

第一次课程实验

持续集成

“

什么是持续集成？

”

持续集成

CONTINUOUS INTEGRATION——CI

- 持续集成指的是，频繁地（一天多次）将代码集成到主干
- 好处：
 - 快速发现错误
 - 防止分支大幅偏离主干
- 目的：让产品可以快速迭代，同时还能保持高质量
- 代码集成到主干之前，必须通过自动化测试。只要有一个测试用例失败，就不能集成。

持续集成

CONTINUOUS INTEGRATION——CI

Add controller:user-info in staff system;Add controller:chat-record c... #83

Merged

matinjugou merged 1 commit into master from huangchao on 4 Jan

Conversation 0

Commits 1

Checks 0

Files changed 7

+63 -1



matinjugou commented on 4 Jan

Owner + ...

...ompany in user system;Record user on userReg;Give staff info on getQueue

>Add controller:user-info in staff system;Add controller:chat-record
C...
...ompany in user system;Record user on userReg;Give staff info on
getQueue

✓ 42db458

matinjugou requested a review from effie-0 on 4 Jan

matinjugou merged commit 80217cc into master on 4 Jan

Hide details

Revert

2 checks passed

✓ continuous-integration/travis-ci/pr The Travis CI build passed

Details

✓ continuous-integration/travis-ci/push The Travis CI build passed

Details

Edit

Reviewers



effie-0



Assignees

No one—assign yourself



Labels

None yet



Projects

None yet



Milestone

No milestone



Notifications

Unsubscribe

You're receiving notifications
from matinjugou.

持续集成

CONTINUOUS INTEGRATION——CI

- 常用的可以与Github集成的持续集成工具
 - Travis CI、Circle CI、Jenkins、AppVeyor、CodeShip、Drone、Semaphore CI、Buildkite、Wercker、TeamCity

“

怎么做持续集成？

”

TRAVIS

- 使用所有互联网软件的第一步：注册并绑定账号
- 根据指引在github上为travis激活权限，可以是部分库，也可以是全部
- 等待travis读取库信息
- 编写.travis.yml文件
- 在代码里写一些乱七八糟的东西，并假装不知道有bug
- 将travis的status image放进README文件中
- Push! OR Pull Request!

TRAVIS

在GITHUB上授权

The screenshot shows a GitHub user profile page with a sidebar on the left containing 'Personal settings' and various account management links. The main content area displays a card for 'Travis CI' which has been installed. The card includes a small icon of a character wearing a hard hat, the name 'Travis CI', a timestamp ('Installed 4 minutes ago'), developer information ('Developed by travis-ci'), and a link to the website (<https://travis-ci.com>). Below the card, there is descriptive text about Travis CI's capabilities and a note about its free availability for open source projects.

Travis CI is already installed as suggested. ×

Personal settings

Profile

Account

Emails

Notifications

Billing

SSH and GPG keys

Security

Sessions

Blocked users

Repositories

Organizations

Saved replies

Applications

Travis CI

Installed 4 minutes ago Developed by [travis-ci](#) <https://travis-ci.com>

Test and deploy with confidence. Trusted by over 800,000 users, Travis CI is the leading hosted continuous integration system.

Supporting over 30 different languages, including Ruby, Mac/iOS, and Docker, Travis CI is built for everyone.

Free for open source, and with a 100 build trial for private projects, getting setup takes just 2 minutes.

Permissions

✓ Read access to code

✓ Read access to metadata and pull requests

✓ Read and write access to checks, commit statuses, deployments, and repository hooks

TRAVIS

查看已经被TRAVIS追踪的库

The screenshot shows the Travis CI interface for the repository `matinjugou/yogurt-backend`. The repository has a green "build passing" status. The build log is visible at the bottom, showing logs for worker information and build system information.

Travis CI Changelog Documentation

Search all repositories

My Repositories Running (0/0) +

- ✓ `matinjugou/yogurt-backend` # 300
Duration: 1 min 29 sec Finished: 9 months ago
- ✓ `matinjugou/yogurt-frontend` # 298
Duration: 1 min 33 sec Finished: 9 months ago
- `matinjugou/Mask_RCNN`
Duration: -
- `matinjugou/leetcode`
Duration: -
- `matinjugou/host`
Duration: -
- `matinjugou/Transfer-Adversarial`
Duration: -

`matinjugou / yogurt-backend` build passing

Current Branches Build History Pull Requests More options

✓ `master` Merge pull request #86 from `matinjugou/test` -o #300 passed
Duration: Ran for 1 min 29 sec Finished: 9 months ago

Test
-o Commit 01b26b8
↳ Compare ed04029..01b26b8
↳ Branch master

Huang Chao

Node.js: node

Job log View config

```
1 Worker information
6 mode of '/usr/local/clang-5.0.0/bin' changed from 0777 (rwxrwxrwx) to 0775 (rwxrwxr-x)
7 Build system information
404
405 removed '/etc/apt/sources.list.d/basho_riak.list'
406 Executing: /tmp/tmp.DJD4kIP9Ri/gpg.1.sh --keyserver
407 hkp://keyserver.ubuntu.com:80
```

TRAVIS

编写YML文件

```
1 sudo: false
2 language: node_js
3 node_js:
4   - node
5 services:
6   - mongodb
7   - redis-server
8   - mysql
9 before_install:
10  - mysql -e 'CREATE DATABASE yogurt_test CHARACTER SET utf8 COLLATE utf8_unicode_ci;'
11  - mysql -e 'CREATE TABLE yogurt_test.staff
```

```
106  - mongo yogurt_test --eval 'db.createUser({user:"travis",pwd:"test",roles:["readWrite"]});'
107  - mongo yogurt_test --eval 'db.createCollection("sessionPair");'
108 cache:
109   directories:
110     - node_modules
111 before_script:
112   - npm prune
113 script:
114   - npm run lint
115   - npm run compile
116   - npm run test
```

TRAVIS

将TRAVIS STATUS IMAGE放进README文件

The screenshot shows a GitHub repository page for `matinjugou / yogurt-backend`. A modal window titled "Status Image" is open, displaying configuration options for a Travis CI build status image. The modal includes fields for "BRANCH" (set to "master") and "Image URL" (set to <https://travis-ci.com/matinjugou/yogurt-backend.svg?branch=master>). Below the modal, a portion of the repository's README file is shown, containing two lines of code that generate a Travis CI status badge:

```
1 # Yogurt-Backend
2 [![Build Status](https://travis-ci.com/matinjugou/yogurt-backend.svg?branch=master)](https://travis-ci.com/matinjugou/yogurt-backend)
```

On the left side of the image, a separate white box displays the text "Yogurt-Backend" and a "build passing" badge.

TRAVIS

PUSH! OR PULL REQUEST!

Add controller:user-info in staff system;Add controller:chat-record c... #83

Merged matinjugou merged 1 commit into master from huangchao on 4 Jan

Conversation 0 Commits 1 Checks 0 Files changed 7 +63 -1

matinjugou commented on 4 Jan

...ompany in user system;Record user on userReg;Give staff info on getQueue

Add controller:user-info in staff system;Add controller:chat-record c... 42db458

...ompany in user system;Record user on userReg;Give staff info on getQueue

matinjugou requested a review from effie-0 on 4 Jan

matinjugou merged commit 80217cc into master on 4 Jan Hide details Revert

2 checks passed

continuous-integration/travis-ci/pr The Travis CI build passed Details

continuous-integration/travis-ci/push The Travis CI build passed Details

Reviewers effie-0

Assignees No one—assign yourself

Labels None yet

Projects None yet

Milestone No milestone

Notifications

Unsubscribe

You're receiving notifications from this organization

TRAVIS

查看TRAVIS的输出

✓ master Merge pull request #86 from matinjugou/test

Test

- o Commit 01b26b8 ↗
- ↳ Compare ed04029..01b26b8 ↗
- ↳ Branch master ↗

Huang Chao

Node.js: node

↻ #300 passed

🕒 Ran for 1 min 29 sec

27 9 months ago

↻ Restart build

```
2200 $ npm run test                                     9.24s
2201
2202 > thinkjs-application@1.0.0 test /home/travis/build/matinjugou/yogurt-backend
2203 > THINK_UNIT_TEST=1 nyc mocha -t 20000 --delay
2204
2205 (node:4685) Warning: process.on(SIGPROF) is reserved while debugging
2206
2207
2208   test
2209     user
2210       GET queue
2211 [2018-01-06 15:34:02.262] [4697] [INFO] - mysql://root:@127.0.0.1:3306/yogurt_test
2212 [2018-01-06 15:34:02.270] [4697] [INFO] - SQL: SELECT * FROM `staff` WHERE ( `onlineStatus` = 1 ) AND ( `queueCount` < 30 ) ORDER BY queueCount ASC, Time: 7ms
2213 [2018-01-06 15:34:02.274] [4697] [INFO] - SQL: SELECT * FROM `staff` WHERE ( `staffId` = '1_s1' ) LIMIT 1, Time: 2ms
2214 [2018-01-06 15:34:02.277] [4697] [INFO] - SQL: SHOW COLUMNS FROM `staff`, Time: 2ms
2215 [2018-01-06 15:34:02.279] [4697] [INFO] - SQL: UPDATE `staff` SET `servingCount`=servingCount+1,`queueCount`=queueCount+1 WHERE ( `staffId` = '1_s1' ), Time: 1ms
2216 [2018-01-06 15:34:02.283] [4697] [INFO] - mongodb://travis:test@127.0.0.1:27017/yogurt_test
2217     ✓ The first user should be arranged (122ms)
2218 [2018-01-06 15:34:02.362] [4697] [INFO] - SQL: SELECT * FROM `staff` WHERE ( `onlineStatus` = 1 ) AND ( `queueCount` < 30 ) ORDER BY queueCount ASC, Time: 1ms
2219 [2018-01-06 15:34:02.364] [4697] [INFO] - SQL: SELECT * FROM `staff` WHERE ( `staffId` = '1_s2' ) LIMIT 1, Time: 1ms
2220 [2018-01-06 15:34:02.365] [4697] [INFO] - SQL: UPDATE `staff` SET `servingCount`=servingCount+1,`queueCount`=queueCount+1 WHERE ( `staffId` = '1_s2' ), Time: 1ms
2221     ✓ The second user should be arranged with different user
2222 [2018-01-06 15:34:02.374] [4697] [INFO] - SQL: SELECT * FROM `staff` WHERE ( `onlineStatus` = 1 ) AND ( `queueCount` < 30 ) ORDER BY queueCount ASC, Time: 1ms
2223     ✓ The third user should not be arranged with different user
```

“

练一练！

— $1+1=2$, 数学简单吧? 现在我们来练一练: 求证黎曼猜想

”

课程实验

UNEXPECTED HOMEWORK

- Clone本次实验的代码，在github上为它创建一个仓库，并将push到这个新的仓库中。
- 使用Travis或者其他持续集成工具为项目配置CI环境
- 根据实验文档完成API的编写，并编写单元测试
- 编写yml配置CI流程，完成自动化测试
- 在你的技术博客上写点东西记录一下心得

课程实验

评价标准

- 规范使用Git
- 正确编写yml文件，正确配置持续集成环境
- 正确并合理的实现API逻辑，**尽可能**的避免可能出现的bug，充分考虑抢票这类事务的特性，考虑业务上线后可能发生的情况
- 正确编写测试代码，对核心逻辑进行有效的单元测试，要求使用持续集成的方式进行自动化测试
- 提交文档要对所做工作有详细的叙述，包括但不限于以下内容：
 - – 交付产品：部署测试号的二维码（保证11月1日前可用）、后台管理员的账号和密码
 - – 产品目标：所开发的系统功能和性能方面的要求
 - – 设计实现：不同模块的设计实现方案以及重点或难点问题解决等
 - – 测试方案：介绍测试方案、测试用例设计、缺陷汇总与测试结果，至少包括单元测试和功能测试，性能测试可选
 - – 系统部署：部署方法与规范
 - – 过程管理：包括迭代过程、任务分工、持续集成等

课程实验

代码解释

- `adminpage`、`userpage`、`wechat` Django的APP
- `codex`、`static`、`templates` 公用组件与公共素材
- `manage.py` Django管理入口
- `configs.example.json`、`requirements.txt` 配置文件
- `.gitignore` Git仓库文件
- `docs`、`LICENSE`、`README.md` 辅助性文档

课程实验

代码解释

```
migrations  
__init__.py  
admin.py  
apps.py  
models.py  
tests.py  
urls.py  
views.py
```

课程实验

代码解释

26 lines (19 sloc) | 745 Bytes

Raw Blame History



```
1 from codex.baseerror import *
2 from codex.baseview import APIView
3
4 from wechat.models import User
5
6
7 class UserBind(APIView):
8
9     def validate_user(self):
10         """
11             input: self.input['student_id'] and self.input['password']
12             raise: ValidationError when validating failed
13         """
14         raise NotImplementedError('You should implement UserBind.validate_user method')
15
16     def get(self):
17         self.check_input('openid')
18         return User.get_by_openid(self.input['openid']).student_id
19
20     def post(self):
21         self.check_input('openid', 'student_id', 'password')
22         user = User.get_by_openid(self.input['openid'])
23         self.validate_user()
24         user.student_id = self.input['student_id']
25         user.save()
```

“

Q&A

— 对于我不会的问题，我只能说：“无可奉告”

”

EMAIL: THSS15_HUANGC@163.COM

