# Git使用方法

**本地使用git（local repository）：**

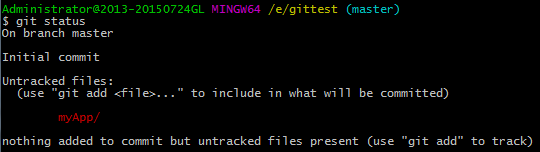
*1.用于添加提交。*

1. 首先在要传的文件夹下初始化一个本地git仓储。例如，在E:\gittest\myApp中，要上传myApp，在E:\gittest\下打开git命令行。初始化一个本地git仓储。



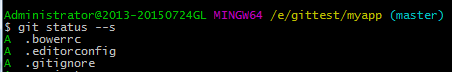
注: 就是在本地文件夹中添加了一个.git的文件夹，用于记录所有的项目变更信息。

1. 查看本地仓储的变更状态

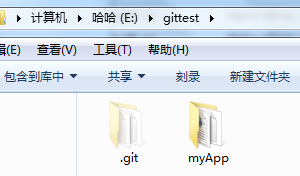


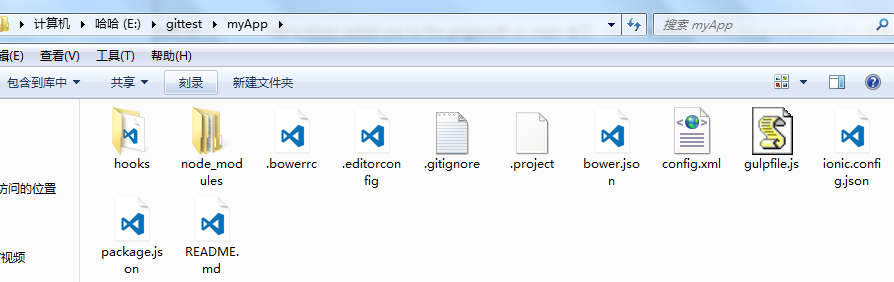
注：untracked files未被跟踪的文件，用于查看本地仓储的状态，第一次查看，显示的是没有被跟踪的文件。

git status –s 输出简要的变更日志。



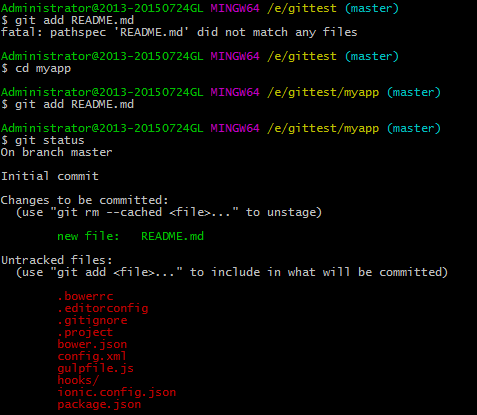
1. 添加本地暂存文件



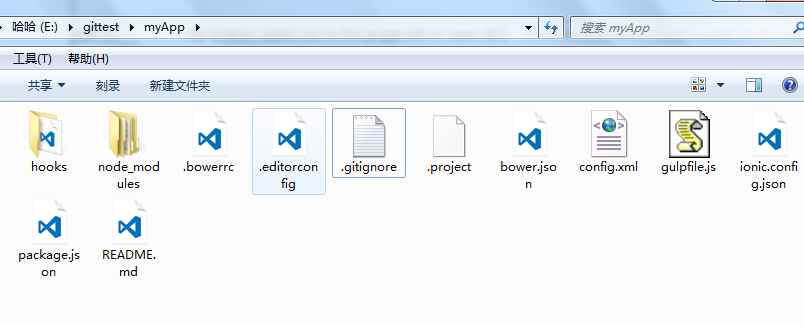


注：第一次用git add readme.md命令时，由于当前目录并没有readme.md这个文件，所以提示找不到任何匹配的文件。切换目录到myapp底下，重新git add readme.md，然后使用git status查看当前状态，绿色的是添加的文件，可以被commit，红色的是未被跟踪的文件，不能被commit。

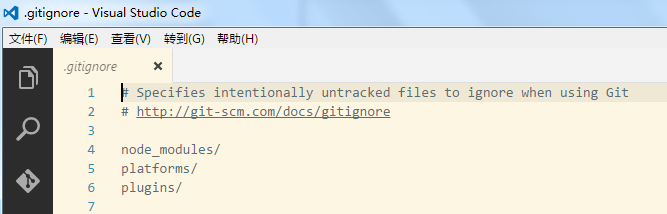
添加所有文件 git add .或者git add --all



1. 类似于node\_modules这种性质的文件是不应该被跟踪的，可以添加本地git忽略清单文件。



添加一个.gitignore文件，此文件用于说明忽略的文件有哪些。文件里直接写不需要被跟踪的文件，如下图所示。



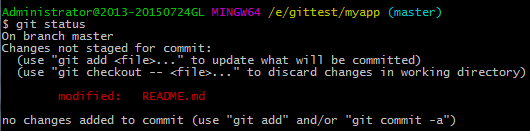
1. 提交被托管的文件变化到本地仓储

git commit：将本地的变化提交到本地的仓储文件夹归档。

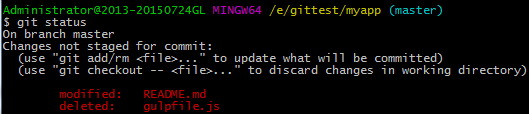
注：要记得每一次提交写日志。-m 后面加上日志内容



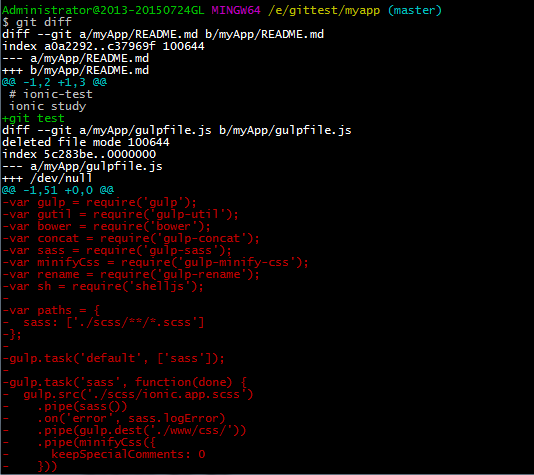
1. 当修改了提交的某一个文件时，git status可以看到：



再删除一个文件，可以看到



1. 可以使用git diff命令查看变更，即当前状态和版本库中状态的对比。

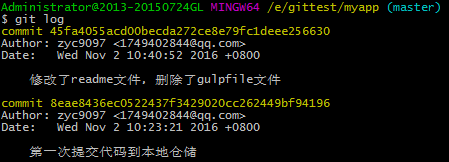


1. 然后可以再次添加并提交。



*2.查看日志，追溯版本*

1）查看提交日志



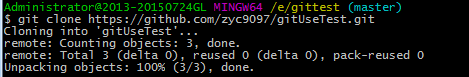
2） 追溯版本。哈希值前6位。



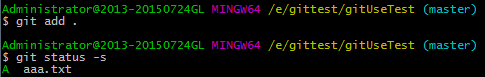
**Git服务器（remote repository）：**

*Git服务提供商：github。Git是一个工具，github是一个网站，提供git服务。*

1. 在github网站上create new repository。
2. 将新建的仓储clone到本地。



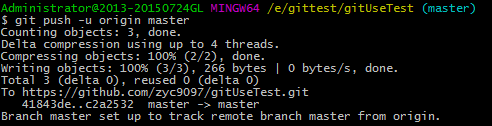
1. 添加，修改，删除文件，然后添加到本地仓储中。



1. 提交代码。



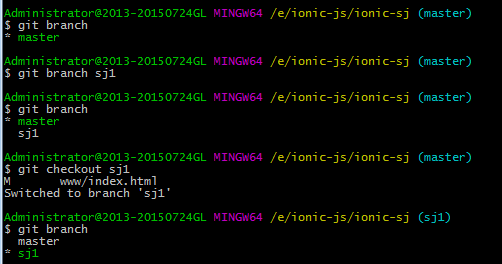
1. 推送到原仓储中。



1. 从remote仓储中pull,更新代码。

$git pull origin master

**git切换分支**

****

**基本上顺序是这样的：**

1 . 修改好了代码，先要提交

git commit -m “commit message"

2 . 然后有两种方法来把你的代码和远程仓库中的代码合并

a. git pull这样就直接把你本地仓库中的代码进行更新但问题是可能会有冲突(conflicts)，个人不推荐

b. 先git fetch origin（把远程仓库中origin最新代码取回），再git merge origin/master（把[本地代码](https://www.baidu.com/s?wd=%E6%9C%AC%E5%9C%B0%E4%BB%A3%E7%A0%81&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YkmWczm1bduWf3nW0YrjFh0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPWDLn1ckrHmsPjTdrjmsPHTz" \t "_blank)和已取得的远程仓库最新代码合并），如果你的改动和远程仓库中最新代码有冲突，会提示，再去一个一个解决冲突，最后再从1开始

3 . 如果没有冲突，git push origin master，把你的改动推送到远程仓库中