

黑盒测试

目录

- [目录](#)
- [流程](#)
- [改动](#)

流程

1. 打开 MySQL 命令行

```
$ ../../../../usr/local/mysql/bin/mysql -uroot -pxhd200176
```

2. 删除并重新建数据库 `e_book`。

```
drop database e_book;
create database e_book;
use e_book
$$
```

3. 执行脚本命令插入三个用户和 1000 本书籍。

```
delimiter $$
drop procedure if exists myproc4;
create procedure myproc4 ()
begin
# 插入 1000 条书籍数据
declare num int ;
set num = 1 ;
while num <= 1000 do
insert into book(id, author, bookname, image, introduction, inventory, isbn, price, state,
type) values (num, 'xuhuidong', 'Algorithm', 'img url', 'This is for test.', 2000, '123-
321-123-321', 99, 1, 'Computer Science');
set num = num + 1 ;
end while;
# 插入唯一的 Admin 和 一个正常用户 和 一个被禁用户
insert into user(id, username, password, email, role, state) values (1, 'Admin',
'Password1234', 'Admin@126.com', 'ADMIN', 'Normal');
insert into user(id, username, password, email, role, state) values (2, 'Customer',
'customer1234', 'Customer@126.com', 'USER', 'Normal');
insert into user(id, username, password, email, role, state) values (3,
'Forbidden@126.com', 'customer1234', 'Forbidden', 'USER', 'Forbidden');
```

```
end$$
call myproc4()$$
```

4. 运行后端代码。
5. 使用 `postman` 进行测试。

测试

注册正常

编写测试代码。

```
// 获取请求中的参数
var query = {};
pm.request.url.query.all().forEach((param) => { query[param.key] = param.value});
// 测试请求返回码是否正常
pm.test("Successful POST request", function () {
    pm.expect(pm.response.code).to.be.eql(200);
});
// 测试返回时间是否符合标准
pm.test("Response time is less than 2000ms", function () {
    pm.expect(pm.response.responseTime).to.be.below(2000);
});
// 测试返回值是否正确
pm.test("Your test name", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.username).to.eql(query.username);
    pm.expect(jsonData.password).to.eql(query.password);
    pm.expect(jsonData.email).to.eql(query.email);
});
```

测试通过。

All Tests	Passed (3)	Failed (0)
Iteration 1		
POST register...	http://localhost:9090/register?username={{register-normal-username}}&password={{register-normal-password}}&re_...	/ register... 200 OK 620 ms 366 B
Pass	Successful POST request	
Pass	Response time is less than 2000ms	
Pass	Your test name	

检查数据库，用户注册成功。

```
[mysql> select * from user;
```

```
[    -> $$
```

id	email	password	role	state	username
1	Admin@126.com	Password1234	ADMIN	Normal	Admin
2	Customer@126.com	customer1234	USER	Normal	Customer
3	Forbidden@126.com	customer1234	Forbidden	USER	Forbidden
4	ebook@126.com	ebook12345	USER	Normal	ebook-normal

```
4 rows in set (0.00 sec)
```

注册异常

编写测试代码。

```
// 测试请求返回码是否正常
```

```
pm.test("Successful POST request", function () {  
  pm.expect(pm.response.code).to.be.oneOf([406, 500]);  
});
```

```
// 测试返回时间是否符合标准
```

```
pm.test("Response time is less than 2000ms", function () {  
  pm.expect(pm.response.responseTime).to.be.below(2000);  
});
```

测试通过。

All Tests Passed (74) Failed (0)

Pass Successful POST request

Pass Response time is less than 2000ms

Iteration 33

POST register... http://localhost:9090/register?username={{register-abnormal-username}}&password={{register-abnormal... / register... 406 Not Acceptable 6 ms 224 B

Pass Successful POST request

Pass Response time is less than 2000ms

登录正常

编写测试代码。

```
// 获取请求中的参数
```

```
var query = {};
```

```
pm.request.url.query.all().forEach((param) => { query[param.key] = param.value});
```

```
// 测试请求返回码是否正常
```

```
pm.test("Successful GET request", function () {
```

```
    pm.expect(pm.response.code).to.be.eql(200);
  });
  // 测试返回时间是否符合标准
  pm.test("Response time is less than 2000ms", function () {
    pm.expect(pm.response.responseTime).to.be.below(2000);
  });
  // 测试返回值是否正确
  pm.test("Your test name", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.username).to.eql(query.username);
    pm.expect(jsonData.password).to.eql(query.password);
  });
```

测试通过。

All Tests

Passed (6)

Failed (0)

Iteration 1

GET login-normal

http://localhost:9090/checkGotoHome?username={{login-normal-username}}&password={{login-normal-password}}

/ login-normal

200 OK

11 ms

362 B

Pass

Your test name

Pass

Successful GET request

Pass

Response time is less than 2000ms

Iteration 2

GET login-normal

http://localhost:9090/checkGotoHome?username={{login-normal-username}}&password={{login-normal-password}}

/ login-normal

200 OK

42 ms

367 B

Pass

Successful GET request

Pass

Your test name

Pass

Response time is less than 2000ms

登录异常

编写测试代码。

```
// 测试请求返回码是否正常
pm.test("Successful GET request", function () {
  pm.expect(pm.response.code).to.be.oneOf([404, 406, 500]);
});
// 测试返回时间是否符合标准
pm.test("Response time is less than 2000ms", function () {
  pm.expect(pm.response.responseTime).to.be.below(2000);
});
```

测试通过。

All Tests

Passed (14)

Failed (0)

Iteration 1

GET

login-abnor...

http://localhost:9090/checkGotoHome?username={{login-abnormal-username}}&password={{login-abnorm...

/ login-a...

404 Not Found

81 ms

219 B

Pass

Successful GET request

Pass

Response time is less than 2000ms

Iteration 2

GET

login-abnor...

http://localhost:9090/checkGotoHome?username={{login-abnormal-username}}&password={{login-abnorm...

/ login-a...

404 Not Found

11 ms

219 B

Pass

Response time is less than 2000ms

Pass

Successful GET request

下订单正常

编写测试代码。

```
// 测试请求返回码是否正常
pm.test("Successful POST request", function () {
    pm.expect(pm.response.code).to.be.eql(200);
});
// 测试返回时间是否符合标准
pm.test("Response time is less than 2000ms", function () {
    pm.expect(pm.response.responseTime).to.be.below(2000);
});
```

测试通过。

All Tests

Passed (17)

Failed (3)

Iteration 1

POST

mak...

http://localhost:9090/addOrderFromUser?username={{makeorder-normal-username}}&password={{makeorder-normal-... / makeor...

200 OK

969 ms

212 B

Pass

Successful POST request

Pass

Response time is less than 2000ms

Iteration 2

POST

mak...

http://localhost:9090/addOrderFromUser?username={{makeorder-normal-username}}&password={{makeorder-normal-p... / makeor...

200 OK

61 ms

212 B

Pass

Successful POST request

Pass

Response time is less than 2000ms

检查数据库下订单成功。

```
[mysql> select * from orders;
-> $$

+----+-----+-----+-----+-----+-----+
| id | address | price | receivername | timestamp | userid |
+----+-----+-----+-----+-----+-----+
| 1 | SJTU | 99000 | huidongxu | 2022-03-20 10:18:50 | 2 |
| 2 | SJTU | 99000 | huidongxu | 2022-03-20 10:18:50 | 2 |
| 3 | SJTU | 99000 | huidongxu | 2022-03-20 10:18:50 | 2 |
| 4 | SJTU | 99000 | huidongxu | 2022-03-20 10:18:50 | 2 |
| 5 | SJTU | 99000 | huidongxu | 2022-03-20 10:18:50 | 2 |
| 6 | SJTU | 99 | huidongxu | 2022-03-20 10:18:50 | 2 |
| 7 | SJTU | 198 | huidongxu | 2022-03-20 10:18:50 | 2 |
| 8 | SJTU | 197901 | huidongxu | 2022-03-20 10:18:50 | 2 |
| 9 | SJTU | 198000 | huidongxu | 2022-03-20 10:18:50 | 2 |
| 10 | SJTU | 1980 | huidongxu | 2022-03-20 10:18:50 | 2 |
+----+-----+-----+-----+-----+-----+

10 rows in set (0.00 sec)
```

下订单异常

编写测试代码。

```
// 测试请求返回码是否正常
pm.test("Successful POST request", function () {
  pm.expect(pm.response.code).to.be.oneOf([406, 500]);
});

// 测试返回时间是否符合标准
pm.test("Response time is less than 2000ms", function () {
  pm.expect(pm.response.responseTime).to.be.below(2000);
});
```

测试通过。

All Tests Passed (17) Failed (3)

Iteration 1

POST mak...

http://localhost:9090/addOrderFromUser?username={{(makeorder-normal-username)}}&password={{(makeorder-normal-p... / makeor...

200 OK 969 ms 212 B

Pass

Successful POST request

Pass

Response time is less than 2000ms

Iteration 2

POST mak...

http://localhost:9090/addOrderFromUser?username={{(makeorder-normal-username)}}&password={{(makeorder-normal-p... / makeor...

200 OK 61 ms 212 B

Pass

Successful POST request

Pass

Response time is less than 2000ms

订单筛选正常

编写测试代码。

```
// 获取请求中的参数
var query = {};
pm.request.url.query.all().forEach((param) => { query[param.key] = param.value});
// 测试请求返回码是否正常
pm.test("Successful GET request", function () {
    pm.expect(pm.response.code).to.be.eql(200);
});
// 测试返回时间是否符合标准
pm.test("Response time is less than 2000ms", function () {
    pm.expect(pm.response.responseTime).to.be.below(2000);
});
// 测试返回值是否正确
pm.test("Your test name", function () {
    var jsonData = pm.response.json();

    jsonData.forEach((order) => {
        console.log(order.timestamp.substring(0, 10));
        console.log(query.startdate);

        pm.expect(Date.parse(query.startdate)).to.be.below(Date.parse(order.timestamp.substring(0, 10)));
        pm.expect(Date.parse(order.timestamp.substring(0, 10))).to.be.below(Date.parse(query.enddate));
    });
    // for (const orderItem of jsonData.orderItems) {
    //     pm.expect(orderItem.order).to.be.below(200);
    // };
});
```

测试通过。

All Tests	Passed (3)	Failed (0)
Iteration 1		
GET order...	http://localhost:9090/getOrdersByBookAndDaterange?username={{orderdaterange-normal-username}}&password={{or... / orderda...	200 OK 793 ms 255 B
Pass	Successful GET request	
Pass	Response time is less than 2000ms	
Pass	Your test name	

订单筛选异常

编写测试代码。

```
// 测试请求返回码是否正常
pm.test("Successful GET request", function () {
    pm.expect(pm.response.code).to.be.oneOf([406, 500]);
});
// 测试返回时间是否符合标准
pm.test("Response time is less than 2000ms", function () {
    pm.expect(pm.response.responseTime).to.be.below(2000);
});
```

测试通过。

All Tests	Passed (12)	Failed (0)
Pass	Response time is less than 2000ms	
Iteration 3		
GET order...	http://localhost:9090/getOrdersByBookAndDaterange?username={{orderdaterange-abnormal-username}}...	/ orderda... 406 Not Acceptable 76 ms 224 B
Pass	Successful GET request	
Pass	Response time is less than 2000ms	
Iteration 4		

用户管理正常

编写测试代码。

```
// 获取请求中的参数
var query = {};
pm.request.url.query.all().forEach((param) => { query[param.key] = param.value});
// 测试请求返回码是否正常
pm.test("Successful POST request", function () {
    pm.expect(pm.response.code).to.be.eql(200);
});
// 测试返回时间是否符合标准
pm.test("Response time is less than 2000ms", function () {
    pm.expect(pm.response.responseTime).to.be.below(2000);
});
// 测试返回值是否正确
pm.test("Your test name", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData).to.be.oneOf([0, 1]);
});
```


测试通过。

All Tests

Passed (12)

Failed (0)

Iteration 1

POST

edituse...

http://localhost:9090/editUser?username={{edituser-normal-username}}&password={{edituser-normal-password}}&... / edituser...

200 OK

588 ms

254 B

Pass

Successful POST request

Pass

Response time is less than 2000ms

Pass

Your test name

Iteration 2

POST

edituse...

http://localhost:9090/editUser?username={{edituser-normal-username}}&password={{edituser-normal-password}}&... / edituser...

200 OK

106 ms

254 B

Pass

Successful POST request

Pass

Response time is less than 2000ms

Pass

Your test name

管理员统计销量

和订单过滤功能是一致的，也是先取出所有用户的订单，然后前端做统计，所以其实测试了订单过滤就相当于测试了统计销量。

改动

- 注册正常的用户名改为 ebook-normal，防止与注册异常的用户名相冲突。
- 下订单的时间需要改动，改成今天左右的时间。