1. Git和SVN的区别

SVN是集中式的版本控制系统，如图1.1所示。集中式的版本控制系统最重要的特征是有一台中央服务器，当需要下载代码时，从中央服务器download即可，修改完代码后，upload即可。



图1. 1 集中式的版本控制系统

Git是分布式的版本控制系统，如图1.2所示。也就是说，代码存在任何一台电脑上，当需要和另一台电脑进行交互时，只需要两台电脑进行代码的冲突处理即可。但通常情况下，也需要有一台中央服务器，但并不存储代码，只是为了方便两台电脑能够彼此找到对方。



图1. 2 分布式的版本控制系统

1. 为何平时都说要将代码传至github上

其实github也是一台简单的PC机，有我们代码的备份。

1. 安装git

Git官网：<https://git-scm.com/downloads>

国内镜像：<https://pan.baidu.com/s/1kU5OCOB#list/path=%2Fpub%2Fgit> （一个百度网盘的地址）

1. 创建版本库

初始化版本库，使其成为git可以管理的版本库。

1. git init

将文件装进版本库

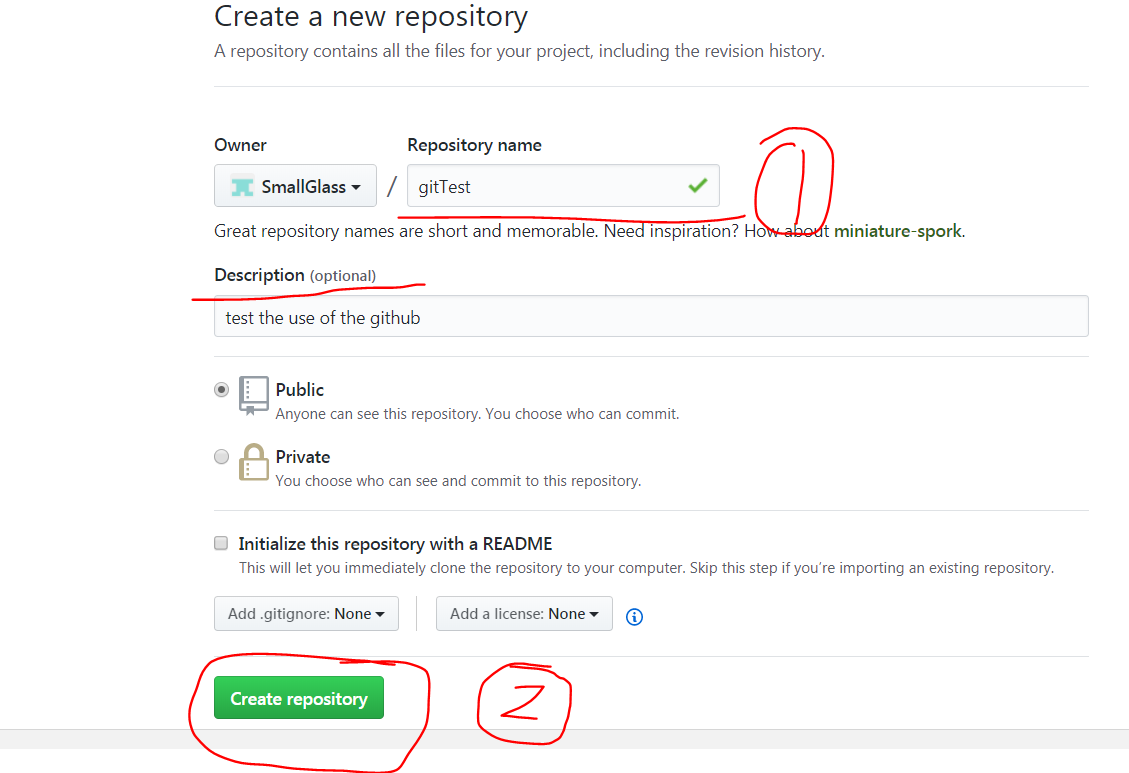
1. git add readme.txt

把文件提交到仓库。

1. git commit -m "wrote a readme file"

git的add操作，一次可以操作多个文件

1. $ git add file1.txt
2. $ git add file2.txt file3.txt
3. $ git commit -m "add 3 files."
4. 在github上创建远程仓库



1. SSH连接

Git使用的是SSH连接，目的是为了在每次提交代码的时候不用输入用户名和密码。实现方法是在本地生成SSH key，并将key加入到github服务器上。

首先，检查是否有SSH key。

6.1 打开Git Bash

6.2

1. ls -al ~/.ssh

如果没有key，生成之，最后一个参数是github邮箱。

1. ssh-keygen -t rsa -b 4096 -C "xinchaozhulang@126.com"

然后，一路enter即可。

6.3 将SSH key加入到 SSH-agent

启动ssh-agent

1. $ eval $(ssh-agent -s)

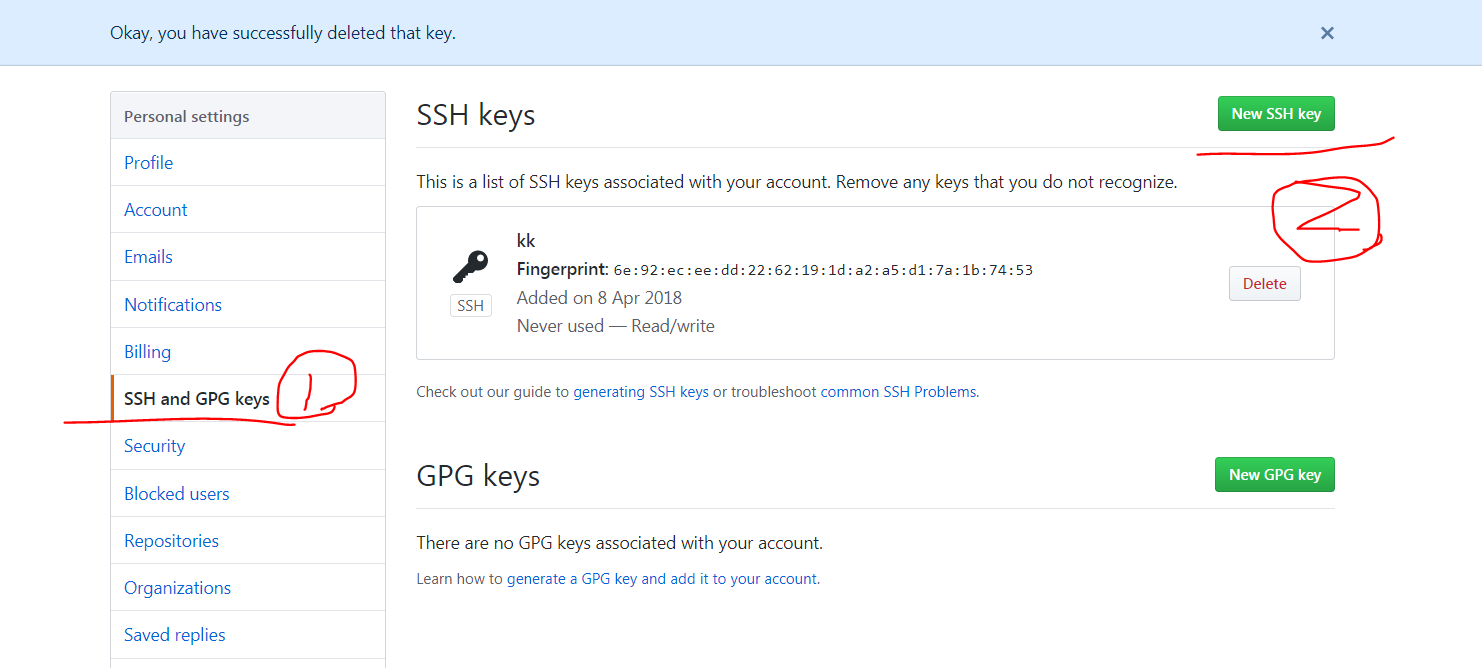
加入SSH-agent

1. $ ssh-add ~/.ssh/id\_rsa

6.4 将SSH key加入到github中

将SSH key加入到剪切板中。

1. clip< ~/.ssh/id\_rsa.pub
2. 在github里面的setting里面设置下就好。



1. 将github作为自己的远程仓库
2. $ git init
3. $ git add \*
4. $ git commit -m "test handle"
5. \*\*\* Please tell me who you are.
6. Run
7. git config --global user.email "you@example.com"
8. git config --global user.name "Your Name"
10. to set your account's default identity.
11. Omit --global to set the identity only in this repository.
13. fatal: unable to auto-detect email address (got 'Administrator@kk-PC.(none)')
15. Administrator@kk-PC MINGW64 ~/gitTest (master)
16. $ git config --global user.email "xinchaozhulang@126.com"
18. Administrator@kk-PC MINGW64 ~/gitTest (master)
19. $ git config --global user.name "kk"
21. $ git commit -m "test handle"
22. $ git remote add origin git@github.com:SmallGlass/gitTest.git
24. Administrator@kk-PC MINGW64 ~/gitTest (master)
25. $ git push -u origin master
26. 查看历史版本号
27. git log
28. 回退到上一个版本
29. git reset --hard HEAD^ readme.md
30. 或者 git reset –hard 版本号 readme.md
31. Git的工作区，暂存区



如果想丢掉工作区的内容，则可以直接

1. git checkout –file

如果文件已经到了暂存区，但还没有提交，则可以

1. git reset HEAD file
2. git checkout –file