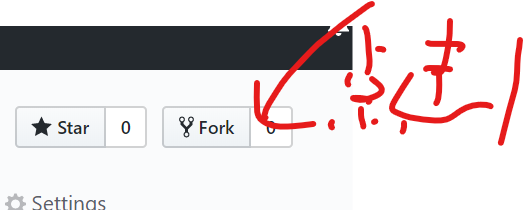
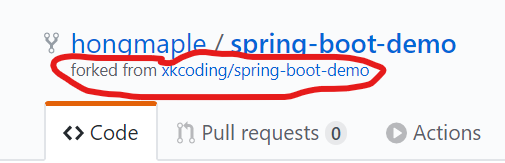
### 项目协同步骤

1. 打开<https://github.com/hongmaple/HtxkEmsm>这个网址
2.   
   点击这个按钮  
     
   把项目fork到自己的github账号中的仓库中
3. 然后你的仓库中就会有这个名字的项目，比如我fork别人的项目，就会出现标红的地方，就说明这个项目是fork别人的，不是源库



1. 然后再用git 克隆**fork**的项目到本地（注意要使用命令行下载，以创建本地.git文件记录本地仓库信息）:  git clone https://github.com/你的域名/你的项目名称.git

5，本地修改代码前，虽然已经把远程github上的项目clone下来，在修改代码之前还需要将远程github上的文件pull一下：

更新 项目名称：origin

git  pull  origin  master

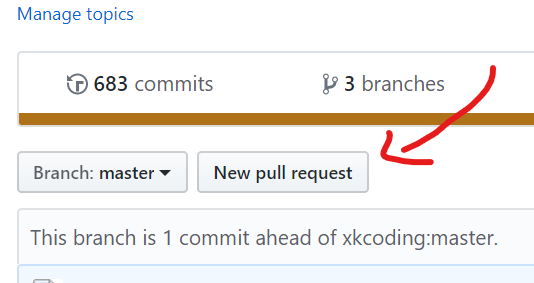
6.然后你把本地修改的，新增的文件提交到缓存区: git  add  new.html   all.php 提交单个文件

git add .提交全部

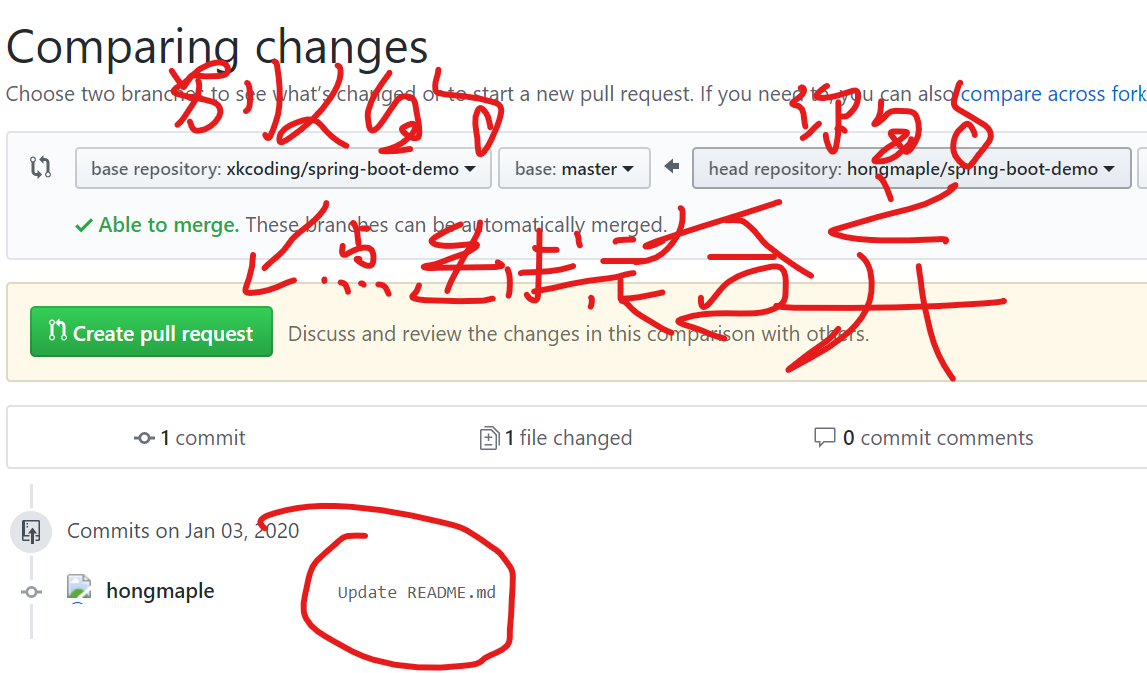
7.将缓存区的提交到本地仓库：git  commit -m "更新了"

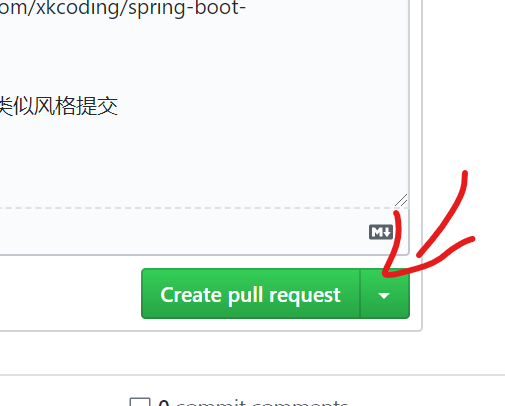
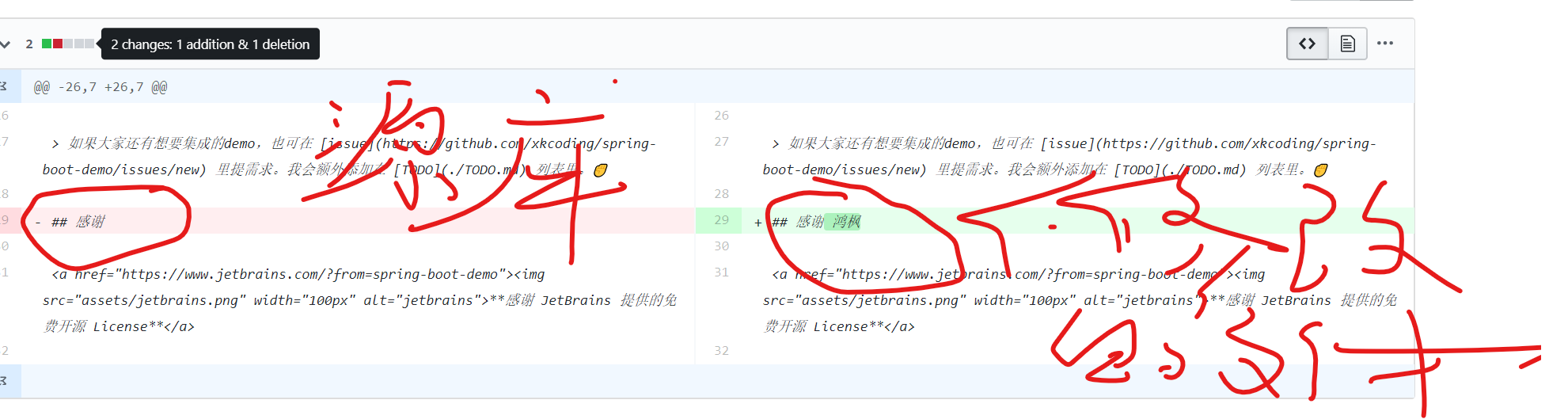
8.然后将本地仓库的文件推送到github上：git   push origin  master: master(本地的分支：github上的分支）

9.在自己fork的github仓库中点击New pull request ，把修改后的项目文件提交到源库进行合并



点击之后会出现

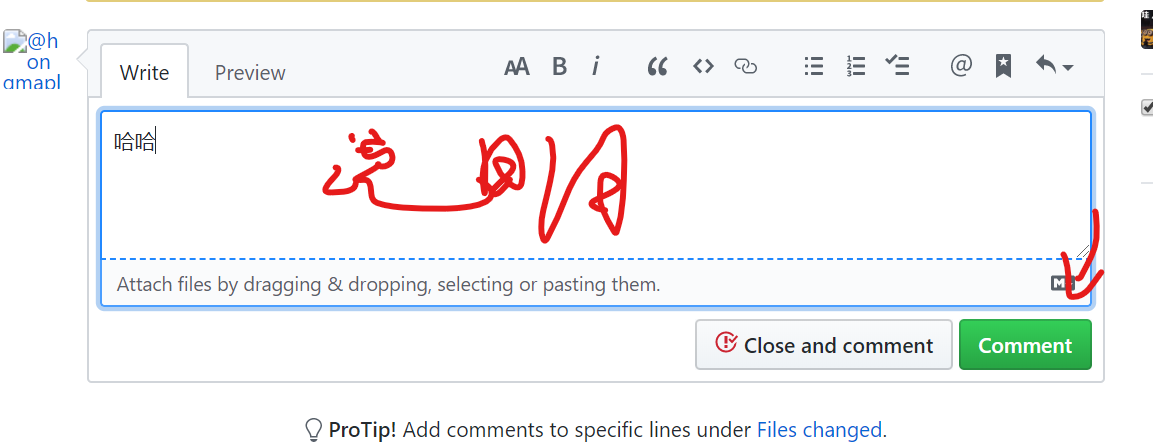




点击之后

填写说明内容

再点击提交



安装git<https://www.cnblogs.com/ximiaomiao/p/7140456.html>

D:\IT资源\第三学期\6.基于SSM轻量级框架应用实战\6.基于SSM轻量级框架应用实战-教学资料\13.项目案例—APP信息管理平台\5.提供的素材\项目案例：APP信息管理平台\软件工具部提供中有git安装包、

如果你是第一个创建项目的人，项目放在你的github或者gitee上：

1，首先你要在本地要上传文件的文件夹中初始化创建一个本地仓库：git  init

2，然后你把本地的文件提交到缓存区: git  add  .

3，然后你把缓存区的文件提交到远端公共 仓库：git commit  -m "一些说明"

4，然后连接gitee:  git  remote add origin  https://gitee.com/你的域名/你的项目名称.git

5，然后将本地仓库的文件推送到gitee上：git   push origin  master: master(本地的分支：gitee上的分支）

如果你是加入别人项目中的管理员，第一次想要修改代码时候：

1，先把gitee上项目代码下载到本地（注意要使用命令行下载，以创建本地.git文件记录本地仓库信息）:  git  clone https://gitee.com/你的域名/你的项目名称.git

2，本地修改代码前，虽然已经把远程gitee上的项目clone下来，在修改代码之前还需要将远程gitee上的文件pull一下：

git  pull  origin  master

3，然后你把本地修改的，新增的文件提交到缓存区: git  add  new.html   all.php

4，将缓存区的提交到本地仓库：git  commit -m "更新了"

5，然后将本地仓库的文件推送到gitee上：git   push origin  master: master(本地的分支：gitee上的分支）

clone 是本地没有 repository 时，将远程 repository 整个下载过来。

pull 是本地有 repository 时，将远程 repository 里新的 commit 数据(如有的话)下载过来，并且与本地代码merge。

如果你是加入别人项目中的管理员，第二次想要修改代码时候（本地已经有.git记录历史）：

1，先将gitee上的文件pull下来，以更新本地仓库：git  pull  origin  master(这个origin是已经remote连接过的）

2，将文件修改完毕后要推送过去，就按选项二的3~5执行一遍。

————————————————

版权声明：本文为CSDN博主「kalinux」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/kalinux/article/details/87716027>

# 同步叉子

同步存储库的分支，以使其与上游存储库保持最新。

[Mac](https://help.github.com/en/github/collaborating-with-issues-and-pull-requests/syncing-a-fork)**[Windows](https://help.github.com/en/github/collaborating-with-issues-and-pull-requests/syncing-a-fork)**[Linux](https://help.github.com/en/github/collaborating-with-issues-and-pull-requests/syncing-a-fork)

必须先[配置指向](https://help.github.com/en/articles/configuring-a-remote-for-a-fork) Git中[上游存储库的远程](https://help.github.com/en/articles/configuring-a-remote-for-a-fork)控制器，然后才能将fork与上游存储库同步。

打开Git Bash。

将当前工作目录更改为本地项目。

从上游存储库中获取分支及其各自的提交。承诺master将存储在本地分支中upstream/master。

$ git fetch upstream

> remote: Counting objects: 75, done.

> remote: Compressing objects: 100% (53/53), done.

> remote: Total 62 (delta 27), reused 44 (delta 9)

> Unpacking objects: 100% (62/62), done.

> From https://github.com/ORIGINAL\_OWNER/ORIGINAL\_REPOSITORY

> \* [new branch] master -> upstream/master

检查您的fork的本地master分支。

$ git checkout master

> Switched to branch 'master'

将更改从合并upstream/master到本地master分支。这使fork的master分支与上游存储库同步，而不会丢失本地更改。

$ git merge upstream/master

> Updating a422352..5fdff0f

> Fast-forward

> README | 9 -------

> README.md | 7 ++++++

> 2 files changed, 7 insertions(+), 9 deletions(-)

> delete mode 100644 README

> create mode 100644 README.md

1. 如果您的本地分支没有任何唯一的提交，则Git将执行“快速转发”：

$ git merge upstream/master

> Updating 34e91da..16c56ad

> Fast-forward

> README.md | 5 +++--

> 1 file changed, 3 insertions(+), 2 deletions(-)

****提示****：同步fork只会更新存储库的本地副本。要在GitHub上更新fork，您必须[推送更改](https://help.github.com/en/articles/pushing-commits-to-a-remote-repository)。