文件编号：3107-SWC2018-20180045

受控状态：■受控 □非受控

保密级别：□公司级 □部门级 ■项目级 □普通级

采纳标准：CMMI DEV V1.2



图美集

**Temage**

项目测试文档

**Version 1.2.2**

2019.5.9

**Written by 3107**



**All Rights Reserved**

目录

[1 引言 1](#_Toc8315777)

[1.1 编写目的 1](#_Toc8315778)

[1.2 项目背景 1](#_Toc8315779)

[1.3 术语和缩略语 1](#_Toc8315780)

[1.4 参考资料 2](#_Toc8315781)

[2 测试计划 2](#_Toc8315782)

[2.1.1 测试策略与目标 2](#_Toc8315783)

[2.1.2 测试范围 5](#_Toc8315784)

[2.1.3 测试环境 5](#_Toc8315785)

[3 单元测试 6](#_Toc8315786)

[3.1 图文匹配模块 6](#_Toc8315787)

[3.1.1 测试用例与执行分析 6](#_Toc8315788)

[3.1.2 分析结果： 9](#_Toc8315789)

[3.1.3 测试结果综合分析及建议 9](#_Toc8315790)

[3.1.4 测试经验总结 9](#_Toc8315791)

[3.2 用户社区模块 9](#_Toc8315792)

[3.2.1 测试用例与执行分析 9](#_Toc8315793)

[3.2.2 分析结果： 16](#_Toc8315794)

[3.2.3 测试结果综合分析及建议 16](#_Toc8315795)

[3.2.4 测试经验总结 16](#_Toc8315796)

[4 功能测试 17](#_Toc8315797)

[4.1 图文匹配功能 17](#_Toc8315798)

[4.1.1 测试用例与执行分析 17](#_Toc8315799)

[4.1.2 分析结果： 18](#_Toc8315800)

[4.1.3 测试结果综合分析及建议 18](#_Toc8315801)

[4.1.4 测试经验总结 19](#_Toc8315802)

[4.2 用户社区功能 19](#_Toc8315803)

[4.2.1 测试用例与执行分析 19](#_Toc8315804)

[4.2.2 结果分析 21](#_Toc8315805)

[4.2.3 测试结果综合分析及建议 21](#_Toc8315806)

[4.2.4 测试经验总结 21](#_Toc8315807)

[5 系统测试 22](#_Toc8315808)

[5.1 模型性能测试 23](#_Toc8315809)

[5.1.1 测试用例与执行分析 23](#_Toc8315810)

[文本分类预期性能指标测试： 23](#_Toc8315811)

[分析结果： 23](#_Toc8315812)

[图文匹配预期性能指标测试： 23](#_Toc8315813)

[分析结果： 23](#_Toc8315814)

[预期性能指标测试： 24](#_Toc8315815)

[分析结果： 24](#_Toc8315816)

[5.1.2 测试结果综合分析及建议 24](#_Toc8315817)

[5.1.3 测试经验总结 24](#_Toc8315818)

[5.2 压力测试 24](#_Toc8315819)

[5.2.1 测试用例与执行分析 24](#_Toc8315820)

[图文匹配压力测试用例： 24](#_Toc8315821)

[注： 25](#_Toc8315822)

[分析结果： 25](#_Toc8315823)

[智能搜索压力测试用例： 25](#_Toc8315824)

[分析结果： 26](#_Toc8315825)

[文本分类压力测试用例： 26](#_Toc8315826)

[分析结果: 26](#_Toc8315827)

[图文匹配（A）、文本分类（B）、智能搜索（C）组合压力测试用例： 26](#_Toc8315828)

[分析结果： 27](#_Toc8315829)

[5.2.2 测试结果综合分析及建议 27](#_Toc8315830)

[5.2.3 测试经验总结 28](#_Toc8315831)

记录更改历史

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **更改原因** | **版本** | **作者** | **更改日期** | **备 注** |
| 1 | 创建 | 1.0.0 | 队员C | 2018/11/17 | 初稿 |
| 2 | 更新 | 1.0.1 | 队员C | 2018/11/18 | 整体修改 |
| 3 | 更新 | 1.0.2 | 队员C | 2018/11/19 | 初赛终稿 |
| 4 | 更新 | 1.1.0 | 队员A | 2019/02/28 | 添加前端单元测试和集成测试 |
| 5 | 更新 | 1.1.1 | 队员D | 2019/02/28 | 添加后端单元测试和功能测试 |
| 6 | 更新 | 1.2.0 | 队员B | 2019/04/04 | 添加系统测试 |
| 7 | 更新 | 1.2.1 | 队员B | 2019/04/10 | 添加测试覆盖率图片 |
| 8 | 更新 | 1.2.2 | 队员B | 2019/05/09 | 修缮相关描述 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 引言

## 编写目的

编写本测试文档的核心意义在于明确本项目的测试目标，为项目的测试建立系统的计划以及分析项目的结构进而设计测试过程以及实现方案。

Temage项目的测试可分前端部分和后端部分。每部分的测试又可分为单元测试和集成测试，确保前后端分别能够独立正常运行。该项目还会进行集成前后端的系统测试，确保系统最终集成之后的正常运行。

## 项目背景

Temage智能图文排版项目是一个运用深度学习技术实现自动图文排版的web应用。用户给定文本和图片，即可根据文章的主题，在线智能排版文本与图片，选择合适的风格，生成一篇优美悦目的文章。能够极大地节省文章从排版到发布的时间，提高用户的效率和生产力。

Temage针对当下大众资讯获取、媒体信息传播的新变化，具有很好的应用前景和广大的目标用户群体。

Temage注重用户的体验，旨在为用户提供优美友好的界面和优质流畅的服务。

Temage前端框架为Vue，后端ServerA/B的框架分别为Django和Tornado。

## 术语和缩略语

[1] Vue：Vue是一个构建数据驱动的 web 界面的渐进式框架。Vue.js的目标是通过尽可能简单的 API 实现响应的数据绑定和组合的视图组件。它不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与单文件组件和 Vue生态系统支持的库结合使用时，Vue也完全能够为复杂的单页应用程序提供驱动。

[2] Jest: 一款令人愉悦的JavaScript前端测试框架。

[3] CI: CI的全称是Continuous Integration，表示持续集成。在CI环境中，开发人员将会频繁地向主干提交代码。这些新提交的代码在最终合并到主干前，需要经过编译和自动化测试流进行验证。持续集成过程中很重视自动化测试验证结果，以保障所有的提交在合并主线之后没有问题，或对可能出现的一些问题进行预警。

[4] CD: CD的全称是Continuous Deployment，表示持续部署。在CD环境中，通过自动化的构建、测试和部署循环来快速交付产品。该产品可以进行灰度测试或是直接上线运营。

[5] Express 是一个简洁而灵活的 node.js Web应用框架, 提供了一系列强大特性帮助你创建各种 Web 应用，和丰富的 HTTP 工具。使用 Express 可以快速地搭建一个完整功能的网站。

[6] MockServer: MockerServer可以模拟通过HTTP或HTTPS集成的任何系统（例如服务、web站点等）。当MockServer收到一个请求时，它将与已配置的活动期望相匹配。期望定义了所采取的行动，例如，可以返回响应。

[7] Gitlab: Gitlab是一个用于代码版本管理的开源项目，使用git作为代码管理工具，内置有持续集成/持续部署的服务。

[8] Chromedriver: 可以用来控制Chrome浏览器的开源工具，可用来进行较为复杂的浏览器自动化测试。

[9] Nightwatch: Nightwatch.js是一个自动化的web应用端到端测试框架，用Node.js写成。

## 参考资料

[1] Jest-npm: Available online on https://www.npmjs.com/package/Jest

[2] Gitlab CI/CD: Available online on https://docs.gitlab.com/ee/ci/

[3] Gitlab Auto DevOps: Available online on https://docs.gitlab.com/ee/ci/examples/devops\_and\_game\_dev\_with\_gitlab\_ci\_cd/index.html

[4] Configuration of your jobs with .gitlab-ci.yml: Available online on https://docs.gitlab.com/ee/ci/yaml/README.html

# 测试计划

### 测试策略与目标

测试过程可大致分成两大部分：前端测试以及后端测试。前后端的测试又可细分为单元测试、集成测试。最终该项目还会进行集成前后端的系统测试，确保系统最终集成之后的正常运行。

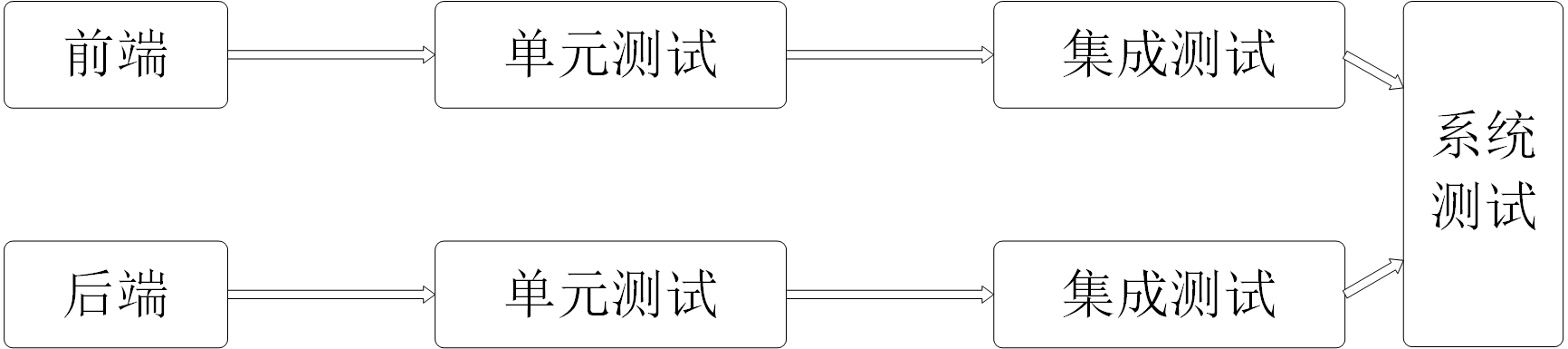


图 2.1 Temage项目测试架构（集成测试和系统测试见 4.功能测试）

前端单元测试主要为编写基本测试用例来检查各项函数的返回值是否正确，此部分测试可以使用测试框架（Jest）进行。为加快前后端开发进程，在前端单元测试中该项目也会使用Express搭建MockServer，模拟后端进行测试。

前端的集成测试要求将通过单元测试的前端各模块聚合起来，配合MockServer，在Nightwatch和chromedriver的模拟环境下进行集成测试，确保各模块之间能够正常协同运作。该项目计划经常性地进行集成测试以确保任何时刻都有一份稳定且版本较新的代码。

后端单元测试的将会使用Django和Tornado自带的单元测试工具。测试策略与前端的代码相似。

后端集成测试将会使用Django的LiveServerTestCase进行，确保前后端能够正确协同工作。

系统测试是检验项目完成度的一大指标，该项目决定使用在Django的LiveServerTestCase中使用selenium模拟操作前端，进行前后端集成的系统测试。

Temage将会在CI/CD架构下进行开发，开发人员在本地的单元测试通过之后，向主代码库提交代码。每次提交均会触发在.gitlab-ci.yml定义的流水线操作。由测试服务器（Gitlab-Runners）拉取代码、自动进行各项测试。若各项测试通过则会向Docker Hub推送新的镜像。服务器拉取新的镜像并部署。最后通知软件测试人员部署结果。若测试出现问题则会以邮件形式提示软件测试人员。



图 2.2 CI/CD架构图

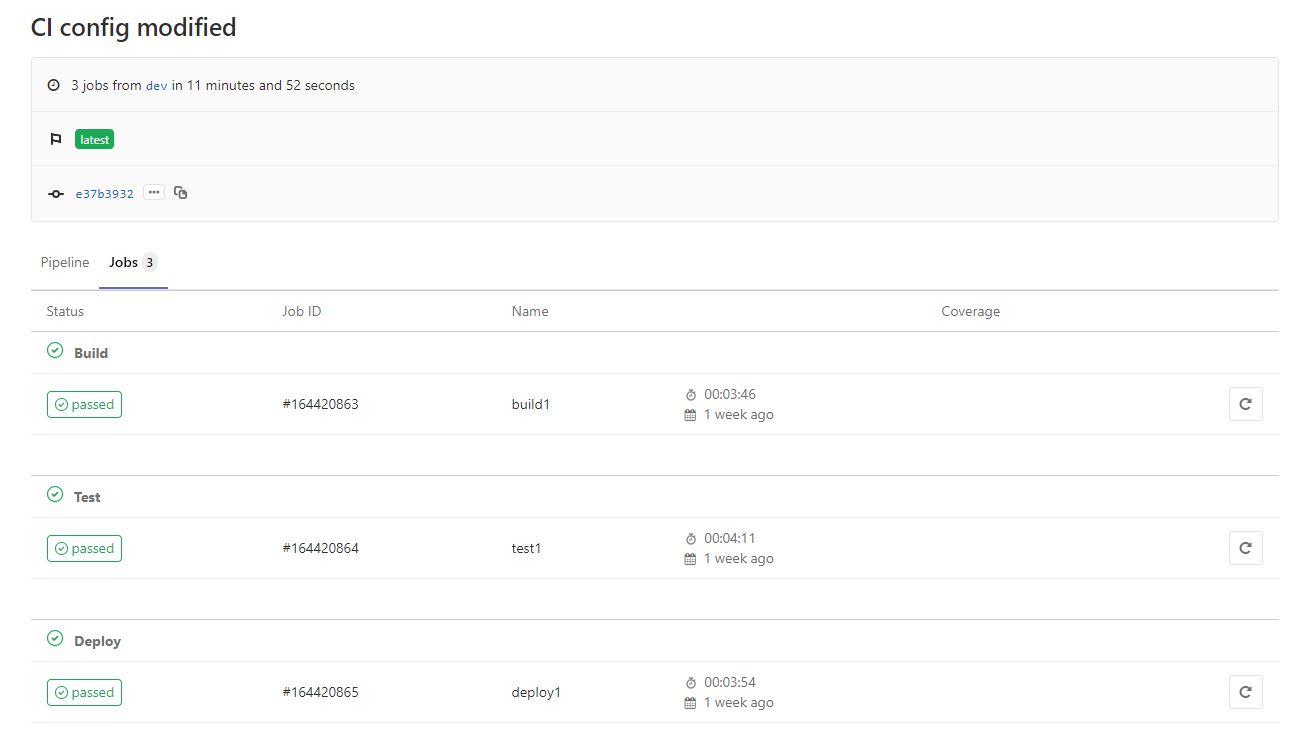


图 2.3 使用Gitlab实现CI/CD

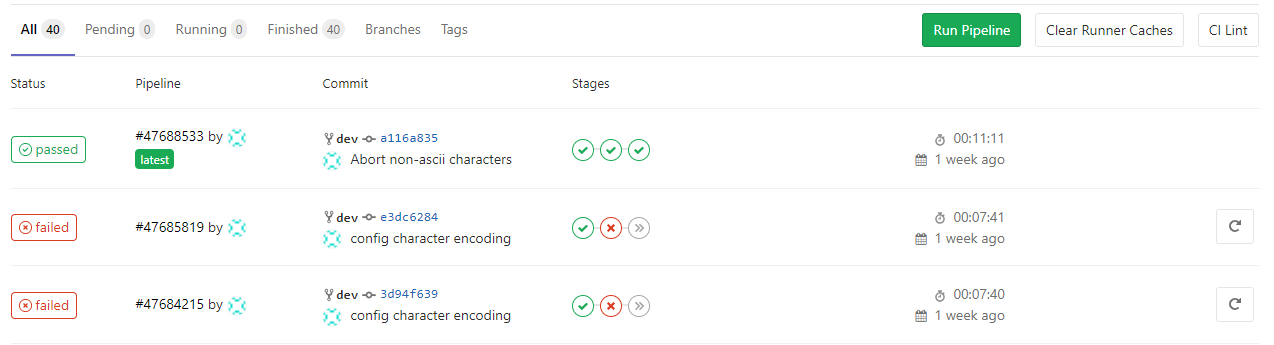


图 2.4 使用Gitlab实现CI/CD

在该架构下，频繁的系统测试和集成测试能加快项目的迭代、更新，在出现问题时也能在早期解决，总体上减少了debug的成本。

项目的测试重点包括：前端页面应能正确解析并显示后端的数据，同时后端应确保业务逻辑的正确性，主要包括用户登录、编辑、推荐、分享等功能。

项目的测试难点包括：该项目选择前后端分离进行开发，则相对应的测试也是较为独立的。该项目建立的MockServer帮助前端独立于后端进行开发，帮助编写单元测试。对此，该项目选择采用Express模拟后端的数据进行测试。在系统各接口均确定之后再进行集成前后端的系统测试。后端的测试将会使用Django的LiveServerTestCase进行单元测试和集成测试。

该项目的测试将分为如下几个层次：



图 2.5 测试分层图（图片来自http://www.anexinet.com/wp-content/uploads/Acceptance.png）

总体上的测试由下至上为单元测试、集成测试、系统测试和用户的接受测试（上线之后）。其中前后端的测试又相对独立。在进行系统测试时则会集成前后端。

该项目的测试目标有如下几点：

1. 单元测试的测试目标首先是验证各个单元的正确性，其次则是检查代码的覆盖率。在保证代码的正确性的前提下，通过编写尽可能多的单元测试来覆盖单元的所有功能。
2. 集成测试的测试目标主要是验证前端或后端的各个模块业务逻辑的正确性。
3. 系统测试的测试目标主要是验证前后端的系统能够正确满足所有需求，其次是系统的各项非功能性需求（如产品的响应时间、产品的兼容性、产品的抗压测试等等）。

### 测试范围

该项目对测试广度的理解为：测试应从尽可能多的角度对产品进行测试。该项目将对前后端分别进行功能性测试，确保使用逻辑的完整与正确；性能测试，保障用户的使用体验；安全测试，防御常见攻击（CSRF等）；易用性测试，确保用户能快速上手；稳定性测试，保障服务抗压能力；兼容性测试，保障web前端在不同主流浏览器上都能正常运行。

该项目对测试深度的理解为：测试应对产品的每一个特性或功能进行尽可能多、尽可能深的测试。通过编写大量测试样例，确保测试将会覆盖前后端的所有代码。每次测试都将会进行代码覆盖率的分析，确保项目的所有特性都能在测试中被覆盖到。

### 测试环境

所有测试将会运行在有英特尔芯片的机器上。其中前端测试运行在Node.js环境中，测试工具为Jest, Express，nightwatch。后端测试运行在Python环境中。测试工具为Django和Tornado自带的单元测试工具。CI/CD平台将使用Gitlab内置的CI/CD服务，在gitlab的共享runner服务器上实现。

# 单元测试

## 图文匹配模块

### 测试用例与执行分析

前端单元测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 001 | | | |
| 测试单元描述 | 测试关键组件exhibition cards渲染 | | | |
| 用例目的 | 验证关键组件exhibition cards渲染正常 | | | |
| 前提条件 | 用户已登录 | | | |
| 特殊的规程说明 | card数据字段与流入数据字段一致, 流入数据有效 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | cards数组, 包含card实例 | exhibition card渲染正确 | exhibition card渲染正确 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 002 | | | |
| 测试单元描述 | 测试关键组件gallery cards渲染 | | | |
| 用例目的 | 验证关键组件gallery cards渲染正常 | | | |
| 前提条件 | 1.用户已登录 | | | |
| 2.画廊数据不为空 | | | |
| 特殊的规程说明 | card数据字段与流入数据字段一致, 流入数据有效 | | | |
| 用例间的依赖关系 | exhibition card渲染正常 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | cards数组, 包含card实例 | gallery card渲染正确 | gallery card渲染正确 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 003 | | | |
| 测试单元描述 | 测试关键组件通知栏渲染 | | | |
| 用例目的 | 验证关键组件通知栏渲染正常 | | | |
| 前提条件 | 1.用户已登录 | | | |
| 2.用户通知不空 | | | |
| 特殊的规程说明 | 通知栏数据字段与流入数据字段一致, 流入数据有效 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 通知类数组, 包含通知实例 | 通知栏渲染正确 | 通知栏渲染正确 | 无 |

后端单元测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 101 | | | |
| 测试单元描述 | 测试Models的多态多对多关系 | | | |
| 用例目的 | 测试model多态多对多关系能否正常工作 | | | |
| 前提条件 | 通过数据库操作创建相应的带有多态多对多关系的实例 | | | |
| 特殊的规程说明 | Profile与Product的对象可以同时拥有同一个Theme，且它们分别与Theme成多对多关系 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 无 | user和第一个product拥有同一个theme | 两者的theme相等 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 102 | | | |
| 测试单元描述 | 登录功能测试 | | | |
| 用例目的 | 测试login登录功能 | | | |
| 前提条件 | 在数据库中创建相关实例 | | | |
| 特殊的规程说明 | 在数据库中存入username="user1", password="123"的用户 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | {username:"user1", password:"123"}请求 | 输出的status\_code = 200 | “200” | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 107 | | | |
| 测试单元描述 | 测试/api/user/register/接口 | | | |
| 用例目的 | 测试用户注册功能 | | | |
| 前提条件 | 通过数据库操作创建相应的对象们 | | | |
| 特殊的规程说明 | 无 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 无 | 注册成功状态码“200” | 200 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 111 | | | |
| 测试单元描述 | 测试/api/workflow/finished\_work接口 | | | |
| 用例目的 | 测试/api/workflow/finished\_work返回预览信息是否正确 | | | |
| 前提条件 | 通过数据库操作创建相应的实例 | | | |
| 特殊的规程说明 | 需登陆后操作 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 无 | 登录后response中带有身份象征的token，token解码后id为2 | id为2 | 根据setup的改动，id可能有所变动 |
| 2 | 带有步骤1中token的request，和当前处理的product的id | 返回作品的html文件路径，和生成图片的宽度 | 作品html的url和宽度width=400 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 112 | | | |
| 测试单元描述 | 测试/api/workflow/download\_picture接口 | | | |
| 用例目的 | 测试/api/workflow/download\_picture返回的下载信息是否正确 | | | |
| 前提条件 | 通过数据库操作创建相应的实例 | | | |
| 特殊的规程说明 | 需登陆后操作 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 无 | 登录后response中带有身份象征的token，token解码后id为2 | id为2 | 根据setup的改动，id可能有所变动 |
| 2 | 带有步骤1中token的request，和需要下载的product的id | 返回下载地址 | 返回下载地址 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 113 | | | |
| 测试单元描述 | 测试/api/workflow/confirm\_store接口 | | | |
| 用例目的 | 测试/api/workflow/confirm\_store的操作是否成功 | | | |
| 前提条件 | 通过数据库操作创建相应的实例 | | | |
| 特殊的规程说明 | 需登陆后操作 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 无 | 登录后response中带有身份象征的token，token解码后id为2 | id为2 | 根据setup的改动，id可能有所变动 |
| 2 | 带有步骤1中token的request，还包含最终保存的product的id和用户评分stars | 返回成功状态码200 | “200” | 无 |

### 分析结果：

|  |  |  |  |
| --- | --- | --- | --- |
| 图文匹配单元测试报告 | | | |
| statements | branches | functions | lines |
| 86.41% | 70.88% | 84.60% | 79.05% |

### 测试结果综合分析及建议

1. 提高代码覆盖率
2. 注意测试用例之间的依赖关系

### 测试经验总结

测试需要包含每一个单元，需要对每一个接口进行测试，MockServer作为中间媒介，一方面使前端能进行独立的测试与运行，另一方面给后端清晰表明具体的开发任务，减少了依赖。

## 用户社区模块

### 测试用例与执行分析

单元测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 001 | | | |
| 测试单元描述 | 测试关键组件exhibition cards渲染 | | | |
| 用例目的 | 验证关键组件exhibition cards渲染正常 | | | |
| 前提条件 | 用户已登录 | | | |
| 特殊的规程说明 | card数据字段与流入数据字段一致, 流入数据有效 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | cards数组, 包含card实例 | exhibition card渲染正确 | exhibition card渲染正确 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 002 | | | |
| 测试单元描述 | 测试关键组件gallery cards渲染 | | | |
| 用例目的 | 验证关键组件gallery cards渲染正常 | | | |
| 前提条件 | 1.用户已登录 | | | |
| 2.画廊数据不为空 | | | |
| 特殊的规程说明 | card数据字段与流入数据字段一致, 流入数据有效 | | | |
| 用例间的依赖关系 | exhibition card渲染正常 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | cards数组, 包含card实例 | gallery card渲染正确 | gallery card渲染正确 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 003 | | | |
| 测试单元描述 | 测试关键组件通知栏渲染 | | | |
| 用例目的 | 验证关键组件通知栏渲染正常 | | | |
| 前提条件 | 1.用户已登录 | | | |
| 2.用户通知不空 | | | |
| 特殊的规程说明 | 通知栏数据字段与流入数据字段一致, 流入数据有效 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 通知类数组, 包含通知实例 | 通知栏渲染正确 | 通知栏渲染正确 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 004 | | | |
| 测试单元描述 | 工作流初始状态测试 | | | |
| 用例目的 | 验证工作流初始状态正常 | | | |
| 前提条件 | 1.用户已登录 | | | |
| 2.工作流内为空 | | | |
| 特殊的规程说明 | 无 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 无 | 工作流初始状态正常 | 工作流初始状态正常 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 005 | | | |
| 测试单元描述 | 工作流第一步状态推进测试 | | | |
| 用例目的 | 验证工作流第一步状态推进正常 | | | |
| 前提条件 | 1.用户已登录 | | | |
| 2.工作流初始状态正常 | | | |
| 特殊的规程说明 | 无 | | | |
| 用例间的依赖关系 | 测试004通过 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 触发下一步按钮 | 工作流推进到状态1 | 工作流推进到状态1 | 不涉及该部分接口测试 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 006 | | | |
| 测试单元描述 | 工作流第二步状态推进测试 | | | |
| 用例目的 | 验证工作流第二步状态推进正常 | | | |
| 前提条件 | 1.用户已登录 | | | |
| 2.工作流第一步状态推进正常 | | | |
| 特殊的规程说明 | 无 | | | |
| 用例间的依赖关系 | 测试005通过 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 触发下一步按钮 | 工作流推进到状态2 | 工作流推进到状态2 | 不涉及该部分接口测试 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 007 | | | |
| 测试单元描述 | 工作流第三步状态推进测试 | | | |
| 用例目的 | 验证工作流第三步状态推进正常 | | | |
| 前提条件 | 1.用户已登录 | | | |
| 2.工作流第二步状态推进正常 | | | |
| 特殊的规程说明 | 无 | | | |
| 用例间的依赖关系 | 测试006通过 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 触发下一步按钮 | 工作流推进到状态3 | 工作流推进到状态3 | 不涉及该部分接口测试 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 008 | | | |
| 测试单元描述 | 工作流异常按钮触发测试 | | | |
| 用例目的 | 验证工作流异常按钮触发后状态正常 | | | |
| 前提条件 | 1.用户已登录 | | | |
| 2.工作流处于初始状态 | | | |
| 特殊的规程说明 | 无 | | | |
| 用例间的依赖关系 | 测试004通过 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 上一步按钮 | 工作流状态依然位于初始值 | 工作流状态依然位于初始值 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 101 | | | |
| 测试单元描述 | 测试Models的多态多对多关系 | | | |
| 用例目的 | 测试model多态多对多关系能否正常工作 | | | |
| 前提条件 | 通过数据库操作创建相应的带有多态多对多关系的实例 | | | |
| 特殊的规程说明 | Profile与Product的对象可以同时拥有同一个Theme，且它们分别与Theme成多对多关系 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 无 | user和第一个product拥有同一个theme | 两者的theme相等 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 102 | | | |
| 测试单元描述 | 登录功能测试 | | | |
| 用例目的 | 测试login登录功能 | | | |
| 前提条件 | 在数据库中创建相关实例 | | | |
| 特殊的规程说明 | 在数据库中存入username="user1", password="123"的用户 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | {username:"user1", password:"123"}请求 | 输出的status\_code = 200 | “200” | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 103 | | | |
| 测试单元描述 | 测试/api/explore/接口 | | | |
| 用例目的 | 测试给前端/api/explore/返回的数据是否符合规范 | | | |
| 前提条件 | 通过数据库操作创建相应的实例 | | | |
| 特殊的规程说明 | 用户需要先登录才能获取这些信息 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 无 | 登录后response中带有身份象征的token，token解码后id为2 | id为2 | 根据setup的改动，id可能有所变动 |
| 2 | 带有步骤1中token的request | 有四个元素（四种信息）的list | 有四个元素（四种信息）的list | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 104 | | | |
| 测试单元描述 | 测试/api/explore/product接口 | | | |
| 用例目的 | 测试给前端/api/work返回的数据是否符合规范 | | | |
| 前提条件 | 通过数据库操作创建相应的实例 | | | |
| 特殊的规程说明 | 用户需要先登录才能获取这些信息 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 无 | 登录后response中带有身份象征的token，token解码后id为2 | id为2 | 根据setup的改动，id可能有所变动 |
| 2 | 带有步骤1中token的request，work的id | product所需的四种信息 | product所需的四种信息 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 105 | | | |
| 测试单元描述 | 测试api接口 | | | |
| 用例目的 | 测试给前端/api/explore/collection返回的数据是否符合规范 | | | |
| 前提条件 | 通过数据库操作创建相应的实例 | | | |
| 特殊的规程说明 | 用户需要先登录才能获取这些信息 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 无 | 登录后response中带有身份象征的token，token解码后id为2 | id为2 | 根据setup的改动，id可能有所变动 |
| 2 | 带有步骤1中token的request | collection中的六张卡片内容 | collection中的六张卡片内容 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 106 | | | |
| 测试单元描述 | 测试/api/explore/get\_recent/接口 | | | |
| 用例目的 | 测试给前端/api/explore/get\_recent/返回的数据是否符合规范 | | | |
| 前提条件 | 通过数据库操作创建相应的实例 | | | |
| 特殊的规程说明 | 用户需要先登录才能获取这些信息 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 无 | 登录后response中带有身份象征的token，token解码后id为2 | id为2 | 根据setup的改动，id可能有所变动 |
| 2 | 带有步骤1中token的request | 最近修改的四个work的信息 | 最近修改的四个work的信息 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 107 | | | |
| 测试单元描述 | 测试/api/user/register/接口 | | | |
| 用例目的 | 测试用户注册功能 | | | |
| 前提条件 | 通过数据库操作创建相应的对象们 | | | |
| 特殊的规程说明 | 无 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 无 | 注册成功状态码“200” | 200 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 108 | | | |
| 测试单元描述 | 测试/api/explore/gallery/接口 | | | |
| 用例目的 | 测试/api/explore/gallery/返回的数据是否符合规范 | | | |
| 前提条件 | 通过数据库操作创建相应的实例 | | | |
| 特殊的规程说明 | 需登陆后操作 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 无 | 登录后response中带有身份象征的token，token解码后id为2 | id为2 | 根据setup的改动，id可能有所变动 |
| 2 | 带有步骤1中token的request | 返回相应gallery数据 | 返回相应gallery数据 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 109 | | | |
| 测试单元描述 | 测试/api/explore/gallery/more\_cards接口 | | | |
| 用例目的 | 测试/api/gallery/more\_cards返回的数据是否符合规范 | | | |
| 前提条件 | 通过数据库操作创建相应的实例 | | | |
| 特殊的规程说明 | 需登陆后操作 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 无 | 登录后response中带有身份象征的token，token解码后id为2 | id为2 | 根据setup的改动，id可能有所变动 |
| 2 | 带有步骤1中token的request | 随机返回4条gallery数据 | 随机返回4条gallery数据 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 114 | | | |
| 测试单元描述 | 测试/api/explore/delete接口 | | | |
| 用例目的 | 测试/api/explore/delete的对product的删除操作是否成功 | | | |
| 前提条件 | 通过数据库操作创建相应的实例 | | | |
| 特殊的规程说明 | 需登陆后操作 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 无 | 登录后response中带有身份象征的token，token解码后id为2 | id为2 | 根据setup的改动，id可能有所变动 |
| 2 | 带有步骤1中token的request，还包含需要删除的product的id | 返回成功状态码200 | “200” | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 115 | | | |
| 测试单元描述 | 测试/api/explore/post\_collect/接口 | | | |
| 用例目的 | 测试/api/explore/post\_collect/的加入收藏夹操作是否成功 | | | |
| 前提条件 | 通过数据库操作创建相应的实例 | | | |
| 特殊的规程说明 | 需登陆后操作 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 无 | 登录后response中带有身份象征的token，token解码后id为2 | id为2 | 根据setup的改动，id可能有所变动 |
| 2 | 带有步骤1中token的request，还包含需要加入收藏夹的product的id | 返回成功状态码200 | “200” | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 116 | | | |
| 测试单元描述 | 测试/api/explore/cancel\_collect/接口 | | | |
| 用例目的 | 测试/api/explore/cancel\_collect/从收藏夹中取消收藏某个product的操作是否成功 | | | |
| 前提条件 | 通过数据库操作创建相应的实例 | | | |
| 特殊的规程说明 | 需登陆后操作 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 无 | 登录后response中带有身份象征的token，token解码后id为2 | id为2 | 根据setup的改动，id可能有所变动 |
| 2 | 带有步骤1中token的request，还包含需要从收藏夹取消的product的id | 返回成功状态码200 | “200” | 无 |

集成测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 201 | | | |
| 测试单元描述 | 测试关键组件exhibition cards渲染 | | | |
| 用例目的 | 验证关键组件exhibition cards渲染正常 | | | |
| 前提条件 | 用户已登录 | | | |
| 特殊的规程说明 | card数据字段与流入数据字段一致, 流入数据有效 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | cards数组, 包含card实例 | exhibition card渲染正确 | exhibition card渲染正确 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 202 | | | |
| 集成测试描述 | 模拟用户登录行为 | | | |
| 用例目的 | 验证前端注册过程正确性, 接口/login/submit/正常运行 | | | |
| 前提条件 | 1.浏览器Cookies含有登录用户信息; | | | |
| 2.用户信息已经被注册; | | | |
| 3.登录字段按需求正确填写 | | | |
| 特殊的规程说明 | 用户名与密码字段要求与数据库内信息一致 | | | |
| 用例间的依赖关系 | 进行注册后再用用户名和密码字段进行登录 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 用户名: username | validate正常 | validate正常 | 无 |
| 2 | 用户密码: 123 | 用户密码: 123 | 用户密码: 123 | 无 |
| 3 | 点击登录按钮, 提交表单 | 页面跳转至用户主页 | 页面跳转至用户主页 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 203 | | | |
| 集成测试描述 | 模拟用户收藏行为 | | | |
| 用例目的 | 验证前端收藏过程正确性, 接口/api/collect/正常运行 | | | |
| 前提条件 | 1.浏览器Cookies含有登录用户信息; | | | |
| 2.被收藏文章作者不是收藏执行用户; | | | |
| 3.被收藏文章不再用户收藏夹中 | | | |
| 特殊的规程说明 | 必须首先执行登录, 保证用户Cookies被储存 | | | |
| 用例间的依赖关系 | 登录成功测试后执行本用例 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 用户名:username; 密码:123; 登录操作 | 登陆成功, 跳转到主页 | 登陆成功, 跳转到主页 | 无 |
| 2 | 1号文章地址 | 跳转到该文章 | 跳转到该文章 | 无 |
| 3 | 收藏操作 | 提示收藏成功 | 提示收藏成功 | 无 |

### 分析结果：

|  |  |  |  |
| --- | --- | --- | --- |
| 用户社区单元测试报告 | | | |
| statements | branches | functions | lines |
| 72.73% | 76.47% | 78.95% | 85.71% |

### 测试结果综合分析及建议

测试数量和测试覆盖率还需进一步的加强。

### 测试经验总结

测试需要包含每一个单元，需要对每一个接口进行测试，MockServer作为中间媒介，一方面使前端能进行独立的测试与运行，另一方面给后端清晰表明具体的开发任务，减少了依赖。

# 功能测试

## 图文匹配功能

### 测试用例与执行分析

测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 201 | | | |
| 测试单元描述 | 模拟用户注册 | | | |
| 用例目的 | 验证前端注册过程正确性, 接口/api/register/正常运行 | | | |
| 前提条件 | 1.浏览器Cookies不含有登录用户信息; | | | |
| 2.用户名未被注册; | | | |
| 3.注册其他字段按需求完整填写 | | | |
| 特殊的规程说明 | 注册时validate操作需要各个字段按要求完整填写 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 用户名: name1 | validate正常 | validate正常 | 无 |
| 2 | 用户密码: 123 | validate正常 | validate正常 | 无 |
| 3 | 电子邮件: 123@sample.com | validate正常 | validate正常 | 无 |
| 4 | 个人描述: describe | validate正常 | validate正常 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 202 | | | |
| 集成测试描述 | 模拟用户登录行为 | | | |
| 用例目的 | 验证前端注册过程正确性, 接口/login/submit/正常运行 | | | |
| 前提条件 | 1.浏览器Cookies含有登录用户信息; | | | |
| 2.用户信息已经被注册; | | | |
| 3.登录字段按需求正确填写 | | | |
| 特殊的规程说明 | 用户名与密码字段要求与数据库内信息一致 | | | |
| 用例间的依赖关系 | 进行注册后再用用户名和密码字段进行登录 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 用户名: username | validate正常 | validate正常 | 无 |
| 2 | 用户密码: 123 | 用户密码: 123 | 用户密码: 123 | 无 |
| 3 | 点击登录按钮, 提交表单 | 页面跳转至用户主页 | 页面跳转至用户主页 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 203 | | | |
| 集成测试描述 | 模拟用户收藏行为 | | | |
| 用例目的 | 验证前端收藏过程正确性, 接口/api/collect/正常运行 | | | |
| 前提条件 | 1.浏览器Cookies含有登录用户信息; | | | |
| 2.被收藏文章作者不是收藏执行用户; | | | |
| 3.被收藏文章不再用户收藏夹中 | | | |
| 特殊的规程说明 | 必须首先执行登录, 保证用户Cookies被储存 | | | |
| 用例间的依赖关系 | 登录成功测试后执行本用例 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 用户名:username; 密码:123; 登录操作 | 登陆成功, 跳转到主页 | 登陆成功, 跳转到主页 | 无 |
| 2 | 1号文章地址 | 跳转到该文章 | 跳转到该文章 | 无 |
| 3 | 收藏操作 | 提示收藏成功 | 提示收藏成功 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 102 | | | |
| 功能描述 | 集成测试django后端workflow模块功能 | | | |
| 用例目的 | 测试workflow模块是否正常工作 | | | |
| 前提条件 | 数据库创建相应实例（用户、收藏夹、缓存等对象） | | | |
| 特殊的规程说明 | 无 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 动作 | 期望结果 | 实际结果 | 备注 |
| 1 | 输入用户名、密码登录网站 | 输出状态码“200”，表示成功 | “200” | 无 |
| 2 | 给出必要信息初步存储作品 | 输出状态码“200”，表示成功 | “200” | 无 |
| 3 | 用户作品最终效果预览 | 输出状态码“200”，表示成功 | “200” | 无 |
| 4 | 用户最后评分，相关信息存入作品 | 输出状态码“200”，表示成功 | “200” | 无 |

### 分析结果：

|  |  |  |  |
| --- | --- | --- | --- |
| 图文匹配功能测试报告 | | | |
| statements | branches | functions | lines |
| 76.92% | 50% | 64.30% | 63.16% |

### 测试结果综合分析及建议

测试全部通过。

### 测试经验总结

功能测试需要把单元测试中的各个部分串联起来，将各个子模块组合成为功能，MockServer作为中间媒介，一方面使前端能进行独立的测试与运行，另一方面给后端清晰表明具体的开发任务，减少了依赖。

## 用户社区功能

### 测试用例与执行分析

测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 201 | | | |
| 测试单元描述 | 模拟用户注册 | | | |
| 用例目的 | 验证前端注册过程正确性, 接口/api/register/正常运行 | | | |
| 前提条件 | 1.浏览器Cookies不含有登录用户信息; | | | |
| 2.用户名未被注册; | | | |
| 3.注册其他字段按需求完整填写 | | | |
| 特殊的规程说明 | 注册时validate操作需要各个字段按要求完整填写 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 用户名: name1 | validate正常 | validate正常 | 无 |
| 2 | 用户密码: 123 | validate正常 | validate正常 | 无 |
| 3 | 电子邮件: 123@sample.com | validate正常 | validate正常 | 无 |
| 4 | 个人描述: describe | validate正常 | validate正常 | 无 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 202 | | | |
| 集成测试描述 | 模拟用户登录行为 | | | |
| 用例目的 | 验证前端注册过程正确性, 接口/login/submit/正常运行 | | | |
| 前提条件 | 1.浏览器Cookies含有登录用户信息; | | | |
| 2.用户信息已经被注册; | | | |
| 3.登录字段按需求正确填写 | | | |
| 特殊的规程说明 | 用户名与密码字段要求与数据库内信息一致 | | | |
| 用例间的依赖关系 | 进行注册后再用用户名和密码字段进行登录 | | | |
| 步骤 | 输入 | 期望输出 | 实际输出 | 备注 |
| 1 | 用户名: username | validate正常 | validate正常 | 无 |
| 2 | 用户密码: 123 | 用户密码: 123 | 用户密码: 123 | 无 |
| 3 | 点击登录按钮, 提交表单 | 页面跳转至用户主页 | 页面跳转至用户主页 | 无 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 用例编号 | | 203 | | | | | | |
| 集成测试描述 | | 模拟用户收藏行为 | | | | | | |
| 用例目的 | | 验证前端收藏过程正确性, 接口/api/collect/正常运行 | | | | | | |
| 前提条件 | | 1.浏览器Cookies含有登录用户信息; | | | | | | |
| 2.被收藏文章作者不是收藏执行用户; | | | | | | |
| 3.被收藏文章不再用户收藏夹中 | | | | | | |
| 特殊的规程说明 | | 必须首先执行登录, 保证用户Cookies被储存 | | | | | | |
| 用例间的依赖关系 | | 登录成功测试后执行本用例 | | | | | | |
| 步骤 | | 输入 | | 期望输出 | | 实际输出 | | 备注 |
| 1 | | 用户名:username; 密码:123; 登录操作 | | 登陆成功, 跳转到主页 | | 登陆成功, 跳转到主页 | | 无 |
| 2 | | 1号文章地址 | | 跳转到该文章 | | 跳转到该文章 | | 无 |
| 3 | | 收藏操作 | | 提示收藏成功 | | 提示收藏成功 | | 无 |
| 用例编号 | 301 | | | | | | | |
| 功能描述 | 集成测试django后端explore模块功能 | | | | | | | |
| 用例目的 | 测试explore模块是否正常工作 | | | | | | | |
| 前提条件 | 无 | | | | | | | |
| 特殊的规程说明 | 无 | | | | | | | |
| 用例间的依赖关系 | 无 | | | | | | | |
| 步骤 | 动作 | | 期望结果 | | 实际结果 | | 备注 | |
| 1 | 输入相关信息注册新用户 | | 输出状态码“200”，表示成功 | | “200” | | 无 | |
| 2 | 输入用户名、密码登录 | | 输出状态码“200”，表示成功 | | “200” | | 无 | |
| 3 | 访问/api/explore/页面获取首页信息 | | 输出状态码“200”，表示成功 | | “200” | | 无 | |
| 4 | 访问/api/explore/gallery/获取游廊信息 | | 输出状态码“200”，表示成功 | | “200” | | 无 | |
| 5 | 获取更多游廊信息 | | 输出状态码“200”，表示成功 | | “200” | | 无 | |
| 6 | 访问/api/explore/collection/获取收藏夹信息 | | 输出状态码“200”，表示成功 | | “200” | | 无 | |
| 7 | 访问/api/explore/get\_recent/获取最近作品信息 | | 输出状态码“200”，表示成功 | | “200” | | 无 | |
| 8 | 获取指定作品的完整信息 | | 输出状态码“200”，表示成功 | | “200” | | 无 | |
| 9 | 将指定作品加入收藏夹 | | 输出状态码“200”，表示成功 | | “200” | | 无 | |
| 10 | 从收藏夹中取消收藏指定作品 | | 输出状态码“200”，表示成功 | | “200” | | 无 | |
| 11 | 从用户下删除指定作品 | | 输出状态码“200”，表示成功 | | “200” | | 无 | |

### 结果分析

|  |  |  |  |
| --- | --- | --- | --- |
| 图文匹配功能测试报告 | | | |
| statements | branches | functions | lines |
| 66.67% | 71.43% | 68.18% | 62.50% |

### 测试结果综合分析及建议

测试全部通过，暂且只包含重要功能测试，建议进一步添加所用功能测试。

### 测试经验总结

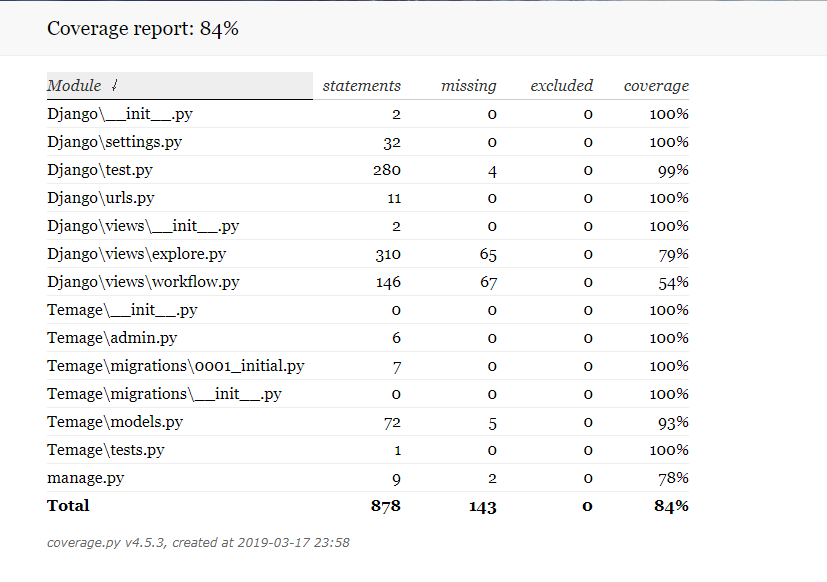
 功能测试需要把单元测试中的各个部分串联起来，将各个子模块组合成为功能，MockServer作为中间媒介，一方面使前端能进行独立的测试与运行，另一方面给后端清晰表明具体的开发任务，减少了依赖。

图 4.1 后端测试覆盖率报告

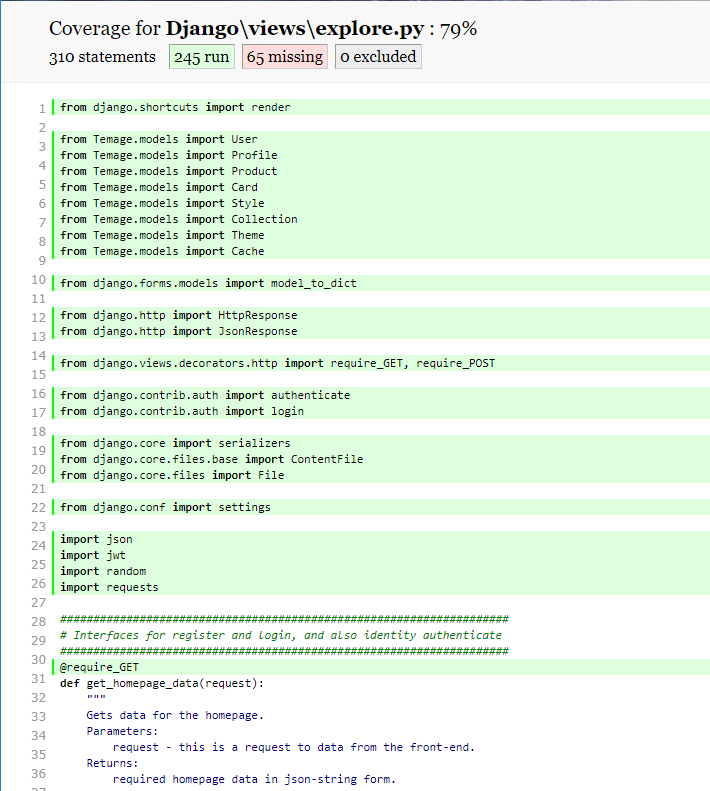


图 4.2 后端测试覆盖率详情报告

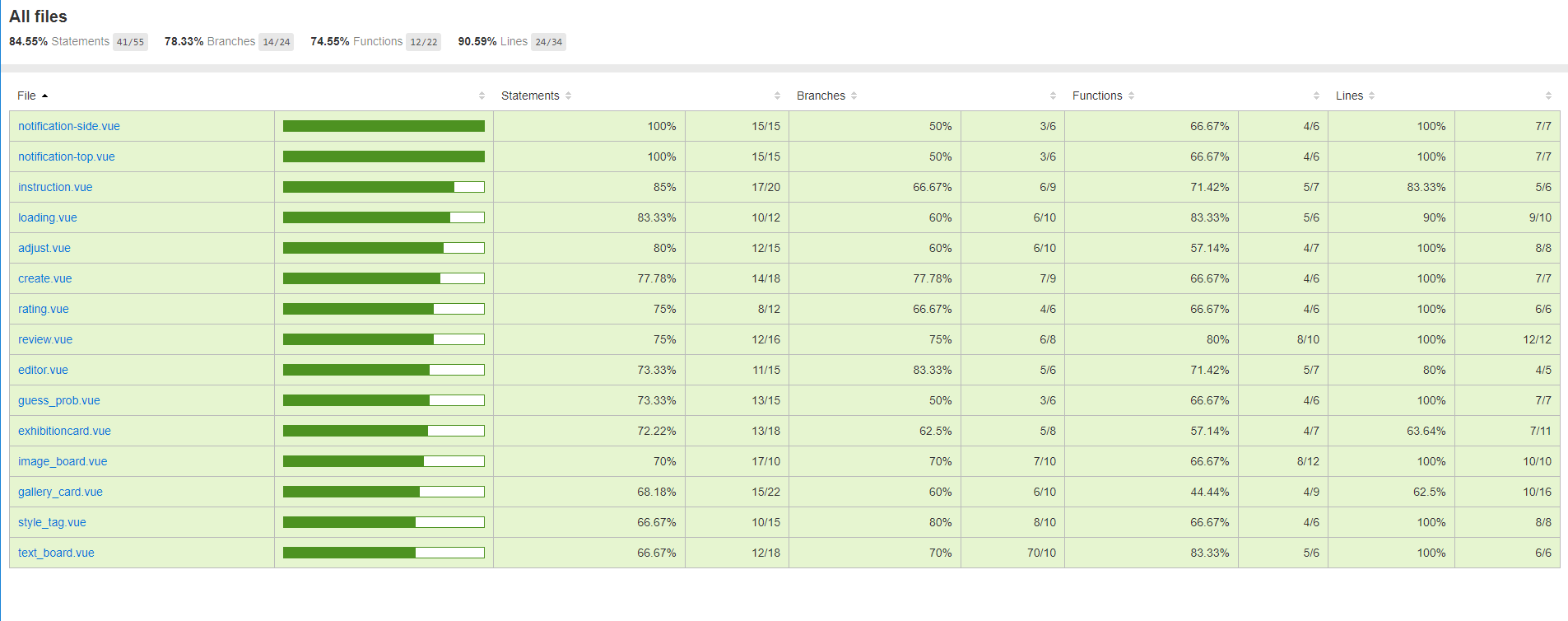


图 4.3 前端覆盖率测试报告

详细的测试报告见repo下的coverage文件夹。

# 系统测试

系统测试分为两大部分，一部分是模型性能测试，一部分为压力测试，Temage使用JMeter来进行系统测试，详细的JMeter测试文件见repo的JMeter文件夹。

在模型性能测试中，Temage主要观察和测试深度学习模型在并发数量较少的情况下调用的性能，并以此进行改进。

在压力测试中，Temage主要观察和测试各核心模块在并发数量较大的情况下的性能，并以此进行改进。

## 模型性能测试

### 测试用例与执行分析

### 文本分类预期性能指标测试：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 001 | | | |
| 性能描述 | 用户上传图片与文本后，获取文本分类 | | | |
| 用例目的 | 用户上传图片与文本后，获取文本分类 | | | |
| 前提条件 | 用户已登录 | | | |
| 特殊的规程说明 | 无 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入/动作 | 期望的性能  (平均值) | 实际性能  (平均值) | 备注 |
| 1 | 输入文本 | 200ms | 183ms |  |
| 2 | 输入图片 | 3000ms | 3466ms | 图片总量为4张，大小为10MB |
| 3 | 获取文本嵌入 | 8000ms | 6583ms |  |
| 4 | 输出文本分类 | 4000ms | 4522ms |  |

### 分析结果：

文本分类和获取文本嵌入需要的时间较多，基本能够达标，文本与图片上传的瓶颈在于带宽。

### 图文匹配预期性能指标测试：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 002 | | | |
| 性能描述 |  | | | |
| 用例目的 | 获取最佳图文匹配 | | | |
| 前提条件 | 用户登录并完成文本分类 | | | |
| 特殊的规程说明 | 无 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入/动作 | 期望的性能  (平均值) | 实际性能  (平均值) | 备注 |
| 1 | 向server发送相关材料 | 10000ms | 12830ms | 发送图片文字，获取匹配方式。 |
| 2 | 根据匹配方式组织图文排版 | 300ms | 413ms |  |

### 分析结果：

图文匹配的瓶颈在于serverB的处理速度，serverB的性能不容乐观。

### 预期性能指标测试：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 003 | | | |
| 性能描述 |  | | | |
| 用例目的 | 通过用户关键字获取相关作品 | | | |
| 前提条件 | 用户登录 | | | |
| 特殊的规程说明 | 无 | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 输入/动作 | 期望的性能  (平均值) | 实际性能  (平均值) | 备注 |
| 1 | 发送关键字 | 200ms | 183ms |  |
| 2 | 向ElasticSearch发送请求 | 600ms | 431ms | 向ElasticSearch发送请求，ElasticSearch中模拟数据有200000条 |
| 3 | 返回相关内容 | 400ms | 421ms | 从数据库中获取相应内容 |

### 分析结果：

只能搜索功能性能良好。

### 测试结果综合分析及建议

有关深度学习的相关服务的性能还有待提高，项目带宽也应当在有条件的条件下适当增加。

### 测试经验总结

深度学习模型有极高的性能需求，Temage通过在保证一定accuracy的条件下，减少神经元数量。

## 压力测试

### 测试用例与执行分析

### 图文匹配压力测试用例：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 功能 | 图文匹配 | | | | | |
| 目的 | 获取图文匹配 | | | | | |
| 方法 | 发送图片和文本，使用双分支模型进行图文匹配 | | | | | |
| 并发用户数与事务执行情况 | | | | | | |
| 并发用户数 | 事务平均响应时间 | 事务最大响应时间 | 平均每秒处理事务数 | 事务成功率 | 每秒点击率 | 平均流量(字节/秒) |
| 20 | 18485ms | 22182ms | 0.05 | 100.0% | 20 | 139059 |
| 25 | 19769ms | 26182ms | 0.05 | 100.0% | 25 | 341504 |
| 40 | 22741ms | 32738ms | 0.04 | 100.0% | 40 | 759398 |
| 50 | 24558ms | 46837ms | 0.04 | 98.0% | 50 | 1183744 |
| 60 | 25477ms | 54892ms | 0.04 | 98.3% | 60 | 1704960 |
| 70 | 26002ms | 54800ms | 0.04 | 97.1% | 70 | 2229654 |
| 100 | 26822ms | 55470ms | 0.04 | 98.0% | 100 | 2655488 |
| 200 | 1235ms | 55996ms | -- | 13.0% | 200 | 3044557 |

### 注：

为一个用户进行一次图文排版服务的操作作为一次事务。

### 分析结果：

通过图表可知，随着用户并发数上升，平均响应时间总体呈上升趋势，在50-70用户并发数测试阶段，上升趋势趋缓，且在70时有下降趋势，这是因为此时CPU使用率达到了80%，触发了弹性伸缩，创建了更多了Pod参与到服务中，当用户并发数达到200时，触发熔断机制，响应时间骤减，除去JMeter Ramp-Up Period时处理了少部分请求外，其余请求直接status返回500，因而平均响应时间极短，事务最大响应时间极长。

### 智能搜索压力测试用例：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 功能 | 智能搜索 | | | | | |
| 目的 | 通过用户输入获取相关作品内容 | | | | | |
| 方法 | 发送用户输入，使用ElasticSearch进行搜索 | | | | | |
| 并发用户数与事务执行情况 | | | | | | |
| 并发用户数 | 事务平均响应时间 | 事务最大响应时间 | 平均每秒处理事务数 | 事务成功率 | 每秒点击率 | 平均流量(字节/秒) |
| 20 | 15ms | 27ms | 66.6 | 100% | 20 | 9922 |
| 25 | 15ms | 27ms | 66.6 | 100% | 25 | 12298 |
| 40 | 14ms | 25ms | 71.4 | 100% | 40 | 19251 |
| 50 | 15ms | 28ms | 66.6 | 100% | 50 | 24005 |
| 60 | 15ms | 29ms | 66.6 | 100% | 60 | 28547 |
| 70 | 17ms | 30ms | 58.8 | 100% | 70 | 34564 |
| 100 | 16ms | 29ms | 62.5 | 100% | 100 | 50148 |
| 200 | 17ms | 29ms | 58.8 | 100% | 200 | 98224 |
| 2000 (持续十轮) | 3ms | 21140ms | -- | 81.1% | 2000 | 316917 |

### 分析结果：

通过图表可知，随着用户并发数上升，平均响应时间并沒有明显的上升趋势。整体趋于稳定，这主要得益于ElasticSearch的分片与集群管理，能够有效的处理高强度的搜索。

当然，在相对极限压力（每秒2000并发用户，持续10秒）下，限于硬件的原因，服务还是会出现错误。同时该极限压力测试结束之后，服务能够自动恢复正常，对正常压力下的请求能够正常响应。

### 文本分类压力测试用例：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 功能 | 文本分类 | | | | | |
| 目的 | 获取文本分类的结果 | | | | | |
| 方法 | 发送文本获取文本嵌入，使用TextCNN进行文本分类 | | | | | |
| 并发用户数与事务执行情况 | | | | | | |
| 并发用户数 | 事务平均响应时间 | 事务最大响应时间 | 平均每秒处理事务数 | 事务成功率 | 每秒点击率 | 平均流量(字节/秒) |
| 20 | 8362ms | 10254ms | 0.10 | 95.0% | 20 | 32143 |
| 25 | 8556ms | 10122ms | 0.10 | 92.0% | 25 | 38470 |
| 40 | 8919ms | 11932ms | 0.09 | 85.0% | 40 | 54778 |
| 50 | 9001ms | 11569ms | 0.10 | 90.0% | 50 | 65552 |
| 60 | 9033ms | 12569ms | 0.10 | 86.7% | 60 | 69824 |
| 70 | 10851ms | 13771ms | 0.09 | 70.0% | 70 | 82436 |
| 100 | 12022ms | 14700ms | 0.08 | 63.0% | 100 | 143858 |
| 200 | 13022ms | 18512ms | 0.05 | 50.0% | 200 | 26271 |

### 分析结果:

与图文匹配压力测试相似，在压力达到某临界值时会触发集群中的“熔断机制”。

### 图文匹配（A）、文本分类（B）、智能搜索（C）组合压力测试用例：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 功能 | 用户进行文本分类、图文匹配、智能搜索 | | | | | | | | | | | | | |
| 目的 | 模拟用户使在线使用Temage | | | | | | | | | | | | | |
| 方法 | 向文本分类、图文匹配、只能搜索 | | | | | | | | | | | | | |
| 并发用户数与事务执行情况 | | | | | | | | | | | | | | |
| 并发用户数 | 事务平均响应时间 | | | 事务最大响应时间 | | | 平均每秒处理事务数 | | | 事务成功率 | | | 每秒点击率 | 平均流量(字节/秒) |
| A | B | C | A | B | C | A | B | C | A | B | C |
| 20 | 9833ms | 19555ms | 13ms | 21434ms | 10254ms | 27ms | 0.05 | 0.10 | 66.6 | 93.0% | 90.0% | 100.0% | 60 | 139059 |
| 25 | 10721ms | 21793ms | 14ms | 28198ms | 10122ms | 27ms | 0.05 | 0.10 | 66.6 | 93.2% | 90.0% | 100.0% | 75 | 341504 |
| 40 | 11457ms | 23129ms | 15ms | 38778ms | 11932ms | 25ms | 0.04 | 0.09 | 71.4 | 87.5% | 85.0% | 100.0% | 120 | 759398 |
| 50 | 13775ms | 25116ms | 13ms | 50481ms | 11569ms | 28ms | 0.04 | 0.10 | 66.6 | 80.0% | 84.0% | 100.0% | 150 | 1183744 |
| 60 | 14951ms | 1106ms | 15ms | 59643ms | 12569ms | 29ms | 0.04 | 0.10 | 66.6 | 66.7% | 86.7% | 100.0% | 180 | 1704960 |
| 70 | 1477ms | 1054ms | 19ms | 58723ms | 13771ms | 30ms | 0.03 | 0.09 | 58.8 | 68.6% | 71.4% | 100.0% | 210 | 2229654 |
| 100 | 1383ms | 822ms | 16ms | 57543ms | 14700ms | 29ms | 0.01 | 0.08 | 62.5 | 52.0% | 70.0% | 100.0% | 300 | 2655488 |
| 200 | 1544ms | 893ms | 17ms | 58660ms | 18512ms | 29ms | 0.01 | 0.09 | 58.8 | 13.0% | 55.0% | 100.0% | 600 | 3044557 |

### 分析结果：

由图表可知，智能搜索功能的性能相当稳定，没有明显的收到其他服务的影响，图文匹配与文本分类两者都是ServerB提供的服务，当用户并发数上升时，ServerB较之前单独测试提早被熔断。我们在部署时使用nodeSelector，使得ServerB所在的虚拟机与ElasticSearch所在的虚拟机相互独立，使用了不同的标签，在部署时进行区分。

图 5.1 不同node节点拥有不同的tags

### 测试结果综合分析及建议

限于硬件条件，应用在相对极限的压力条件下可能会无法以较高的成功率响应所有请求。但kubernetes + Istio能够自动从故障中恢复。

### 测试经验总结

使用kubernetes+Istio管理集群，对集群配置熔断和弹性伸缩，能够使Temage在较高并发用户数量的情况下拥有稳定服务，在极大用户并发的情况下，能够对高性能需求的模型相关服务熔断，保证用户能够正常的游廊浏览。