[记录自己使用GitHub的点点滴滴](https://www.cnblogs.com/wj-1314/p/9901763.html)

**前言**

　　现在大多数开发者都有自己的GitHub账号，很多公司也会以是否有GitHub作为一项筛选简历以及人才的选项了，可见拥有一个GitHub账号的重要性，本文就从最基本的GitHub账号的注册到基本的使用进行学习记录，一方面方便自己，另一方面希望对你有所帮助~

　　之前写过一篇文章，浅谈使用git 进行版本控制（博客链接：[https://www.cnblogs.com/wj-1314/p/7992543.html](https://www.cnblogs.com/wj-1314/p/7992543.html" \t "_blank)）

　　说实话，当时是学习可视化的时候，接触到git ，学了些皮毛，但是自己在实践中并没有真正的用到GitHub这个平台。那时候在大学，还无忧无虑的，但是一年后，也就是今天，我在家里写了代码，但是在公司也想写，才发现了GitHub的好处了，今天迫不及待的使用了这个东西，并且把一些笔记记录下来。

**GitHub简介**

 　　GitHub是用于版本控制和协作的代码托管平台，它可以让您和其他人在任何地方协同工作。GitHub 可以托管各种Git版本库，并提供一个web界面，但与其它像 SourceForge或Google Code这样的服务不同，GitHub的独特卖点在于从另外一个项目进行分支的简易性。

 　　为一个项目贡献代码非常简单：首先点击项目站点的“fork”的按钮，然后将代码检出并将修改加入到刚才分出的代码库中，最后通过内建的“pull request”机制向项目负责人申请代码合并。已经有人将GitHub称为代码玩家的MySpace。

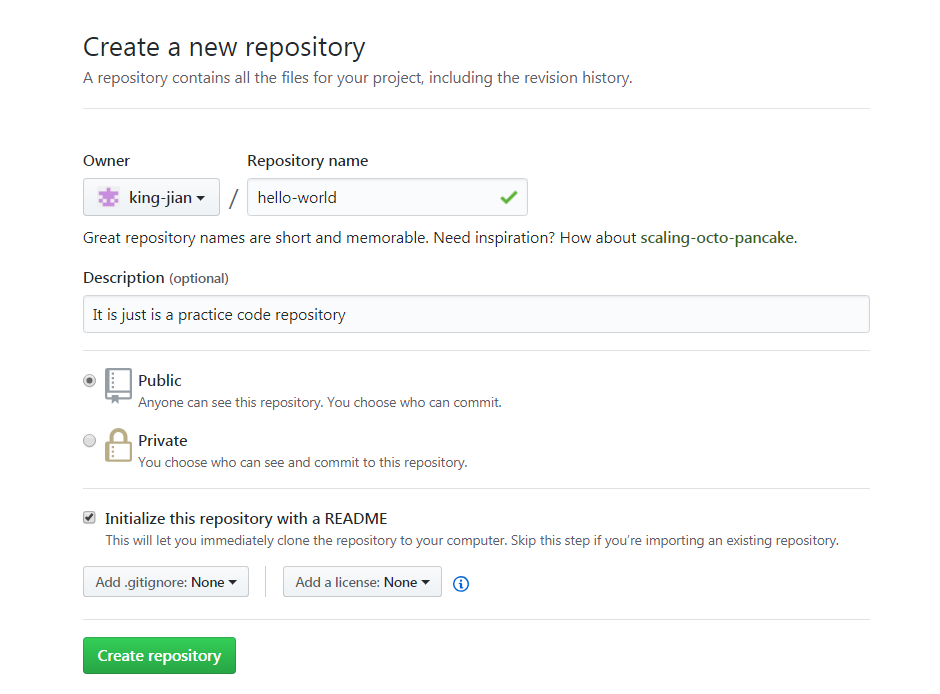
**一：创建新的Git仓库**

　　本文将学习使用GitHub基本知识，如存储库，分支，提交和Pull请求，我将创建自己的hello world存储库并学习GitHub的Pull Request工作流，这是一种创建和检查代码的流行方法。

**1：创建存储库**

　　一个库通常用于举办单个项目，存储库可以包含文件夹和文件，图像，视频，电子表格和数据集等等，你的项目需要的任何内容，我们认为包括README或者包含项目信息的文件。GitHub可以在创建新存储库的同时轻松添加一个。

　　如下图所示：我们可以添加一个新的仓库，并且添加描述，最后**单击Create repository**。

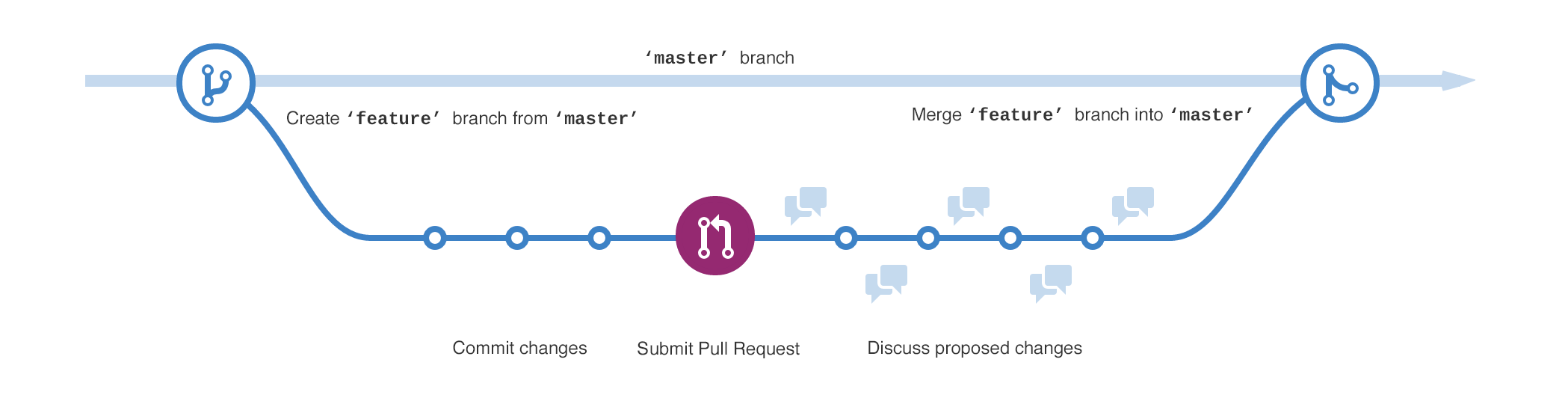


**2：创建一个分支（Create a Branch）**

　　分支是一次处理不同版本的存储库的方法。

　　默认情况下，我们的存储库有一个名为master的分支branch，该分支被认为是权威分支。我们在使用分支进行试验并在提交之前进行编辑master。

　　当你在分支机构上创建master 分支时,我们正在制作该master时间点的副本或者快照，如果其他人master在我们的分支机构上工作时对branch做了更改，则可以提取这些更新。



　　在GitHub，我们的开发人员，编写人员和设计人员使用分支来保持错误修复和功能工作与我们的master分支分开，当更改准备就绪时候，他们讲其分支合并到master。

**3：创建一个新分支**

* 1，转到新的仓库，也就是我们的仓库hello-world
* 2，单击文件列表顶部的下拉列表：master
* 3，在新分支文本框中输入分支名称readme-edits
* 4，选择蓝色的创建分支框或者按键盘的Enter

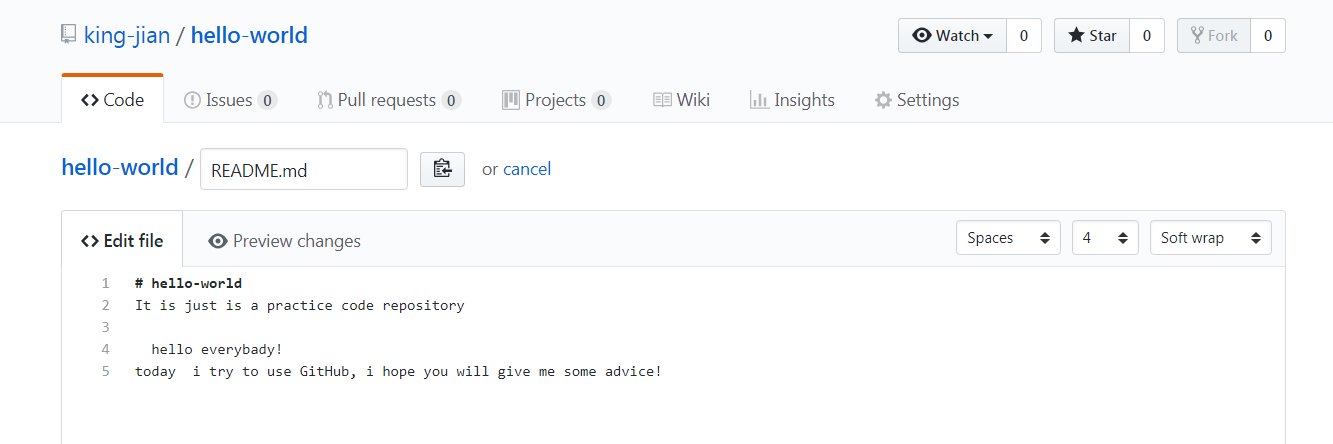


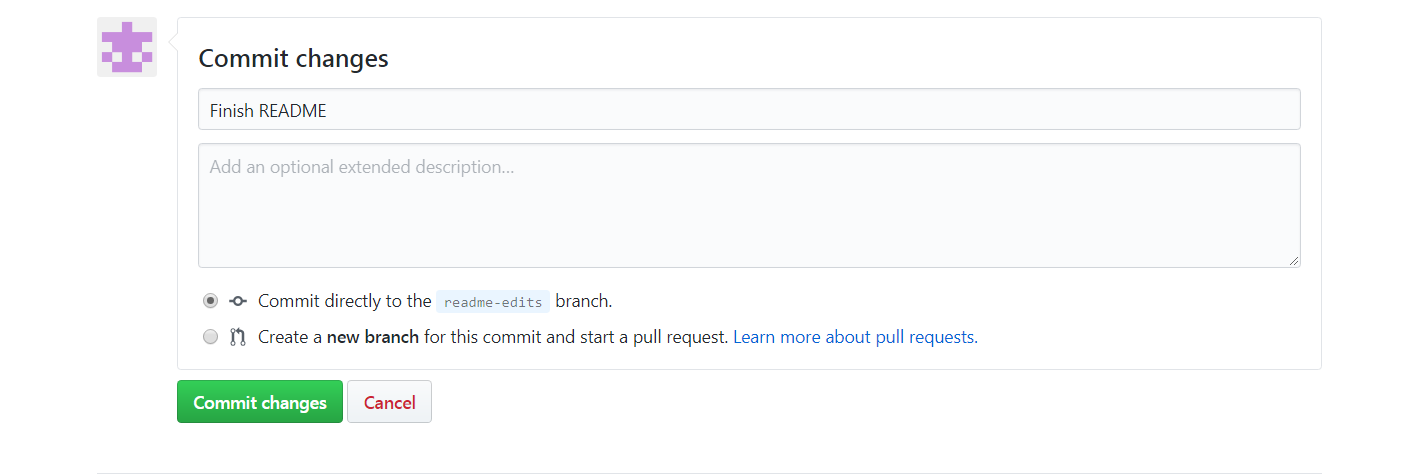
　　现在我们有两个分支，master和readme-edits，他们看起来一样，但是时间不会很长，接下来我们增加我们的改变在新的分支上。

**4：制作并提交更改**

　　现在我们在readme-edits分支的代码视图中，这是一个master的副本，让我们做一些修改吧。

* 1，单击该README.md文件
* 2，点击要编辑的文件上糊涂右上角的铅笔图标
* 3，在编译器中，写一点东西
* 4，编写描述更改的提交消息
* 5，单击提交按钮

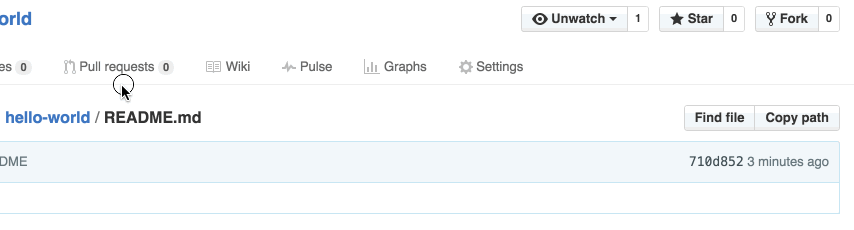




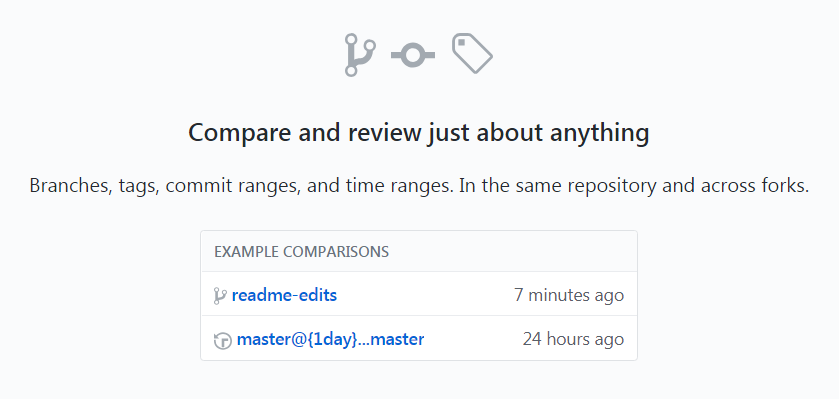
**5：打开Pull请求**

　　Pull Requests是GitHub上合作的核心，当我们打开拉取请求时候，你提出了更改并请求某人审核并提取我们的贡献，并将其合并到他们的分支中，拉请求显示来自于两个分支的内容的差异。

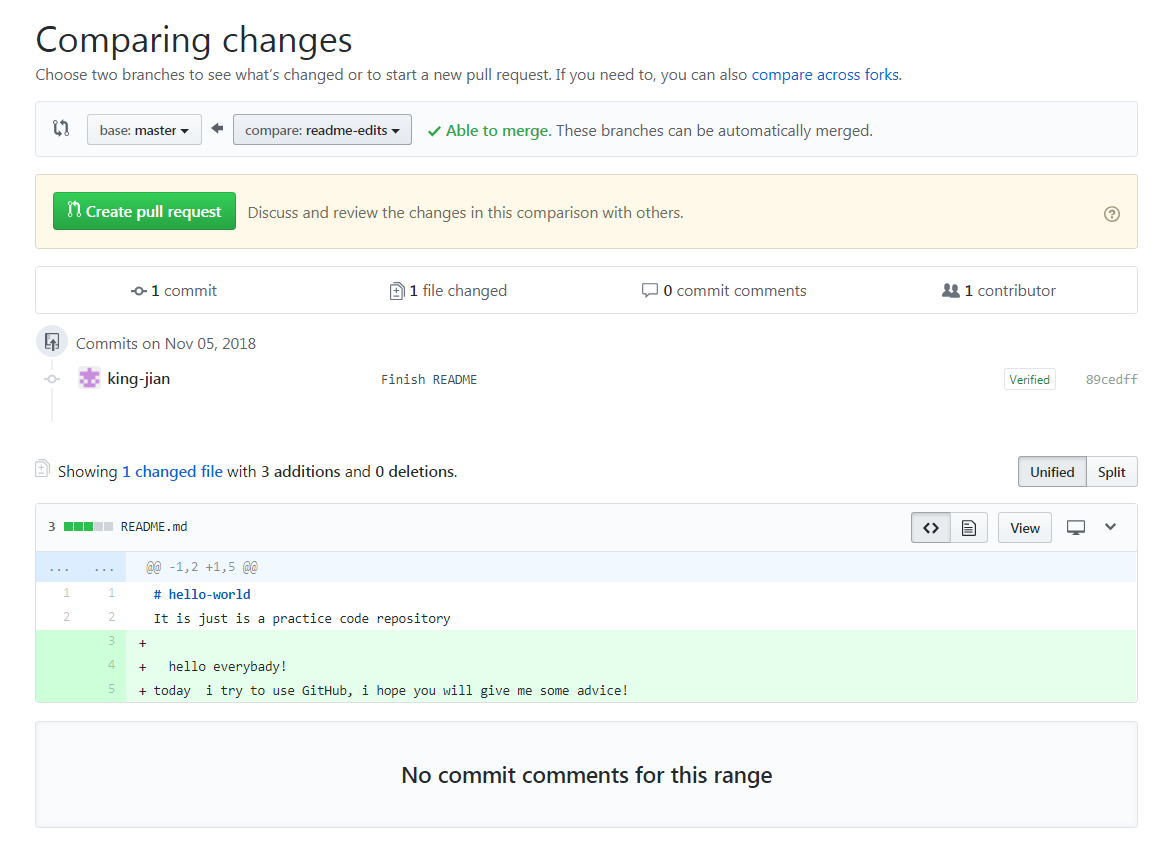
**5.1 点击Pull Requests**



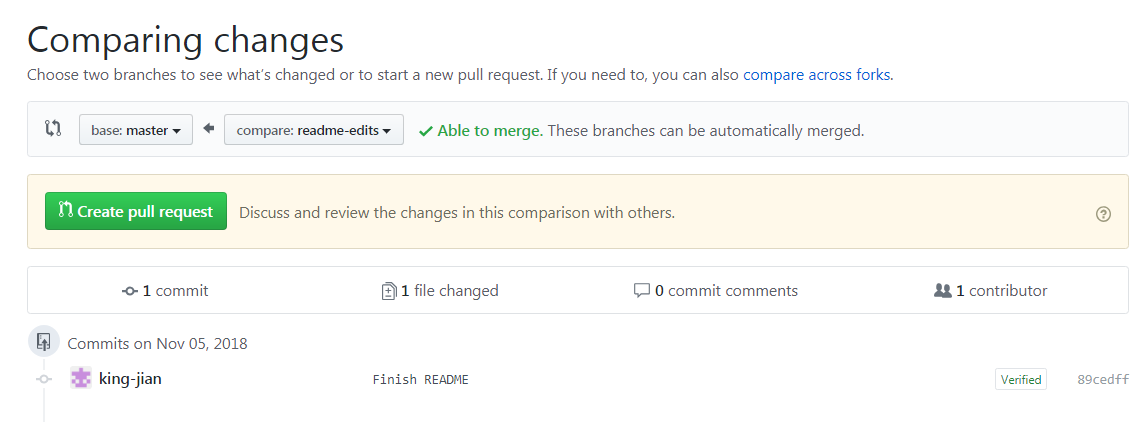
**5.2 在Example Comparisons中比对代码**



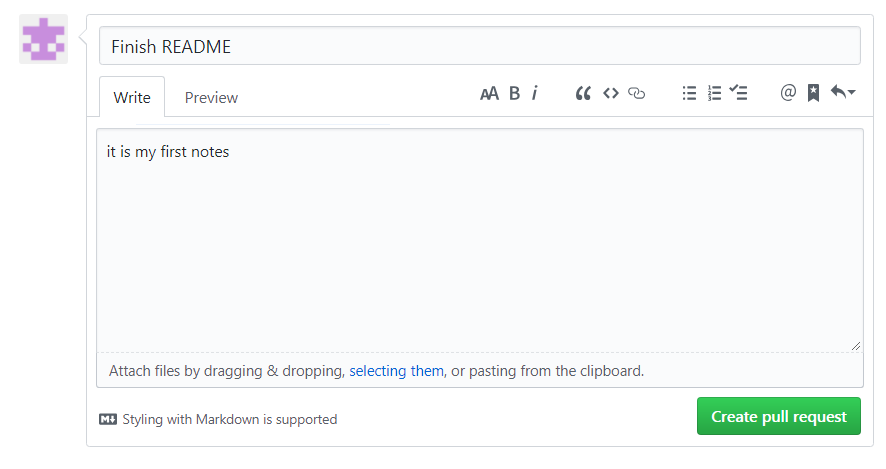
**5.3 在比较页面查看差异，确保其是我们要提交的内容**



**5.4 如果是，点击Create pull request**



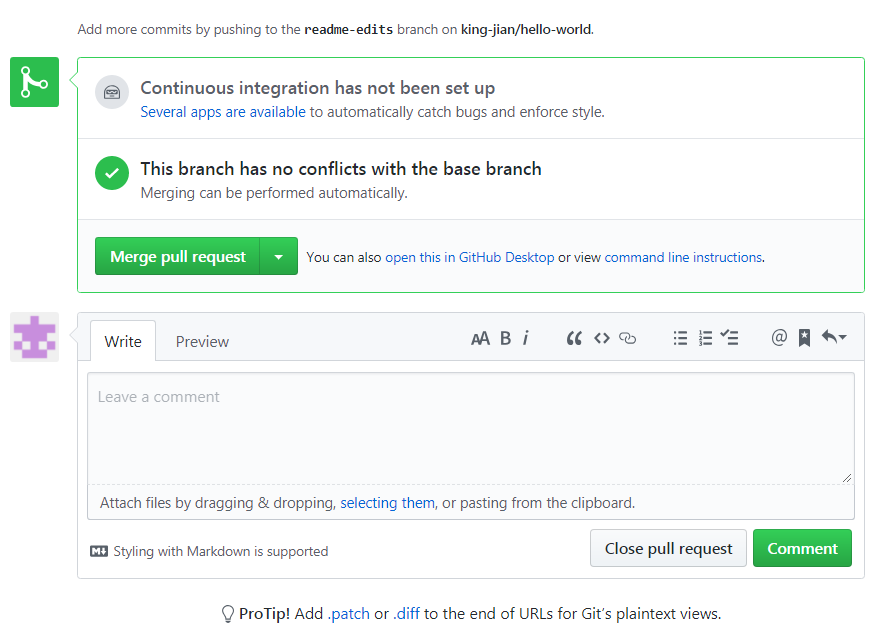
**5.5 为我们的拉取请求提供标题，并写下更改的简要说明**



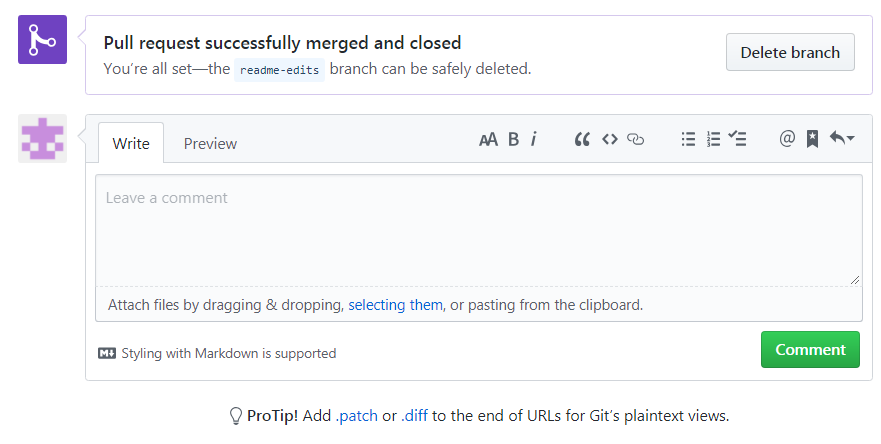
**6：合并我们的Pull Requests**

　　最后将我们的更改结合在一起，将我们的readme-edits分支合并到master分支上。

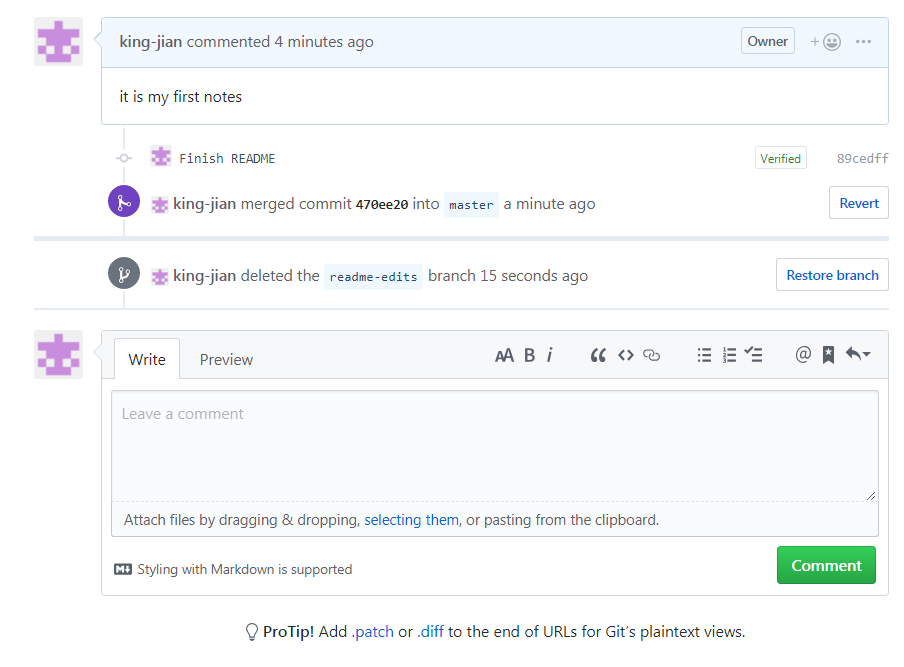
**6.1 单击Merge pull request，然后单击Confirm merge**



**6.2 Delete branch**

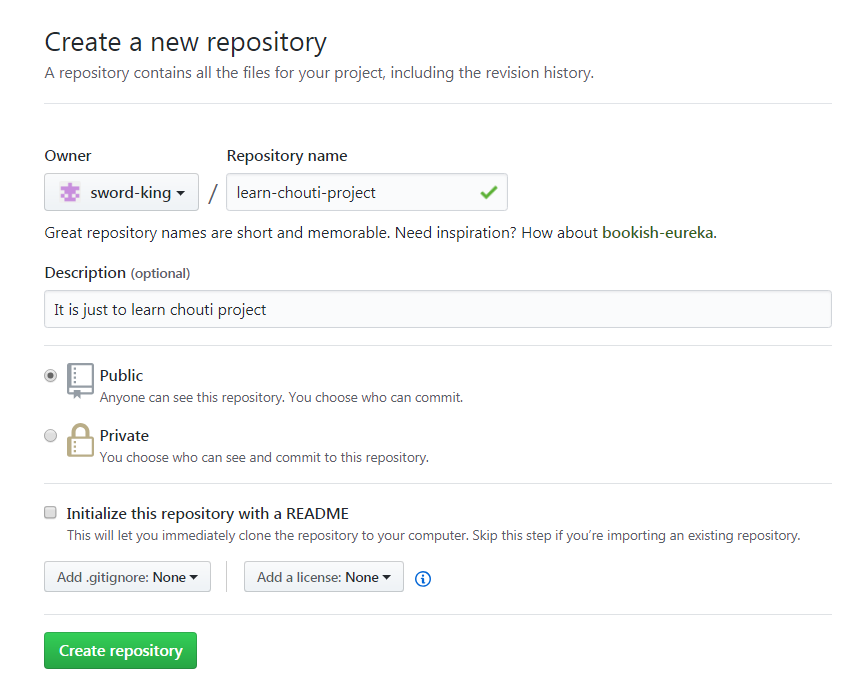


**6.3 恢复的话点击Restore branch**

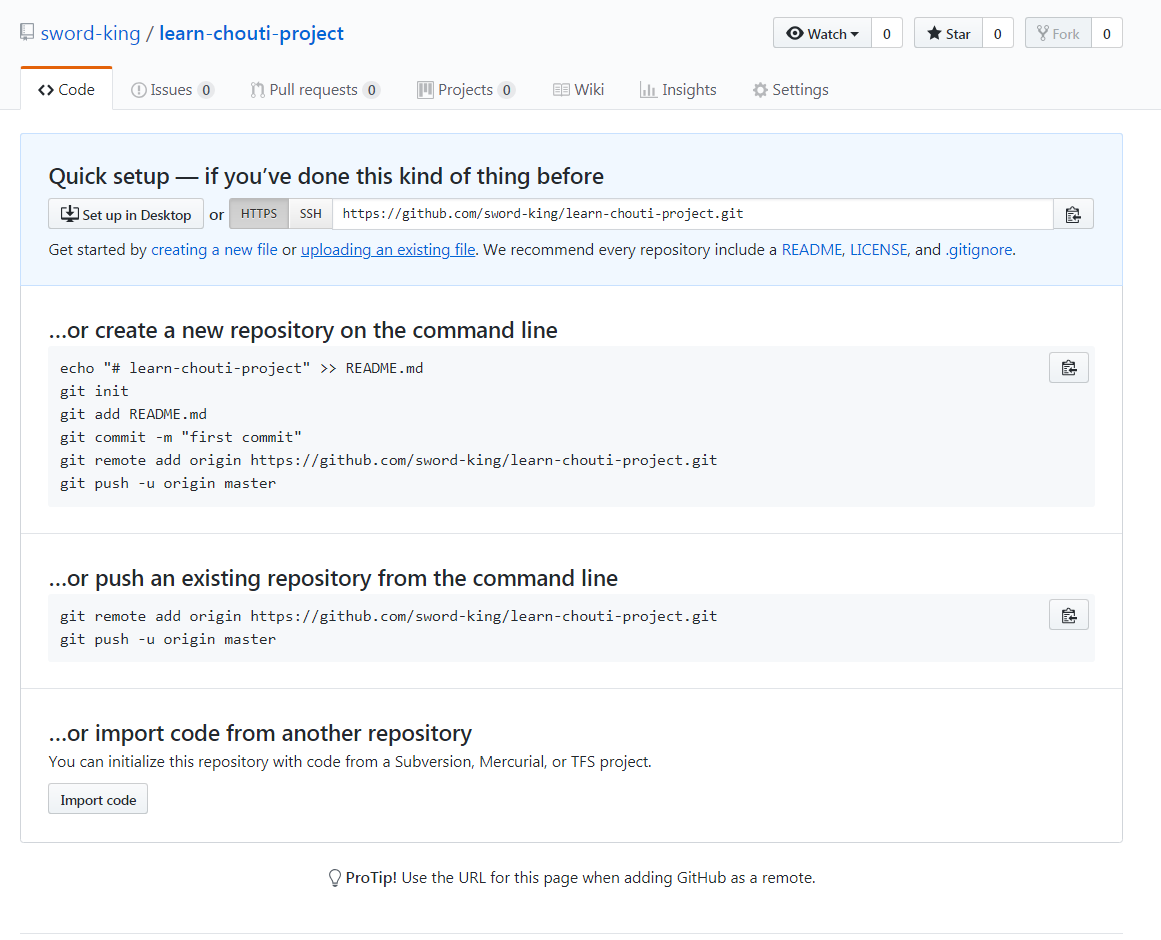


**二：如何一步步的在GitHub上传自己的项目**

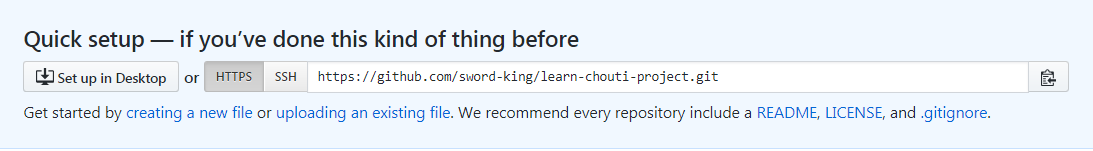
**1 创建一个新的项目，填写项目名称，描述等**



**2 创建完成后，跳转到下面页面**



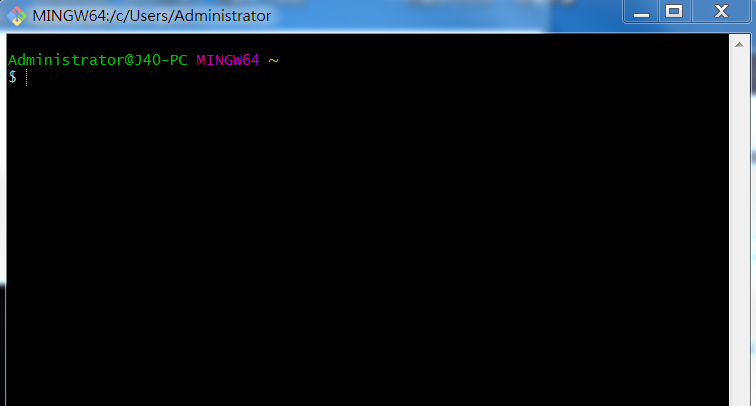
　　那么请记住下面的地址：



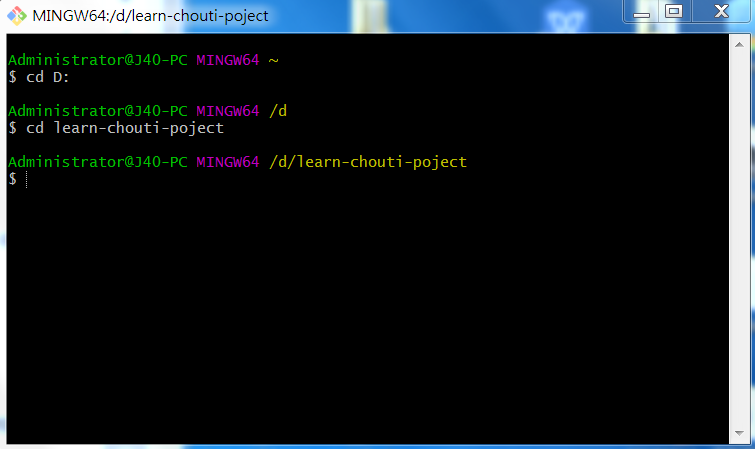
**3 我们需要下载Git，并安装。**

　　如果没有安装，请查看windows的安装博客（这里不做详细介绍）：[https://www.cnblogs.com/wj-1314/p/7993819.html](https://www.cnblogs.com/wj-1314/p/7993819.html" \t "_blank)

**4 进入Git Bash，出现如下界面**



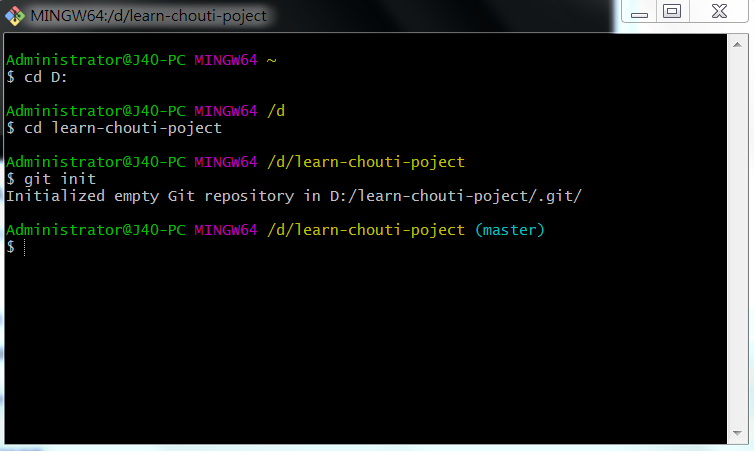
**5 cd 进入到我们放项目的地址**



**6 输入git init**

|  |  |
| --- | --- |
| 1 | 在当前项目工程下履行这个号令，相当于把当前项目git化 |

　　在当前项目的目录中生成本地的git管理（我们会发现当前目录下多了一个.git文件夹）



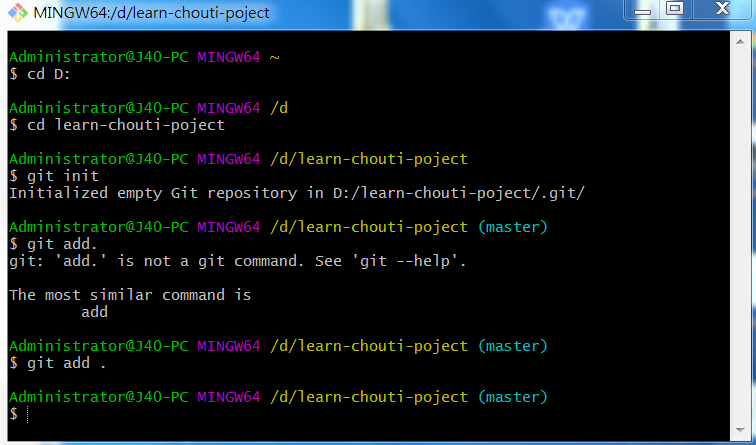
https://img2018.cnblogs.com/blog/1226410/201811/1226410-20181105113957127-38142511.png

**7 输入git add .**

|  |  |
| --- | --- |
| 1 | 把当前目次下代码参加git的跟踪中，意思就是交给git经管，提交到本地库 |

　　这个是将项目上所有的文件添加到仓库中，如果只想添加某个特定的文件，只需要将.换成特定的名称即可。

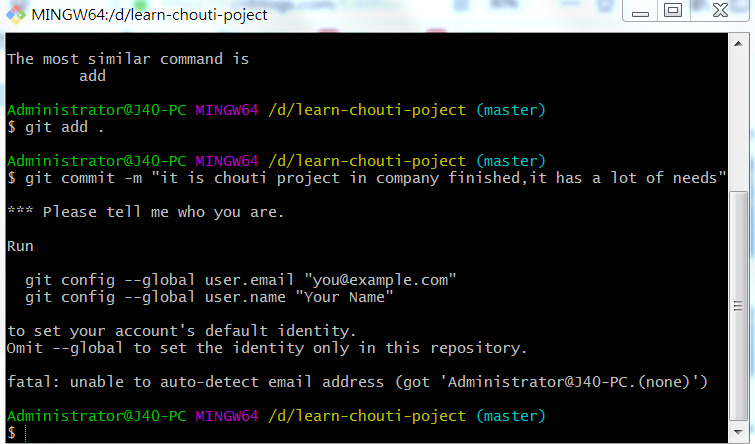
**（下面会报错，我们发现add 和.之间有空格）**



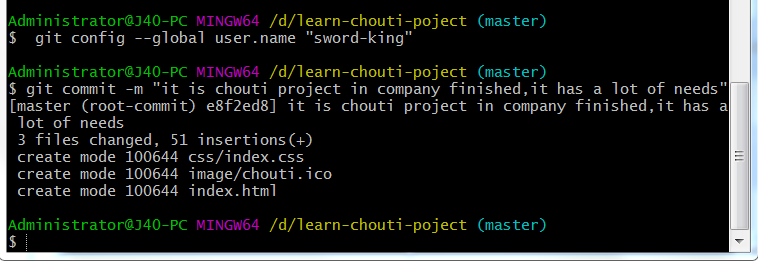
**8 输入git commit -m "first commit"**

|  |  |
| --- | --- |
| 1 | 相当于写点提交信息 |

　　表示我们对这次提交的注释，双引号里面的内容可以根据个人的需求改



**9 出现上面的内容，我们需要输出自己的账号或者名字，再执行上面的代码就会成功**

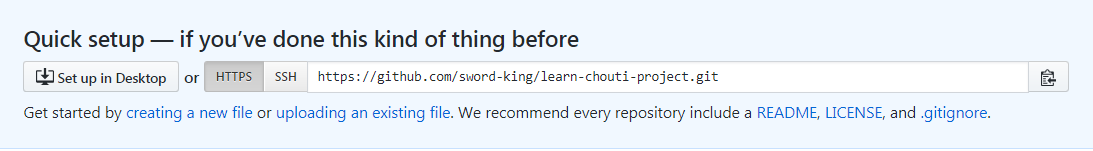


**10 关联自己的仓库url地址**

　　这里自己找自己的url地址

|  |  |
| --- | --- |
| 1 | git remote add origin https://自己的仓库url地址 |

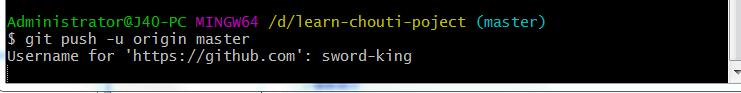
 　　下面展示本人的：

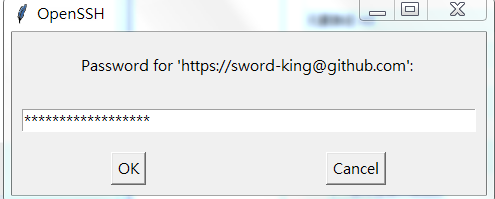


**11 上传代码 输入 git push -u origin master（意思：上传到GitHub仓库）**

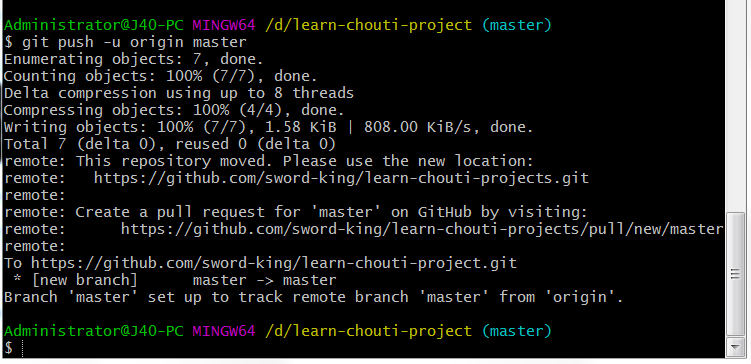
|  |  |
| --- | --- |
| 1 | 将本地库提交到github上。 |

　　执行完毕后，如果没有异常，会等待几秒，然后跳出一个让我们输入Username 和password的窗口，我们只需要输入个人的github登录账号和密码即可。

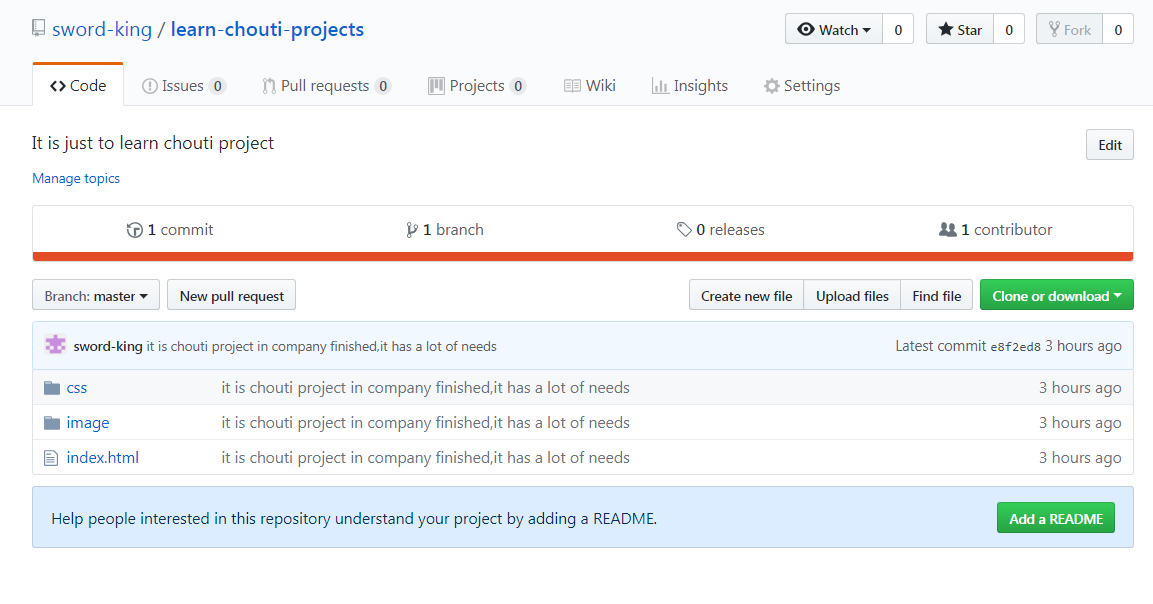




 　　最后上传完毕

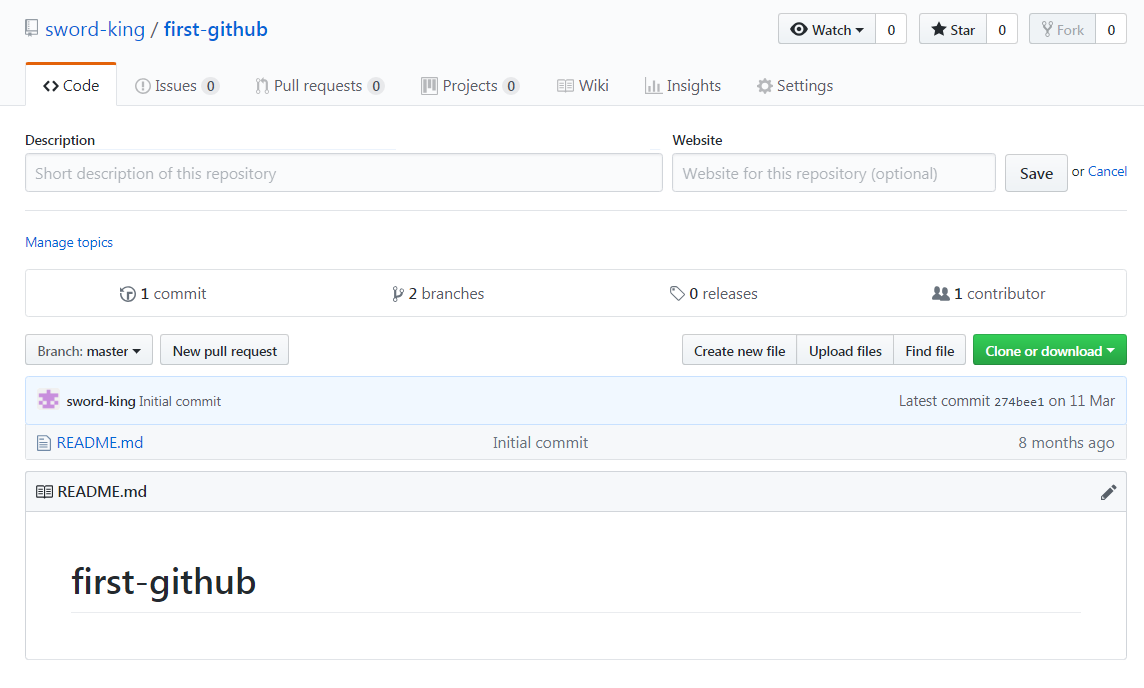


**12 上传成功，进入到GitHub中查看**

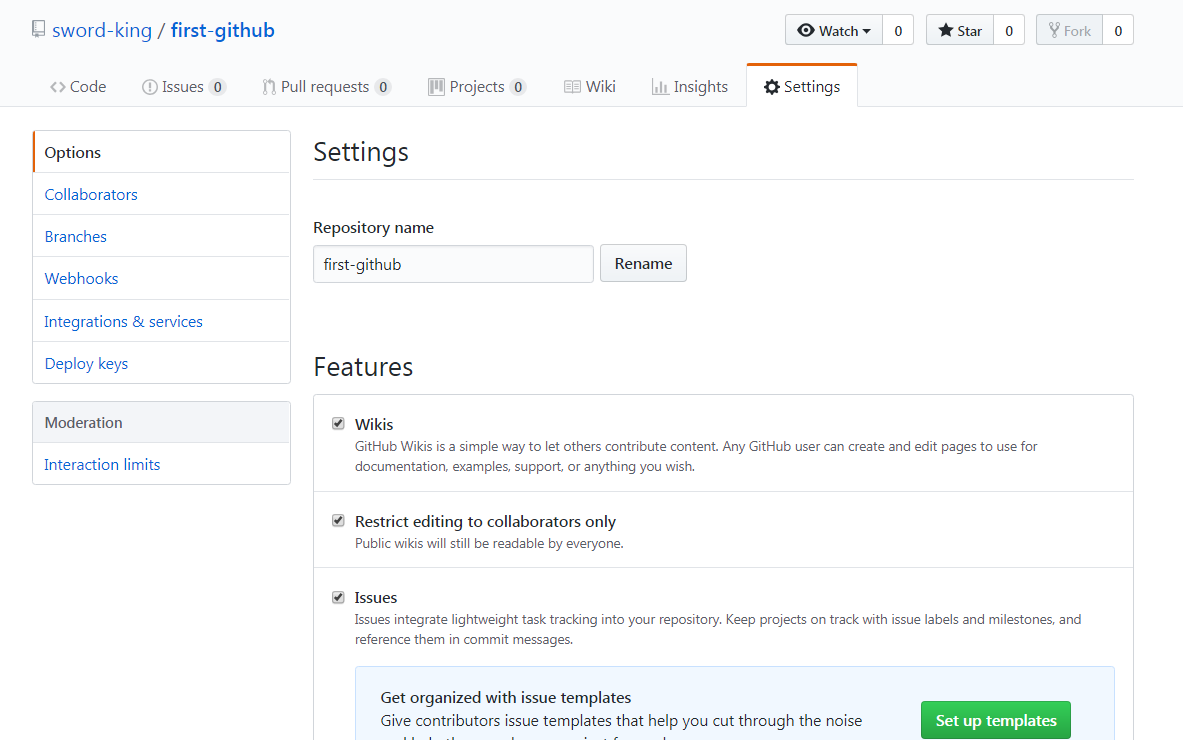


**三：GitHub如何删除项目**

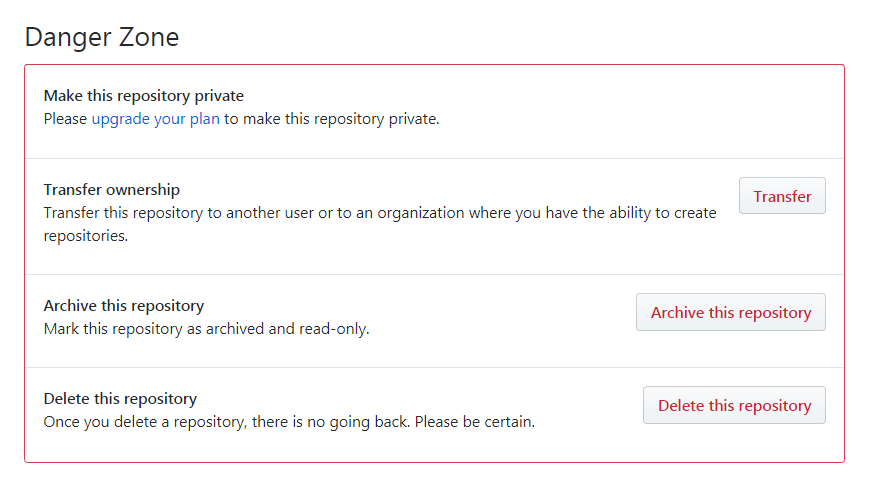
**1，首先找到需要删除的项目，点开**



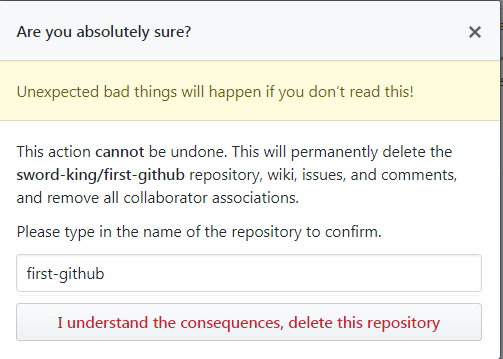
**2，找到settings，点开**



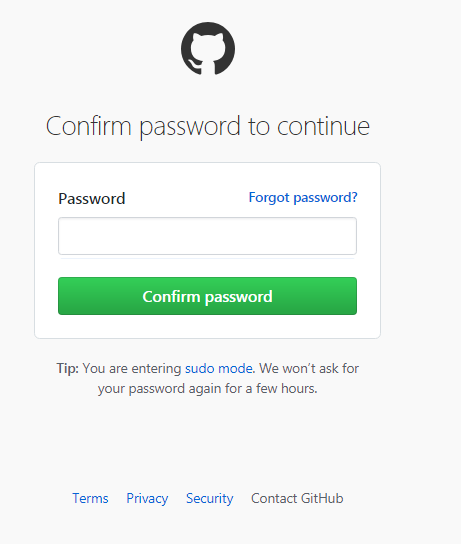
**3，将滚动条滑到底部，找到Danger Zone下的Delete this repository**



**4，点击，会弹出一个警告框，将该项目名称输入进行确认**



**5，这里会弹出账号重新进行确认，输入密码进行确认即可。**

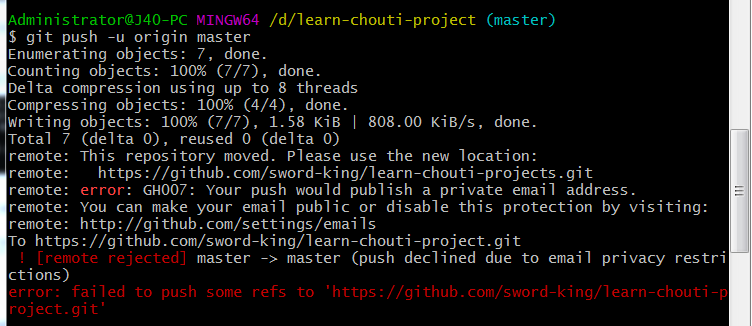


**6，删除成功后，会重新回到个人主界面提醒项目删除成功**

https://img2018.cnblogs.com/blog/1226410/201811/1226410-20181105110759389-1901254973.png

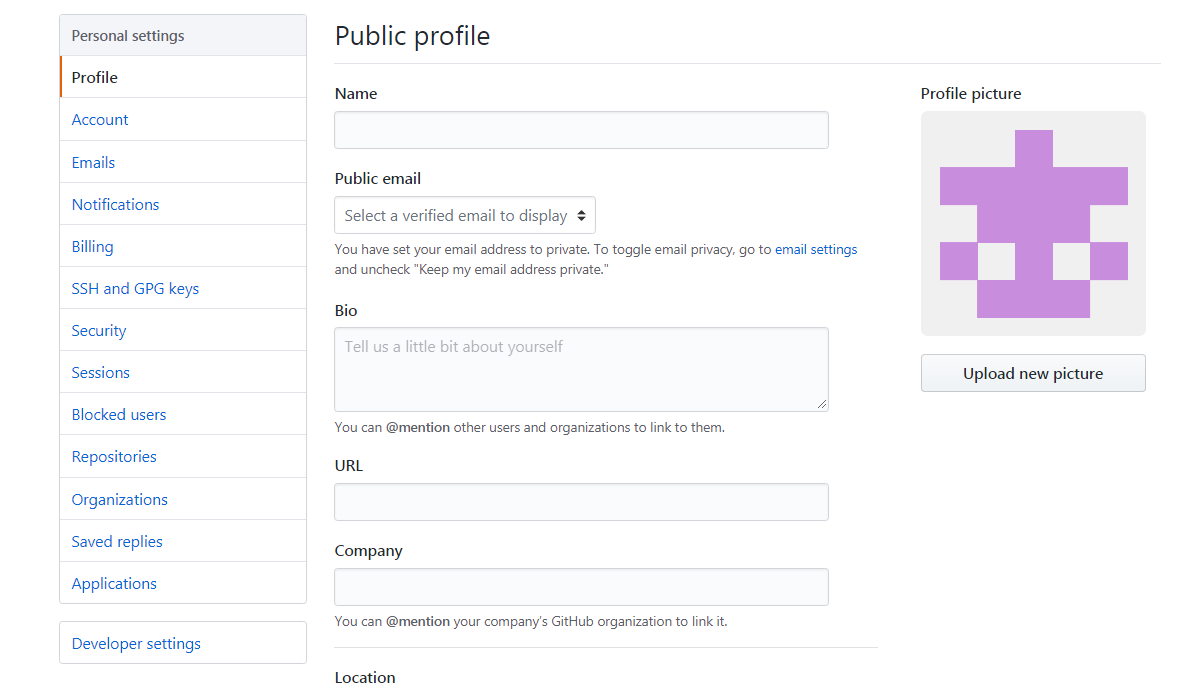
**四，报错push declined due to email privacy restrictions的解决方法**

　　当你上传代码到最后一步，发现无法push，并且会出现如下错误：

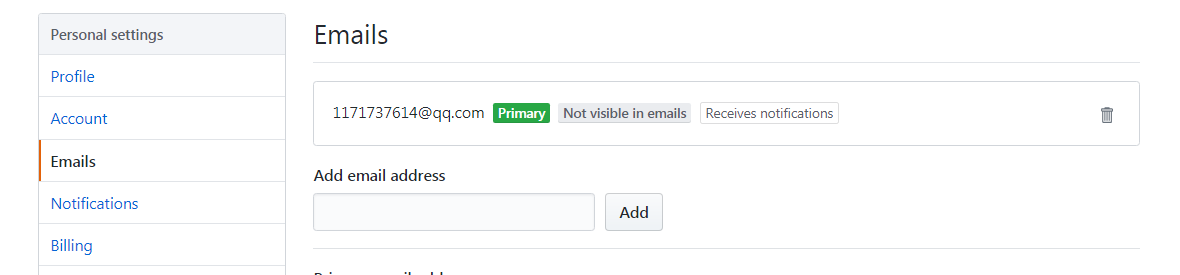


　　那么如何解决呢？

**1，进入GitHub主页，进入setting**



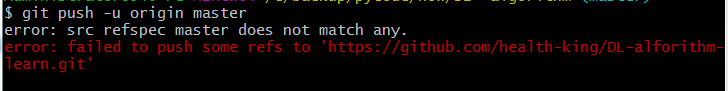
**2，点击emails**



**3，取消Block command line pushes that expose my email的勾即可**

**五，报错error: src refspec master does not match any.的解决方法**

　　上传代码到最后一步，出现此错误，如何解决呢？（就是无法匹配master）



**1：错误产生的原因**

　　引起该错误的原因是，目录中没有文件，空目录是不能提交上去的。

**2：解决方法**

　　这个仔细检查，本地的文件名称是否和GitHub上的对应。我的就是因为名字不对应导致的错误，也就是自己粗心！！！

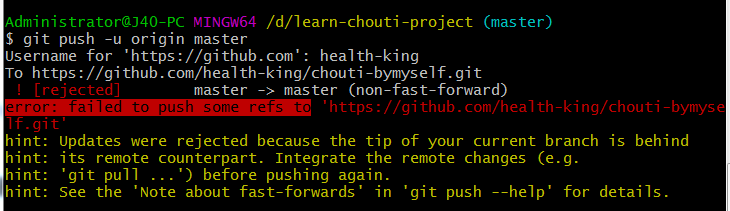
　　其二，就是之前没有使用如下代码：

|  |  |
| --- | --- |
| 1 | git add . |

　　　没有对代码进行跟踪，我找了两天，才发现自己每次都少了这一步，也会报同样的错误，非常粗心！！！

**六，报错error: failed to push some refs to的解决方法**

　　上传代码到最后一步，出现此错误，如何解决呢？



**1：错误产生的原因**

　　引起该错误的原因是，我们在GitHub中对代码进行了在线的修改；或者我们直接在GitHub上的某个库照片那个添加了readme文件或者其他文件，但是没有对本地库进行同步，所以这时候我们要想commit到remote 的GitHub库中就会有push失败的问题，

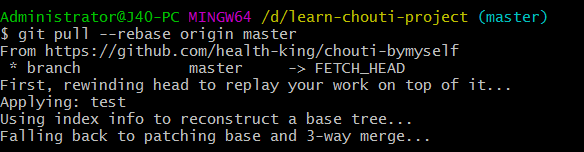
**2：解决方法**

　　这个问题就是因为远程库与本地库不一致造成的，我们只需要把远程库同步到本地库就可以了。指令如下：

|  |  |
| --- | --- |
| 1 | git pull --rebase origin master |

　　git pull –rebase origin master意为先取消commit记录，并且把它们临时 保存为补丁(patch)(这些补丁放到”.git/rebase”目录中)，之后同步远程库到本地，最后合并补丁到本地库之中。

　　如图所示：



然后再上传，指令如下：

|  |  |
| --- | --- |
| 1 | git push  -u origin master |

　　结果如图所示（此时解决问题）：

