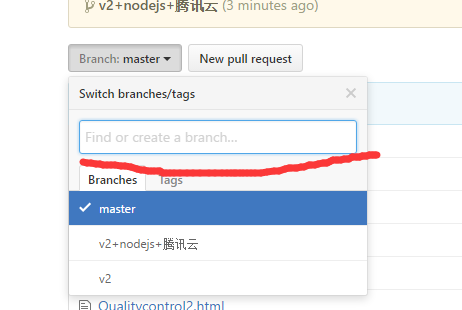
1. 创建分支  
     
   在红线框里输入即可。
2. 本地仓库提交到GitHub

首先github创建空仓库,复制空仓库的Git地址 XXX

然后本地先git init

创建.gitignore文件

检查用户是否和GitHub用户一样,

git remote add origin xxx

再git pull xxx

再git add .

git commit –m ‘up’

git push --set-upstream origin master

1. npm发布插件

<https://www.jianshu.com/p/7c29e3e933b0>

补充 :命令行npm login就不用注册了,

有更新先更改package.json和package-lock里的版本号在npm publish

记住账号名是bill-mark 不是bill.mark

开发插件用var不用let 或者修改webpack

因为 node对es6语法还不是完全支持

<https://blog.csdn.net/a8725585/article/details/74559803>

4.查看远程分支  
gtbranch -a

5.子分支合并主分支/  
 git merge master

如果子分支推送到主分支,切换到主分支,再merge子分支即可

1. 拉取仓库子分支子目录

创建目录

mkdir xxx(目录)

初始化

git init

拉取objects信息

git remote add -f origin xxx(远程git地址)

开启只拉取子目录

git config core.sparsecheckout true

设置要拉取的子目录

echo "xxx" >>.git/info/sparse-checkout (xxx为子目录名)

拉取远程仓库

Git pull origin x2 (x2远程分支名)

提交

git push origin x1:x2 (x1 本地分支名 x2远程分支名)

1. 删除本地缓存

git rm -r --cached .

8.本地分支关联远程分支

查看当前分支与远程分支关系  git branch -vv

如果返回信息没包含远程分支说明没关联

第一种(已经创建了本地分支)

指定当前本地分支与远程仓库链接关系

假设:本地分支名为master 远程分支名为dev

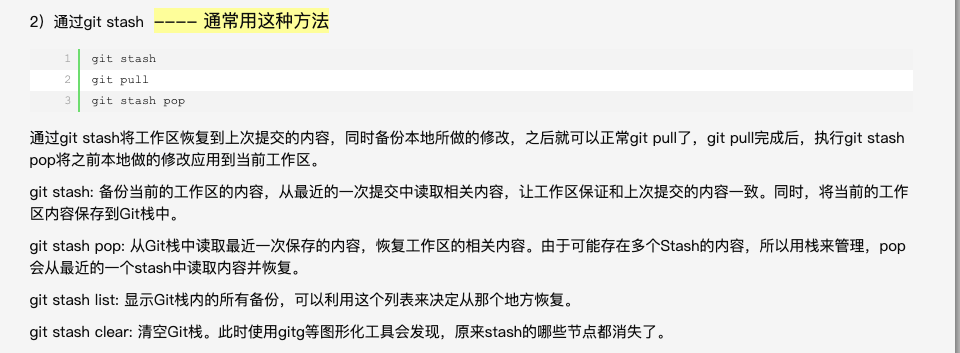
git branch --set-upstream master origin/dev

第二种(本地分支不存在)

git branch new\_branch origin/remote\_branch

1. git冲突

git checkout -- 文件名



<https://www.cnblogs.com/wenlj/p/5866356.html>



<https://www.jianshu.com/p/daba0bb673a0>

1. git撤销单文件修改

git checkout -- 文件名

1. commit的时候， 跳过eslit 的强制检测
2. git commit --no-verify -m 'update: 这里是你想写的描述'

11.查看隐藏文件

#显示所有文件（包含隐藏文件）  
ls -a

#只显示隐藏文件  
l.  
或者  
ls -d .\*