




 holyzfy / frontend_guidelines


 Watch


 Star


 Fork


 Code

 Issues 3

 Pull requests 0

 Wiki


 Pulse

 Graphs

Branch: master

frontend_guidelines / git.md

[Find file](#) [Copy path](#)

 holyzfy Update git.md

4958668 on 3 Feb

1 contributor

103 lines (68 sloc) 2.63 KB

Raw

Blame

History







git分支管理策略

常用分支

1. master : 主分支, 存放成熟的代码, 可以随时发布, 请不要强制push
2. dev : 平时在这个分支上开发
3. fix : 当发现 master 上分支的代码有bug时, 请在此分支上开发, 测试通过后再合并到 master

约定 dev , fix 是本地分支, 不要push到远程仓库

一次完整的操作流程

1. git clone https://github.com/holyzfy/frontend_guidelines.git
2. 切换并创建分支 dev :

```
git checkout -b dev
```

-b参数表示创建分支, 后续再切换到 dev 分支时请去掉此参数;
3. 一些开发动作, 创建, 编辑文件...
4. 把上一步的变更内容添加到暂存区;

```
git add .
```
5. 提交到本地仓库:

```
git commit -m '请简要描述本次的提交'
```
6. 准备合并到master分支前先拉取更新, 看别人是否向 master 分支推送过代码:

```
git checkout master  
git pull
```
7. 如果拉取到更新, 就回到 dev 分支, 把 master 分支的新内容合并到 dev :

```
git checkout dev  
git merge master
```

如果没有发生冲突, git会自动合并, 这时候会打开vim的编辑界面, 提示你写提交说明:

```
Merge branch 'master' into dev  
# Please enter a commit message to explain why this merge is necessary,  
# especially if it merges an updated upstream into a topic branch.
```

