## git config

用户目录下，全局有效：

git config --global credential.helper store

git config --global http."sslVerify" false

~/.gitconfig

~/.git-credentials

当前工程目录有效：

git config --local credential.helper store

git config --local http."sslVerify" false

${workspace}/.git/.gitconfig

${workspace}/.git/.git-credentials

更多命令查看：

$ git config --help

file:///D:/software/Git/mingw64/share/doc/git-doc/git-config.html

附录：

.gitconfig

```text

[user]

    name = xywanhh

    email = xywanhh@qq.com

[credential]

    helper = store

[filter "lfs"]

    process = git-lfs filter-process

    required = true

    clean = git-lfs clean -- %f

    smudge = git-lfs smudge -- %f

[http]

    sslVerify = false

```

.git-credentials

```text

https://useremail:pwd@github.com

https://useremail:pwd@gitee.com/

```

## git clean

从命令执行的当前目录递归清除未跟踪的文件

清除前可以列出哪些文件

git clean -fd -n

git clean -nfd

git clean -f  清除文件

git clean -d  清除目录

git clean -fd

git clean -fx   remove ignored files

submodule也需要清除，需要再加个-f

git clean -dffx

git clean -d -f -fx

## 常见操作

1. 合并2个不同remote的branch

远程仓A: origin

远程仓B: remote

B dev --> A dev

git clone git\_A

git checkout dev

git pull

git remote -v

git remote add remote git\_B

git remote -v

git fetch remote （使用git fetch不会自动合并）

git checkout -b dev1 remote/dev

git branch -a

git checkout master

git merge dev1

(如果有comflic，解决后)

git commit -m "msg"

git push origin dev1

注意：

在git merge、git pull、git push的时候，可能会出现fatal: refusing to merge unrelated histories，原因是不同仓库的分支不相关，加上 --allow-unrelated-histories

2. clone代码

git clone [-b dev] git\_url

git checkout dev

git submodule update --init

(如果存在.gitmodules)

git submodule foreach git checkout dev

3. update代码

git remote remove origin

git remote add origin git\_new\_url

git clean -d -fx (git clean -d -ffx)  清除未git跟踪的文件

git reset --hard  清除modified状态的文件

git submodule foreach git reset --hard

git fetch -all  拉所有分支的更新 git fetch origin dev

git checkout dev

git reset --hard origin/dev  把index区跟工作区强制重置，保持和远程仓一致

(如果存在.gitmodules文件)

git submodule sync

git submodule foreach git clean -d -fx

git submodule foreach git fetch --all

git submodule foreach git checkout dev

git submodule foreach git reset --hard origin/dev

注意：

fatal: Peer's Certificate issuer is not recognized

git config --local http."sslVerify" false

## 常见问题

1. git Filename too long

原因：

一般在windows下出现，git允许创建4096长度的文件名，windows最多是260，git用的是旧版本的windows api。

解决：

git config --system core.longpaths true

git config --global core.longpaths true

2. git clean 无法清除submodule目录

原因：

git clean -d -fx只能递归清除工程未跟踪的文件（包含目录，.gitignore里的文件），子模块需要额外 -f

解决：

git clean -d -f -fx

https://stackoverflow.com/questions/61212/how-to-remove-local-untracked-files-from-the-current-git-working-tree

git clean -f :/ works as if you had run it in the root repo dir. See also later answers also accounting for submodules with git clean -ffxd

3. git remote remove/rm

Remove the remote named <name>. All remote-tracking branches and configuration settings for the remote are removed.

这里修改远程仓的同时，配置文件也会改。

java在多线程执行git fetch、git remote remove/add时，会有问题，有的线程会读取不到user.name,user.pwd，需要同步解决。

4. 在windows上调整目录名称大小写时没生效

原因：

Windows系统默认不区分大小写。

解决：

这里有坑，比如git上修改目录名称，远程库上显示修改成功，linux机器和远程库目录一致，windows本地机器目录异常，所以只是调整大小写，建议在linux系统上搞。