

git：是一个版本管理工具

github：是一个免费的代码托管网站，相当于免费的 git 远程仓库

git 可以在本地玩，也可以链接github使用远程仓库

1: 有本地库，链接github 仓库

要关联一个远程库，使用命令 `git remote add origin git@server-name:path/repo-name.git`;

关联后，使用命令 `git push -u origin master` 第一次推送master分支的所有内容;

此后，每次本地提交后，只要有必要，就可以使用命令 `git push origin master` 推送最新修改;

2: 无本地库，直接克隆github 仓库

第1步：**创建SSH Key**。在用户主目录下，看看有没有.ssh目录，如果有，再看看这个目录下有没有 `id_rsa` 和 `id_rsa.pub` 这两个文件，如果已经有了，可直接跳到下一步。如果没有，打开Shell（Windows下打开Git Bash），创建SSH Key：

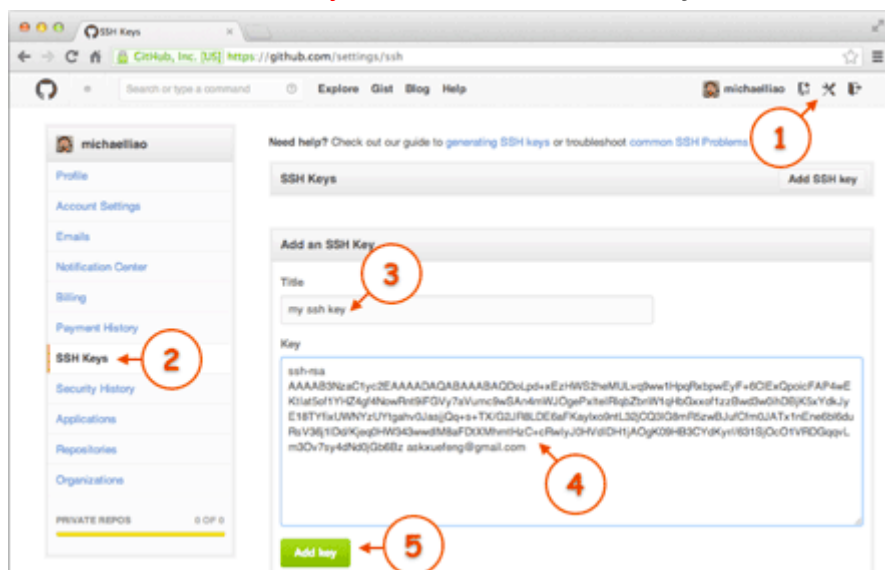
```
$ ssh-keygen -t rsa -C "youremail@example.com"
```

你需要把邮件地址换成你自己的邮件地址，然后一路回车，使用默认值即可，由于这个Key也不是用于军事目的，所以也无需设置密码。

如果一切顺利的话，可以在用户主目录里找到.ssh目录，里面有 `id_rsa` 和 `id_rsa.pub` 两个文件，这两个就是SSH Key的密钥对，`id_rsa` 是私钥，不能泄露出去，`id_rsa.pub` 是公钥，可以放心地告诉任何人。

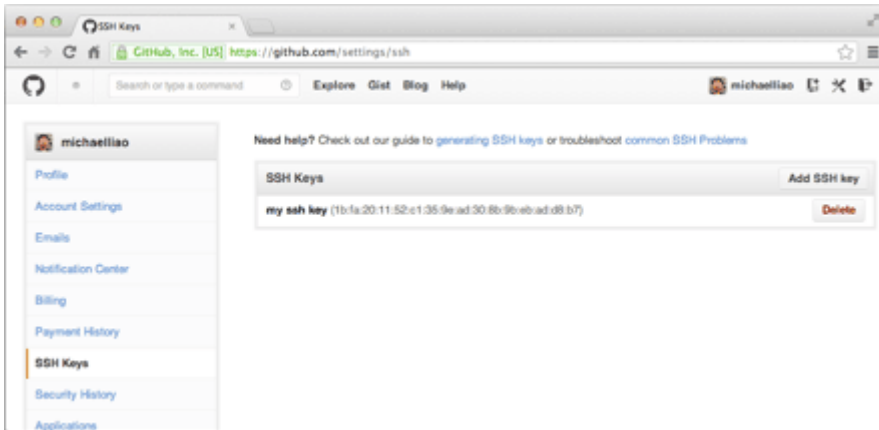
第2步：登陆GitHub，打开“Account settings”，“SSH Keys”页面：

然后，点“**Add SSH Key**”，填上任意Title，在Key文本框里粘贴 `id_rsa.pub` 文件的内容：



github-addkey-1

点“Add Key”，你就应该看到已经添加的Key：



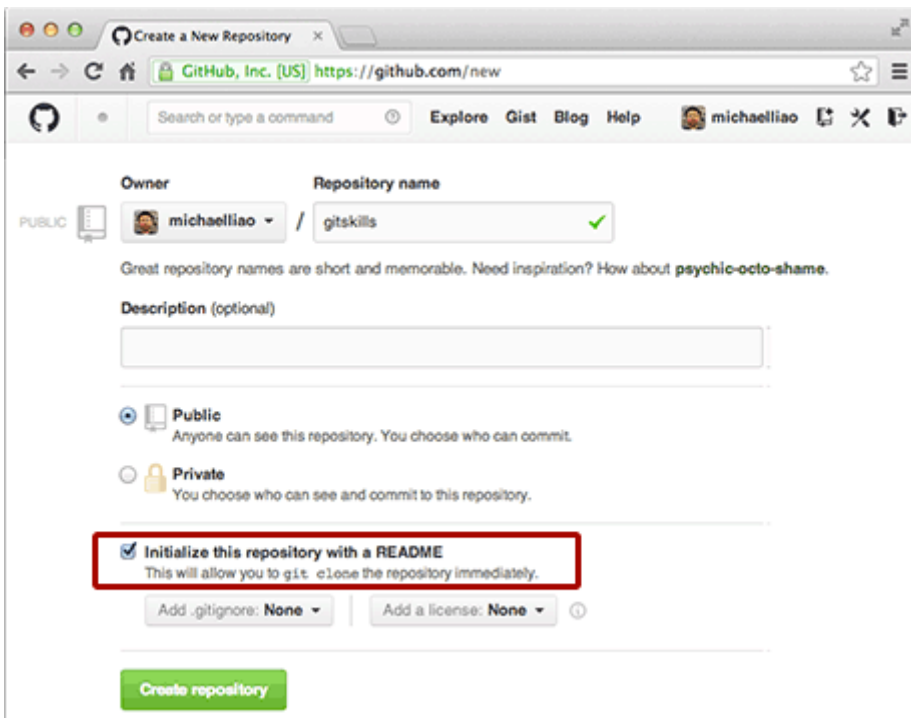
github-addkey-2

为什么GitHub需要SSH Key呢？因为GitHub需要识别出你推送的提交确实是你推送的，而不是别人冒充的，而Git支持SSH协议，所以，GitHub只要知道了你的公钥，就可以确认只有你自己才能推送。

当然，GitHub允许你添加多个Key。假定你有若干电脑，你一会儿在公司提交，一会儿在家里提交，只要把每台电脑的Key都添加到GitHub，就可以在每台电脑上往GitHub推送了。最后友情提示，在GitHub上免费托管的Git仓库，任何人都可以看到喔（但只有你自己才能改）。所以，不要把敏感信息放进去。

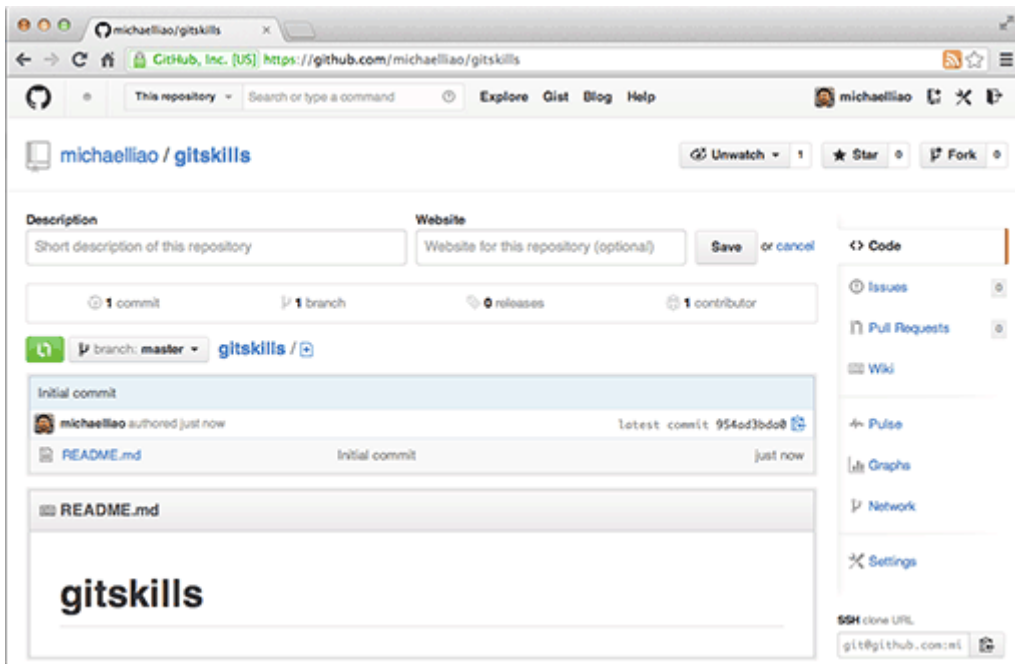
第三步：**克隆远程库**

首先，登陆GitHub，创建一个新的仓库，名字叫gitskills：



github-init-repo

我们勾选Initialize this repository with a README，这样GitHub会自动为我们创建一个README.md文件。创建完毕后，可以看到README.md文件：



github-init-repo-2

现在，远程库已经准备好了，下一步是用命令`git clone`克隆一个本地库：

```
$ git clone git@github.com:michaelliao/gitskills.git
```

```
Cloning into 'gitskills'...
```

```
remote: Counting objects: 3, done.
```

```
remote: Total 3 (delta 0), reused 0 (delta 0)
```

```
Receiving objects: 100% (3/3), done.
```

```
$ cd gitskills
```

```
$ ls
```

```
README.md
```

注意把Git库的地址换成你自己的，然后进入`gitskills`目录看看，已经有`README.md`文件了。

如果有多个人协作开发，那么每个人各自从远程克隆一份就可以了。

你也许还注意到，GitHub给出的地址不止一个，还可以用

`https://github.com/michaelliao/gitskills.git`这样的地址。实际上，Git支持多种协议，默认的`git://`使用ssh，但也可以使用`https`等其他协议。

使用`https`除了速度慢以外，还有个最大的麻烦是每次推送都必须输入口令，但是在某些只开放http端口的公司内部就无法使用`ssh`协议而只能用`https`。

第四步：提交

```
git push -u origin master
```