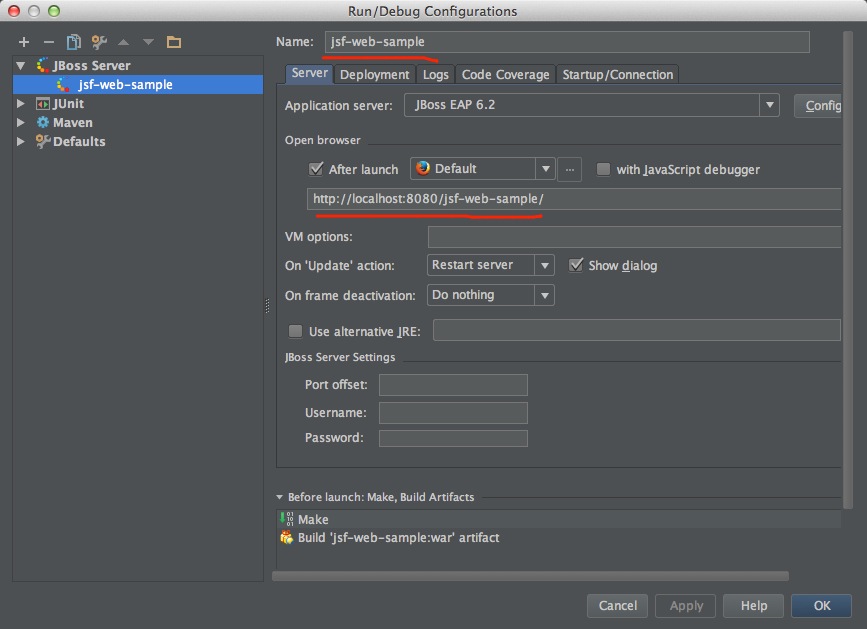
接下来，看看如何部署到jboss上，然后在Edit Configurations...这里



点击+号，添加配置，但是这次选择Jboss

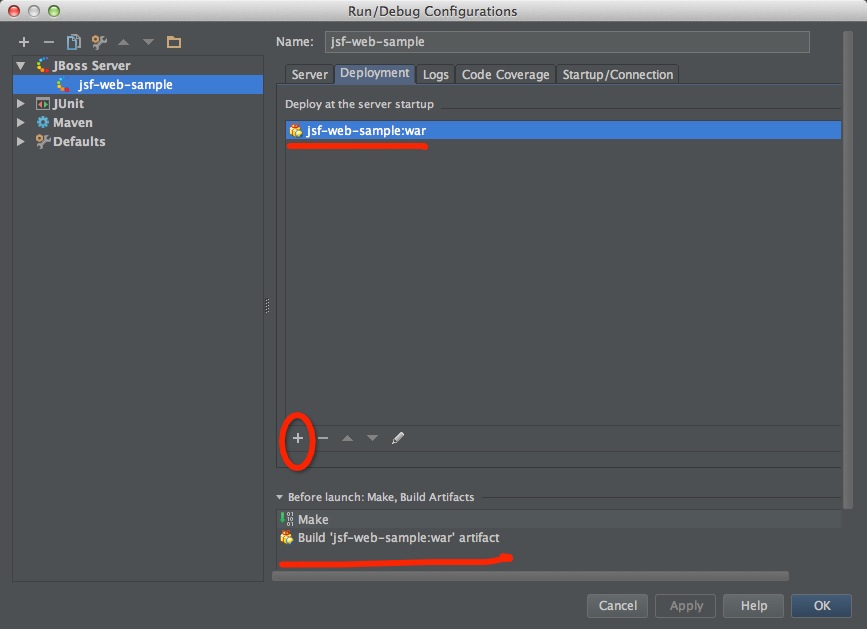


在server tab页中，填写Name，以及部署成功后，要打开的浏览器地址

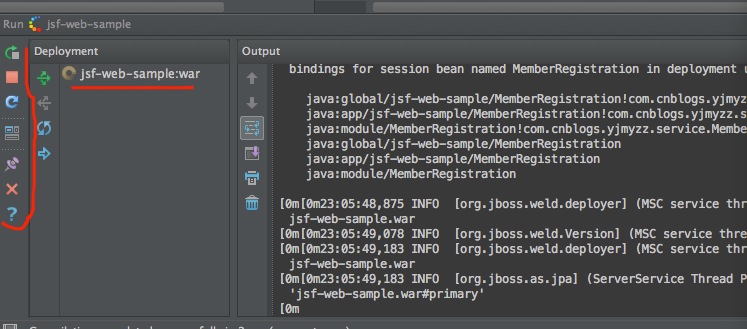


在Deployment tab页，参考下图，点击+号，添加要部署的Artifact

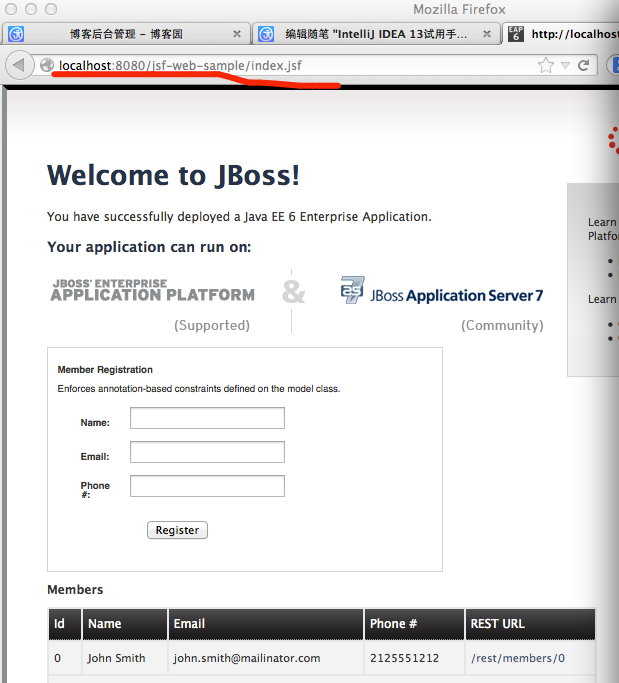
注: 部署在jboss上时，部署的artifact名，一定要以:war结尾，大家不要修改，否则会报 invalid extension之类的错误



点击ok，回到idea主界面后，会发现Application Server面板里，已经有这一项配置了



左侧有一排竖着的小图标按钮，第一个就是启动，点击它，就能启用jboss并部署jsf-web-sample这个应用，顺利的话，就能直接在浏览器里打开了



当然，idea的功能远不止这些，大家入门之后，自行体会吧.