https://blog.csdn.net/qq_45893999/article/details/106273214

- 本地项目—>远程仓库—>本地关联远程—>推送最新代码

Git add .

Git commit . -m ‘’

我都操作好了，但是 git push 时出错。原因如下：

在 github 创建项目时，点了自动创建 readme，但是我没有 pull 更新本地工程.结果就造成我提交代码提交不上去。这相当于多人同时操作一个项目，别人提交了一个代码并 merge 到主分支了，现在我提交代码前，没有更新，所以会引起冲突。那么正确的做法顺序是，先把远程的 pull 下来并合并到本地，使用命令

`git pull --rebase`

- **执行** `git pull --rebase` **的时候必须保持本地目录干净。** 即：不能存在状态为 `modified` 的文件。（存在 `Untracked files` 是没关系的）
- **如果出现冲突，可以选择手动解决冲突后继续** `rebase`**，也可以放弃本次** `rebase`

手动解决后，

```
git rebase --continue
```

参考来自这里

作者：kevinsEegets
链接：https://www.jianshu.com/p/14f9eb70e99e
来源：简书
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

这步完成的操作是将远程分支拉到本地更新合并。

而后可以再进行提交更新。

`git push origin main`

Git 现在默认主分支不再是 master，而是 main 了，可以通过网站看到，主分支是 main 了。

当然如果在创建项目时，没有选择自动创建 readme 文档，那本地没有存在差异，不会因为这个造成冲突。但是如果多人合作开发，总会存在这个问题。所以正确的做法是，先 pull 一下远程，然后再把自己的和远程的合并。具体还需要一步一步操作体验。

多人基于同一个远程分支开发的时候，如果想要顺利 push 又不自动生成 merge commit，建议在每次提交都按照如下顺序操作：

```
$ git stash
```

```
$ git pull --rebase

$ git push

$ git stash pop
```

其中 git stash 的使用可以参考这里，一般是先将自己没改完的工作储藏一下，然后开发完一个地方后，再回来做这块工作。https://www.cnblogs.com/tocy/p/git-stash-reference.html

# Failed to connect to github.com port 443 after 21098 ms: Timed out

这个错误的解决方法：

**解决步骤：我们直接在终端先输入设置代理的命令，再输入取消代理的命令即可解决**

```
git config --global https.proxy
```

git config --global --unset https.proxy

---

MR 之前把本地和远程分支都删掉了，咋办？

昨天提交了代码 mr，结果因为提交生产代码少，我看发出来邮件了，就以为是已经 merge 了，（其实只要提交 mr 都会发邮件，和是不是 merge 了没关系，）然后我就迅速地把本地和远程分支都删了。

删除本地分支　git branch --delete　branchname

删除远程分支　git push origin --delete　branchname

本地不再跟踪远程中已经删除的分支　　git fetch --all --prune

然后这样就删光光了。但是我删完了才意识到，我的 mr 还没通过。。。。。。于是 mr 自动关掉，同时由于我的远程分支已经删掉，mr 上的 changes 也没了。。。。。悲剧人生都是从无知开始。。。

于是，我就开始 git log -g

然后就发现了自己的 commit 历史记录，commit_id 知道了，于是

git rebase　commit_id　看了一下，是我提交的 commit。

然后怎么恢复我刚才的分支呢，假设刚才的分支名称是 git_branch。

此时万能的搜索给了我帮助。先创建你的分支 git_branch，从主分支 master 创建一个你的分支 git_branch，并切换到 git_branch 分支。

git checkout -b git_branch  master

然后把刚才的 commit_id 合并到分支 git_branch 中。

git cherry-pick commit_id

git push

好完成这个操作后，你删除的分支 git_branch 在本地和远程上都可以看到了，刚才提交的 mr 上也可以正常看到提交的 changes 等等了。

当然这个前提是，你提交的 commit 一定要有 message，而且不是很久，否则一大串别人的冲突都够头大的。