[**https://blog.csdn.net/pets\_1021/article/details/131536418**](https://blog.csdn.net/pets_1021/article/details/131536418) **最新版本安装教程**

**创建账号部分比较简单，参照语雀即可**

Administrator@VJWP5NF92RVKZOC MINGW64 ~ (master)

$ cd d:/java/workspace/boookManage\_Library-jdbc **进入库所在目录**

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (master)

$ git clone -b https://github.com/yyybx/test1.get

fatal: You must specify a repository to clone.

usage: git clone [<options>] [--] <repo> [<dir>]

-v, --verbose be more verbose

-q, --quiet be more quiet

--progress force progress reporting

--reject-shallow don't clone shallow repository

-n, --no-checkout don't create a checkout

--bare create a bare repository

--mirror create a mirror repository (implies bare)

-l, --local to clone from a local repository

--no-hardlinks don't use local hardlinks, always copy

-s, --shared setup as shared repository

--recurse-submodules[=<pathspec>]

initialize submodules in the clone

--recursive ... alias of --recurse-submodules

-j, --jobs <n> number of submodules cloned in parallel

--template <template-directory>

directory from which templates will be used

--reference <repo> reference repository

--reference-if-able <repo>

reference repository

--dissociate use --reference only while cloning

-o, --origin <name> use <name> instead of 'origin' to track upstream

-b, --branch <branch> checkout <branch> instead of the remote's HEAD

-u, --upload-pack <path>

path to git-upload-pack on the remote

--depth <depth> create a shallow clone of that depth

--shallow-since <time>

create a shallow clone since a specific time

--shallow-exclude <revision>

deepen history of shallow clone, excluding rev

--single-branch clone only one branch, HEAD or --branch

--no-tags don't clone any tags, and make later fetches not to fo

llow them

--shallow-submodules any cloned submodules will be shallow

--separate-git-dir <gitdir>

separate git dir from working tree

-c, --config <key=value>

set config inside the new repository

--server-option <server-specific>

option to transmit

-4, --ipv4 use IPv4 addresses only

-6, --ipv6 use IPv6 addresses only

--filter <args> object filtering

--also-filter-submodules

apply partial clone filters to submodules

--remote-submodules any cloned submodules will use their remote-tracking b

ranch

--sparse initialize sparse-checkout file to include only files

at root

--bundle-uri <uri> a URI for downloading bundles before fetching from ori

gin remote

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (master)

$ git init **初始化库**

Reinitialized existing Git repository in D:/java/workspace/boookManage\_Library-j

dbc/.git/

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (master)

$ git temote add test1 d:/java/workspace/boookManage\_Library-jdbc

**拼写错误temote -> remote**

git: 'temote' is not a git command. See 'git --help'.

The most similar command is

remote

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (master)

$ git remote add test1 d:/java/workspace/boookManage\_Library-jdbc

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (master)

$ ssh-keygen -t rsa -C [745376088@qq.com](mailto:745376088@qq.com) **生成连接Github的ssh-key**

Generating public/private rsa key pair.

Enter file in which to save the key (/c/Users/Administrator/.ssh/id\_rsa):

**存储ssh-key的路径**

Created directory '/c/Users/Administrator/.ssh'.

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /c/Users/Administrator/.ssh/id\_rsa

Your public key has been saved in /c/Users/Administrator/.ssh/id\_rsa.pub

The key fingerprint is:

SHA256:qEXHbTUucrN0vk6YqEiAnLQv/bQMAJoHc3bxKOt5EW4 [745376088@qq.com](mailto:745376088@qq.com)

The key's randomart image is:

+---[RSA 3072]----+

| . o |

| + . . o . |

|+.+ + o + B o |

|=Bo= o o = \* |

|+=+ E o S . . |

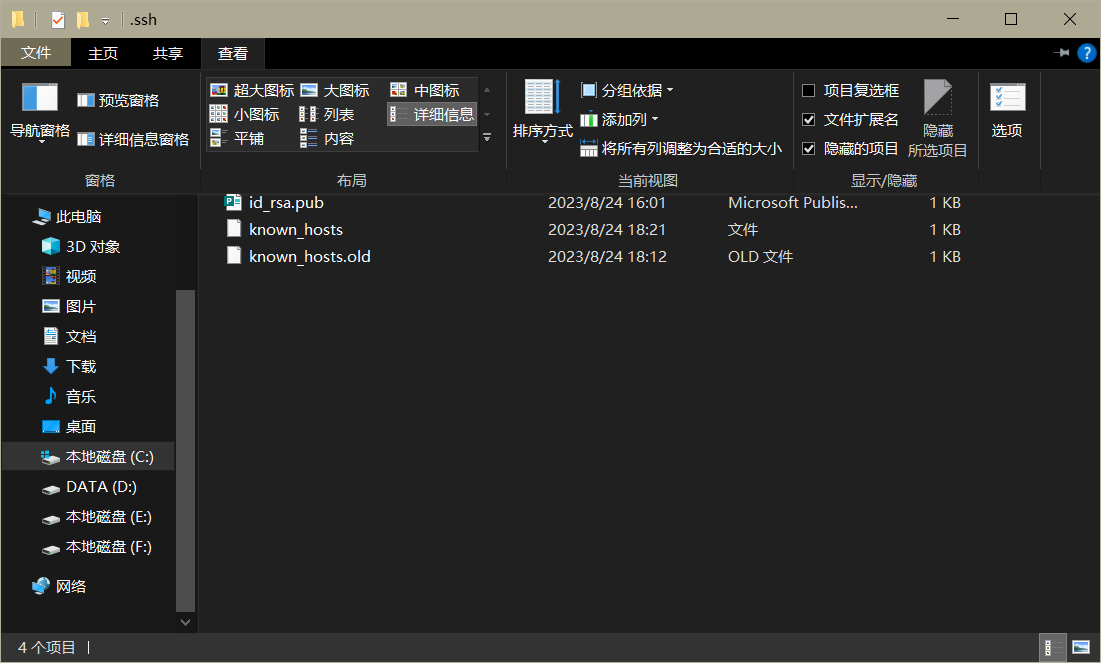
| o++ + . o . |

| .o++. . o o |

| .o=... o |

| .+. . |

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



**Key存储在id\_rsa.pub中，可以直接用记事本打开**



**注意不包括”ssh-rsa”这部分**

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (master)

$ ssh -T [git@github.com](mailto:git@github.com) **连接Git与GitHub**

The authenticity of host 'github.com (127.0.0.1)' can't be established.

ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/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.

ssh\_dispatch\_run\_fatal: Connection to 127.0.0.1 port 22: Broken pipe

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (master)

$ git clone -b dev https://github.com/yyybx/test1.git

Cloning into 'test1'...

fatal: unable to access 'https://github.com/yyybx/test1.git/': SSL certificate p

roblem: unable to get local issuer certificate

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (master)

$ mkdir test1-git+test **在当前目录下新建目录（文件夹）**

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (master)

$ cd test-git+test **进入目录**

bash: cd: test-git+test: No such file or directory

**以下是在GitHub新建存储库时的连接测试**

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (master)

$ echo "# test1" >> README.md

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (master)

$ git init

Reinitialized existing Git repository in D:/java/workspace/boookManage\_Library-j

dbc/.git/

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (master)

$ add README.md

bash: add: command not found

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (master)

$ git add README.md

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (master)

$ branch -M main

bash: branch: command not found

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (master)

$ git branch -M main

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (main)

$ git remote add origin git@github.com:yyybx/test1.git

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (main)

$ push -u origin main

bash: push: command not found

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (main)

$ git push -u origin main

Enumerating objects: 3, done.

Counting objects: 100% (3/3), done.

Delta compression using up to 4 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (3/3), 569 bytes | 11.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

To github.com:yyybx/test1.git

\* [new branch] main -> main

branch 'main' set up to track 'origin/main'.

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (main)

$ git remote **检查现在的本地库**

origin

test1

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (main)

$ git remote -v **检查现在所有的库**

origin git@github.com:yyybx/test1.git (fetch)

origin git@github.com:yyybx/test1.git (push)

test1 d:/java/workspace/boookManage\_Library-jdbc (fetch)

test1 d:/java/workspace/boookManage\_Library-jdbc (push)

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (main)

$ git clone -b dev <https://github.com/yyybx/test1.git> **不信任网络**

Cloning into 'test1'...

fatal: unable to access 'https://github.com/yyybx/test1.git/': SSL certificate p

roblem: unable to get local issuer certificate

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (main)

$ git config --global http.sslVerify false **解决了不信任网络的问题**

**解决方法https://blog.csdn.net/weixin\_44014995/article/details/109900149**

**解决了不信任服务器无法克隆远程库的问题**

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (main)

$ git clone -b dev <https://github.com/yyybx/test1.git> **分支名错误dev -> main**

Cloning into 'test1'...

fatal: Remote branch dev not found in upstream origin

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (main)

$ git clone -b main <https://github.com/yyybx/test1.git> **克隆库到本地**

Cloning into 'test1'...

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 3 (delta 0), pack-reused 0

Receiving objects: 100% (3/3), done.

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (main)

$ git checkout main **使用主分支**

Already on 'main'

M readme.md

Your branch is up to date with 'origin/main'.

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (main)

$ git pull **检查主分支是否有需要同步的更新**

Already up to date.

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (main)

$ git branch test\_new **新建测试分支**

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (main)

$ get checkout test\_new **get -> git**

bash: get: command not found

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (main)

$ git checkout test\_new **切换至新建的测试分支**

Switched to branch 'test\_new'

M readme.md

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (test\_new)

$ git add test.txt **向向当前分支提交更新**

fatal: pathspec 'test.txt' did not match any files

**提交失败，解决方法https://blog.csdn.net/qq\_37855074/article/details/90181162**

**解决了提交文件时did not match any files问题**

**只需查看目录然后进入需要上传文件所在目录**

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (test\_new)

$ ls -ah **查看目录**

./ libs/

../ readme.md

.classpath resources/

.git/ src/

.project test/

.settings/ test1/

bin/ test1-git+test/

bookManage\_Library.sql '~$ Microsoft Word 文档.docx'

dangdang\_books\_utf8\_sample\_20180529\_1000.xls '新建 Microsoft Word 文档.docx'

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc (test\_new)

$ cd test1 **前往待提交文件所在目录**

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc/test1 (main)

$ git add test.txt **向当前分支提交修改**

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc/test1 (main)

$ git commit -m "测试文件"

**提交修改（事务）引号里是备注信息Windows系统用双引号Linux系统用单引号**

[main 60fe6cc] 测试文件

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 test.txt

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc/test1 (main)

$ git checkout main **切换至main分支**

Already on 'main'

Your branch is ahead of 'origin/main' by 1 commit.

(use "git push" to publish your local commits)

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc/test1 (main)

$ git push **在main分支上确认提交**

warning: ----------------- SECURITY WARNING ----------------

warning: | TLS certificate verification has been disabled! |

warning: ---------------------------------------------------

warning: HTTPS connections may not be secure. See https://aka.ms/gcm/tlsverify f

or more information.

info: please complete authentication in your browser...

warning: ----------------- SECURITY WARNING ----------------

warning: | TLS certificate verification has been disabled! |

warning: ---------------------------------------------------

warning: HTTPS connections may not be secure. See https://aka.ms/gcm/tlsverify f

or more information.

Enumerating objects: 4, done.

Counting objects: 100% (4/4), done.

Delta compression using up to 4 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (3/3), 283 bytes | 14.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

To https://github.com/yyybx/test1.git

25b4939..60fe6cc main -> main

Administrator@VJWP5NF92RVKZOC MINGW64 /d/java/workspace/boookManage\_Library-jdbc/test1 (main)

$ git log **检查提交历史**

commit 60fe6cc86dbac158ef6b738176df57890db95ad8 (HEAD -> main, origin/main, orig

in/HEAD)

Author: root <root@example.com>

Date: Thu Aug 24 18:35:12 2023 +0800

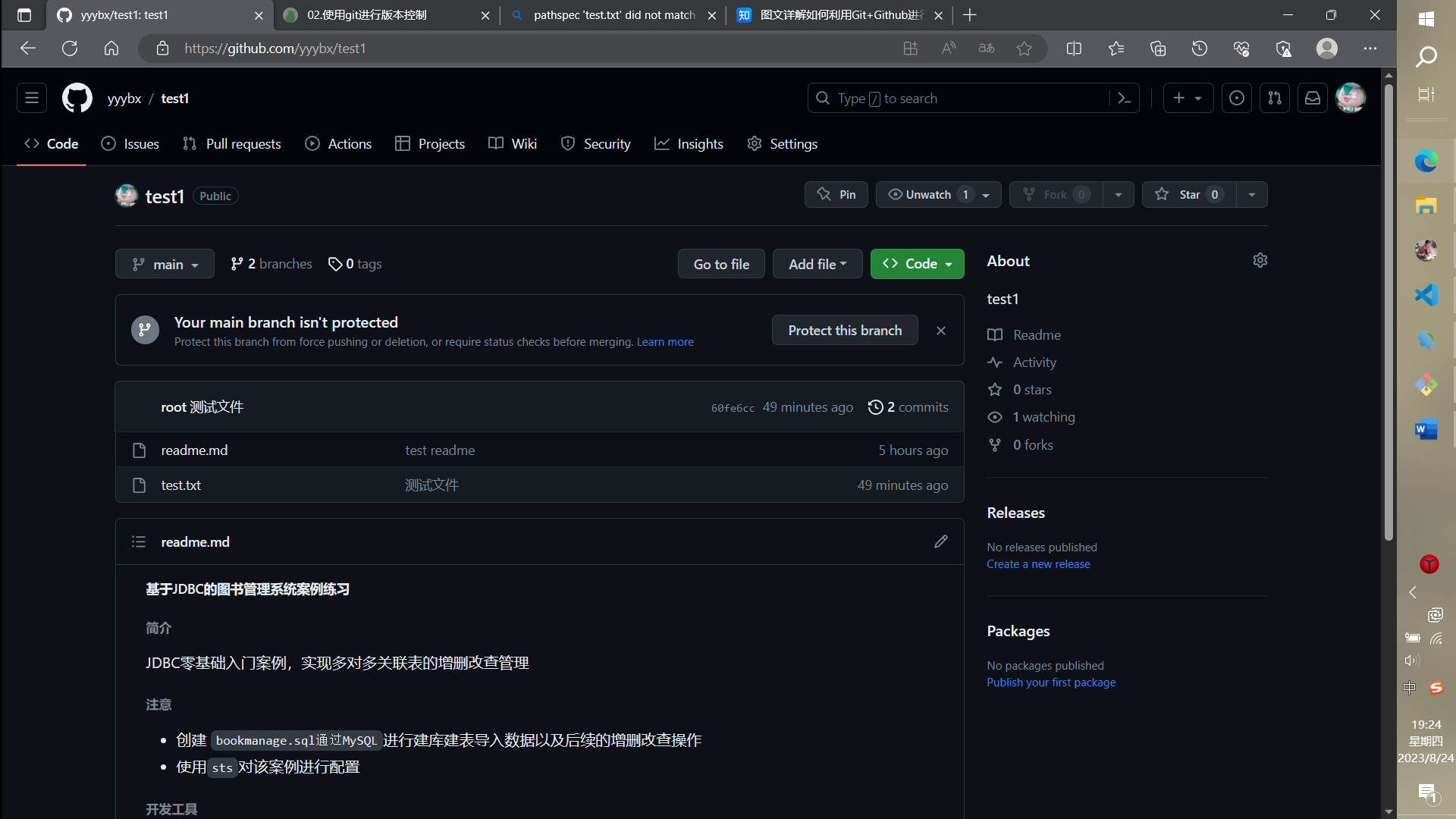
测试文件

commit 25b4939deaf68f4634908f250f0635208f9c83a8

Author: root <root@example.com>

Date: Thu Aug 24 13:36:59 2023 +0800

test readme



此时成功向GitHub上我的项目中添加了测试文件

常用Git命令（可以通过git help在内部查看）

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)

clone Clone a repository into a new directory

init Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)

add Add file contents to the index

mv Move or rename a file, a directory, or a symlink

restore Restore working tree files

rm Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)

bisect Use binary search to find the commit that introduced a bug

diff Show changes between commits, commit and working tree, etc

grep Print lines matching a pattern

log Show commit logs

show Show various types of objects

status Show the working tree status

grow, mark and tweak your common history

branch List, create, or delete branches

commit Record changes to the repository

merge Join two or more development histories together

rebase Reapply commits on top of another base tip

reset Reset current HEAD to the specified state

switch Switch branches

tag Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)

fetch Download objects and refs from another repository

pull Fetch from and integrate with another repository or a local branch

push Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some

concept guides. See 'git help <command>' or 'git help <concept>'

to read about a specific subcommand or concept.

See 'git help git' for an overview of the system.

https://zhuanlan.zhihu.com/p/23478654

协作开发，包含除库创建者之外成员如何进行库的文件流交互

<https://zhuanlan.zhihu.com/p/262926829#:~:text=Git%E4%BD%BF%E7%94%A8%E5%A4%9A%E4%B8%AA%E8%BF%9C%E7%AB%AF%E4%BB%93%E5%BA%93%E5%8D%8F%E4%BD%9C%E5%BC%80%E5%8F%91%201%201%EF%BC%9A%E5%88%9B%E5%BB%BA%E7%A7%81%E6%9C%89%E5%BA%93%E5%A4%87%E7%94%A8%202%202%3A%20%E5%88%9D%E5%A7%8B%E5%8C%96%E6%9C%AC%E5%9C%B0%E9%A1%B9%E7%9B%AE%203%203%3A,%E7%94%9F%E4%BA%A7%E7%8E%AF%E5%A2%83%E7%9A%84%E4%BB%A3%E7%A0%81%E9%83%A8%E7%BD%B2%207%207%3A%20%E5%85%B6%E4%BB%96%E5%90%8C%E4%BA%8B%E5%8A%A0%E5%85%A5%E5%BC%80%E5%8F%91%208%208%3A%20%E7%9B%B4%E6%8E%A5%E7%BC%96%E8%BE%91git%E7%9A%84%E9%85%8D%E7%BD%AE%E6%96%87%E4%BB%B6%20%E6%9B%B4%E5%A4%9A%E9%A1%B9%E7%9B%AE>

多仓库远程开发

有问题可以查看语雀，菜鸟教程相关部分