Git命令

1. 创建仓库

git init repo

1. 克隆仓库

git clone ssh://git@git.xesv5.com:10088/wxgo/push-irc.git

1. 设置用户名和邮箱

git config --global user.name “zhangxiaodong2”

git config --global user.email “zhangxiaodong2@100tal.com”

1. 查看分支

git branch

1. 查看状态

git status

1. 拉取代码

拉取master分支：git pull origin master

拉取所有分支：git fetch

1. 查看改动

* 尚未缓存的改动：git diff
* 查看已缓存的改动： git diff --cached
* 查看已缓存的与未缓存的所有改动：git diff HEAD
* 显示摘要而非整个 diff：git diff --stat

1. 添加文件夹/文件

git add folder or file

1. 提交

git commit -a -m ‘注释’

1. 取消已缓存的内容

git reset HEAD xx.go

执行 git reset HEAD 以取消之前 git add 添加，但不希望包含在下一提交快照中的缓存。

git reset --hard HEAD

工作目录中所有未提交的内容清空

1. 撤销提交

git revert HEAD

撤销最近的一次提交

git revert HEAD^

撤销上上次提交

git revert commit\_id

用新的一次commit回滚之前的某次commit

git reset --hard commit\_id

直接删除指定的commit

1. 删除文件

删除工作区文件：git rm file

删除工作区、暂存区文件：git rm -f file

从暂存区移除，保留在工作区：git rm --cached file

1. 移动或者重命名

git add xx.go

git mv xx.go xxxx.go

1. 推送

推送到远程仓库：git push origin master

强制推送到远程仓库：git push origin master -f

1. 合并

dev分支合并到当前master分支：git merge dev

如果有冲突，解决具体文件的冲突

git add 冲突文件

git merge continue

//git commit -a -m “解决xx功能冲突”

git push origin master

1. 分支

列出分支：git branch

切换分支：git checkout branch\_name

切换并新建分支：git checkout -b branch\_name

新建分支：git branch new\_branch

删除本地分支：git branch -d branch\_name

删除远程分支：git push origin --delete branch\_name

1. 提交历史

详细信息：git log

简略信息：git log --oneline

显示合并信息：git log --oneline --graph

逆向显示所有日志：git log --reverse --oneline

根据时间查询：git log --oneline --before={2018-12-07} --after={2018-12-05}

选项：

--no-merges:因此合并

--author：作者

1. Tag

在release分支上执行（-a：加注释）：git tag v1.0

推送tag：git push origin v1.0

推送所有tag：git push origin --tags

删除tag：git tag -d v1.0

查看修改内容：git show v1.0

1. Rebase

git checkout my\_branch

git rebase master

git rebase --continue

终止rebase：Git rebase --abort

1. Stash

压栈未提交的修改：git stash

弹出未提交的修改：git stash pop

查看现有stash列表：git stash list

移除stash：git stash drop stash\_name

查看指定stash的diff：git stash show stash\_name

1. Cherry-pick

将单一commit引入当前分支：git cherry-pick commit\_id

查看commit id： git log

如果有冲突，解决冲突，git commit