

# GitHub团队项目合作流程

已在另一篇博客中写出关于以下问题的解决，[点此进入](#)：

- 同步团队项目到本地时出现冲突怎么办？
- 项目负责人merge一个Pull Request后发现有点错怎么回退？

## 目录：

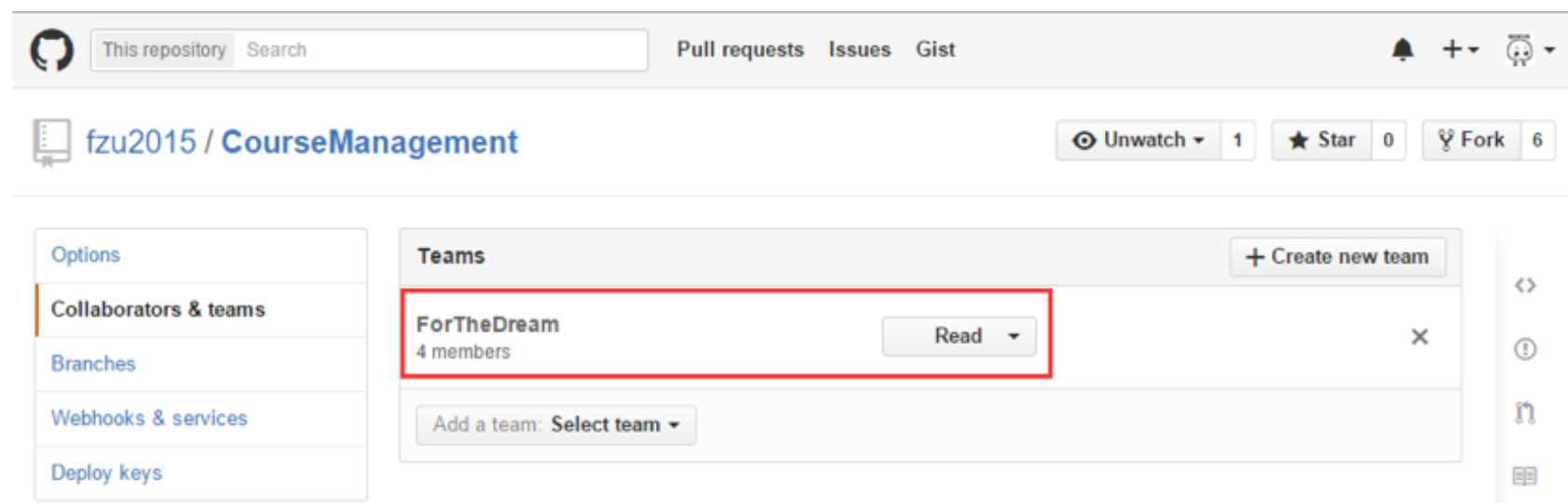
- 零、前期准备
- 一、创建开发分支
- 二、Fork项目到个人的仓库
- 三、Clone项目到本地
- 四、和团队项目保持同步
- 五、push修改到自己的项目上
- 六、请求合并到团队项目上
- 七、团队项目负责人审核及同意合并请求

注：其中 零、一、七 是由团队项目负责人来完成的。开发人员只要从 二开始就行了。

## 零、前期准备：

首先把队友直接push的权限关掉，即设置成Read。这样可以防止队友误操作，未经审核就把代码push到团队项目上。

Teams用来分配issue的时候会用到，所以保留下来，并不是没有用。

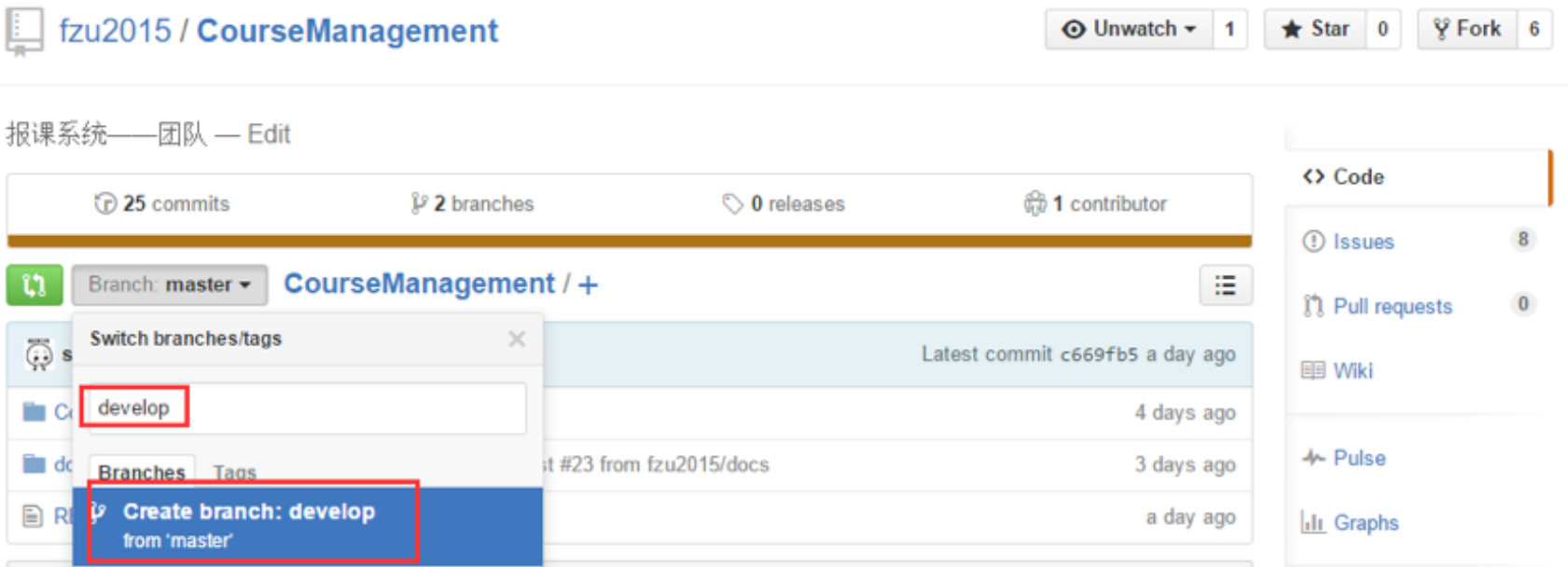


# 一、创建开发分支

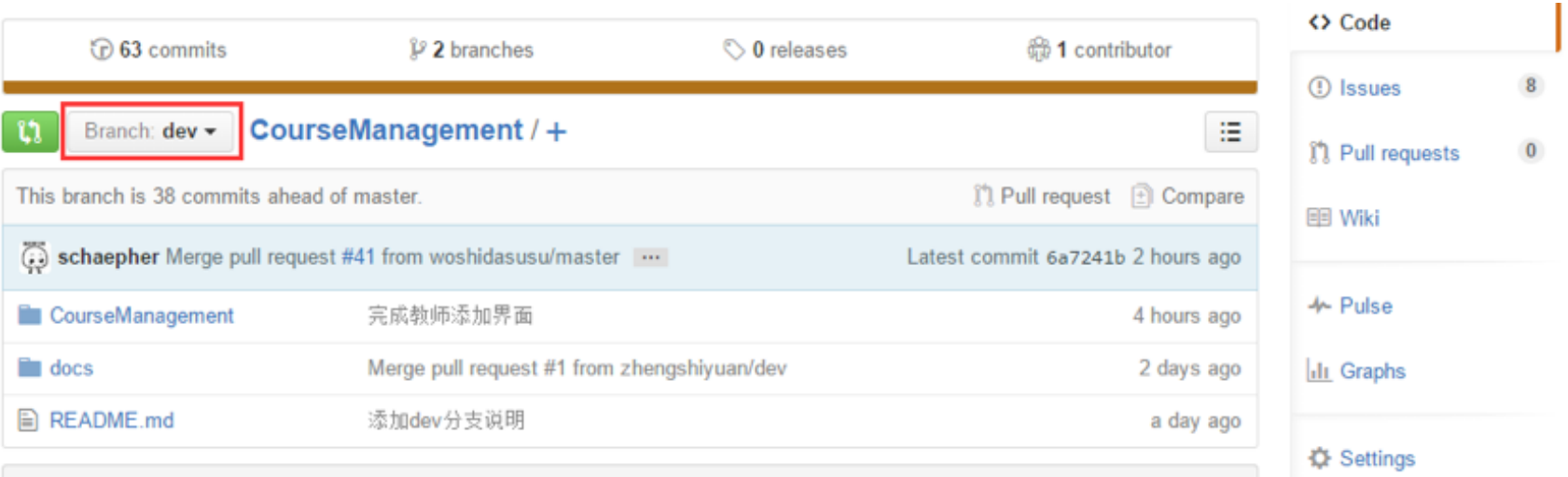
master分支一般用来发布稳定版本，dev分支（开发分支）用来发布开发版本。

输入分支名称后，下面会跳出Create branch，点击即可创建。

下面图片写的是develop，是因为我们这个项目已经有dev分支了。如果你们没有dev分支，那么名字改成dev即可。这个影响不大。

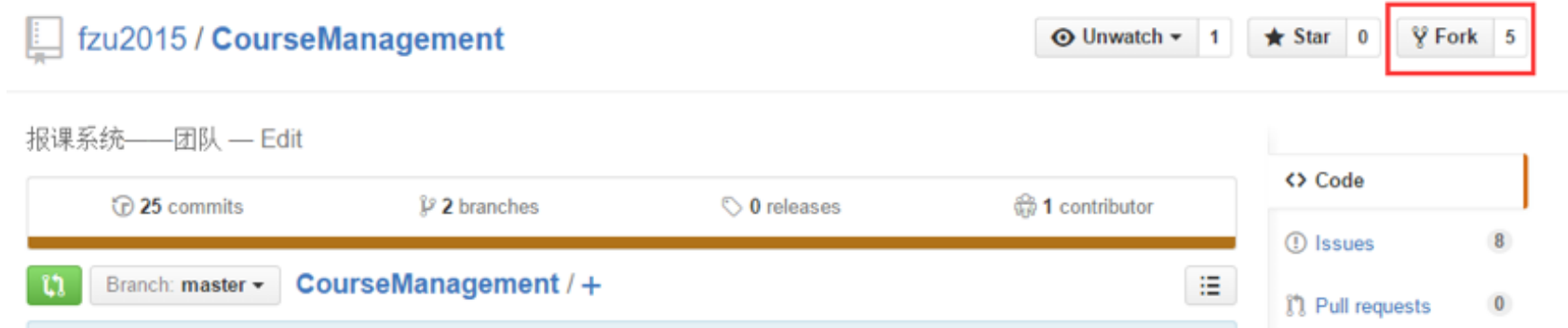


分支创建完毕后，会自动跳转到dev分支。由于dev分支是从master分支上创建的，因此内容与master分支一致。

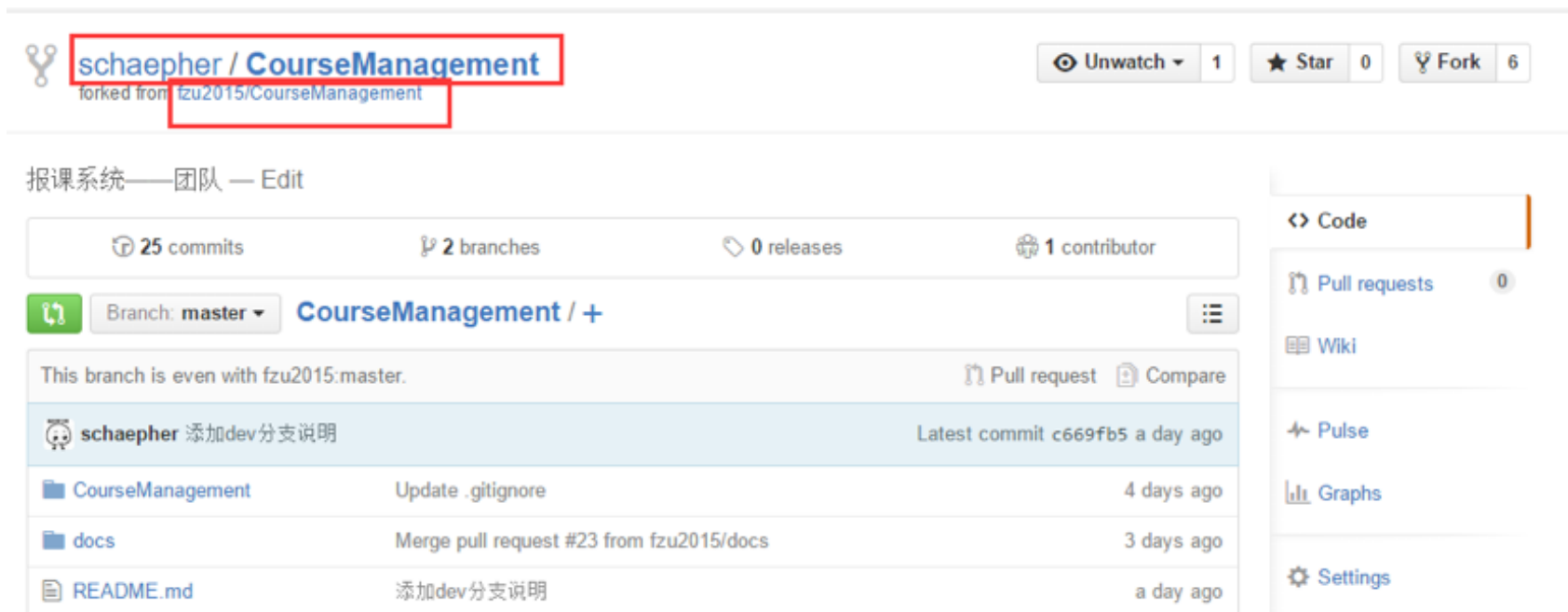


# 二、Fork项目到个人的仓库

点击右上角的Fork，并选择你的账号（一般在第一个）。就可以Fork团队项目到个人仓库啦。



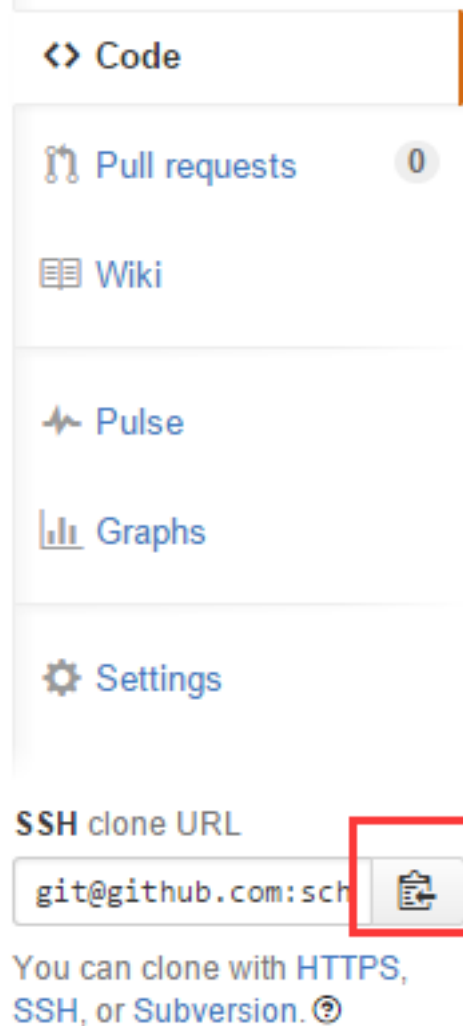
## Fork完成后



## 三、Clone项目到本地

首先是clone， clone的地址可以直接点击按钮复制（如下图）。

推荐使用SSH协议，用HTTP协议有时会出问题。  
注意， 这里clone的是你自己仓库里的项目



打开git命令行，输入指令和刚才复制的地址，回车即可克隆到本地

```
D:\repository> git clone git@github.com:schaephe/CourseManagement.git
Cloning into 'CourseManagement'...
Warning: Permanently added 'github.com,192.30.252.131' (RSA) to the list of known hosts.
remote: Counting objects: 634, done.
remote: Compressing objects: 100% (208/208), done.
Rreceiving objects: 80% (508/634), 1.43 MiB | 38.00remote: Total 634 (delta 100), reused 0
Receiving objects: 81% (514/634), 1.43 MiB | 38.00 KiB/s
Receiving objects: 100% (634/634), 1.47 MiB | 41.00 KiB/s, done.
Resolving deltas: 100% (241/241), done.
Checking connectivity... done.
```

此时你只能看到master分支，并没有把dev分支clone下来。使用 `git branch` 命令查看本地分支，发现本地只有master分支。如下图的①

```
D:\repository> cd .\CourseManagement
D:\repository\CourseManagement [master]> git branch ①
* master
D:\repository\CourseManagement [master]> git branch -a ②
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/dev
  remotes/origin/master
D:\repository\CourseManagement [master]> git checkout -b dev origin/dev ③
Branch dev set up to track remote branch dev from origin.
Switched to a new branch 'dev'
```

如上图的②，使用 `git branch -a` 查看所有分支，就能看到远程分支。根据远程分支，我们可以创建一个新的本地分支dev，并把该项目的dev分支的内容放到本地dev分支。如上图③。

git checkout -b dev origin/dev 的意思是，创建一个dev分支（-b），并把远程dev分支（origin/dev）的内容放在该分支内。接着切换到该分支（checkout）

现在使用 git branch 可以查看两个分支，并且用 ls 或者 dir 就能看到 dev 分支的内容了。想切换回 master 分支的时候，再用 git checkout master 即可。

```
D:\repository\CourseManagement [dev]> git branch
```

```
* dev
master
```

```
D:\repository\CourseManagement [dev]> ls
```

目录: D:\repository\CourseManagement

Mode	LastWriteTime	Length	Name
d----	2015/11/2 22:55		CourseManagement
d----	2015/11/2 22:55		docs
-a----	2015/11/2 22:54	259	README.md

上面的操作完成后，你就可以在本地进行开发了。但是如果要将你修改完的代码合并到团队项目上，还需要进行下面的操作。

## 四、和团队项目保持同步

首先查看有没有设置upstream，使用 git remote -v 命令来查看。如下图

①

```
D:\repository\CourseManagement [dev]> git remote -v
```

```
origin git@github.com:schaepfer/CourseManagement.git <fetch> ①
```

```
origin git@github.com:schaepfer/CourseManagement.git <push> ②
```

```
D:\repository\CourseManagement [dev]> git remote add upstream git@github.com:fzu2015/CourseManagement.git
```

```
D:\repository\CourseManagement [dev]> git remote -v ③
```

```
origin git@github.com:schaepfer/CourseManagement.git <fetch>
```

```
origin git@github.com:schaepfer/CourseManagement.git <push>
```

```
upstream git@github.com:fzu2015/CourseManagement.git <fetch>
```

```
upstream git@github.com:fzu2015/CourseManagement.git <push>
```

如果没有显示upstream，则使用 git remote add upstream 团队项目地址 命令。如上图②

接着再次使用 git remote -v，如果如上图③，显示出了upstream，那么就设置好了

开始同步。首先执行 git fetch upstream 获取团队项目最新版本。如下

图①

```
D:\repository\CourseManagement [dev] > git fetch upstream ①
Warning: Permanently added 'github.com,192.30.252.130' (RSA) to the list of known hosts.
remote: Counting objects: 121, done.
remote: Compressing objects: 100% (94/94), done.
remote: Total 121 (delta 35), reused 0 (delta 0), pack-reused 2 receiving objects: 39% (48/121), 124.00 KiB : 159.00 KiB/s
Receiving objects: 100% (121/121), 165.59 KiB : 159.00 KiB/s, done.
Resolving deltas: 100% (35/35), done.
From github.com:fzu2015/CourseManagement
 * [new branch]      dev       -> upstream/dev
 * [new branch]      master    -> upstream/master
D:\repository\CourseManagement [dev] > git merge upstream/dev ②
Updating 6a7241b..522eb13
Fast-forward
```

此时并没有把最新版本合并到你本地的分支上，因此还需要一步。如上图②，当前分支是dev分支，执行 `git merge upstream/dev` 命令后，会将源分支（upstream/dev）合并到当前分支（dev）。

如果你是在本地的master分支上开发，那么在使用该命令前，先切换到master分支。

merge的时候，有可能碰到冲突。需要解决冲突才能继续下面的操作。  
冲突的解决可以参考→ [冲突的解决](#)

## 五、push修改到自己的项目上

解决冲突后，就可以使用 `git push` 命令将本地的修改同步到自己的GitHub仓库上了。

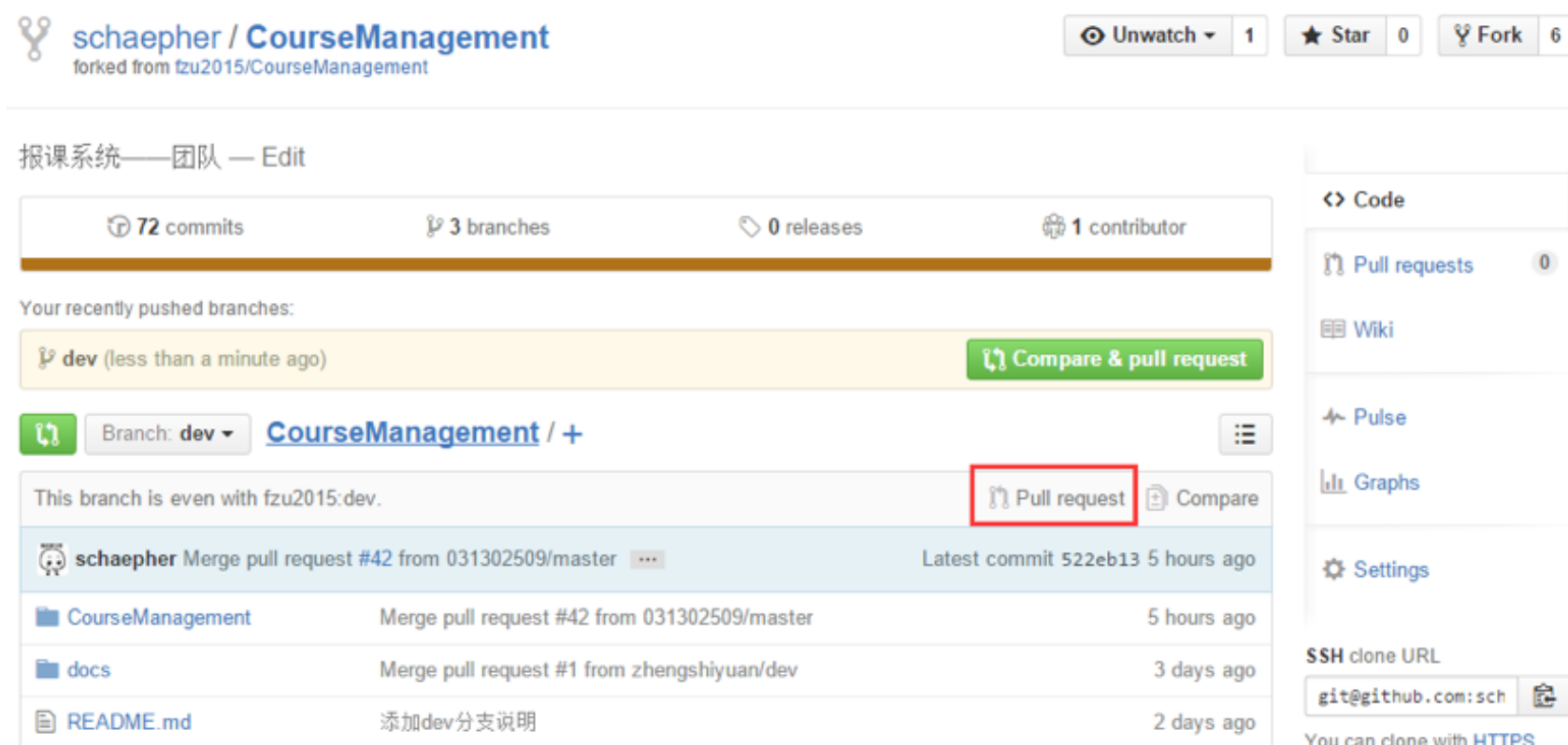
注意，在当前所在分支使用push，会push到与这个分支相关联的远程仓库分支。这里dev分支与origin/dev关联，因此push到GitHub上的dev分支。

```
D:\repository\CourseManagement [dev] > git push
Warning: Permanently added 'github.com,192.30.252.129' (RSA) to the list of known hosts.
Counting objects: 198, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (61/61), done.
Writing objects: 100% (121/121), 160.32 KiB : 0 bytes/s, done.
Total 121 (delta 50), reused 106 (delta 35)
To git@github.com:schaepher/CourseManagement.git
 6a7241b..522eb13 dev -> dev
```

## 六、请求合并到团队项目上

首先到你的GitHub上，进入你Fork的仓库里。点击红框处的Pull request



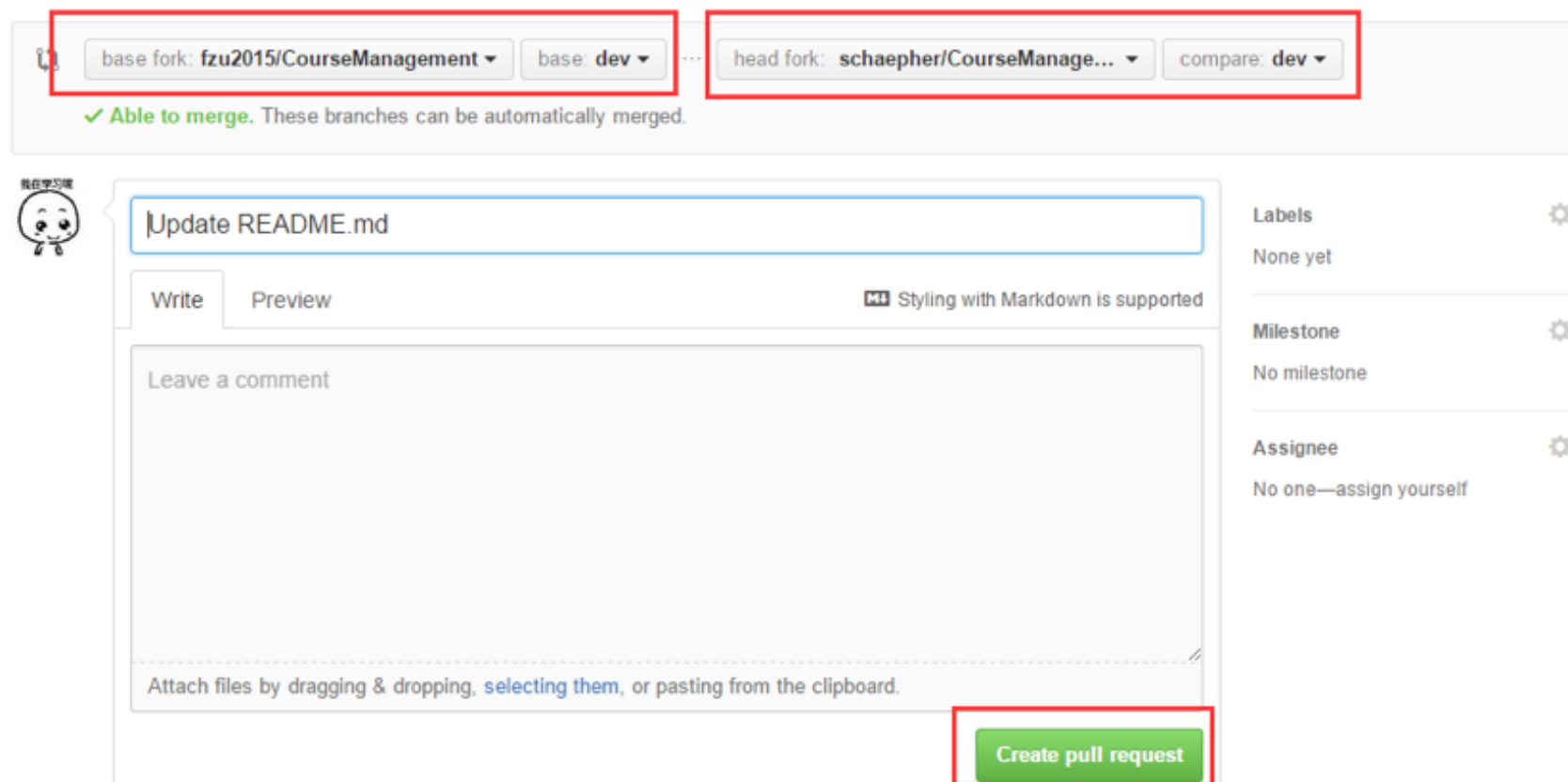


下图左边红框，表示要合并到fzu2015/CourseManagement项目的dev分支。

下图右边红框，表示要从自己仓库的dev分支发起合并请求。点击红框处的 Create pull request就可以发送合并请求了。

## Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).



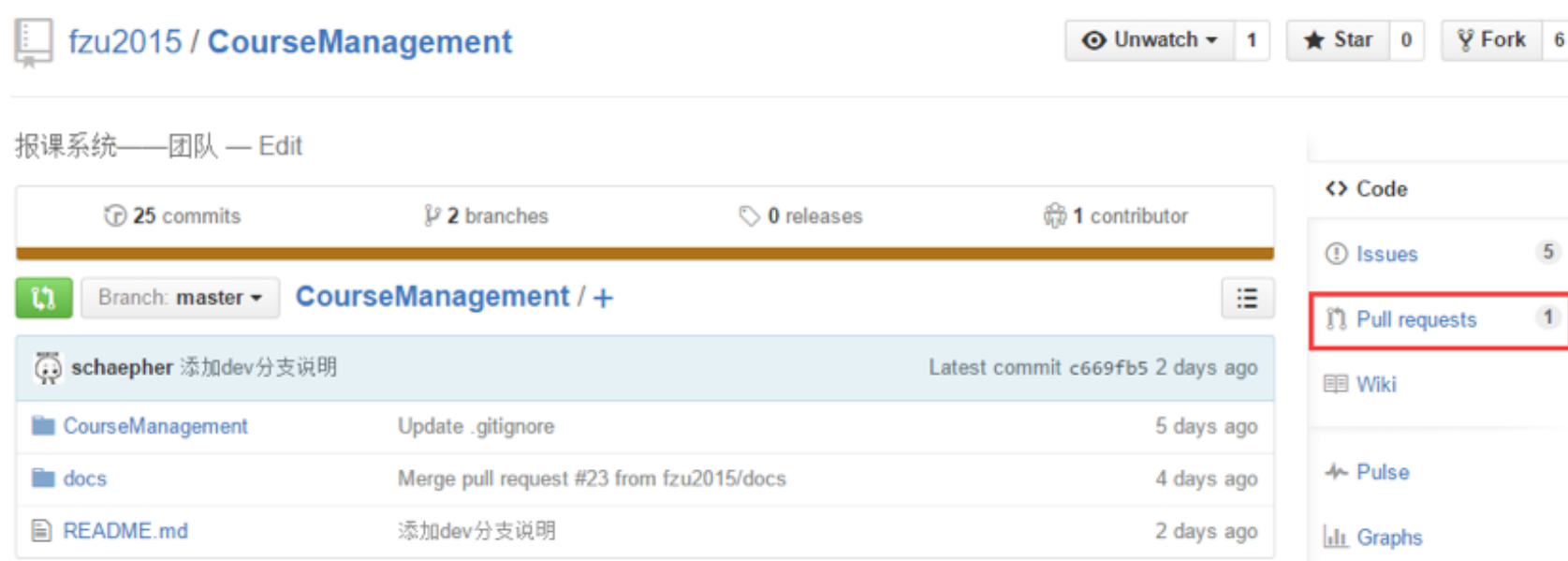
当然，在发送请求之前，你可以检查一下你都改了哪些东西。在上面那个页面往下拉，就可以看到两者的对比。如下图



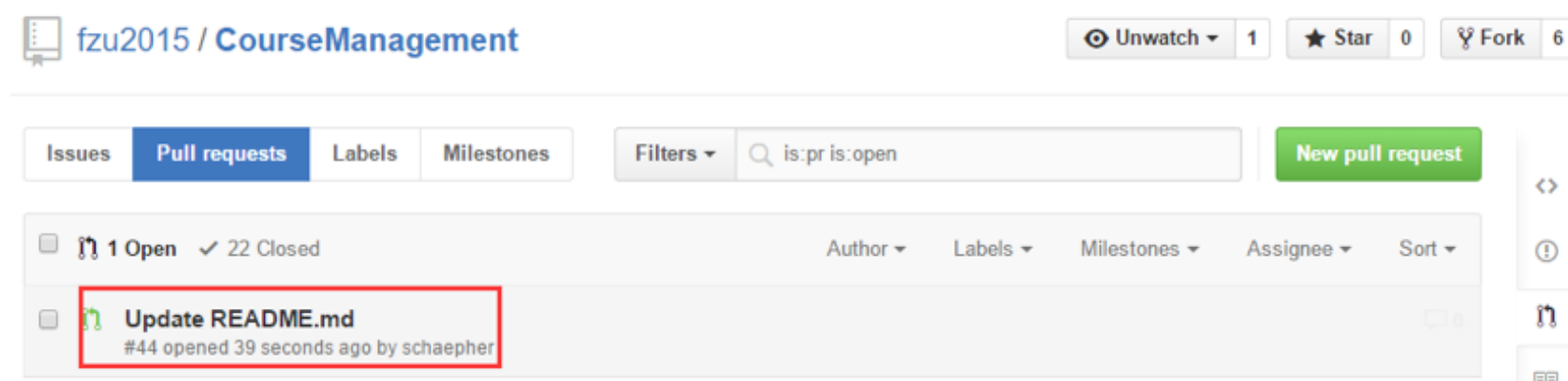
以上操作结束后，团队成员的流程就结束了。最后一步交给团队项目负责人来完成。

## 七、团队项目负责人审核及同意合并请求

首先进入GitHub的团队项目仓库中。此时右边的Pull requests显示当前项目有几个Pull request。点击进入查看。



选择一个Pull request



项目负责人审核有两个要注意的地方



- 一个是下图的①。一定要看清楚是合并到哪个分支。这里是从schaepher的dev分支合并到fzu2015的dev分支。
- 另一个是下图的②。点击进去后，就可以查看该Pull request对项目做了哪些修改。这样如果有问题，可以及时发现，并关闭该Pull request。

如果关闭了，一定要告诉队友，否则他可能会不知道。虽然也可以直接在下面发布Comment告诉他，但队友不一定看到。

Update README.md #44

Open schaeper wants to merge 3 commits into fzu2015:dev from schaeper:dev ①

Conversation 0 Commits 3 Files changed 1 ② +2 -2

**schaeper** commented 22 minutes ago  
No description provided.

**schaeper** added some commits 26 minutes ago

- Update README.md 081295a
- Update README.md 0598fbb
- Update README.md 3fdfe11

Add more commits by pushing to the dev branch on schaeper/CourseManagement.

**This branch is up-to-date with the base branch**  
Merging can be performed automatically.

**Merge pull request** ③ You can also open this in GitHub Desktop or view command line instructions.

Labels: None yet  
Milestone: No milestone  
Assignee: No one—assign yourself  
Notifications: Unsubscribe  
1 participant

- 如果没有问题，可以点击Merge pull request。这样就合并好了。