

2019年04月26日 阅读 362

如何在本地管理和切换多个 github 账号?

前言

大多数的我们都会遇到这样一个问题：公司有一个 github 账户，这个是专门为工作准备的。而我们自己也有一个自己的私人 github 账户，有事没事做做自己的项目，写写自己感兴趣的东西。可是，我们怎么在电脑上把公司 github 切换到自己的私人github 账号上呢？

我公司内部建议用 smartgit 来精简 git 的操作，但是，我却没办法在上面切换成自己的账号，这意味着我必须在自己的电脑上使用自己的私人账号。oh, 这真是一个烦人的问题。

经过在 google 上查找资料 and 不断地试错，我终于成功的解决了这个问题。现在，让我来介绍以下这个是怎么解决的吧。

本质上，这只是一个平衡 git 和 ssh 配置的问题——实际上这并没有看上去那么糟糕。
—[Michael Herman](#)

操作过程

它的操作包括

- 创建ssh密钥
- 将密钥添加到 github 账户；
- 创建 config 文件，管理单独的 key
- 更新存储的 key

1. 创建 SSH 密钥

以我为例，我有两个 github 账户，一个工作用的，用户名是 yuanzhen-kooboo,; 另一个是私人的：huangyuanzhen。所以，我要创建两个密钥，每个账号一个：

操作为：

- 打开cmd;
- 依次输入命令:

```
cd ~/.ssh  
ssh-keygen -t rsa -C "1356409766@qq.com"  
ssh-keygen -t rsa -C "3083074260@qq.com"
```

cmd

- 当出现 "Enter file in which to save the key" 的提示时，将文件保存为 id_rsa_<>。在我的示例中，我将文件保存为 ~/.ssh/id_rsa_personal 和 ~/.ssh/id_rsa_company;

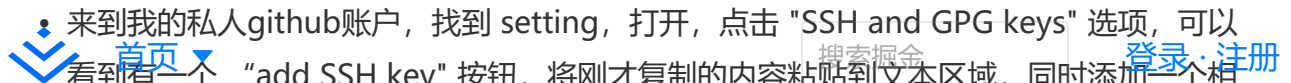
效果如图：

找到 C:\Users\huangyuanzhen\.ssh 下，可以看到，生成了下面四个文件：

- id_rsa_personal
- id_rsa_personal.pub
- id_rsa_company
- id_rsa_company.pub

2. 将新密钥绑定到 github 账号

- 用记事本打开 id_rsa_personal.pub 文件，全选复制;

- 
- 来到我的私人github账户，找到 setting，打开，点击 "SSH and GPG keys" 选项，可以看到有一个 "add SSH key" 按钮，将刚才复制的内容粘贴到文本区域，同时添加一个相关标题；成功之后是这个样子的：

- 相对的，在其他账户上重复相对应的操作；以我的为例，则是把 id_rsa_company.pub 的内容粘贴到我工作账户 yuanzhen-kooboo 的 ssh 上；

3. 创建一个配置 config 文件来管理 key

在 ~/.ssh/ 目录下创建一个 config 文件

```
echo test>config
```

找到这个文件，并用编辑器打开(我的是 vscode)，然后将下面内容写入文件，保存：

```
# huangyuanzhen
Host personal
    HostName github.com
    User git
    IdentityFile ~/.ssh/id_rsa_personal

# yuanzhen-kooboo
Host company
    HostName github.com
    User git
    IdentityFile ~/.ssh/id_rsa_work
```

这里我们的主机名称不是 github.com，而是将其命名为 personal 和 company。不同之处在于，我们现在附加了之前创建的新标识文件: id_rsa_<>;

4. 更新存储的key

在更新存储之前，我们要先检查一下本地的 OpenSSH 服务有没有开启。不然会出错。

开启 ssh 服务的流程为:



[首页](#) ▼

搜索掘金

[登录](#) · [注册](#)

1. 设置 → 管理可选功能 → 添加功能 → [OpenSSH服务器]
2. 计算机管理 → 服务和应用程序 → 服务 → OpenSSH Authentication Agent&OpenSSH Server → 右击

启动之后看到的是这样子的:

开始 SSH 服务之后, 我们就可以使用ssh 命令了。

清除当前存储的认证:

```
C:\Users\huangyuanzhen>ssh-add -D
// All identities removed.
```

cmd

增加新的keys:

```
C:\Users\huangyuanzhen\.ssh>ssh-add id_rsa_company
Identity added: id_rsa_company (id_rsa_company)

C:\Users\huangyuanzhen\.ssh>ssh-add id_rsa_personal
Identity added: id_rsa_personal (id_rsa_personal)
```

cmd

验证一下! github 是否能识别到这些 keys; 在cmd中输入:

```
ssh -T personal
```

cmd

可以看到 "Hi huangyuanzhen! You've successfully authenticated, but GitHub does not provide shell access." 的提示语。这表明, github 能识别这些 keys 了。Cool !

5. 测试 clone 和 push

测试 git clone

以我的私人账户为例，我想把 huangyuanzhen 账号上的 Look-Thinking 仓库克隆到本地，然后操作。

在cmd 上输入：

```
git clone git@personal:huangyuanzhen/Look-Thinking.git
```

cmd

可以看到可以成功把该仓库克隆过来：

如果要操作仓库，执行命令 "`cd Look-Thinking → code .`"就可以操作了。是不是感觉非常方便？

测试 git push

还是以在我的私人 github 账号 huangyuanzhen 上操作为例。在 账号上创建 work-test 仓库；然后在本地创建 test 文件夹：

```
E:\mkdir test
E:\cd test
E:\test>echo test>readme.md
```

cmd

创建好 readme.md 文件后,将其 push 到 github ；

```
git init
git add .
git commit -am "first commit"
git remote add origin git@personal:huangyuanzhen/test.git
git push -u origin master
```

cmd

将文件push成功之后是这样的：

可以在github账户上看到在本地对 readme.md的操作结果。git pull 同理。

我们怎样用命令行切换账号呢？

如果之前已经用 git remote add 和远程 仓库建立过连接，我们需要先清除当前连接，不然会报错：“fatal: remote origin already exists.”；清除当前连接之后，然后连接新的账号里的仓库，如：

```
git remote rm origin
git remote git add origin git@company:yuanzhen-kooboo/work-test.git
```

cmd

这里，我们就断开了之前和 huangyuanzhen 账号的连接，然后切换到了 yuanzhen-kooboo 账号。

这样配置好之后，不需要再用 smartgit 等一些辅助工具，直接在终端使用命令行操作，我感觉十分方便了呢！动手试试吧。

注意：上面命令行是在windows上操作，如果是在别的操作系统上，直接换成对应的操作命令即可。

资料

- [Managing Multiple Github Accounts](#)
- [Quick Tip: How to Work with GitHub and Multiple Accounts](#)
- [图解 -- Win10 OpenSSH](#)

关注下面的标签，发现更多相似文章

GitHub



首页 ▾

搜索掘金

[登录](#) · [注册](#)

安装掘金浏览器插件

打开新标签页发现好内容，掘金、GitHub、Dribbble、ProductHunt 等站点内容轻松获取。快来安装掘金浏览器插件获取高质量内容吧！

评论

输入评论...