

Git使用教程

2017年11月21日 藏雪

查看、添加、提交、删除、找回，重置修改文件

git help <command> # 显示command的help

git show # 显示某次提交的内容 git show \$id

git add <file> # 将工作文件修改提交到本地暂存区

git add . # 将所有修改过的工作文件提交暂存区

git rm <file> # 从版本库中删除文件

git rm <file> --cached # 从版本库中删除文件，但不删除文件

git reset <file> # 从暂存区恢复到工作文件

git reset -- . # 从暂存区恢复到工作文件

git reset --hard # 恢复最近一次提交过的状态，即放弃上次提交后的所有本次修改

git revert <\$id> # 恢复某次提交的状态，恢复动作本身也创建次提交对象

git revert HEAD # 恢复最后一次提交的状态

查看文件diff

git diff <file> # 比较当前文件和暂存区文件差异 git diff
git diff <\$id1> <\$id2> # 比较两次提交之间的差异
git diff <branch1>..<branch2> # 在两个分支之间比较
git diff --staged # 比较暂存区和版本库差异
git diff --cached # 比较暂存区和版本库差异
git diff --stat # 仅仅比较统计信息

查看提交记录

git log git log <file> # 查看该文件每次提交记录
git log -p <file> # 查看每次详细修改内容的diff
git log -p -2 # 查看最近两次详细修改内容的diff
git log --stat #查看提交统计信息

tig

Mac上可以使用tig代替diff和log, brew install tig

Git 本地分支管理

查看、切换、创建和删除分支

git branch -r # 查看远程分支

git branch <new_branch> # 创建新的分支

git branch -v # 查看各个分支最后提交信息

git branch --merged # 查看已经被合并到当前分支的分支

git branch --no-merged # 查看尚未被合并到当前分支的分支

git checkout <branch> # 切换到某个分支

git checkout -b <new_branch> # 创建新的分支，并且切换过去

git checkout -b <new_branch> <branch> # 基于branch创建新的new_branch

git checkout \$id # 把某次历史提交记录checkout出来，但无分支信息，切换到其他分支会自动删除

git checkout \$id -b <new_branch> # 把某次历史提交记录checkout出来，创建成一个分支

分支合并

git merge <branch> # 将branch分支合并到当前分支

git merge origin/master --no-ff # 不要Fast-Foward合并，这样可以生成merge提交

Git暂存管理

git stash # 暂存

git stash list # 列所有stash

git stash apply # 恢复暂存的内容

git stash drop # 删除暂存区

Git远程分支管理

git pull # 抓取远程仓库所有分支更新并合并到本地

git pull --no-ff # 抓取远程仓库所有分支更新并合并到本地，不要快进合并

git fetch origin # 抓取远程仓库更新

git merge origin/master # 将远程主分支合并到本地当前分支

git push # push所有分支

git push origin master # 将本地主分支推到远程主分支

git push -u origin master # 将本地主分支推到远程(如无远程主分支则创建，用于初始化远程仓库)

git push origin <local_branch> # 创建远程分支， origin是远程仓库名

Git远程仓库管理

git remote -v # 查看远程服务器地址和仓库名称

git remote show origin # 查看远程服务器仓库状态

git remote add origin git@ github:robbin/robbin_site.git # 添加远程仓库地址

git remote set-url origin git@ github.com:robbin/robbin_site.git # 设置远程仓库地址(用于修改远程仓库地址) git remote rm <repository> # 删除远程仓库

创建远程仓库

git push -u origin master # 客户端首次提交

git push -u origin develop # 首次将本地develop分支提交到远程develop分支，并且track

git remote set-head origin master # 设置远程仓库的HEAD指向master分支

也可以命令设置跟踪远程库和本地库

git branch --set-upstream master origin/master

git branch --set-upstream develop origin/develop

常用命令介绍

目前测试环境需要本地绑定一下hosts

gitlab.example.com 192.168.1.42

克隆项目有两种方式http和ssh:

http方式: git clone http://192.168.1.42/credit/credit-util.git

clone默认是拉取master代码

如果想拉取分支代码则输入: git clone http://192.168.1.42/credit/credit-util.git -b 分支名

ssh方式:

如果52119是端口: git clone ssh://git@192.168.1.42:52119/credit/credit-util.git

如果是默认端: git clone git@192.168.1.42/credit/credit-util.git

输入自己的注册的用户名和密码后就可以拉取代码到本地。

可以通过git branch查看在哪个分支右面如图:

规范: 开发尽量避免在master开发, 每次项目拉取一个独立分支

创建新分支命令: git checkout -B 分支名称

新分支提交到服务器: git push --set-upstream origin zangxue-test3

```
Test@bogon ~$ git clone http://192.168.1.42/credit/credit-util.git
Cloning into 'credit-util'...
Username for 'http://192.168.1.42': test
Password for 'http://test@192.168.1.42':
```

```
bogon:master zsnow$ git branch
  master
* zangxue-test
bogon:master zsnow$
```

每次提交一个文件需要下面3步

1.**git add** 不跟文件则代表文件夹内有变化的加入到暂存区, 跟文件名如 **git add filename** 则只把filename提交到暂存区, 只想提交有变动的文件使用 **git add -u**。

2.**git commit -m "提交备注"** 把缓存区文件提交到本地库

3.**git push** 把本地文件提交到gitlab服务器端。如果想强制提交**git push --force** 慎用!!

小技巧: 如果只是修改文件, 则可以直接git commit -m '提交备注'

如果需要删除文件则:

1.**git rm 文件名**

2.**git commit -m "提交备注"** 把缓存区文件提交到本地库

3.**git push** 把本地文件提交到gitlab服务器端。

通过**git show**命令可以查看本次提交了的具体内容, 也可以通过**git log -p**查看

meger命令操作

1、本地 merge远程 master代码

1. git pull
2. git merge origin/master

2、master merge 远程代码

1. git pull
2. git merge origin/zangxue-test #将服务器上面的 zangxue-test分支 merge到master上
3. git push

想要分支与master最新版本相同则使用：

1. git fetch --all
2. git reset --hard origin/master #把当前工作空间恢复到远程服务器master最近一次节点。
3. git push --force #强制推送

删除文件操作

1. git rm zangxue
2. git commit -m '提交备注'
3. git push

代码提交日志查看

git show 能看出最近一次提交具体变化

git log 能看出最近多次提交日志， git log -p 看具体信息

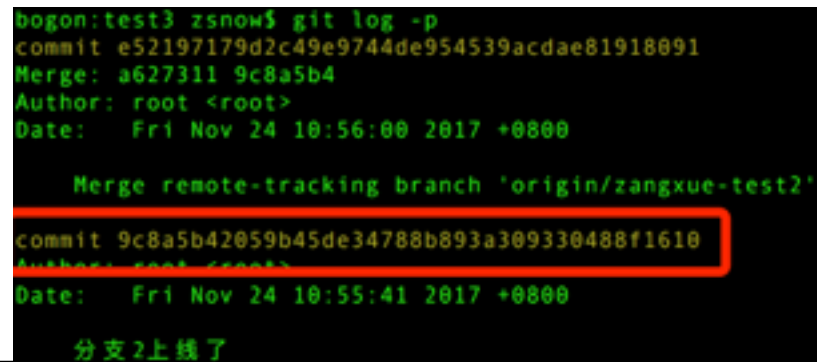
回滚

使用git log 查看到需要回归的 commit号

使用： git reset --hard commit号 就可以回滚到指定版本。

如图： git reset --hard 9c8a5b42059b45de34788b893a309330488f1610

就会回滚到 9c8a5b42059b45de34788b893a309330488f1610这次的提交。



```
bogon:test3 zsnw$ git log -p
commit e52197179d2c49e9744de954539acdae81918091
Merge: a627311 9c8a5b4
Author: root <root>
Date: Fri Nov 24 10:56:00 2017 +0800

    Merge remote-tracking branch 'origin/zangxue-test2'

commit 9c8a5b42059b45de34788b893a309330488f1610
Author: root <root>
Date: Fri Nov 24 10:55:41 2017 +0800

    分支2上线了
```

查看本地账号：

git config --list

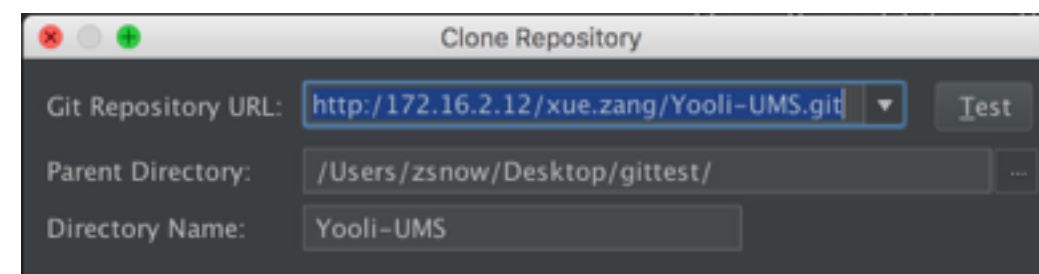
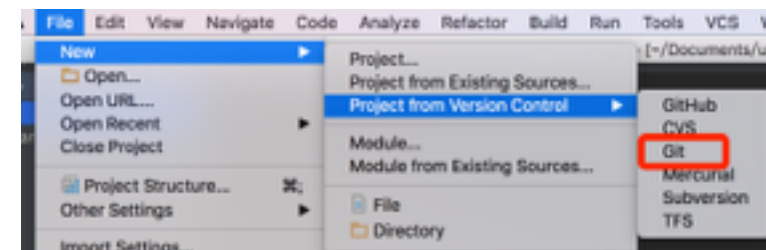
IDEA--Git

一、配置获取远程项目

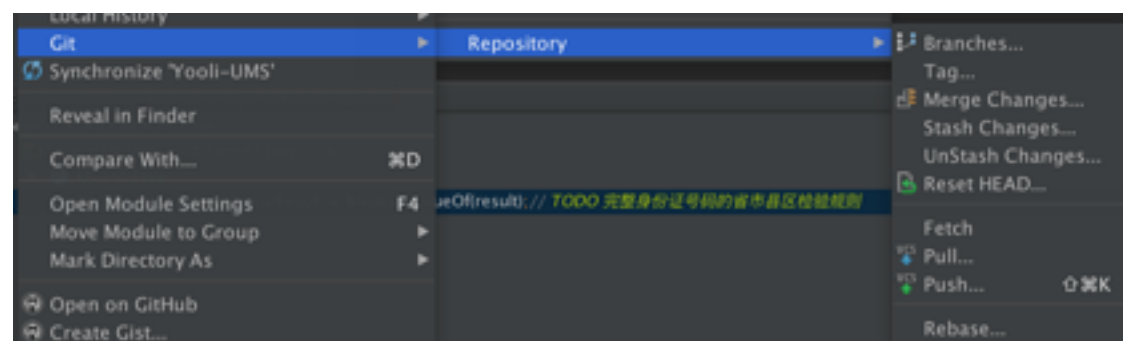
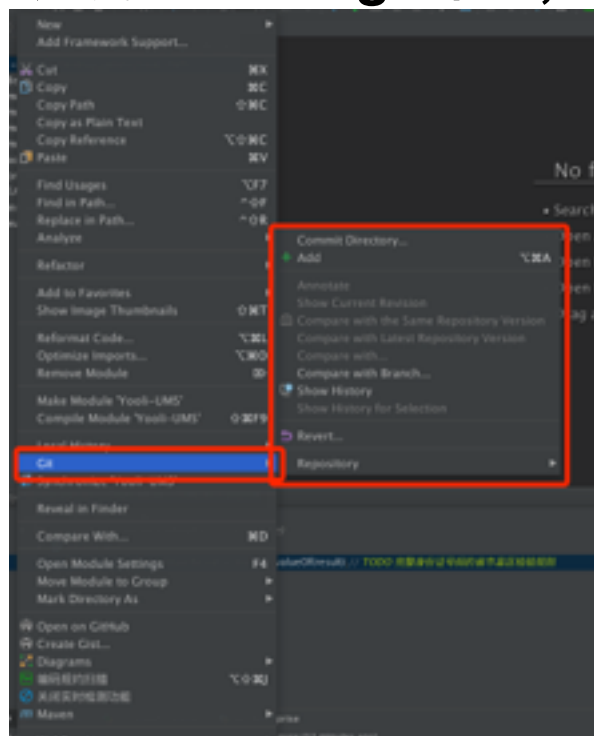
File-->New-->Project From Version Control-->Git

输入对应的git地址即可，点击clone就可以拉取到指定位置。

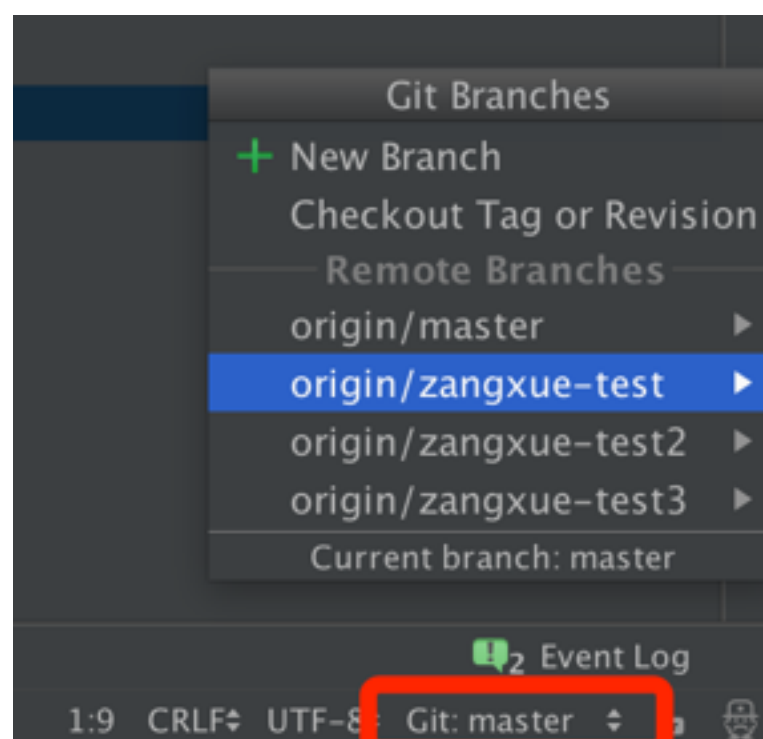
第一次使用需要输入用户名和密码



二、使用集成的git命令，在文件夹上鼠标右键就可以找到。



三、查看当前工作分支



Yooli

Yooli-Finance

Yooli-Asset

d