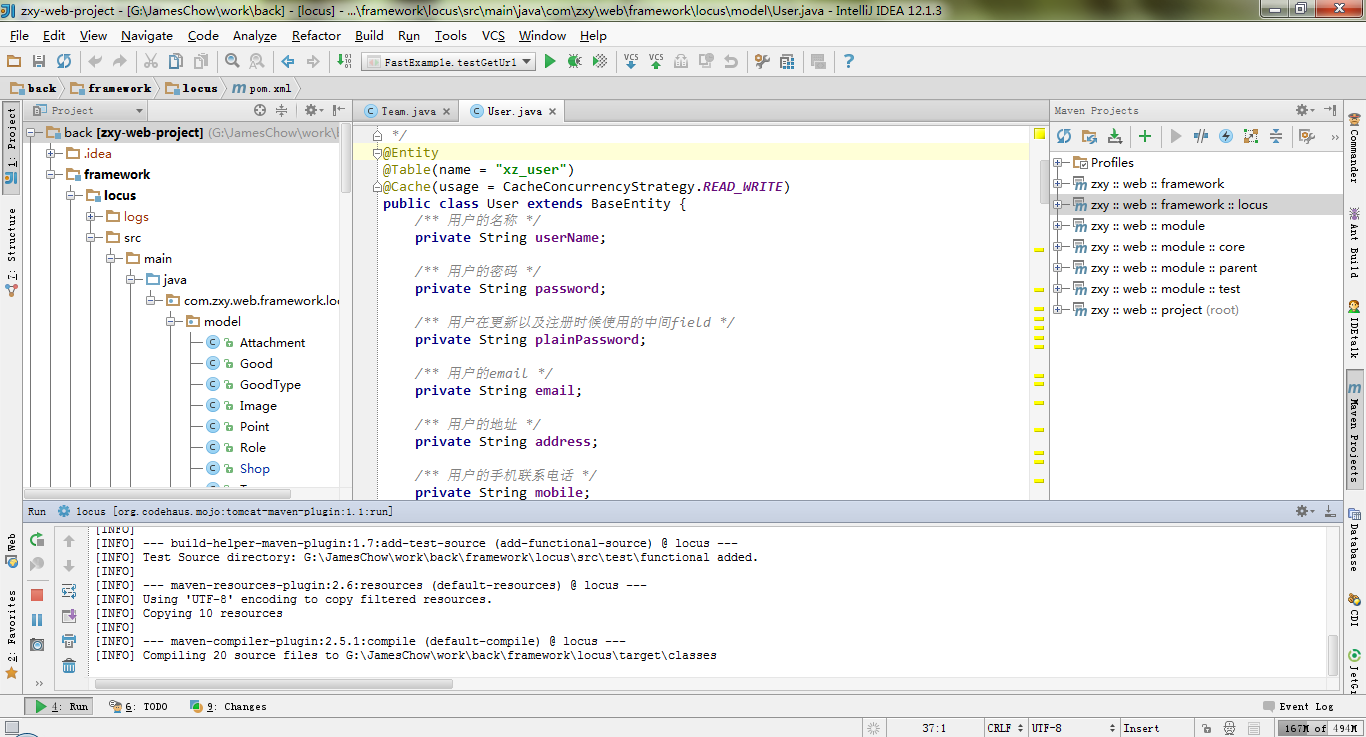
**Intellij Idea 使用文档**

* **James Chow Created**

1. **总体介绍**

Intellij 本身是基于纯JAVA语言实现编写的IDE工具，他有许多的版本，WebStorm、PhpStorm、RubyStorm这三个各有侧重点，WebStorm:注重前端的开发，如果开发Node.js、html5、javascript可以使用WebStorm，他的JS提示功能相当强大。编写Node.js也非常的爽。同理PhpStorm RubyStorm偏重的就是php ruby。回到JAVA版本的Intellij，他集成了许多常用的插件，这是Eclipse无法比拟的，虽然Eclipse开源社区很活跃，有很多插件。但是效率使用方便性上明显不如Intellij。Intellij集成了Android, Maven, GitHub,Gradle等等。

1. **界面介绍**



Intellij是分为5个部分的，图上是显示4块（算上顶部的工具栏是五块区域）。

区域1:代码区，编写代码在这块区域下

区域2:文件导航区，查看文件。项目的目录结构在这个区域，而显示在这个区域的菜单主要有Project、Structure（显示一个类的结构视图）

区域3:为一些插件视图界面，比如我现在打开的maven视图可以方便的使用maven命令对项目进行管理操作。右侧的视图主要有maven、ant、gradle、database等，常用的就是ant、maven、gradle等。

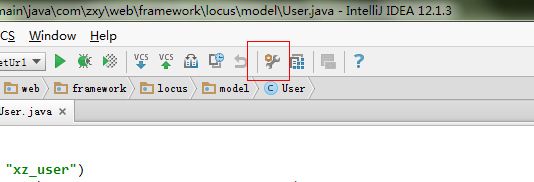
区域4:为run、debug、svn、git等窗口（相当重要）

重要的区域:1、2、4 第三块区域不是每一个项目都会用到，一般的项目基本不会用到右边侧的区域。

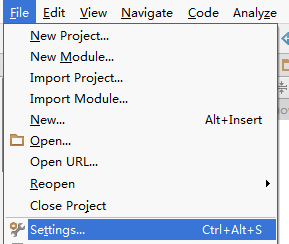
1. **基本设置**
2. **进入设置**

对Intellij的设置全部都在Settings中.进入settings总共有如下的几种方式

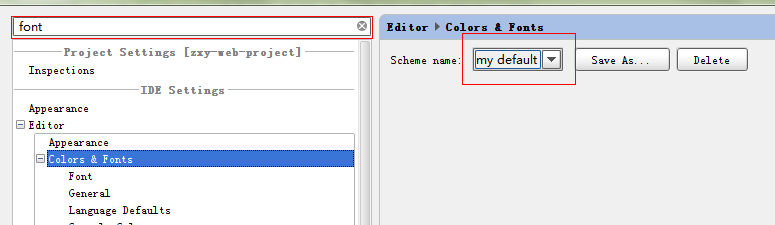
1. 点击工具栏上如图的按钮



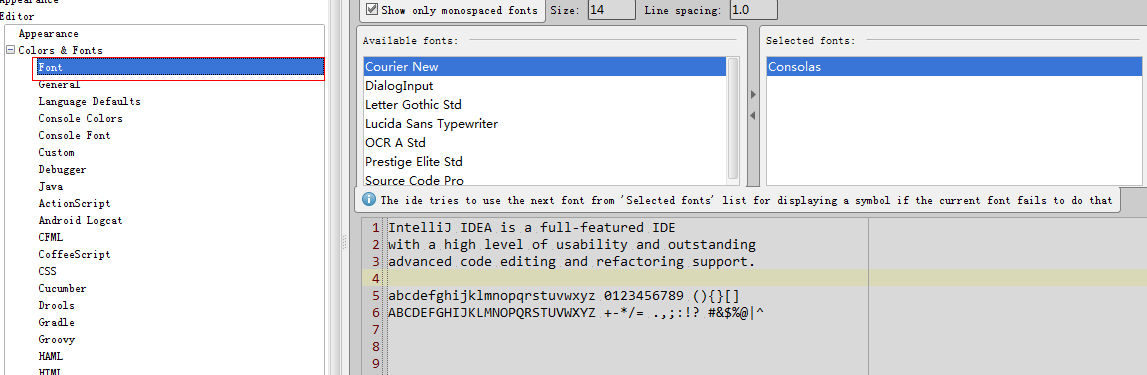
1. File -> Settings这种方式可以进入设置



1. 初次使用Intellij 12设置跟上面的不太一样是点击Configure->Settings（不再截图）。
2. **常用设置**
3. **字体设置**



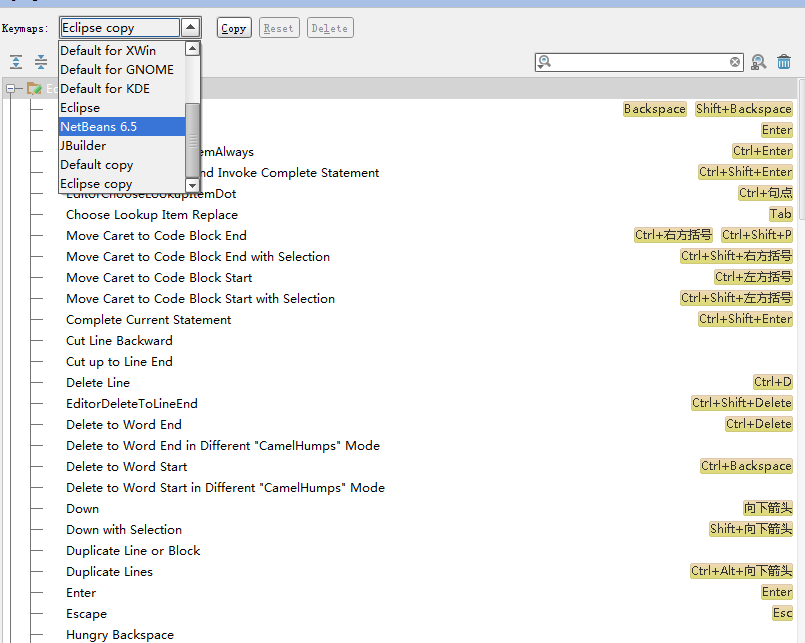
选择默认的字体，save As保存成自已喜欢的名字



选择Font左侧是可供选择的字体，右侧是你想用的字体。设置很简单（0智商的估计都可以完成这样的操作）标注一下：intellij 11系列跟 intellij 12系列的差别是有不少的，Intellij 12界面架构跟11是不一样的，intellij 12支持jdk8等一系列的新特性。当然在某些设置方面是不一样的，intellij 11的字段设置跟 12稍不同，不过我相信做程序员（我老周的朋友肯定都会设置的）。

1. **快捷键设置**

搜索keymap 进行快捷键设置界面



如果你使用eclipse的话就Copy一份Eclipse的快捷键。改名比如我这里改名为Eclipse Copy

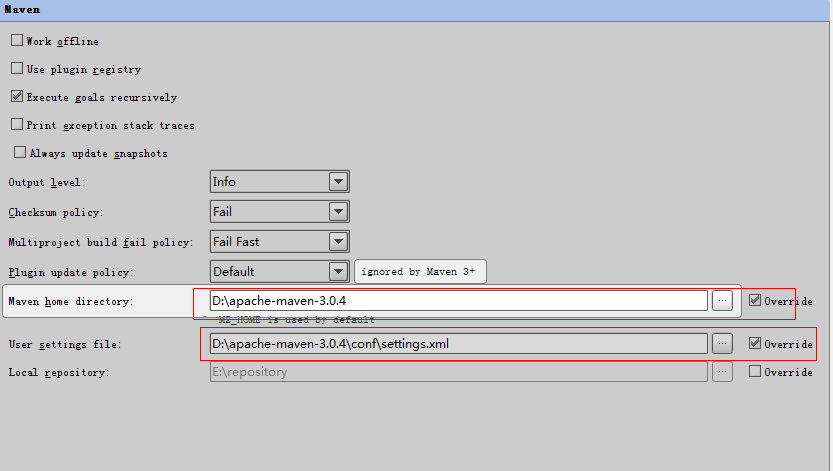


Main menu -> Code -> Completion

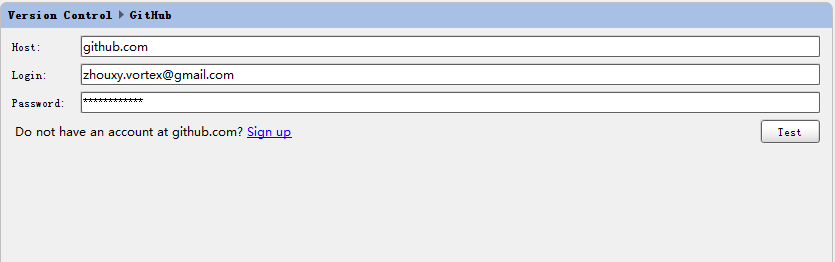
将Basic 和 Cyclic Expand Word改成Eclipse默认的。

标注:intellij中Basic 和 SmartType两者提示是不一样的。Basic是基本的（Alt + / 那个快捷键）只是提示变量名啊，已经引入的类名啊。 这里并不像Eclipse一样把所有的类都提示出来，这样会造成效率的浪费，Alt+Shift+空格 这是Intellij里面特有的一个快捷键这个快捷键是提示现在所有类。在Intellij里就算Copy了Eclipse快捷键之后还是有些是不一样的，比如Eclipse里提示Ctrl+1这个键被替代成了Alt+Enter，生成Get Set方法那个快捷键在Eclipse中是Alt+Shift+S 在Intellij中是Alt+Insert

1. **Maven GitHub设置**



Maven没有什么好说的，关于Maven的知识请自行到官网上去学习。这里我不再说了，这里只说设置，搜索Maven进入界面去设置



GitHub的设置就是搜索Github进去输入你所注册的github账号以及密码



如果使用Git作为项目管理的工具，就要去下载安装Git工具包，然后在 git设置界面进行设置，指定文件的位置。

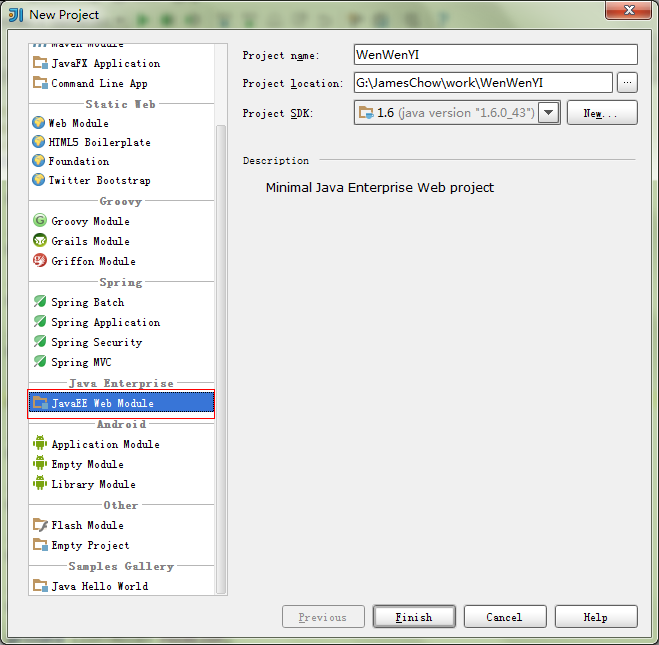
1. **Svn设置等**

这些设置在一般情况下是不动的。因为Intellij给我们集成好了插件（说习惯了插件，其实就是一个IDE的组件而已，所以用起来比Eclipse要爽的多）

1. **项目实践使用技巧**

在介绍项目实际中开发东西的之前，先说明一下，在Intellij中Project可以理解成Eclipse下的WorkSpace概念，而Module可以理解为Eclipse里面的一个Project。特殊情况就是Maven项目中Module是可以互相嵌套，互相依赖的，这正是符合Maven的特点。所以吗老周有话说就是这TM用起来真的很爽。

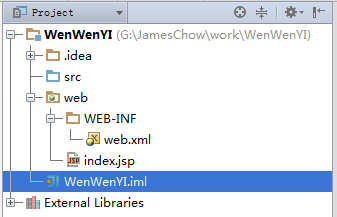
* 1. **普通J2EE项目创建运行等**



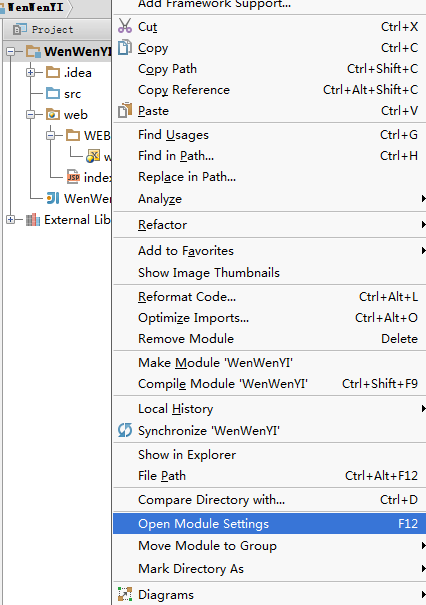
Spring类型的项目可以选择建立，不过Intellij会花时间去官网下载jar包。下次再使用就不会再去下载了，MyEclipse是下载的时候就集成在安装包里面了，所以MyEclipse很大，就是这个原因。



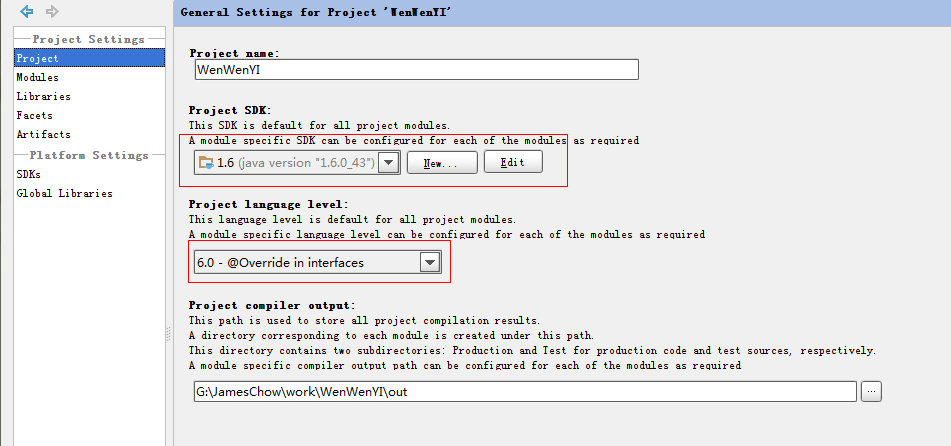
初次使用可能没有指定jdk我们选择new去指定自己jdk所在位置，这是全局性的，不会有任何问题，设置也跟eclipse很简单。



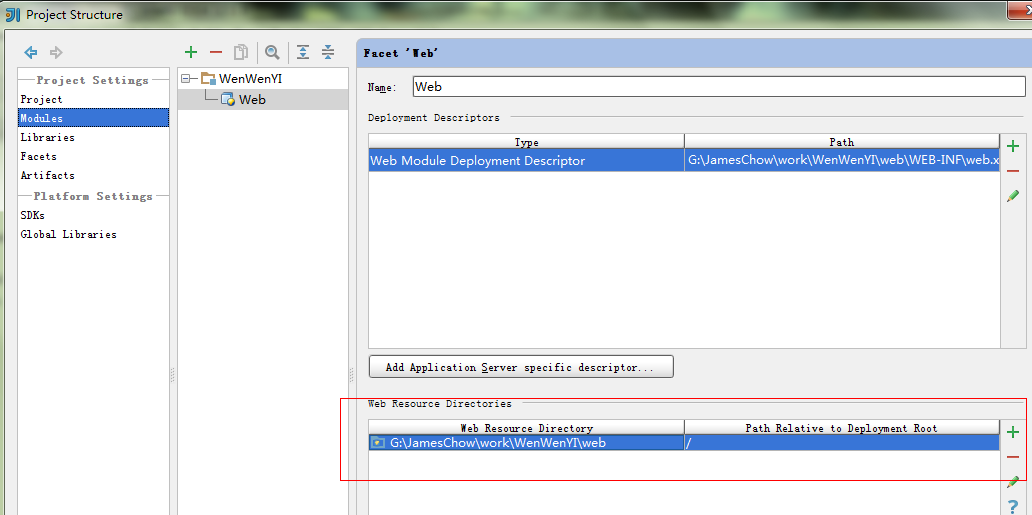
最普通的J2EE项目是这样的目录结构。



要添加JAR包右击项目，选择Open Module Settings，或者选中项目之后按下F12这个键也可以进行Module的设置。

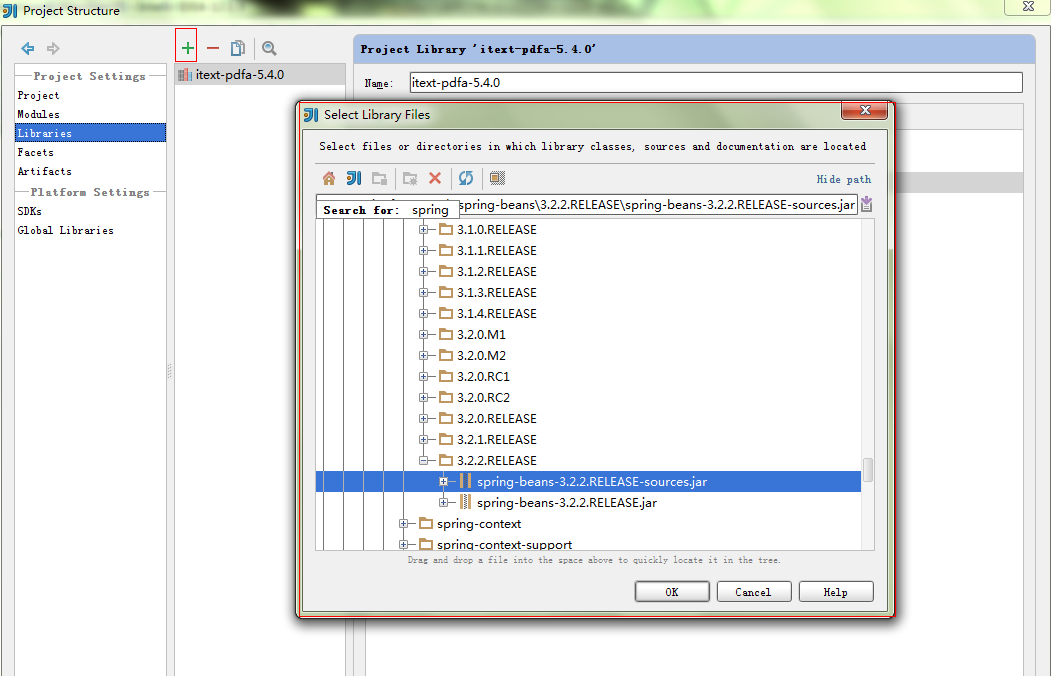


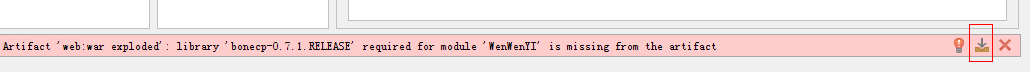
设置界面是这样的，我这里就不再解释什么意思了，无非就是项目名啊，编译级别之类的，可以根据项目的需要进行不同的设置工作。



进入Modules这个设置选项中，主要关注一下Path Relative to Deployment Root这里就是指定访问的URL比如这里叫/JamesChow那么tomcat使用默认的配置就是<http://localhost:8080/JamesChow> Intellij默认的是<http://localhost:8080/> 就行了

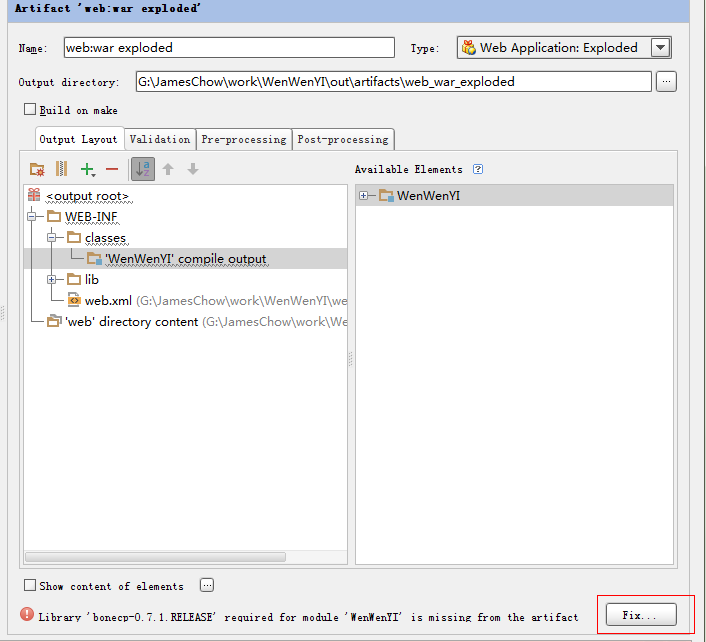
加入JAR包到编译里面的



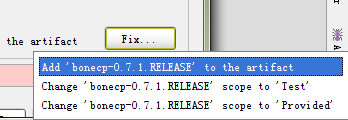


这里的提示就是把jar包放到lib下面，一个JAR包总共有几种存活周期

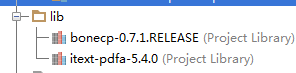
1. production 就是生产环境例如Spring里面的beans这些jar包模块，要随项目一起发布的。
2. test就是比如junit这些测试类，只在测试的时候会使用，它是不需要跟随项目一起发布出去的
3. provided就是例如HttpServlet这种类，在我们编写Servlet的时候需要继承，但是jdk是不带这个接口的，我们需要指定JAR包（Tomcat是在lib目录下的servlet.jar）。但是当项目真正部署的时候我们又不需要这个JAR包一起发布，因为J2EE容器给咱们提供了。



点击那个按钮到达这个项目

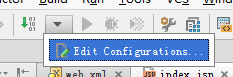


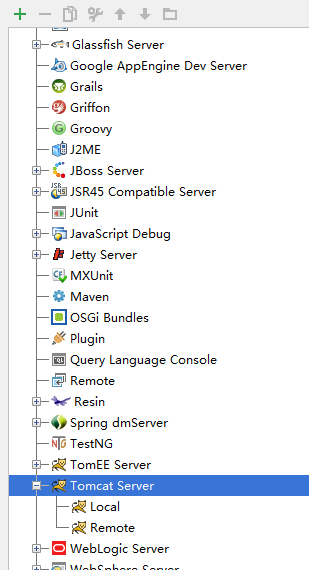
看到这样的提示了吧，我们选择第一个

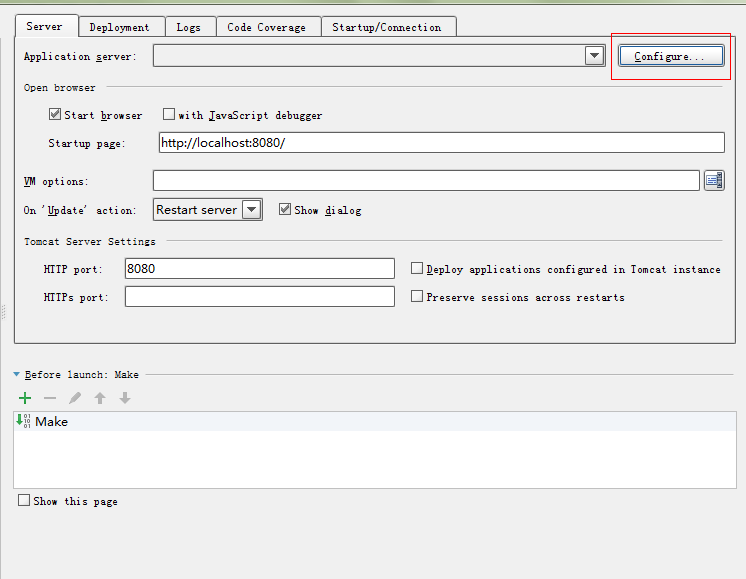


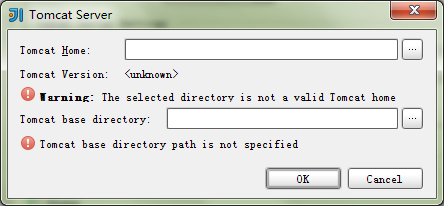
可以看到他给我们加入到了lib这个目录下面

编写完成自己的页面之后，下面就是要把项目跑起来，下面讲解TOMCAT的配置

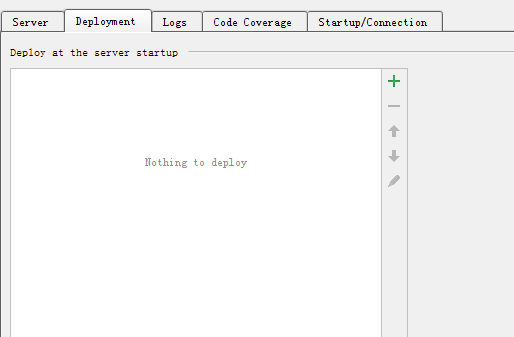


找到tomcat server -> Local

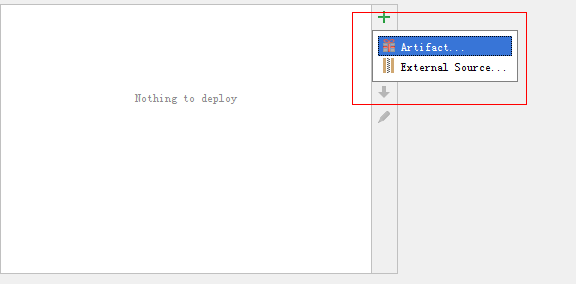


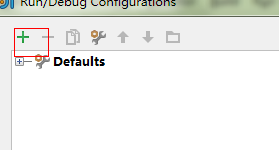
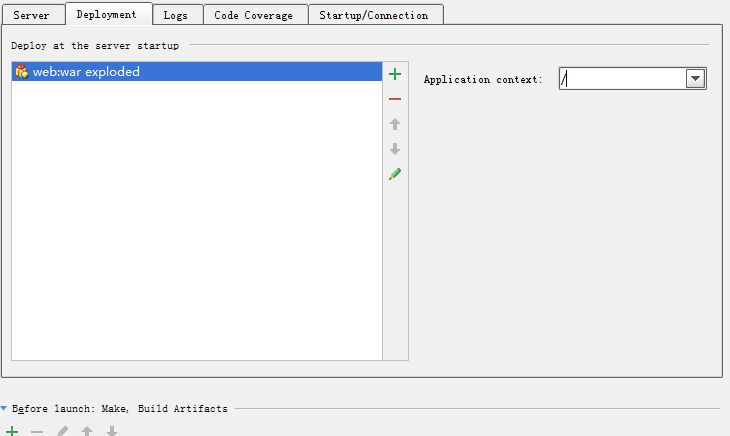


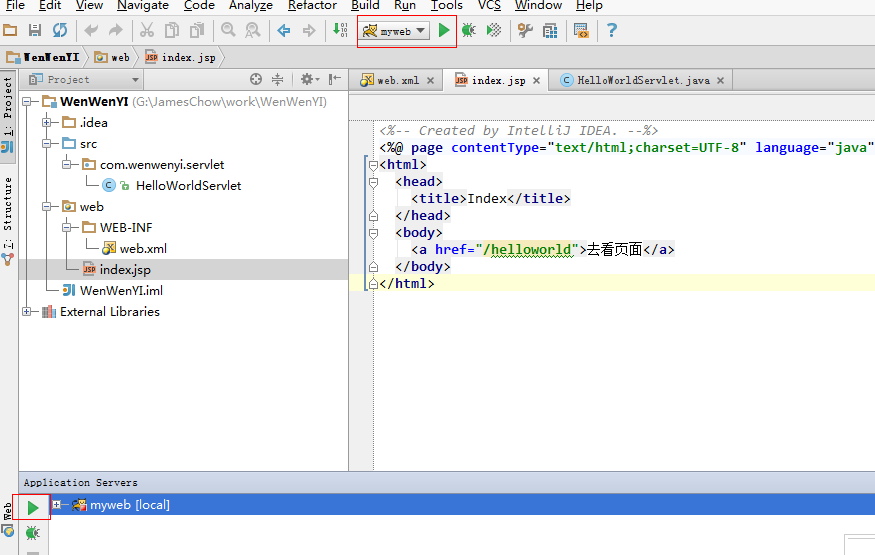
选择自己的Tomcat位置



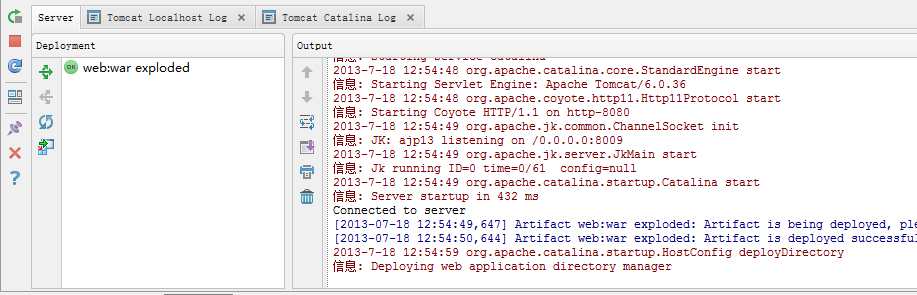
选择Deployment





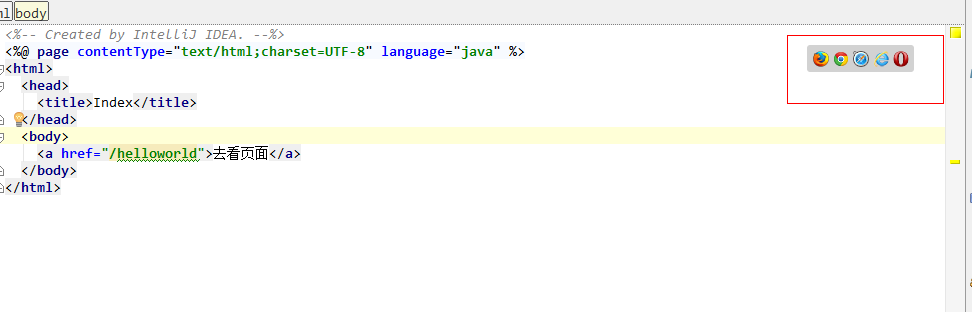


添加完自己刚刚配置的tomcat之后就可以运行了



这个为reploy键就是可以自动重新部署



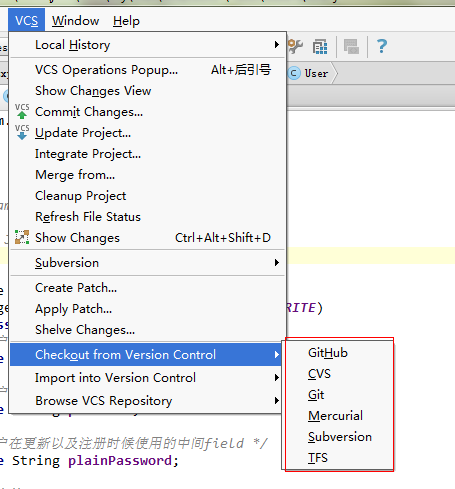


当停留在页面上的时候可以选择不同的浏览器进行查看。方便程序员进行调试。

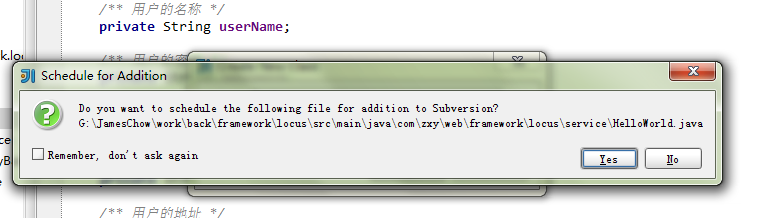
上面就是一个简单的创建到部署的过程。

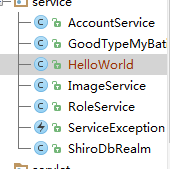
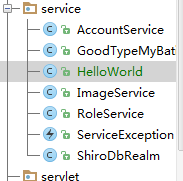
* 1. **充分应用其提示功能**

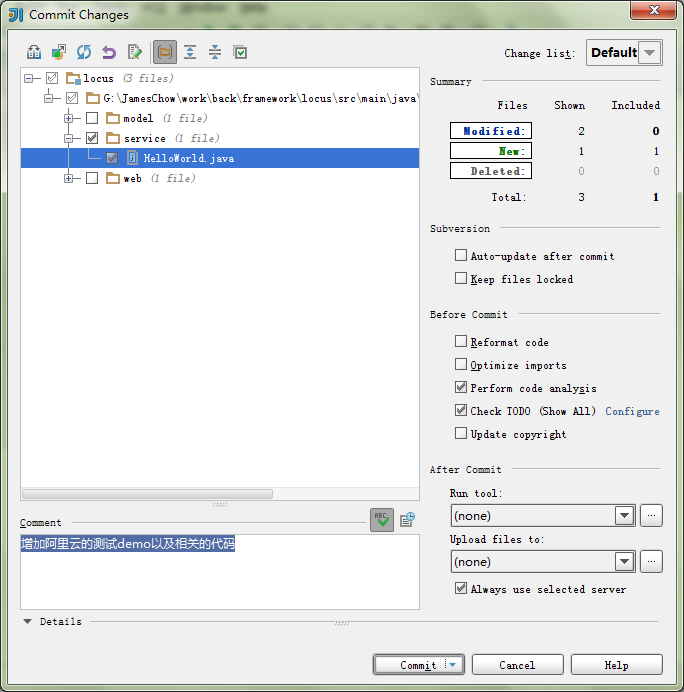
**4.2.1 版本管理器对项目进行管理**



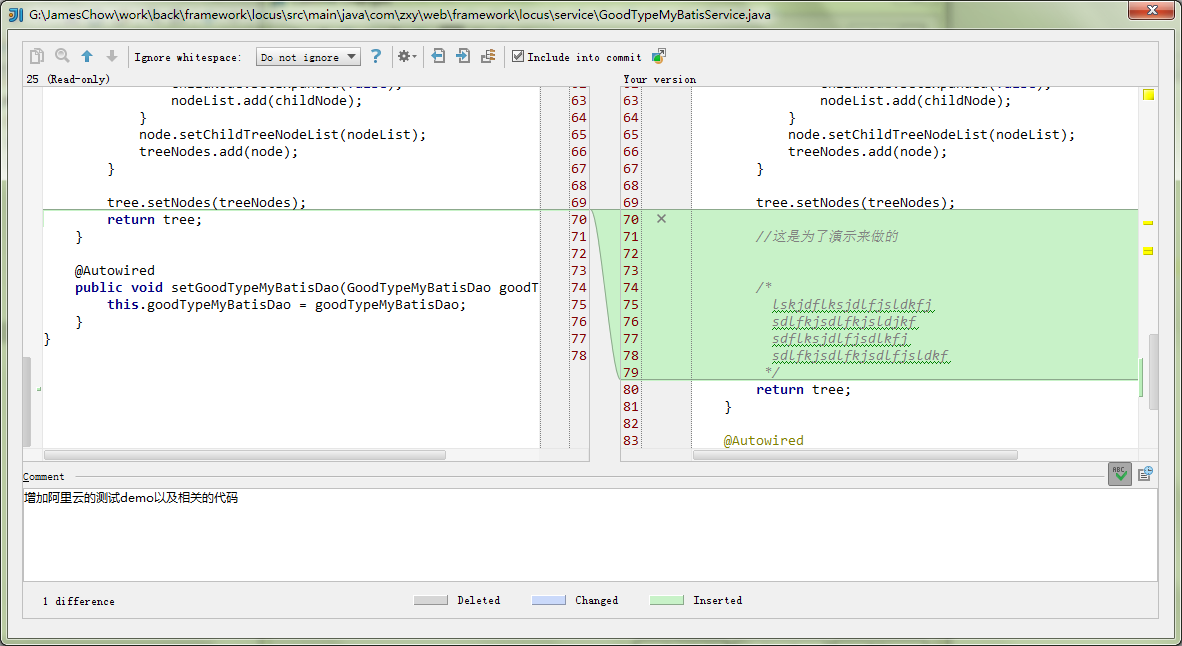
一般在实际的生产环境中，项目都是存放在各种管理软件中。我们可以从图中的管理软件中检出自己项目。

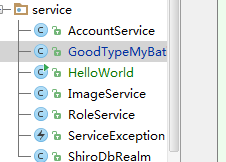


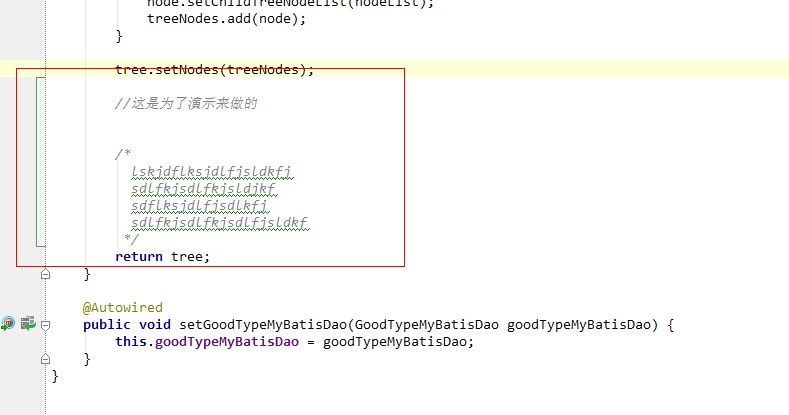
当你新建一个类或者任何一个文件的时候，他会询问你是否要把这个添加到管理文件中去。SVN是没有这个概念的，只有在git中才有这样的概念，没有”add”的文件是不会被提交上去的。当选择no的时候，整个文件会变为红色，那么在提交的时候这个文件会被忽略。如果选择是的话就是绿色。提交的时候这个文件也会被提交。



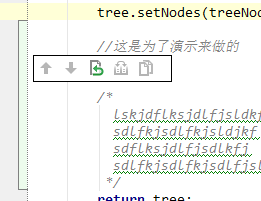
提交的时候可以看到具体的信息，这时候可以进行文件对比。双击要看的文件



以前提交过的文件，如果被修改了就会变为蓝色。



在我的项目中随便增加了一点代码，那么可以看到左边的位置多了绿条，这说明这是新加的代码。

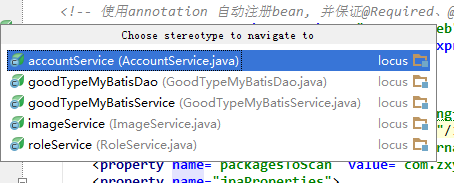


可以选择回滚，比较等操作。非常的方便，这里不一一演示了。

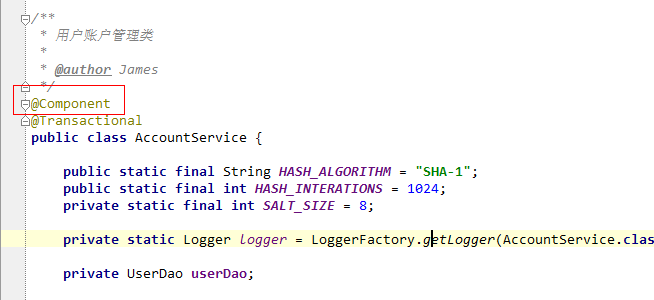
**4.2.2 Spring等框架的提示工作**



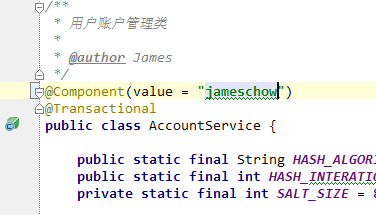
比如我们这里一个配置就是扫描某个包下的Bean只要是@Service @Component等注解标注的类就视为BEAN，那么注意左侧的那绿叶，点击之后就可以查看它实际扫描了哪些类。



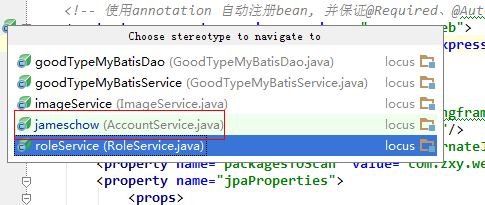
这样就可以直接跟踪进JAVA类了



因为使用@Component并没有命名，所以Bean的id就是accountService

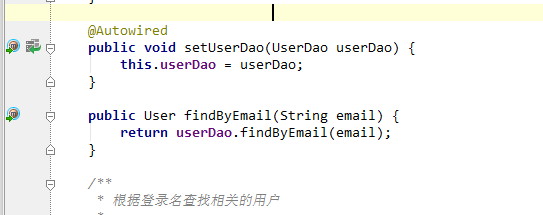


我们将其改为jameschow



再回到前面的配置文件，可以看到这里还标注了这个类被修改过了，现在ID是多少。

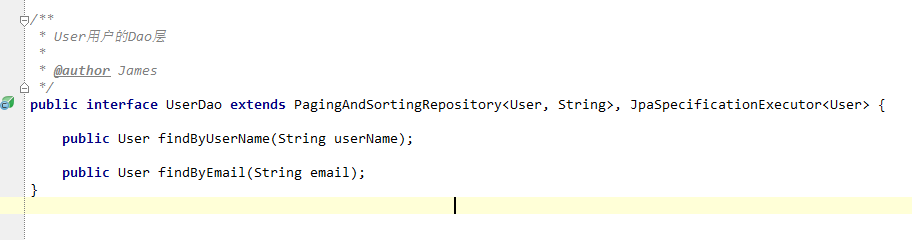
同样的在类名里有绿叶就是表示这个是一个BEAN可以找到其声明的地方，非常方便。



这个表示其是受事务管理的，点击可以到达事务的声明点



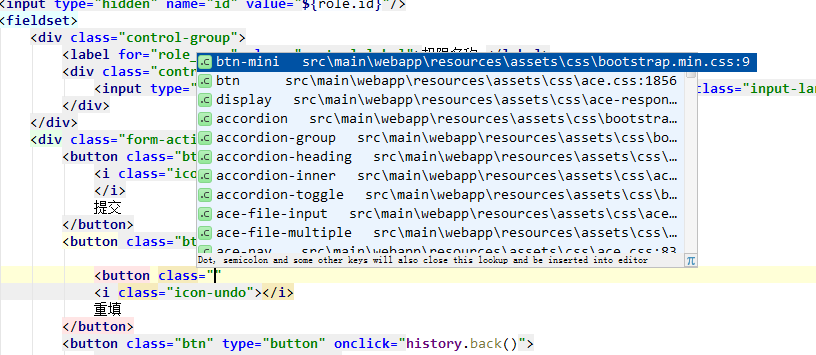
这个表示这是通过Spring注入进来的点击可以找到注入bean的位置





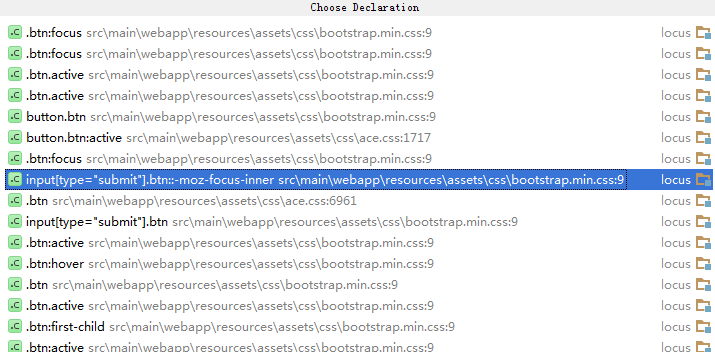
当然SpringMVC也有许多功能，考虑到老师你使用的情况较少我就不一一介绍了。

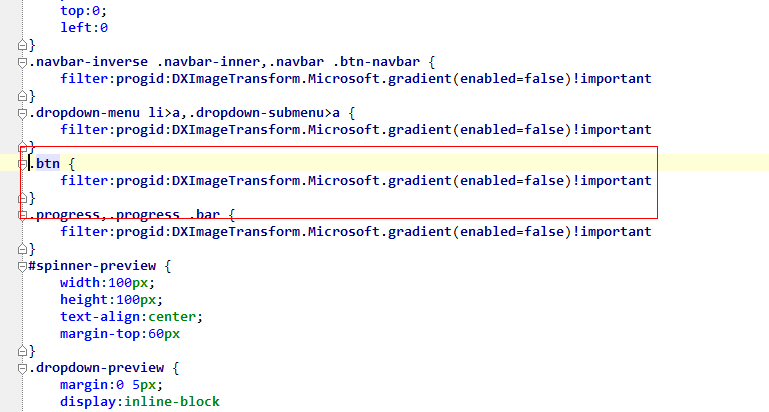
**4.2.3 前端框架的提示**



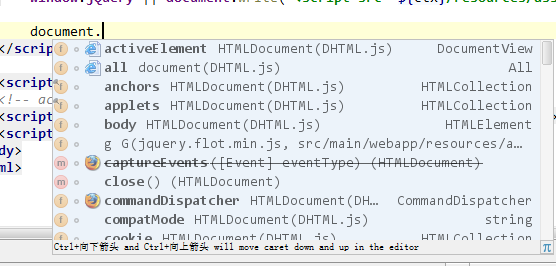
Jsp文件里面一个元素要应用样式，会进行准确的提示。

Ctrl+鼠标 就可以定位到所在CSS声明的位置

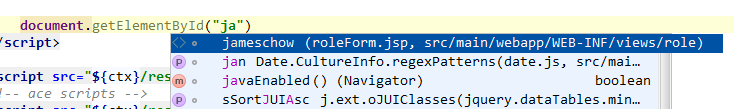




Js也一样的，第一个方法都可以准备定位到不同的地方。

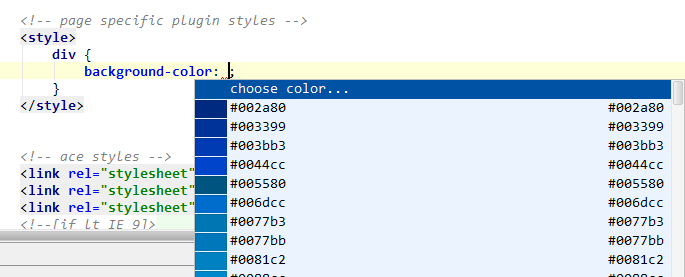


这是其JS提示，不同浏览器符合代表在不同的浏览器写法中兼容性。

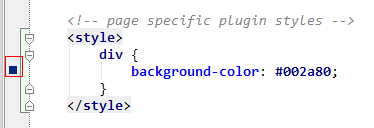


Intellij还可以定位html元素的位置的，同样的ctrl+鼠标 就可以定位到该html声明的位置





书写CSS的时候，只要关于色彩类的就可以直接选择。有拾色板的。



定义完成之后，还可以在左侧看到其表现的色彩。基本的使用就这些了