PR: Pull Request 怎么进行

Pull Requests

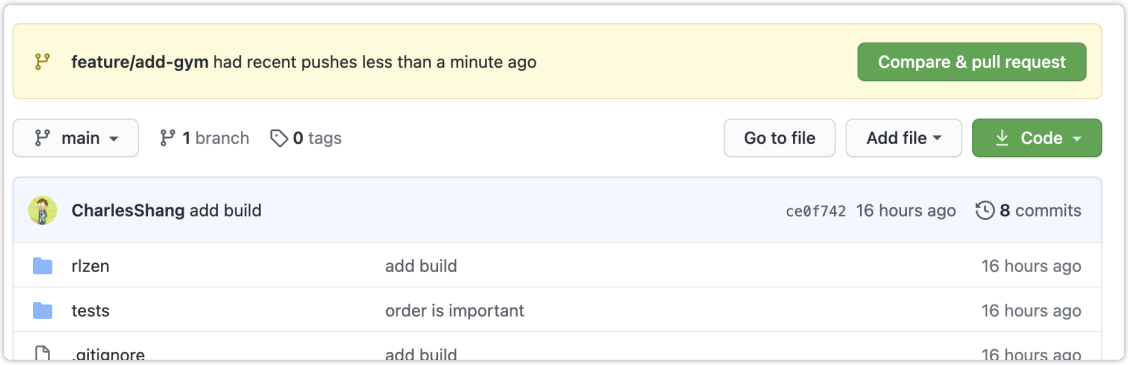
PR 是多个人进行协作开发、交流的主要方式。如果一个仓库涉及到多个开发者，就应该采用这种方式进行合作。一般来说除了最开始建立仓库的时候，任何人都不应该直接向 master 分支提交内容。最开始建立仓库，提交一些基础模块、baseline的时候除外。

1 准备

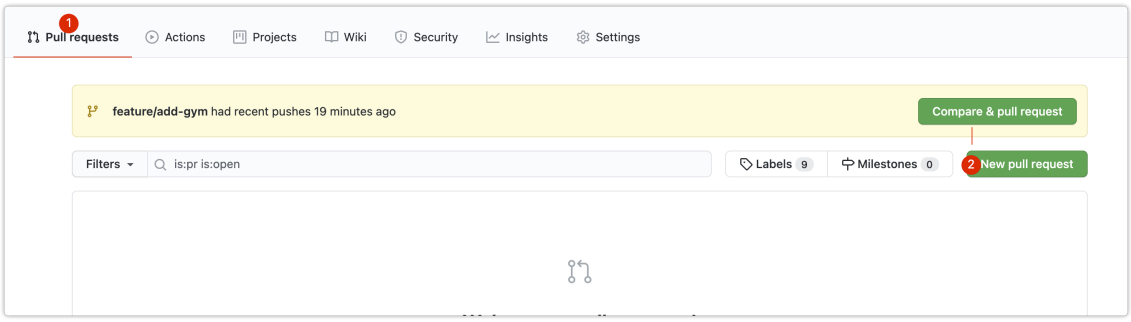
确保你做的修改已经被 push 到某个分支，或者是某个 fork 中。假设你已经完善好了自己的代码，本地测试通过，有信心将自己的改动加入到 master/main 分支中。

2 提交 PR

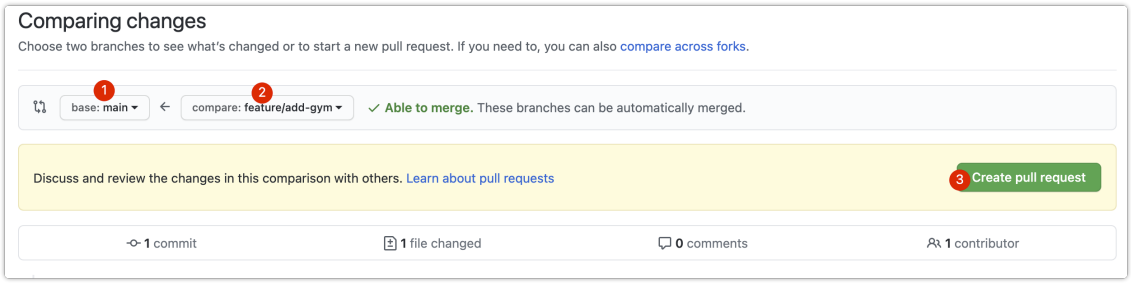
1. 创建一个 pull request 请求.
    1. 分支上的修改可以自动触发 github 的 *Compare & Pull Request* 按钮，点击之



2. 如果是 fork 的仓库会在你自己的仓库中出现一个按键.
3. 另外，可以点击页面上方的 Pull Request 创建一个新的PR

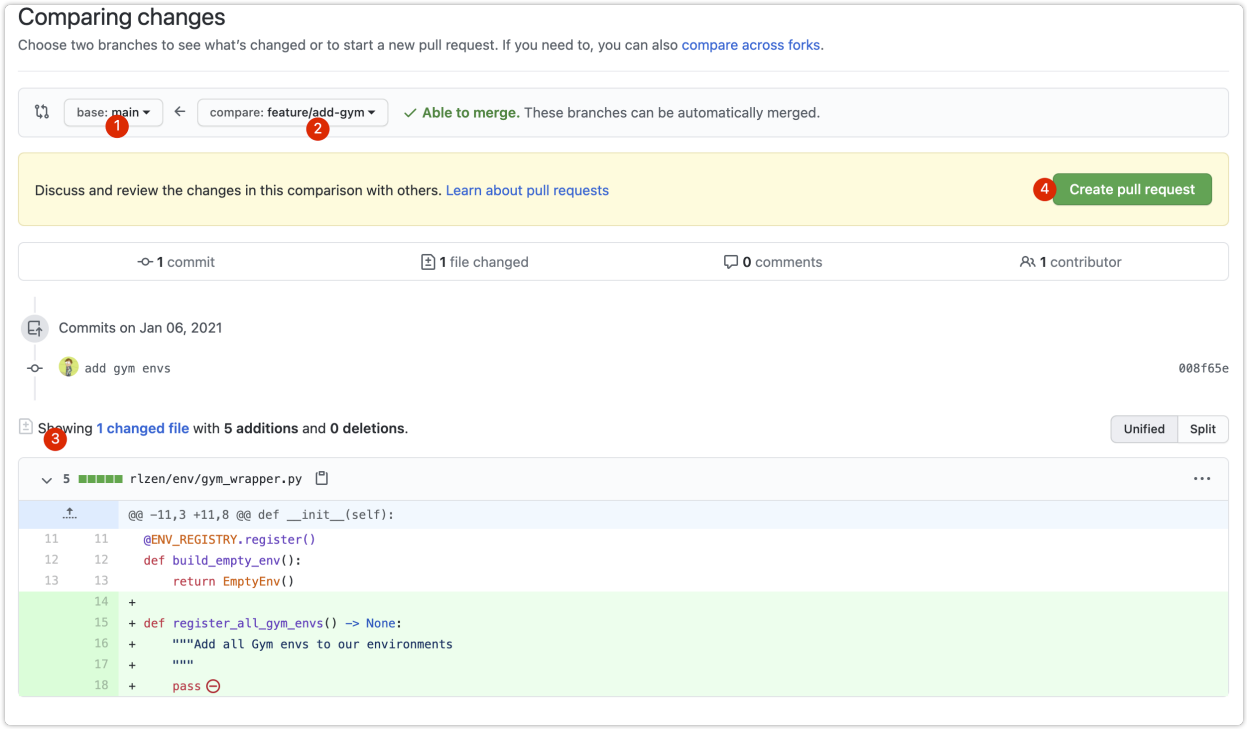


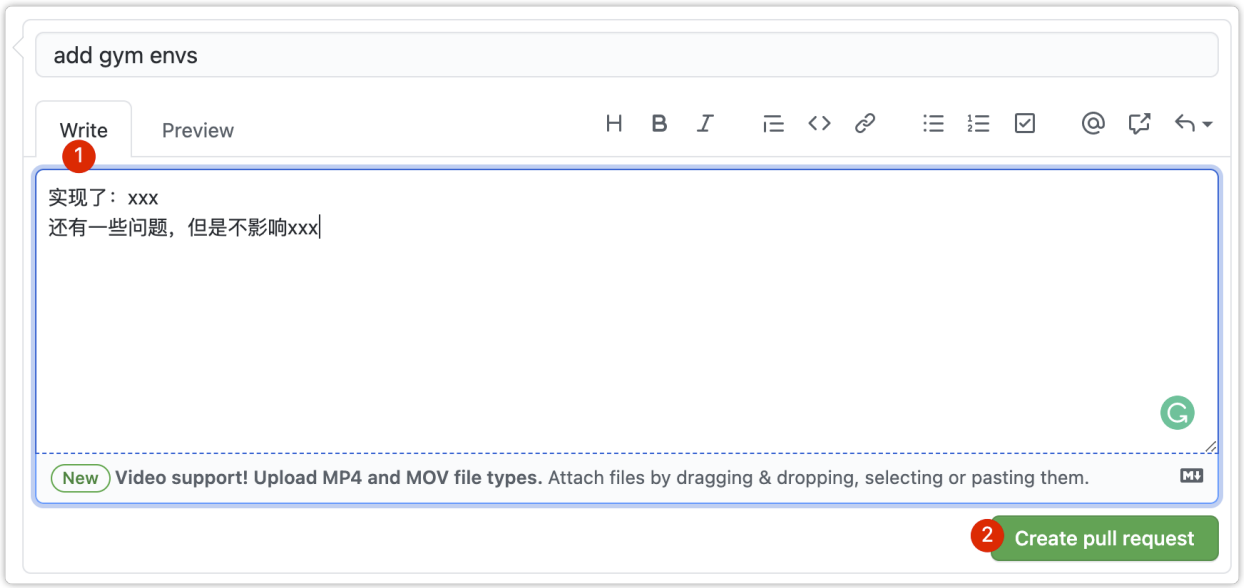
4. 手动创建 PR 流程：1 选择目标分支、 2 选择修改的分支



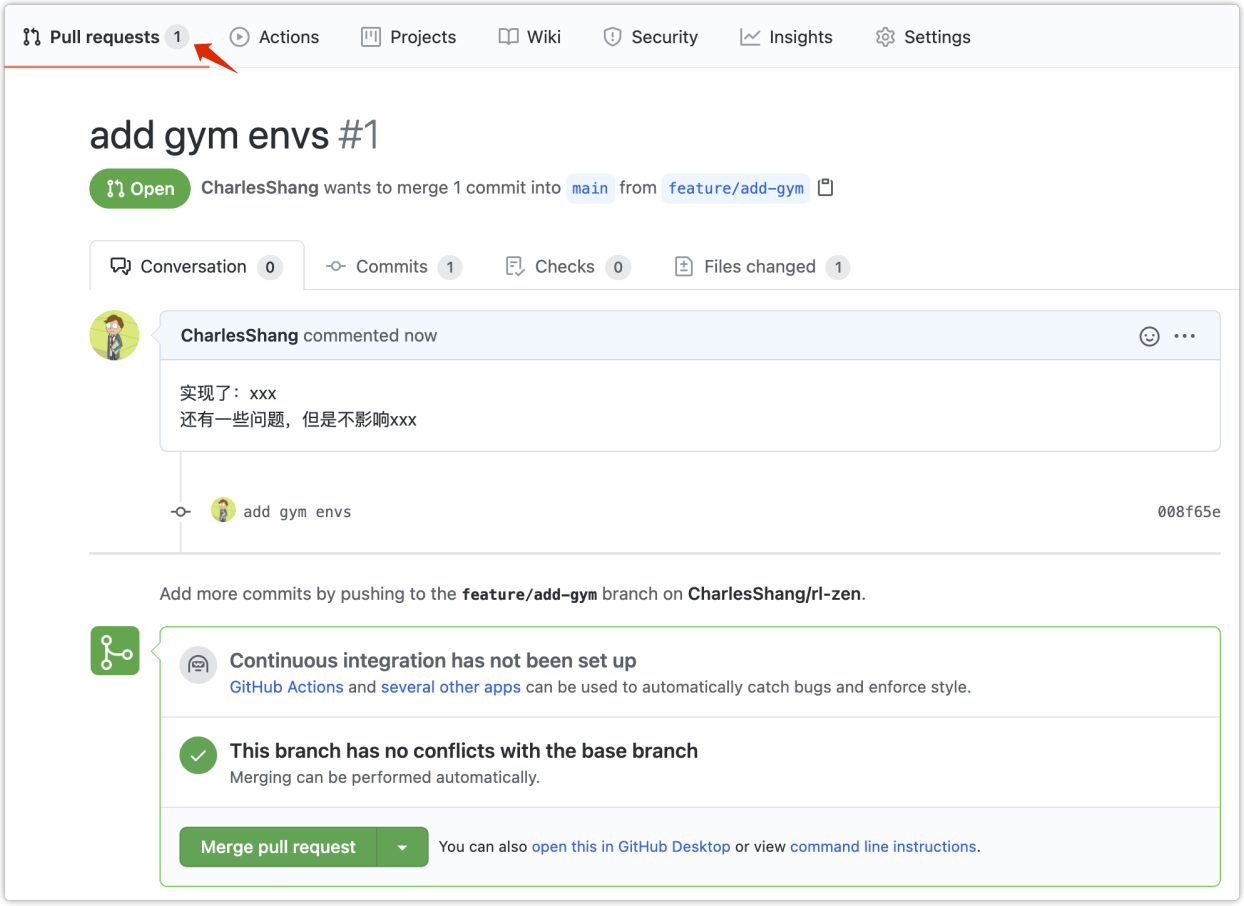
3 比较代码的改动

1. **比较代码的改动**: 再一次检查你修改的代码、修改的文件





创建成功 PR 后页面如下图。



4 Code Review

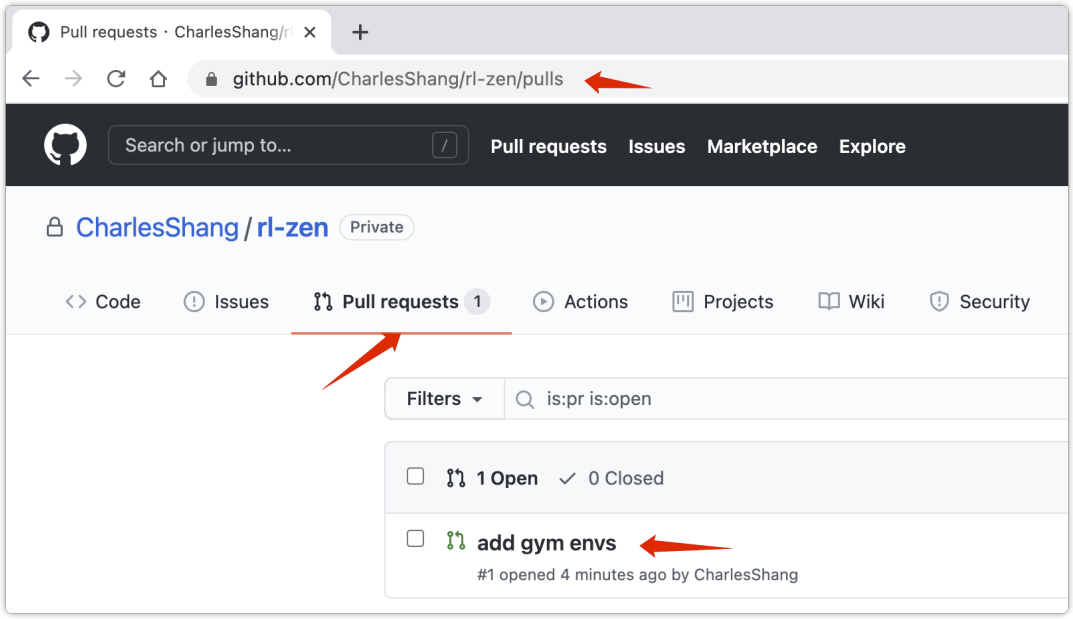
Code Review 工作说白了就是自己写了代码，让同组的人来看看。其主要的目的是（按从基本到高阶依次来说）：

1. 让相关人员知晓你的工作
2. 检查是否有bug
3. 代码风格是否合乎规范
4. 是否有更好的方式实现同一个功能

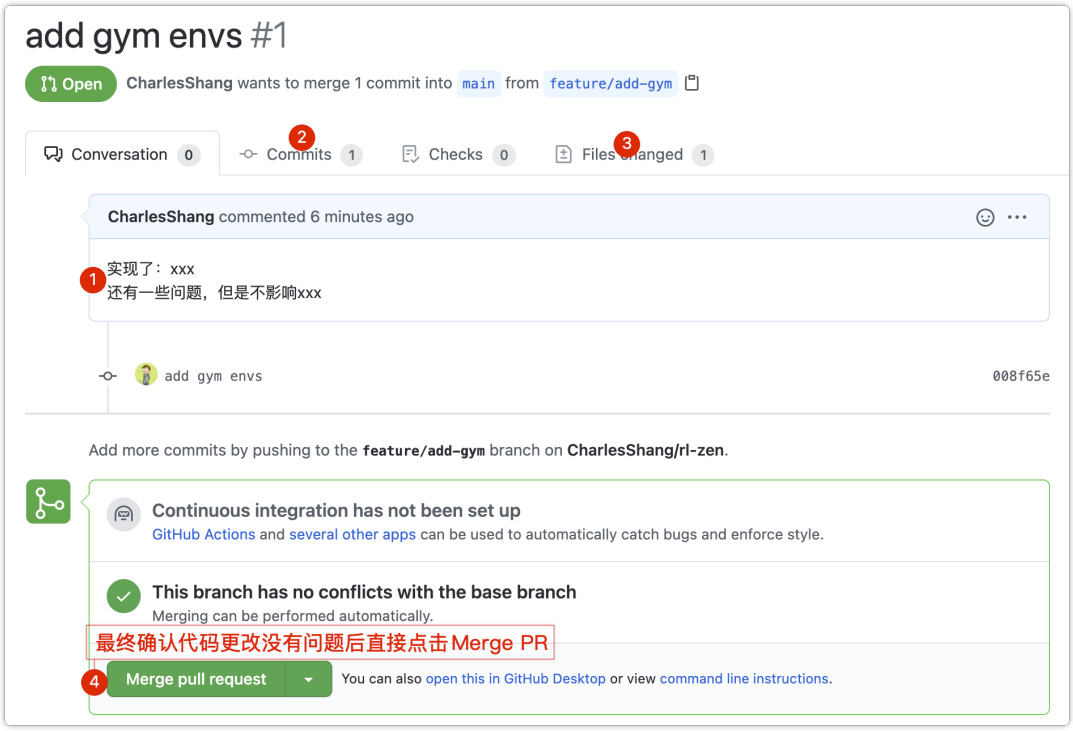
主要思想是说，并不是你做完一件事都万事大吉了，而是需要把这件事告诉别人（扩大自己的影响力），让人检查自己的工作，收集可能的意见，尽可能提升自己的工作能力。大公司里Google 的程序员会主动找大牛给自己做 Code Review 的。Code Review 可以涉及很多个人觉得有用的具有实际操作性的建议，*细致*到变量命名方式（例如，Google 发布的编程规范、Python PEP规范等）、实现方式（例如有些地方更适合把一个复杂的 dict 实现成一个单独的 class）、功能描述是否合理，文件结构（比如一个新的文件是否应该

出现在这个文件中）等。Code Review 是把多个协作者聚集到一起，共同合作、共同提高的一个核心步骤。

1. **参与 Code Review：**
    1. 粗略的看一下改动的文件和代码
        1. 在页面的Pull Reqest下就能看到当前提交的所有的PRs

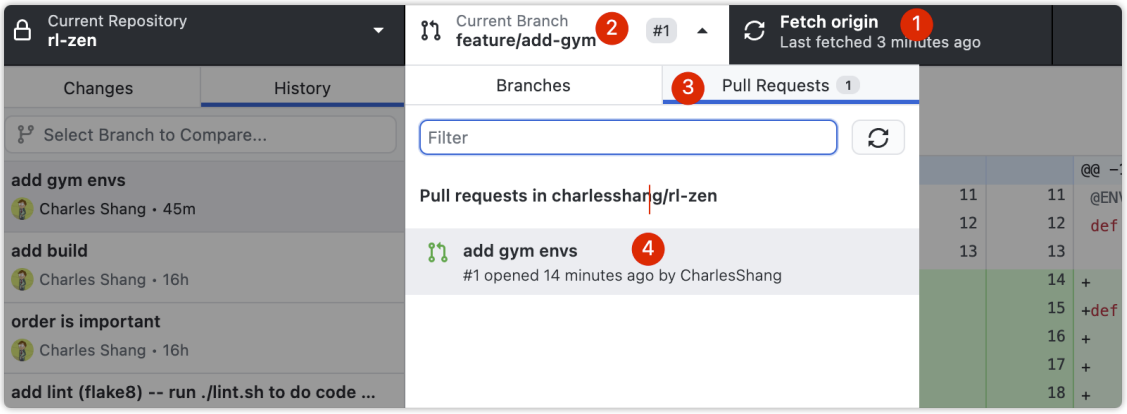


可以进一步浏览某个 PR 所涉及到的所有更改，可以针对具体某行代码**进行评论**等



2. Reviewer 自己执行一下测试、编译、运行

   例如在 Github Desktop 下就能看到PR，可以将PR同步到本地，编译、测试、运行



5 合并到主分支

合并操作

6 可能产生的问题

Merging Conflict