

Mac 多个 Git 账户配置

概述

代码管理工具从流行的 CVS,到 SVN，到如今的 Git, 尤其是 Github 的存在，加之 git 在代码管理的种种优势，git 已经被越来越多的程序员喜爱。在日常开发中，我们可能面临着这样的尴尬：公司本身有 git 服务器，管理自己的代码。爱折腾的程序员，可在各大第三方代码管理平台，[github](#), [coding.net](#), [码云](#)，也有着自己的账号,对于 github,可能也有两个账号；这是就需要配置多个 SSH，为了方便在一台电脑上对多个 git 仓库提交代码，现在将其总结如下

具体操作

SSH key 可以让你在你的电脑和 Git 服务器之间建立安全的加密连接。一般你的 ssh key 存储在 这个目录下 `/Users/{username}/.ssh`

```
cd /Users/{accountname}/.ssh
```

查看.ssh 目录会有这样几个文件

```
-rw-r--r--@ 1 {username} staff 280B 8 30 15:12 config
-rw----- 1 {username} staff 3.2K 8 29 17:42 id_rsa
-rw-r--r-- 1 {username} staff 748B 8 29 17:42 id_rsa.pub
-rw----- 1 {username} staff 3.2K 8 29 14:31 id_rsa_github
-rw-r--r-- 1 {username} staff 748B 8 29 14:31 id_rsa_github.pub
-rw----- 1 {username} staff 1.6K 8 30 15:32 id_rsa_gitlab
-rw-r--r-- 1 {username} staff 397B 8 30 15:32 id_rsa_gitlab.pub
-rw-r--r--@ 1 {username} staff 1.5K 9 2 10:06 known_hosts
```

如果你的目录下，没有你的目录下没有这些文件，没关系，下面将详细讲解这些文件的作用，以及如何生成

由于笔者已经配置过自己 [github](#) 和公司 [gitlab](#)，所以会比较多，这里不必在意，现在我就[码云](#)为例，说明详细配置

生成 SSH key

```
ssh-keygen -t rsa -C "xxx@xx.com"
```

输入上述命令，点击回车后，会出现以下提示

```
Generating public/private rsa key pair.  
  
Enter file in which to save the key (/Users/{username}/.ssh/id_rsa):
```

这里是提示你生成的 **ssh key** 的存储路径和名称；如果你是简单的配置一个账号，直接回车，自动取默认路径和名称。如何你是为多个 **git** 账号配置私钥/公钥，你需要自己指定路径和名称，笔者用的是 **id_rsa_oschina**,名称可自己定义，具体如下：

```
Generating public/private rsa key pair.  
  
Enter file in which to save the key (/Users/{username}/.ssh/id_rsa):  
/Users/{username}/.ssh/id_rsa_oschina
```

继续回车

```
Enter passphrase (empty for no passphrase):  
  
Enter same passphrase again:
```

这里取默认值**空**,回车即可，最后出现如下图的回显，就说明你配置成功了

```
Your identification has been saved in /Users/{username}/.ssh/id_rsa_oschina.  
  
Your public key has been saved in /Users/{username}/.ssh/id_rsa_oschina.pub.  
  
The key fingerprint is:  
  
SHA256:lEmncZqtuXuHgZ4XtkVMkazLaTC5XgN0VLjYi3T8Fk8 xxx@xxx.com
```

The key's randomart image is:

+---[RSA 2048]-----+

| o o..=+o |

| . @. + o X|

| B..B o |

| . oB B . E|

| So X = + |

| ..* X o . |

| ..+ 0 o |

| o.* . |

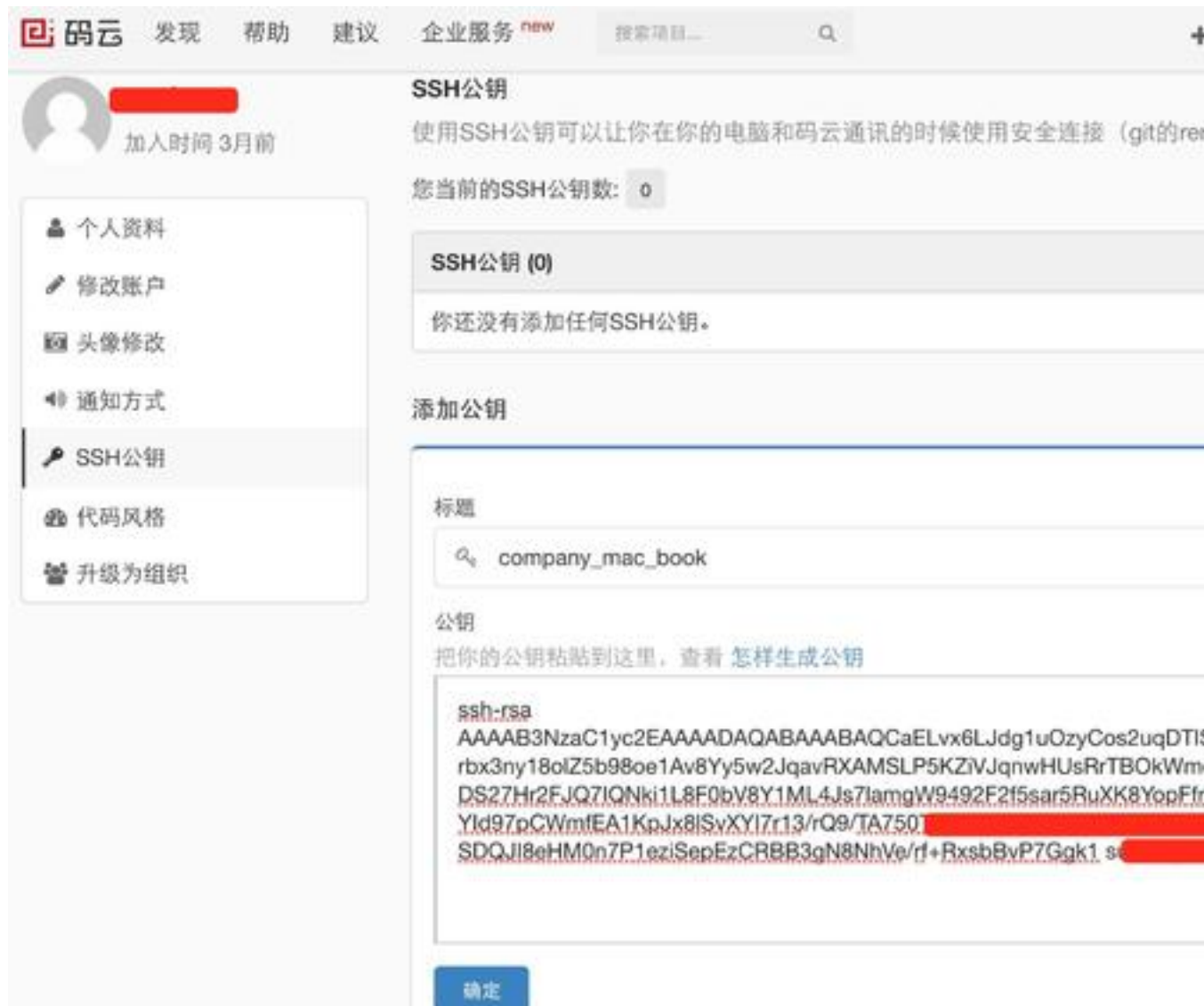
| .o . |

+-----[SHA256]-----+

将生成的公钥拷贝到剪贴板上，到 **git** 管理页面贴入即可

```
pbcopy < ~/.ssh/id_rsa_oschina.pub
```

其中 `id_rsa_oschina.pub`，就是你刚才输入的名称
oschina 配置：



oschina 图片

测试连接是否成功

```
ssh -T git@git.oschina.net
```

若返回

```
Welcome to Git@OSC, yourname!
```

则证明添加成功。

若返回

```
The authenticity of host 'git.oschina.net (116.211.167.152)' can't be
established.

ECDSA key fingerprint is SHA256:FQGC9Kn/eye1W8icdBgrQp+KkGYoFgbVr17bmjey0Wc.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added 'git.oschina.net,116.211.167.152' (ECDSA) to the
list of known hosts.

Permission denied (publickey).
```

则未成功，执行

```
ssh-add ~/.ssh/id_rsa_oschina
```

如果 ssh-add 不成功，可以直接进入下一步
如果成功再次重复执行这一步

至此是否已经配置完成，答案否

如果你在 github 等上面有两个账号，可能就会导致 git 提交失败，那么这里就需要配置~/.ssh/config 这个文件

打开 config 文件

```
open ~/.ssh/config
```

在文件中添加如下

```
#oschina

Host oschina （名称自定义）

    HostName git.oschina.net （服务器地址）

    user git （`git`@git.oschina.net 与 这里`git`名称是一致的，也可自定义，但不建议修改）
```

```
IdentityFile ~/.ssh/id_rsa_oschina (密钥存储路径)
```

QA

Q: 按照上述步骤执行了, 仍然 `Permission denied (publickey)`

试如下方案:

- 清空 `~/.ssh/known_hosts` 文件
- 执行 `ssh-add -D` (删除所有), 再次执行 `ssh-add -A` (添加所有)

Q: 此次配置成功后, 待下次重新启动电脑后, `git` 提交又提示 `Permission denied (publickey)`

- 每次重新执行 `ssh-add -A`
- 每次重新执行嫌麻烦, `ssh-add -A -K`, 添加到钥匙串内, 这样下次如果没有知道密钥, 那么则会自动取钥匙中存取的密钥

Q: 报错如下:

```
remote: Permission to username1/xxxx.git denied to username2.

fatal: unable to access 'https://github.com/username1/xxxx.git/': The
requested URL returned error: 403
```

- 原因可能是你有两个 `github` 账户, 以前使用一个登陆并管理代码, 切换另一个账号管理代码时, 钥匙串存储仍然是第一个用户的密码
- 执行 `git credential-osxkeychain erase`

然后进行测试

```
ssh -vT git@github.com
```

• 1

不通的仓库需要设置不同的用户名和邮箱, 如果之前有设置需要 `unset`

```
git config --global --unset user.name
```

```
git config --global --unset user.email
```

- 1
- 2

然后在不同的仓库下设置

```
git config user.name "yourname"
```

```
git config user.email "youremail"
```