

一台电脑上的GIT同时使用两个GITHUB账户

1.需求:

有多个github账号时,想在git上同时使用,两者互不干扰。

2.解决方案

2.1 生成两个SSH key

这里以两个账号的配置为例,多于两个账号的情况配置和两个账号一样,所以学会了两个账号怎么配置了,再多账号也是一样可以顺利配置成功的。

根据你的Github账号,分别生成对应的key。为了方便举例,这里使用“one”和“two”两个账户。下同。

生成SSH key的具体命令如下:

windows用户:

- 1) 首先进入 C:\Users\用户名\.ssh/ 目录下,例如我的: C:\Users\panhu\.ssh\
- 2) 然后使用如下命令生成两个钥匙,中间一路回车

```
ssh-keygen -t rsa -f ./id_rsa_one -C "one@qq.com"
ssh-keygen -t rsa -f ./id_rsa_two -C "two@qq.com"
```

Linux用户:

- 1) 使用 cd ~/.ssh 切换工作目录
- 2) 后使用如下命令生成两个钥匙,中间一路回车

```
ssh-keygen -t rsa -f ~/.ssh/id_rsa_one -C "one@qq.com"
ssh-keygen -t rsa -f ~/.ssh/id_rsa_two -C "two@qq.com"
```

最后会在对应的文件夹下生成如下文件(该图为 windows 用户生成的),两份包含私钥和公钥的4个文件,后缀为.pub的文件为公钥文件。

名称	修改日期	类型	大小
id_rsa_one	2022/7/15 21:39	文件	3 KB
id_rsa_one.pub	2022/7/15 21:39	Microsoft Publis...	1 KB
id_rsa_two	2022/7/15 21:39	文件	3 KB
id_rsa_two.pub	2022/7/15 21:39	Microsoft Publis...	1 KB

特别注意: linux或mac用户一定要在 ~/.ssh 路径下运行命令行,不然生成的文件不会出现在当前目录,Windows用户则在“C:\Users\用户名.ssh”目录下运行命令行。

2.2 创建config文件并配置

1) 继续在.ssh目录下创建 config 文件

```
$ touch config // 创建 config
$ vim config   // 编辑 config
```

2) 在config文件中添加以下内容:

Linux用户的配置文件

```
# one(one@gmail.com)
# 修改点1, 给HOST取个别名(英文的), 比如 one.github.com
Host one.github.com
HostName github.com
PreferredAuthentications publickey
IdentityFile ~/.ssh/id_rsa_one
User one

# two(two@gmail.com)
# 修改点2, 给HOST取个别名(英文的), 比如 two.github.com
Host two.github.com
HostName github.com
PreferredAuthentications publickey
IdentityFile ~/.ssh/id_rsa_two
User two
```

Windows用户的配置文件

```
# one(one@gmail.com)
# 修改点1, 给HOST取个别名(英文的), 比如 one.github.com
Host one.github.com
HostName github.com
PreferredAuthentications publickey
# 修改点2, 将用户名修改为你对应的用户名
IdentityFile C:/Users/用户名/.ssh/id_rsa_one
User one

# two(two@gmail.com)
# 修改点3, 给HOST取个别名(英文的), 比如 two.github.com
Host two.github.com
HostName github.com
PreferredAuthentications publickey
# 修改点4, 将用户名修改为你对应的用户名, 比如我的是 panhu
IdentityFile C:/Users/用户名/.ssh/.ssh/id_rsa_two
User two
```

这里说明一下配置各字段的含义

```
Host myhost (这里是自定义的host简称, 以后连接远程服务器就可以用命令ssh myhost)
HostName 主机名可用ip也可以是域名(如:github.com或者bitbucket.org)
Port 服务器open-ssh端口(默认: 22,默认时一般不写此行)
PreferredAuthentications 配置登录时用什么权限认证--可设为publickey,password
publickey,keyboard-interactive等
IdentityFile 证书文件路径(如 ~/.ssh/id_rsa_*)
User 登录用户名(如: git)
```

每个账号单独配置一个Host, 每个Host要取一个别名, 一般为每个Host主要配置HostName和IdentityFile两个属性, 配置完保存即可。

Host的名字可以自定义名字, 不过这个会影响git相关命令, 例如: Host mygithub 这样定义的话, 使用命令git clone git@mygithub:PopFisher/AndroidRotateAnim.git, git@后面紧跟的名字改为mygithub

2.3 部署SSH key

这里以配置github为例, 分别登陆两个github账号, 进入Settings -> SSH and GPG keys, 点击"new SSH key", 把下面两个公钥(.pub文件)的内容分别添加到相应的github账号中, 其中Title为自定义的名字, Key为.pub文件的内容, 最后点击"Add SSH key"即可。

2.4 远程测试

输入以下命令进行测试

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

此处 "[one.github.com](#)" 和 "[two.github.com](#)" 为config文件中你配置的Host名字

运行命令后如果出现 "Hi xxxx! You've successfully authenticated, but GitHub does not provide shell access.", 其中"xxxx"为你github 配置的 User 名字, 这时恭喜你, 配置成功了~

2.5 使用

1) clone代码到本地

原来的写法为:

```
# 这里的 learngit 就是github上你要克隆的仓库的名字, "one的用户名就是步骤2.4运行命令后提示的
xxxx"
git clone git@github.com:one的用户名/learngit.git
```

现在的写法为:

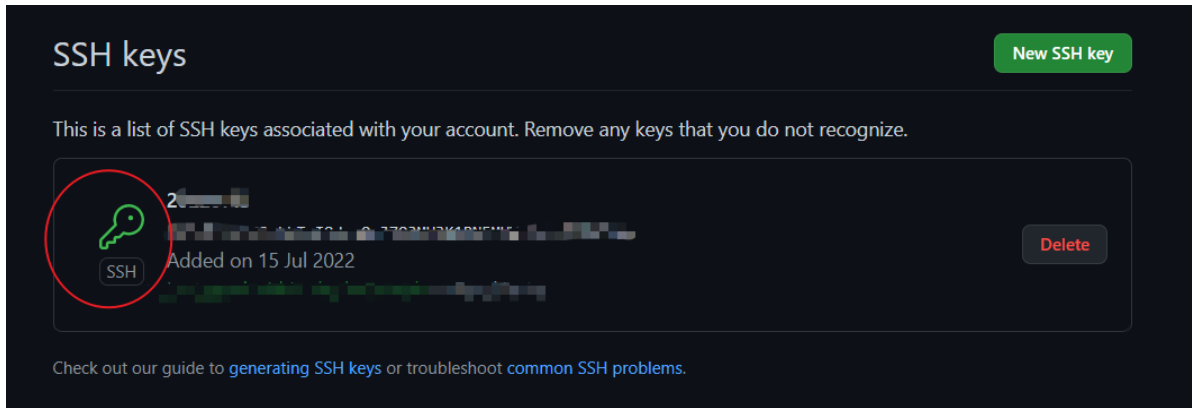
```
# 这里的 one.github.com 和 two.github.com 就是上面提到的给 HOST 取的别名
# one的用户名就是步骤2.4运行命令后提示的 xxxx,two的用户名同理
git clone git@one.github.com:one的用户名/learngit.git
git clone git@two.github.com:two的用户名/learngit.git
```

为仓库设置局部的用户名和邮箱:

```
# 取消全局 用户名/邮箱 配置
git config --global --unset user.name
git config --global --unset user.email

# 分别进入one账户和two账户克隆下来的仓库，单独为每个仓库设置 用户名/邮箱
git config user.name "one_name" ; git config user.email "one@qq.com"
git config user.name "two_name" ; git config user.email "two@qq.com"
```

上述都成功后，会发现钥匙会由灰变绿。



参考文献：

<https://www.freesion.com/article/3894842447/>