

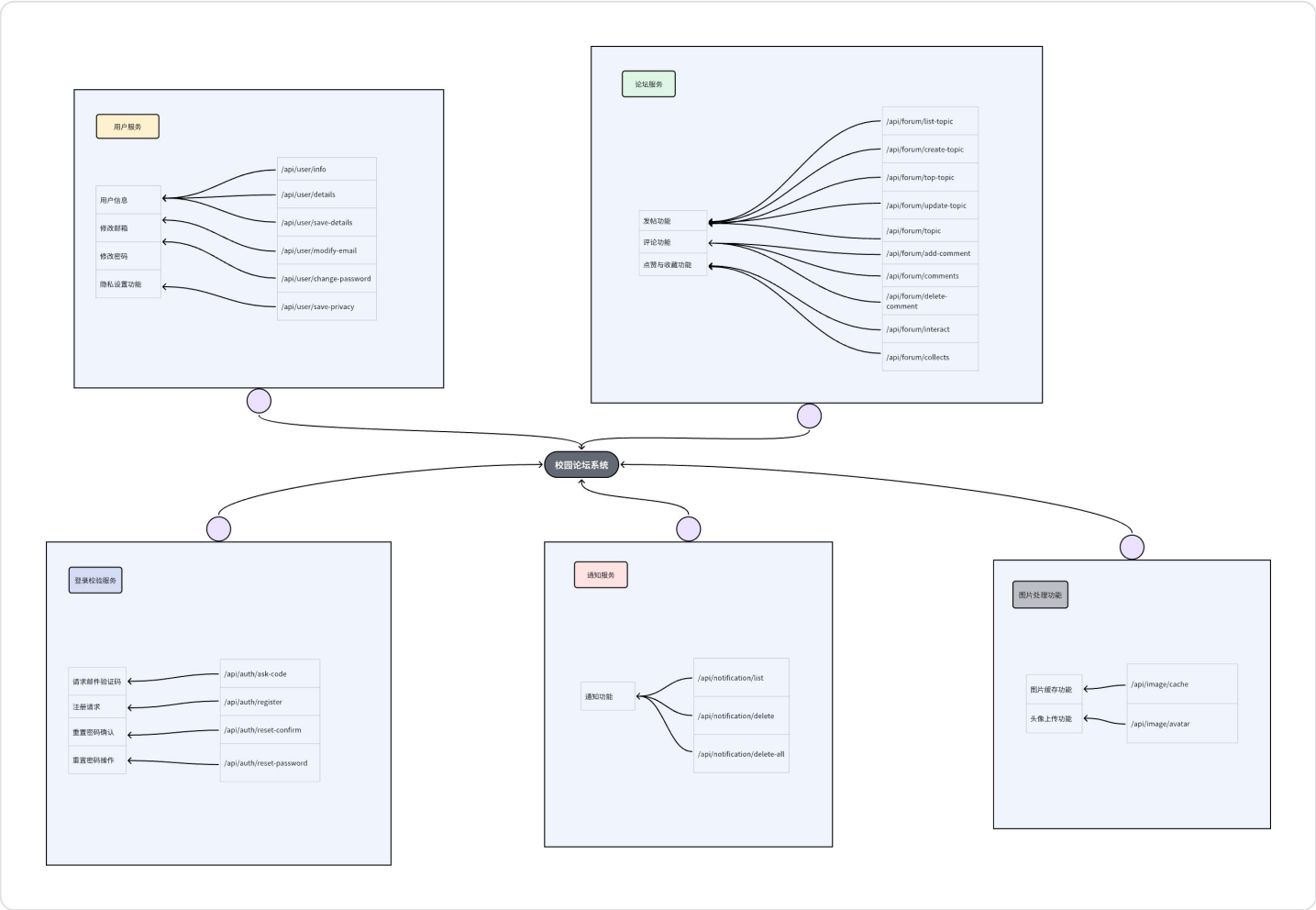
# 项目测试文档——集成测试

## 一. 测试计划

- **目标：**验证各模块集成后的系统功能正确性，确保系统各部分之间正确协作。
- **范围：**包括用户账号管理、话题管理、评论系统、通知系统、点赞和收藏功能。
- **测试环境：**基于IDEA环境的笔记本电脑，使用SpringBoot框架和Maven管理依赖。
- **角色和职责：**测试工程师负责测试用例的编写和执行，开发人员负责修复发现的缺陷。

## 二. 组件划分与测试案例

### 架构图



### 1. 用户账号管理

- **组件：**AccountController, AuthorizeController
- **功能：**注册、登录、更新资料、密码修改

• 测试函数参考表：

标识符	名称	代码行
LLD_001_FUN_001	testGetAccountInfo()	8
LLD_001_FUN_002	testGetAccountDetails()	8
LLD_001_FUN_003	testSaveAccountDetails()	18
LLD_001_FUN_004	testModifyEmail()	13
LLD_001_FUN_005	testChangePassword()	13
LLD_001_FUN_006	testSavePrivacySettings()	11
LLD_001_FUN_007	testGetPrivacySettings()	7
LLD_001_FUN_008	testAskVerifyCode()	5
LLD_001_FUN_009	testRegister()	11
LLD_001_FUN_0010	testResetConfirm()	8
LLD_001_FUN_0011	testResetPassword()	11

测试案例：

- 注册：确保当提供有效信息时用户可以成功注册（包括成功收到邮件信息）；无效信息（如重复的用户名或邮箱、不符合要求的密码）时拒绝注册。
- 登录：验证正确和错误的登录凭据处理。
- 更新资料：确保用户可以更新自己的资料，包括邮箱、密码和头像。
- 密码修改：验证用户在提供正确的旧密码时可以成功更改密码，并确保在旧密码错误时拒绝更改。

2. 话题管理

- 组件：ForumController
- 功能：列出、置顶、获取详情、创建、更新话题
- 测试函数参考表：

标识符	名称	代码行
LLD_002_FUN_001	testListTopic()	13
LLD_002_FUN_002	testTopTopic()	11

LLD_002_FUN_003	testTopic()	7
LLD_002_FUN_004	testCreateTopic()	11
LLD_002_FUN_005	testUpdateTopic()	15

测试案例：

- 列出话题：验证在用户登录状态下，通过API列出话题的操作是否成功。
- 置顶话题：测试获取置顶话题列表的功能。
- 获取话题详情：模拟用户请求查看某个话题的详细信息，验证 `topicService.getTopic` 方法是否能正确处理并返回话题的详细信息。
- 创建话题：验证用户能否成功创建新的话题。
- 更新话题：类似于创建话题，但这里是测试更新已存在话题的功能。

3. 评论系统

- 组件：ForumController
- 功能：添加评论、获取评论列表、删除评论
- 测试函数参考表：

标识符	名称	代码行
LLD_003_FUN_001	testAddComment()	12
LLD_003_FUN_002	testComments()	19
LLD_003_FUN_003	testDeleteComment()	21

测试案例：

- 添加评论：验证用户在登录状态下可以对帖子添加评论。
- 获取评论列表：验证用户可以获取包括所有评论的列表。
- 删除评论：验证用户可以删除自己的评论，检查数据库更新情况。

4. 通知系统

- 组件：NotificationController
- 功能：获取用户通知列表、删除用户通知、删除用户所有通知
- 测试函数参考表：

标识符	名称	代码行
LLD_004_FUN_001	testListNotification()	15
LLD_004_FUN_002	testDeleteNotification()	17
LLD_004_FUN_003	testDeleteAllNotification()	8

测试案例：

- 获取用户通知列表：验证系统能否成功返回指定用户的全部通知列表。
- 删除用户通知：确认单个用户通知可以被成功删除。
- 删除用户所有通知：验证用户的所有通知能够被一次性成功删除。

## 5. 点赞和收藏功能

- 组件：ForumController
- 功能：话题互动、获取收藏列表
- 测试函数参考表：

标识符	名称	代码行
LLD_005_FUN_001	testInteract()	12
LLD_005_FUN_002	testCollects()	13

测试案例：

- 话题互动：模拟用户对某个话题进行互动操作（如点赞或取消点赞），验证互动接口的调用逻辑是否正确执行，无异常发生。
- 获取收藏列表：验证系统能否成功返回指定的全部收藏列表。

## 6. 图像管理功能

- 组件：ImageController
- 功能：帖子中图片上传和展示、用户头像上传和展示
- 测试函数参考表：

标识符	名称	代码行
LLD_006_FUN_001	uploadImageSuccess()	15

LLD_006_FUN_002	uploadAvatarSuccess()	7
LLD_006_FUN_003	uploadImageFailureDueToSize()	13
LLD_006_FUN_004	uploadAvatarFailureDueToSize()	11
LLD_006_FUN_005	uploadImageServiceFailure()	8
LLD_006_FUN_006	uploadAvatarServiceFailure()	13

测试案例：

- 话题互动：模拟用户对某个话题进行互动操作（如点赞或取消点赞），验证互动接口的调用逻辑是否正确执行，无异常发生。
- 获取收藏列表：验证系统能否成功返回指定的全部收藏列表。

### 三. 测试方法

- 集成测试策略
  - **自底向上策略**：从底层的单元模块开始逐步集成和测试，先集成和测试较低层次的模块，然后逐步向上进行。
  - **自顶向下策略**：从顶层模块开始逐步集成和测试，先集成和测试较高层次的模块，然后逐步向下进行。
  - **大爆炸策略**：将所有模块一起集成和测试，适用于规模较小的系统。
  - **混合策略**：结合自底向上和自顶向下策略，根据具体情况进行灵活调整。
- 参考等价类划分方法
- 参考边界值分析方法
- 测试脚本使用 Java 语言实现
- 某函数的缺陷被修正后必须回归与该函数相关的所有集成测试用例

### 四. 测试环境

1. 现有一台笔记本电脑
2. 基于idea环境
3. 配备有springboot框架
4. 通过Maven来管理引入依赖（包括Mockito的依赖，junit单元测试依赖等等）

### 五. 测试对象

- 1. 用户账号管理系统
- 2. 帖子管理系统
- 3. 评论系统
- 4. 通知系统
- 5. 点赞和收藏系统

## 六.用例分析与设计

### 1.用户账号管理系统

#### 1.1 testGetAccountInfo()的测试分析与设计

该测试主要用于测试获取账号信息的接口 `/api/user/info`，输入用户的ID，系统会查询相应的用户账号的信息，包括用户名、密码、邮箱、照片等账户信息。当测试成功后则返回http响应状态为200的状态码，同时返回到的账号信息应该为对应的人物信息。

##### 1.1.1 标识符定义

IT\_TD\_001\_001

##### 1.1.2 被测特性

输入想要查找的用户ID，返回相应的查询结果

##### 1.1.3 测试方法

对于输入的账户ID的等价类划分可以考虑存在和不存在的两种情况。

##### 1.1.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_001_001_001	获取用户信息	高

##### 1.1.5 测试通过/失败标识

所有的测试用例都被执行同时不存在错误。

##### 1.1.6 用例设计

测试项编号	IT_TD_001_001_001
覆盖接口	/api/user/info

请求方式	Get	
优先级	高	
测试项描述	获取用户信息	
前置条件	用户登录主界面，输入账户ID进行查询	
用例序号	输入	期望结果
001	userId = "1"	HTTP:200 OK true

## 1.2 testGetAccountDetails()的测试分析与设计

该测试主要用于测试获取账号的详细信息的部分，输入用户的ID，系统会查询相应的用户账号的全部信息，包括用户名、性别、密码、邮箱、照片、电话号等更加详尽账户信息。当测试成功后则返回http响应状态为200的状态码，同时返回到的账号信息应该为对应的人物的详细信息。

### 1.2.1 标识符定义

IT\_TD\_001\_002

### 1.2.2 被测特性

输入想要查找的用户ID，返回相应的查询结果

### 1.2.3 测试方法

对于输入的账户ID的等价类划分可以考虑存在和不存在的两种情况。

### 1.2.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_001_002_001	获取用户详细信息	高

### 1.2.5 测试通过/失败标识

所有的测试用例都被执行同时不存在错误。

### 1.2.6 用例设计

测试项编号	IT_TD_001_002_001	
覆盖接口	/api/user/details	
请求方式	Get	

优先级	高	
测试项描述	获取用户详细信息	
前置条件	用户登录主界面，输入账户ID进行查询	
用例序号	输入	期望结果
001	userId = "1"	HTTP:200 OK true

### 1.3 testSaveAccountDetails()的测试分析与设计

该测试主要用于保存账户的详细信息，输设置新的用户名、密码、邮箱、电话号等详细的账户信息。当测试成功后则返回http响应状态为200的状态码。

#### 1.3.1 标识符定义

IT\_TD\_001\_003

#### 1.3.2 被测特性

输入详细的账户信息，查看状态码为200OK则请求成功。

#### 1.3.3 测试方法

对于输入的账户信息的等价类划分可以根据不同的账户信息进行划分。

#### 1.3.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_001_003_001	将输入的用户信息保存到数据库中	高

#### 1.3.5 测试通过/失败标识

所有的测试用例都被执行同时不存在错误。

#### 1.3.6 用例设计

测试项编号	IT_TD_001_003_001
覆盖接口	/api/user/save-details
请求方式	Post
优先级	高
测试项描述	将输入的用户信息保存到数据库中



前置条件	用户登录主界面，输入用户详细信息并保存	
用例序号	输入	期望结果
001	detailsSaveVO.setUsername("newuser"); detailsSaveVO.setGender(0); detailsSaveVO.setPhone("1234567890"); detailsSaveVO.setQq("123456"); detailsSaveVO.setWx("userwx"); detailsSaveVO.setDesc("New description");	HTTP:200 OK true

1.4 testModifyEmail()的测试分析与设计

该测试主要用于邮箱信息修改，用户输入想要修改后的邮箱信息，同时获取邮箱验证码，输入后修改邮箱。

1.4.1 标识符定义

IT\_TD\_001\_004

1.4.2 被测特性

输入新的邮箱信息以及获取的验证码，查看状态码为200OK则请求成功。

1.4.3 测试方法

针对不同的邮箱发送验证码进行测试。

1.4.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_001_004_001	输入新的邮箱以及验证码修改邮箱信息	高

1.4.5 测试通过/失败标识

所有的测试用例都被执行同时不存在错误。

1.4.6 用例设计

测试项编号	IT_TD_001_004_001
覆盖接口	/api/user/modify-email
请求方式	Post

优先级	高	
测试项描述	输入新的邮箱以及验证码修改邮箱信息	
前置条件	用户进入到修改邮箱的界面	
用例序号	输入	期望结果
001	modifyEmailVO.setEmail("new.email@example.com"); modifyEmailVO.setCode("123456");	HTTP:200 OK true

1.5 testChangePassword()的测试分析与设计

该测试函数用于修改用户修改账号密码，输入旧密码以及新密码即能设置。

1.5.1 标识符定义

IT\_TD\_001\_005

1.5.2 被测特性

输入旧密码和新密码，提交后，查看状态码为200OK则请求成功。

1.5.3 测试方法

对新旧密码进行随机编辑，进行测试。

1.5.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_001_005_001	修改密码	高

1.5.5 测试通过/失败标识

所有的测试用例都被执行同时不存在错误。

1.5.6 用例设计

测试项编号	IT_TD_001_005_001	
覆盖接口	/api/user/change-password	
请求方式	Post	
优先级	高	

测试项描述	修改密码	
前置条件	用户进入到修改密码的界面，输入旧密码以及新密码进行修改	
用例序号	输入	期望结果
001	changePasswordVO.setPassword("old Password"); changePasswordVO.setNew_password("newPassword");	HTTP:200 OK true

## 1.6 testSavePrivacySettings()的测试分析与设计

该测试函数用来测试用户信息隐私设置的操作。

### 1.6.1 标识符定义

IT\_TD\_001\_006

### 1.6.2 被测特性

选择想要保持隐私的信息，不向外部环境公开，设置该信息的状态为false，则该信息无法被其他人来查询。

### 1.6.3 测试方法

选择一条想要设置为隐私的一条信息，调整他的status为false，让他不能被其他人可见。

### 1.6.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_001_006_001	信息隐私设置	高

### 1.6.5 测试通过/失败标识

所有的测试用例都被执行同时不存在错误。

### 1.6.6 用例设计

测试项编号	IT_TD_001_006_001
覆盖接口	/api/user/save-privacy
请求方式	Post
优先级	高

测试项描述	信息隐私设置	
前置条件	用户进入到信息设置界面	
用例序号	输入	期望结果
001	privacySaveVO.setType("phone"); privacySaveVO.setStatus(false);	HTTP:200 OK 设置成功

## 1.7 testGetPrivacySettings()的测试分析与设计

主要测试获取隐私信息这样的功能。

### 1.7.1 标识符定义

IT\_TD\_001\_007

### 1.7.2 被测特性

当隐私信息设置好之后会存储到对应的隐私数据数据库，那么我们去访问数据库获取相应的数据。

### 1.7.3 测试方法

利用get方法从数据库中获取隐私设置数据

### 1.7.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_001_007_001	获取隐私设置数据	高

### 1.7.5 测试通过/失败标识

所有的测试用例都被执行同时不存在错误。

### 1.7.6 用例设计

测试项编号	IT_TD_001_007_001
覆盖接口	/api/user/privacy
请求方式	Get
优先级	高
测试项描述	获取隐私设置数据

前置条件	用户进入到信息设置界面	
用例序号	输入	期望结果
001	“”	HTTP:200 OK 找到隐私信息

## 1.8 testAskVerifyCode()的测试分析与设计

主要测试获取邮箱验证码功能。

### 1.8.1 标识符定义

IT\_TD\_001\_008

### 1.8.2 被测特性

用户请求通过邮箱获取验证码，模拟用户认证的过程。

### 1.8.3 测试方法

利用get方法从指定api获取验证码。

### 1.8.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_001_008_001	获取邮箱验证码成功	高

### 1.8.5 测试通过/失败标识

所有的测试用例都被执行同时不存在错误。

### 1.8.6 用例设计

测试项编号	IT_TD_001_008_001	
覆盖接口	/api/auth/ask-code	
请求方式	Get	
优先级	高	
测试项描述	获取邮箱验证码成功	
前置条件	用户登录主界面，在获取时间间隔外	
用例序号	输入	期望结果

001	email = "test@example.com" type = "register"	HTTP:200 OK 获取邮箱验证码成功
-----	---	--------------------------

## 1.9 testRegister()的测试分析与设计

主要测试用户注册功能。

### 1.9.1 标识符定义

IT\_TD\_001\_009

### 1.9.2 被测特性

用户请求注册，测试新用户能否正常交互。

### 1.9.3 测试方法

利用post方法向数据库写入新的用户信息。

### 1.9.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_001_009_001	用户注册成功	高

### 1.9.5 测试通过/失败标识

所有的测试用例都被执行同时不存在错误。

### 1.9.6 用例设计

测试项编号	IT_TD_001_009_001	
覆盖接口	/api/auth/register	
请求方式	Post	
优先级	高	
测试项描述	用户注册成功	
前置条件	用户登录主界面，输入账户ID进行查询	
用例序号	输入	期望结果
001	email = "test@example.com"	HTTP:200 OK

	password = "password123" code = "123456"	用户注册成功
--	---	--------

## 1.10 testResetConfirm()的测试分析与设计

主要测试系统密码重置确认相关功能。

### 1.10.1 标识符定义

IT\_TD\_001\_0010

### 1.10.2 被测特性

用户请求密码重置，测试用户能否正常交互。

### 1.10.3 测试方法

集成测试，结合模拟get的HTTP请求验证重置密码与数据库交互的正确性。

### 1.10.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_001_0010_001	密码重置成功	高

### 1.10.5 测试通过/失败标识

所有的测试用例都被执行同时不存在错误。

### 1.10.6 用例设计

测试项编号	IT_TD_001_0010_001	
覆盖接口	/api/auth/reset-confirm	
请求方式	Post	
优先级	高	
测试项描述	密码重置成功	
前置条件	用户登录主界面，输入账户ID进行查询	
用例序号	输入	期望结果
001	email = "test@example.com" code = "123456"	HTTP:200 OK 密码重置成功

## 1.11 testResetPassword()的测试分析与设计

主要测试系统密码重置操作相关功能。

### 1.11.1 标识符定义

IT\_TD\_001\_0011

### 1.11.2 被测特性

用户请求密码重置，测试用户能否正常交互。

### 1.11.3 测试方法

集成测试，结合模拟post的HTTP请求验证重置密码与数据库交互的正确性。

### 1.11.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_001_0011_001	密码重置操作成功	高

### 1.11.5 测试通过/失败标识

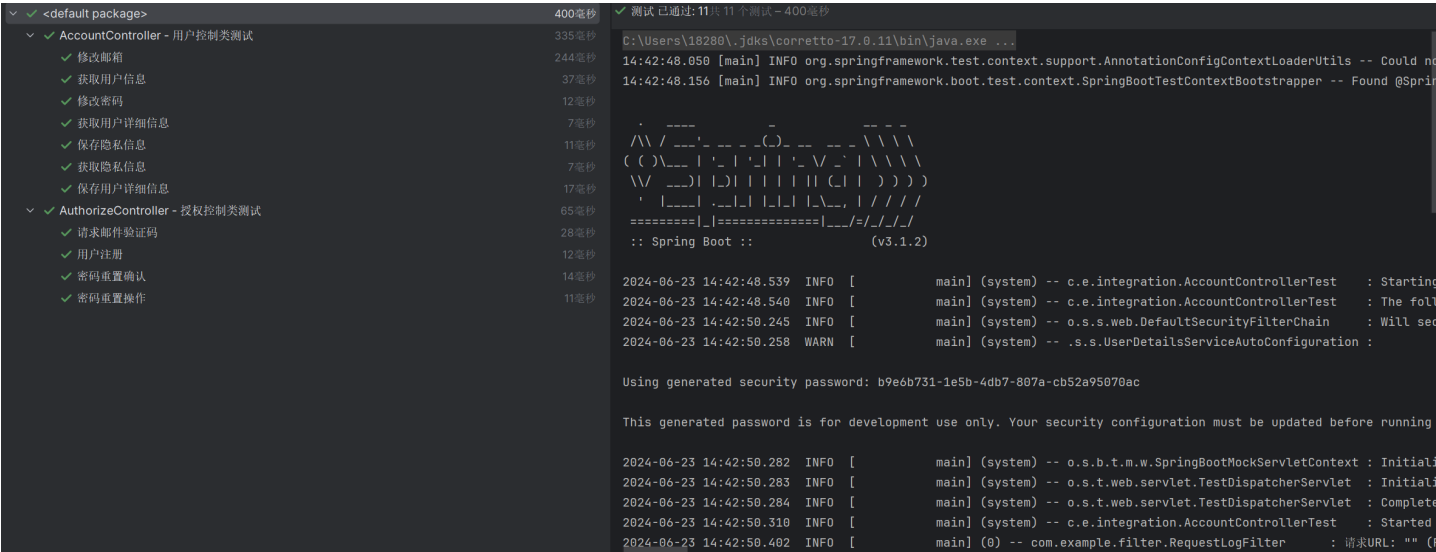
所有的测试用例都被执行同时不存在错误。

### 1.11.6 用例设计

测试项编号	IT_TD_001_0011_001	
覆盖接口	/api/auth/reset-password	
请求方式	Post	
优先级	高	
测试项描述	密码重置操作成功	
前置条件	用户登录主界面，输入账户ID进行查询	
用例序号	输入	期望结果
001	email = "test@example.com" password = "newPass123" code = "123456"	HTTP:200 OK 密码重置操作成功

## 测试截图





## 2.话题管理系统

### 2.1 testListTopic()的测试分析与设计

主要测试系统获取话题列表的功能。

#### 2.1.1 标识符定义

IT\_TD\_002\_001

#### 2.1.2 被测特性

系统应能正确列出指定条件下的所有话题列表，包括分页和筛选功能。

#### 2.1.3 测试方法

通过模拟HTTP请求验证接口输出，使用get方法，确保 `topicService` 与数据库交互正确。

#### 2.1.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_002_001_001	获取话题列表成功	高
IT_TD_002_001_002	获取话题列表失败	低

#### 2.1.5 测试通过/失败标识

- 通过：HTTP响应状态码为200，返回的数据结构正确，且与预期的数据库查询结果一致。
- 失败：HTTP响应非200，数据缺失或不符合预期，分页逻辑错误。

#### 2.1.6 用例设计

--	--

测试项编号	IT_TD_002_001_001	
覆盖接口	/api/forum/list-topic	
请求方式	Get	
优先级	高	
测试项描述	获取话题列表成功	
前置条件	用户进入论坛页面	
用例序号	输入	期望结果
001	page = "0" type = "1"	HTTP:200 OK 找到话题列表

测试项编号	IT_TD_002_001_002	
覆盖接口	/api/forum/list-topic	
请求方式	Get	
优先级	低	
测试项描述	获取话题列表失败	
前置条件	用户进入论坛页面	
用例序号	输入	期望结果
001	page = "-1" type = "-1"	HTTP:非200 OK 找不到话题列表

## 2.2 testTopTopic()的测试分析与设计

主要测试系统话题置顶相关功能。

### 2.2.1 标识符定义

IT\_TD\_002\_002

### 2.2.2 被测特性

系统应能列出所有置顶的话题。

### 2.2.3 测试方法

使用get方法，验证API输出是否包含正确的置顶话题列表。

2.2.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_002_002_001	获取置顶话题成功	高
IT_TD_002_002_002	获取置顶话题失败	低

2.2.5 测试通过/失败标识

- 通过：成功返回置顶话题列表，状态码200。
- 失败：无置顶话题时返回空列表而非错误，状态码非200。

2.2.6 用例设计

测试项编号	IT_TD_002_002_001	
覆盖接口	/api/forum/top-topic	
请求方式	Get	
优先级	高	
测试项描述	获取置顶话题成功	
前置条件	用户进入论坛页面	
用例序号	输入	期望结果
001		HTTP:200 OK 找到置顶话题

测试项编号	IT_TD_002_002_002	
覆盖接口	/api/forum/top-topic	
请求方式	Get	
优先级	低	
测试项描述	获取置顶话题失败	
前置条件	用户进入论坛页面	
用例序号	输入	期望结果

001		HTTP:非200 OK 找不到置顶话题
-----	--	-------------------------

## 2.3 testTopic()的测试分析与设计

主要测试系统获取话题详情相关功能。

### 2.3.1 标识符定义

IT\_TD\_002\_003

### 2.3.2 被测特性

根据话题ID正确返回话题详情。

### 2.3.3 测试方法

通过get方法，测试话题ID的有效性检查，详情内容准确性，根据话题ID能否正确返回话题详情。

### 2.3.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_002_003_001	获取话题详情成功	高
IT_TD_002_003_002	获取话题详情失败	低

### 2.3.5 测试通过/失败标识

正确返回预期详情为通过，否则失败。

### 2.3.6 用例设计

测试项编号	IT_TD_002_003_001	
覆盖接口	/api/forum/topic	
请求方式	Get	
优先级	高	
测试项描述	获取话题详情成功	
前置条件	用户进入论坛页面	
用例序号	输入	期望结果
001	tid = "1"	HTTP:200 OK

		找到话题详情
--	--	--------

测试项编号	IT_TD_002_003_002	
覆盖接口	/api/forum/topic	
请求方式	Get	
优先级	低	
测试项描述	获取话题详情失败	
前置条件	用户进入论坛页面	
用例序号	输入	期望结果
001	tid = "-1"	HTTP:非200 OK 找不到话题详情

2.4 testCreateTopic()的测试分析与设计

主要测试系统创建话题相关功能。

2.4.1 标识符定义

IT\_TD\_002\_004

2.4.2 被测特性

用户能够创建新话题，数据被正确存储。

2.4.3 测试方法

使用post方法，创建后数据库记录验证，返回的响应确认创建成功。

2.4.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_002_004_001	创建话题成功	高
IT_TD_002_004_002	创建话题失败	低

2.4.5 测试通过/失败标识

创建成功且数据库有对应记录为通过，否则失败。

2.4.6 用例设计

测试项编号	IT_TD_002_004_001	
覆盖接口	/api/forum/create-topic	
请求方式	Post	
优先级	高	
测试项描述	创建话题成功	
前置条件	用户进入论坛页面	
用例序号	输入	期望结果
001	type = "1" title = "New Topic" content = "123"	HTTP:200 OK 成功创建话题

测试项编号	IT_TD_002_004_002	
覆盖接口	/api/forum/create-topic	
请求方式	Post	
优先级	低	
测试项描述	创建话题失败	
前置条件	用户进入论坛页面	
用例序号	输入	期望结果
001	type = "-1" title = null content = null	HTTP:非200 OK 创建话题失败

2.5 testUpdateTopic()的测试分析与设计

主要测试系统更新话题功能。

2.5.1 标识符定义

IT\_TD\_002\_005

2.5.2 被测特性

用户可以修改已有话题的信息。

2.5.3 测试方法

使用post方法，更新后的数据验证，确保变更被正确反映。

2.5.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_002_005_001	更新话题成功	高
IT_TD_002_005_002	更新话题失败	低

2.5.5 测试通过/失败标识

话题信息更新成功且数据库同步更新为通过，否则失败。

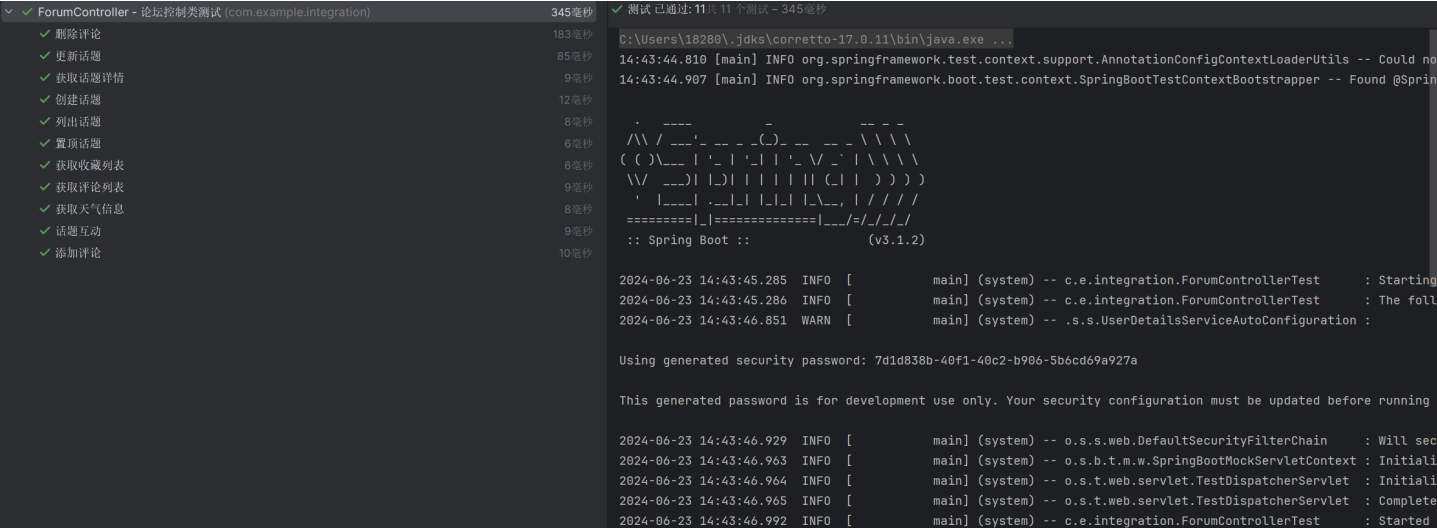
2.5.6 用例设计

测试项编号	IT_TD_002_005_001	
覆盖接口	/api/forum/update-topic	
请求方式	Post	
优先级	高	
测试项描述	更新话题成功	
前置条件	用户进入论坛页面	
用例序号	输入	期望结果
001	id = "1" type = "1" title = "Updated Topic" content = "123"	HTTP:200 OK 成功更新话题

测试项编号	IT_TD_002_005_002	
覆盖接口	/api/forum/update-topic	
请求方式	Post	

优先级	低	
测试项描述	更新话题失败	
前置条件	用户进入论坛页面	
用例序号	输入	期望结果
001	id = "-1" type = "-1" title = null content = null	HTTP:非200 OK 更新话题失败

测试截图



3.评论系统

3.1 testAddComment()的测试分析与设计

主要测试系统添加评论相关功能。

3.1.1 标识符定义

IT\_TD\_003\_001

3.1.2 被测特性

- 用户能够在指定话题下添加评论，评论内容应被正确保存至数据库。

3.1.3 测试方法

- 集成测试，结合模拟post的HTTP请求验证评论服务与数据库交互。

3.1.4 测试项标识



测试项标识符	测试项描述	优先级
IT_TD_003_001_001	添加评论成功	高
IT_TD_003_001_002	添加评论失败	低

3.1.5 测试通过/失败标识

- 通过：评论成功创建，数据库中有对应记录，HTTP响应状态码200。
- 失败：无效输入返回错误信息，数据库无新增记录，状态码非200。

3.1.6 用例设计

测试项编号	IT_TD_003_001_001	
覆盖接口	/api/forum/add-comment	
请求方式	Post	
优先级	高	
测试项描述	添加评论成功	
前置条件	用户进入话题页面	
用例序号	输入	期望结果
001	tid = "1" quote = "-1" content = "This is a comment"	HTTP:200 OK 添加评论成功

测试项编号	IT_TD_003_001_002	
覆盖接口	/api/forum/add-comment	
请求方式	Post	
优先级	低	
测试项描述	添加评论失败	
前置条件	用户进入话题页面	
用例序号	输入	期望结果
001	tid = "-1"	HTTP:非200 OK

	quote = "-1" content = null	添加评论失败
--	--------------------------------	--------

### 3.2 testComments()的测试分析与设计

主要测试系统获取评论列表相关功能。

#### 3.2.1 标识符定义

IT\_TD\_003\_002

#### 3.2.2 被测特性

- 能够根据话题ID正确获取其关联的评论列表，支持分页和排序。

#### 3.2.3 测试方法

- 集成测试，结合模拟get的HTTP请求测试API能否返回正确的评论列表。

#### 3.2.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_003_002_001	获取评论列表成功	高
IT_TD_003_002_002	获取评论列表失败	低

#### 3.2.5 测试通过/失败标识

- 通过：返回正确的评论列表，符合分页和排序要求，状态码200。
- 失败：返回错误的评论列表，分页或排序逻辑错误，状态码非200。

#### 3.2.6 用例设计

测试项编号	IT_TD_003_002_001	
覆盖接口	/api/forum/comments	
请求方式	Get	
优先级	高	
测试项描述	获取评论列表成功	
前置条件	用户进入话题页面	
用例序号	输入	期望结果

001	tid = "1" page = "0"	HTTP:200 OK 获取评论列表成功
-----	-------------------------	-------------------------

测试项编号	IT_TD_003_002_002	
覆盖接口	/api/forum/comments	
请求方式	Get	
优先级	低	
测试项描述	获取评论列表失败	
前置条件	用户进入话题页面	
用例序号	输入	期望结果
001	tid = "-1" page = "-1"	HTTP:非200 OK 获取评论列表失败

### 3.3 testDeleteComment()的测试分析与设计

主要测试系统删除评论相关功能。

#### 3.3.1 标识符定义

IT\_TD\_003\_003

#### 3.3.2 被测特性

- 用户能够删除自己发布的评论，且删除后评论不再显示于评论列表。

#### 3.3.3 测试方法

- 集成测试，结合模拟get的HTTP请求测试API能否返回正确删除指定的评论。

#### 3.3.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_003_003_001	删除评论成功	高
IT_TD_003_003_002	删除评论失败	低

#### 3.3.5 测试通过/失败标识

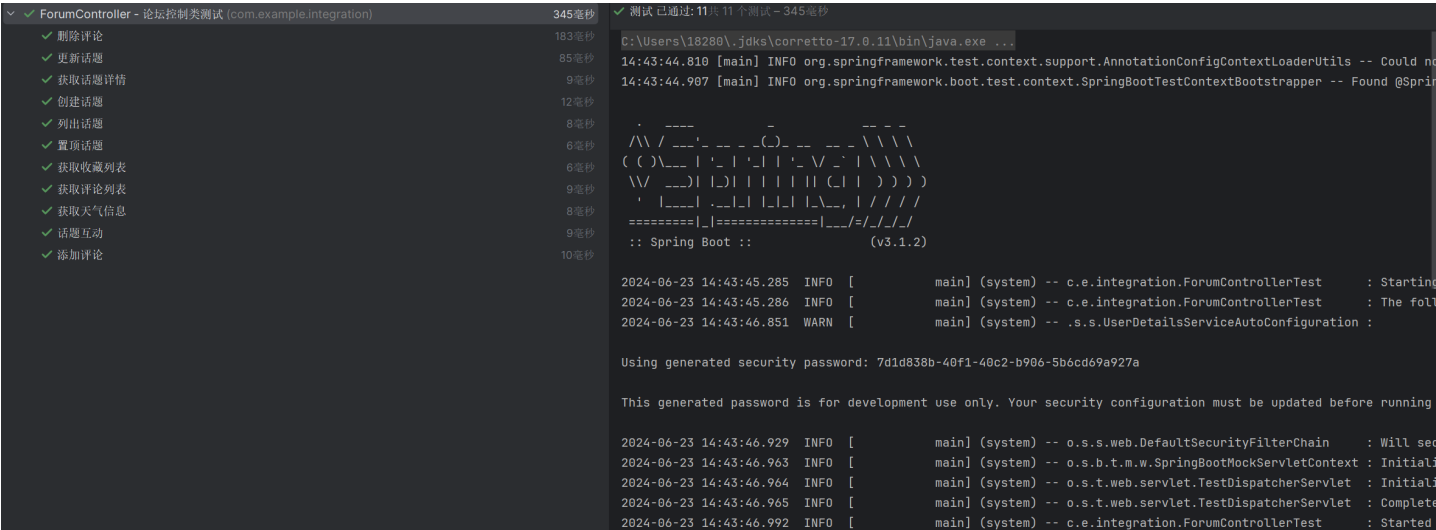
- 通过：评论成功删除，从数据库移除，再次请求评论列表时该评论不存在。
- 失败：无权限用户尝试删除他人评论时操作失败，状态码非200。

3.3.6 用例设计

测试项编号	IT_TD_003_003_001	
覆盖接口	/api/forum/delete-comment	
请求方式	Get	
优先级	高	
测试项描述	删除评论成功	
前置条件	用户进入话题页面	
用例序号	输入	期望结果
001	id = "1"	HTTP:200 OK 删除评论成功

测试项编号	IT_TD_003_003_002	
覆盖接口	/api/forum/delete-comment	
请求方式	Get	
优先级	低	
测试项描述	删除评论失败	
前置条件	用户进入话题页面	
用例序号	输入	期望结果
001	id = "-1"	HTTP:非200 OK 删除评论失败

测试截图



## 4.通知系统

### 4.1 testListNotification()的测试分析与设计

主要测试系统获取用户通知列表相关功能。

#### 4.1.1 标识符定义

IT\_TD\_004\_001

#### 4.1.2 被测特性

- 系统应能根据用户ID列出该用户的全部通知列表，支持分页和过滤。

#### 4.1.3 测试方法

- 集成测试，结合模拟get的HTTP请求验证通知服务与数据库交互的正确性。

#### 4.1.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_004_001_001	获取用户通知列表成功	高
IT_TD_004_001_002	获取用户通知列表失败	低

#### 4.1.5 测试通过/失败标识

- 通过：成功返回用户的通知列表，状态码200，数据准确。
- 失败：无法获取通知列表，数据不准确，状态码非200。

#### 4.1.6 用例设计

测试项编号	IT_TD_004_001_001
-------	-------------------

覆盖接口	/api/notification/list	
请求方式	Get	
优先级	高	
测试项描述	获取用户通知列表成功	
前置条件	用户进入系统，完成身份验证	
用例序号	输入	期望结果
001		HTTP:200 OK 获取用户通知列表成功

测试项编号	IT_TD_004_001_002	
覆盖接口	/api/notification/list	
请求方式	Get	
优先级	低	
测试项描述	获取用户通知列表失败	
前置条件	用户进入系统，完成身份验证	
用例序号	输入	期望结果
001		HTTP:非200 OK 获取用户通知列表失败

## 4.2 testDeleteNotification()的测试分析与设计

主要测试系统删除用户通知相关功能。

### 4.2.1 标识符定义

IT\_TD\_004\_002

### 4.2.2 被测特性

- 用户能够删除指定的通知，系统应正确处理该请求并更新通知状态。

### 4.2.3 测试方法

- 集成测试，结合模拟get的HTTP请求验证删除操作后数据库记录的变更。

4.2.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_004_002_001	删除用户通知成功	高
IT_TD_004_002_002	删除用户通知失败	低

4.2.5 测试通过/失败标识

- 通过：通知成功删除，数据库记录更新，状态码200。
- 失败：无法删除通知，状态码非200，数据库记录未改变。

4.2.6 用例设计

测试项编号	IT_TD_004_002_001	
覆盖接口	/api/notification/delete	
请求方式	Get	
优先级	高	
测试项描述	删除用户通知成功	
前置条件	用户进入系统，完成身份验证	
用例序号	输入	期望结果
001	id = "1"	HTTP:200 OK 删除用户通知成功

测试项编号	IT_TD_004_002_002	
覆盖接口	/api/notification/delete	
请求方式	Get	
优先级	低	
测试项描述	删除用户通知失败	
前置条件	用户进入系统，完成身份验证	
用例序号	输入	期望结果

001	id = "-1"	HTTP:非200 OK 删除用户通知失败
-----	-----------	--------------------------

### 4.3 testDeleteAllNotification()的测试分析与设计

主要测试系统删除用户所有通知相关测试。

#### 4.3.1 标识符定义

IT\_TD\_004\_003

#### 4.3.2 被测特性

- 用户能够清空自己的所有通知，系统需正确处理并更新用户通知状态。

#### 4.3.3 测试方法

- 集成测试，结合模拟get的HTTP请求验证批量删除操作后用户通知列表为空。

#### 4.3.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_004_003_001	删除用户所有通知成功	高
IT_TD_004_003_002	删除用户所有通知失败	低

#### 4.3.5 测试通过/失败标识

- 通过：所有通知成功删除，数据库中用户通知记录为空，状态码200。
- 失败：未能清空通知，状态码非200，数据库记录未清空。

#### 4.3.6 用例设计

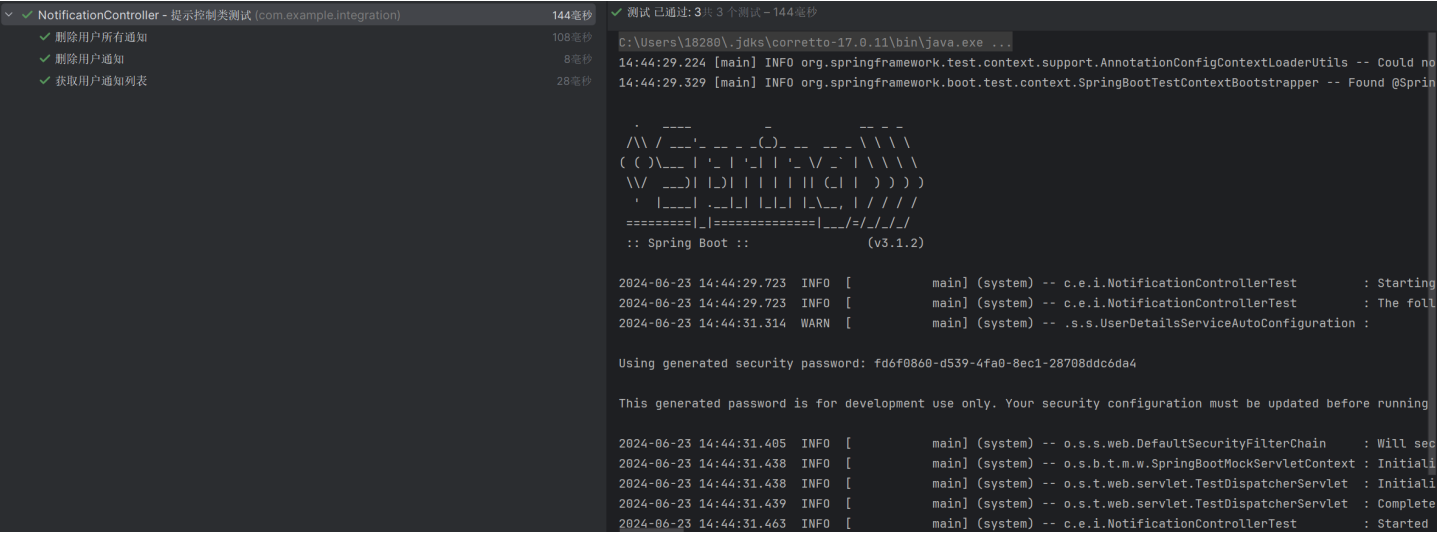
测试项编号	IT_TD_004_003_001	
覆盖接口	/api/notification/delete-all	
请求方式	Get	
优先级	高	
测试项描述	删除用户所有通知成功	
前置条件	用户进入系统，完成身份验证	
用例序号	输入	期望结果



001	HTTP:200 OK 删除用户所有通知成功
-----	---------------------------

测试项编号	IT_TD_004_003_002	
覆盖接口	/api/notification/delete-all	
请求方式	Get	
优先级	低	
测试项描述	删除用户所有通知失败	
前置条件	用户进入系统，完成身份验证	
用例序号	输入	期望结果
001		HTTP:非200 OK 删除用户所有通知失败

测试截图



5.点赞和收藏系统

5.1 testInteract()的测试分析与设计

主要测试系统话题互动相关功能。

5.1.1 标识符定义

IT\_TD\_005\_001

5.1.2 被测特性

- 用户能对话题进行互动操作（如点赞、取消点赞、回复等），系统需正确处理互动请求并更新相应数据。

5.1.3 测试方法

- 集成测试，结合模拟的HTTP请求验证话题互动逻辑及数据库更新。

5.1.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_005_001_001	话题互动成功	高
IT_TD_005_001_002	话题互动失败	低

5.1.5 测试通过/失败标识

- 通过：互动操作成功执行，数据库状态更新正确，返回状态码200。
- 失败：无效互动请求，权限校验失败，状态码非200。

5.1.6 用例设计

测试项编号	IT_TD_005_001_001	
覆盖接口	/api/forum/interact	
请求方式	Get	
优先级	高	
测试项描述	话题互动成功	
前置条件	用户进入系统，完成身份验证	
用例序号	输入	期望结果
001	tid = "1" type = "like" state = "true"	HTTP:200 OK 话题互动成功

测试项编号	IT_TD_005_001_002	
覆盖接口	/api/forum/interact	
请求方式	Get	

优先级	低	
测试项描述	话题互动失败	
前置条件	用户进入系统，完成身份验证	
用例序号	输入	期望结果
001	tid = "-1" type = "-1" state = "false"	HTTP:非200 OK 话题互动失败

## 5.2 testCollects()的测试分析与设计

主要测试系统获取收藏列表相关功能。

### 5.2.1 标识符定义

IT\_TD\_005\_002

### 5.2.2 被测特性

- 用户能查看自己收藏的话题列表，系统应正确展示用户收藏的所有话题。

### 5.2.3 测试方法

- 集成测试，结合模拟的HTTP请求验证收藏列表接口与数据库交互的准确性。

### 5.2.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_005_002_001	获取收藏列表成功	高
IT_TD_005_002_002	获取收藏列表失败	低

### 5.2.5 测试通过/失败标识

- 通过：成功返回用户收藏的话题列表，数据准确，状态码200。
- 失败：无法获取收藏列表，数据不准确或状态码非200。

### 5.2.6 用例设计

测试项编号	IT_TD_005_002_001
覆盖接口	/api/forum/collects



## 6.图像管理系统

### 6.1 uploadImageSuccess()的测试分析与设计

主要测试上传图像相关功能。

#### 6.1.1 标识符定义

IT\_TD\_006\_001

#### 6.1.2 被测特性

- 服务能够处理有效的图片上传请求。
- 服务返回正确的图片URL。

#### 6.1.3 测试方法

- 使用 MockMvc 进行模拟HTTP的multipart请求。

#### 6.1.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_006_001_001	上传图像成功	高

#### 6.1.5 测试通过/失败标识

- 通过: 返回状态码200并且JSON路径 \$.data 包含正确的URL。
- 失败: 状态码不是200或者返回的数据不包含正确的URL。

#### 6.1.6 用例设计

测试项编号	IT_TD_006_001_001	
覆盖接口	/api/image/cache	
请求方式	POST	
优先级	高	
测试项描述	上传图片成功	
前置条件	用户进入系统，完成身份验证	
用例序号	输入	期望结果
001		

	<pre>file = MockMultipartFile("file", "image.jpg", "image/jpeg", "test image content".getBytes())  userId = 1</pre>	HTTP:200 OK, JSON路径 \$.data 包含 /image.jpg ，显示图片上传成功
--	---	---

## 6.2 uploadAvatarSuccess()的测试分析与设计

主要测试上传头像相关功能。

### 6.2.1 标识符定义

IT\_TD\_006\_002

### 6.2.2 被测特性

- 服务能够处理有效的头像上传请求。
- 服务返回正确的头像URL。

### 6.2.3 测试方法

- 使用 MockMvc 进行模拟HTTP的multipart请求。

### 6.2.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_006_002_001	上传头像成功	高

### 6.2.5 测试通过/失败标识

- 通过: 返回状态码200并且JSON路径 \$.data 包含正确的URL。
- 失败: 状态码不是200或者返回的数据不包含正确的URL。

### 6.2.6 用例设计

测试项编号	IT_TD_006_002_001
覆盖接口	/api/image/avatar
请求方式	POST
优先级	高
测试项描述	上传头像成功
前置条件	用户进入系统，完成身份验证

用例序号	输入	期望结果
002	file = MockMultipartFile("file", "avatar.jpg", "image/jpeg", "test avatar content".getBytes())  userId = 1	HTTP:200 OK, JSON路径 \$.data 包 含 /avatar.jpg ，显示图片上传成功

### 6.3 uploadImageFailureDueToSize()的测试分析与设计

主要测试上传图像超出大小限制的情况。

#### 6.3.1 标识符定义

IT\_TD\_006\_001

#### 6.3.2 被测特性

- 服务能够正确处理文件大小超限的情况。
- 服务返回相应的错误代码和消息。

#### 6.3.3 测试方法

- 使用 MockMvc 进行模拟HTTP的multipart请求。

#### 6.3.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_006_001_002	拒绝过大的文件上传。  返回正确的错误代码和错误消息。	高

#### 6.3.5 测试通过/失败标识

- 通过: 返回状态码400并且JSON路径 \$.code 和 \$.message 符合预期。
- 失败: 状态码不是400或错误信息不正确。

#### 6.3.6 用例设计

测试项编号	IT_TD_006_001_002
覆盖接口	/api/image/cache
请求方式	POST
优先级	高

测试项描述	上传图片失败，因文件太大	
前置条件	用户进入系统，完成身份验证	
用例序号	输入	期望结果
003	<pre>file = MockMultipartFile("file", "image.jpg", "image/jpeg", new byte[1024 * 1024 * 6])  userId = 1</pre>	HTTP:400 Bad Request, JSON路径 \$.code 和 \$.message 包含错误信息，提示“头像图片不能大于5MB”

## 6.4 uploadAvatarFailureDueToSize()的测试分析与设计

主要测试上传头像超出大小限制的情况。

### 6.4.1 标识符定义

IT\_TD\_006\_002

### 6.4.2 被测特性

- 服务能够处理有效的图片上传请求。
- 服务返回正确的图片URL。

### 6.4.3 测试方法

- 使用 MockMvc 进行模拟HTTP的multipart请求。

### 6.4.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_006_002_002	拒绝过大的头像文件上传。 返回正确的错误代码和错误消息。	高

### 6.4.5 测试通过/失败标识

- 通过: 返回状态码400并且JSON路径 \$.code 和 \$.message 符合预期。
- 失败: 状态码不是400或错误信息不正确。

### 6.4.6 用例设计

测试项编号	IT_TD_006_002_002
覆盖接口	/api/image/avatar



请求方式	POST	
优先级	高	
测试项描述	上传头像失败，因文件太大	
前置条件	用户进入系统，完成身份验证	
用例序号	输入	期望结果
004	file = MockMultipartFile("file", "avatar.jpg", "image/jpeg", new byte[1024 * 101])  userId = 1	HTTP:400 Bad Request, JSON路径 \$.code 和 \$.message 包含错误信息，提示“头像图片不能大于100KB”

## 6.5 uploadImageServiceFailure()的测试分析与设计

主要测试上传图像服务错误的情况。

### 6.5.1 标识符定义

IT\_TD\_006\_001

### 6.5.2 被测特性

- 服务在遇到内部错误时能够返回适当的错误信息。

### 6.5.3 测试方法

- 使用 MockMvc 进行模拟HTTP的multipart请求，并通过Mockito模拟服务层返回null。

### 6.5.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_006_001_003	处理服务层错误，返回错误状态码和消息。	高

### 6.5.5 测试通过/失败标识

- 通过: 返回状态码400并且JSON路径 \$.code 和 \$.message 包含适当的错误信息。
- 失败: 状态码不是400或错误信息不适当。

### 6.5.6 用例设计

测试项编号	IT_TD_006_001_003
-------	-------------------

覆盖接口	/api/image/cache	
请求方式	POST	
优先级	高	
测试项描述	上传图片失败，因服务错误	
前置条件	用户进入系统，完成身份验证	
用例序号	输入	期望结果
005	file = MockMultipartFile("file", "image.jpg", "image/jpeg", "test image content".getBytes())  userId = 1	HTTP:400 Bad Request, JSON路径 \$.code 和 \$.message 包含错误信息，提示“图片上传失败，请联系管理员！”

## 6.6 uploadAvatarServiceFailure()的测试分析与设计

主要测试上传头像服务错误的情况。

### 6.6.1 标识符定义

IT\_TD\_006\_002

### 6.6.2 被测特性

- 服务在遇到内部错误时能够返回适当的错误信息。

### 6.6.3 测试方法

- 使用 MockMvc 进行模拟HTTP的multipart请求，并通过Mockito模拟服务层返回null。

### 6.6.4 测试项标识

测试项标识符	测试项描述	优先级
IT_TD_006_002_003	处理服务层错误，返回错误状态码和消息。	低

### 6.6.5 测试通过/失败标识

- 通过: 返回状态码400并且JSON路径 \$.code 和 \$.message 包含适当的错误信息。
- 失败: 状态码不是400或错误信息不适当。

### 6.6.6 用例设计

测试项编号	IT_TD_006_002_003	
覆盖接口	/api/image/avatar	
请求方式	POST	
优先级	低	
测试项描述	上传头像失败，因服务错误	
前置条件	用户进入系统，完成身份验证	
用例序号	输入	期望结果
006	file = MockMultipartFile("file", "avatar.jpg", "image/jpeg", "test avatar content".getBytes())  userId = 1	HTTP:400 Bad Request, JSON路径 \$.code 和 \$.message 包含错误信息，提示“头像上传失败，请联系管理员！”

测试截图

