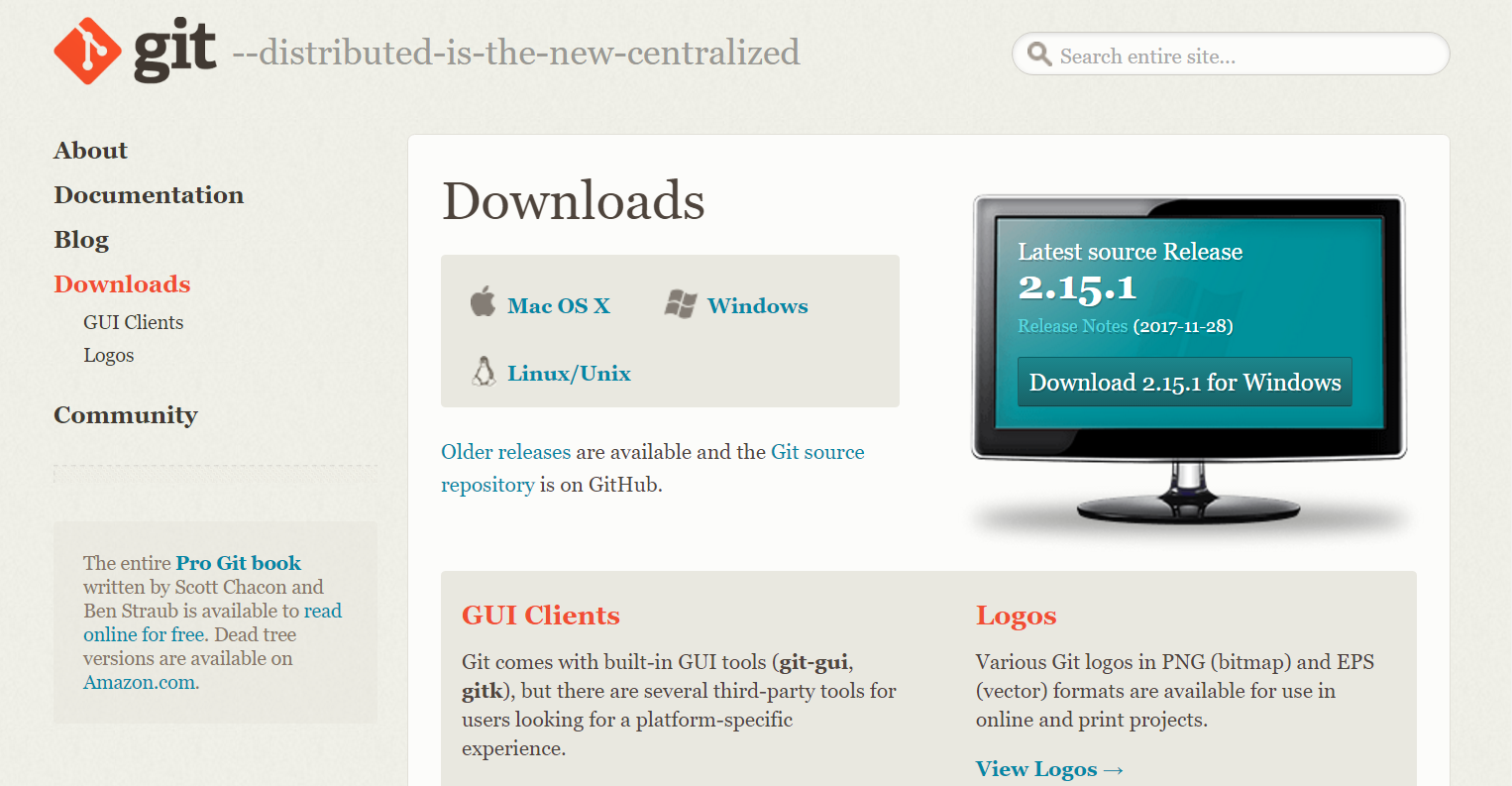
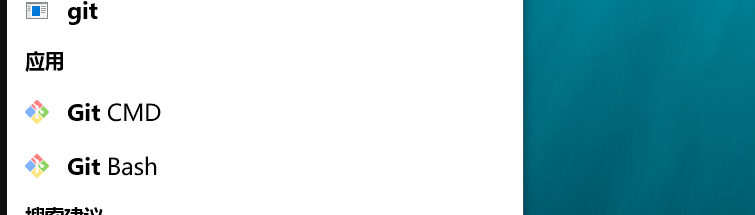
**1.下载Git软件：https://git-scm.com/downloads，据说ios自带的有git软件，这个我就不太清楚了。**



**2.下载之后安装就很简单了，一路下一步就可以了。安装完成后鼠标右击和者开始->程序会出现，打开Git Bash，进入bash界面。**



**接下来是重点，若这一步没有通过的话；秘钥没有认证完成，就不能上传文件，这个已经踩坑了，一直英语提示tell who are you ;**

**3.邮箱注册**

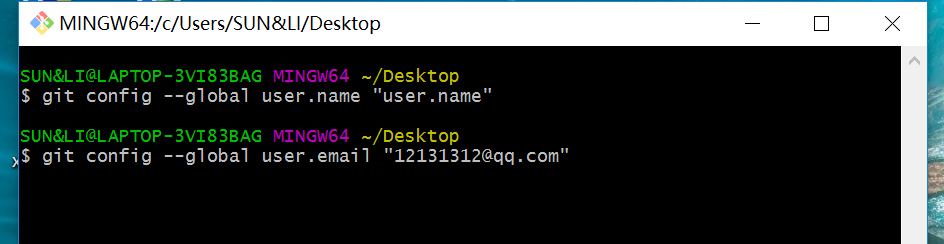
在git bash界面输入如下内容即可完成邮箱的注册：

$ git config --global user.name "user.name"

（说明：双引号中需要你的用户名，这个可以随便输入，比如“zhangsan”）

$ git config --global user.email "yourmail@youremail.com.cn"

（说明： 双引号中需要输入你的有效邮箱，比如“12131312@qq.com”）

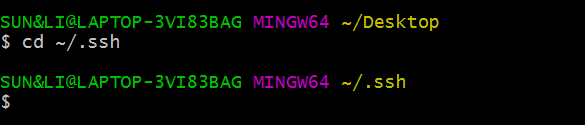


**4.查看是否存在密钥ssh keys**

**https://images2017.cnblogs.com/blog/1271277/201801/1271277-20180111173820301-1356290773.png**

若出现“No such file or directory”,则表示需要创建一个ssh keys。

我的秘钥已经设置过，如下面的状态：



**5.如果没设置，就创建新的ssh keys，不然git不能上传文件**

$ssh-keygen -t rsa -C "你的邮箱名"

$指定目录: C:\deskbook\（说明：若在此处不输入路径，而直接按回车，则ssh keys生成后存放的路径为C:\User\.ssh）

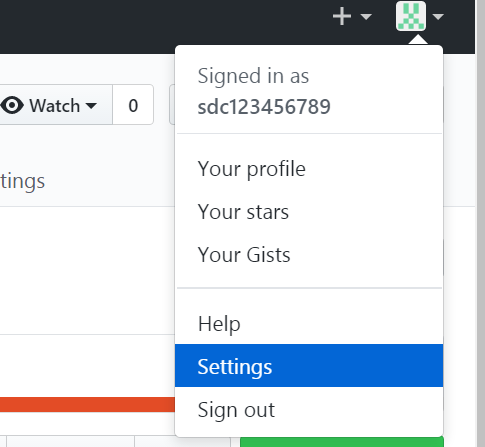
$输入密码: 123456

$确认密码: 123456

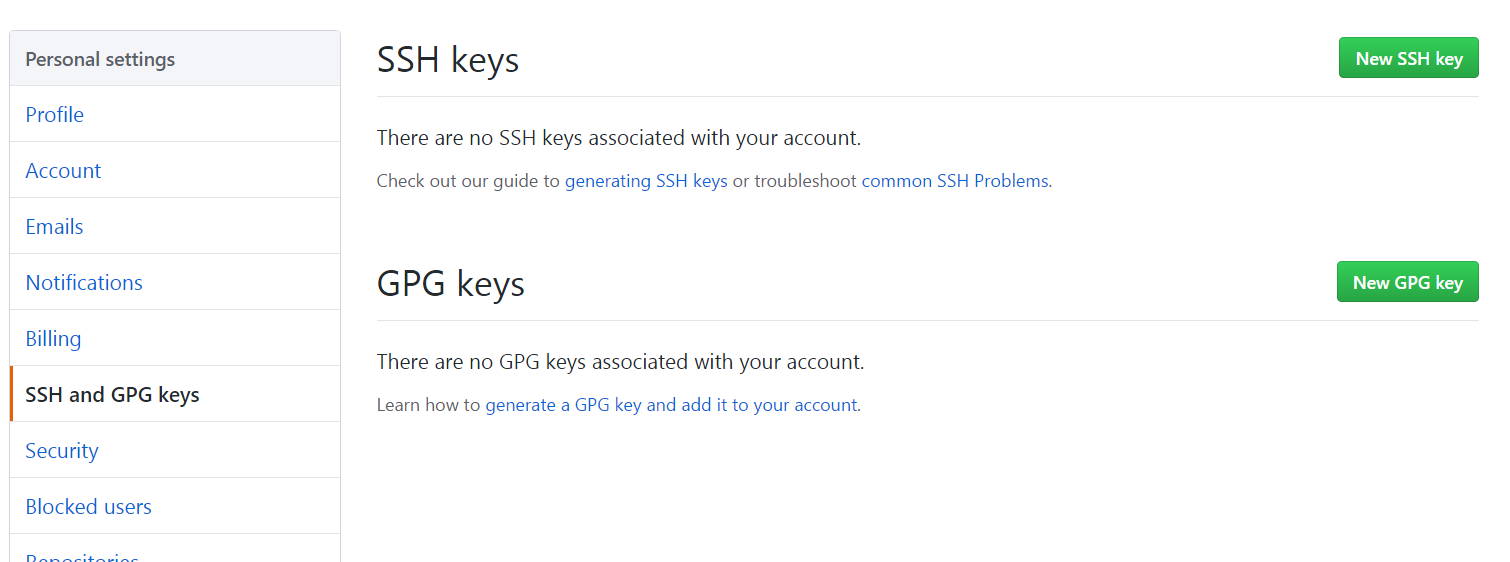
如此即可在C:\deskbook\文件夹中生成ssh keys。包括两个文件rd\_rsa和id\_rsa.pub

https://images2017.cnblogs.com/blog/1271277/201801/1271277-20180111174654957-1634964178.png

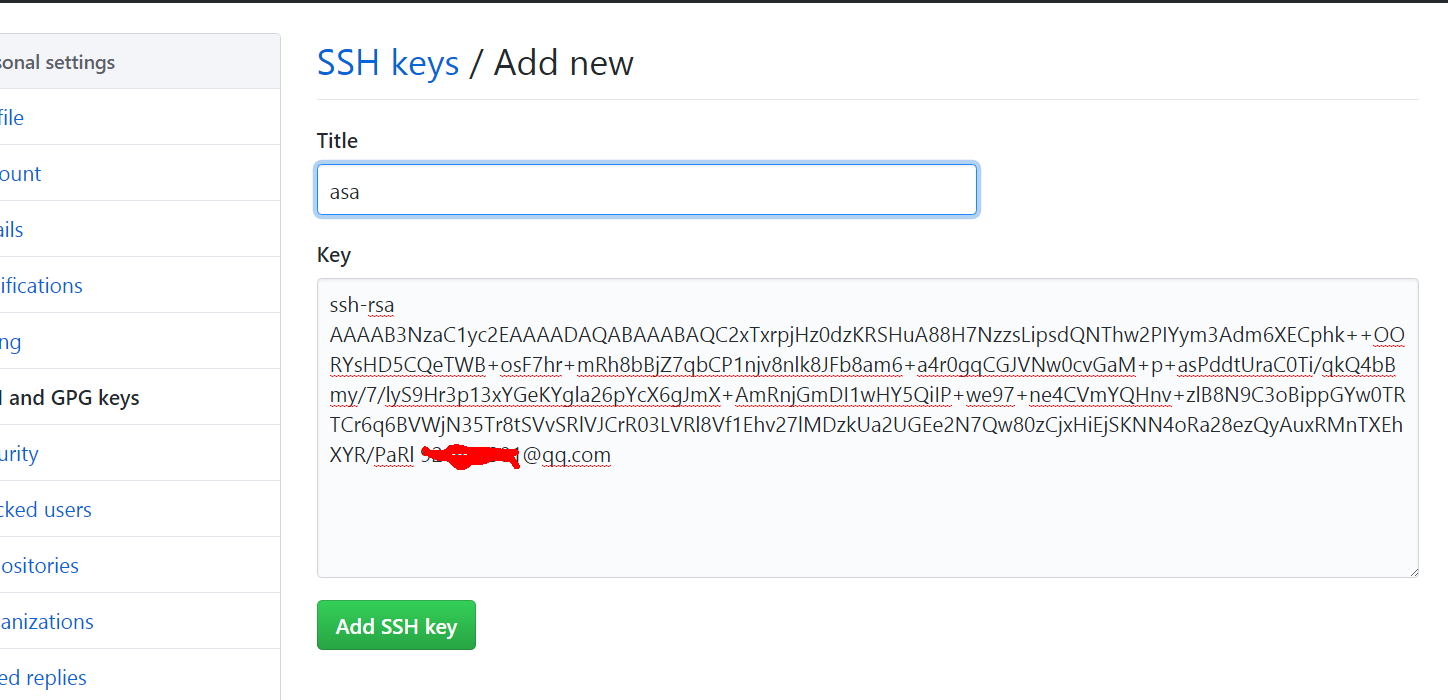
然后找到rd\_rsa和id\_rsa.pub所在目录打开idb\_rsa.pub（可以把后缀名改成.txt),登录自己的GitHub账号，找到Settings,



进入Settings后,点击SSH and GPG keys,然后再点击右上角添加新密钥按钮New SSH key,



然后，将idb\_rsa.pub里的内容拷贝到Key内，Title内容随便填，确定即可。

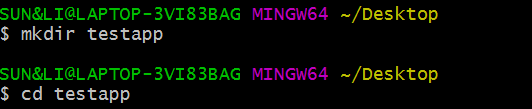


密钥添加完成，下次在使用时就不需要再添加密钥了。

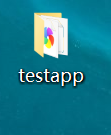
**6.开始上传本地文件到git上。**

我们需要先创建一个本地的版本库（其实也就是一个文件夹）。你可以直接右击新建文件夹，也可以右击打开Git bash命令行窗口通过命令来创建。

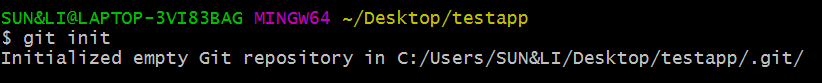
现在我通过命令行在桌面新建一个testapp文件夹（你也可以在其他任何地方创建这个文件夹），并且进入这个文件夹



这时桌面上就会出现一个文件夹，你可以将你的项目代码拷贝进来



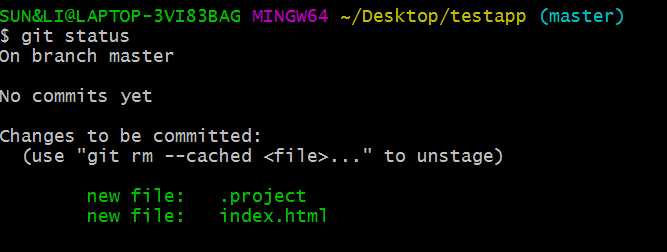
**7.通过命令git init把这个文件夹变成Git可管理的仓库**



这时你会发现TEST里面多了个.git文件夹，它是Git用来跟踪和管理版本库的。如果你看不到，是因为它默认是隐藏文件，那你就需要设置一下让隐藏文件可见。

https://images2017.cnblogs.com/blog/1271277/201801/1271277-20180111181252004-126982353.png

**8.这时候你就可以把你的项目粘贴到这个本地Git仓库里面（粘贴后你可以通过git status来查看你当前的状态），然后通过git add把项目添加到仓库（或git add .把该目录下的所有文件添加到仓库，注意点是用空格隔开的）。在这个过程中你其实可以一直使用git status来查看你当前的状态。如果文件内有东西会出现红色的字，不是绿色，这不是错误。**

****

这里提示你虽然把项目粘贴过来了，但还没有add到Git仓库上，然后我们通过git add .把刚才复制过来的项目全部添加到仓库上。

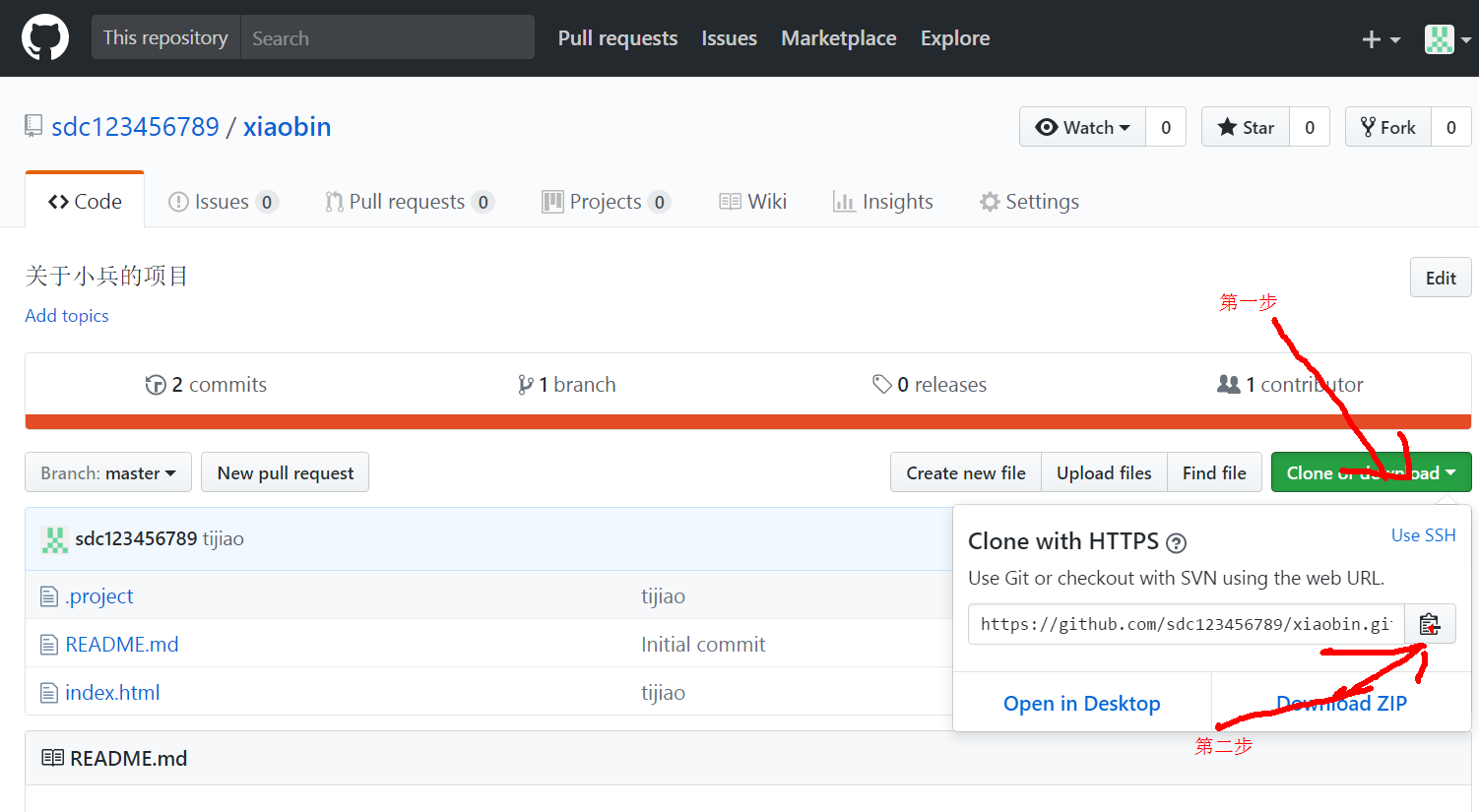
https://images2017.cnblogs.com/blog/1271277/201801/1271277-20180111181900801-1376666535.png

**9.用git commit -m "日志" 把项目提交到仓库。**

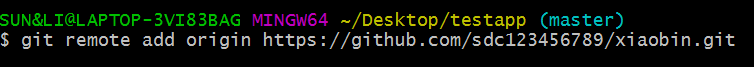


**10.在Github上创建一个Git仓库。**

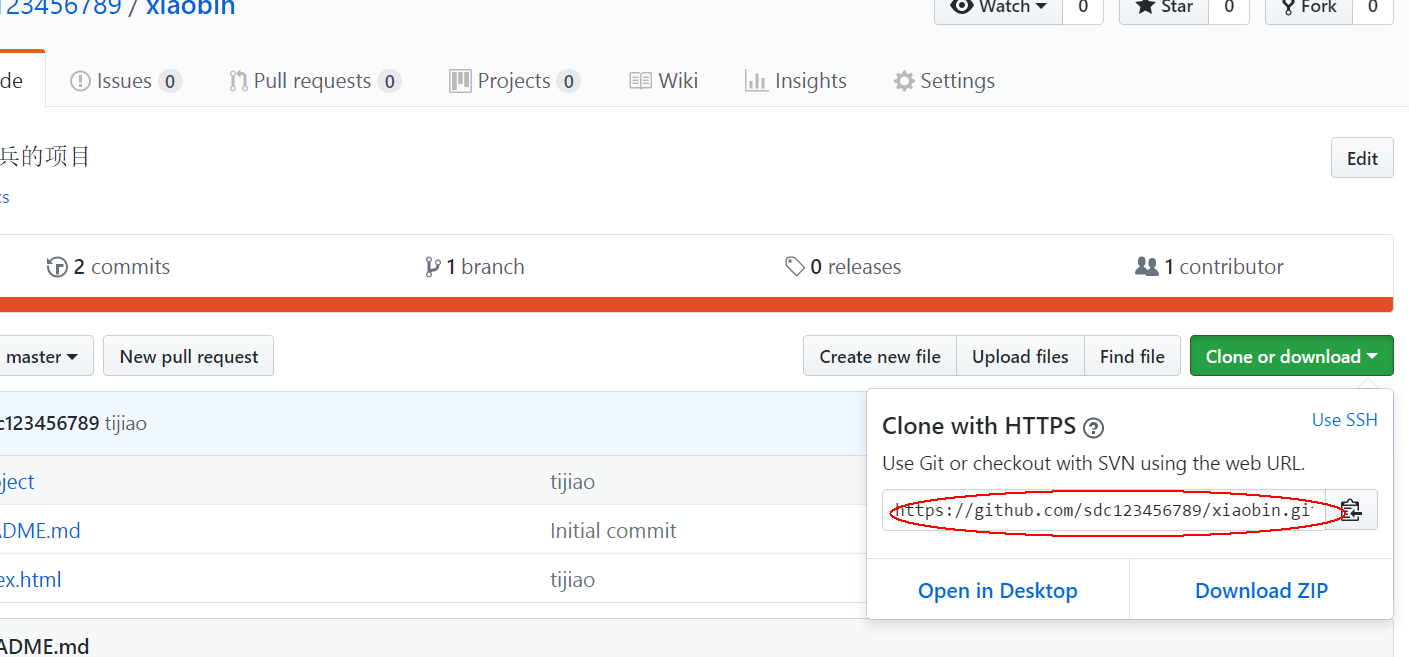
  你可以直接点New repository来创建，比如我创建了一个xiaobing的仓库.进入仓库



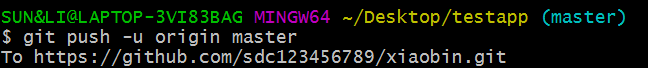
**11.在Github上创建好Git仓库之后我们就可以和本地仓库进行关联了，根据创建好的Git仓库页面的提示，可以在本地testapp仓库的命令行输入：**



注意origin后面加的是你Github上创建好的仓库的地址,就是上图第二步点击复制的地址。

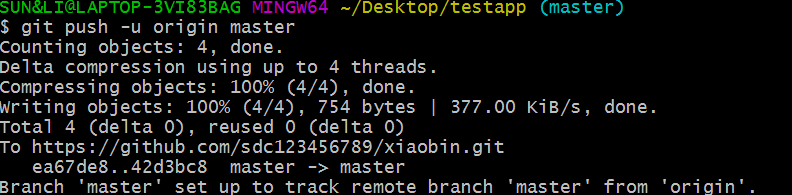


关联好之后我们就可以把本地库的所有内容推送到远程仓库（也就是Github）上了，通过：$ git push -u origin master

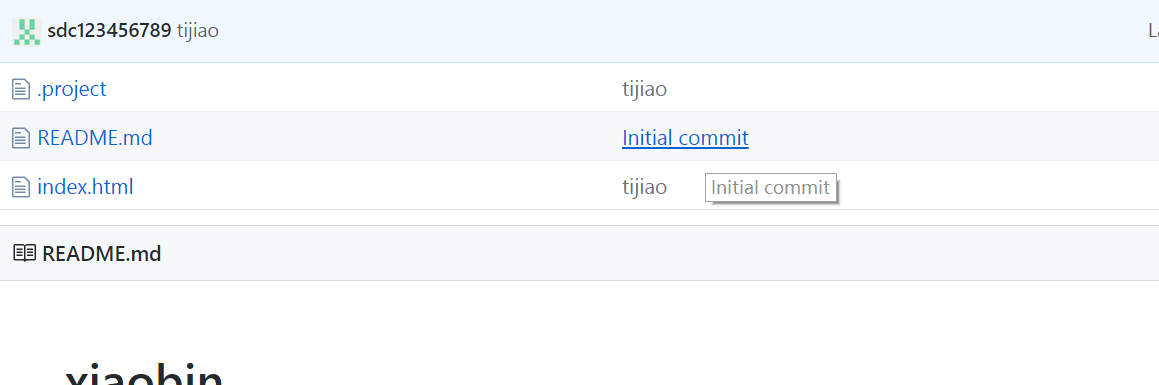


 由于新建的远程仓库是空的，所以要加上-u这个参数，等远程仓库里面有了内容之后，下次再从本地库上传内容的时候只需下面这样就可以了：

**$ git push origin master**  
  
上传项目的过程可能需要等一段时间，完成之后是这样的：

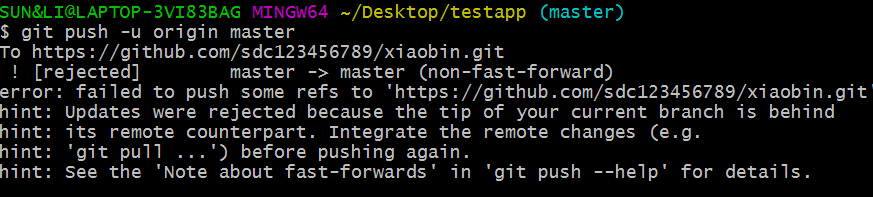


这时候你再重新刷新你的Github页面进入刚才新建的那个仓库里面就会发现项目已经成功上传了：



 至此就完成了将本地项目上传到Github的整个过程。

另外，这里有个坑需要注意一下，就是在上面第七步创建远程仓库的时候，如果你勾选了Initialize this repository with a README（就是创建仓库的时候自动给你创建一个README文件），那么到了第九步你将本地仓库内容推送到远程仓库的时候就会报一个to   https://github.com/sdc123456789/xiaobin的错



 这是由于你新创建的那个仓库里面的README文件不在本地仓库目录中，这时我们可以通过以下命令先将内容合并以下：

**$ git pull --rebase origin master**

https://images2017.cnblogs.com/blog/1271277/201801/1271277-20180111184252785-1951335363.png

  这时你再push就能成功了。

   总结：其实只需要进行下面几步就能把本地项目上传到Github

　　注意：如果中途弹出输入框让你填写用户名和密码，只需把GitHub的账号和密码填写上即可。

     1、在本地创建一个版本库（即文件夹），通过git init把它变成Git仓库；

     2、把项目复制到这个文件夹里面，再通过git add .把项目添加到仓库；

     3、再通过git commit -m "注释内容"把项目提交到仓库；

     4、在Github上设置好SSH密钥后，新建一个远程仓库，通过git remote add origin https://github.com/guyibang/TEST2.git将本地仓库和远程仓库进行关联；

     5、最后通过git push -u origin master把本地仓库的项目推送到远程仓库（也就是Github）上；（若新建远程仓库的时候自动创建了README文件会报错，解决办法看上面）。

 以上只是我初次使用的总结，后面有时间我会把git工具小乌龟使用进行总结。