**全面介绍[Android](http://lib.csdn.net/base/android" \o "Android知识库" \t "http://blog.csdn.net/gao_chun/article/details/49817229/_blank)** **Studio中[Git](http://lib.csdn.net/base/git" \o "Git知识库" \t "http://blog.csdn.net/gao_chun/article/details/49817229/_blank)的使用**

# 一、Git 与GitHub

这里首先介绍下Git与GitHub是什么东西，他们之间有何区别呢？

Git ：分布式版本控制系统，最初用在**[Linux](http://lib.csdn.net/base/linux" \o "Linux知识库" \t "http://blog.csdn.net/gao_chun/article/details/49817229/_blank)**上，可以和SVN、CVS等作为同一个概念理解，目前世界上最先进的分布式版本控制系统（没有之一）。

GitHub：基于Git的项目托管平台，简单的说就是一个网站，通过注册账号后，可以使用Git服务，它提供了可以在上面创建资源仓库的Web页面，通过shell命令或者图形用户界面可以和远程项目同步更新，对项目进行管理。还是不太清楚？就是一个项目由多个人一起干，开发中对于项目里的文件，每个人都有可能拿同一文件进行来编辑，为了协同开发，就有了像Git（版本控制）这样的东西。而GitHub呢，就相当于项目放置的平台罢了，不过它里面有很多不同的开源项目，往往是很多人协同开发的， 借助Git来管理，相对于Git本地仓库来说，它是一个远程仓库。

# 二、安装Git

关于Git的安装可去网上搜索安装教程，这里就不再赘述了。

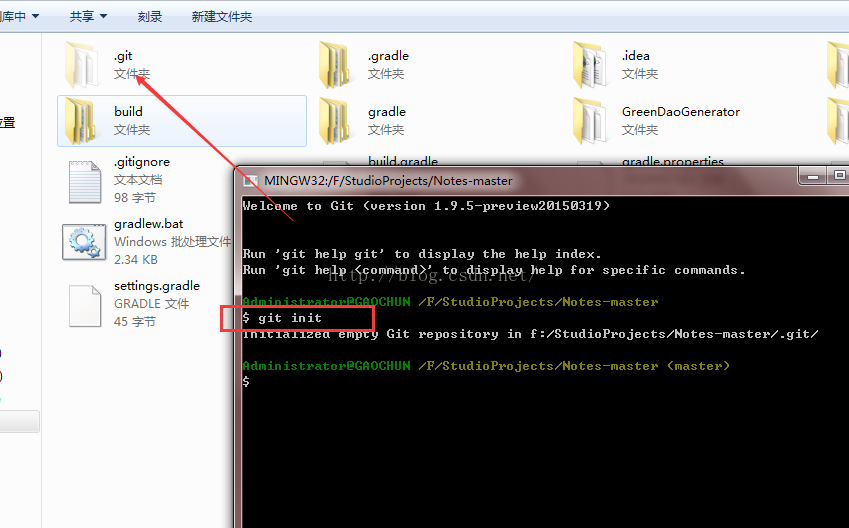
下载链接：[http://git-scm.com/download/](http://git-scm.com/download/" \t "http://blog.csdn.net/gao_chun/article/details/49817229/_blank)

# 三、创建本地仓库并提交代码

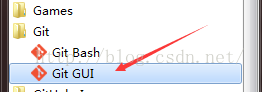
关于如何创建本地仓库并管理项目，可以使用Windows中Git的图形用户界面进行操作，同样也可以使用命令来创建。至于Git客户端大家可以去网上下载安装。这里比如你已经安装了Git客户端，我们下面来做个示例：

1、首先对需要作为仓库的目录右键并选择Git Bash，之所以在当前目录下，是因为打开命令操作框后，当前就是根路径了，这个因人而异，个人比较懒，省得打开了再切换。

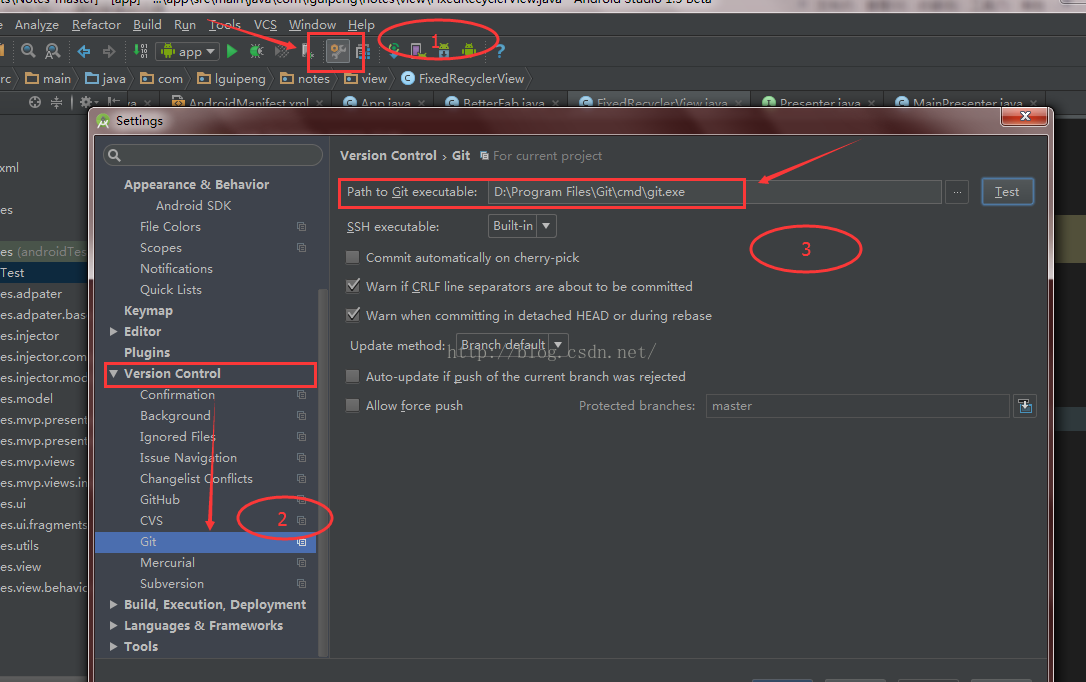
2、在命令端输入Git init 来初始化一个仓库，即会在该仓库的根目录下生成一个.git的目录，如下图：



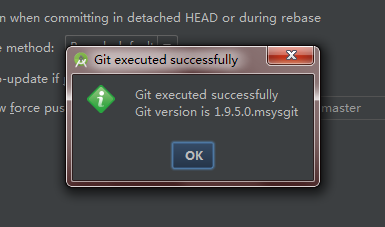
附：上面提到过同样可以采用图形界面来创建，打开 Git Gui，选择创建新的版本库。如下所示：

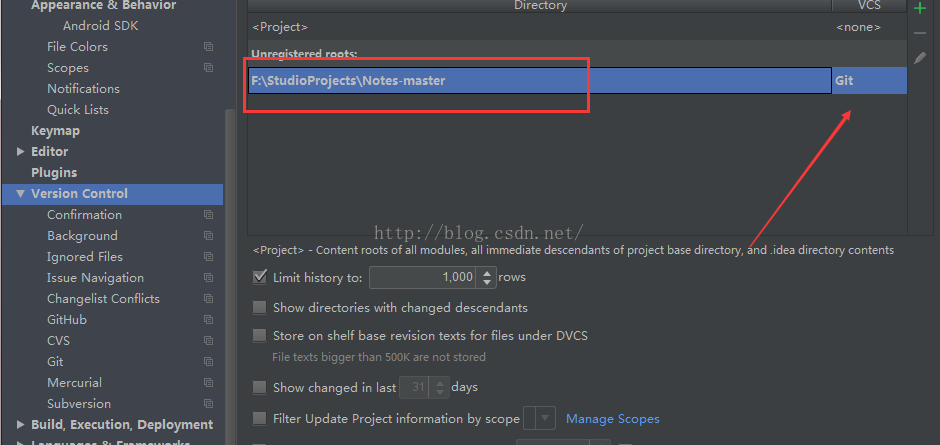
3、OK！上述工作我们已经创建了仓库，接下来打开Studio，我们来配置一下Git，在Settings中选择Version Control 并配置Git，不多说，上个图：



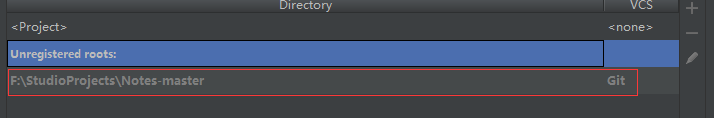
配置完后我们点下路径后的Test按钮，若出现该Success提示框则表明配置成功：

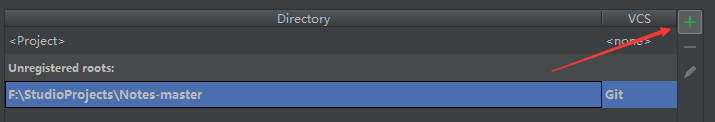


注：配置好后我们点击Version Control的时候，在右侧会提示该项目所采用的版本控制工具，如：



若发现该路径为灰色，需要选中，点击"+"

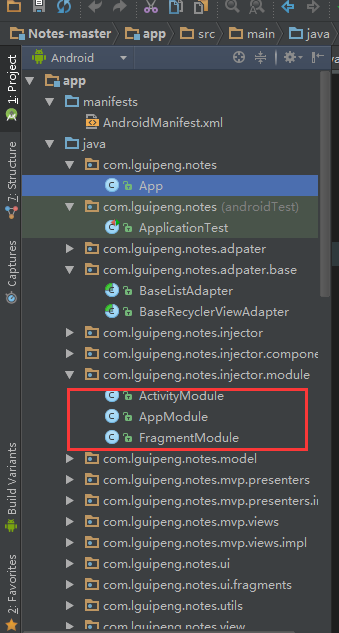




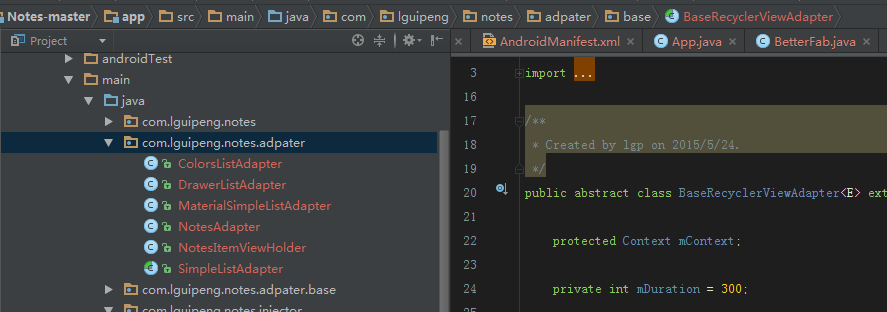
因为我们是在一个已有的项目上创建的仓库，所以配置完后需要为项目指定版本控制工具，也可以在最开始Create项目的时候直接指定仓库路径。

4、下面我们来将项目上传到仓库。

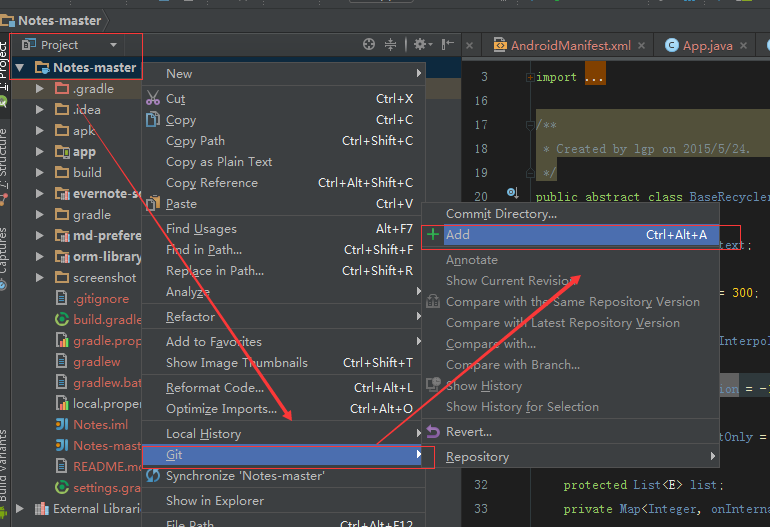
① 起初，我们项目所有文件颜色，都是 [白色：正常文件的颜色]



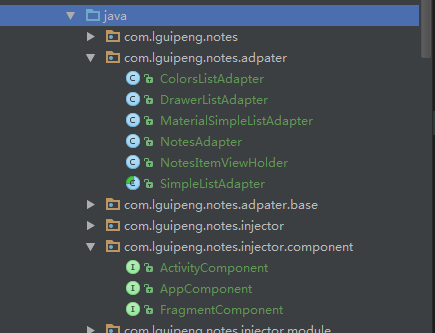
②但是当我们为项目指定仓库路径后，所有文件颜色的颜色变了 [红色：指定仓库路径后，未Add的文件]



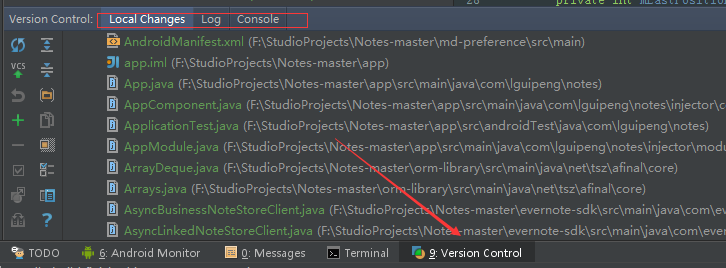
③切换为Project视图，对项目右键，Add文件。（在创建仓库的时候.gitignore文件已默认生成，可以修改添加需要ignore的文件）



Add成功后，我们在看看文件的颜色为 [绿色：已Add，但未commit的文件]

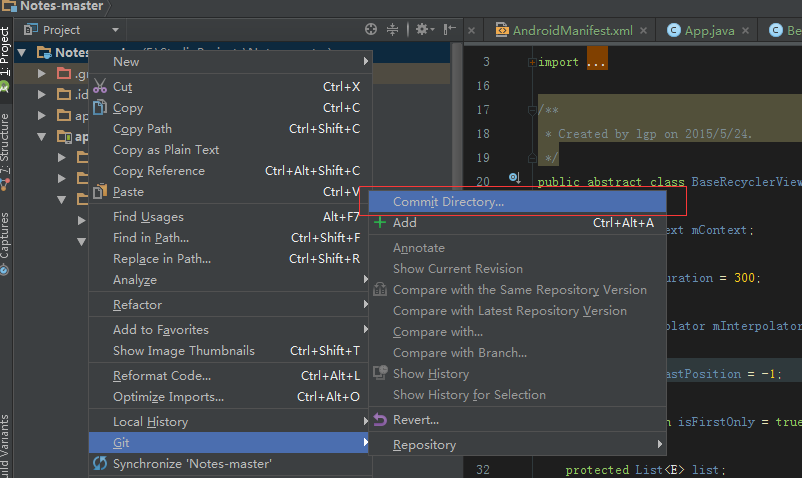


也可查看Log日志

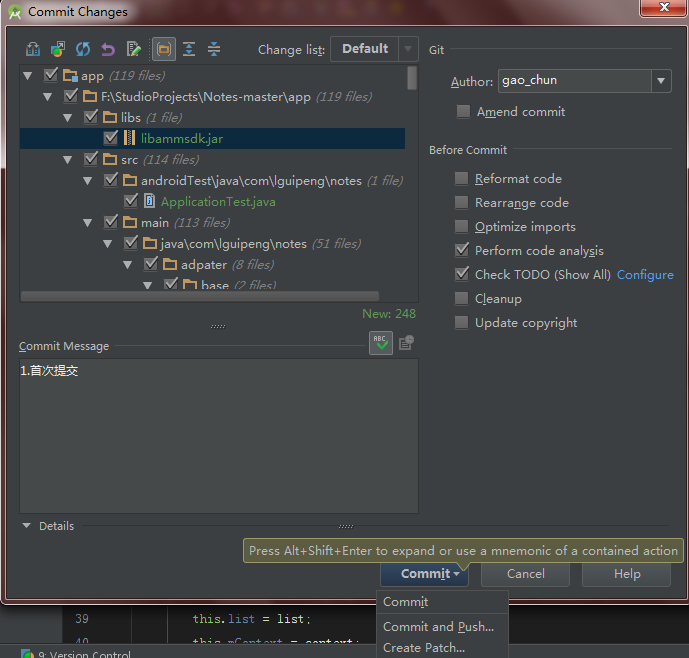


④Commit文件可通过 项目右键--> Git --> Commit Directory...  也可点击 工具栏上的两个按钮。



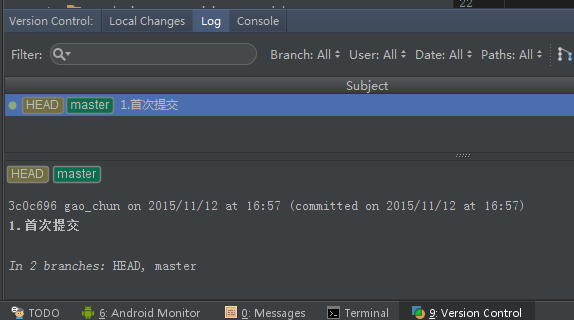


填写提交信息，然后可先Commit 然后再 Push 或者直接选择 Commit And Push ...

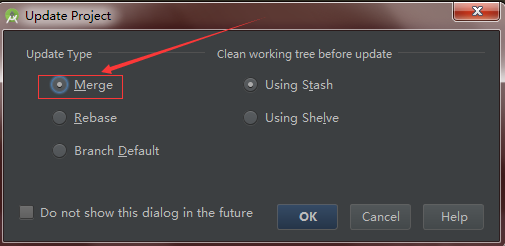


最后在提交完成时，由于项目中可能存在一些警告信息，提交后会出现提示框，选择提交便可。

⑤当我们对某个类进行修改后，会发现文件颜色又变了 [墨蓝色：提交成功后修改过的文件]，到此就将项目上传至仓库了，可以通过Studio中的Version Control一目了然的查看提交Log。



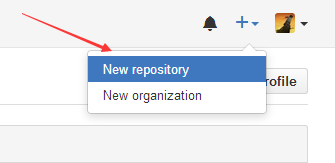
⑥关于如何下拉代码，我们可以点击工具栏上的按钮，然后在在弹出框中选择 Merge 合并。



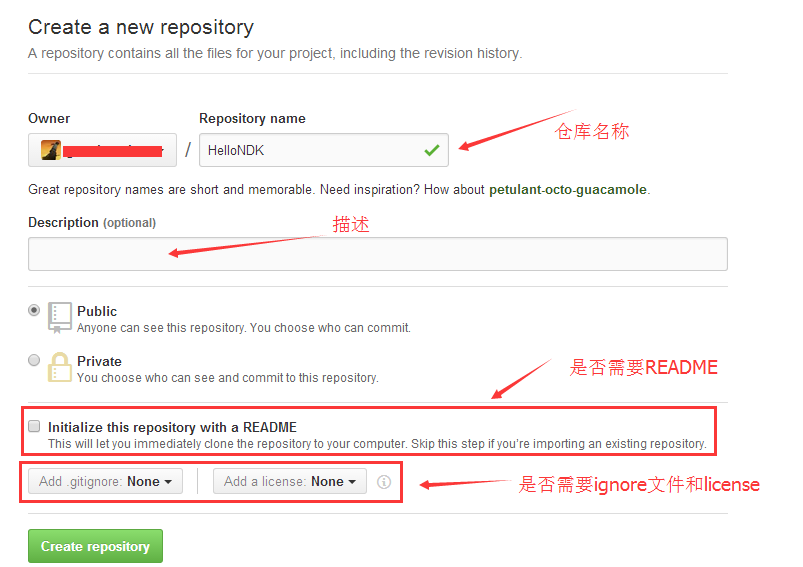
# 四、Studio上传项目至GitHub

说完本地Git仓库，那么如何将项目上传至远程的GitHub仓库呢？其实操作也基本上差不多。

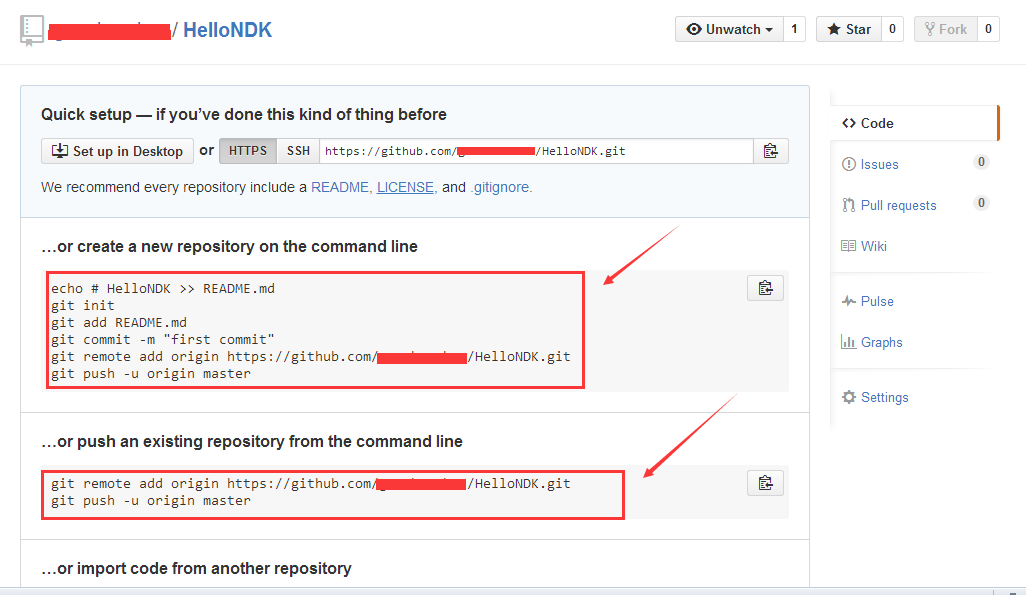
我们首先需要有一个GitHub账号，我们登录上去后new 一个仓库：



我们这里用一个示例做**[测试](http://lib.csdn.net/base/softwaretest" \o "软件测试知识库" \t "http://blog.csdn.net/gao_chun/article/details/_blank)**，名字叫做HelloNDK，并根据需要创建相关文件。



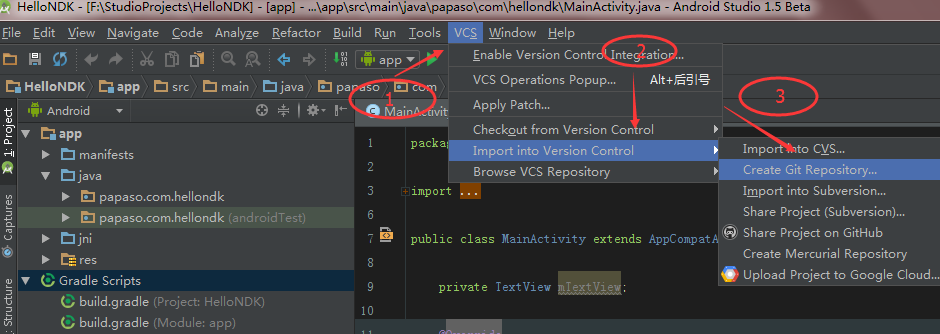
Create成功后，我们可以看到如下界面，这是GitHub给出的一些操作提示：



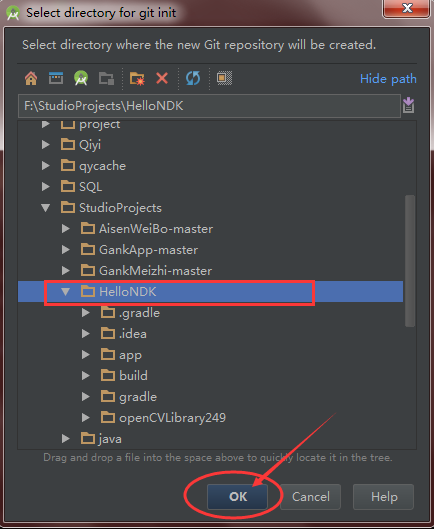
关于远程仓库创建暂时打住，文章后方一会继续补充一些关于.gitignore文件和README相关的知识点。这里我们就去看看如何用Studio将项目上传到刚刚创建的远程库吧。

注：和之前上传本地仓库相同，你可以去新建项目，也可能是已有的项目，新建项目的话，在Project location指定好项目路径就好，这里不多说了。

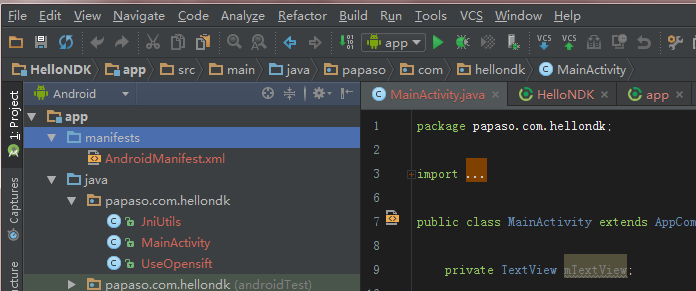
远程仓库建好了，本地项目也准备好了，那么我们现在需要怎么做呢？其实和上传到本地仓库一个道理，需要为本项目创建Git，所以我们接下来这么操作：



然后出现以下提示框，选择根目录，点击OK便可。

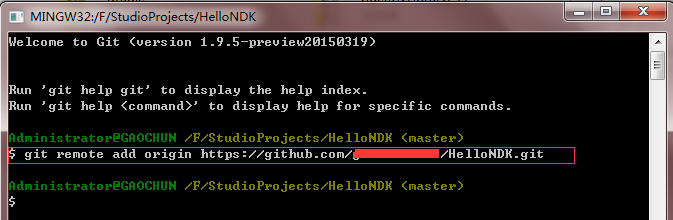


此时，有没有发现你项目中的文件颜色，已经全部变成了红色呢！

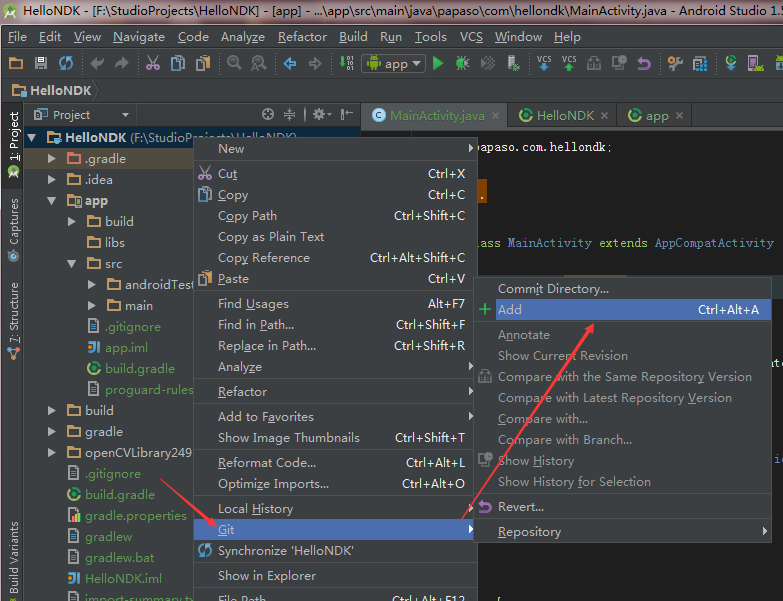


接下来需要做与之前本地仓库不同的事情，啥事呢？当然是 remote add 了，若你已经安装了Git for windows 客户端，则进入当前项目根目录，右键选择Git Bash，然后出现如下界面，并输入命令：

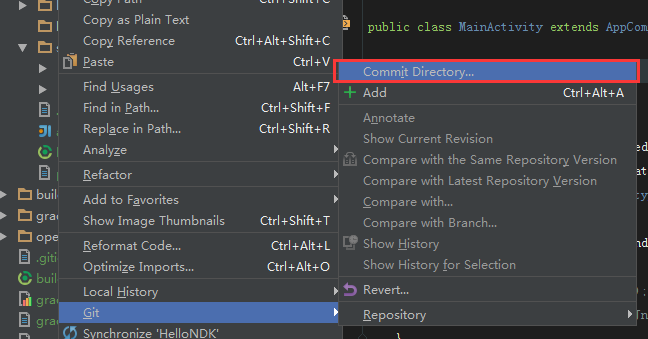
git remote add origin https://github.com/[username]/HelloNDK.git

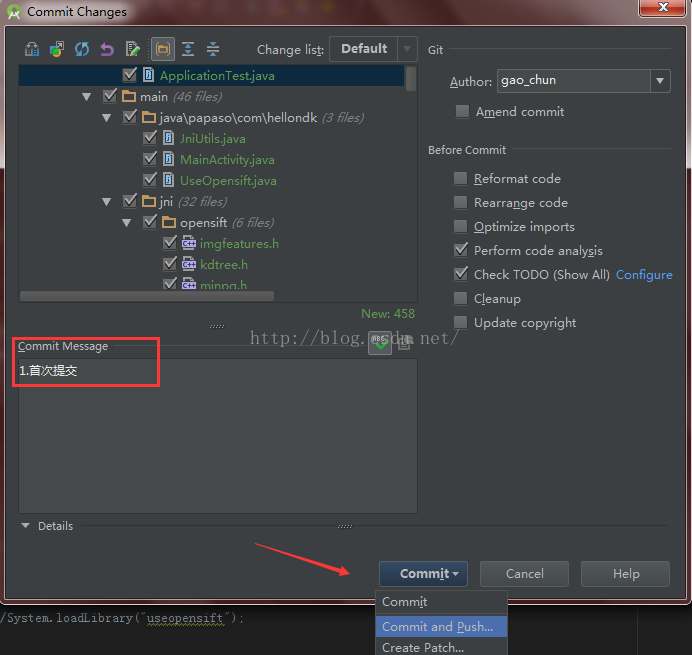


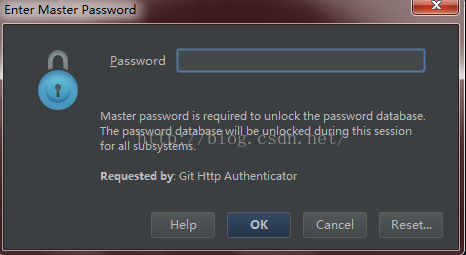
此时将会远程添加到你的Git repository，OK，我们开始Add文件，Add后文件变为绿色



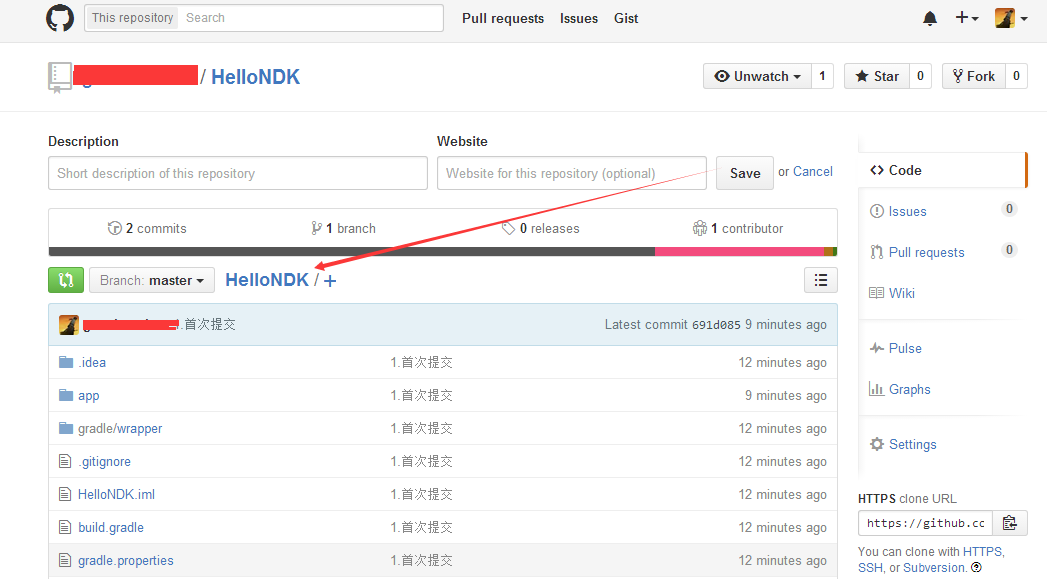
接下来Commit And Push ... ，其实操作和之前上传到本地仓库一样，有点不同的就是在push的时候，需要再次验证GitHub密码。







我们可以返回到GitHub页面，来看看之前创建的仓库是否有我们刚刚提交的代码：

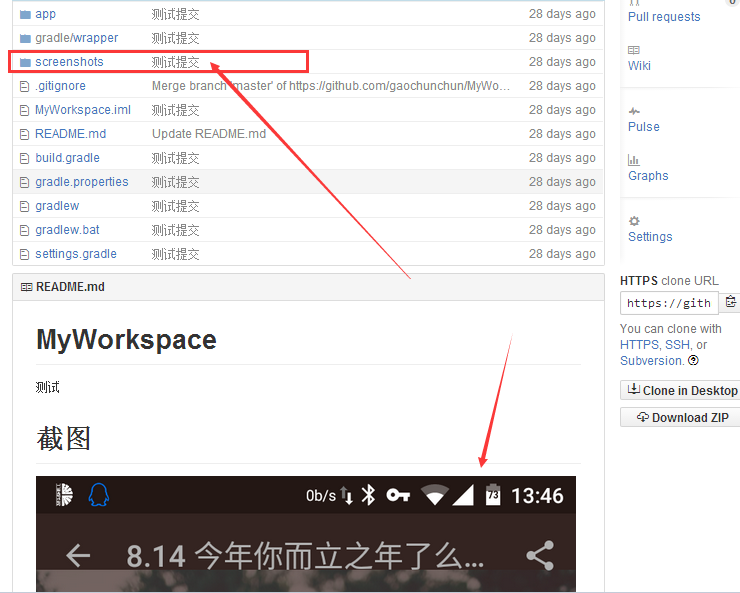


# 五、.gitignore文件和README文件

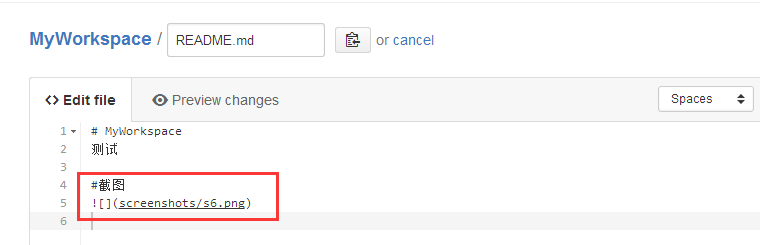
关于.gitignore文件如何写这里给个小参考：

1. **<<<<<<<** **HEAD**
2. .idea
3. .gradle
4. /local.properties
5. /.idea/workspace.xml
6. /.idea/libraries
7. /build
8. /captures
9. =======
10. **>>>>>>>** 09d68fcea9510dd91ecfb99b4c06bb2dfc187b22
11. # Built application files
12. \*.apk
13. \*.ap\_
15. # Files for the Dalvik VM
16. \*.dex
18. # Java class files
19. \*.class
21. # Generated files
22. bin/
23. gen/
25. # Gradle files
26. .gradle/
27. build/
28. /\*/build/
30. # Local configuration file (sdk path, etc)
31. local.properties
33. # Proguard folder generated by Eclipse
34. proguard/
36. # Log Files
37. \*.log
38. **<<<<<<<** **HEAD**

如何在README文件中显示图片呢，比如如下效果：



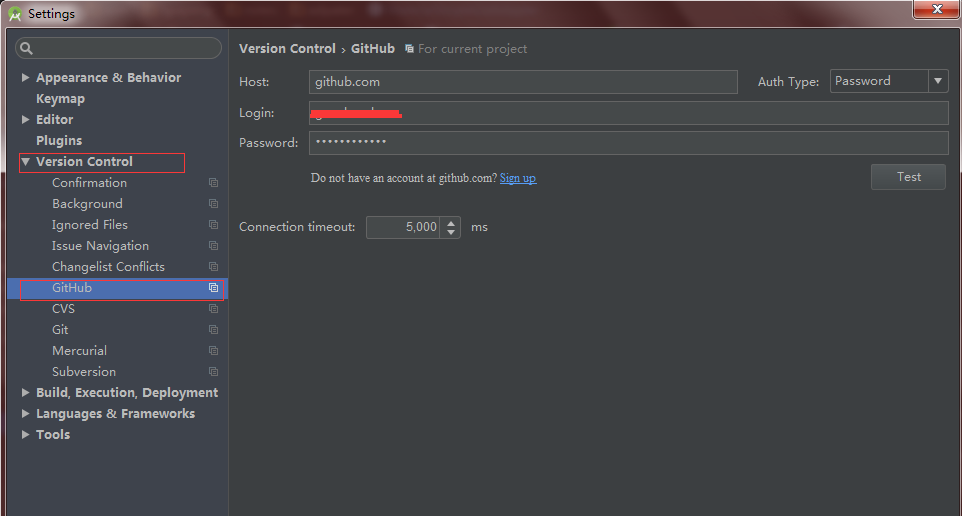
这里也给个小参考，在上传提交前，将截图保存在项目的某个文件夹下，然后在README文件中引用即可。（这个大家尽可参考其他作者开源项目中的README文件）



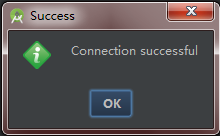
# 六、Studio中配置GitHub并下拉代码

那么如何将GitHub上将项目使用Studio下载到本地呢，下面也来总结记录下。

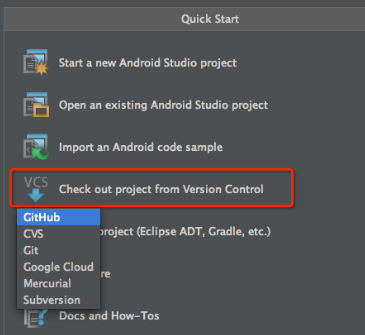
一、首先来配置Studio，打开 Settings 中 Version Control 下的GitHub，输入账号密码等信息。

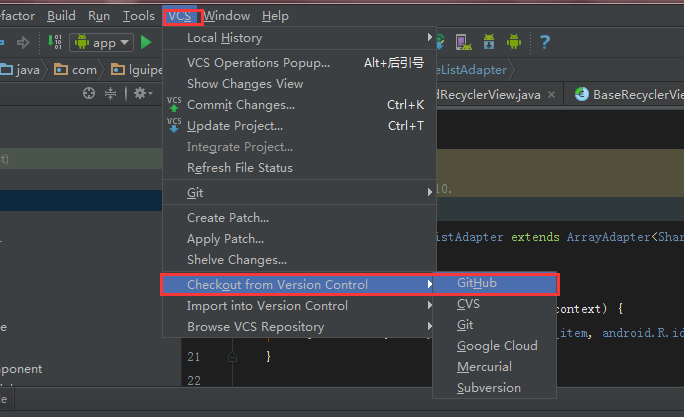


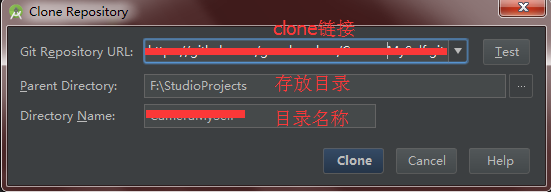
二、点击Test，可能需要再次确认密码，之后点击OK。



三、从首界面Check也好，从当前项目Check也好，操作还是挺简单的。







**错误总结：**

如果你在创建仓库的时候创建了.gitignore文件，或许在提交的时候会出现如下错误：[Updates were rejected because the remote contains work that you do not have](http://stackoverflow.com/questions/18328800/github-updates-were-rejected-because-the-remote-contains-work-that-you-do-not-h" \t "http://blog.csdn.net/gao_chun/article/details/_blank)

This happens if you initialized a new github repo with README and/or LICENSE file

git remote add origin [//your github url]

//pull those changes

git pull origin master

//now, push your work to your new repo

git push origin master

Now you will be able to push your repository to github. Basically, you have to merge those new initialized files with your work. git pull fetches and merges for you. You can also fetch and merge if that suits you.