

Hubery_Jun

博客园

首页

新随笔

联系

订阅

管理

随笔 - 171 文章 - 1 评论 - 4 阅读 - 18万

Git 开发流程

1. 流程

- `develop` 开发分支：开发人员每天都需要拉取/提交最新代码的分支
- `test` 测试分支：开发人员开发完并自测通过后，发布到测试环境的分支
- `release` 预发布分支：测试环境测试通过后，将测试分支的代码发布到预发环境的分支（这个得看公司支不支持预发环境，没有的话就可以不采用这条分支）
- `master` 线上分支：预发环境测试通过后，运营/测试会将此分支代码发布到线上环境
- `hotfix` 分支：在 `master` 发现新的 `bug` 时，需要创建一个 `hotfix`，完成后，合并到 `master` 和 `develop` 分支

大致流程

- 开发人员每天都需要拉取、提交最新的代码到 `develop` 分支
- 开发完毕后，开始 **集成测试**，测试无误后提交到 `test` 分支，并发布到测试环境，交由测试人员测试
- 测试环境通过后，发布到 `release` 分支上，进行预发布环境测试
- 预发环境通过后，发布到 `master` 分支上并打上标签 `tag`
- 如果线上分支出现 `bug`，这时开发者应该基于预发布（没有预发布环境就用 `master` 分支），新建一个 `bug` 分支用来临时解决 `bug`，处理完成后申请合并到预发布分支（好处是不会影响正在开发中的功能）

1.1 克隆指定分支

场景：要求使用非master分支（如next分支）进行开发

```
git clone -b next "git地址"
```

1.2 优雅的在分支上开发

1、克隆到本地的分支为 `next` 分支，但是不能直接在上面开发，而是需要另外创建分支：

```
# 创建并切换到新的分支 next-dev, 其中 add_login 为分支说明（可不加）
git checkout -b next-dev/add_login
```

2、在 `next-dev` 分支上开发完后，将代码提交到本地仓库：

```
git add . # "."的意思就是保存添加所有修改到暂存区
git commit -m "注释" # 暂存区中的修改提交到本地仓库，commit 规范可参考 6.1 commit 规范
```

3、将 `next-dev` 分支合并到 `next` 分支：

```
# 先切换到 next 分支
git checkout next # 切换之前一定要先执行第二步

# 从远端拉取最新的代码，当你合并分支的时候，可能其他同事又提交了新的内容，所以最好先拉取一次最新的代码
git pull origin next

# 合并（如果合并发现有冲突，可以使用可视化工具查看冲突的代码，如：pycharm、BCompare4
```

公告

昵称：Hubery_Jun
园龄：3年7个月
粉丝：8
关注：2
+加关注

2022年6月						
日	一	二	三	四	五	六
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

Web 框架(23)
Django(18)
项目(16)
DRF(12)
Vue(9)
Python 进阶(9)
MySQL(9)
Spark(8)
Python模块(7)
hive(6)
更多

随笔分类

Git(1)
Linux(16)
Python(85)
大数据系列(23)
工具(5)
每日一技(7)
前端(17)

```
git merge next-dev

# 将本地代码推送到远程
git push origin next:next      # ":"前面的是本分支的名字, ":"后面的是远程分支的名字
```

注意：合并之前先 `commit`，合并之前先 `git pull`

2. 常用命令

```
# ssh 方式需要配置 SSH Key, https 需要每次都输入用户名和密码
$ cd /d/python-code/
$ git clone git@github.com:wangy8961/flask-vuejs-madblog.git

git init          # 初始化本地仓库
git status        # 查看状态
git add .         # 添加到暂存库
git commit -m 'xxx' # 提交
git push -u origin master # 推送到远端

git remote add origin "HTTPS地址" # 关联远程仓库

# 从远端拉取最新代码到本地
git pull          # 自动合并
git fetch         # 不合并

# 分支
git checkout -b feat/add-artical # 创建新的分支 feat, 并切到新分支, add-artical 为对这个分支的描述
git branch -d dev # 删除 dev 分支
```

3. 拉取最新代码

```
# 查看已关联的仓库
$ git remote -v
origin https://gitee.com/hubery_jun/flask-vuejs-madblog (fetch)
origin https://gitee.com/hubery_jun/flask-vuejs-madblog (push)

# 类似于 git pull, 也是用于拉取最新代码
$ git fetch
# 或拉取指定的远程主机上的分支, 如 origin 上的 master
$ git fetch origin master
```

git fetch 与 git pull 的区别

- `git fetch` :
 - 远端跟踪分支：可以更改远端跟踪分支
 - 拉取：会将数据拉取到本地仓库，但是不会自动合并或修改当前的工作
 - `commitID`：本地库中 `master` 的 `commitID` 不变，还是等于 1
- `git pull` :
 - 远端跟踪分支：无法对远端跟踪分支操作，必须先切回到本地分支然后创建一个新的 `commit` 提交
 - 拉取：从远处获取最新版本，并合并到本地，会自动合并或修改当前的工作
 - `commitID`：本地库中 `master` 的 `commitID` 发生改变，变成了 2

用法

1、git pull

```
git pull <远程主机名> <远程分支名>:<本地分支名>

# 将远端主机 origin 的 master 分支拉下来, 与本地 dev 分支合并, 若未指定本地分支 (那么将与本地当前分支合并)
git pull origin master:dev

# 用 git fetch 表示
git fetch origin master:dev
git merge dev
```

2、git fetch

数据库(16)
学习链接(1)

随笔档案

2022年1月(1)
2021年12月(15)
2021年11月(1)
2021年10月(3)
2021年9月(1)
2021年8月(3)
2021年6月(1)
2021年5月(3)
2021年4月(4)
2021年2月(1)
2021年1月(1)
2020年10月(4)
2020年9月(42)
2020年8月(1)
2020年7月(1)
更多

相册

img(1)

阅读排行榜

- Linux 之 .bashrc 文件作用(61760)
- 解决 Ajax 发送 post 请求出现 403 Forbidden 的三种方式(33936)
- Python 列表切片(9991)
- Django 之验证码实现(6940)
- 使用 Python 批量将数据插入到 ES中(5723)

评论排行榜

- Django 之验证码实现(2)
- 【flask + vue 前后端分离博客】使用 axios 访问接口 (二) (1)
- Celery 基本使用(1)

推荐排行榜

- Linux 之 .bashrc 文件作用(4)
- Git 开发流程(1)
- Linux 之添加系统环境变量(1)
- uWSGI + Nginx + Django 部署(1)
- Celery 基本使用(1)

最新评论

- Re: Celery 基本使用
写得赞，但主要还是celery太难了。此国产分布式函数调度框架，从用法调用难度，用户所需代码量，超高并发性，qps控频精确程度，支持的中间件类型，任务控制方式，稳定程度等19个方面全方位超过ce...
--北风之神0509
- Re: 【flask + vue 前后端分离博客】使用 axios 访问接口 (二)
后端怎么接受？
--wangdachui2020
- Re: Django 之验证码实现
@ huhuahuhu有没有代码，发来瞧瞧...
--Hubery_Jun
- Re: Django 之验证码实现

目录
导航

```
# 方法一
git fetch origin master # 从远端 origin 仓库的 master 分支拉取最新代码
git log -p master.. origin/master # 比较本地的仓库和远端仓库的区别
git merge origin/master # 将远端拉取的最新代码合并到本地仓库，远端和本地的合并

# 方法二
git fetch origin master:temp # 从远端 origin 仓库的 master 分支拉取最新代码到本地并在本地创建一个新的分支 temp
git diff # 比较 master 分支和 temp 分支的不同
git merge temp # 合并 temp 分支到 master 分支
git branch -d temp # 删除 temp 分支
```

注意： `git fetch` 更安全也更符合实际要求，可以在合并前，先查看更新情况，根据实际情况再决定是否合并

有一个小问题，，，
手动生成验证码的方法，相应到前端页面上，第一次请求页面的时候，验证码为空，只有点击验证码区域产生请求才会响应到页面上

目录
导航

--huhuahuhu

4. git push 常用用法

一般形式：

```
git push <远程主机名> <本地分支名> <远程分支名>

# origin 为远程主机名，将本地 next 分支推送到远程 next 分支
git push origin next:next
```

省略远程分支：

```
# 表示将本地分支推送到与之存在追踪关系的远程分支（通常两者同名），如果该远程分支不存在，则会被新建
git push origin master
```

如果省略本地分支名，则表示删除指定的远程分支，因为这等同于推送一个空的本地分支到远程分支，等同于 `git push origin --delete master`：

```
git push origin : refs/for/master
```

如果当前分支与远程分支存在追踪关系，则本地分支和远程分支都可以省略，将当前分支推送到 `origin` 主机的对应分支：

```
git push origin
```

如果当前分支只有一个远程分支，那么主机名都可以省略，形如 `git push`，可以使用 `git branch -r`，查看远程的分支名：

```
git push
```

5. 分支管理

```
git checkout -b dev # 创建并切换到分支
git branch -d dev # 删除分支
git branch # 查看当前分支
```

6. 代码提交

添加到存储库、合并分支

```
git add . # 添加所有
git commit -m 'xxxx' # 提交
git checkout master # 切换到主分支
git merge dev # 合并 dev 分支到 master 主分支
git branch -d dev # 删除 dev 分支
git branch # 查看当前分支
```

6.1 commit 规范

`commit` 的内容也应该遵守规范，一般来说是

- 1. `fix:xx`：表示修改了XX代码
- 2. `feat:xx`：新增了XX需求

- 3. style:xx : 修改了部分的样式
- 4. delete:xx : 删除了某些无用的部分

推送到远程

```
git push -u origin master      # 推送到远端 master
```

注意：合并分支必须先切换分支，删除分支不是必须的！但是流程最好是：**先创建分支 ----> 合并分支 ----> 提交 ----> 删除分支 ----> 创建分支（第二天）**

打标签

```
git tag v0.1      # 打标签 v0.1
git tag           # 查看所有标签
git show v0.1     # 查看 v0.1 标签（内容）

# 1. 同步单个标签
$ git push origin v0.1

# 2. 同步所有标签
$ git push --tags

# 或者：
$ git push origin --tags
```

分类: 工具

标签: Git

好文要顶

关注我

收藏该文

Hubery_Jun

关注 - 2

粉丝 - 8

+加关注

1

0

« 上一篇: [如何在 Windows 上管理多个 Python 版本](#)
» 下一篇: [Apscheduler 定时任务](#)

posted @ 2020-09-03 15:50 Hubery_Jun 阅读(1667) 评论(0) 编辑 收藏 举报

[刷新评论](#) [刷新页面](#) [返回顶部](#)

登录后才能查看或发表评论，立即 [登录](#) 或者 [逛逛](#) 博客园首页

编辑推荐:

- 在 4GB 物理内存的机器上，申请 8G 内存会怎么样？
- 文字轮播与图片轮播？CSS 不在话下
- 技术管理者的困惑——技术与管理应该如何平衡？
- 深入探究 MinimalApi 是如何在 Swagger 中展示的
- 原理：C++为什么一般把模板实现放入头文件

最新新闻:

- 前阿里智能网卡负责人蒋晓维加入大禹智芯，任职首席科学家
 - 谷歌将Pi计算到后100万亿位数
 - Visual Studio Code 1.68 发布
 - TIOBE 6 月榜单：C++ 即将超越 Java
 - 华为管理如何“拧麻花”？
- » 更多新闻...