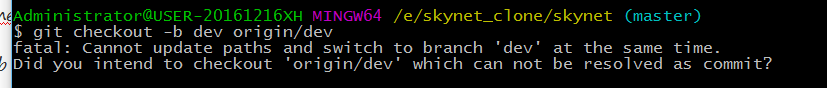
Git问题

01、fork开源项目 创建新分支的问题



解答：

你本地还没有这个分支在线上的记录，需要先fetch一下。  
git fetch origin lightmerge  
之后再使用checkout -b 试试看。

02、Git 真正理解 git fetch, git pull 以及 FETCH\_HEAD

git push.

这个很简单, 其实和后面的差不多, 这里就不讲了.

唯一需要注意的地方是:

git push origin :branch2, 表示将一个内容为空的同名分支推送到远程的分支.(说白了, 即删除远程主机的branch2分支), 但是这并不会消除之前的comment内容, 而且你一旦提交了一些大的文件(例如: 图片之类的), 通过这个操作, 是不会将这些文件占用的空间消除的. 如果要真正的删除一个文件, 除了删除整个项目, Github网站也有提供办法, 不过还没看懂.

git fetch, 理解fetch的含义, 是远程协作的关键.

而理解 fetch 的关键, 是理解 FETCH\_HEAD.

这里需要解释下什么是FETCH\_HEAD??

FETCH\_HEAD指的是: 某个branch在服务器上的最新状态'. 每一个执行过fetch操作的项目'都会存在一个FETCH\_HEAD列表, 这个列表保存在 .git/FETCH\_HEAD 文件中, 其中每一行对应于远程服务器的一个分支. 当前分支指向的FETCH\_HEAD, 就是这个文件第一行对应的那个分支.

一般来说, 存在两种情况:

如果没有显式的指定远程分支, 则远程分支的master将作为默认的FETCH\_HEAD.

如果指定了远程分支, 就将这个远程分支作为FETCH\_HEAD.

常见的git fetch 使用方式包含以下四种:

git fetch

这一步其实是执行了两个关键操作:

创建并更新所有远程分支的本地远程分支.

设定当前分支的FETCH\_HEAD为远程服务器的master分支 (上面说的第一种情况)

需要注意的是: 和push不同, fetch会自动获取远程`新加入'的分支.

git fetch origin

同上, 只不过手动指定了remote.

git fetch origin branch1

设定当前分支的 FETCH\_HEAD' 为远程服务器的branch1分支`.

注意: 在这种情况下, 不会在本地创建本地远程分支, 这是因为:

这个操作是git pull origin branch1的第一步, 而对应的pull操作,并不会在本地创建新的branch.

一个附加效果是:

这个命令可以用来测试远程主机的远程分支branch1是否存在, 如果存在, 返回0, 如果不存在, 返回128, 抛出一个异常.

git fetch origin branch1:branch2

只要明白了上面的含义, 这个就很简单了,

首先执行上面的fetch操作

使用远程branch1分支在本地创建branch2(但不会切换到该分支), 如果本地不存在branch2分支, 则会自动创建一个新的branch2分支, 如果本地存在branch2分支, 并且是`fast forward', 则自动合并两个分支, 否则, 会阻止以上操作.

git fetch origin :branch2

等价于: git fetch origin master:branch2

git pull

只要理解了git fetch, git pull就太简单了.

git pull 等价于以下两步:

经命令中的pull换成fetch, 执行之...

git merge FETCH\_HEAD

唯一需要提及的一点是:

我认为pull操作, 不应该涉及三方合并 或 衍合 操作 换个说法: pull 应该总是 fast forward 的. 为了达到这样一个效果, 在真正push操作之前, 我倾向于使用衍合, 在本地对代码执行合并操作.