

常用 git 命令以及处理技巧

基础常用命令

```
1 # 克隆
2 git clone git@github.com:fengcms/fengcms.git
3
4 # 拉
5 git pull
6
7 # 添加某一个文件
8 git add "src/xxx.js"
9
10 # 添加所有文件，严禁使用 `git add .` 这样的命令！这完全是不同的含义！
11 git add -A
12
13 # 暂存—通过 vim 编辑器
14 git commit -a
15
16 # 暂存—通过命令行
17 git commit -m "what i've done"
18
19 # 推
20 git push
21
22 # 重置未提交文件修改
23 git checkout src/xxx.js
24
25 # 重置未提交文件夹
26 git checkout src/xxx/
27
28 # 查看所有分支
29 git branch -a
30
31 # 切换分支
32 git checkout feature_1_3_0
```

基础状态查看类命令

```
1 # 查看当前修改
2 git status
3
4 # 查看当前修改详情
5 git diff
```

```
6
7 # 查看最近几次的提交，可以取得 commit 哈希值
8 git log -2
9
10 # 查看某次提交具体干了啥
11 # 哈希值通过上个命令获取，不用复制全部，复制最前面十几位就可以了。
12 git show [commit]
```

非常用，但必须掌握的命令

```
1 # 以当前所在分支代码为基础，创建一个新分支
2 git checkout -b feature_1_3_1
3
4 # 将当前分支推送到远程
5 git push --set-upstream origin feature_1_3_1
6
7 # 删除本地分支
8 git branch -d feature_1_3_0
9
10 # 强制删除本地分支
11 git branch -D feature_1_3_0
12
13 # 删除远程分支
14 git push origin --delete feature_1_3_0
15
16 # 更新远程分支信息
17 # 常用语远程已经删除分支，但本地依然缓存了远程分支的情况
18 git fetch origin --prune
19
20 # 合并某分支到当前分支
21 git merge feature_1_3_0
22
23 # 重置到最近一次提交的代码
24 # 不会删除你新添加的文件
25 git reset --hard
26
27 # 重置到某次提交的代码
28 # 可以从新往旧恢复，可以从旧恢复到新
29 git reset [commit]
```

git 命令很多，也不容易记忆，善用 ctrl + r 快捷键，可以在命令行中进行历史命令搜索。因此，只要记住关键词就可以了。经过测试，windows 下的 git-bash 命令行工具也是支持这种搜索的。

其他 git 使用技巧

无密码操作

我们可以通过设置 ssh 秘钥来进行无密码操作 git 仓库。但在 windows 上 ssh 秘钥需要配置并长期开启一个软件(IDE 另说), 因此, 我们可以通过另一个方式来实现无密码操作。

在需要操作的项目文件夹下, 找到 **.git/config** 文件, 大概内容如下:

```
1 [core]
2     repositoryformatversion = 0
3     filemode = true
4     bare = false
5     logallrefupdates = true
6 [remote "origin"]
7     url = http://192.168.1.66/Front-end/waybill-fe.git
8     fetch = +refs/heads/*:refs/remotes/origin/*
```

我们找到其中的 **url** 参数, 调整为如下的格式, 就可以免输密码了。

```
1 url = http://__USERNAME__:__PASSWORD__@192.168.1.66/Front-end/waybill-f
  e.git
```

注意, 这样做会比较容易泄露密码, 请自行确定密码安全等级。此方法常用于在 windows 系统上, 操作别人的电脑时别人使用了 ssh 但你又不知道如何开启他的配置, 但此时又必须提交代码的情况下, 临时切换为自己的用户名密码的方法。当然, 方法给了, 怎么用是自己考虑的事情。

调整默认编辑器为 vim

在一些 linux 系统中, 默认的文本编辑器是 nano。这样我们在 `git commit -a` 命令提交代码的时候, 就很不方便 (或者认为很方便?), 因此我们想要重置为 vim 编辑器, 命令如下:

```
1 git config --global core.editor vim
```

或者, 直接编辑 **~/.gitconfig** 文件, 添加 [core] 节点信息如下, 比如我的:

```
1 [user]
2     email = web@fengcms.com
3     name = fungleo
4 [core]
5     editor = vim
```

小结

命令行操作可能没有图形界面操作方便, 但是命令行操作是高度统一的, 不会因为对方使用的操作系统, GUI 软件亦或是其他因素发生变化。掌握命令行操作后, 可以跨系统跨平台的操作和解决问题。因此希望大家能够努力掌握一下, 亦或是收藏这些命令, 在需要的时候可以有一本《肘后方》。