

本地与GitHub关联并进行Git管理的两种方法

0.准备工作

GitHub连接已有仓库时使用的认证方式是SSH公开密钥

- 创建一个用于认证的SSH Key并将其添加至GitHub

1. 首先检查是否已经为计算机生成过SSH密钥，在Git bash命令行下先后输入以下两条命令：`cd ~/.ssh` 然后 `ls` 若没有则显示：

```
$ cd ~/.ssh
bash: cd: /c/Users/zdy/.ssh: No such file or directory
```

或直接一条命令：`cat ~/.ssh/id_rsa.pub`，若没有则显示：

```
$ cat ~/.ssh/id_rsa.pub
cat: /c/Users/zdy/.ssh/id_rsa.pub: No such file or directory
```

2. 若没有生成过，使用下列命令生成：`ssh-keygen -t rsa -C`
`"your_email@example.com"` (邮箱要用GitHub注册的邮箱!!!)

```
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/zdy/.ssh/id_rsa):
Created directory '/c/Users/zdy/.ssh'.
```

回车一次输入密码，再次回车确认密码：

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

最后生成的结果是：

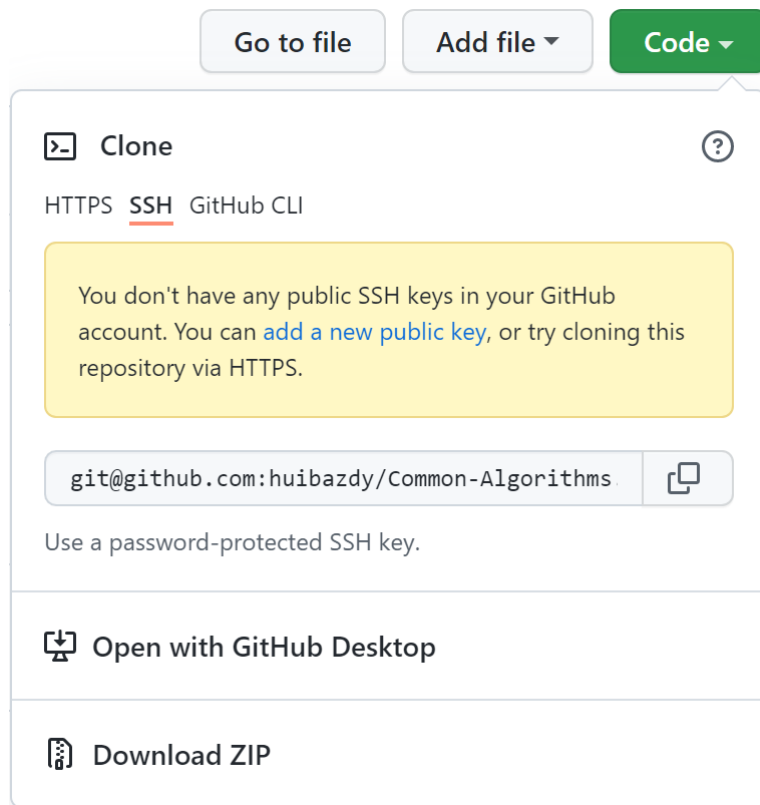
```
Your identification has been saved in /c/Users/zdy/.ssh/id_rsa
Your public key has been saved in /c/Users/zdy/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:A4xQUNSAfi6dSe9fwjTk5/9wq6yWZNxCN7SUetyhLY zhangdaoyuan1@xiaomi.com
The key's randomart image is:
+---[RSA 3072]-----+
|  o*=o      .      |
|  .. o.      o .    |
|  ... o . + o      |
|  . o .  o = *      |
|  . . . .S.E +      |
|  . . o = +o =      |
|  o . o + +++      |
|  .      . o.+o+     |
|  .+=.o.o.o.        |
+---[SHA256]-----+
```

其中 `id_rsa` 是私有密钥，`id_rsa.pub` 是共有密钥。可以在路径 `C:\Users\zdy.ssh`中看到生成的两个文件

名称	修改日期	类型	大小
 id_rsa	2022/3/28 13:54	文件	3 KB
 id_rsa.pub	2022/3/28 13:54	Microsoft Publisher...	1 KB

- 在GitHub中添加公有密钥

在GitHub中打开一个新建的仓库，找到配置ssh的位置，你会看到要求你添加共有密钥的提示：



点击提示的添加新的个公开密钥的链接，复制之前生成密钥路径下id_rsa.pub中的文本内容并粘贴至下图2处：（步骤1可以省略，会默认以邮箱名命名该ssh）

SSH keys / Add new

1

Title

company_laptop_id_rsa_public

2

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQGDQU8q35eVYf4FVcLMI7v8TiCna926A7xujN9UPEvQp/7iHakZJMoH4i6pvxDML
3//2krZwTONzwpHKfh4/wGOdt5RyW7+TqRmCNsxoN0g7Vxlod1t3KPL+pvzB0etwDr+eAmbufz+uTcNmDZckRY2+n
AK4Cw2o/Hw372x6qtXdN2Xli+LTrR+uQvHDom8O24oTXzJHNaiCmJIKRik+rqhbkdje6jt7I0DMJT/clXv0oobUsqPzMHR
bt52eJTRUtZFEKIXPpv7cbWXC5jVOr7y5IArx+6aRc5J4jCYASX+KCRH76hrBerV7Qd6AKNQAYVnh4pWyBg96f3qPhJ712
Q/UR7m5UBQqjQwIWELWW+WqyD8V78rCPO2hhYOLh1M9KS9pNuZtxV16UWW3ys2EZC1f5mlDn2TjwxVDFJkiTFTCP
NMHbPWfzyM95Pe6zpJu2wgPoX2mHiGZFvuvPMLbEaSzWDhAsFMhWopigujGh6gBBzN8AdTXFThphaquv8=
zhangdaoyuan1@xiaomi.com
```

3

Add SSH key

添加结束后可以在个人账户设置页面中看到以下内容：

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

SSH and GPG keys

Organizations

Moderation

Code, planning, and automation

Repositories

Packages

Pages

Saved replies

Security

Code security and analysis

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.



company_laptop_id_rsa_public
SHA256:AAxQUNSAf116d5e9fwjTK5/9wq6yuzIXCN75UetyhLY
Added on 28 Mar 2022
Never used — Read/write

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

GPG keys

New GPG key

There are no GPG keys associated with your account.

Learn how to [generate a GPG key](#) and [add it to your account](#).

Vigilant mode

☐ Flag unsigned commits as unverified

This will include any commit attributed to your account but not signed with your GPG or S/MIME key.
Note that this will include your existing unsigned commits.

[Learn about vigilant mode.](#)

公钥文件id_rsa.pub中的文本内容可以在Git bash命令行中以命令：`cat`

`~/.ssh/id_rsa.pub` 查看。在添加SSH Key成功后，创建GitHub账号的邮箱中会受到一封已添加新公钥的邮件。

- 完成上述设置后，本地计算机就可以利用私钥与GitHub进行认证通信了，现在进行验证：

键入命令 `ssh -T git@github.com`

```
zdy@SK-20210902RBPT MINGW64 ~ (master)
$ ssh -T git@github.com
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqu.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? |
```

输入 `yes` 并回车，提示警告信息

```
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Enter passphrase for key '/c/Users/zdy/.ssh/id_rsa':
```

输入之前设置的认证密码并回车，完成连接

```
Hi huibazdy! You've successfully authenticated, but GitHub does not provide shell access.
```

1. 先本地，后GitHub

- 新建一个本地仓库

```
zdy@SK-20210902RBPT MINGW64 /E/GitHub
$ mkdir LocalTest

zdy@SK-20210902RBPT MINGW64 /E/GitHub
$ cd LocalTest/

zdy@SK-20210902RBPT MINGW64 /E/GitHub/LocalTest
$ pwd
/E/GitHub/LocalTest
```

- 使本地仓库受Git管理

```
zdy@SK-20210902RBPT MINGW64 /E/GitHub/LocalTest
$ git init
Initialized empty Git repository in E:/GitHub/LocalTest/.git/

zdy@SK-20210902RBPT MINGW64 /E/GitHub/LocalTest (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

- 向仓库中添加文件或修改

```
zdy@SK-20210902RBPT MINGW64 /E/GitHub/LocalTest (master)
$ vim hello.cpp

zdy@SK-20210902RBPT MINGW64 /E/GitHub/LocalTest (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    hello.cpp

nothing added to commit but untracked files present (use "git add" to track)

zdy@SK-20210902RBPT MINGW64 /E/GitHub/LocalTest (master)
```

- 提交修改到暂存区

```
zdy@SK-20210902RBPT MINGW64 /E/GitHub/LocalTest (master)
$ git add hello.cpp

zdy@SK-20210902RBPT MINGW64 /E/GitHub/LocalTest (master)
$ git status
On branch master

No commits yet

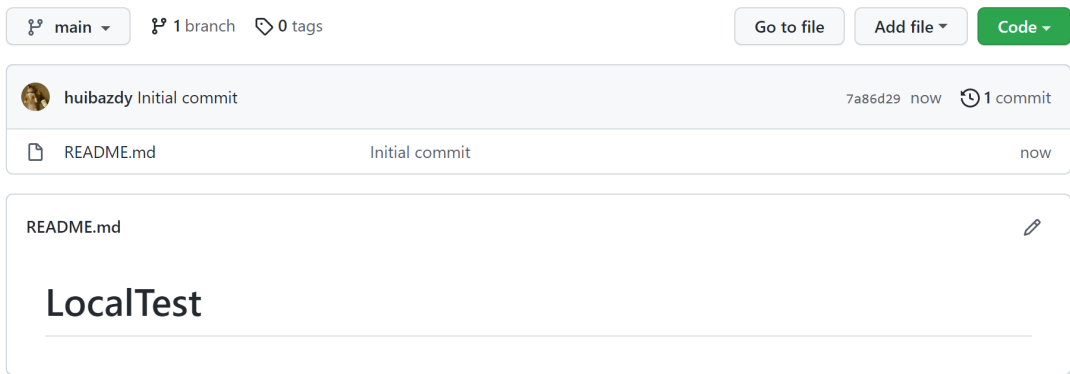
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   hello.cpp
```

- 提交到本地仓库（此处采取提交简单修改描述的方式）

```
zdy@SK-20210902RBPT MINGW64 /E/GitHub/LocalTest (master)
$ git commit -m "first commit"
[master (root-commit) a9cb790] first commit
1 file changed, 8 insertions(+)
create mode 100644 hello.cpp
```

- 推送到远程仓库（此处指的是GitHub）

1. 在GitHub上创建一个同名仓库作为本地仓库的远程仓库（最好不要勾选with a README)



2. 添加远程仓库—— `git remote add`

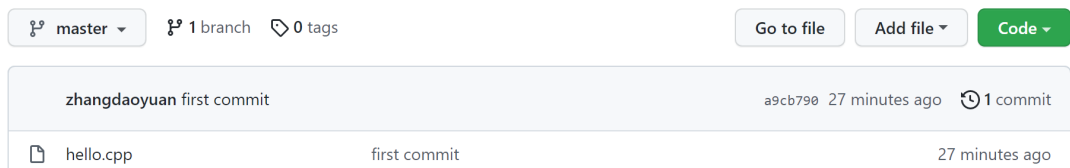
```
zdy@SK-20210902RBPT MINGW64 /E/GitHub/LocalTest (master)
$ git remote add origin git@github.com:huibazdy/LocalTest.git
```

3. 推送至远程仓库—— `git push`

第一次由于远程仓库为空，需要在 `git push` 后加上 `-u` 参数。其他分支也需要加上这个参数。

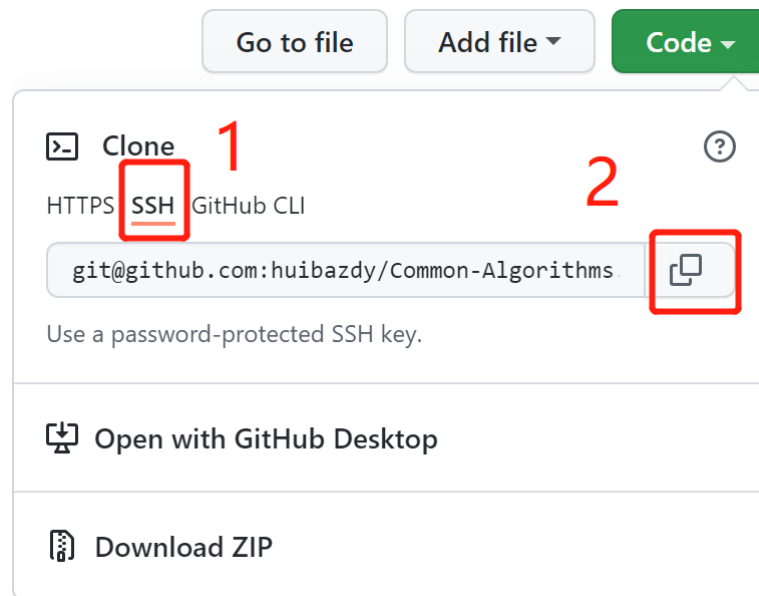
```
zdy@SK-20210902RBPT MINGW64 /E/GitHub/LocalTest (master)
$ git push -u origin master
Enter passphrase for key '/c/Users/zdy/.ssh/id_rsa':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 297 bytes | 297.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/huibazdy/LocalTest/pull/new/master
remote:
To github.com:huibazdy/LocalTest.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

之后可以在仓库中看到添加的hello.cpp文件：



2. 先GitHub，后本地

- 在新建的仓库后复制仓库的ssh链接



- 在Git bash命令行中切换到想要保存仓库的路径下，使用 `git clone` 命令克隆这个仓库到本地

```
zdy@SK-20210902RBPT MINGW64 /E/GitHub
$ git clone git@github.com:huibazdy/Common-Algorithms.git
Cloning into 'Common-Algorithms'...
Enter passphrase for key '/c/Users/zdy/.ssh/id_rsa':
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

- 在路径下查看克隆下来的仓库

E:\GitHub\Common-Algorithms				
名称	修改日期	类型	大小	
.git	2022/3/28 15:10	文件夹		
README.md	2022/3/28 15:10	Markdown File	1 KB	

- 切换到仓库路径下，用 `git status` 查看仓库状态

```
zdy@SK-20210902RBPT MINGW64 /E/GitHub
$ cd Common-Algorithms/

zdy@SK-20210902RBPT MINGW64 /E/GitHub/Common-Algorithms (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

- 在仓库路径下用vim新建一个hello.c文件，并再次查看仓库状态可以看到仓库存在 Untracked files

```
zdy@SK-20210902RBPT MINGW64 /E/GitHub/Common-Algorithms (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
      hello.c

nothing added to commit but untracked files present (use "git add" to track)
```

- **提交相关修改**（此时的修改是新建的文件hello.c）到仓库Common-Algorithms中

1. `git add` --提交到暂存区
2. `git commit` --提交到本地仓库（使文件变得可追踪）

```
zdy@SK-20210902RBPT MINGW64 /E/GitHub/Common-Algorithms (main)
$ git add hello.c

zdy@SK-20210902RBPT MINGW64 /E/GitHub/Common-Algorithms (main)
$ git commit
```

3. 在提交后，弹出的vim页面编辑该次提交的详细修改信息，保存并退出

[illegible]

```
zdy@SK-20210902RBPT MINGW64 /E/GitHub/Common-Algorithms (main)
$ git commit
[main 0ad0a41] first commit
1 file changed, 7 insertions(+)
create mode 100644 hello.c
```

- #### 4. git log --查看仓库提交历史

```
zdy@SK-20210902RBPT MINGW64 /E/GitHub/Common-Algorithms (main)
$ git log
commit 0ad0a4150a88a486c5972a1cfd6cfa27b15e5bf6 (HEAD -> main)
Author: zhangdaoyuan <zhangdaoyuan1@xiaomi.com>
Date: Mon Mar 28 15:26:34 2022 +0800

    first commit

    this is the first commit.

commit 83d2e793482d12ed693f9788e6aa2e7d41f05682 (origin/main, origin/HEAD)
Author: huiba_450_zdy <57480678+huibazdy@users.noreply.github.com>
Date: Mon Mar 28 11:18:29 2022 +0800

    Initial commit
```

- ## 5. git push --推到远程仓库 (此处指的是GitHub)

```
zdy@SK-20210902RBPT MINGW64 /E/GitHub/Common-Algorithms (main)
$ git push
Enter passphrase for key '/c/Users/zdy/.ssh/id_rsa':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 355 bytes | 355.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:huibazdy/Common-Algorithms.git
83d2e79..0ad0a41  main -> main
```

可以观察到GitHub仓库中增加了hello.c文件：

zhangdaoyuan first commit ...			0ad0a41 1 hour ago	🕒 2 commits
📄	README.md	Initial commit	6 hours ago	
📄	hello.c	first commit	1 hour ago	

3. 用到的Git命令小结

- git init
- git add
- git commit
- git status
- git remote add
- git push
- git log

4. 其他常用Git命令

- git branch
- git checkout
- git rm
- git reset