

# git 常用命令详解



作者 子疯zp (/u/82a05afc4a68) (+关注)

2017.04.11 17:33 字数 2144 阅读 138 评论 0 喜欢 0

(/u/82a05afc4a68)

## 一、Git 命令初识

在正式介绍Git命令之前，先介绍一下Git 的基本命令和操作，对Git命令有一个总体的认识

示例：从Git 版本库的初始化，通常有两种方式：

1) git clone：这是一种较为简单的初始化方式，当你已经有一个远程的Git版本库，只需要在本地克隆一份

例如：git clone git://github.com/someone/some\_project.git some\_project

上面的命令就是将'git://github.com/someone/some\_project.git'这个URL地址的远程版本库，完全克隆到本地some\_project目录下

2) git init 和 git remote：这种方式稍微复杂一些，当你本地创建了一个工作目录，你可以进入这个目录，使用'git init'命令进行初始化；Git以后就会对该目录下的文件进行版本控制，这时候如果你需要将它放到远程服务器上，可以在远程服务器上创建一个目录，并把可访问的URL记录下来，此时你就可以利用'git remote add'命令来增加一个远程服务器端，

例如：git remote add origin git://github.com/someone/another\_project.git

上面的命令就会增加URL地址为'git://github.com/someone/another\_project.git'，名称为origin的远程服务器，以后提交代码的时候只需要使用 origin别名即可

## 二、Git 常用命令

### 1、远程仓库相关命令

检出仓库：\$ git clone git://github.com/jquery/jquery.git

查看远程仓库：\$ git remote -v

添加远程仓库：\$ git remote add [name] [url]

删除远程仓库：\$ git remote rm [name]

修改远程仓库：\$ git remote set-url --push [name] [newUrl]

拉取远程仓库：\$ git pull [remoteName] [localBranchName]

推送远程仓库：\$ git push [remoteName] [localBranchName]

如果想把本地的某个分支test提交到远程仓库，并作为远程仓库的master分支，或者作为另外一个名叫test的分支，如下：

\$git push origin test:master // 提交本地test分支作为远程的master分支

\$git push origin test:test // 提交本地test分支作为远程的test分支

## 2、分支(branch)操作相关命令

查看本地分支: `$ git branch`

查看远程分支: `$ git branch -r`

创建本地分支: `$ git branch [name]` ----注意新分支创建后不会自动切换为当前分支

切换分支: `$ git checkout [name]`

创建新分支并立即切换到新分支: `$ git checkout -b [name]`

删除分支: `$ git branch -d [name]` ---- -d选项只能删除已经参与了合并的分支, 对于未有合并的分支是无法删除的。如果想强制删除一个分支, 可以使用-D选项

合并分支: `$ git merge [name]` ----将名称为[name]的分支与当前分支合并

创建远程分支(本地分支push到远程): `$ git push origin [name]`

删除远程分支: `$ git push origin :heads/[name]` 或 `$ git push origin :[name]`

创建空的分支: (执行命令之前记得先提交你当前分支的修改, 否则会被强制删干净没得后悔)

`$git symbolic-ref HEAD refs/heads/[name]`

`$rm .git/index`

`$git clean -fdx`

## 3、版本(tag)操作相关命令

查看版本: `$ git tag`

创建版本: `$ git tag [name]`

删除版本: `$ git tag -d [name]`

查看远程版本: `$ git tag -r`

创建远程版本(本地版本push到远程): `$ git push origin [name]`

删除远程版本: `$ git push origin :refs/tags/[name]`

合并远程仓库的tag到本地: `$ git pull origin --tags`

上传本地tag到远程仓库: `$ git push origin --tags`

创建带注释的tag: `$ git tag -a [name] -m 'yourMessage'`

## 4、子模块(submodule)相关操作命令

添加子模块：\$ git submodule add [url] [path]

如：\$git submodule add git://github.com/soberh/ui-libs.git src/main/webapp/ui-libs

初始化子模块：\$ git submodule init ----只在首次检出仓库时运行一次就行

更新子模块：\$ git submodule update ----每次更新或切换分支后都需要运行一下

删除子模块：（分4步走哦）

1) \$ git rm --cached [path]

2) 编辑“.gitmodules”文件，将子模块的相关配置节点删除掉

3) 编辑“.git/config”文件，将子模块的相关配置节点删除掉

4) 手动删除子模块残留的目录

## 5、忽略一些文件、文件夹不提交

在仓库根目录下创建名称为“.gitignore”的文件，写入不需要的文件夹名或文件，每个元素占一行即可，如

target

bin

\*.db

## 附：Git 常用命令速查

git branch 查看本地所有分支

git status 查看当前状态

git commit 提交

git branch -a 查看所有的分支

git branch -r 查看远程所有分支

git commit -am "init" 提交并且加注释

git remote add origin git@192.168.1.119:ndshow

git push origin master 将文件给推到服务器上

git remote show origin 显示远程库origin里的资源

git push origin master:develop

git push origin master:hb-dev 将本地库与服务器上的库进行关联

git checkout --track origin/dev 切换到远程dev分支

git branch -D master develop 删除本地库develop

git checkout -b dev 建立一个新的本地分支dev

git merge origin/dev 将分支dev与当前分支进行合并

git checkout dev 切换到本地dev分支

git remote show 查看远程库

git add .

git rm 文件名(包括路径) 从git中删除指定文件

git clone git://github.com/schacon/grit.git 从服务器上将代码给拉下来

git config --list 看所有用户

git ls-files 看已经被提交的

git rm [file name] 删除一个文件

git commit -a 提交当前repos的所有的改变

git add [file name] 添加一个文件到git index

git commit -v 当你用 -v 参数的时候可以看commit的差异

git commit -m "This is the message describing the commit" 添加commit信息

git commit -a -a是代表add, 把所有的change加到git index里然后再commit

git commit -a -v 一般提交命令

git log 看你commit的日志

git diff 查看尚未暂存的更新

git rm a.a 移除文件(从暂存区和工作区中删除)

git rm --cached a.a 移除文件(只从暂存区中删除)

git commit -m "remove" 移除文件(从Git中删除)

git rm -f a.a 强行移除修改后文件(从暂存区和工作区中删除)

git diff --cached 或 \$ git diff --staged 查看尚未提交的更新

git stash push 将文件给push到一个临时空间中

git stash pop 将文件从临时空间pop下来

-----

git remote add origin git@github.com:username/Hello-World.git

git push origin master 将本地项目给提交到服务器中

-----

git pull 本地与服务器端同步

-----

git push (远程仓库名) (分支名) 将本地分支推送到服务器上去。

git push origin server fix:awesome branch

-----

git fetch 相当于是从远程获取最新版本到本地，不会自动merge

git commit -a -m "log\_message" (-a是提交所有改动，-m是加入log信息) 本地修改同步至服务器端：

git branch branch\_0.1 master 从主分支master创建branch\_0.1分支

git branch -m branch\_0.1 branch\_1.0 将branch\_0.1重命名为branch\_1.0

git checkout branch\_1.0/master 切换到branch\_1.0/master分支

du -hs

git branch 删除远程branch

git push origin:branch\_remote\_name

git branch -r -d branch\_remote\_name

-----

初始化版本库，并提交到远程服务器端

mkdir WebApp

cd WebApp

git init本地初始化

touch README

git add README添加文件

```
git commit -m 'first commit'
```

```
git remote add origin git@github.com:daixu/WebApp.git
```

增加一个远程服务器端

上面的命令会增加URL地址为'git@github.com:daixu/WebApp.git', 名称为origin的远程服务器库, 以后提交代码的时候只需要使用 origin别名即可

IOS (/nb/11293657)

举报文章 © 著作权归作者所有



子疯zp (/u/82a05afc4a68) ♂

写了 13117 字, 被 35 人关注, 获得了 26 个喜欢

(/u/82a05afc4a68)

+ 关注

滴水穿石非一日之功, 冰冻三尺非一日之寒!

帅哥美女, 能不能赏瓶矿泉水~ 😊

赞赏支持

喜欢 (/sign\_in?utm\_source=desktop&utm\_medium=not-signed-in-like-button) 0







更多分享

(http://cwb.assets.jianshu.io/notes/images/11221464/w



登录后发表评论

(/sign\_in?utm\_source=desktop&utm\_medium=not-signed-in-comment-form)

↑

🔗

评论

智慧如你, 不想发表一点想法 (/sign\_in?utm\_source=desktop&utm\_medium=not-signed-in-nocomments-text)咩

~

