

Evan

Only let oneself become strong enough, good enough, can afford the life that you want to.

☰ 目录视图

☰ 摘要视图 | |  订阅

程序员，为什么写不好一份简历？ 征文 | 你会为 AI 转型么？ 赠书：7月大咖新书机器学习/Android/python

Git新手上路，让你快速掌握Git的基本使用

标签： [git](#) [github](#) [基本操作](#)

2016-03-24 12:01 400人阅读 评论(0) 收藏 举报

☰ 分类： [Git](#)

❗ 版权声明：本文为博主原创文章，未经博主允许不得转载。

github是一个基于git的代码托管平台，付费用户可以建私人仓库，我们一般的免费用户只能使用公共仓库，也就是代码要公开。这对于一般人来说公共仓库就已经足够了。

1.注册账户以及创建仓库

要想使用github第一步当然是注册github账号了。之后就可以创建仓库了（免费用户只能建公共仓库），Create a NewRepository，填好名称后Create，之后会出现一些仓库的配置信息，这也是一个Git的简单教程。

2.安装客户端tortoiseGit

github是服务端，要想在自己电脑上使用git我们还需要一个git客户端，我这里选用TortoiseGit，他给我们提供了图形界面的操作。在安装之前首先需要安装git，下载地址<http://msysgit.github.com/>，TortoiseGit下载地址：

<http://code.google.com/p/tortoisegit>

3.创建版本库

创建一个版本库非常简单，首先，选择一个合适的地方，创建一个空目录：

```
$ mkdir learngit    （目录可以自己选择:D/Mine）
```

```
$ cd learngit    （目录可以自己选择:D/Mine）
```

```
$ pwd  
/Users/michael/learngit
```

pwd命令用于显示当前目录。这个仓库位于/Users/michael/learngit。

如果你使用Windows系统，为了避免遇到各种莫名其妙的问题，请确保目录名（包括父目录）不包含中文。

第二步，通过git init命令把这个目录变成Git可以管理的仓库：

```
$ git init
```

Initialized empty Git repository in /Users/michael/learngit/.git/

瞬间Git就把仓库建好了，而且告诉你是一个空的仓库（empty Git repository），细心的读者可以发现当前目录下多了一个。

目录是Git来跟踪管理版本库的，没事千万不要手动修改这个目录里面的文件，不然改乱了，就把Git仓库给破坏了。

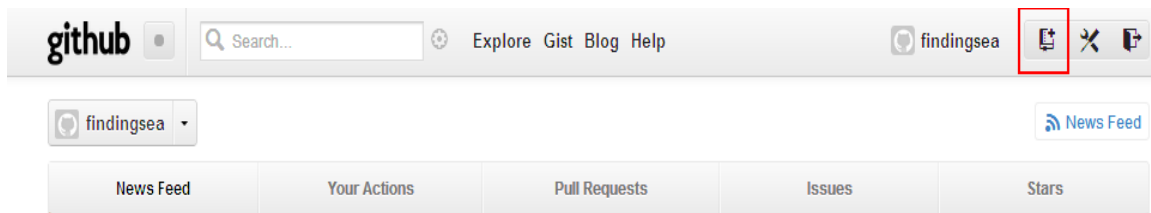
如果你没有看到.git目录，那是因为这个目录默认是隐藏的。

在现有仓库创建版本库：

在本地仓库里右键选择Git InitHere，会多出来一个.git文件夹，这就表示本地git创建成功。

右键GitBash进入git命令行，为了把本地的仓库传到github，还需要配置ssh key。

在github首页的右上角，点击红框中的Create New Repo。



进入新建仓库的界面

Owner: findingsea

Repository name:

Great repository names are short and memorable. Need inspiration? How about [tripping-octo-sansa](#).

Description (optional):

☒ Public Anyone can see this repository. You choose who can commit.

☐ Private You choose who can see and commit to this repository.

☐ Initialize this repository with a README
This will allow you to `git clone` the repository immediately.

Add .gitignore:

Create repository

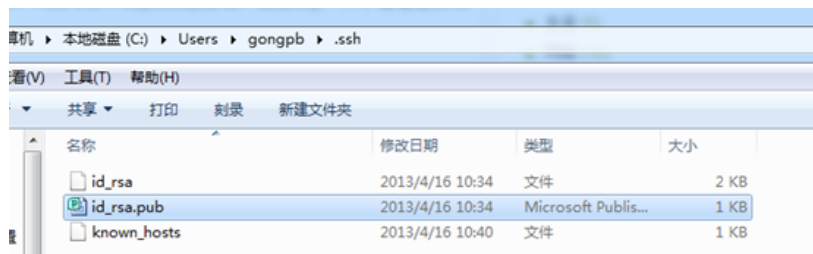
填一下仓库名称，Initialize this repository with a README是可选的，不过本人建议最好选上，可以在后面省一个步骤。填好之后，点Create repository就行了。

4.配置Git

（1）首先在本地创建ssh key；

```
$ ssh-keygen -t rsa -C "your_email@youremail.com"
```

后面的your_email@youremail.com改为你的邮箱，之后会要求确认路径和输入密码，我们这使用默认的一路回车就行。成功的话会在~/下生成.ssh文件夹，进去，打开id_rsa.pub，复制里面的key。回到github，进入AccountSettings，左边选择SSH Keys，Add SSH Key,title随便填，粘贴key。



(2) 为了验证是否成功，在gitbash下输入：

在初始化版本库之前，先要确认认证的公钥是否正确，如下：

```
ssh -T git@github.com
```

正确地结果如下：

```
Warning: Permanently added 'github.com,207.97.227.239' (RSA) to the list of known hosts.
```

```
Hi findingsea! You've successfully authenticated, but GitHub does not provide shell access.
```

会有一个Warning，不用理会

如果是第一次的会提示是否continue，输入yes就会看到：You've successfully authenticated, but GitHub does not provide shell access。这就表示已成功连上github。

(3) 接下来我们要做的就是将本地仓库传到github上去，在此之前还需要设置username和email，因为github每次commit都会记录他们。

[plain]

```
01. $ git config --global user.name "your name"
02. $ git config --global user.email "your_email@youremail.com"
```

(4) 接下来对库进行clone，如下：

```
git clone https://github.com/findingsea/myRepoForBlog.git
```

上面的地址可以在如下界面找到：




clone成功如下：

```
Cloning into 'myRepoForBlog'...
Warning: Permanently added 'github.com,207.97.227.239' (RSA) to the list of known hosts.
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (3/3), done.
```

后面的findingsea和myRepoForBlog表示你再github的用户名和刚才新建的仓库，加完之后进入.git，打开config，这里会多出一个remote “origin” 内容，这就是刚才添加的远程地址，也可以直接修改config来配置远程地址。

5.上传README.md文件

这个时候，我们的GitHub文件夹下就多了一个myRepoForBlog文件夹，进入文件夹目录，对仓库进行初始化，如果我们之前没有勾选创建

README，则要先创建README.md文件，不然上传文件会报错。如果在第一步就勾选过了  **Initialize this repository with a README** This will allow you to git clone the repository immr，则可以直接进入下一步。

```
git init
touch README.md
git add README.md
git commit -m 'first_commit'
git remote add origin https://github.com/findingsea/myRepoForBlog.git
git push origin master
```

6.push文件

创建完README.md后，就可以push了，代码类似。

```
git add .
git commit -m 'first_commit'
git remote add origin https://github.com/findingsea/myRepoForBlog.git
git push origin master
```

如果执行git remote add origin https://github.com/findingsea/myRepoForBlog.git，出现错误：

```
fatal: remote origin already exists
```

则执行以下语句：

```
git remote rm origin
```

再往后执行git remote add origin<https://github.com/findingsea/myRepoForBlog.git> 即可。

在执行git push origin master时，报错：

```
error:failed to push som refs to.....
```

则执行以下语句：

```
git pull origin master
```

先把远程服务器github上面的文件拉先来，再push 上去。

PS: .gitignore文件

.gitignore顾名思义就是告诉git需要忽略的文件，这是一个很重要并且很实用的文件。一般我们写完代码后会执行编译、调试等操作，这期间会产生很多中间文件和可执行文件，这些都不是代码文件，是不需要git来管理的。我们在gitstatus的时候会看到很多这样的文件，如果用git add -A来添加的话会把他们都加进去，而手动一个个添加的话也太麻烦了。这时我们就需要.gitignore了。比如一般c#的项目我的.gitignore是这样写的：

bin

.suo

obj

bin和obj是编译目录，里面都不是源代码，忽略；suo文件是vs2010的配置文件，不需要。这样你在gitstatus的时候就只会看到源代码文件了，就可以放心的git add -A了。

gitpush命令会将本地仓库推送到远程服务器。

git pull命令则相反。

修改完代码后，使用gitstatus可以查看文件的差别，使用git add 添加要commit的文件，也可以用git add -i来智能添加文件。之后gitcommit提交本次修改，git push上传到github。

顶 踩
0 0

- 上一篇 蓝桥杯 无穷分数
- 下一篇 照片EXIF信息的读取和改写的JAVA实现