

🔒

CDMS.Zhouxuan.2023Fall.DASE / AllStuRead

Private

👁 取消关注

61

☆ 点赞 1

🔗 派生 1

<> 代码

💬 话题 4

🔗 合并请求 0

📄 版本发布 0

📖 百科

🔄 动态

标签

里程碑

创建话题

# #1 第一次大作业

已关闭

由 陈熙之 于 5 个月前创建 · 8 条评论

陈熙之

评论于 5 个月前

😊 ...

## bookstore

### 功能

实现一个提供网上购书功能的网站后端。

网站支持书商在上面开商店，购买者可以通过网站购买。

买家和卖家都可以注册自己的账号。

一个卖家可以开一个或多个网上商店，  
买家可以为自己的账户充值，在任意商店购买图书。

支持 下单->付款->发货->收货 流程。

1.实现对应接口的功能，见项目的 doc 文件夹下面的 .md 文件描述 （60%）

其中包括：

1)用户权限接口，如注册、登录、登出、注销

2)买家用户接口，如充值、下单、付款

3)卖家用户接口，如创建店铺、填加书籍信息及描述、增加库存

通过对应的功能测试，所有 test case 都 pass

2.为项目添加其它功能：（40%）

1)实现后续的流程

发货 -> 收货

2)搜索图书

用户可以通过关键字搜索，参数化的搜索方式；  
如搜索范围包括，题目，标签，目录，内容；全站搜索或是当前店铺搜索。  
如果显示结果较大，需要分页  
(使用全文索引优化查找)

3)订单状态，订单查询和取消定单

用户可以查自己的历史订单，用户也可以取消订单。

取消定单可由买家主动地取消定单，或者买家下单后，经过一段时间超时仍未付款，定单也会自动取消。

## bookstore目录结构

```
bookstore
|-- be                                后端
    |-- model                        后端逻辑代码
    |-- view                         访问后端接口
    |-- ....
|-- doc                               JSON API规范说明
|-- fe                               前端访问与测试代码
    |-- access                       效率测试
    |-- bench
```

分支/标记未指定

标签

未选择标签

里程碑

未选择里程碑

指派成员

未指派成员

2 名参与者

陈熙之

51h 36min 55s

陈熙之

51h 36min 55s

通知

取消订阅

总花费时间：51h 36min 55s

陈熙之

51h 36min 55s

到期时间

📅 2023-11-11 过期

依赖话题

此话题当前没有任何依赖。

```
|-- data
|-- book.db          sqlite 数据库(book.db, 较少量的测试数据)
|-- book_lx.db       sqlite 数据库(book_lx.db,  较大量的测试数据, 要从网盘下载)
|-- scraper.py        从豆瓣爬取的图书信息数据的代码
|-- test             功能性测试 (包含对前60%功能的测试, 不要修改已有的文件, 可以提pull request或
|-- conf.py          测试参数, 修改这个文件以适应自己的需要
|-- conftest.py      pytest初始化配置, 修改这个文件以适应自己的需要
|-- ....
|-- ....
```

## 安装配置

安装 python (需要 python3.6 以上)

进入 bookstore 文件夹下:

安装依赖

```
pip install -r requirements.txt
```

执行测试

```
bash script/test.sh
```

(注意: 如果提示 "RuntimeError: Not running with the Werkzeug Server", 请输入下述命令, 将 flask 和 Werkzeug 的版本均降低为2.0.0。)

```
pip install flask==2.0.0
pip install Werkzeug==2.0.0
```

## 要求

2~3人一组, 做好分工, 完成下述内容:

1.bookstore 文件夹是该项目的 demo, 采用 Flask 后端框架与 SQLite 数据库, 实现了前60%功能以及对应的测试用例代码。  
**要求大家创建本地 MongoDB 数据库, 将 bookstore/fe/data/book.db 中的内容以合适的形式存入本地数据库, 后续所有数据读写都在本地的 MongoDB 数据库中进行**

书本的内容可自行构造一批, 也可参从网盘下载, 下载地址为:

```
https://pan.baidu.com/s/1bjCOW8Z5N_C1cqU54Pdt8g
```

提取码:

```
hj6q
```

2.在完成前60%功能的基础上, 继续实现后40%功能, 要有接口、后端逻辑实现、数据库操作、代码测试。对所有接口都要写 test case, 通过测试并计算测试覆盖率 (尽量提高测试覆盖率)。

3.尽量使用索引, 对程序与数据库执行的性能有考量

4.尽量使用 git 等版本管理工具

5.不需要实现界面, 只需通过代码测试体现功能与正确性

## 报告内容

1.每位组员的学号、姓名, 以及分工

2.文档数据库设计: 文档 schema

3.对60%基础功能和40%附加功能的接口、后端逻辑、数据库操作、测试用例进行介绍, 展示测试结果与测试覆盖率。

4.如果完成, 可以展示本次大作业的亮点, 比如要求中的"3 4"两点。

注: 验收依据为报告, 本次大作业所作的工作要完整展示在报告中。

## 验收与考核准测

- 提交 代码+报告 压缩包到 作业提交入口
- 命名规则: 2023\_ECNU\_PJ1\_第几组(.zip)
- 提交截止日期: **2023.11.10 23:59**



考核标准:

1. 没有提交或没有实质的工作, 得D
2. 完成"要求"中的第1点, 可得C
3. 完成前3点, 通过全部测试用例且有较高的测试覆盖率, 可得B
4. 完成前2点的基础上, 体现出第3 4点, 可得A
5. 以上均为参考, 最后等级会根据最终的工作质量有所调整

 bookstore.zip

30 KiB

  陈熙之 到期时间 2023-11-30 5 个月前 已添加

  陈熙之 开始工作 5 个月前

  陈熙之 到期时间 2023-11-30 5 个月前 已删除

  陈熙之 到期时间 2023-11-11 5 个月前 已添加



陈熙之 评论于 5 个月前

所有者  ...



同学们对第一次大作业有什么问题可以在此话题下回复






唐小卉 评论于 5 个月前


 ...

"通过对应的功能测试, 所有 test case 都 pass"这个要求没太理解, 能否稍微解释一下


  陈熙之 关闭这个话题 5 个月前

  陈熙之 停止工作 5 个月前  
 51h 36min 55s

  陈熙之 重新打开此话题 5 个月前



陈熙之 评论于 5 个月前

所有者  ...


通过运行test.sh可以发现现在有很多测试用例没有通过, 可以在test文件夹中查看每个具体的测试用例和测试方法, 前60%的分数就来自于补完现有代码, 使test文件夹中所有测试用例通过, 详细相关的可以自行查阅pytest相关文档 > "通过对应的功能测试, 所有 test case 都 pass"这个要求没太理解, 能否稍微解释一下




陈熙之 评论于 4 个月前

所有者  ...


更改数据库至 MongoDB 是指把所有用 SQLite 数据库保存的数据都要用 Mongoddb 保存, 包括 book.db 和 be.db




陈熙之 评论于 4 个月前

所有者  ...

提醒一下, test 在实际开发流程中是在代码都大概完善的情况下, 寻找没有考虑到的 bug 和功能上的缺少的一个方法, 而不是用来调试代码的。需要调试代码请以 debug 模式启动 app.py, 使用 postman 等软件对本地运行的服务端进行模拟访问来调试代码, 否则代码还没完成就直接运行 test 会有许多无法定位的错误。



陈熙之 评论于 4 个月前

所有者  ...

总结一下近期同学们问的较多的问题:

1. 在 Windows 环境下无法运行 script/test.sh。

有几种解决办法:

- i. 使用 `git bash` 执行 `bash script/test.sh` ;
  - ii. 安装 WSL Linux 子系统，通过子系统的来运行；
  - iii. (最推荐) 使用 PyCharm 的终端（界面左下角），直接在你配置好的 Python 环境下运行 `test.sh` 中的内容，具体操作参考视频：<https://www.bilibili.com/video/BV1Lu4y1h7Pn/>
2. 对覆盖率的要求具体指什么？

作业说明文档中的描述是：

对所有接口都要写 `test case`，通过测试并计算测试覆盖率（尽量提高测试覆盖率）

并没有要求到100%，这是为什么呢？在运行 `test.sh` 之后，会在根目录下生成一个 `htmlcov` 文件夹其中有每个文件的覆盖率可视化展示，以 `be\model\user.py` 为例，用浏览器打开后可以看到这个文件的测试覆盖率为76%，被 `test` 函数运行到的行被标为绿色，未被运行到的行为红色。那么为什么只有76%呢？因为我们这里使用了 `try ... except ...` 语句用于错误捕获，但是由于只有代码运行时出现严重错误导致无法继续运行才会进入 `except` 语句中，所以这是 `test` 函数无法测试到的。这就导致了这个文件中所有的 `except` 中的代码行都被标为红色，进一步导致了覆盖率的降低。

但是使用 `try ... except ...` 肯定是不使用更好的代码习惯，所以我们评价时只是要求尽可能覆盖除 `except` 外的代码，覆盖率只要不是太低就可以了。

还是以这个文件为例，虽然覆盖率只有76%，但是其中 `try` 中的代码基本上都被覆盖完了，证明这也是一个比较好的测试。

3. 其他测试都过了，`test_bench`为什么过不了？
- `test_bench` 和其他 `test` 不同，是用于测试后端的吞吐量的（本质上是测试数据库的吞吐量），在 `bench` 文件夹中可以看到具体实现，首先是把 `book.db` 中的内容通过调用插入书本的后端插入到 MongoDB 的数据库中，然后通过大量线程同时调用下订单和付款的后端接口，来测试读写性能，所以这就需要把 `book.db` 保留在 `data` 文件夹中，否则会报错，并且需要清空 MongoDB 中已有的书本数据，否则在插入书本时会报插入重复数据的错误。



陈熙之 评论于 4 个月前

所有者

修正两个错误：

- 1. `be/model/user.py` 中的 `jwt_encode` 函数的 `return` 语句（第22行）改为 `return encoded.decode("utf-8")`。
- 2. 文档中说明所有对 SQL 的操作都要替换成对 MongoDB 的操作，这是一个错误，`fe/access/book.py` 中的 SQL 不需要更改，因为这里是读取 `book.db` (或者 `book_lx.db`) 中的数据，并调用 `add_book` 函数来插入到自定义的数据库中用于测试插入数据的性能，所以和自定义数据库无关。但是如果已经改为 MongoDB 操作并且能正常运行 `test_bench`，那就不需要改回 SQL。



陈熙之 评论于 4 个月前

所有者

第三个错误：`fe/test/gen_book_data.py`，`fe/test/test_add_book.py`，`fe/test/test_add_stock_level.py` 三个测试文件中，`book_db = book.BookDB()` 应改为 `book_db = book.BookDB(conf.Use_Large_DB)`，同时在文件头补上 `from fe import conf`，这样就能不改文件名，而是通过更改 `fe/conf.py` 中的 `Use_Large_DB` 来控制是否使用 `book_lx.db` 了，由马也弛同学报告并修复。

陈熙之 关闭这个话题 3 个月前



撰写 预览

B I S H<sub>1</sub> H<sub>2</sub> H<sub>3</sub> H<sub>Δ</sub> H<sub>▼</sub> </> “ ≡ ≡ 🔗 🖼️ 📄 - ✍️ 📁

lines: 0 words: 0 0:0

拖动文件或者点击此处上传。

评论