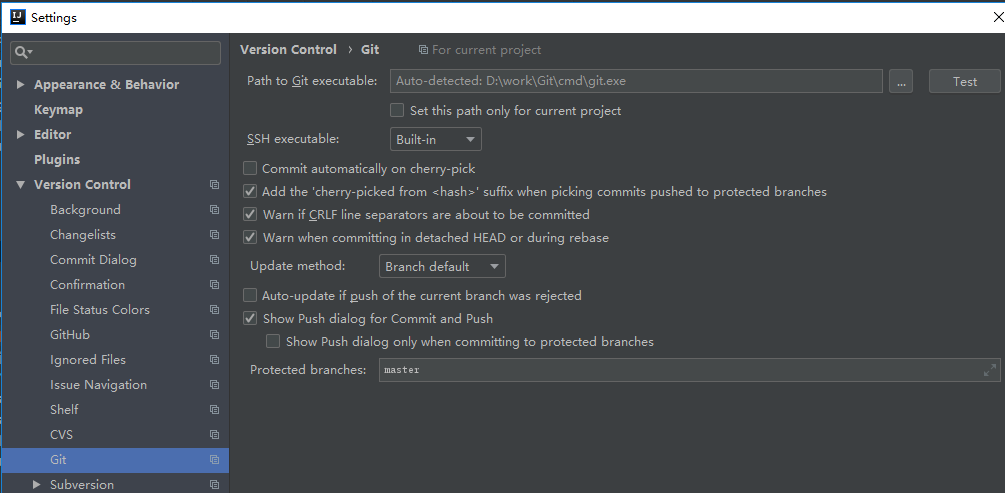
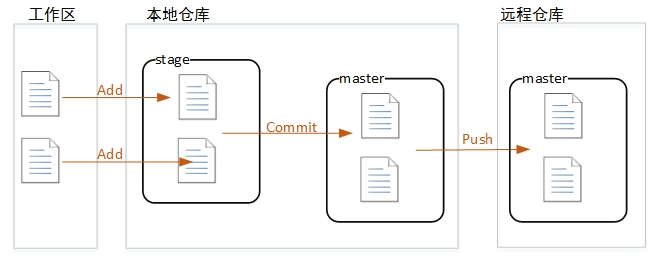
# IDEA中Git指南

安装git客户端后，发现IDEA自动配置了git位置，如下：



## 项目初始提交

下图是Git与提交有关的三个命令对应的操作，Add命令是把文件从IDE的工作目录添加到本地仓库的stage区，Commit命令把stage区的暂存文件提交到当前分支的仓库，并清空stage区。Push命令把本地仓库的提交同步到远程仓库。



### .gitignore文件编写

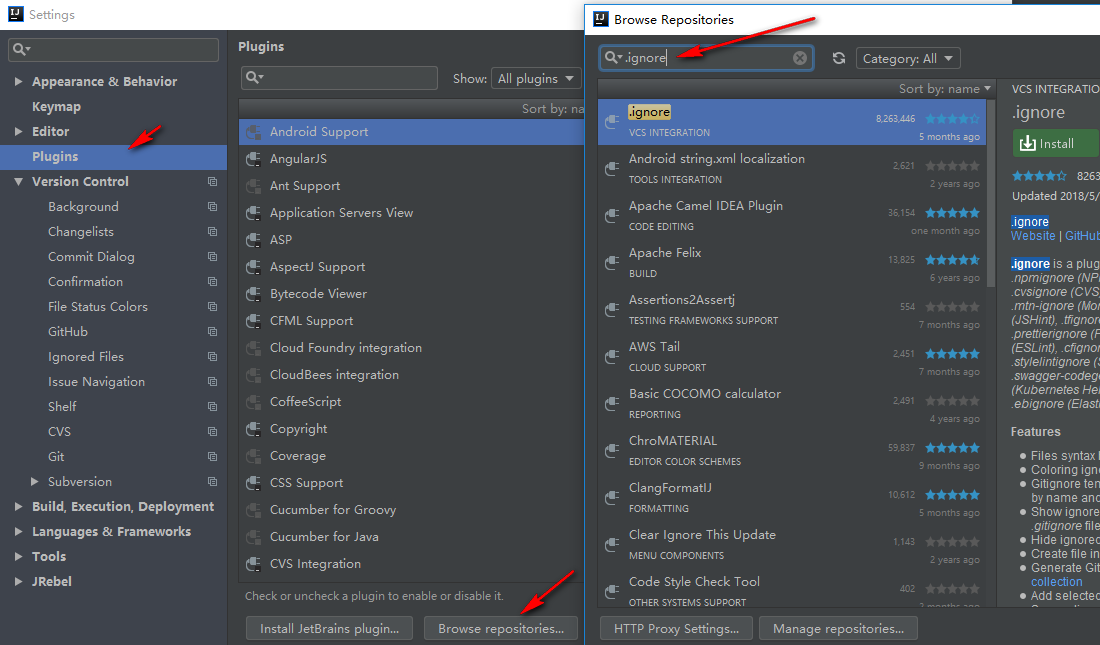
创建maven项目，使用git提交，有时需要忽略不必要的文件或文件夹，只保留一些基本。

例如如下截图，实际开发中我们只需提交：src,.gitignore,pom.xml



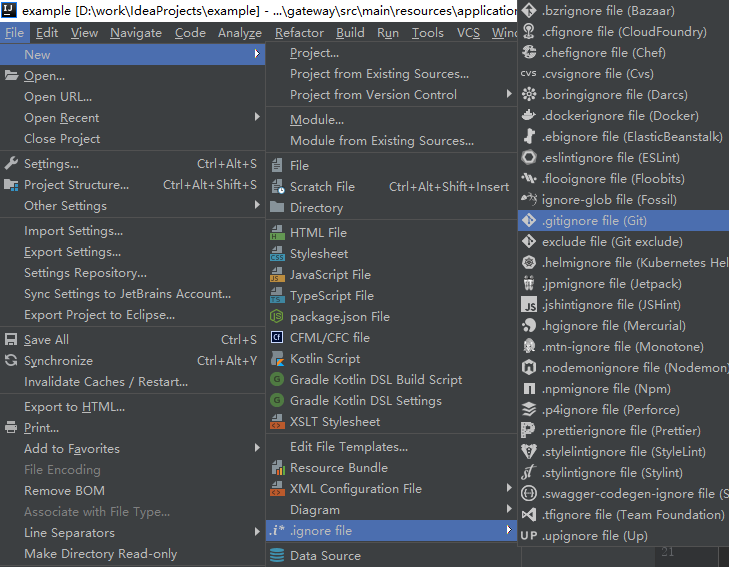
而自己项目文件一般都保留，但是有些则不必要提交，如maven项目的target文件夹等等。

使用.ignore插件生成.gitignore文件

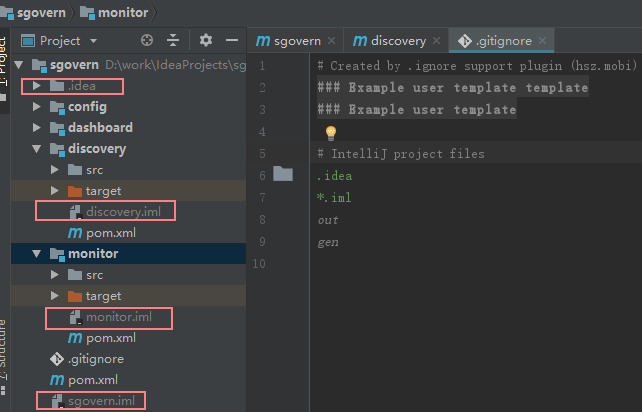


安装后重启生效。

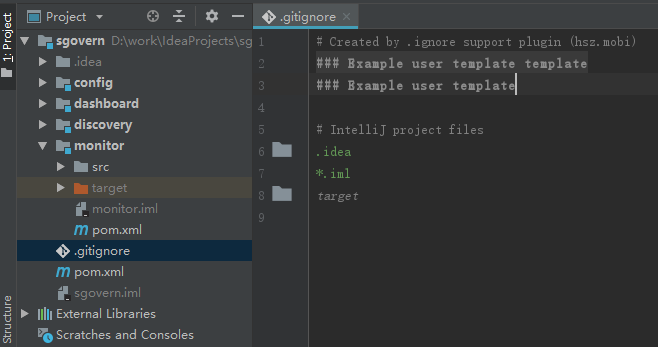
选择某工程，



然后选择要忽略的文件



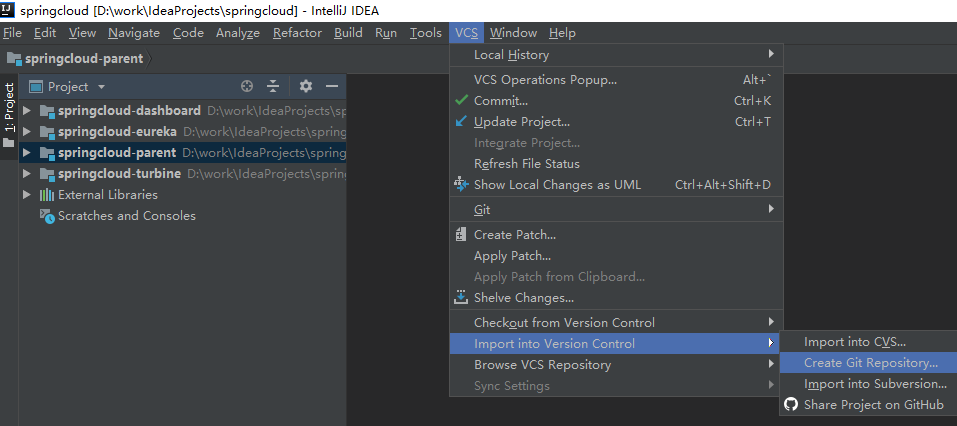
然后就会看到了文件置灰了。



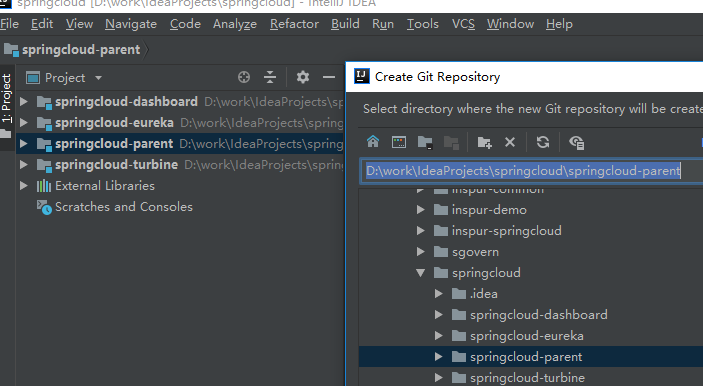
<https://www.cnblogs.com/javabg/p/8567790.html>

### 创建本地仓库

创建好项目，选择VCS - > Import into Version Control -> Create Git Repository

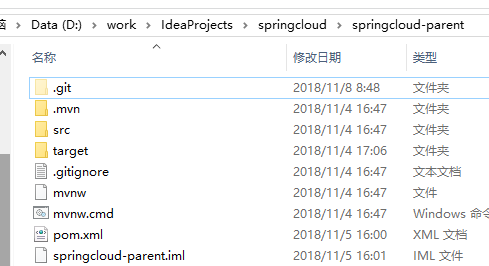


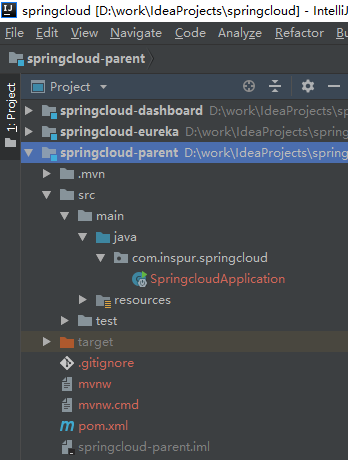
接下来指定本地仓库的位置，按个人习惯指定即可，例如这里选择了项目源代码同目录，这里选择的是模块目录。



这样我们就把idea工程创建了一个Git Repository了，在工程文件目录下就会多了一个

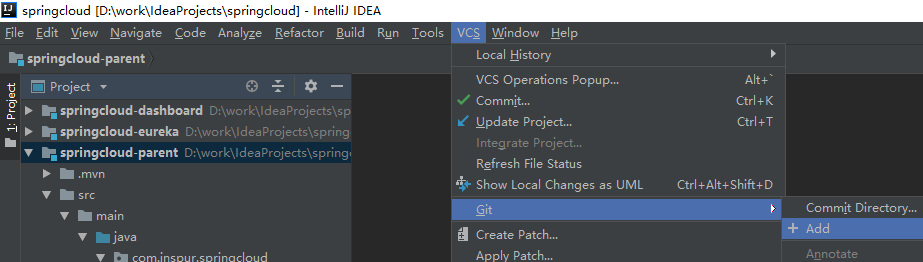
.git文件夹（隐藏文件夹）。这时项目中的文件会变成红色，表示该文件还没有交给git，需要你再执行下git的add命令。

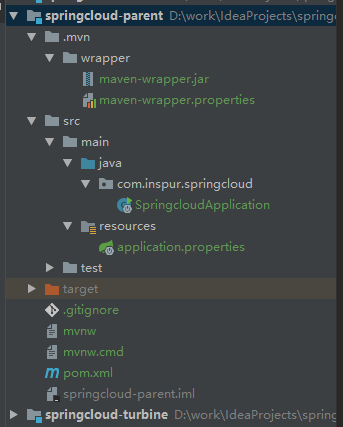




### 提交至本地git仓库

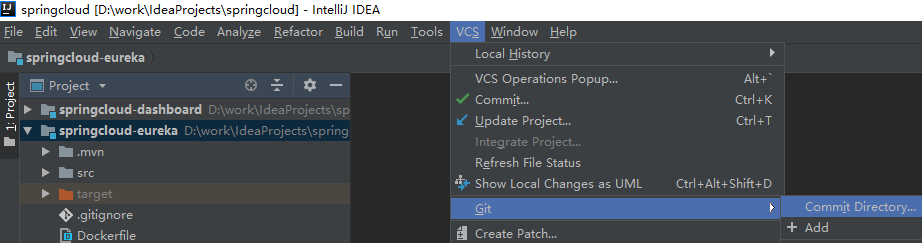
3、提交到本地git仓库：在模块名点右键或点菜单VCS，选择Git -> +Add，选择你要提交到版本控制的文件，然后点Commit。（此时只是提交到了本地git仓库）





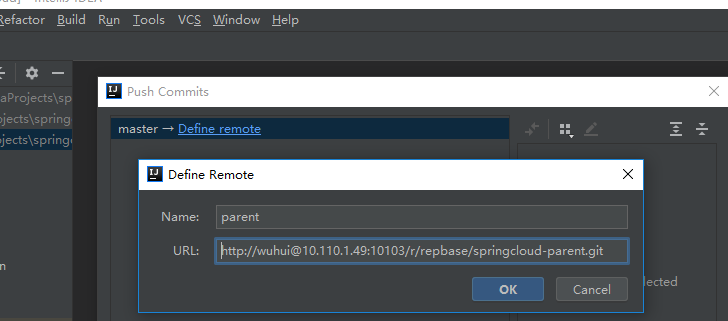
文件变成绿色，表示该文件已经执行了git的add命令。

然后，commit Directory

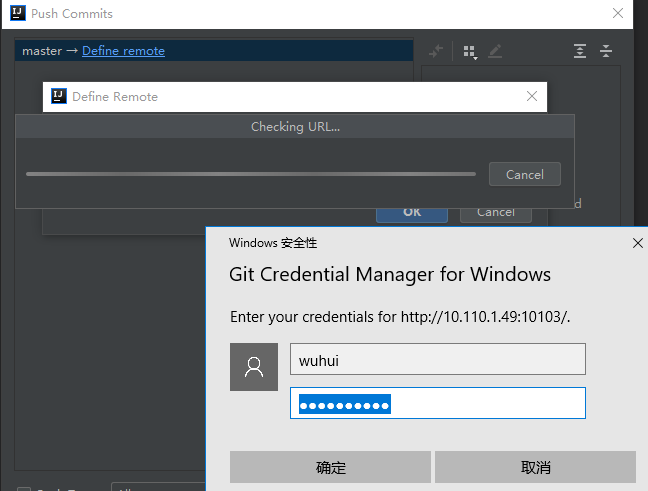


### Push至远程仓库

4、将本地仓库push到服务器远程仓库。选择VCS -> Git ->Push，在弹出的界面中点Define Remote，输入git服务器URL，Name可以写成模块名方便识别，然后Push即可。



此后，弹出密码页面



错误解决：

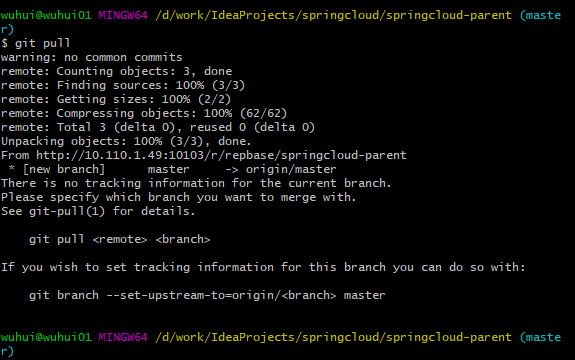
Push时出现如下错误



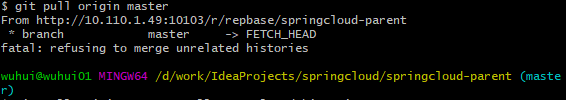
1.切换到自己项目所在的目录，右键选择GIT BASH Here

2.在terminl窗口中依次输入命令：

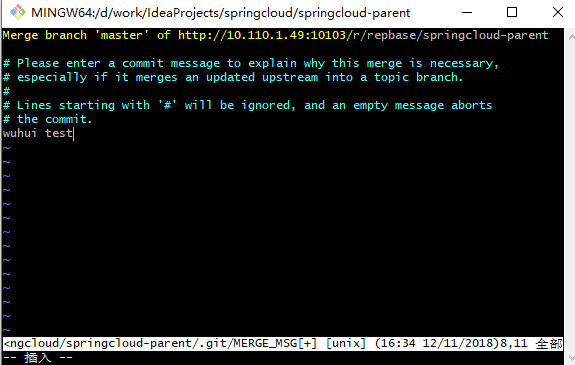
git pull



git pull origin master

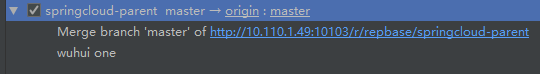


git pull origin master --allow-unrelated-histories



ESC🡪:wq

3.在idea中重新push自己的项目，成功！！！



成功提示如下：

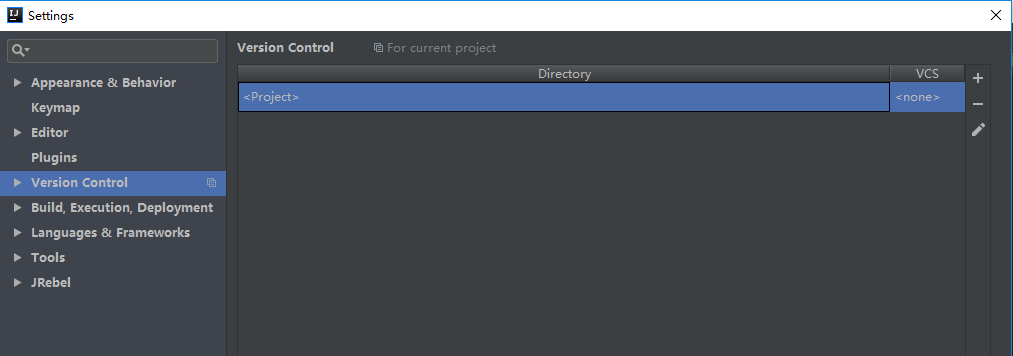


### 多module 项目提交至不同的git仓库

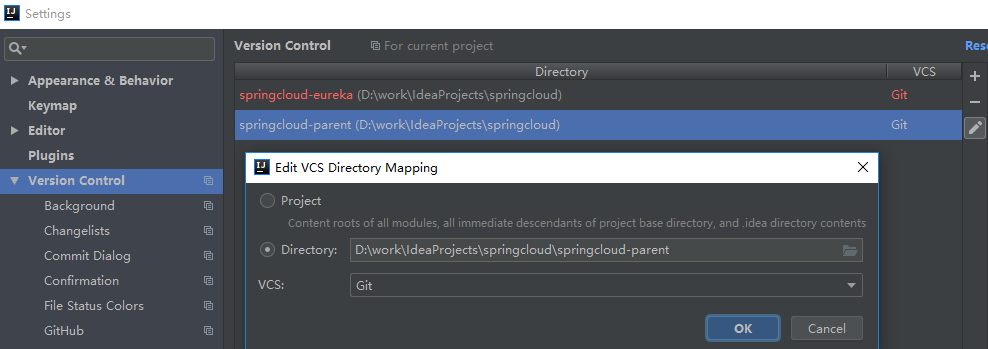
https://www.cnblogs.com/cnsugar/p/7998819.html

说明：本文档适用于一个工程多个模块的项目，每个模块对应不同的git服务器地址。

1、打开 File -> Settings，选择 Version Control



选择 <Project> 点击右侧的 编辑 图标



https://cloud.tencent.com/developer/news/40670

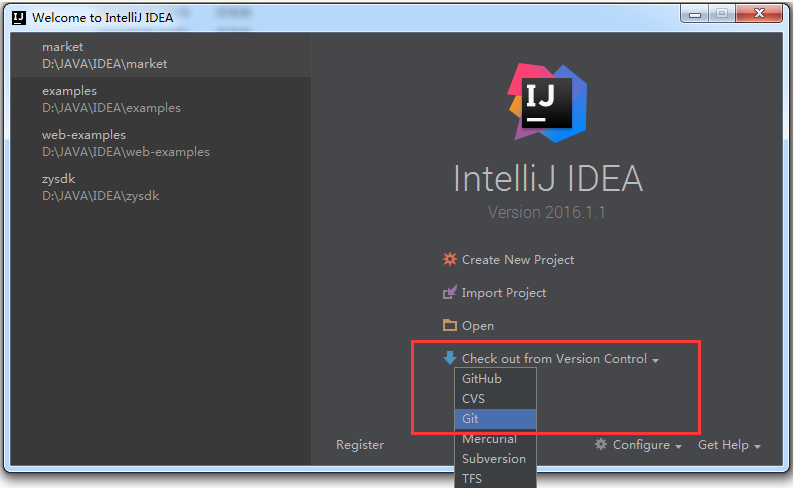
https://www.cnblogs.com/alone-striver/p/7745744.html

点击右上角的+添加Directory，选择本地模块目录，VCS选择Git，然后点OK

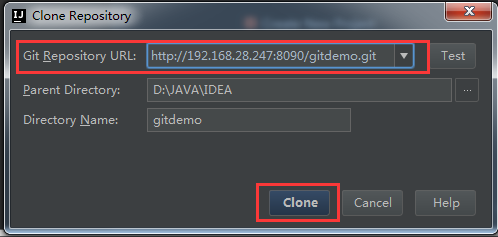
## 检出项目

### 从远程Git仓库上获取项目源码

即克隆项目，操作如下：



输入Push时填写的远程仓库地址

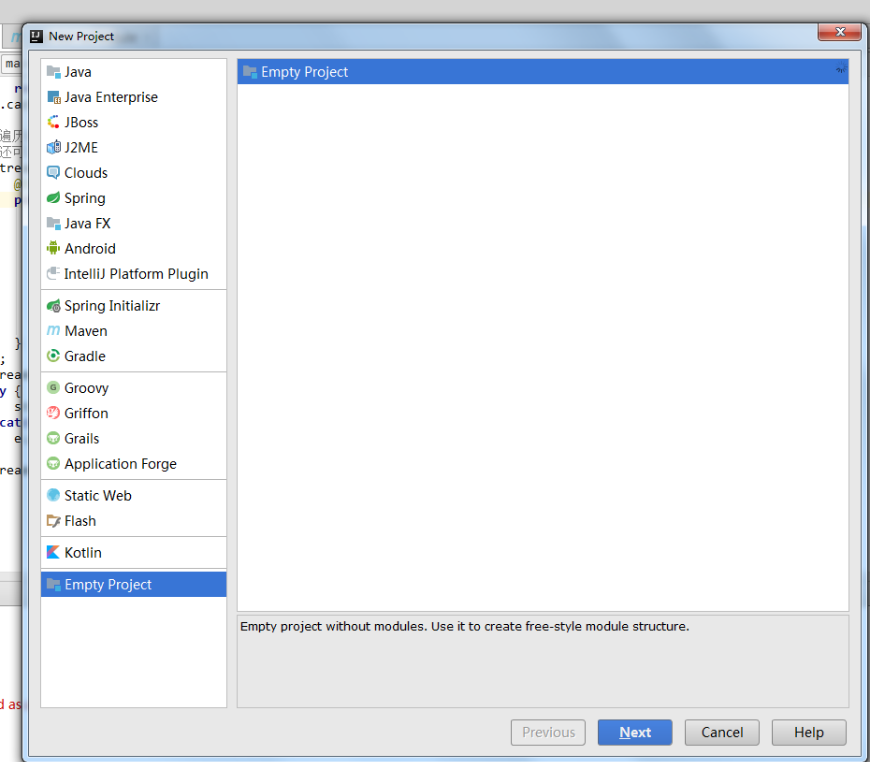


接下来按向导操作，即可把项目从远程仓库克隆到本地仓库和IDE工作区。

### 检出项目的模块

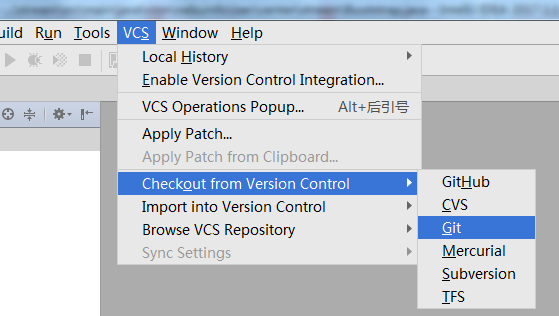
#### 创建空工程

1、创建一个空的工程。File -> New -> Project。

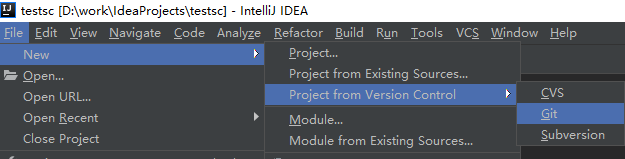


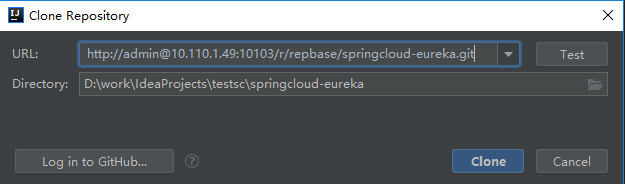
#### 克隆模块

2、选择VCS -> Checkout from Version Control -> Git，输入Git服务器地址，Parent Directory 选择刚刚创建的工程目录，点Clone。注意!!!：检出成功之后会提示是否创建IDEA工程，选否。重复此步骤检出其他模块。

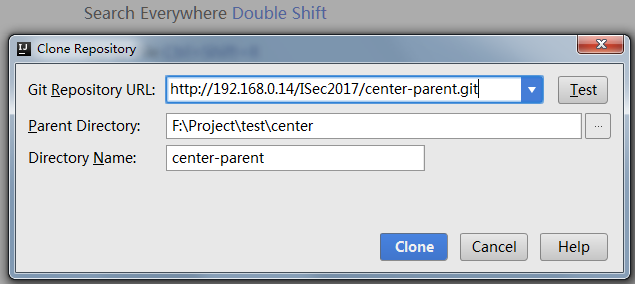


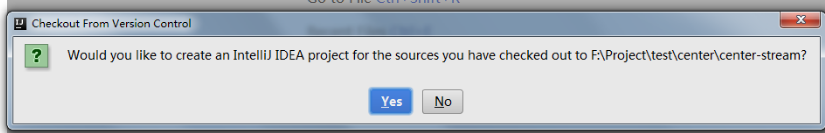
或者





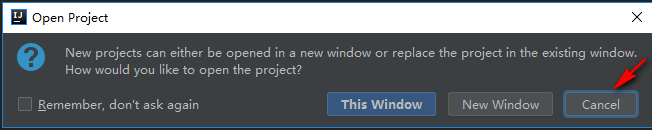
注：老版本：



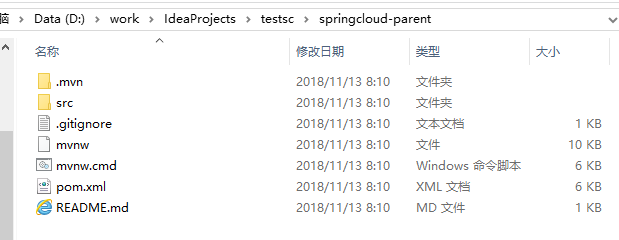


--------------------------------------------------------------------------------------------------------------

Clone完后，提示如下，选择 cancel

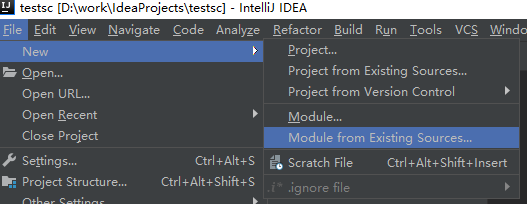


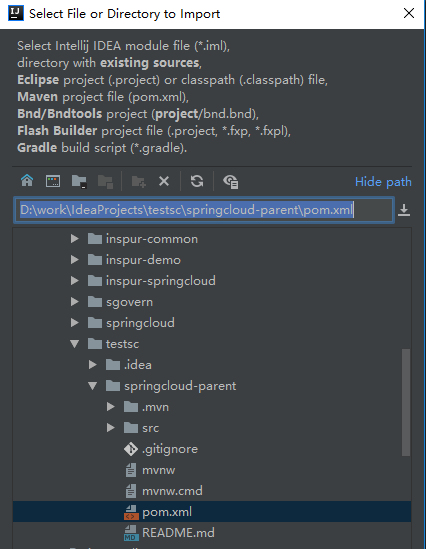
此时，已经从远程仓库下载下来了



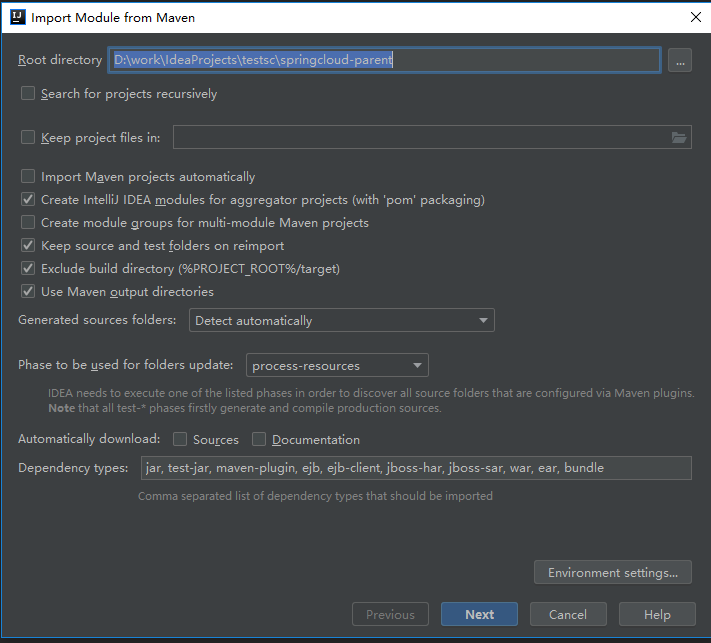
#### 导入模块

3、导入模块：File -> New -> Module from Existing Sources，按普通导入模块即可。

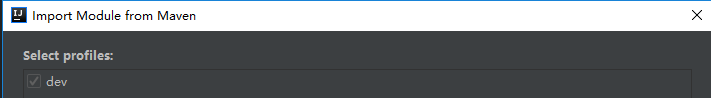


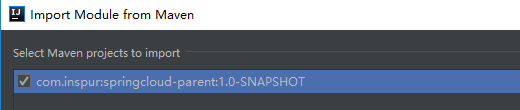


选择pom.xml文件



默认设置即可。





## 修改后提交

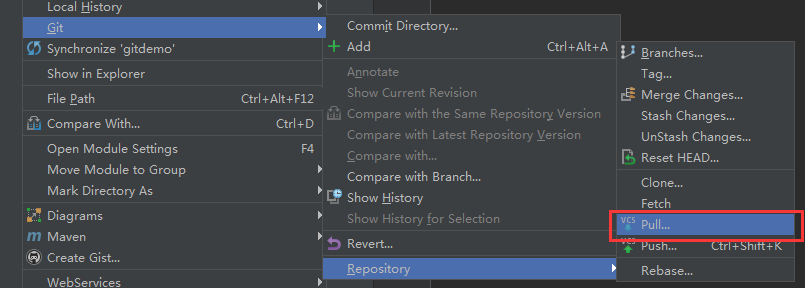
场景三：修改了部分源码，提交到远程仓库

这个操作和首次提交的流程基本一致，分别是 Add -> Commit -> Push。请参考场景一

## 获取他人提交更新

场景四：小张从远程仓库获取小袁的提交

获取更新有两个命令：Fetch和Pull，Fetch是从远程仓库下载文件到本地的origin/master，然后可以手动对比修改决定是否合并到本地的master库。Pull则是直接下载并合并。如果各成员在工作中都执行修改前先更新的规范，则可以直接使用Pull方式以简化操作。

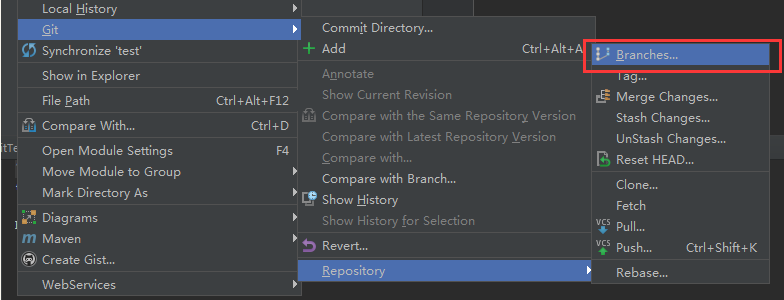


## 分支开发

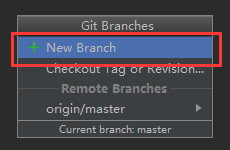
场景五：小袁接受了一个新功能的任务，创建了一个分支并在分支上开发

建分支也是一个常用的操作，例如临时修改bug、开发不确定是否加入的功能等，都可以创建一个分支，再等待合适的时机合并到主干。

创建流程如下：

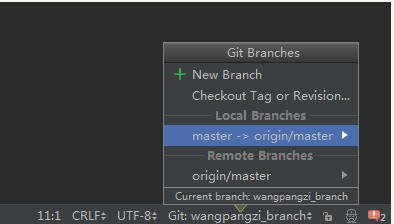


选择New Branch并输入一个分支的名称

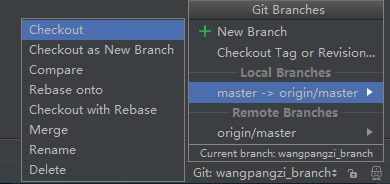


创建完成后注意IDEA的右下角，如下图，Git: wangpangzi\_branch表示已经自动切换到wangpangzi\_branch分支，当前工作在这个分支上。

点击后弹出一个小窗口，在Local Branches中有其他可用的本地分支选项，点击后选择Checkout即可切换当前工作的分支(见场景7操作切换其他分支)。



如下图，点击Checkout

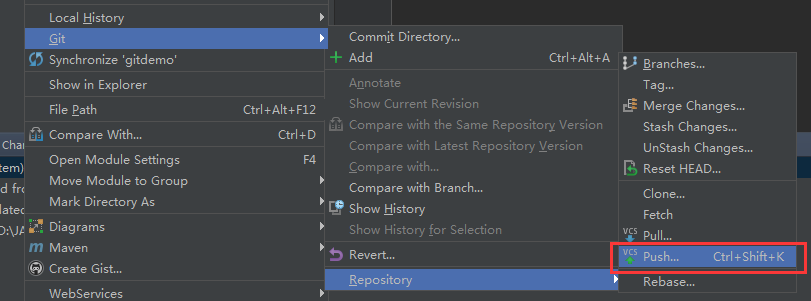


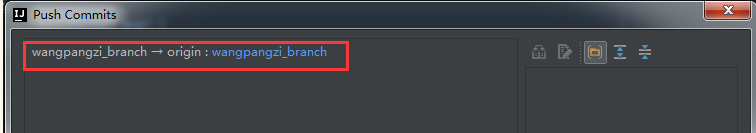
注意，这里创建的分支仅仅在本地仓库，如果想让组长小张获取到这个分支，还需要提交到远程仓库。

## 分支提交至远程git仓库

场景六：小袁把分支提交到远程Git仓库

切换到新建的分支，使用Push功能

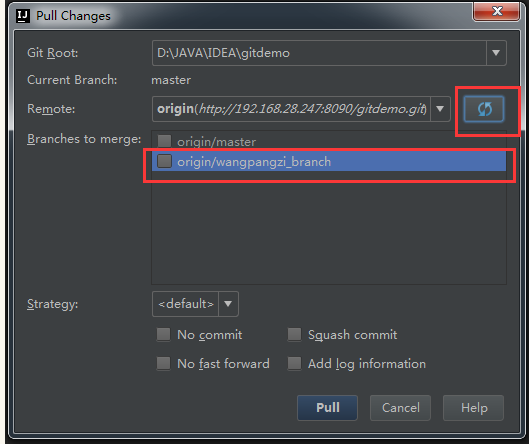




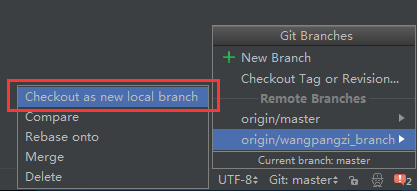
## 获取他人提交的分支

场景七：小张获取小袁提交的分支

使用Pull功能打开更新窗口，点击Remote栏后面的刷新按钮，会在Branches to merge栏中刷新出新的分支。这里并不想做合并，所以不要选中任何分支，直接点击Pull按钮完成操作。



更新后，再点击右下角，可以看到在Remote Branches区已经有了新的分支，点击后在弹出的子菜单中选择Checkout as new local branch，在本地仓库中创建该分支。完成后在Local Branches区也会出现该分支的选项，可以按上面的方法，点击后选择Checkout切换。

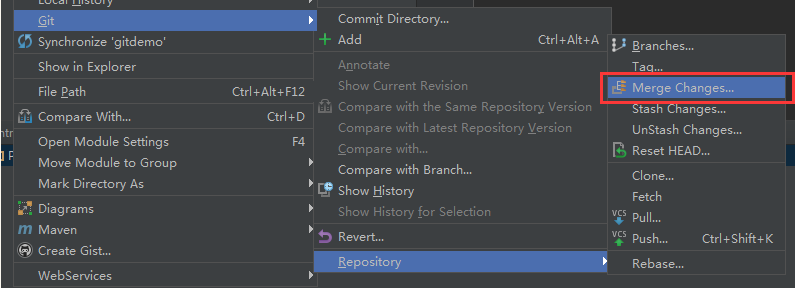


## 分支合并到主干

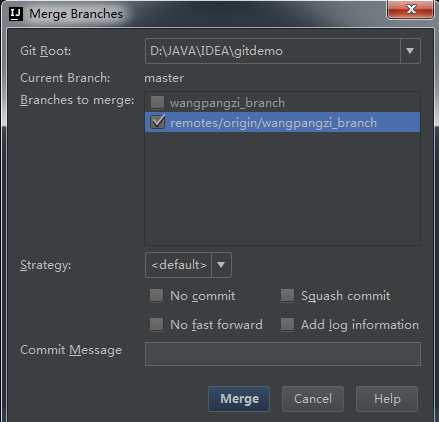
场景八：小张把分支合并到主干

新功能开发完成，体验很好，项目组决定把该功能合并到主干上。

切换到master分支，选择Merge Changes

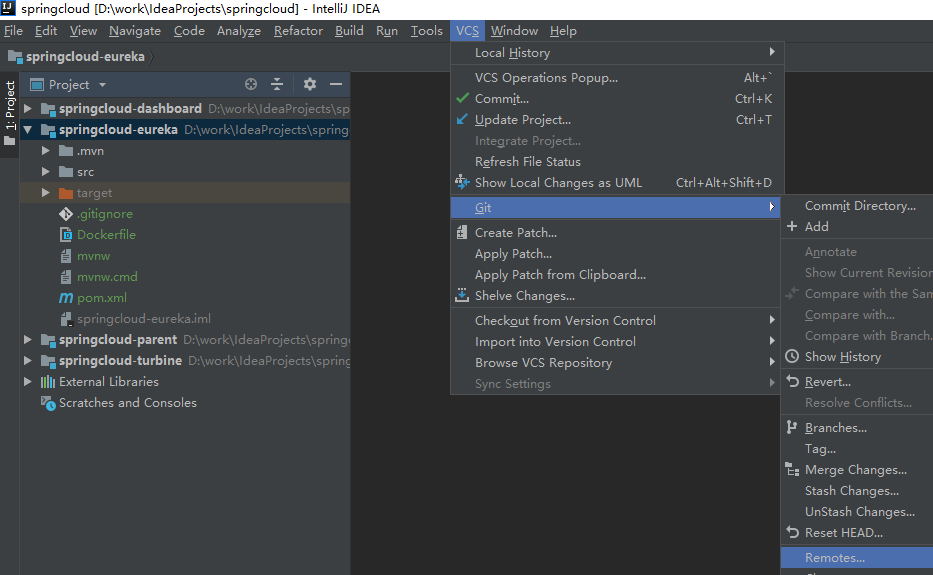


选择要合并的分支，点击Merge完成



## 日常操作

### 配置remotes



https://www.cnblogs.com/java-maowei/p/5950930.html