

Git学习

git是免费、开源的[分布式版本控制工具](#)，可以快速高效地处理从小型到大型地各种项目。

github

- 代码推送 push
- 代码拉取 pull
- 代码克隆 clone
- SSH免密登录

版本控制

版本控制就是一种[记录文件内容变化](#)、以便将来查阅特定版本修订情况的系统。

版本控制最重要的是可以记录文件修改历史记录，从而让用户能够查看历史版本，方便版本切换。

代码托管中心

代码托管中心是基于网络服务器的远程代码仓库，一般简单成为[远程库](#)

- 局域网
 - GitLab
- 互联网
 - GitHub（外网）
 - Gitee码云（国内）

Git常用命令

命令名称	作用
git config --global user.name 用户名	设置用户签名
git config --global user.email 邮箱	设置用户邮箱
git init	初始化本地库
git status	查看本地库状态
git add 文件名(git add . 添加当前目录下的所有文件)	将文件添加到暂存区

命令名称	作用
git commit -m "日志信息" 文件名	提交到本地库
git reflog	查看历史记录
git reset --hard 版本号	版本穿梭

设置用户签名和邮箱

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

```
git config --global user.email 3080680474@qq.com
```

签名的作用是区分不同操作者身份。用户的签名信息在每一个版本提交信息中能够看到，以此确认本次提交是水做的。**Git首次安装必须设置一下用户签名，否则无法提交代码。**

注意：这里设置用户签名和将来登录GitHub的账号没有任何关系

初始化本地仓库

在需要设置仓库的文件路径下输入 `git init` 即可初始化本地库。

查看本地库状态

使用 `git status` 即可查看本地库状态。

```
Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/学习笔记 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   "JS\345\255\246\344\271\240.md"
    new file:   "css\343\200\201js\346\213\223\345\261\225.md"
    new file:   "\345\211\215\347\253\257\351\235\242\350\257\225\346\211\21
3\345\206\231\351\242\230.md"
    new file:   "\347\256\227\346\263\225\347\254\224\350\256\260.md"
    new file:   "\347\275\221\347\273\234\345\255\246\344\271\240.md"

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    "git\345\255\246\344\271\240.md"
```

untracked files代表没有添加到暂存区的文件，git并没有追踪到，可以使用 `git add 文件名` 添加到暂存区。

```

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/学习笔记 (master)
$ git add git学习.md

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/学习笔记 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   "JS\345\255\246\344\271\240.md"
    new file:   "css\343\200\201js\346\213\223\345\261\225.md"
    new file:   "git\345\255\246\344\271\240.md"
    new file:   "\345\211\215\347\253\257\351\235\242\350\257\225\346\211\213\345\206\231\351\242\230.md"
    new file:   "\347\256\227\346\263\225\347\254\224\350\256\260.md"
    new file:   "\347\275\221\347\273\234\345\255\246\344\271\240.md"

```

将文件从暂存区中删除

`git rm --cached 文件名` 即可把文件从暂存区中删除。只是从暂存区中删除，工作区中并没有删除。

```

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/学习笔记 (master)
$ git rm --cached git学习.md
rm 'git学习.md'

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/学习笔记 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   "JS\345\255\246\344\271\240.md"
    new file:   "css\343\200\201js\346\213\223\345\261\225.md"
    new file:   "\345\211\215\347\253\257\351\235\242\350\257\225\346\211\213\345\206\231\351\242\230.md"
    new file:   "\347\256\227\346\263\225\347\254\224\350\256\260.md"
    new file:   "\347\275\221\347\273\234\345\255\246\344\271\240.md"

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    "git\345\255\246\344\271\240.md"

```

提交到本地库

`git commit -m "日志信息"` 即可将文件添加到本地库。

```

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/学习笔记 (master)
$ git commit -m "学习笔记2023.5.27"
[master (root-commit) fde23bf] 学习笔记2023.5.27
6 files changed, 2917 insertions(+)
create mode 100644 "JS\345\255\246\344\271\240.md"
create mode 100644 "css\343\200\201js\346\213\223\345\261\225.md"
create mode 100644 "git\345\255\246\344\271\240.md"
create mode 100644 "\345\211\215\347\253\257\351\235\242\350\257\225\346\211\234\345\206\231\351\242\230.md"
create mode 100644 "\347\256\227\346\263\225\347\254\224\350\256\260.md"
create mode 100644 "\347\275\221\347\273\234\345\255\246\344\271\240.md"

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/学习笔记 (master)
$ git status
On branch master
nothing to commit, working tree clean

```

使用 `git reflog` 可以查看提交信息。其中 `fde23bf` 就是本次提交的版本信息。

```

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/学习笔记 (master)
$ git reflog
fde23bf (HEAD -> master) HEAD@{0}: commit (initial): 学习笔记2023.5.27

```

`git log` 不仅可以看到提交信息，也可以看到提交者。

```

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/学习笔记 (master)
$ git log
commit fde23bf5221697c6539f53a565dc2415d0767f8f (HEAD -> master)
Author: efohuajiao <3080680474@qq.com>
Date: Sat May 27 14:35:17 2023 +0800

    学习笔记2023.5.27

```

修改文件

如果文件在提交到本地库后经过修改，使用 `git status` 查看本地库状态时会显示修改后的文件，本地库状态会提示修改的文件。

```

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/学习笔记 (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   "git\345\255\246\344\271\240.md"

no changes added to commit (use "git add" and/or "git commit -a")
Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/学习笔记 (master)

```

这时就需要重新提交修改的文件到本地库，然后本地库就的版本就会指向最新的提交版本。

```
Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/学习笔记 (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   "git\345\255\246\344\271\240.md"

no changes added to commit (use "git add" and/or "git commit -a")

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/学习笔记 (master)
$ git add git学习.md

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/学习笔记 (master)
$ git commit -m "second"
[master d7268eb] second
1 file changed, 46 insertions(+), 1 deletion(-)

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/学习笔记 (master)
$ git reflog
d7268eb (HEAD -> master) HEAD@{0}: commit: second
fde23bf HEAD@{1}: commit (initial): 学习笔记2023.5.27
```

历史版本

`git reflog` 可以查看版本信息。

`git log` 可以看到版本详细信息。

版本穿梭

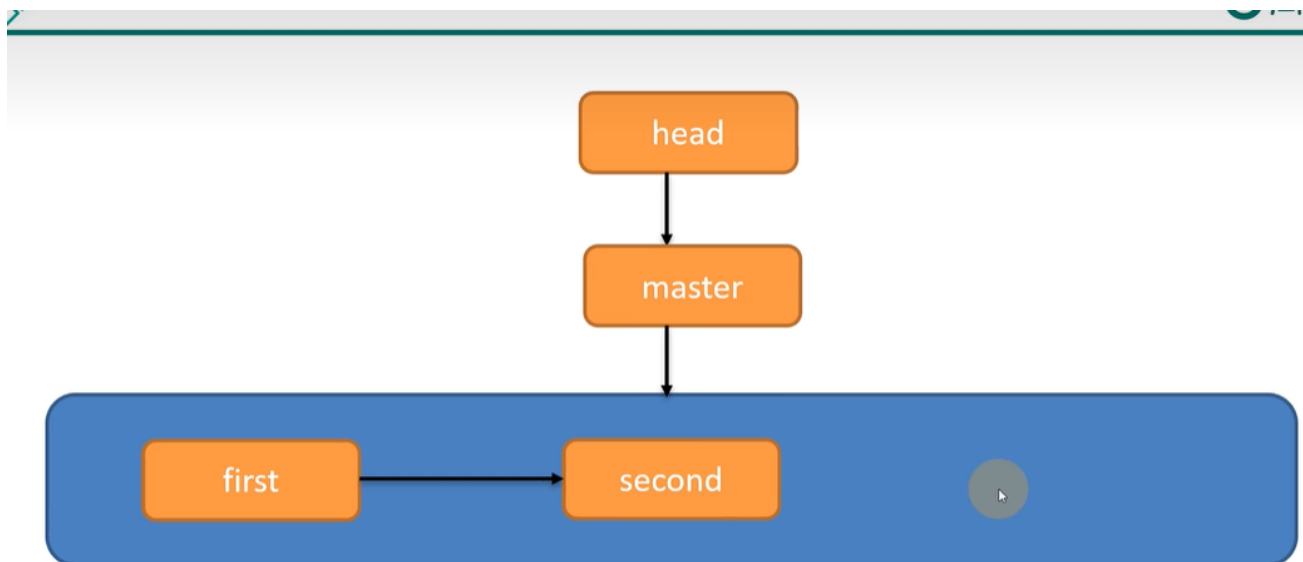
步骤：

1. 通过 `git reflog` 查看版本号
2. 使用 `git reset --hard 版本号` 即可跳转到对应的版本，文件也会回退到对应的版本。

```
Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/学习笔记 (master)
$ git reset --hard d7268eb
HEAD is now at d7268eb second

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/学习笔记 (master)
$ git reflog
d7268eb (HEAD -> master) HEAD@{0}: reset: moving to d7268eb
a1f09df HEAD@{1}: commit: third
d7268eb (HEAD -> master) HEAD@{2}: commit: second
fde23bf HEAD@{3}: commit (initial): 学习笔记2023.5.27
```

`git`切换版本，底层其实是移动的HEAD指针。



分支

在版本控制过程中，同时推进多个任务，为每个任务就可以创建单独的分支，使用分支意味着程序员可以把自己的工作从开发主线上分离开来，开发自己分支的时候，不会影响主线分支的运行。

分支的操作

命令名称	作用
git branch -v	查看分支
git branch 分支名	创建分支
git checkout 分支名	切换分支
git merge 分支名	把指定的分支合并到当前分支上

```
Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/git练习 (master)
$ git branch
* master

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/git练习 (master)
$ git branch -v
* master ab8a1bc first commit

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/git练习 (master)
$ git branch hot-fix

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/git练习 (master)
$ git checkout hot-fix
Switched to branch 'hot-fix'

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/git练习 (hot-fix)
$ vim hello.txt
```

图中后面蓝色括号里的表示**当前分支**。

每一个分支修改的文件仅在当前分支下有效，切换分支后，文件会变成对应分支的文件。

hot-fix分支下的hello.txt文件。

```
Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/git练习 (hot-fix)
$ cat hello.txt
hello world!
hello git!
It's hot-fix branch modified!
```

master分支下的hello.txt文件。

```
Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/git练习 (hot-fix)
$ git checkout master
Switched to branch 'master'

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/git练习 (master)
$ cat hello.txt
hello world!
hello git!
```

合并分支

master是主分支，合并分支要在master分支下进行合并！！

`git merge 分支名`

```
Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/git练习 (master)
$ git merge hot-fix
Updating ab8a1bc..8329d9f
Fast-forward
 hello.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/git练习 (master)
$ cat hello.txt
hello world!
hello git!
It's hot-fix branch modified!
```

合并冲突

合并分支时，两个分支，在同一个文件的同一个位置有两套完全不同的修改。Git无法替我们决定使用哪一个，必须认为决定新代码内容。

```
Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/git练习 (master)
$ git merge hot-fix
Auto-merging hello.txt
CONFLICT (content): Merge conflict in hello.txt
Automatic merge failed; fix conflicts and then commit the result.

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/git练习 (master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   hello.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

这时就要人为修改文件，决定文件的内容。

```
MINGW64:/d/fronted/git练习
hello world!
hello git!
It's hot-fix branch modified!
It's master branch modified!
hot-fix modified again!
<<<<<<< HEAD
master again!
=====
hello,I'm hot-fix!
>>>>>>> hot-fix
~
~
~
~
~
~
~
```

```
MINGW64:/d/fronted/git练习
hello world!
hello git!
It's hot-fix branch modified!
It's master branch modified!
hot-fix modified again!
master again!
hello,I'm hot-fix!
~
~
~
~
~
~
~
```

```
Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/git练习 (master|MERGING)
$ git add .

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/git练习 (master|MERGING)
$ git commit -m "merge again"
[master 164e63a] merge again

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/git练习 (master)
$ |
```

远程仓库操作（github为例）

命令名称	作用
git remote -v	查看当前所有远程地址别名
git remote add 别名 远程地址	起别名，添加远程地址
git remote rm 别名	删除远程仓库
git push 别名 分支	推送本地分支上的内容到远程仓库
git clone 远程地址	将远程仓库的内容克隆到本地
git pull 远程库地址别名 远程分支名	将远程仓库对于分支最新内容拉下来后与当前本地分支直接合并

创建远程仓库别名

添加别名: `git remote add 别名 远程地址`

查看别名: `git remote -v`

```
Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/学习笔记 (master)
$ git remote add fronted_study https://github.com/efohuajiao/fronted_study.git

Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/学习笔记 (master)
$ git remote -v
fronted_study https://github.com/efohuajiao/fronted_study.git (fetch)
fronted_study https://github.com/efohuajiao/fronted_study.git (push)
```

将本地库代码推送到远程库

`git push 别名 分支`

```
Redmi@DESKTOP-6B4IKVM MINGW64 /d/fronted/js/friend_contact (master)
$ git push friendContact_fronted master
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% (2/2), 224 bytes | 224.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/efohuajiao/friendContact_fronted.git
 78e910f..c916f48 master -> master
```

将远程库代码下载到本地库

`git pull 别名 分支`

获取到更新后的代码，使本地库代码与远程库代码同步。

克隆远程库代码

`git clone 远程库地址`

clone所做

1. 拉取远程库的代码
2. 初始化本地仓库
3. 创建别名

团队协作

需要添加成员后，成员才能对代码进行推送。

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Codespaces

Pages

Security

Code security and analysis

Deploy keys

General

Repository name

fronted_study

Rename

☐ Template repository

Template repositories let users generate new repositories with the same directory structure and files. [Learn more.](#)

☐ Require contributors to sign off on web-based commits

Enabling this setting will require contributors to sign off on commits made through GitHub's web interface. Signing off is a way for contributors to affirm that their commit complies with the repository's terms, commonly the [Developer Certificate of Origin \(DCO\)](#). [Learn more about signing off on commits.](#)

Default branch

The default branch is considered the "base" branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.

master

Social Preview

WHO HAS ACCESS

PUBLIC REPOSITORY

This repository is public and visible to anyone.

Manage

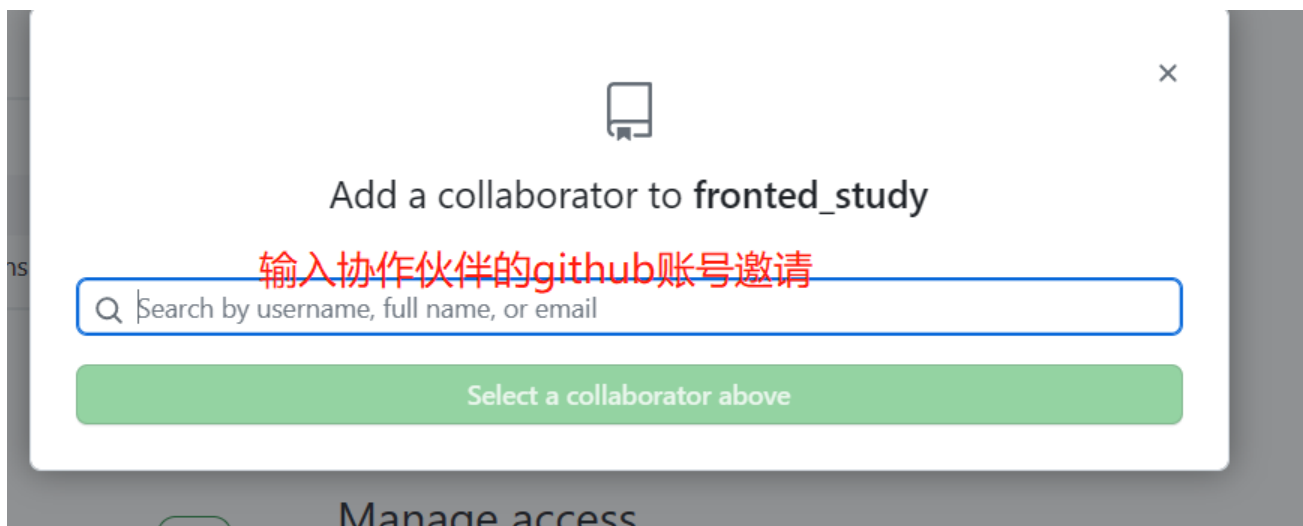
DIRECT ACCESS

0 collaborators have access to this repository. Only you can contribute to this repository.

Manage access

You haven't invited any collaborators yet

Add people



将随后生成的链接发给对方，**对方同意后**即可加入创作团队，进行代码推送。

远程仓库操作（SSH）


1. 创建SSH

`ssh-keygen -t rsa -C github账号` 即可生成ssh密钥。




2. 在C盘用户目录下的.ssh文件夹中即可看到公钥和私钥。

粘贴	组织	新建	打开	选择
📁 > 此电脑 > 本地磁盘 (C:) > 用户 > Redmi > .ssh				
名称		修改日期	类型	大小
personal	id_rsa	2023/4/30 15:43	文件	3 KB
	id_rsa.pub	2023/4/30 15:43	PUB 文件	1 KB
	known_hosts	2023/4/30 15:53	文件	1 KB

.pub是公钥，复制公钥，在github中添加公钥。




[Pull requests](#) [Issues](#) [Codespaces](#) [Marketplace](#) [Explore](#)


  


Top Repositories


New


Find a repository...

 [efohuajiao/efohuajiao.github.io](#)

 [efohuajiao/friendContact_frontend](#)

 [efohuajiao/fronted_study](#)

 [Molly-Lotusdream/HTML-MasterGo](#)

 [yybgx/Imitation_gold_digger](#)

Recent activity


When you take actions across GitHub, we'll provide links to that activity here.


For you Beta **Following**

Discover interesting projects and people to populate your personal news feed.


Your news feed helps you keep up with recent activity on repositories you [watch](#) or [star](#) and people you [follow](#).

Explore GitHub

 **ProTip!** The feed shows you events from people you [follow](#) and repositories you [watch](#) or [star](#).

 [Subscribe to your news feed](#)

Signed in as **efohuajiao**

 Set status

[Your profile](#)

[Your repositories](#)

[Your projects](#)


[Your stars](#)

[Your gists](#)

[Your sponsors](#)


[Upgrade](#)

[Try Enterprise](#)


[Feature preview](#) 


[Help](#)

Exp [Settings](#)

 Sign out


netease-music


 基于 Vue2、Vue-CLI3 的高仿网易云 mac 客户端播放器 (PC) Online Music Player





huajiao (efohuajiao)
Your personal account


Go to your personal profile

 **Public profile**


 Account


 Appearance


 Accessibility


 Notifications


Access


 Billing and plans


 Emails

 Password and authentication

 Sessions

 **SSH and GPG keys**

 Organizations

 Moderation

Code, planning, and automation

Public profile

Name

huajiao

Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.

Public email

Select a verified email to display

You have set your email address to private. To toggle email privacy, go to [email settings](#) and uncheck "Keep my email address private."

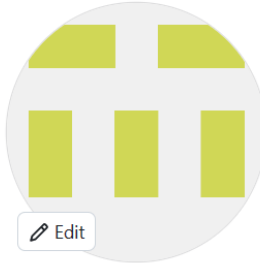
Bio

Tell us a little bit about yourself

You can @mention other users and organizations to link to them.

Pronouns

Profile picture



Edit



huajiao (efohuajiao)
Your personal account

[Go to your personal profile](#)

[Public profile](#)

[Account](#)

[Appearance](#)

[Accessibility](#)

[Notifications](#)

[Access](#)

[Billing and plans](#)

SSH keys

[New SSH key](#)

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

Authentication Keys



myblog

SHA256:FhZSzuJGR582dQQUb37Azipu2H03gSgW1UGpf1Bas/s

Added on Apr 30, 2023

Last used within the last 4 weeks — Read/write

[Delete](#)

SSH keys / Add new

Title

名字随便命名

Key type

Authentication Key

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

输入刚刚复制的公钥

[Add SSH key](#)

点击Add SSH key即可生成SSH密钥

3. 随后即可通过SSH进行代码的拉取、推送等操作

master

1 branch

0 tags

Go to file

Add file

<> Code

加速

efohuajiao 删除

public

src

.gitignore

README.md

babel.config.js

jsconfig.json

package-lock.js

package.json

vue.config.js

Local

Codespaces

Clone

HTTPS SSH GitHub CLI

git@github.com:efohuajiao/friendContact_front

Use a password-protected SSH key.

Open with GitHub Desktop

Download ZIP

9 commits

s ago

s ago

o

o

o

o

o

2023/5/22

第一次上传 2023.3.20

5 days ago

2 months ago

About

游戏伙伴匹配系统
用Vue+Vant UI框架

Readme

0 stars

1 watching

0 forks

Releases

No releases published
[Create a new release](#)

Packages