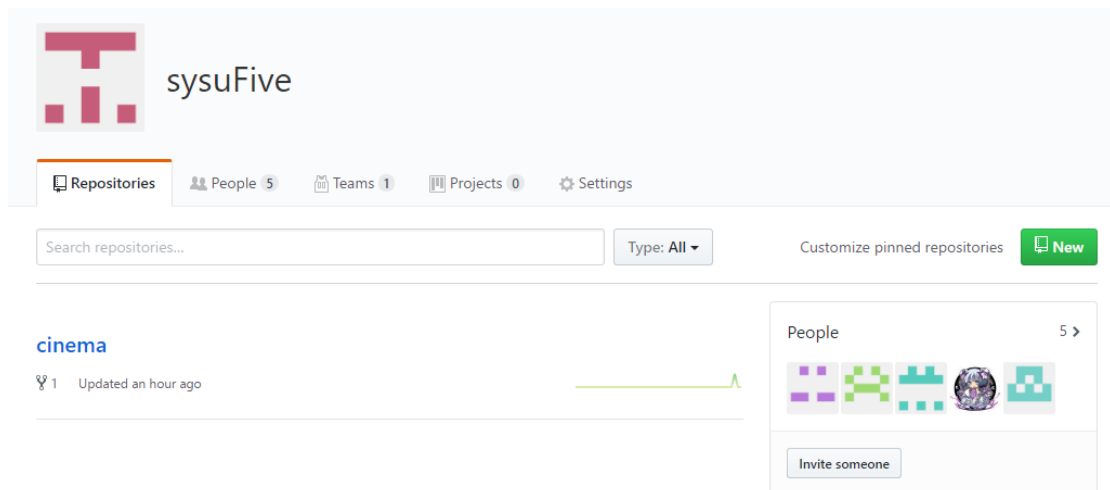


# git 和 BUG 管理系统

安装和学习报告：

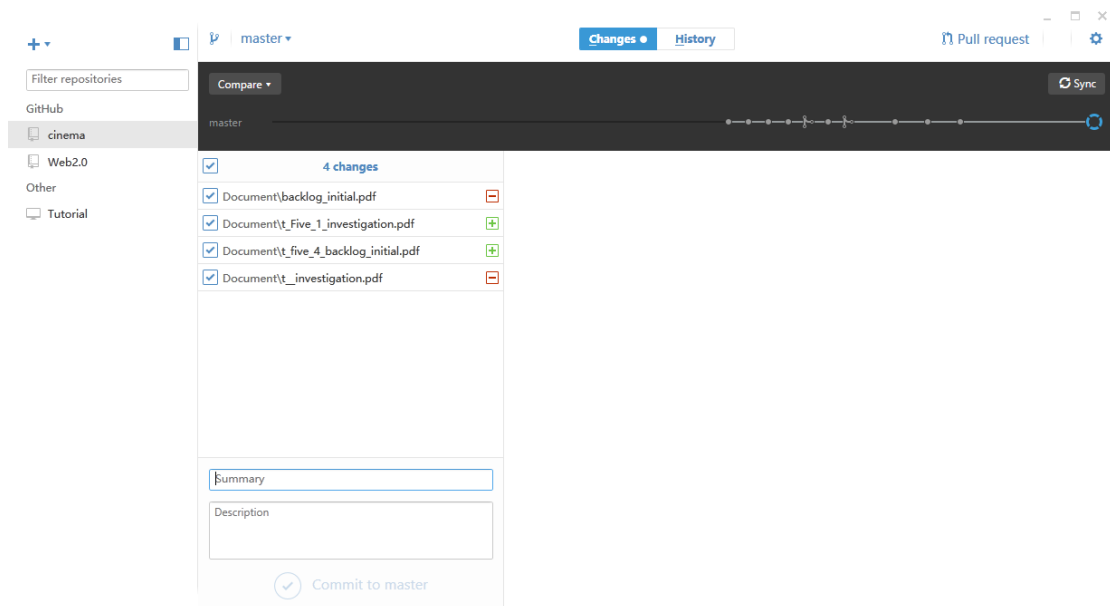
- 1.在 Github 上注册一个账号
- 2.创建小组组织：



- 3.在组织中新建一个 Repository

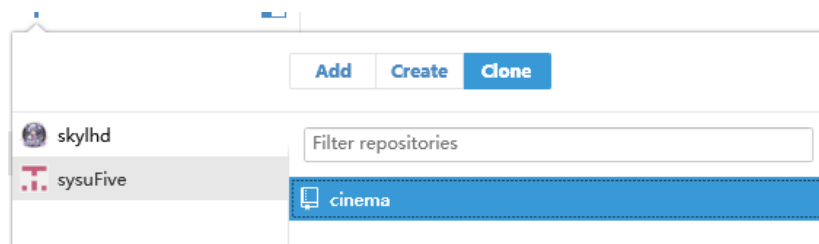


4. 下载 GitHub Desktop 并登录自己账号：



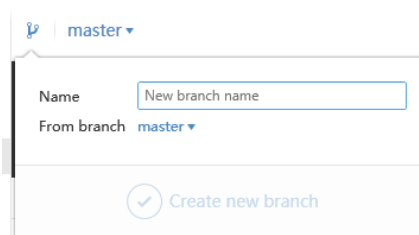
这里可以看到自己 GitHub 上面的所有内容。

## 5. clone 项目：



选择组织的项目，克隆到本地。

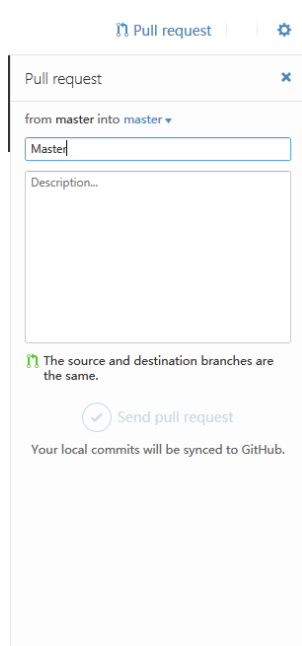
## 6. 创建新的分支：



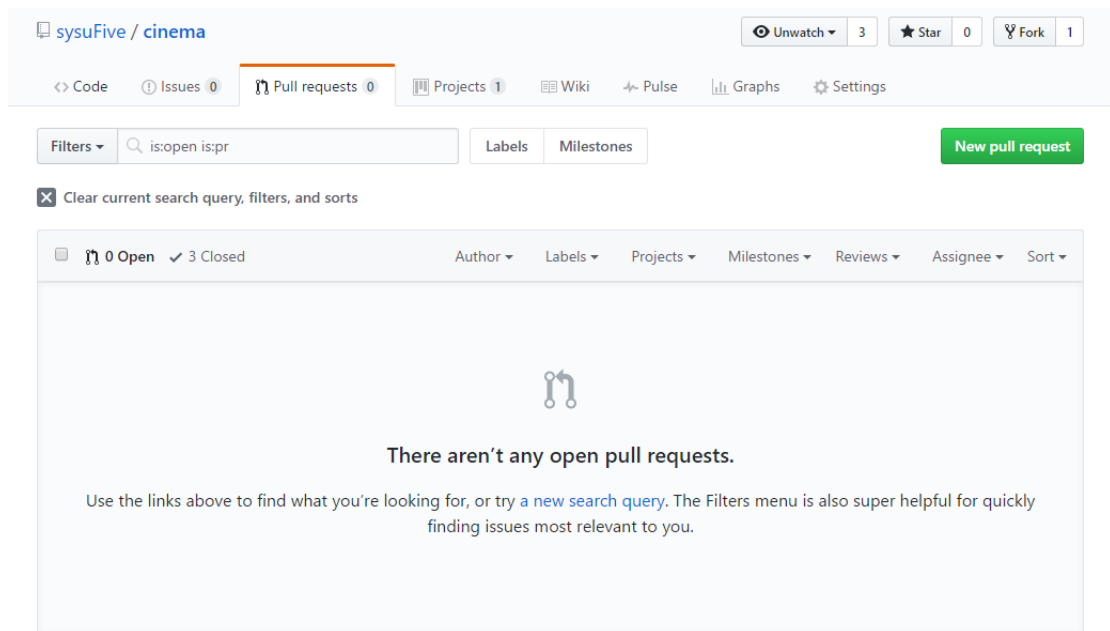
当要进行模块的开发或者修改时，新建一个分支，然后在新的分支上进行开发和修改。

## 7. Pull Request 和 Merge

当我们完成模块的开发和修改之后，通过发出 Pull Request 来发出请求：



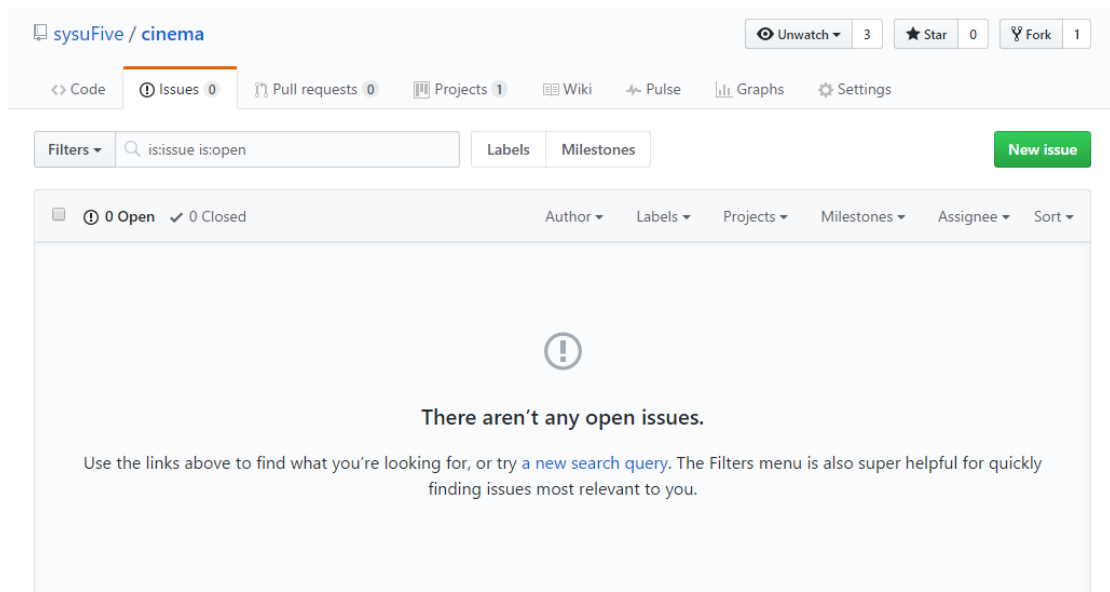
当一个 Pull Request 发出之后，我们可以在网页上看到：




如果有新的 Pull Request 时，这里会有对新的 Pull Request 处理的选项，这时可以选择 Merge Branch，把子分支与 Master 分支合并，这样就完成一个模块的开发或修改了。

## 8. Bug 管理

在项目主页点击 Issues 进入管理页：



通过 New Issue 可以创建新的 Issue:



There is a bug

Write

Preview

AA ▾ B i “ < > ↻ ⋮ ⋮ ⋮ ↶ @ 📌

The description of the bug

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

MD Styling with Markdown is supported

Submit new issue


创建的 Issue 可以在主页面看到:

|                          |          |                |                              |          |            |              |            |        |
|--------------------------|----------|----------------|------------------------------|----------|------------|--------------|------------|--------|
| <input type="checkbox"/> | 🔔 1 Open | ✓ 1 Closed     | Author ▾                     | Labels ▾ | Projects ▾ | Milestones ▾ | Assignee ▾ | Sort ▾ |
| <input type="checkbox"/> | 🔔        | There is a bug | #6 opened just now by skylhd |          |            |              |            |        |

点击进去之后可以留言，也可以 Close Issue，也就意味解决了这个 Bug 之后就可以把这个 Issue 关闭了:

## There is a bug #6


🔔 Open skylhd opened this issue 27 seconds ago · 0 comments



skylhd commented 27 seconds ago

Member + 😊 ✎

The description of the bug



Write

Preview

AA ▾ B i “ < > ↻ ⋮ ⋮ ⋮ ↶ @ 📌

Leave a comment

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

MD Styling with Markdown is supported

Close issue

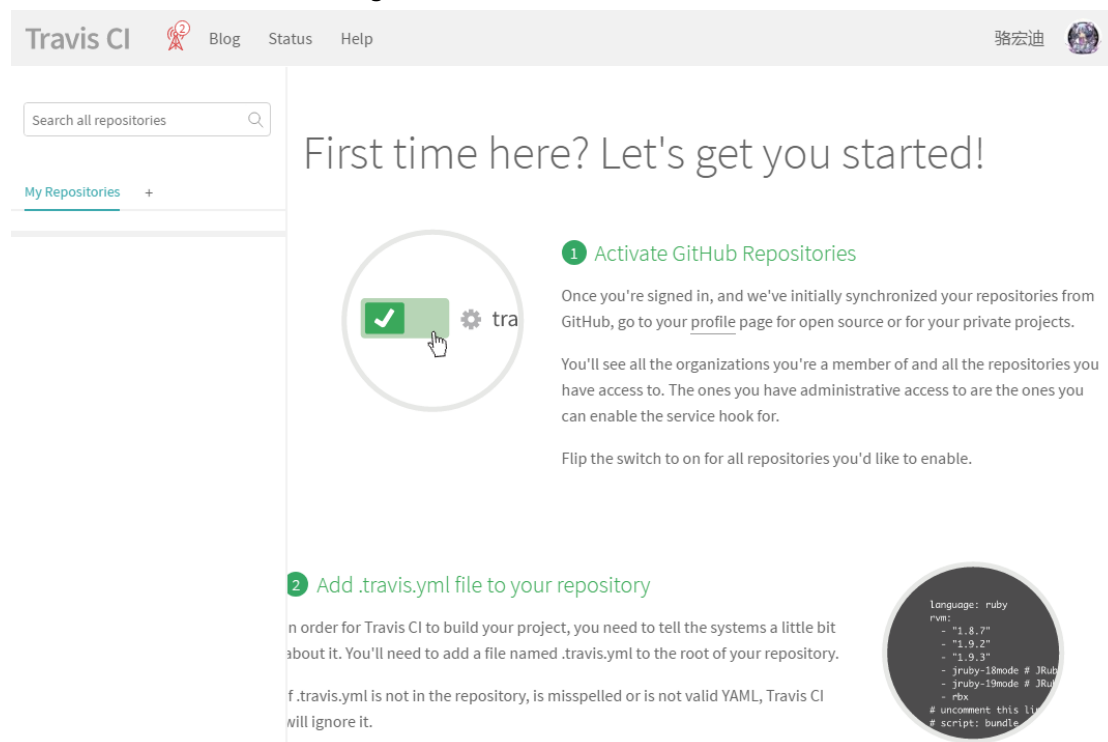
Comment

# 持续集成环境

我使用的是 Travis CI

安装和学习报告：

1.进入 Travis CI 官网，并登录 github 账号



The image shows the Travis CI 'First time here? Let's get you started!' guide. It includes a search bar for repositories, a 'My Repositories' section, and two main steps:

- 1 Activate GitHub Repositories**  
Once you're signed in, and we've initially synchronized your repositories from GitHub, go to your [profile](#) page for open source or for your private projects.  
You'll see all the organizations you're a member of and all the repositories you have access to. The ones you have administrative access to are the ones you can enable the service hook for.  
Flip the switch to on for all repositories you'd like to enable.
- 2 Add .travis.yml file to your repository**  
In order for Travis CI to build your project, you need to tell the systems a little bit about it. You'll need to add a file named `.travis.yml` to the root of your repository.  
If `.travis.yml` is not in the repository, is misspelled or is not valid YAML, Travis CI will ignore it.

A circular inset shows a sample `.travis.yml` file:

```
language: ruby
rvm:
  - "1.8.7"
  - "1.9.2"
  - "1.9.3"
  - jruby-18mode # JRuby
  - jruby-19mode # JRuby
  - rbx
# uncomment this line to run the script: bundle
```

2.进入个人中心，在组织中找到自己的小组，看到 **cinemaServer** 项目，点击项目左边的按钮，变成绿色：



The image shows the Travis CI user profile and repository management interface. The user is 'sysuFive' with 4 repositories. The interface includes a 'Sync account' button and a message: 'We're only showing your public repositories. You can find your private projects on [travis-ci.com](#).'

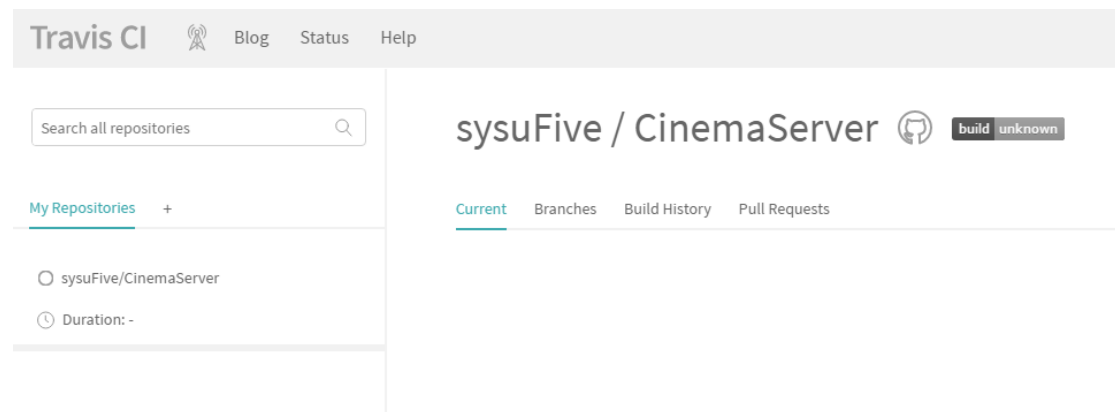
The 'Organizations' section shows 'sysuFive' with 3 repositories. Below this, there is a message: 'Is an organization missing? [Review and add your authorized organizations.](#)'

The 'Repositories' section shows a list of repositories:

- ☐ sysuFive/cinema
- ☐ sysuFive/CinemaClient
- ☒ sysuFive/CinemaServer

The 'Repositories' section also includes a 'See what's new!' button and a 'Beta Features' section.

3.回到 Travis CI 的主页，就可以看到我们的项目出现在列表中：



首先我们更改项目的一些设置：

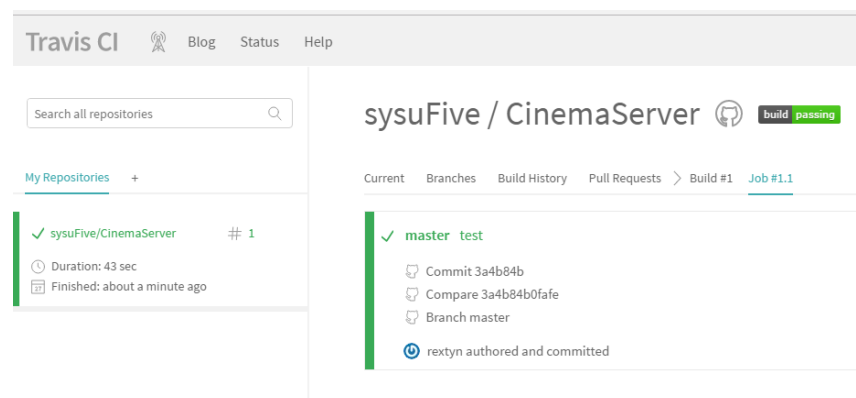


将当且仅当有 `.travis.yml` 文件时才进行 Build 的选项勾上。

然后我们来编写 `.travis.yml` 文件，我们服务器端使用的是 Java 的 Spring + Maven 的架构：

```
1 language: java
2
3 branches:
4   only:
5     - master
6
7 script:
8   - mvn test
```

编写好文件之后，进行 PullRequest 和 Merge，然后我们来查看 TravisCI 的结果：



已经通过了持续集成环境的检查。