

# Git之同一台电脑连接多个远程仓库 - 简书

## 问题描述

有时候我们需要在同一台电脑上连接多个远程仓库，比如连接两个GitHub账号，那么需要两个条件。

1. 生成两对 私钥/公钥，并且密钥文件命名不能重复。
2. push 到remote时区分两个账户，推送到相应的仓库。

## 相应配置

1. 在MAC的终端中输入以下命令，查看密钥。

```
ls ~/.ssh
```

如果有 `id_rsa` 和 `id_rsa.pub`，说明已存在一对密钥/公钥。

2. 创建新的 密钥/公钥，并指定密钥名称，比如`id_rsa_x`（x为任意名称）

```
ssh-keygen -t rsa -f ~/.ssh/id_rsa_x -C "yourmail@xxx.com"
```

操作完成后，该目录会多出 `id_rsa_x` 和 `id_rsa_x.pub` 两个文件。

3. 在 `~/.ssh/` 文件夹下创建一个 `config` 文件

```
$ touch config$ vim config
```

编辑config文件，配置不同的仓库指向不同的密钥文件。

```
# 第一个账号，默认使用的账号Host github.comHostName github.comUser gitIdentityFile ~/.ssh/id_rsa# 第二个账号Host second.github.com # second为前缀名，可以任意设置HostName github.comUser gitIdentityFile ~/.ssh/id_rsa_x
```

## 原理分析

1. ssh 客户端是通过类似 `git@github.com:githubUserName/repName.git **` 的地址来识别使用本地的哪个私钥的，地址中的 User 是@前面的git， Host 是@后面的github.com。

2. 如果所有账号的 User 和 Host 都为 git 和 github.com，那么就只能使用一个私钥。所以对User 和 Host 进行配置，让每个账号使用自己的 Host，每个 Host 的域名做 CNAME 解析到 github.com，如上面配置中的Host second.github.com。

3. 配置了别名之后，新的地址就是`git@second.github.com:githubUserName/repName.git**`（在添加远程仓库时使用）。

这样 ssh 在连接时就可以区别不同的账号了。

4. 查看SSH 密钥的值，分别添加到对应的 GitHub 账户中

```
$ cat id_rsa.pub$ cat id_rsa_x.pub
```

把这两个值分别 copy 到 GitHub 账号中的 SSH keys 中保存。

5. 清空本地的 SSH 缓存, 添加新的 SSH 密钥 到 SSH agent中

```
$ ssh-add -D$ ssh-add id_rsa$ ssh-add id_rsa_x
```

最后确认一下新密钥已经添加成功

```
$ ssh-add -l
```

6. 测试 ssh 链接

```
ssh -T git@github.comssh -T git@second.github.com# xxx! You've successfully authenticated, but GitHub does not provide bash access.# 出现上述提示, 连接成功
```

7. 取消 git 全局用户名/邮箱的设置, 设置独立的 用户名/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# 进入项目文件夹, 单独设置每个repo 用户名/邮箱$ git config user.email "xxxx@xx.com"$ git config user.name "xxxx"
```

查看git项目的配置

```
git config --list
```

8. 命令行进入项目目录, 重建 origin (whatever 为相应项目地址)

```
$ git remote rm origin# 远程仓库地址, 注意Host名称$ git remote add origin git@second.github.com:githubUserName/repName.git$ git remote -v # 查看远程
```

10. 远程 push 测试

首先在 GitHub 上新建一个名为 testProj 的远程仓库, 然后再在本地建一个本地仓库。

```
$ cd ~/documnts$ mkdir testProj
```

1. 进入 testProj 文件夹, 创建 README.md文件

2. 初始化此文件夹为git

3. 添加并提交README.md到Git本地仓库

4. 添加远程仓库

5. 把README.md推送到远程仓库

```
$ cd testProj$ echo"# ludilala.github.io" >> README.md$ git init$ git add README.md$ git commit -m "first commit"# 如果前面已添加远程连接, 就无需再次添加$ git remote add origin https://github.com/ludilalaa/ludilala.github.io.git$ git push -u origin master
```

