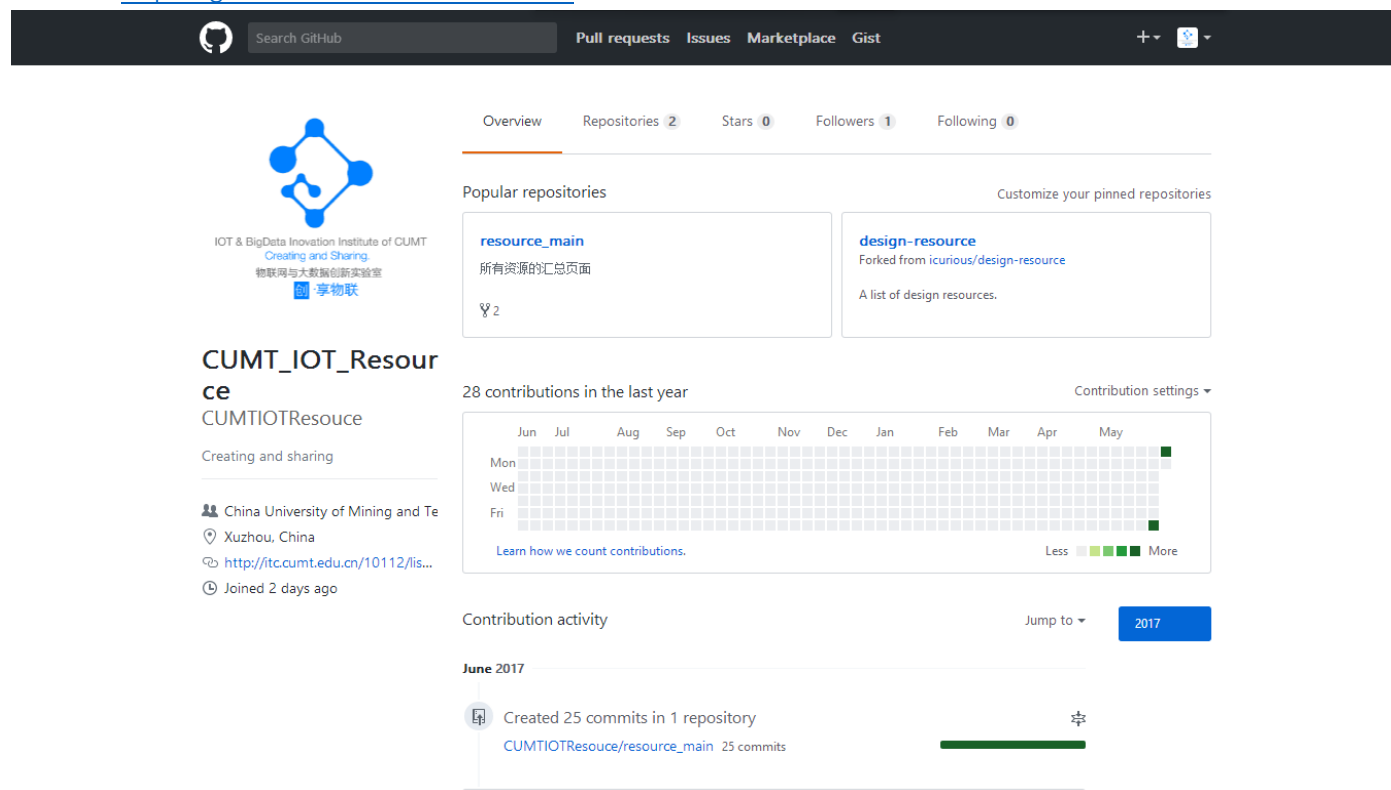


实验室公共 Github 资源账号 Q&A

[@Author:iCurious @Time:2017-06-12 @Rights:All rights reserved by CUMT IOT & BIGDATA Institute.]

主界面：<https://github.com/CUMTIOTResource>



账号：cumt_iot_resource@163.com

● Github 是什么？有什么用？

[Github 介绍及功能](<https://www.zhihu.com/question/20070065>)

[Git 教程]

(<http://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/001373962845513aefd77a99f4145f0a2c7a7ca057e7570000>)

● 这个项目还有账号是用来干嘛的？

这个项目主要是面向物联网实验室成员，一起分享和整合现有的资源，做到有传承，有干货。

每个人都可以成为其中的贡献者，将自己认为好的，值得分享的资源，或者之前做过的，能够开放源代码的项目，竞赛等等贡献出来，分享给本局的，以及后面的成员。由于是免费的账户，所以默认只能是 public 的，非实验室的成员也可以看到主页面上的内容。这个账号还有页面不直接提供资源以供下载，而是一个开放项目。如果你想添加资源和资料，比如视频资料，你可以存放在例如百度网盘上，然后生成链接和分享码，只需要将链接和分享码添加到 https://github.com/CUMTIOTResource/resource_main 的 readme 文件中就可以了。

你可以系统地学习 Git 知识及命令，从头探索 Github 的使用。

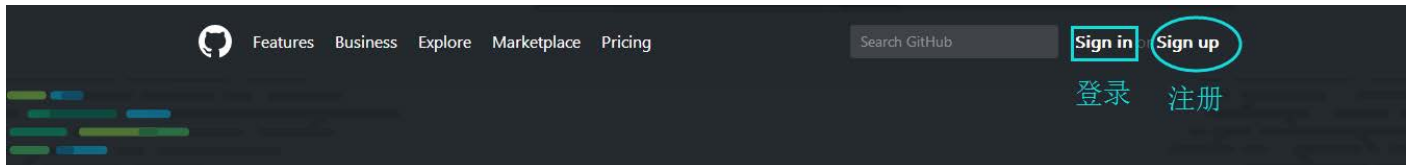
也可以不必掌握任何相关操作和技巧，只需要认识几个基本的英文单词，就可以参与到本实验室资源整理和维护的项目中。

本项目纯属自愿，但实验室内部至少应当周期性维护，比如有同学参加了电子设计大赛，完成了较为完整的工程项目，并且愿意开源出来，就可以让他/她/他们将自己的作品设计方案，文档，源代码等资料的链接添加到

https://github.com/CUMTIOTResource/resource_main 中（注意，不是把全部文件传到本项目中，这样整个项目将会庞大不堪，甚至很容易超过容量）。

● 如果我想成为本项目的贡献者 Contributors

1. 搜索或者直接登录 github.com

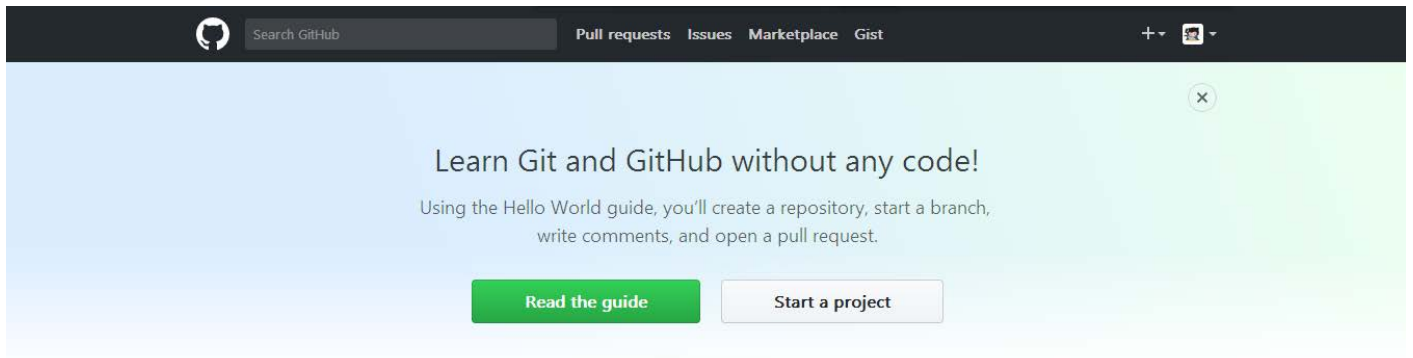


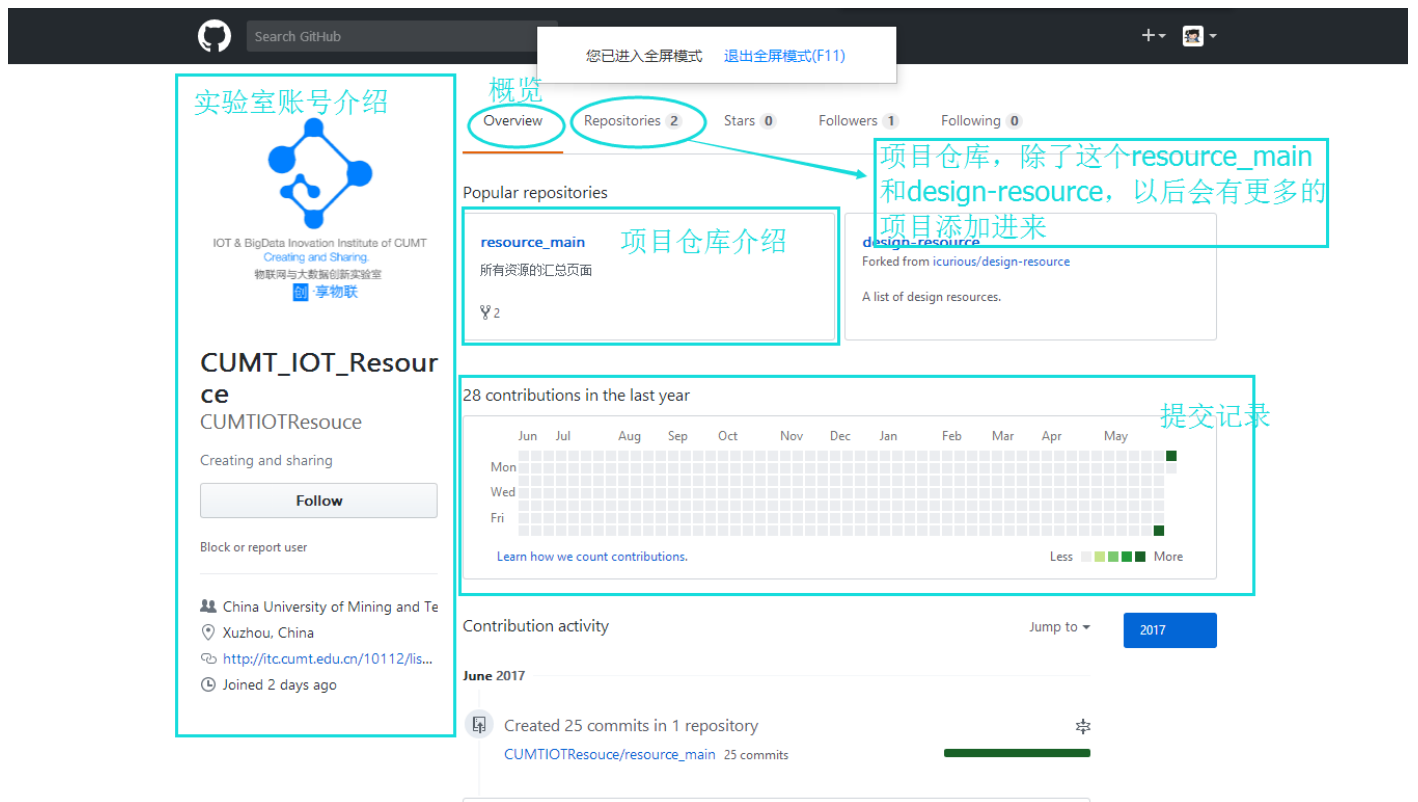
2. 注册 Github 账号

A screenshot of the GitHub 'Join GitHub' registration page. The page has a light gray background. At the top, it says 'Join GitHub' and 'The best way to design, build, and ship software.' Below this is a three-step process: Step 1: Create personal account, Step 2: Choose your plan, and Step 3: Tailor your experience. The 'Create your personal account' section is active. It contains three input fields: 'Username' (with the placeholder '昵称/用户名'), 'Email Address' (with the placeholder '自己的邮箱'), and 'Password'. A red arrow points from the 'Email Address' field to a text box on the right that says '推荐采用163/outlook/hotmail/gmail/cumt邮箱等'. Below the input fields is a green 'Create an account' button. To the right of the input fields, there is a section titled 'You'll love GitHub' with a list of benefits: 'Unlimited collaborators', 'Unlimited public repositories', 'Great communication', 'Frictionless development', and 'Open source community'.

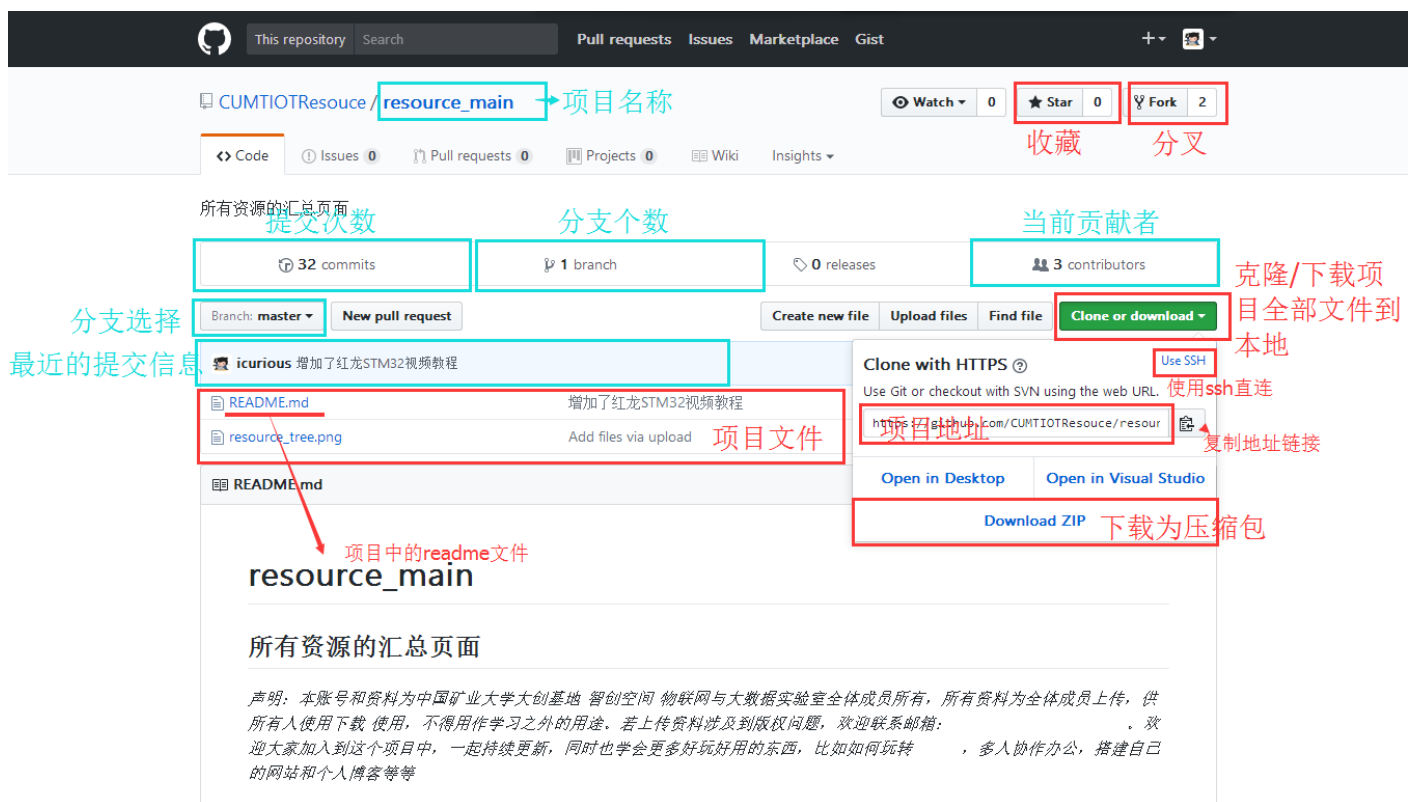
3. 到自己的邮箱中激活账号

4. 开始自己的项目，直接在浏览器地址栏中输入实验室资源账号主界面：<https://github.com/CUMTIOTResource>

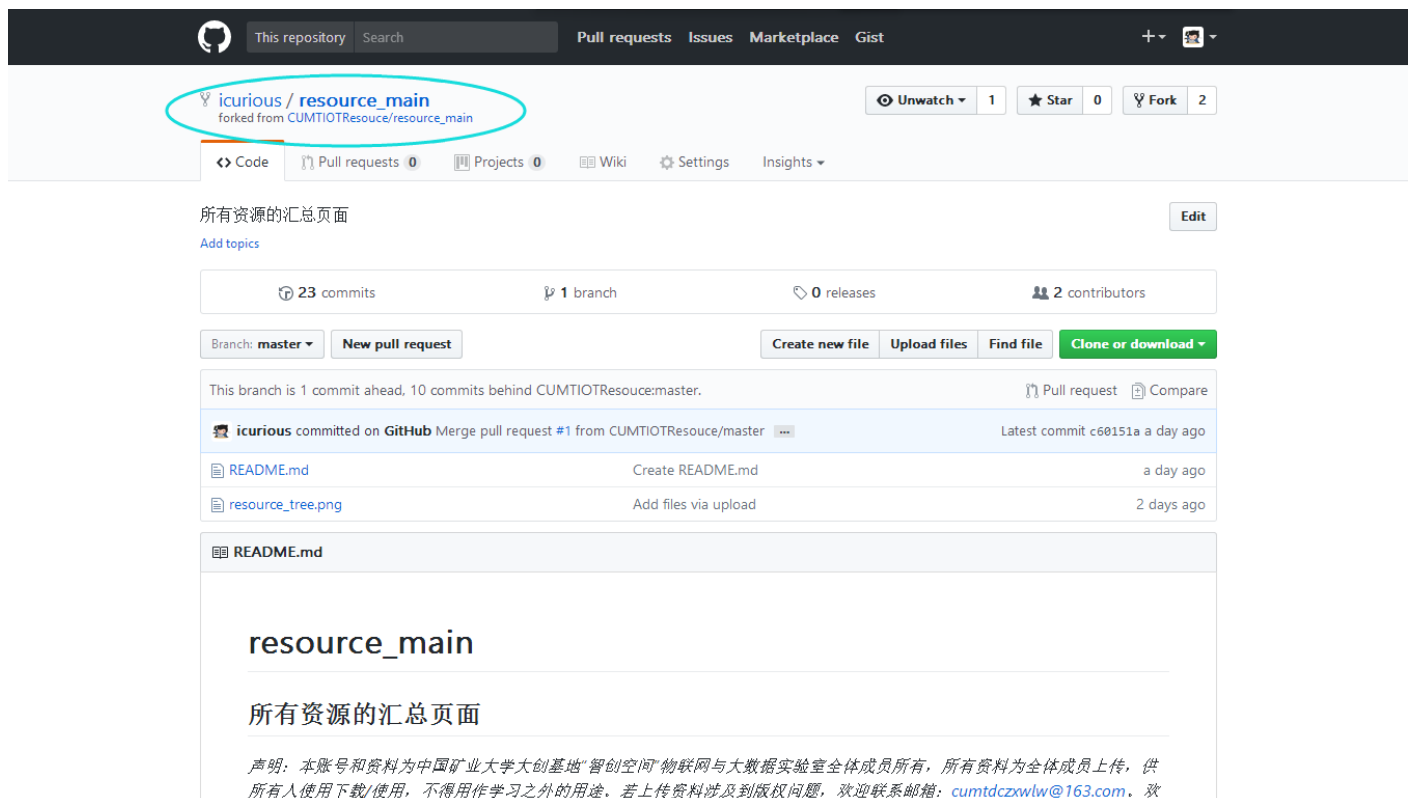
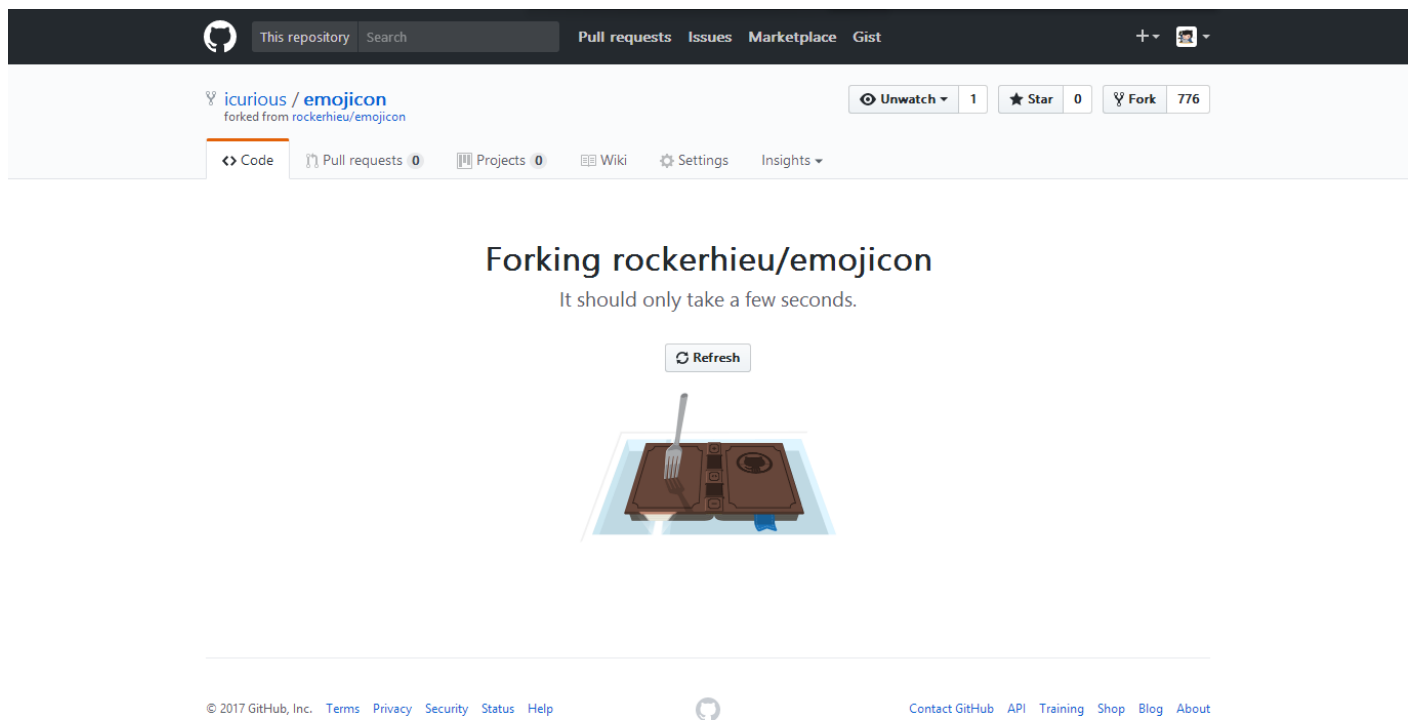




5. 点击 resource_main 这个项目（或者其他你感兴趣的项目）

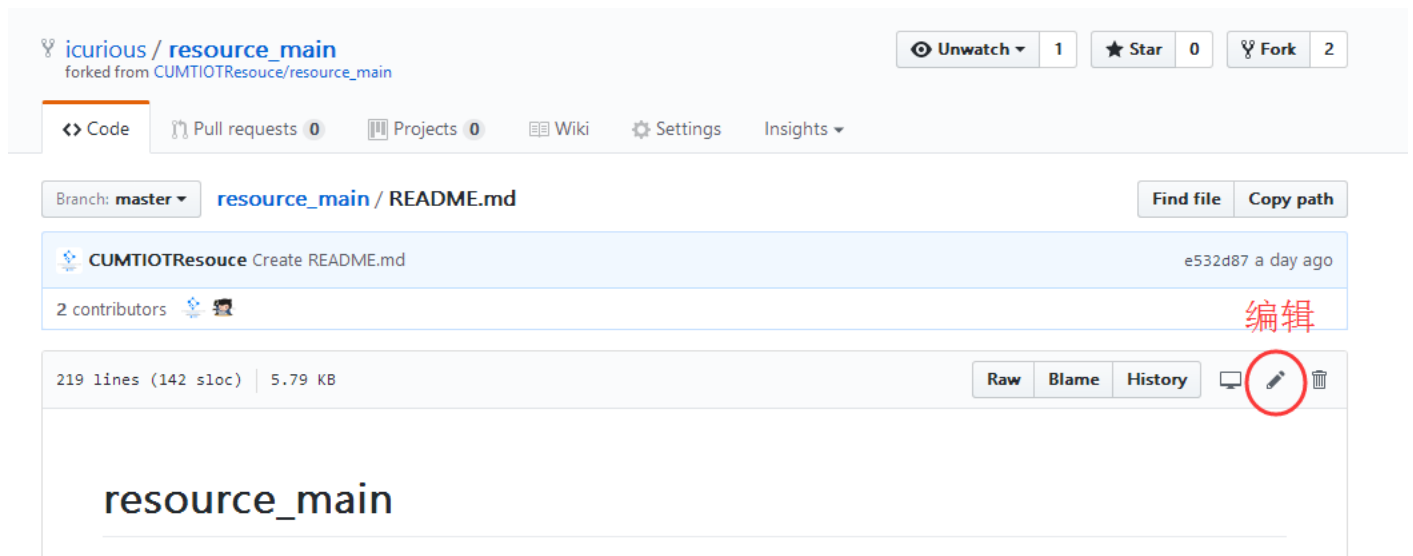


6. 点击 resource_main 这个项目（或者其他你感兴趣的项目）页面右上角的 fork 图标,你就拥有了和项目主干一模一样的工程文件了。



7. 点击 README.md 文件

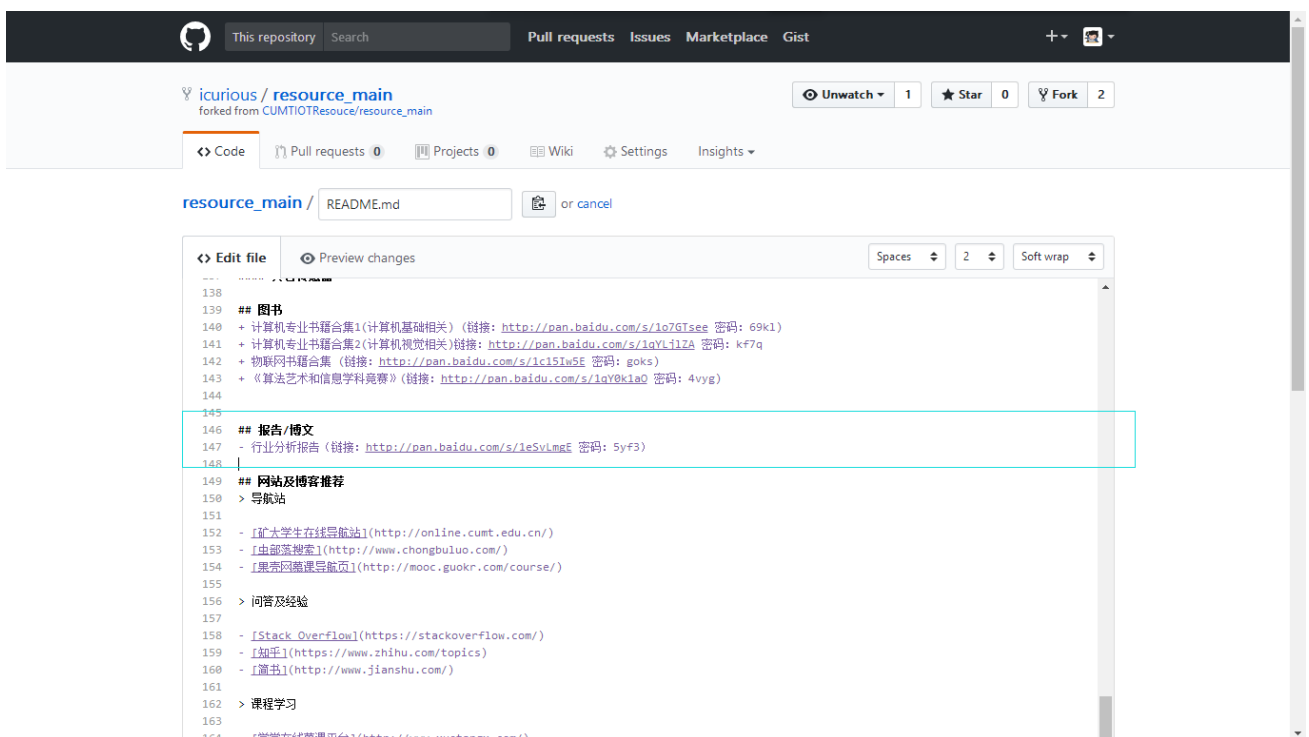
8. 点击编辑



9. 将你想要分享资源的链接添加进其中
比如，这里想分享一下百度网盘中的行业分析报告



先用百度网盘自带的工具生成链接和密码，复制。
然后粘贴到 readme 文件中的相应位置。



10. 确认变更。这个时候，你的这个分支的内容就变了。可以反复更改，随心所欲，不会影响到主干。

Commit changes

添加了行业分析报告

变更留言

来自麦肯锡，胡润百富，中国信息通信研究院，清科，罗兰贝格，新浪，兰德RAND，波士顿资讯（BCG）的专业行业分析报告

变更详细信息

☒ Commit directly to the `master` branch.

☐ Create a **new branch** for this commit and start a pull request. [Learn more about pull requests.](#)

确认变更

Commit changes

Cancel

11. 回到项目主界面，向项目主干发出修改请求。（如果你不想修改主干，就不需要点）

This repository

Search

Pull requests

Issues

Marketplace

Gist

+ ▾

icurious / resource_main

forked from CUMTIOTResource/resource_main

Unwatch ▾ 1

★ Star 0

Fork 2

Code

Pull requests 0

Projects 0

Wiki

Settings

Insights ▾

所有资源的汇总页面

Edit

Add topics

24 commits

1 branch

0 releases

2 contributors

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

This branch is 2 commits ahead, 10 commits behind CUMTIOTResource:master.

Pull request

Compare

icurious committed on GitHub 添加了行业分析报告 ...

Latest commit 33c5894 2 minutes ago

README.md

添加了行业分析报告

2 minutes ago

resource_tree.png

Add files via upload

2 days ago

README.md

12. 点击 create pull request

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights ▾

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base fork: CUMTIOTResource/resour...

base: master ▾

head fork: icurious/resource_main ▾

compare: master ▾

✓ Able to merge. These branches can be automatically merged.

Create pull request

Discuss and review the changes in this comparison with others.

2 commits

1 file changed

0 commit comments

1 contributor

Commits on Jun 11, 2017

icurious Merge pull request #1 from CUMTIOTResource/master ... c60151a

Commits on Jun 12, 2017

icurious 添加了行业分析报告 ... 33c5894

Showing 1 changed file with 1 addition and 0 deletions.

Unified

Split

1 README.md

显示和主干的对比

<> View

144 @@ -144,6 +144,7 @@

145

146 ## 报告/博文

147 +- 行业分析报告（链接：http://pan.baidu.com/s/1eSvLmgE 密码：5yf3）

148 ## 网站及博客推荐

149 > 导航站

150 > 导航站

CUMTIOTResource / resource_main

Watch 0 Star 0 Fork 2

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights

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](#).

base fork: CUMTIOTResource/resour... base: master ... head fork: icurious/resource_main compare: master ✓ Able to merge. These branches can be automatically merged.

我添加了一些行业报告

Write Preview

如果觉得不错，实验室就添进去吧

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

☒ Allow edits from maintainers. [Learn more](#)

Create pull request

2 commits 1 file changed 0 commit comments 1 contributor

This repository Search Pull requests Issues Marketplace Gist

CUMTIOTResource / resource_main Watch 0 Star 0 Fork 2

Code Issues 0 Pull requests 1 Projects 0 Wiki Insights

我添加了一些行业报告 #3

Open icurious wants to merge 2 commits into CUMTIOTResource:master from icurious:master

Conversation 0 Commits 2 Files changed 1 +1 -0

icurious commented just now

如果觉得不错，实验室就添进去吧

icurious added some commits a day ago

- Merge pull request #1 from CUMTIOTResource/master c68151a
- 添加了行业分析报告 55c9894

Add more commits by pushing to the master branch on icurious/resource_main.

✓ This branch has no conflicts with the base branch

Only those with write access to this repository can merge pull requests.

Write Preview

Leave a comment

Reviews: No reviews

Assignees: No one assigned

Labels: None yet

Projects: None yet

Milestone: No milestone

Notifications: Unsubscribe

You're receiving notifications because you authored the thread

<https://github.com/icurious>

13. 管理员回应通过，新添加的内容就添加到主干中了。

14. 成功！恭喜你成为新的贡献者。之后的流程一样。

● 如果我想成为本项目的维护者（管理员）

1. 登录账号

The world's leading... GitHub, Inc. (US) | <https://github.com>

输入网址<https://github.com> 或者直接使用搜索引擎搜索GitHub

Sign in Sign up

Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside millions of other developers.

Username: Pick a username

Email: Your email address

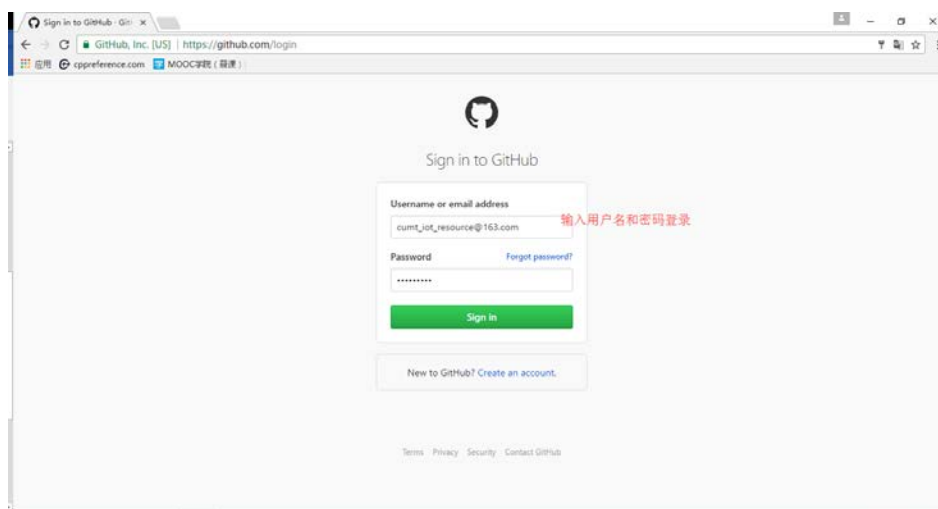
Password: Create a password

Use at least one letter, one numeral, and seven characters.

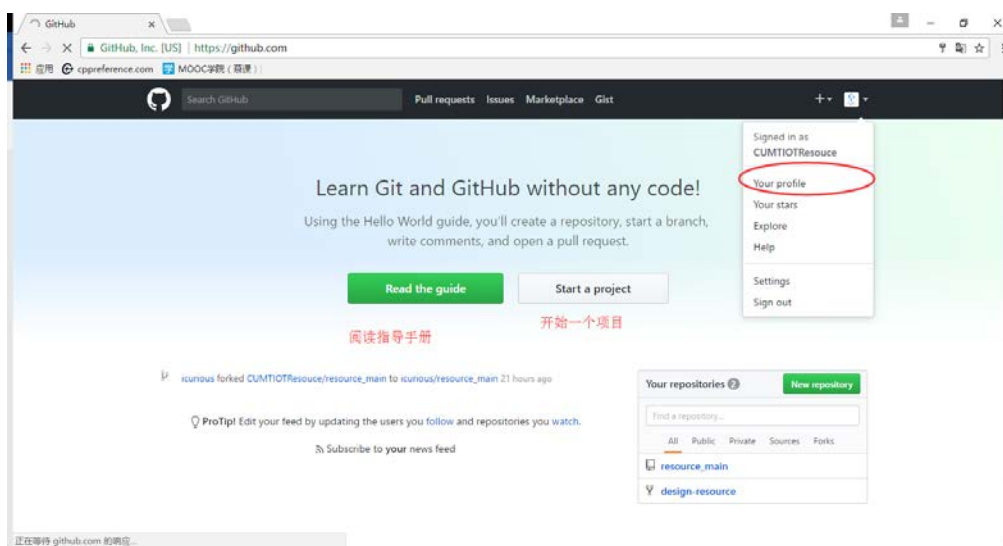
Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy. We'll occasionally send you account related emails.

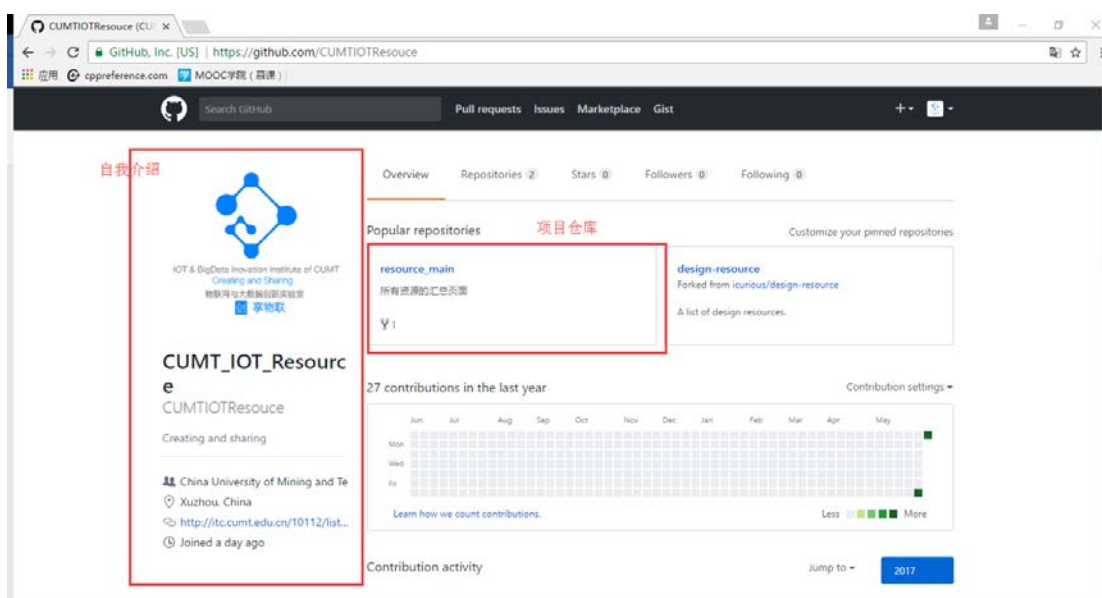
2. 输入用户名和密码



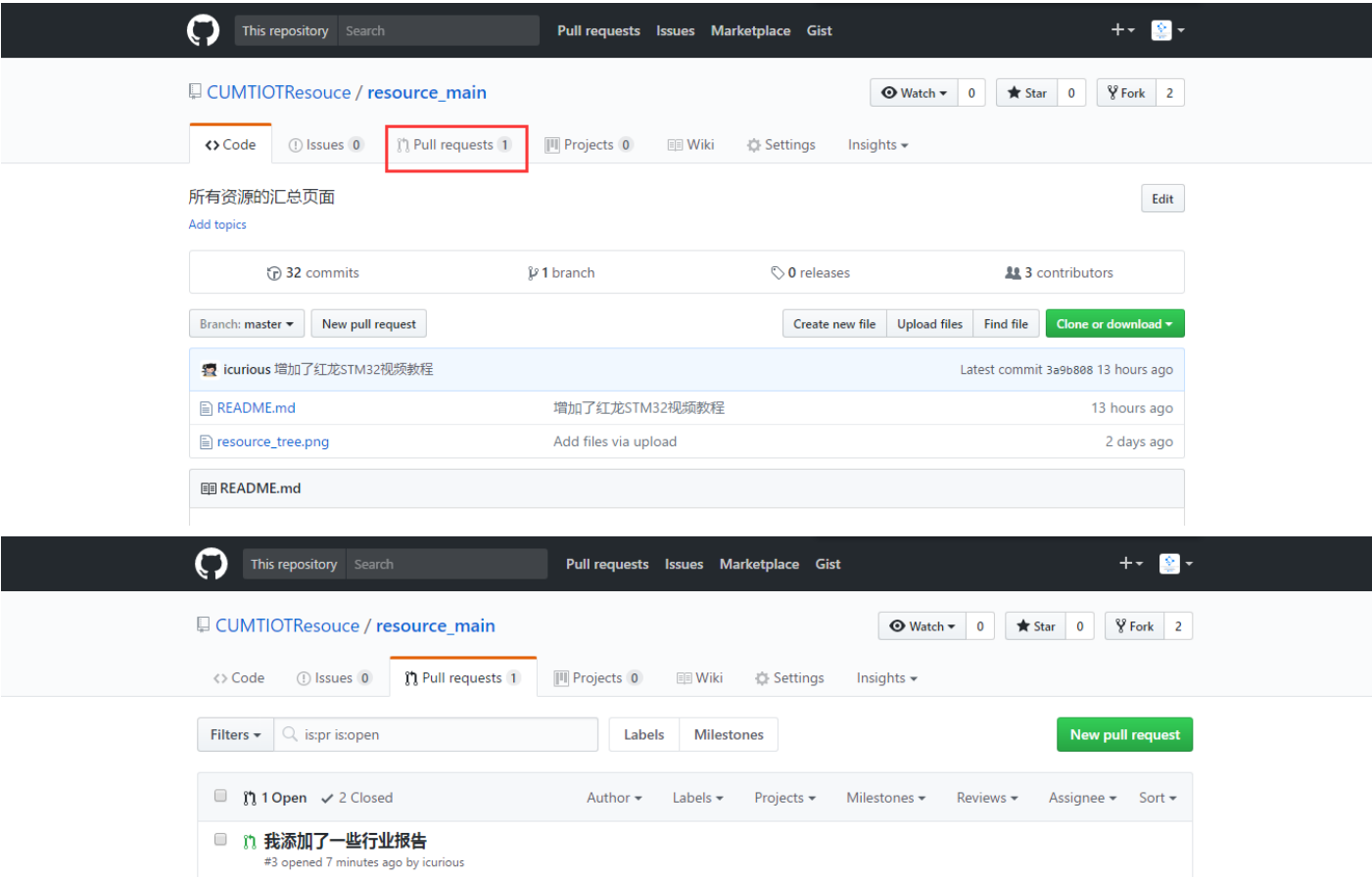
3. 点击 profile



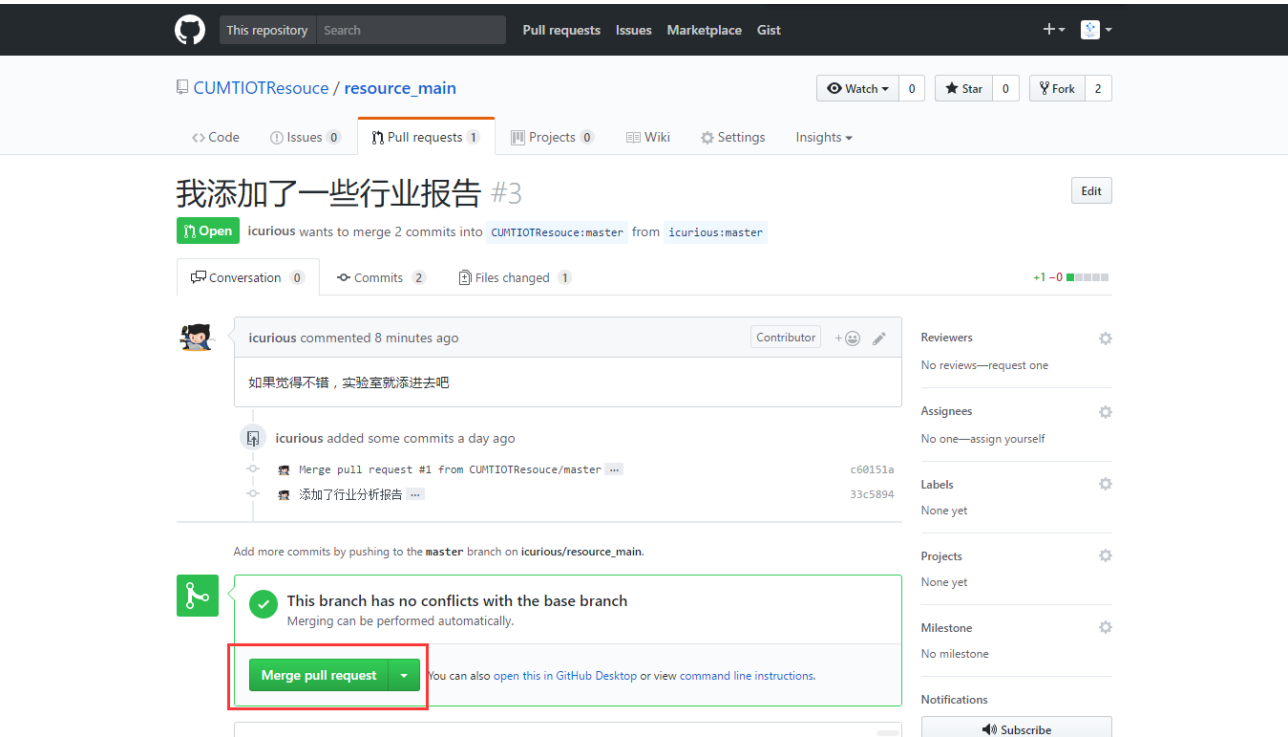
4. 转到要维护的修改的项目



5. 查看有无新的 pull request



6. 对新的 pull request 进行审核，如果觉得合适就合并到主分支中



Navigation: <> Code 0 Issues 0 Pull requests 1 Projects 0 Wiki Settings Insights

我添加了一些行业报告 #3

icurious wants to merge 2 commits into CUMTIOTResource:master from icurious:master

Conversation 0 Commits 2 Files changed 1 +1 -0

icurious commented 9 minutes ago
如果觉得不错，实验室就添进去吧

icurious added some commits a day ago

Merge pull request #1 from CUMTIOTResource/master ... c60151a
添加了行业分析报告 ... 33c5894

Add more commits by pushing to the master branch on icurious/resource_main.

Merge pull request #3 from icurious/master
好的，资源不错

Confirm merge Cancel

Write Preview AA B i " < > : : : @

Leave a comment

1 participant

Watch 0 Star 0 Fork 2

Navigation: <> Code 0 Issues 0 Pull requests 1 Projects 0 Wiki Settings Insights

我添加了一些行业报告 #3

CUMTIOTResource merged 2 commits into CUMTIOTResource:master from icurious:master just now

Conversation 0 Commits 2 Files changed 1 +1 -0

icurious commented 10 minutes ago
如果觉得不错，实验室就添进去吧

icurious added some commits a day ago

Merge pull request #1 from CUMTIOTResource/master ... c60151a
添加了行业分析报告 ... 33c5894

CUMTIOTResource merged commit 2ea9be8 into CUMTIOTResource:master just now

Revert

7. 合并成功 !!!

图书

- 计算机专业书籍合集1(计算机基础相关) (链接：<http://pan.baidu.com/s/1o7GTsee> 密码：69kl)
- 计算机专业书籍合集2(计算机视觉相关)链接：<http://pan.baidu.com/s/1qYljZA> 密码：kf7q
- 物联网书籍合集 (链接：<http://pan.baidu.com/s/1c15lw5E> 密码：goks)
- 《算法艺术和信息学科竞赛》(链接：<http://pan.baidu.com/s/1qY0k1aO> 密码：4vyg)

报告/博文

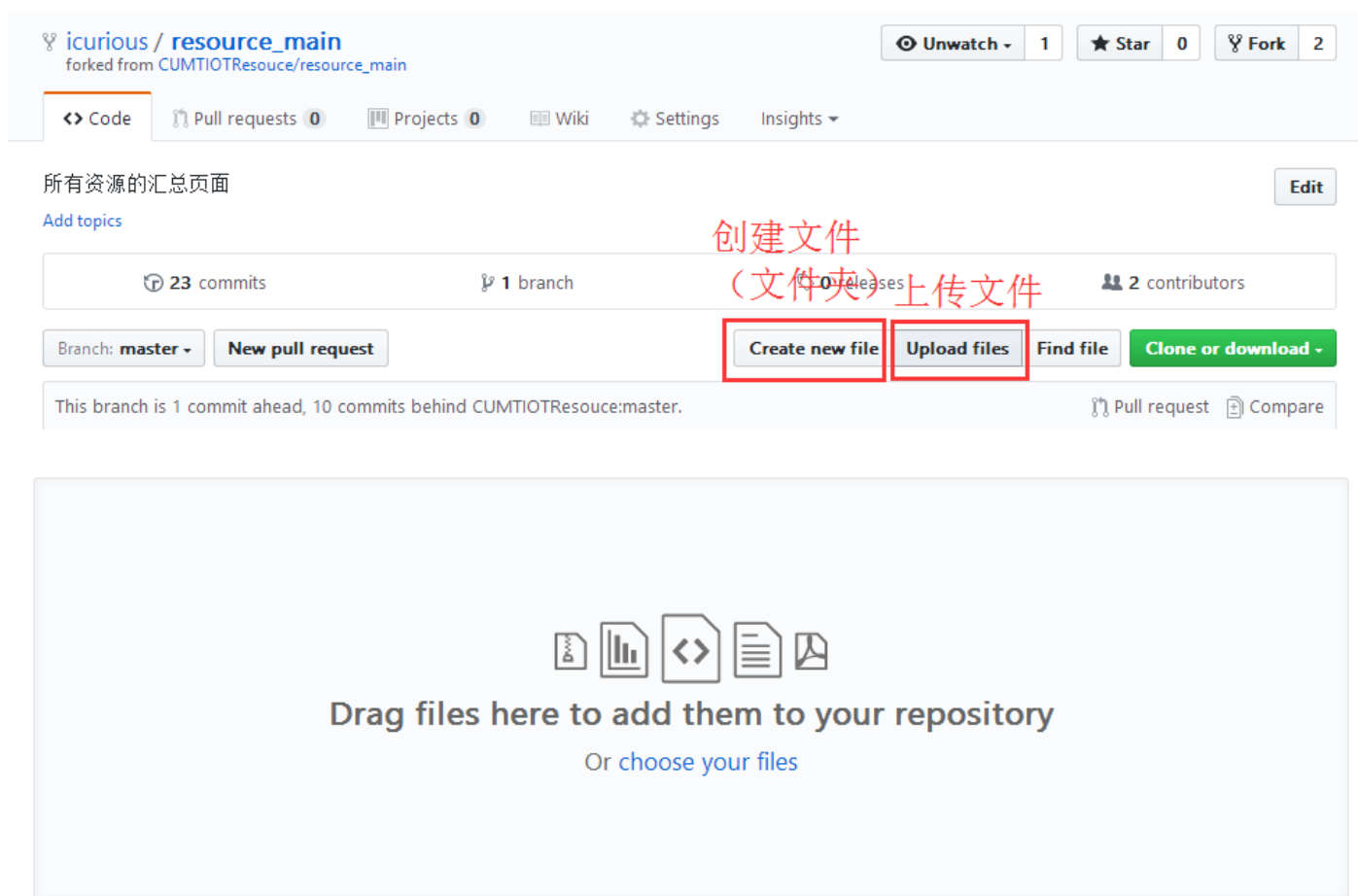
- 行业分析报告 (链接：<http://pan.baidu.com/s/1eSvLmgE> 密码：5yf3)

● 如果我什么都不想，只想查看和下载上面的资源

https://github.com/CUMTIOTResource/resource_main 所有人都是可以访问，直接点 readme 上的链接就可以了。

- 我想修改（增删改）工程中的文件中该如何操作？

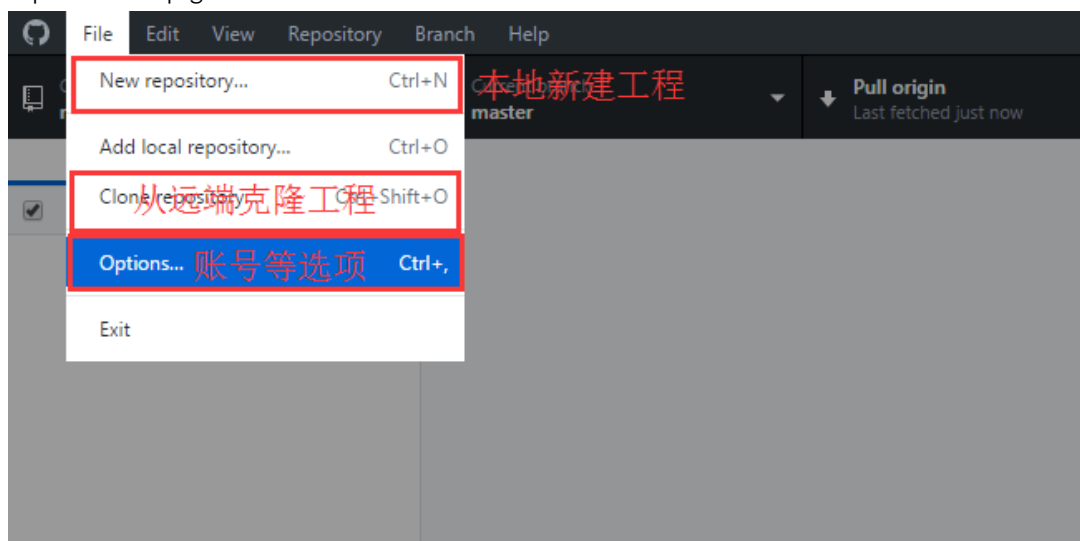
方式一 直接在网站上添加（难度系数 0）

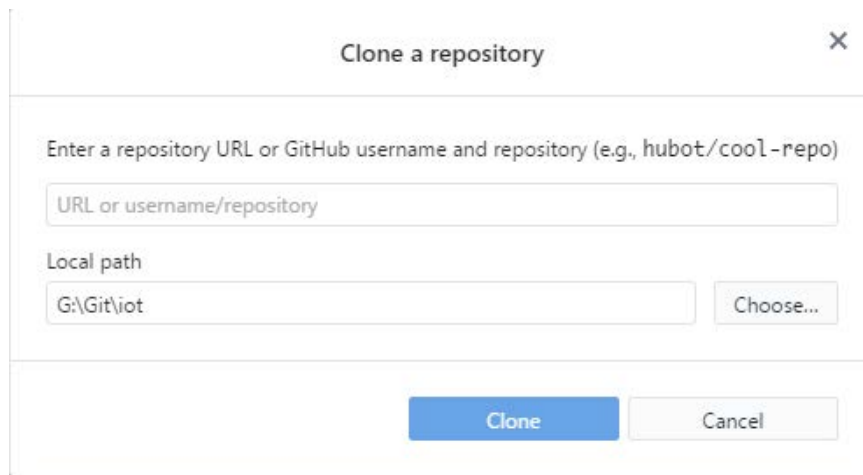


拖拽文件或者选择文件就行。

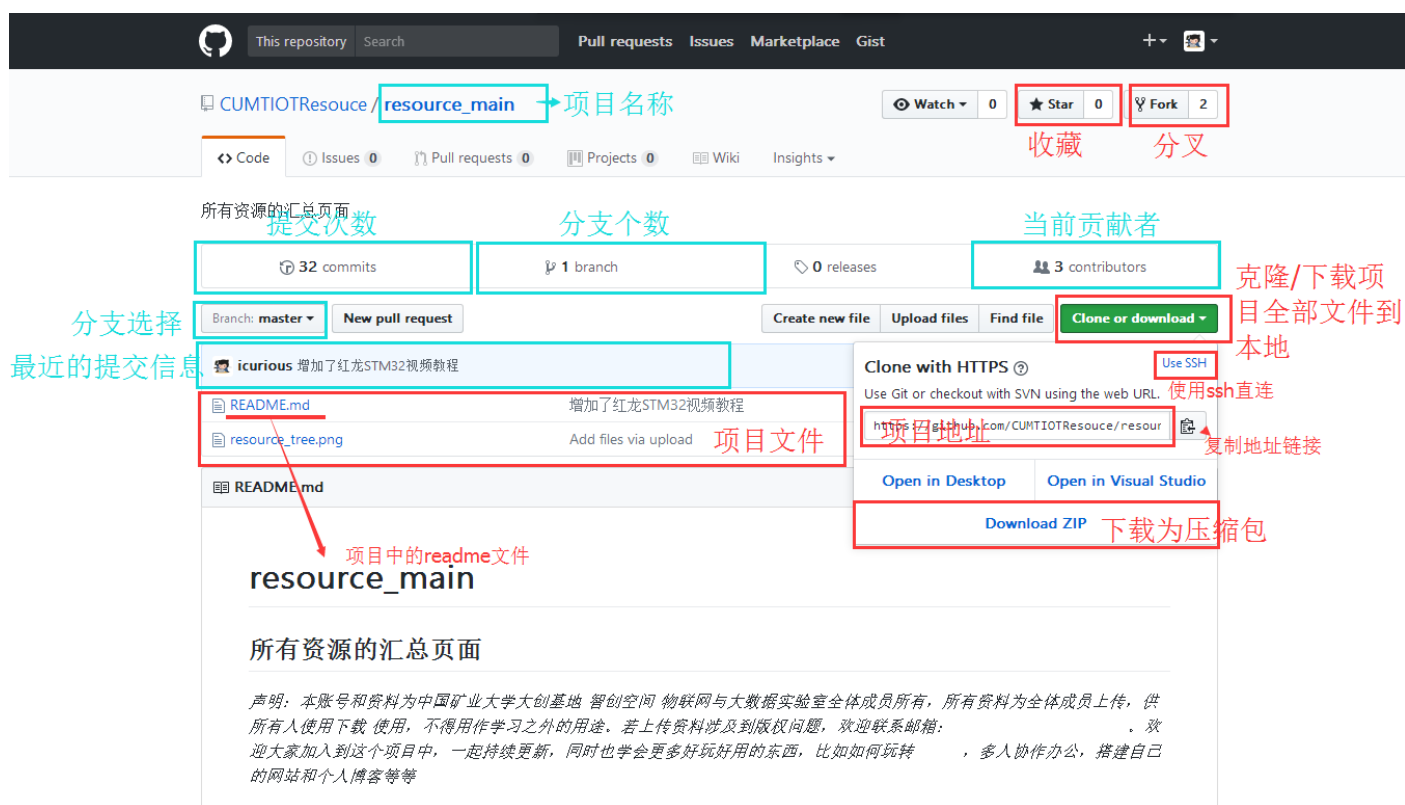
方式二 使用 Github 客户端(难度系数 1)

客户端下载 <https://desktop.github.com/>

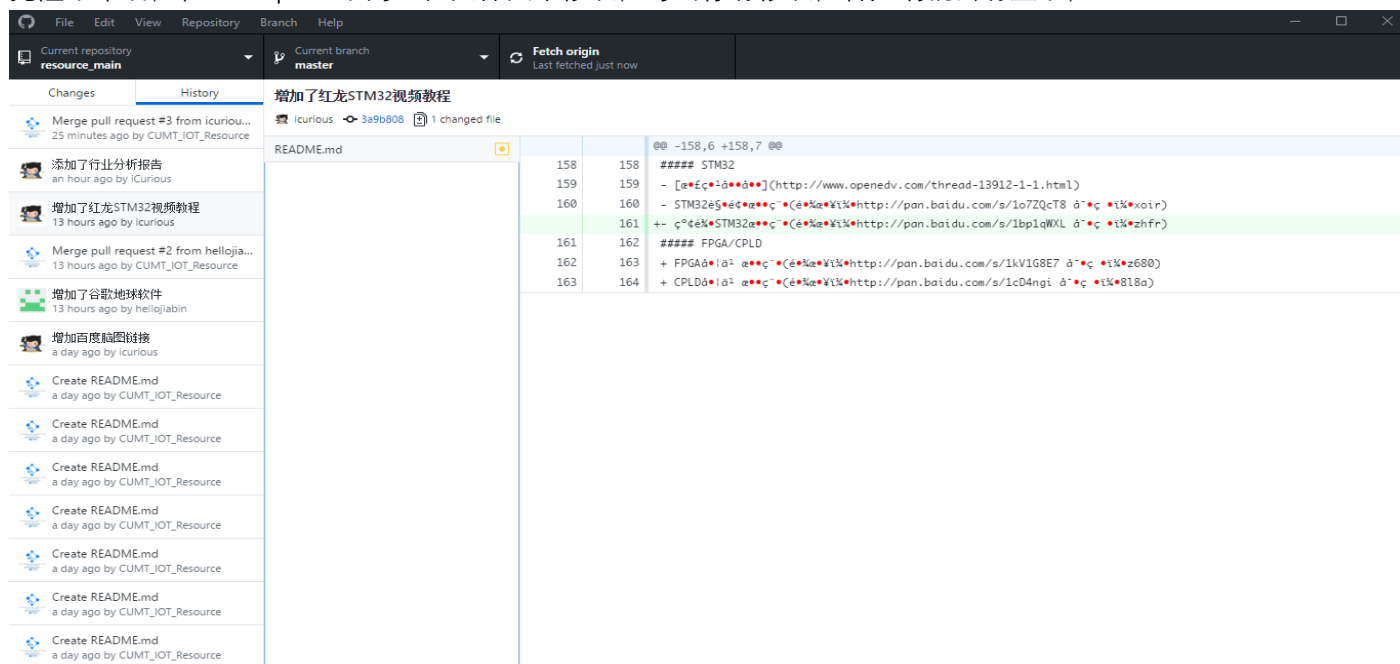


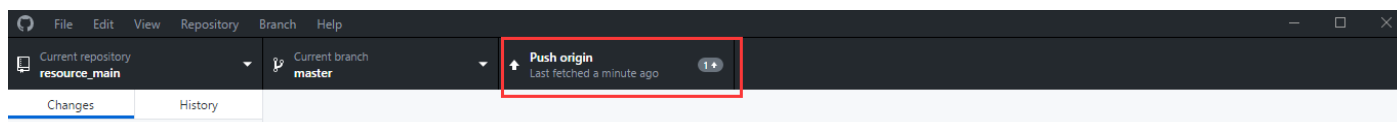
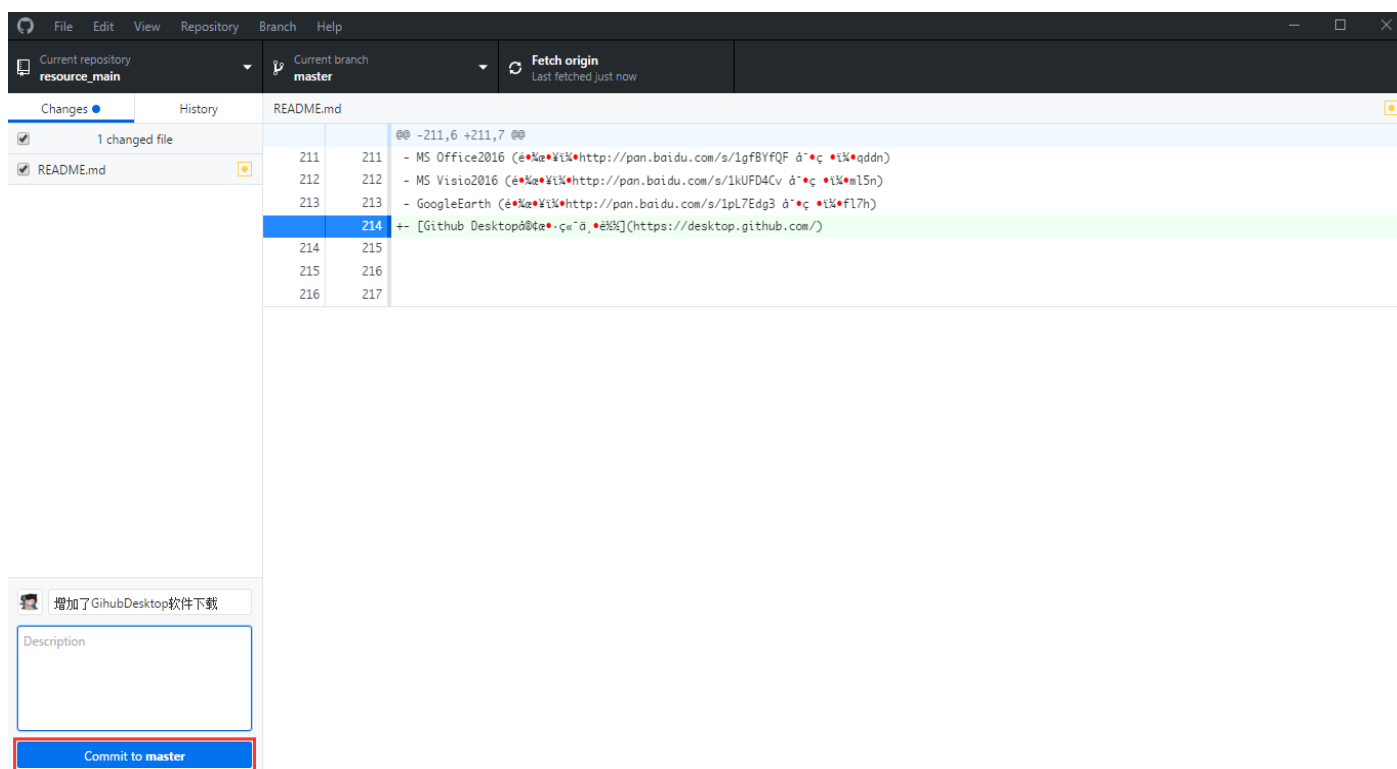


链接的获取在下图中：



克隆下来后，在 local path 下的工程文件夹中修改，每当你有修改，客户端就会有显示，

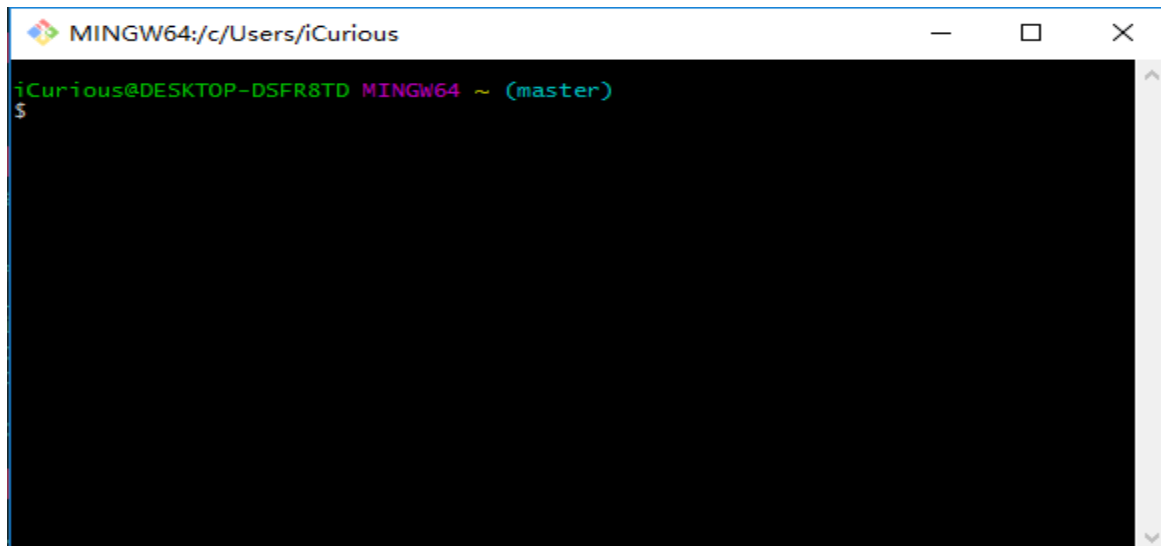




上传成功。

方式二 使用 Git bush (难度系数 3)

0 下载 gitbush <https://git-for-windows.github.io/>



1.在 GitHub 上建立项目

登录 GitHub 后，你可以在右边靠中那里找到一个按钮“New Repository”，点击过后，填入项目名称、说明和网址过后就可以创建了，然后会出现一个提示页面，记下类似 `git@github.com:XXX/XXX.git` 的地址，这个就是你这个项目的地址了。

2.配置 Git 以及上传代码

安装 Git 成功后，如果是 Windows 下，选择 Git Bash，在命令行中完成一切，可能开始有点麻烦，不过就那几条命令行，用几次就记住啦。首先初始设置 Git：

```
1 git config --global user.name "Your Real Name"
```

```
2 git config --global user.email you@email.address
```

然后开始进行最麻烦的一步了，你需要上传文件到 GitHub 的 Git 系统上，得需要一个 SSH 密匙来认证，下面就开始生成密钥和提交密钥。打开 Git Bash,创建 SSH key:

```
1 ssh-keygen -C 'your@email.address' -t rsa
```

然后要你输入 SSH 密匙的存放位置，可以不管，直接回车使用默认路径。再输入你想要的密码，SSH key 就生成了。现在你需要将这个 Key 提交到 GitHub，首先打开 Key 保存的位置，里面会有三个文件，找到 `id_rsa.pub`，用文本编辑器打开，复制里面的全部字符。到 GitHub，在右上方工具栏里找到 Account Settings。在这个页面上有一个 SSH Public Keys 标签，选择 Add another public key。Title 可以随便填一个，Key 就粘贴刚才的字符，提交。

完成这些工作后，就可以上传自己的代码了。找到自己要分享上传的代码文件夹，右击选择 Git Bash，或者在 Git Bash 中进入这个文件夹。建立一个仓库：

```
1 git init
```

选择要添加进仓库的文件：

```
1 git add .
```

一般如果你想分享这个文件夹里的所有代码，就在 `add` 后面加“.”，上面的例子就是这样，如果传指定的，只需要把“.”改为文件名即可，现在只是选择了要加入仓库的文件，下面才是添加进入仓库：

```
1 git commit -m 'Test'
```

-m 后面跟一个参数，表示说明，将代码提交到 GitHub 后，将会在代码文件信息上显示这个说明，如下图标记的地方。

搞了这么久，现在才开始把本地仓库上传到 GitHub 了，下面两行命令搞定问题：

首先需要 pull 下代码，才能 push 上传成功

(`git pull origin master`)

继续下面的操作

```
1 git remote add origin git@github.com:XXX/XXX.git
```

```
2 git pull origin master
```

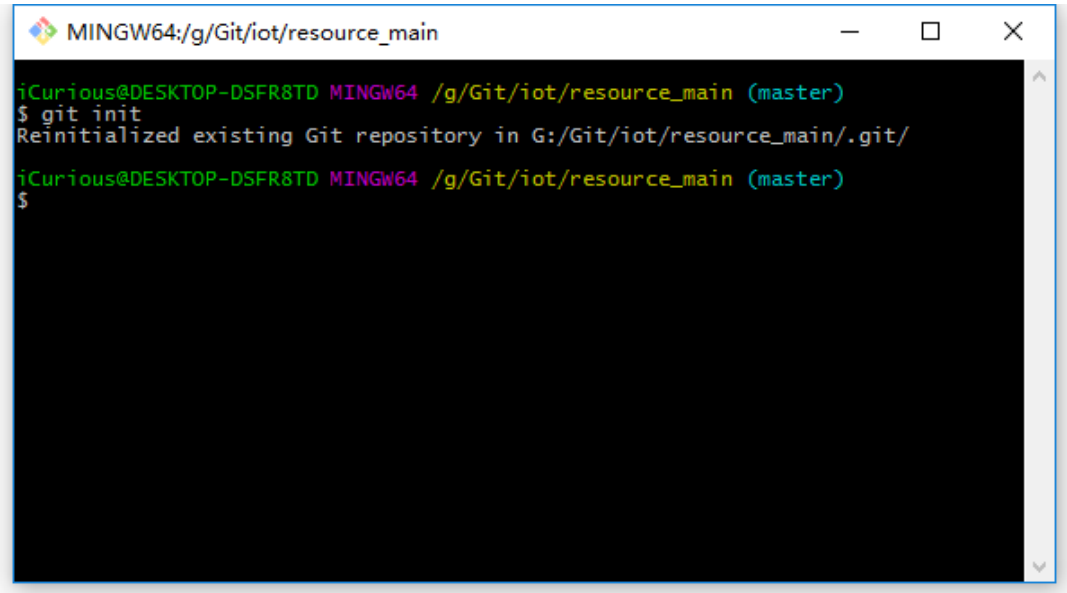
```
3 git push -u origin master
```

这个 `git@github.com:XXX/XXX.git` 就是上面创建项目是生成的地址。现在打开你的项目网址，你就可以发现你的代码已经展示出来了。如果你要更新代码的话，就重复上面的吧。

例如：



输入 git init



输入 `git add -A`(表示添加所有, 还有其他参数和选项)

```
MINGW64:/g/Git/iot/resource_main

iCurious@DESKTOP-DSFR8TD MINGW64 /g/Git/iot/resource_main (master)
$ git init
Reinitialized existing Git repository in G:/Git/iot/resource_main/.git/

iCurious@DESKTOP-DSFR8TD MINGW64 /g/Git/iot/resource_main (master)
$ git add -A

iCurious@DESKTOP-DSFR8TD MINGW64 /g/Git/iot/resource_main (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   README.md

iCurious@DESKTOP-DSFR8TD MINGW64 /g/Git/iot/resource_main (master)
$ git commit -m"update"
[master b2af7d2] update
1 file changed, 2 insertions(+), 1 deletion(-)
```

输入 `git push https://github.com/xxx/yyy.git master`

```
MINGW64:/g/Git/iot/resource_main

iCurious@DESKTOP-DSFR8TD MINGW64 /g/Git/iot/resource_main (master)
$ git init
Reinitialized existing Git repository in G:/Git/iot/resource_main/.git/

iCurious@DESKTOP-DSFR8TD MINGW64 /g/Git/iot/resource_main (master)
$ git add -A

iCurious@DESKTOP-DSFR8TD MINGW64 /g/Git/iot/resource_main (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   README.md

iCurious@DESKTOP-DSFR8TD MINGW64 /g/Git/iot/resource_main (master)
$ git commit -m"update"
[master b2af7d2] update
1 file changed, 2 insertions(+), 1 deletion(-)

iCurious@DESKTOP-DSFR8TD MINGW64 /g/Git/iot/resource_main (master)
$ git push https://github.com/icurious/resource_main.git master
Counting objects: 37, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (37/37), done.
Writing objects: 100% (37/37), 4.68 KiB | 0 bytes/s, done.
Total 37 (delta 12), reused 0 (delta 0)
remote: Resolving deltas: 100% (12/12), completed with 1 local object.
To https://github.com/icurious/resource_main.git
   33c5894..b2af7d2  master -> master

iCurious@DESKTOP-DSFR8TD MINGW64 /g/Git/iot/resource_main (master)
$ |
```

上传成功。

更多操作和使用技巧, 输入 `git --help`

```

iCurious@OESKTOP-DSFR8TD MINGW64 ~ (master)
$ git --help
usage: git [--version] [--help] [-C <path>] [-c name=value]
       [--exec-path=<path>] [--html-path] [--man-path] [--info-path]
       [-p | --paginate] [--no-pager] [--no-replace-objects] [--bare]
       [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
       <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  reset      Reset current HEAD to the specified state
  rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect     Use binary search to find the commit that introduced a bug
  grep       Print lines matching a pattern
  log        Show commit logs
  show       Show various types of objects
  status     Show the working tree status

grow, mark and tweak your common history
  branch     List, create, or delete branches
  checkout   Switch branches or restore working tree files
  commit     Record changes to the repository
  diff       Show changes between commits, commit and working tree, etc
  merge      Join two or more development histories together
  rebase     Reapply commits on top of another base tip
  tag        Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
  fetch      Download objects and refs from another repository
  pull       Fetch from and integrate with another repository or a local branch
  push       Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.

```

或者看下面这张图：

Git 常用命令速查表

master : 默认开发分支 Head : 默认开发分支
origin : 默认远程版本库 Head^ : Head 的父提交

创建版本库

```

$ git clone <url>          #克隆远程版本库
$ git init                 #初始化本地版本库

```

修改和提交

```

$ git status              #查看状态
$ git diff                #查看变更内容
$ git add .               #跟踪所有改动过的文件
$ git add <file>          #跟踪指定的文件
$ git mv <old> <new>      #文件改名
$ git rm <file>           #删除文件
$ git rm --cached <file>  #停止跟踪文件但不删除
$ git commit -m "commit message"
                           #提交所有更新过的文件
$ git commit --amend      #修改最后一次提交

```

查看提交历史

```

$ git log                 #查看提交历史
$ git log -p <file>       #查看指定文件的提交历史
$ git blame <file>        #以列表方式查看指定文件的提交历史

```

撤销

```

$ git reset --hard HEAD   #撤销工作目录中所有未提交文件的修改内容
$ git checkout HEAD <file> #撤销指定的未提交文件的修改内容
$ git revert <commit>     #撤销指定的提交

```

分支与标签

```

$ git branch              #显示所有本地分支
$ git checkout <branch/tag> #切换到指定分支或标签
$ git branch <new-branch> #创建新分支
$ git branch -d <branch>   #删除本地分支
$ git tag                 #列出所有本地标签
$ git tag <tagname>        #基于最新提交创建标签
$ git tag -d <tagname>     #删除标签

```

合并与衍合

```

$ git merge <branch>      #合并指定分支到当前分支
$ git rebase <branch>     #衍合指定分支到当前分支

```

远程操作

```

$ git remote -v           #查看远程版本库信息
$ git remote show <remote> #查看指定远程版本库信息
$ git remote add <remote> <url> #添加远程版本库
$ git fetch <remote>      #从远程库获取代码
$ git pull <remote> <branch> #下载代码及快速合并
$ git push <remote> <branch> #上传代码及快速合并
$ git push <remote> :<branch/tag-name> #删除远程分支或标签
$ git push --tags         #上传所有标签

```

● 学了这个有什么用？

学会了这些，你就能：

- 多人协作办公，团队合作，共同完成一件工程
- 成为一名开源者，和众多开源者一起分享自己的经验、兴趣、见解、代码
- 搭建自己的博客、网站
- 用它来写一本书
- 向 HR 和未来的 BOSS 展示自己
-

● Readme.md 是用什么写成的？有什么格式要求？

其是用 markdown 书写的，这里顺便介绍一下。

标题

在文字书写不同数量的#可以完成不同的标题，如下：

一级标题

二级标题

三级标题

四级标题

五级标题

六级标题

等号及减号也可以进行标题的书写，不过只能书写二级标题，并且需要写在文字的下面，减号及等号的数量不会影响标题的基数，如下：

二级标题

=====

二级标题

列表

无序列表的使用，在符号 “-” 后加空格使用。如下：

- 无序列表 1
- 无序列表 2
- 无序列表 3

如果要控制列表的层级，则需要在符号 “-” 前使用空格。如下：

- 无序列表 1
- 无序列表 2
 - 无序列表 2.1
 - 列表内容
 - 列表内容

有序列表的使用，在数字及符号 “.” 后加空格几个，如下：

1. 有序列表 1
2. 有序列表 2
3. 有序列表 3

有序列表如果要区分层级，也可以在数字前加空格。

引用

引用的格式是使用符号 “>” 后面书写文字，及可以使用引用。如下：

- > 这个是引用
- > 是不是和电子邮件中的
- > 引用格式很像

粗体和斜体

粗体的使用是在需要加粗的文字前后各加两个 “**”，而斜体的使用则是在需要斜体的文字前后各加一个 “*””，如果要使用粗体和斜体，那么就是在需要操作的文字前后各加三个 “***”。如下：

- **这个是粗体**
- *这个是斜体*
- ***这个是粗体加斜体***

图片和链接

在文中直接加链接，中括号中是需要添加链接的文字，圆括号中是需要添加的链接，如下：

[link text](http://example.com/ "optional title")

在引用中加链接，第一个中括号添加需要添加的文字，第二个中括号中是引用链接的 id，之后在引用中，使用 id 加链接：如下：

[link text][id]
[id]: http://example.com/ "optional title here"

在文中直接引用链接，直接使用尖括号，把链接加入到尖括号中就可以实现，如下：

<<http://example.com/>> or <address@example.com>

插入互联网上图片，格式如下：

![这里写图片描述](<http://img3.douban.com/mpic/s1108264.jpg>)
![这里写图片描述][jane-eyre-douban]
[jane-eyre-douban]: <http://img3.douban.com/mpic/s1108264.jpg>

代码块和语法高亮

在需要高亮的代码块的前一行及后一行使用三个反引号 “`” ，同时第一行反引号后面表面代码块所使用的语言，如下：

```
` `ruby
require 'redcarpet'
markdown = Redcarpet.new("Hello World!")
puts markdown.to_html
` `
```

分割线

可以在一行中用三个以上的星号、减号、底线来建立一个分隔线，同时需要在分隔线的上面空一行。如下：

```
---
****
---
```

删除线的使用，在需要删除的文字前后各使用两个符合 “~” ，如下

~~Mistaken text.~~

表格

可以使用冒号来定义表格的对齐方式，如下：

Tables Are Cool
----- :-----: -----:
col 3 is right-aligned \$1600
col 2 is centered \$12
zebra stripes are neat \$1

Latex 公式

使用 MathJax 渲染*LaTeX* 数学公式，详见[\[math.stackexchange.com\]](http://math.stackexchange.com)

行内公式，数学公式为：

$\Gamma(n) = (n-1)! \quad \text{for all } n \in \mathbb{N}$

块级公式：

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

更多 LaTeX 语法请参考 [这儿](#)。

UML 图

可以渲染序列图：

```
sequence
张三->李四: 嘿，小四儿，写博客了没？
Note right of 李四: 李四愣了一下，说：
李四-->张三: 忙得吐血，哪有时间写。

```

或者流程图：

```
flow
st=>start: 开始
e=>end: 结束
op=>operation: 我的操作
cond=>condition: 确认？

st->op->cond
cond(yes)->e
cond(no)->op

```

- 关于 **序列图** 语法，参考 [这儿](#)
- 关于 **流程图** 语法，参考 [这儿](#)