Part IV: Continuous Integration

Introduction to continuous integration

What is continuous integration? It means the practice of merging the working copies of all developers into the shared mainline several times a day. This kind of software testing technique is proposed by Grady Booch. It has four important components.

- Firstly, continuous integration is originated from extreme programming development process. Due to high amount of development, there would be lots of changes of code and different progress align with different colleagues. It is super important under such extreme programming environment
- Secondly, continuous integration needs to be performed, even for minor changes. Therefore, every developer are in the same pace for the project.
- Thirdly, every developer in the group and in the project needs to commit their changes every day. Therefore, every developer are in the same pace for the project.
- Forthly, every version, especially the latest version, needs to build and pass all the tests. Otherwise, this program cannot work, and need to check its methods and functions.

Importance of continuous integration

After talking about the basic concepts of continuous integration, we come to the question why we need continuous integration. Combined with its four important components, we concluded five reasons of necessity of continuous integration.

- Firstly, combined with all of the four basic concepts, it will be quicker to predict the total development time for this project.
- Secondly, due to the second, third, forth basic concepts, it's easier to detect bugs.
    - Because developer need to commit their changes even for minor changes, it is way easier to detect bugs. We can *strangled the baby bugs in the cradle*
    - Also, bugs can be detected separately because each change is separated. It's easy to trace each commit and each bug.
    - Here, we can use differential debugging. It can help by comparing known good codes with faulty codes.

Using continuous integration

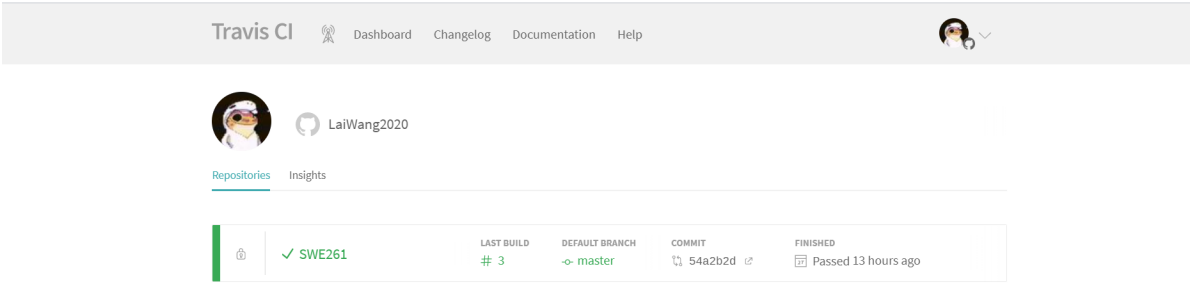
Talking about so many advantages of continuous integration, next step, we want to combine continuous integration with our `JSoup` project.

With the help of TravisCI

First of all, we need to sign up for TravisCI. TravisCI is a very famous tool for continuous integration. We can easily sync our projects with TravisCI and test it. According to the official document, we can use TravisCI in the follwing steps.

- Push our code to github
- Github triggers TravisCI file to build our project and see how it works.
- Appears whether our build passes or fails. (Hope it passes!)
- TravisCI deploys to Heroku
- TravisCI tells our account how it works.

We signed up for an account for TravisCI named LaiWang2020, and sync up with our github account. After signing up, it can choose one / more of your repository to build. We are testing with our `JSoup` project, which is named `SWE261`.



Create a file to build

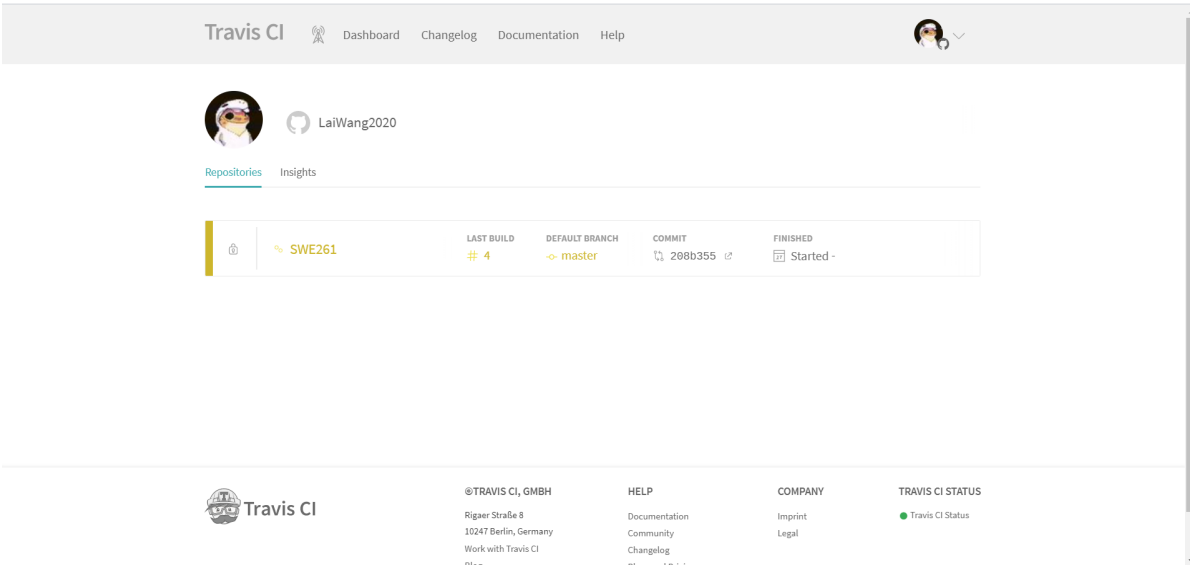
To use TravisCI, we need to create a .travis.yml file for our project. Because TravisCI cannot choose a repository that doesn't belong to you, even for joint collaboration, Lai Wang signed up for an account and synced his GitHub. He forked our `JSoup` project again, and TravisCI project is firstly done in here: https://github.com/LaiWang2020/SWE261/blob/master/.travis.yml. .travis.yml file contains information below.

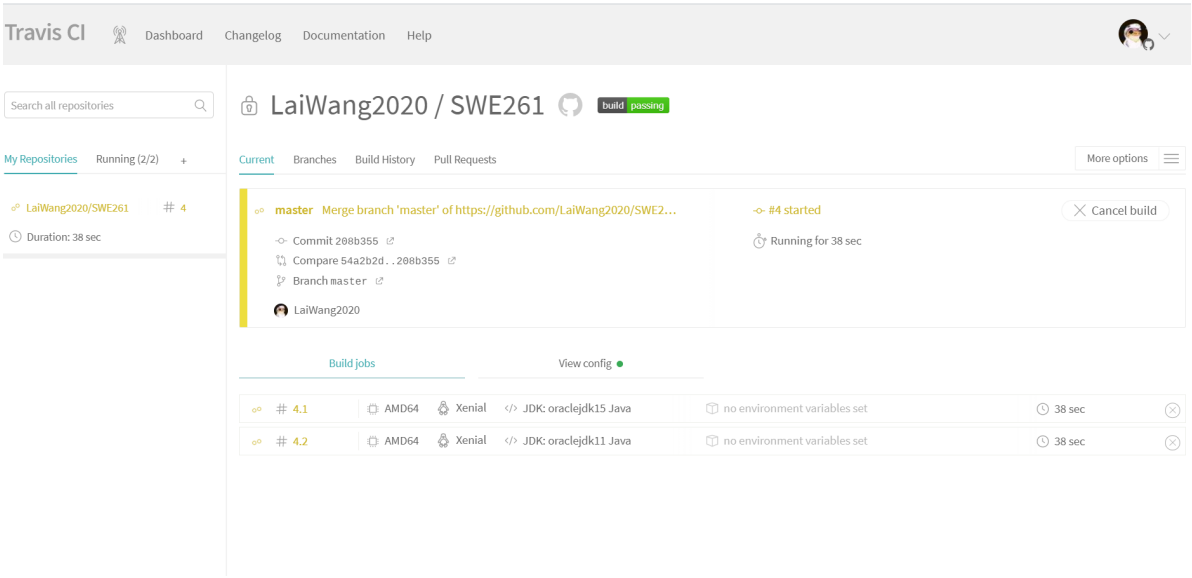
```
language: java

jdk:
  - oraclejdk15
  - oraclejdk11
```

Commit the changes and verify

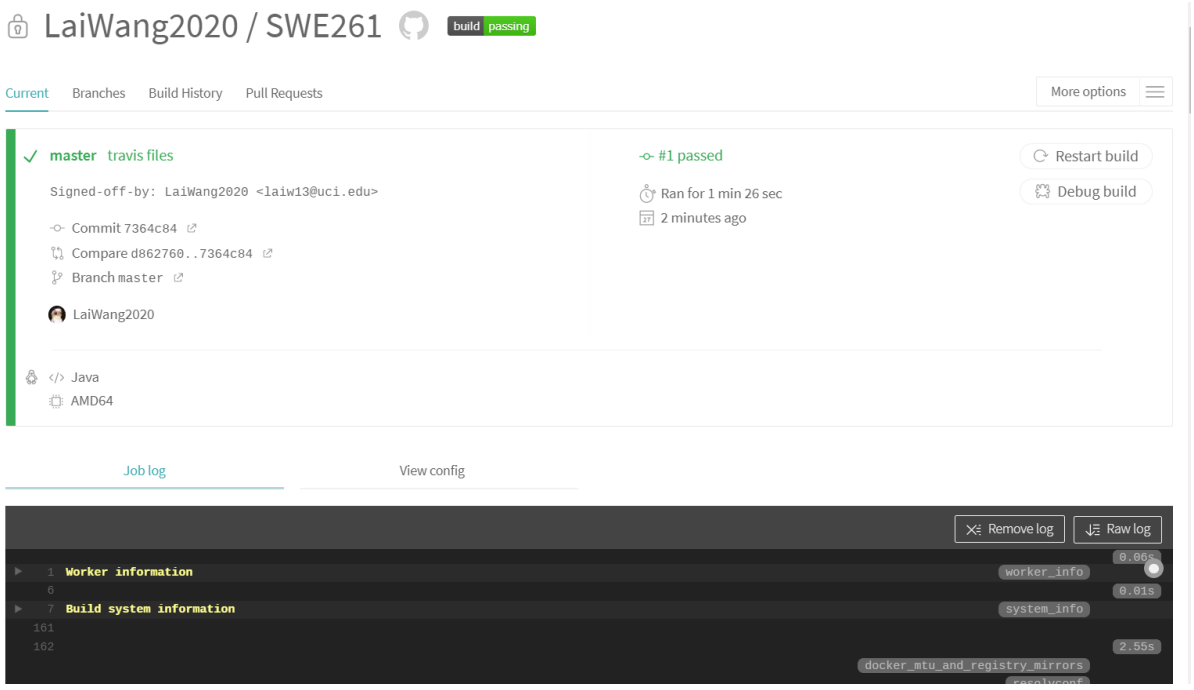
After we add this .travis.yml file, we committed our repository from local to GitHub. Our TravisCI will appear like this.



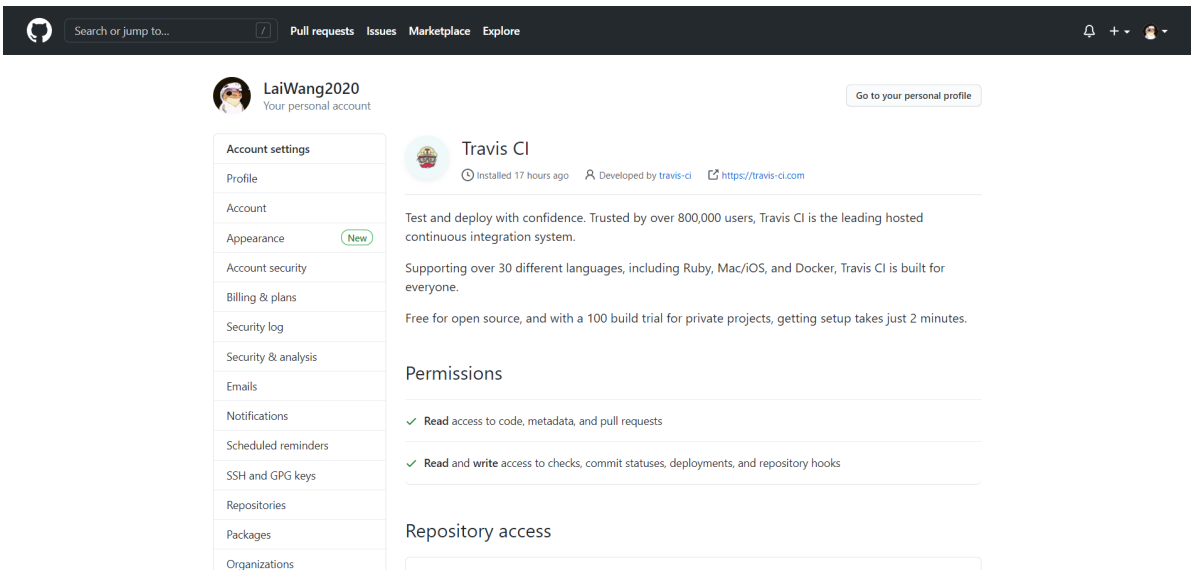


After approximately three minutes, the build process will be done, and it appears like this.

It ran for 1 min 26 sec, and build passes.



Adding extension to GitHub, provided by TravisCI, GitHub can also appear the result of our project `JSoup`. The extension is appeared in the account settings



And the `build passing` is appeared in our repository.

Awesome! It;s more like a professional code repository.