

Git常用命令

Prepare By Vincent
Q3, 2017



如何提交代码 (1/2 - 基于克隆仓库)

- 克隆远程仓库
`git clone <repo>`
- 完成代码任务后, 把所有修改(包括新文件)加入
`git add -all`
- 此时可以查看一下提交状态(非必要)
`git status`
- 首先提交到本地仓库 (comment – 说明这些提交的修改)
`git commit -m “comment”`
- 通常要先合并远程仓库的提交到本地 origin-仓库名 master-分支名(master指主分支), 留意是否有冲突
`git pull origin master`
- 没有冲突的话最后推送至远程仓库
`git push origin master`



如何提交代码 (1/2 – 新建仓库)

- 新建仓库

`git init <dir>` (或进入新建的目录 => `git init`)

- 设置过滤条件 新建文件 `.gitignore`

`touch .gitignore` (编辑过滤条件)

- 提交所有代码

`git add -all`

- 通常要需要查看准备要提交的文件是否正确

`git status`

- 先提交到本地 (如果没有新文件要提交可增加参数`-a`, 意思是同时增加修改并提交, 已执行过 `git add -all`即可省去 `-a`)

`git commit -a -m "comment"` (或 `git commit -m "msg"`)

- 设置远程仓库并推送

`git remote add origin <repo>; git push origin master`



如何设置过滤

- 工程项目会产生很多非源代码中间文件，不应该提交在 **repo** 目录下面 创建 **.gitignore** 文件，并需要提交

```
$ cat .gitignore
```

```
## 后面 / 指目录，表示忽略整个工程项目下面
```

```
## 所有子目录下的 build 目录
```

```
build/
```

```
## 通配符, 忽略后缀为 .log 的所有文件
```

```
*.log
```

```
## 前面 / 指当前工程根目录下面的gen目录,不包含子目录
```

```
/gen
```

```
## 指定忽略某个文件
```

```
.config.json
```



常用命令列表 (1/2)

Case	Command Line	Comment
如何初始化仓库	<code>git init</code>	先进入新建空目录
如何克隆远程仓库	<code>git clone <repo></code>	<code><repo></code> 指仓库地址
如何增加新文件	<code>git add --all</code>	除了新文件还包括所有修改
如何查看提交状态	<code>git status</code>	提交信息/是否已提交等
如何查看远程仓库	<code>git remote -v</code>	<code>-verbose</code> 远程仓库信息
如何设置远程仓库	<code>git remote add origin <repo></code>	<code>origin</code> 是远程仓库名(常用名)
如何修改远程仓库	<code>git remote set-url origin <repo></code>	
如何提交代码	<code>git commit -m "comments"</code>	<code>comments</code> 指此次提交的说明
如何推送到远程仓库	<code>git push origin master</code>	地本提交推送到远程
如何合并远程提交到本地	<code>git pull origin master</code>	



常用命令列表 (2/2)

Case	Command Line	Comment
如何移除已提交文件	<code>git rm <fn></code>	fn 指文件名
如何移除已提交目录	<code>git rm <dir></code>	<dir> 指目录名
如何重置未提交修改	<code>git reset --hard</code>	修改后要重置为初始状态
如何取消当前提交	<code>git reset --soft HEAD~</code>	重置到未提交状态, 保留修改
如何创建远程仓库	<code>git init --bare <repo></code>	登录至远程仓库所在服务器
如何强制推送至远程仓库	<code>git push origin master -f</code>	-force 加此参数相当于本地仓库完全覆盖远程仓库
如何创建新分支	<code>git branch <branch-name></code>	
基于现有分支创建分支	<code>git checkout -b <branch-name> <master></code>	master 指现在分支名, 或 commit-id
如何显示所有分支	<code>git branch -a</code>	

