

我们以github为例，介绍本地仓库如何与github上仓库联动。

github官网: <https://github.com/>

值得注意的是，国内对GitHub限速，除魔法上网之外的方法有时候会出现短暂掉线，连接不上GitHub的情况，这是正常的，多刷新几次，或者等一会就可以了。

(魔法上网大家就各显神通吧，没法说)

首先注册一个账户，略。

## 一、SSH配置

为了让本地仓库和github账号上的仓库之间建立连接，我们需要在本地配置ssh-key。

ssh简要来说是一种常见的安全连接协议。

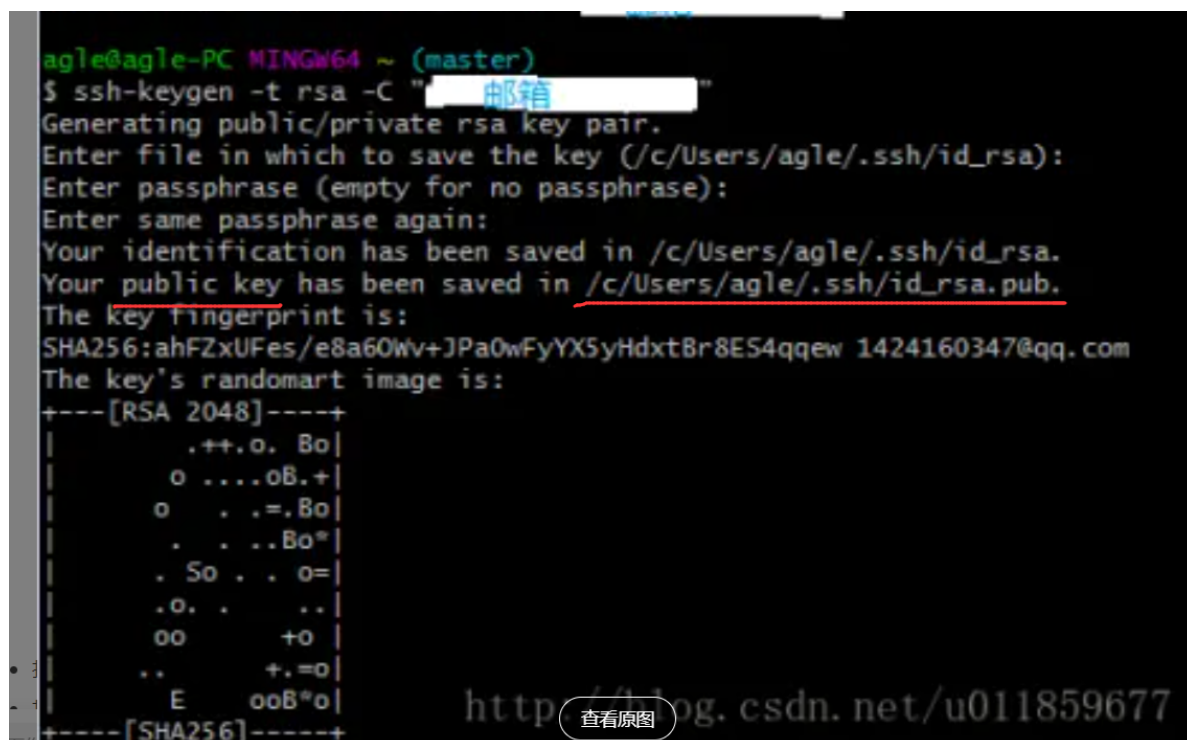
我们在任意位置右击 -> Open Git Bash here, 输入命令。

```
ssh-keygen -t rsa -C "你的账户邮箱（写全，包括后面@...com）"
```

(-t rsa 指的是采用RSA方式加密)

如果你了解ssh，请在下面的弹框中自行配置passphrase，否则不输入一路回车。

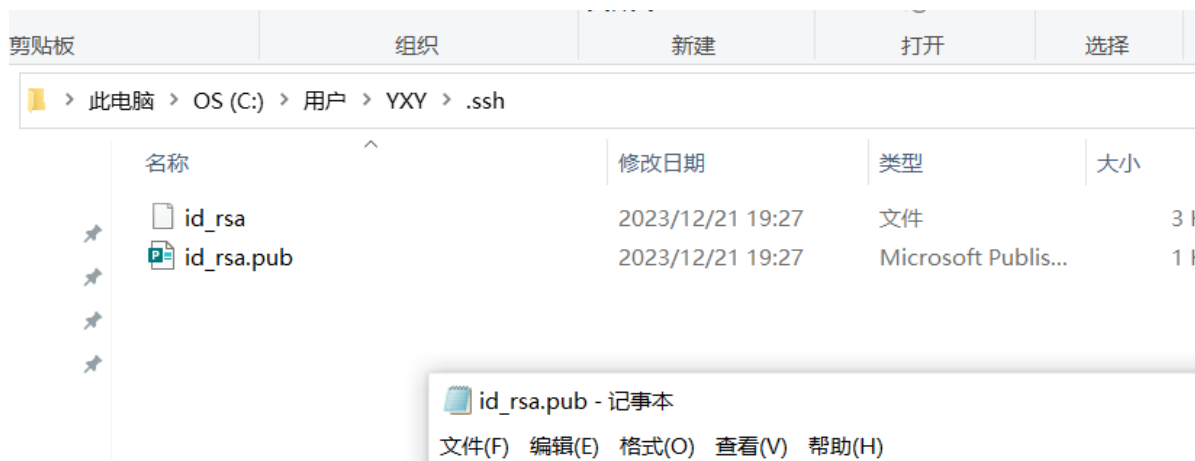
(我这边已经创建过了，这是截的一个网图)



```
agle@agle-PC MINGW64 ~ (master)
$ ssh-keygen -t rsa -C "邮箱"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/agle/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/agle/.ssh/id_rsa.
Your public key has been saved in /c/Users/agle/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:ahFZxUFes/e8a60Wv+JPa0wFyYX5yHdxtBr8E54qqew 1424160347@qq.com
The key's randomart image is:
+---[RSA 2048]---+
|      .+++.o. Bo|
|      o ....oB.+|
|      o  .  .=.Bo|
|      .  .  ..Bo*|
|      . So . . o=|
|      .o. .    ..|
|      oo      +o |
|      ..      +.=o|
|      E      ooB*o|
+---[SHA256]-----+
http://blog.csdn.net/u011859677
```

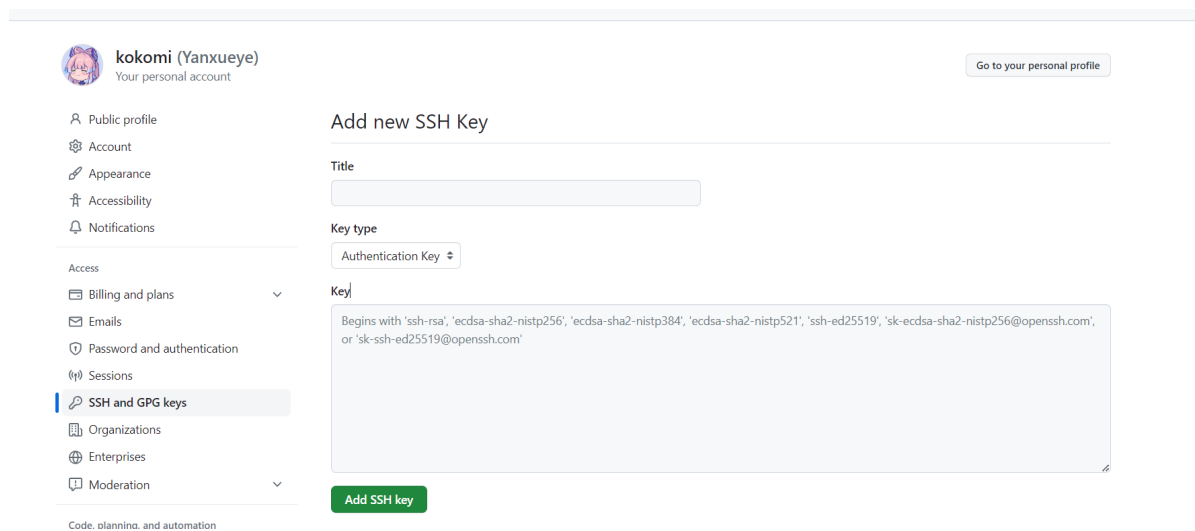
然后我们按照红线所示路径找到这个文件（记得打开显示隐藏文件，agle是系统本地用户名）

用记事本打开文件，复制里面全部内容



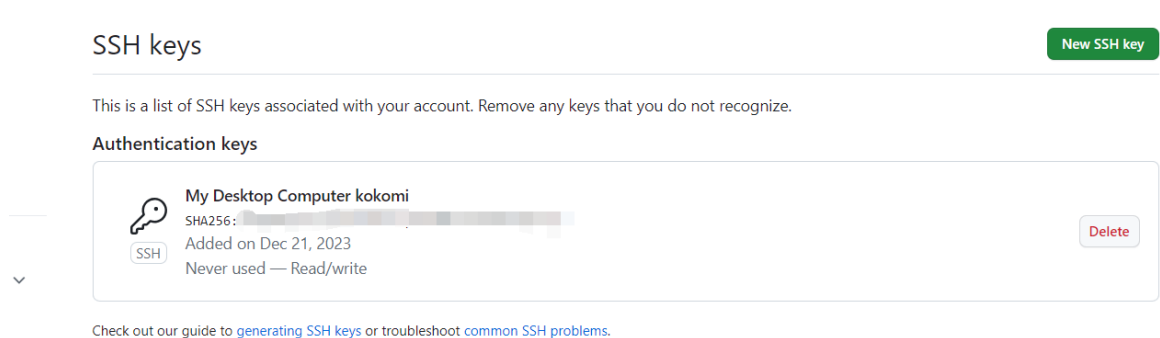
打开GitHub，点击头像，Settings -> SSH and GPG keys

点击右上角 New SSH key，然后进入这个界面。



Title随便写，Key框里粘贴复制内容。

完成后点击Add SSH key，然后输入密码验证一下。完成之后结果如下，然后你写的邮箱里应该会收到ssh-key有关的邮件。



然后再次打开git bash

输入 `ssh -T git@github.com`

随后系统会询问我们是否要继续连接，输入yes然后回车，然后github的信息就被写进hosts文件了，这意味着我们和GitHub账号绑定了。

```
YXY@DESKTOP-PSGQT5E MINGW64 /d/Git/onlineRepo/TestRepository (main)
$ ssh -T git@github.com
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Hi Yanxueye! You've successfully authenticated, but GitHub does not provide shell access.
```

## 二、Github和本地仓库传输命令

### 1.将本地仓库上传

我们需要首先自己在GitHub上创建一个Repository（后面用Repo代替，这是大家默认的简写）

头像 -> Your repositories -> 右上角 New

Great repository names are short and memorable. Need inspiration? How about [improved-rotary-phone](#) ?

Description (optional)

☒ Public

Anyone on the internet can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Initialize this repository with:

☐ Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: Java ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: GNU General Public License v3.0 ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

① You are creating a public repository in your personal account.

Create repository

库名和描述好说。

下面选择库的属性，公开或私有，字面意思。

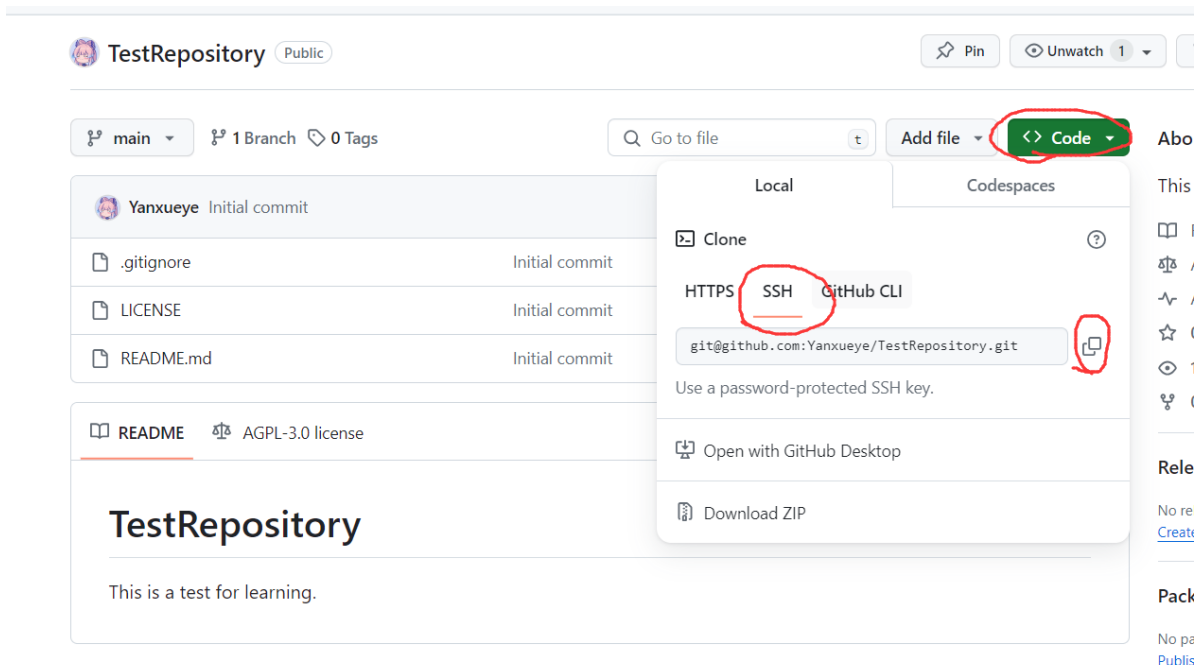
README file是关于你的项目的介绍，大项目一般都有，是markdown文件，可以勾选也可以不勾选，勾选后GitHub会为你创建Readme.md。

.gitignore文件指的是在git工具处理过程中被忽略的文件，这里包含一些模板，我们选择Java模板就可以了，它会帮我们过滤掉\*.class, \*.log, \*.jar等等运行文件或Java配置文件。

license是使用许可证，我们使用GNU General Public License v3.0就足够了，其他的详情自行百度。

接下来我们需要让这个仓库和本地仓库建立联系。

首先复制Repo的SSH地址。



在你本地写好的仓库位置打开Git Bash

输入以下命令

```
git remote add origin 复制的内容（右键Paste）
```

其中 `git remote` 命令是控制远程仓库的命令。

`origin`是默认远程仓库别名。就像一个三孩家庭孩子除了自己的名字还叫老大老二老三一样。

复制的内容是仓库的ssh

`add` 命令在当前本地库的位置加入了绑定的新远程库。

```
YXY@DESKTOP-PSGQT5E MINGW64 /d/Git/repository2 (master)
$ git remote add origin git@github.com:Yanxueye/TestRepository.git
```

输入完后没有任何反应表示添加成功。

我们使用以下命令来将本地库上传到云端（其实是本地分支和Github远程分支的合并，下面实际上是本地master分支和远程分支main（github默认分支）的合并）

```
git push -u origin master
```

（如果报错比如差异过大等请尝试 `git push -u origin master -f` 进行强制推送）

```

YXY@DESKTOP-PSGQT5E MINGW64 /d/Git/repository2 (master)
$ git push -u origin master
Enumerating objects: 30, done.
Counting objects: 100% (30/30), done.
Delta compression using up to 12 threads
Compressing objects: 100% (27/27), done.
Writing objects: 100% (30/30), 3.88 KiB | 1.29 MiB/s, done.
Total 30 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/Yanxueye/TestRepository/pull/new/master
remote:
To github.com:Yanxueye/TestRepository.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

```

然后我们的GitHub上收到一条合并分支的消息，点Compare & pull request

**master** had recent pushes 9 minutes ago
 Compare & pull request

main
2 Branches
0 Tags

Add file
<> Code

**Yanxueye** Initial commit
 88aced3 · yesterday
1 Commits

.gitignore	Initial commit	yesterday
LICENSE	Initial commit	yesterday
README.md	Initial commit	yesterday

**README**
 AGPL-3.0 license

### Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#). [Learn more about diff comparisons here](#).

base: main
compare: master
✓ Able to merge. These branches can be automatically merged.

**Add a title**

**Add a description**

Write

Preview

H B I L <>

- 
- 
-

This is a new version of the project.

Markdown is supported
 Paste, drop, or click to add files

Reviewers
 No reviews

Assignees
 No one—[assign yourself](#)

Labels
 None yet

Projects
 None yet

Milestone
 No milestone

Development
 Use [closing keywords](#) in the description to automatically close issues

Helpful resources

Create pull request

一路确认，哪里亮了点哪里（如果遇到Conflicts，请按照前面分支内容处理）。最后结果是这样的

This screenshot shows the GitHub pull request interface for pull request #2. The header indicates it was merged 1 minute ago by user Yanxueye. The main content area shows a comment: "This is a new version of the project." Below the comment, a commit message "Hello" is visible. The bottom section shows the pull request was successfully merged and closed, with a message: "You're all set—the master branch can be safely deleted." A "Delete branch" button is present. On the right side, there are tabs for "Review", "Assign", "Labels", "Project", and "Milestone".

回头看创建的库，ok，更新完毕。

This screenshot shows the GitHub repository interface. At the top, there's a navigation bar with "main" branch selected, "3 Branches", and "0 Tags". A search bar "Go to file" and buttons "Add file" and "Code" are also visible. Below the navigation bar, a table lists the repository files and their commit history:

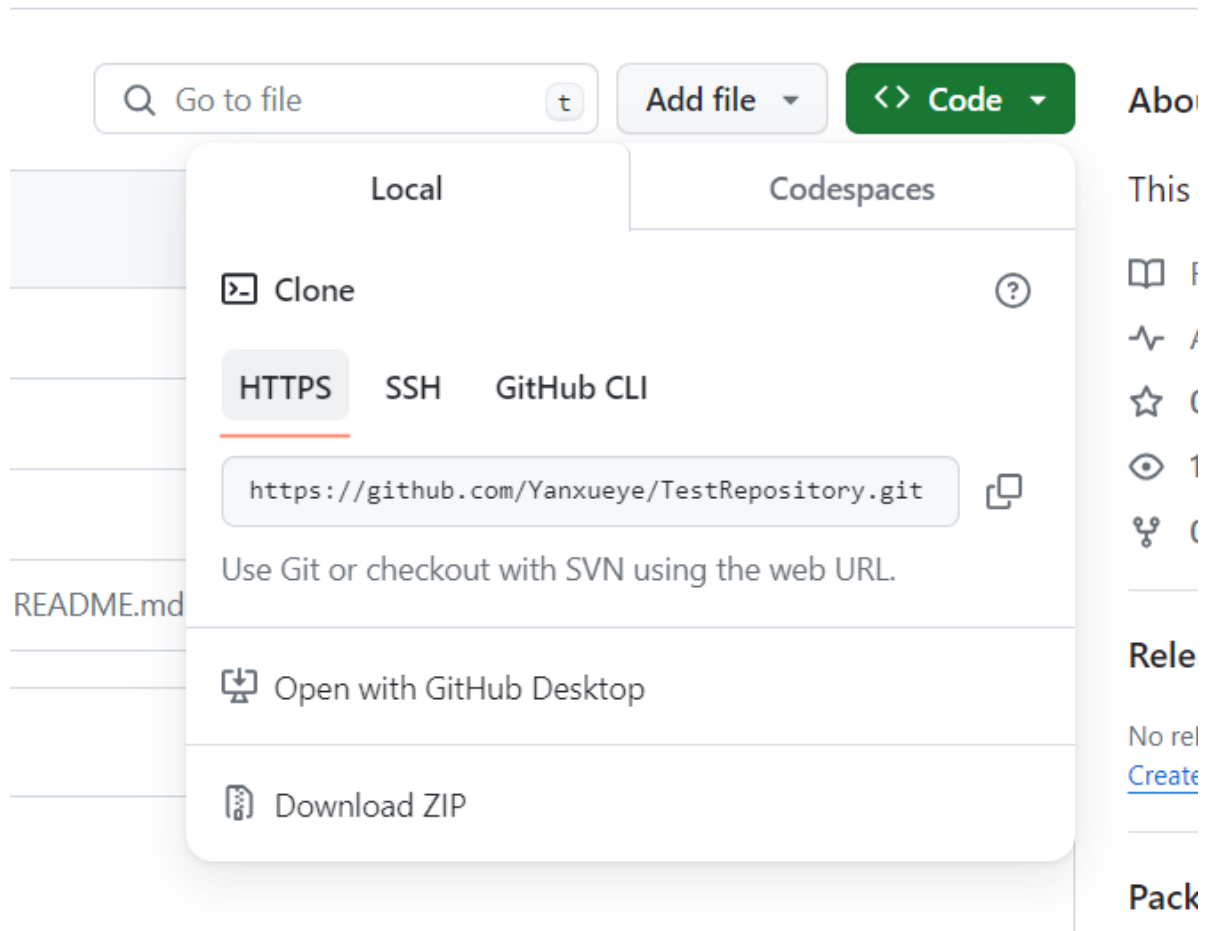
File	Commit	Time
Hi	Recursion	11 minutes ago
.gitignore	Test	2 hours ago
Hello.txt	Hello	33 minutes ago
README.md	Merge branch 'main' into develop	5 minutes ago

Below the file list, the README file is selected, showing the title "TestRepository" and the content "This is a test for learning."

## 2.Copy别人的项目到本地

使用 git clone 命令。


点击项目右上角<>Code，切换到HTTPS，然后复制。



然后在本地要存储的位置右击打开Git Bash

输入 `git clone <鼠标右键Paste粘贴网址>`

```
XY@DESKTOP-PSGQT5E MINGW64 /d/Git
$ git clone https://github.com/Yanxueye/TestRepository.git
Cloning into 'TestRepository'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), 12.40 KiB | 470.00 KiB/s, done.
```

 TestRepository 2023/12/21 21:4

然后这个项目就被下载下来了，非常简单。

GitHub本身也有很多有趣的操作，比如云复制fork，评论issues，提交修改申请Pull Requests等等，大家感兴趣可以自己搜索>\_<，其实所有的操作GitHub官方也有Help文档，只不过都是英文hhh。