## Django 项目测试管理工具使用说明（基于 pytest）

### 已使用工具和插件

**pytest**：Python 测试框架，负责测试执行

**pytest-django**：Django 集成插件，用于识别 Django 的测试环境（如数据库、settings 等）

**anyio**：异步支持插件

**pytest.ini**：配置文件，设置 Django 项目、测试路径等

### 如何运行测试

**1.安装依赖**

pip install pytest pytest-django anyio

**2.运行测试**

在项目根目录执行：

pytest

或者指定具体测试文件运行：

pytest bookstore/tests/test\_views.py

### 测试失败：302 != 200

> assert response.status\_code == 200

E assert 302 == 200

E + where 302 = <HttpResponseRedirect ...>.status\_code

**原因：**

返回了 302 重定向，意味着当前视图需要 登录认证，而测试中使用的 client 没有登录。

**解决方案：**

对需要登录的视图，应使用 authenticated\_client（已登录的测试客户端）进行测试。

替换测试函数中的 client 为 authenticated\_client：

def test\_search\_books(authenticated\_client, test\_book):

...

response = authenticated\_client.get(url)

或者：

使用 Django 的 @login\_required 装饰器后，视图就必须使用认证客户端进行访问；

若不需要登录，确认 view 的权限设置是否错误。

### 测试用例示例说明

**test\_models.py**

test\_book\_creation：验证书籍模型的字段

test\_order\_total\_calculation：验证订单金额的计算是否正确

test\_customer\_credit\_rating：验证顾客信用评级的自动更新机制

**test\_views.py**

test\_search\_books：模拟搜索接口（注意是否需要登录）

test\_book\_details：验证图书详情接口返回正确数据

test\_create\_order：模拟下订单接口行为，确认数据创建成功

### pytest.ini 配置文件说明

[pytest]

DJANGO\_SETTINGS\_MODULE = bookshop.settings

python\_files = tests.py test\_\*.py \*\_tests.py