

远程仓库

第一步: 创建SSH Key。在用户目录下, 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"
```

把邮件地址换成自己的邮件地址, 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, 所以无需设置密码。

如果一切顺利的话, 可以在用户目录里找到 `.ssh` 目录, 里面有 `id_rsa` 和 `id_rsa.pub` 这两个文件, 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

第二步: 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` 文件内容:

为什么GitHub需要SSH Key呢?因为GitHub需要识别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, 就可以确定只有你自己才能推送。

当然, GitHub允许你添加多个Key。假定你有多台电脑, 你一会在公司提交, 一会在家提交, 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。

如果你不想让别人看到Git库, 有两个办法, 一个是交点保护费, 让GitHub把公开的仓库变成私有的, 这样别人就看不见了(不可读更不可写)。另一个办法是自己动手, 搭一个Git服务器, 因为是你自己的Git服务器, 所以别人也是看不见的。这个方法我们后面会讲到的, 相当简单, 公司内部开发必备。

确保你拥有一个GitHub账号后, 我们就即将开始远程仓库的学习。

“

小结

“有了远程仓库, 妈妈再也不用担心我的硬盘了。”