

集成测试计划与报告

集成测试计划

1. 引言

1.1 目的

本文是“StarGames星游游戏零售平台”系统的集成测试的大纲，主要描述如何进行集成测试活动、如何控制集成测试活动、集成测试活动的流程以及集成测试活动的工作安排等，以保证程序集成起来能正常工作，保证程序的完整运行。

1.2 范围

本测试计划主要是针对软件的集成测试:不含硬件、系统测试以及单元测试（完成单元测试是前提）。

主要的任务:

- 测试在把各个模块连接起来的时候，穿越模块接口的数据是否会丢失;
- 测试各个子功能组合起来，能否达到预期要求的父功能;
- 一个模块的功能是否会对另一个模块的功能产生不利的影响;
- 全局数据结构是否有问题;
- 单个模块的误差积累起来，是否会放大，从而达到不可接受的程度。

主要测试方法是黑盒测试方法。必要的集成测试是回归测试。

本文主要的读者对象是项目负责人、集成部门负责人、集成测试设计师。

1.3 术语

algorithm（算法）

- 一个定义好的有限规则集，用于在有限步骤内解决一个问题;
- 执行一个特定任务的任何操作序列。

algorithm analysis（算法分析）

一个软件的验证确认任务，用于保证选择的算法是正确的、合适的和稳定的，并且满足所有精确性、规模和时间方面的要求。

Alpha Testing（Alpha测试）

由选定的用户进行的产品早期性测试。这个测试一般在可控制的环境下进行的。

analysis（分析）

- 分解到一些原子部分或基本原则，以便确定整体的特性；
- 一个推理的过程，显示一个特定的结果是假设前提的结果；
- 一个问题的方法研究，并且问题被分解为一些小的相关单元作进一步详细研究。

anomaly（异常）

在文档或软件操作中观察到的任何与期望违背的结果。

application software（应用软件）

满足特定需要的软件。

architecture（构架）

一个系统或组件的组织结构。

ASQ（自动化软件质量（Automated Software Quality））

使用软件工具来提高软件的质量。

audit（审计）

一个或一组工作产品的独立检查以评价与规格、标准、契约或其它准则的符合程度。

audit trail（审计跟踪）

系统审计活动的一个时间记录。

Automated Testing（自动化测试）

使用自动化测试工具来进行测试，这类测试一般不需要人干预，通常在GUI、性能等测试中用得较多。

1.4 测试环境

序号	描述	配置
1	浏览器	edge, Chorme
2	输入习惯	中文
3	操作系统环境	Windows 11
4	测试工具	Selenium, Apifox
5	数据库	MySQL

2. 集成策略

2.1 进入标准

编码完成，单元测试完成。集成测试计划完成，时间表、工具及其人员安排到位。

2.2 集成内容

1. 函数集成

如函数间接口、函数是否调用正常。

2. 功能集成

如不同函数间实现的业务功能。

3. 数据集成

如数据传递是否正确，对于传入值的控制范围是否一致等。

4. 子系统集成

如把不同通信子系统、业务子系统及报表子系统进行集成。

2.3 集成策略

本系统的集成测试采用自底向上的集成（Bottom-Up Integration）方式。自底向上集成方式从程序模块结构中最底层的模块开始组装和测试。因为模块是自底向上进行组装的，对于一个给定层次的模块，它的子模块(包括子模块的所有下属模块)事前已经完成组装并经过测试，所以不再需要编制桩模块。选择这种集成方法，管理方便，测试人员能较好地锁定软件故障所在位置。

集成测试中的主要步骤：

- 制订并审核集成测试计划。
- 测试用例分析、设计及评审。
- 测试的实施。
- 测试的执行。
- 测试的分析和评估。

2.4 集成顺序

1. 软件集成顺序

自底向上，先函数、数据、功能再子系统。

2. 软件/硬件集成顺序

无。

3. 测试过程描述

3.1 软件集成测试

4. 集成测试验收标准

4.1 模块验收标准

- 接口：接口提供的功能或者数据正确。
- 功能点：验证程序与产品描述、用户文档中的全部说明相对应，一致性。
- 流程处理：验证程序与产品描述、用户文档中的全部说明相对应，一致性。
- 外部接口：验证程序与产品描述、用户文档中的全部说明相对应，一致性。

4.2 集成测试验收标准

首先，集成测试用例中所设计的功能测试用例必须全部通过,性能及其他类型测试用例95%以上通过。在未通过的测试用例中，不能含有“系统崩溃”和“严重错误”这两种错误，“一般错误”小于1%。测试结果与测试用例中期望的结果一致，测试通过，否则表明测试未通过。

5. 测试工具

5.1 测试工具

测试中心平台：JUnit

性能测试工具：JMeter

集成测试工具：Python、Apifox

5.2 其他工具

6. 挂起、恢复和退出条件

6.1 挂起

- 进入第一轮集成测试，测试人员大体了解一下产品情况，如果发现在单元内存在三个以上错误或缺陷以及操作性的错误，退回单元测试组测试；
- 遇到有项目优先级更高的集成测试任务；
- 在复测过程中发现产品无法运行下去；
- 人员、设备不足；
- 重大突发紧急情况。

6.2 恢复

- 符合进入集成测试条件；
- 项目优先级更高的集成测试任务暂告完成；
- 复测过程中产品可以运行下去；
- 人员、设备到位；

- 突发事件处理完成；
- 退出；
- 项目因故终止；
- 不可抗力：合同专用条款中约定等级以上的自然灾害也属不可抗力；
- 其他原因的测试工作频频被挂起或者挂起后迟迟恢复不了，并超过了客户要求的期限。

7. 责任人和时间表

7.1 责任人

测试负责人：孙致远

控制并完成测试任务和测试过程，决定测试人员提交上来的Bug是否需要修改。

测试设计人员：冯小东 孙致远

设计集成测试用例。

测试人员：冯小东 孙致远 陈宇田 曹晓慈

按照测试用例进行测试活动。

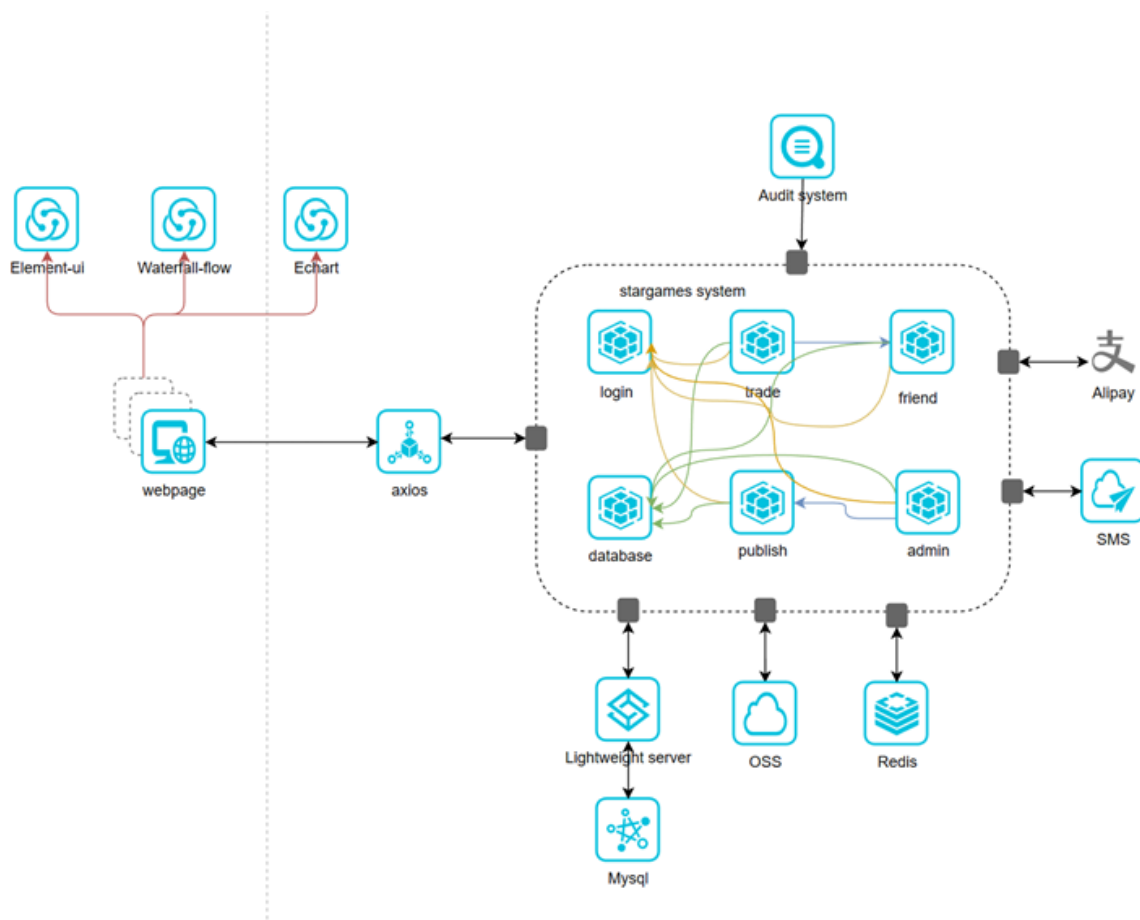
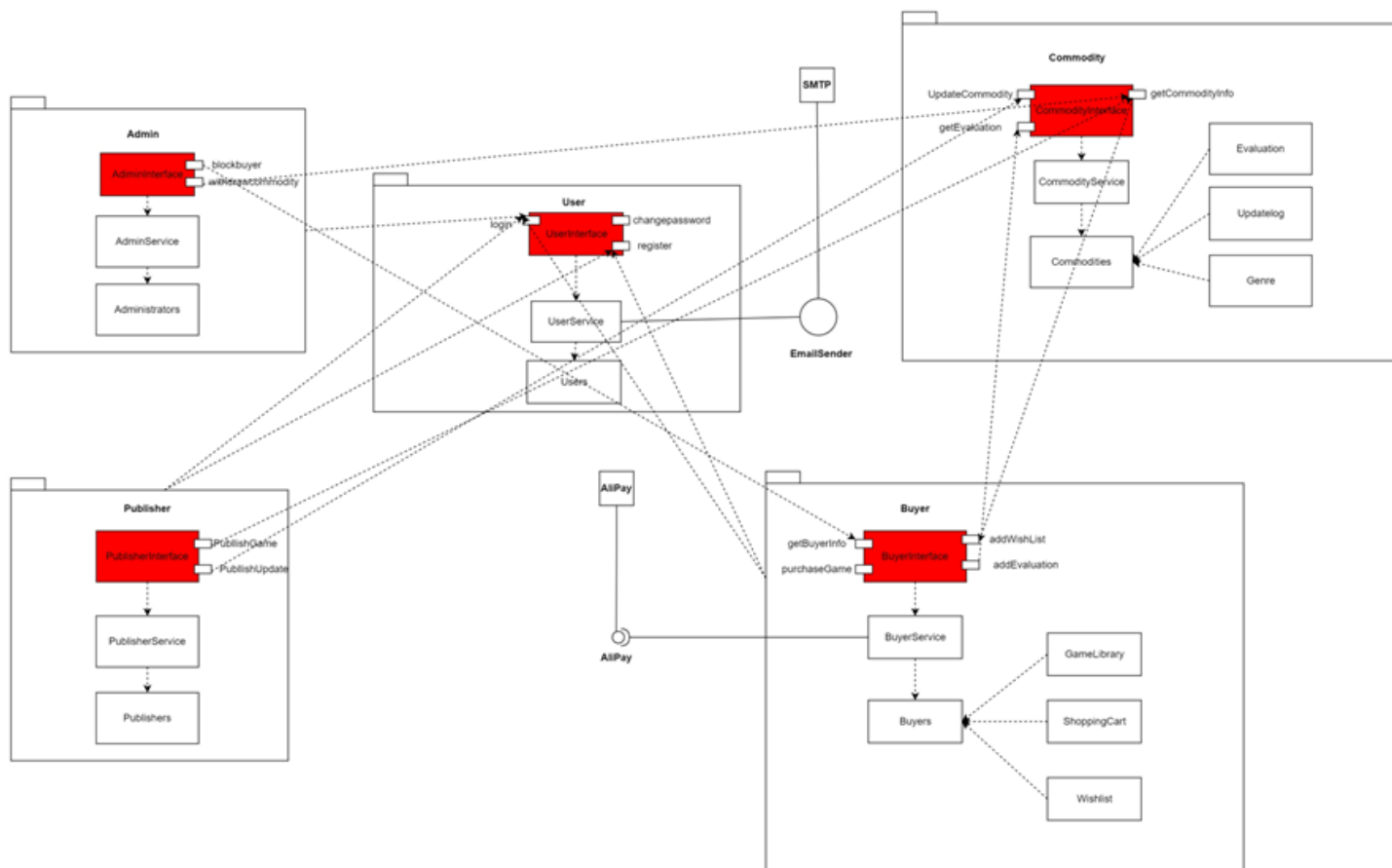
开发人员：陈宇田 曹晓慈

程序Bug修改，开发人员之间协调。

8. 集成测试依据

集成测试的依据是概要设计规约。

下图是被测系统的前端和后端接口、后端类方法之间的接口及与第三方系统的接口关系图，基于此图中的关系，对系统进行集成测试。



集成测试报告

1. 测试内容

- 对Service集成为Controller的集成测试

包括以下10个接口，测试工具：Apifox

序号	名称	简述
1	CommodityInfo(String CommodityID)	获取商品信息
2	AddToGameLibrary(String BuyerID,String CommodityID)	添加商品到游戏库
3	AddEvaluation(String CommodityID,String UserID,String Evaluation)	为游戏添加评测
4	checkIdentity(String UserID, String Password)	检查登录者身份
5	GetCommodityInfo(String CommodityID)	获取游戏商品信息
6	Search_By_name(String name)	搜索商品
7	AddShoppingCart(String CommodityID,String UserID)	添加商品到购物车
8	RemoveShoppingCartItems(String BuyerID,String CommodityID)	移除购物车中商品
9	AddWishList(String CommodityID,String UserID)	添加商品到愿望单
10	RemoveWishListItems(String BuyerID,String CommodityID)	移除愿望单中商品

- 子系统（Controller）之间调用接口的集成测试

包括以下4个接口，测试工具：Python脚本

序号	名称	简述	集成的类
1	GetSellerInfo(String SellerID)	获取商家信息	Publisher、Commodity
2	getAllGenres()	获取种类信息	Genres、Commodity
3	GetCommodityFullInfo(String CommodityID)	获取商品完整的信息列表	Buyer、Commodity
4	GetBuyerInfo(String BuyerID)	获取买家详细信息	Friend、Buyer

2. 测试方法

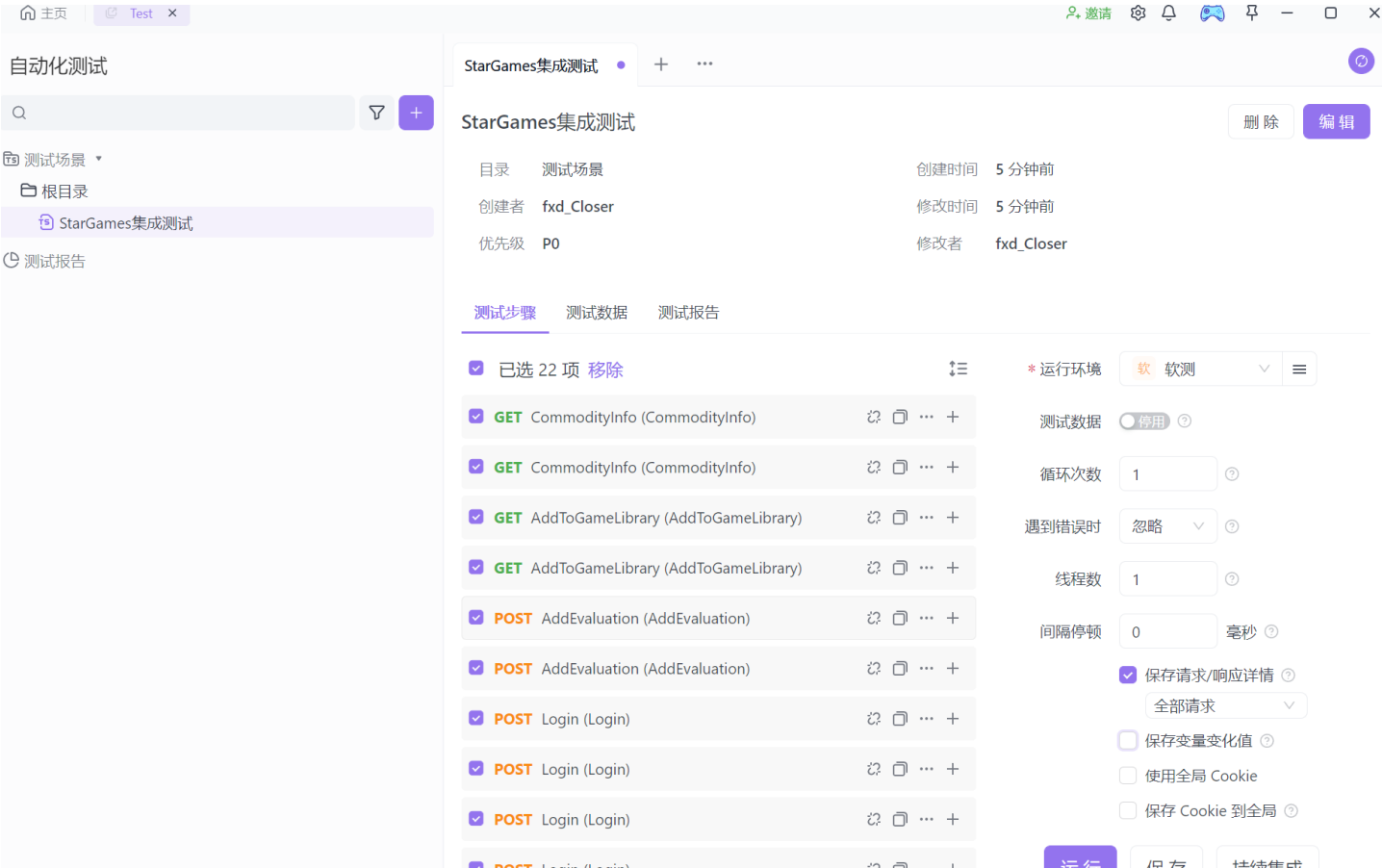
2.1 Python脚本

编写python测试脚本，对测试用例接口进行测试，核心代码如下：

```
1 import requests
2 import pandas as pd
3 urlBase = 'http://124.70.163.146:8080/api/v1'
4
5 def getRes():
6     res = {}
7     res['SG_Int_001_001'] = requests.get(url=urlBase + '/CommodityInfoService/Co
8     # omitted
9     return res
10
11
12 total = 0
13 success = 0
14 fail = 0
15 df = pd.read_excel('集成测试.xlsx')
16 print(df)
17 df.insert(loc=2, column='real', value='')
18 df.insert(loc=3, column='result', value='')
19 res = getRes()
20 for i in range(df.shape[0]):
21     total += 1
22     df.loc[i, 'real'] = res[str(df.loc[i, 'id'])]
23     if str(df.loc[i, 'real']) != str(df.loc[i, 'expect']):
24         df.loc[i, 'result'] = "未通过测试"
25         fail += 1
26     else:
27         df.loc[i, 'result'] = "通过测试"
28         success += 1
29 df.to_csv('result.csv')
30 print('共测试了', total, '个测试用例, 成功:', success, '失败:', fail)
```

2.2 Apifox自动化测试

使用Apifox的“自动化测试”对接口进行集成测试，分别根据等价类进行无效等价类和有效等价类的测试，并得出测试报告。



3. 测试用例分析和设计

采用等价类划分的方法；按照变量输出的结果进行等价类划分，分为有效等价类和无效等价类。除此以外，参考使用边界值分析法、错误猜测方法和覆盖分析。

经过讨论，我们选定了14个比较复杂、业务重要的接口，进行作为测试用例，进行集成测试，接口列表如下

被测接口及标识表如下所示：

序号	标识符	名称	简述
1	SG_Int_001	CommodityInfo(String CommodityID)	获取商品信息
2	SG_Int_002	AddToGameLibrary(String BuyerID,String CommodityID)	添加商品到游戏库
3	SG_Int_003	AddEvaluation(String CommodityID,String UserID,String Evaluation)	为游戏添加评测
4	SG_Int_004	checkIdentity(String UserID, String Password)	检查登录者身份
5	SG_Int_005	GetCommodityInfo(String CommodityID)	获取游戏商品信息
6	SG_Int_006	Search_By_name(String name)	搜索商品
7	SG_Int_007	AddShoppingCart(String CommodityID,String UserID)	添加商品到购物车

8	SG_Int_008	RemoveShoppingCartItems(String BuyerID,String CommodityID)	移除购物车中商品
9	SG_Int_009	AddWishList(String CommodityID,String UserID)	添加商品到愿望单
10	SG_Int_010	RemoveWishListItems(String BuyerID,String CommodityID)	移除愿望单中商品

序号	标识符	名称	简述
1	SG_Int_011	GetSellerInfo(String SellerID)	获取商家信息
2	SG_Int_012	getAllGenres()	获取种类信息
3	SG_Int_013	GetCommodityFullInfo(String CommodityID)	获取商品完整的信息
4	SG_Int_014	GetBuyerInfo(String BuyerID)	获取买家详细信息

3.1 SG_Int_001 测试分析与设计

3.1.1 概述

标识符：SG_Int_001

名称：CommodityInfo

简述：获取商品信息

3.1.2 测试项标识

测试项标识符	等价类类型划分	测试项描述	优先级
SG_Int_001_001	有效等价类	测试ID包含在数据库中的情况	高
SG_Int_001_002	无效等价类	测试ID不包含在数据库中的情况	中

3.1.3 测试用例

测试用例1

	A	B	C
1	测试项编号	SG_Int_001_001	
2	优先级	高	
3	测试项描述	测试ID包含在数据库中的情况	
4	前置条件	无	
5	用例序号	输入	期望结果
6	1	CommodityID == 100001	<pre>{ "COMMODITY_ID": "100001", "COMMODITY_NAME": "God of War: Ragnarok", "PRICE": 568.00, "PUBLISH_TIME": "2022-12-31T16:33:18", "DESCRIPTION": "芬布尔之冬正在肆虐。奎托斯和阿特柔斯必须探索九界各界寻找真相，阿斯加德大军也为迎接预言中的末日大战全力备战。一路上，他们将探索令人惊叹的神话景观，并面对各种各样的敌对生物、怪物和北欧诸神。诸神黄昏劫难迫近。", "PICTURE_URL": "https://gimg2.baidu.com/image_search/src=http%3A%2F%2Fn.sinaimg.cn%2Fsinakd20220119ac%2F650%2Fw1500h750%2F20220119%2Fb724-7fc72986eb63b24bdd742624306a36e9.jpg&refer=http%3A%2F%2Fn.sinaimg.cn&app=2002&size=f9999,10000&q=a80&n=0&g=0n&fmt=auto?sec=1672390858&t=4f923bd5e3c31f17a65b7a00bea35852", "DOWNLOAD_URL": null, "STATUS": 1 }</pre>

测试用例2

	A	B	C
1	测试项编号	SG_Int_001_002	
2	优先级	中	
3	测试项描述	测试ID不包含在数据库中的情况	
4	前置条件	无	
5	用例序号	输入	期望结果
6	1	CommodityID == 100045	{"STATUS":-1}

3.2 SG_Int_002 测试分析与设计

3.2.1 概述

标识符：SG_Int_002

名称：AddToGameLibrary

简述：添加商品到游戏库

3.2.2 测试项标识

测试项标识符	等价类类型划分	测试项描述	优先级
SG_Int_002_001	有效等价类	测试商品未加入游戏库中的情况	高
SG_Int_002_002	无效等价类	测试商品已加入游戏库中的情况	中

3.2.3 测试用例

测试用例1

	A	B	C
1	测试项编号	SG_Int_002_001	
2	优先级	高	
3	测试项描述	测试商品未加入游戏库中的情况	
4	前置条件	用户以买家身份登录	
5	用例序号	输入	期望结果
6	1	BuyerID==10002 ,CommodityID ==100006	{ "STATUS": 1 }

测试用例2

	A	B	C
1	测试项编号	SG_Int_002_002	
2	优先级	中	
3	测试项描述	测试商品已经加入游戏库中的情况	
4	前置条件	用户以买家身份登录	
5	用例序号	输入	期望结果
6	1	BuyerID==10001 ,CommodityID == 100001	{ "STATUS": -1, "REASON": "您已经 拥有该游戏" }

3.3 SG_Int_003 测试分析与设计

3.3.1 概述

标识符：SG_Int_003

名称：AddEvaluation

简述：为游戏添加评测

3.3.2 测试项标识

测试项标识符	等价类类型划分	测试项描述	优先级
SG_Int_003_001	有效等价类	测试商品未评测的情况	高
SG_Int_003_002	无效等价类	测试商品已经评测的情况	高

3.3.3 测试用例

测试用例1

	A	B	C
1	测试项编号	SG_Int_003_001	
2	优先级	高	
3	测试项描述	测试商品未评测的情况	
4	前置条件	用户以买家身份登录	
5	用例序号	输入	期望结果
6	1	UserID==10001,CommodityID==100005,Evaluation==这个游戏好好玩	{ "STATUS": 1 }

测试用例2

	A	B	C
1	测试项编号	SG_Int_003_002	
2	优先级	高	
3	测试项描述	测试商品已经评测的情况	
4	前置条件	用户以买家身份登录	
5	用例序号	输入	期望结果
6	1	UserID==10001,CommodityID==100001,Evaluation==这个游戏好好玩	{ "STATUS": 1 }

3.4 SG_Int_004 测试分析与设计

3.4.1 概述

标识符：SG_Int_004

名称：checkIdentity

简述：检查登录者身份

3.4.2 测试项标识

测试项标识符	等价类类型划分	测试项描述	优先级
SG_Int_004_001	有效等价类	测试用户作为买家登录的情况	高
SG_Int_004_002	有效等价类	测试用户作为卖家登录的情况	高
SG_Int_004_003	有效等价类	测试用户作为管理员登录的情况	高
SG_Int_004_004	有效等价类	测试用户名密码错误的情况	高

3.4.3 测试用例

测试用例1

	A	B	C
1	测试项编号	SG_Int_004_001	
2	优先级	高	
3	测试项描述	测试用户作为买家登录的情况	
4	前置条件	已注册	
5	用例序号	输入	期望结果
6	1	UserID==10001, Password==123 456	0

测试用例2

	A	B	C
1	测试项编号	SG_Int_004_002	
2	优先级	高	
3	测试项描述	测试用户作为卖家登录的情况	
4	前置条件	已注册	
5	用例序号	输入	期望结果
6	1	UserID==10006, Password==123 456	1

测试用例3

	A	B	C
1	测试项编号	SG_Int_004_003	
2	优先级	高	
3	测试项描述	测试用户作为管理员登录的情况	
4	前置条件	已注册	
5	用例序号	输入	期望结果
6	1	UserID==10008, Password==123 456	2

测试用例4

	A	B	C
1	测试项编号	SG_Int_004_004	
2	优先级	高	
3	测试项描述	测试用户用户名密码错误的情况	
4	前置条件	已注册	
5	用例序号	输入	期望结果
6	1	UserID==40008, Password==123 4231asda	2

3.5 SG_Int_005 测试分析与设计

3.5.1 概述

标识符：SG_Int_005

名称：GetCommodityInfo

简述：获取游戏商品信息

3.5.2 测试项标识

测试项标识符	等价类类型划分	测试项描述	优先级
SG_Int_005_001	有效等价类	测试商品存在的情况情况	高
SG_Int_005_002	无效等价类	测试商品不存在的情况	中

3.5.3 测试用例

测试用例1

	A	B	C
1	测试项编号	SG_Int_005_001	
2	优先级	高	
3	测试项描述	测试商品存在的情况情况	
4	前置条件	用户以买家身份登录	
5	用例序号	输入	期望结果
6	1	CommodityID==100001	<pre>{ "COMMODITY_ID": "100001", "COMMODITY_NAME": "God of War: Ragnarok", "PRICE": 568.00, "PUBLISH_TIME": "2022-12-31T16:33:18", "DESCRIPTION": "芬布尔之冬正在肆虐。奎托斯和阿特柔斯必须探索九界各界寻找真相，阿斯加德大军也为迎接预言中的末日大战全力备战。一路上，他们将探索令人惊叹的神话景观，并面对各种各样的敌对生物、怪物和北欧诸神。诸神黄昏劫难迫近。", "PICTURE_URL": "https://gimg2.baidu.com/image_search/src=http%3A%2F%2Fn.sinaimg.cn%2Fsinakd20220119ac%2F650%2Fw1500h750%2F20220119%2Fb724-7fc72986eb63b24bdd742624306a36e9.jpg&refer=http%3A%2F%2Fn.sinaimg.cn&app=2002&size=f9999,10000&q=a80&n=0&g=0n&fmt=auto?sec=1672390858&t=4f923bd5e3c31f17a65b7a00bea35852", "DOWNLOAD_URL": null, "STATUS": 1 }</pre>

	A	B	C
1	测试项编号	SG_Int_005_002	
2	优先级	中	
3	测试项描述	测试商品存在的情况情况	
4	前置条件	用户以买家身份登录	
5	用例序号	输入	期望结果
6	1	CommodityID==105423	{"STATUS":-1}

3.6 SG_Int_006 测试分析与设计

3.6.1 概述

标识符：SG_Int_006

名称：GSearch_By_name

简述：搜索商品

3.6.2 测试项标识

测试项标识符	等价类类型划分	测试项描述	优先级
SG_Int_006_001	有效等价类	测试商品存在的情况情况	高
SG_Int_006_002	无效等价类	测试商品不存在的情况	中

3.6.3 测试用例

测试用例1

	A	B	C
1	测试项编号	SG_Int_006_001	
2	优先级	高	
3	测试项描述	测试商品存在的情况情况	
4	前置条件	用户以买家身份登录	
5	用例序号	输入	期望结果
6	1	name==原神	[[{"COMMODITY_ID":"100016","COMMODITY_NAME":"原神","PRICE":648.00,"PUBLISH_TIME":"2022-12-31T22:13:36","DESCRIPTION":"excellent","PICTURE_URL":" https://stargames-1314598070.cos.ap-nanjing.myqcloud.com/pic/t7dxy8lowi.jpg ","DOWNLOAD_URL":" https://www.bilibili.com/ ","GENRE":["恐怖类"]}]]

测试用例2

	A	B	C
1	测试项编号	SG_Int_006_002	
2	优先级	中	
3	测试项描述	测试商品存在的情况情况	
4	前置条件	用户以买家身份登录	
5	用例序号	输入	期望结果
6	1	name==你好	[]

3.7 SG_Int_007 测试分析与设计

3.7.1 概述

标识符：SG_Int_007

名称：AddShoppingCart

简述：添加商品到购物车

3.7.2 测试项标识

测试项标识符	等价类类型划分	测试项描述	优先级
SG_Int_007_001	有效等价类	测试商品未被添加到购物车的情况	高
SG_Int_007_002	无效等价类	测试购物车中已有该商品的情况	中

3.7.3 测试用例

测试用例1

	A	B	C
1	测试项编号	SG_Int_007_001	
2	优先级	高	
3	测试项描述	测试商品未被添加到购物车的情况	
4	前置条件	用户以买家身份登录	
5	用例序号	输入	期望结果
6	1	UserID==10001, CommodityID==100002	{"STATUS":1}

测试用例2

	A	B	C
1	测试项编号	SG_Int_007_002	
2	优先级	中	
3	测试项描述	测试购物车中已有该商品的情况	
4	前置条件	用户以买家身份登录	
5	用例序号	输入	期望结果
6	1	UserID==10001, CommodityID==100013	{"STATUS":-1,"REASON":"您的购物车当中已经有该游戏"}

3.8 SG_Int_008 测试分析与设计

3.8.1 概述

标识符：SG_Int_008

名称：RemoveShoppingCartItems

简述：移除购物车中商品

3.8.2 测试项标识

测试项标识符	等价类类型划分	测试项描述	优先级
SG_Int_008_001	有效等价类	测试购物车中有该商品的情况	高
SG_Int_008_002	无效等价类	测试购物车中无该商品的情况	中

3.8.3 测试用例

测试用例1

	A	B	C
1	测试项编号	SG_Int_008_001	
2	优先级	高	
3	测试项描述	测试购物车中有该商品的情况	
4	前置条件	用户以买家身份登录	
5	用例序号	输入	期望结果
6	1	BuyerID==10001 ,CommodityID= =100002	TRUE

测试用例2

	A	B	C
1	测试项编号	SG_Int_008_002	
2	优先级	中	
3	测试项描述	测试购物车中无该商品的情况	
4	前置条件	用户以买家身份登录	
5	用例序号	输入	期望结果
6	1	BuyerID==10001 ,CommodityID= =100010	FALSE

3.9 SG_Int_009 测试分析与设计

3.9.1 概述

标识符：SG_Int_009

名称：AddWishList

简述：添加商品到愿望单

3.9.2 测试项标识

测试项标识符	等价类类型划分	测试项描述	优先级
SG_Int_009_001	有效等价类	测试商品未被添加到愿望单的情况	高
SG_Int_009_002	无效等价类	测试愿望单中已有该商品的情况	中

3.9.3 测试用例

测试用例1

	A	B	C
1	测试项编号	SG_Int_009_001	
2	优先级	高	
3	测试项描述	测试商品未被添加到愿望单的情况	
4	前置条件	用户以买家身份登录	
5	用例序号	输入	期望结果
6	1	UserID==10001,CommodityID==100006	{"STATUS":1}

测试用例2

	A	B	C
1	测试项编号	SG_Int_009_002	
2	优先级	中	
3	测试项描述	测试愿望单中已有该商品的情况	
4	前置条件	用户以买家身份登录	
5	用例序号	输入	期望结果
6	1	UserID==10001,CommodityID==100013	{"STATUS":-1,"REASON":"您的购物车当中已经有该游戏"}

3.10 SG_Int_010 测试分析与设计

3.10.1 概述

标识符：SG_Int_010

名称：RemoveWishListItems

简述：移除愿望单中商品

3.10.2 测试项标识

测试项标识符	等价类类型划分	测试项描述	优先级
SG_Int_010_001	有效等价类	测试愿望单中有该商品的情况	高
SG_Int_010_002	无效等价类	测试愿望单中无该商品的情况	中

3.10.3 测试用例

测试用例1

	A	B	C
1	测试项编号	SG_Int_010_001	
2	优先级	高	
3	测试项描述	测试愿望单中有该商品的情况	
4	前置条件	用户以买家身份登录	
5	用例序号	输入	期望结果
6	1	BuyerID==10001 ,CommodityID= =100006	TRUE

测试用例2

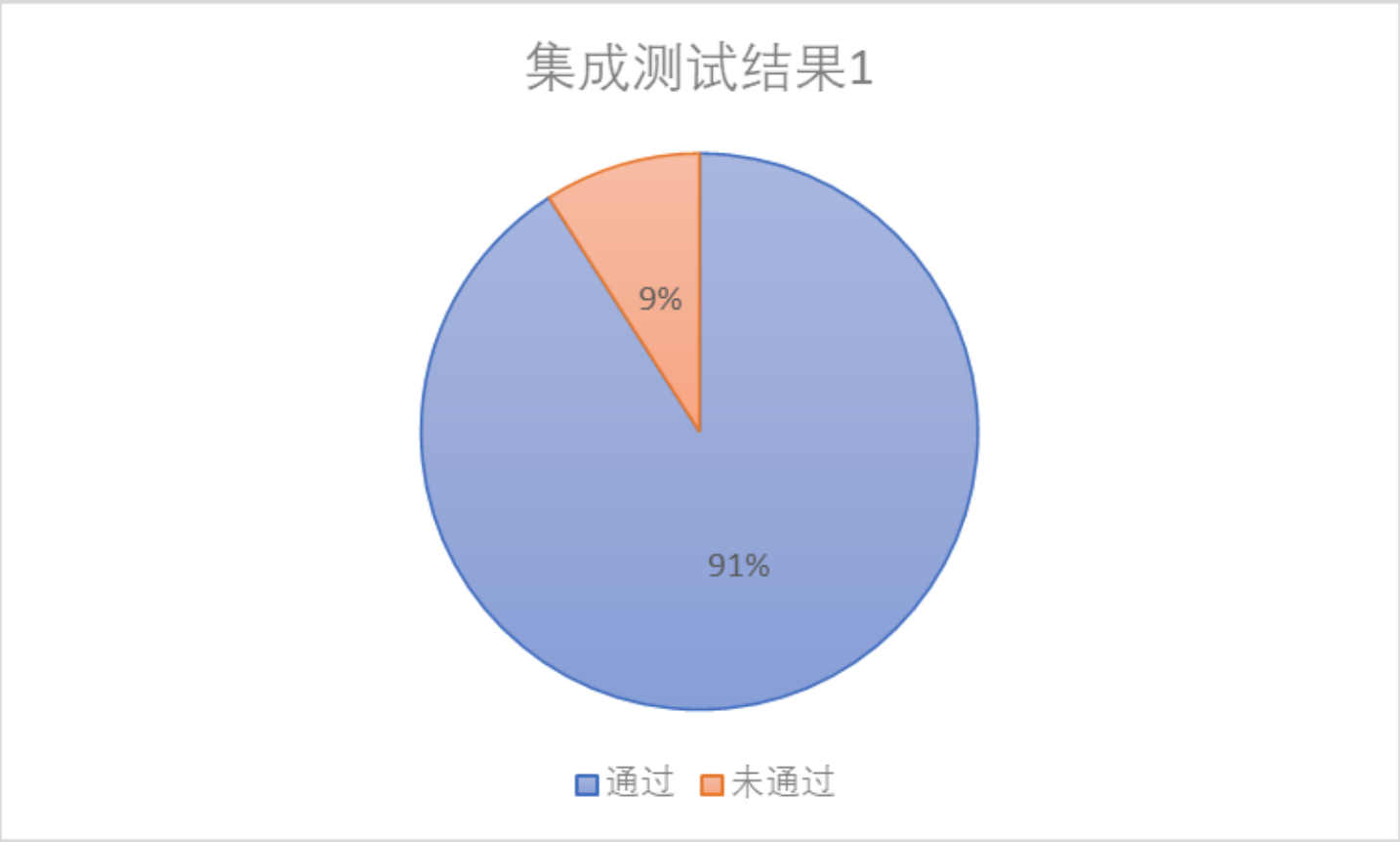
	A	B	C
1	测试项编号	SG_Int_010_002	
2	优先级	中	
3	测试项描述	测试愿望单中无该商品的情况	
4	前置条件	用户以买家身份登录	
5	用例序号	输入	期望结果
6	1	BuyerID==10001 ,CommodityID= =100024	FALSE

4. 测试结果

4.1 子系统内集成测试结果

4.1.1 第一次测试：

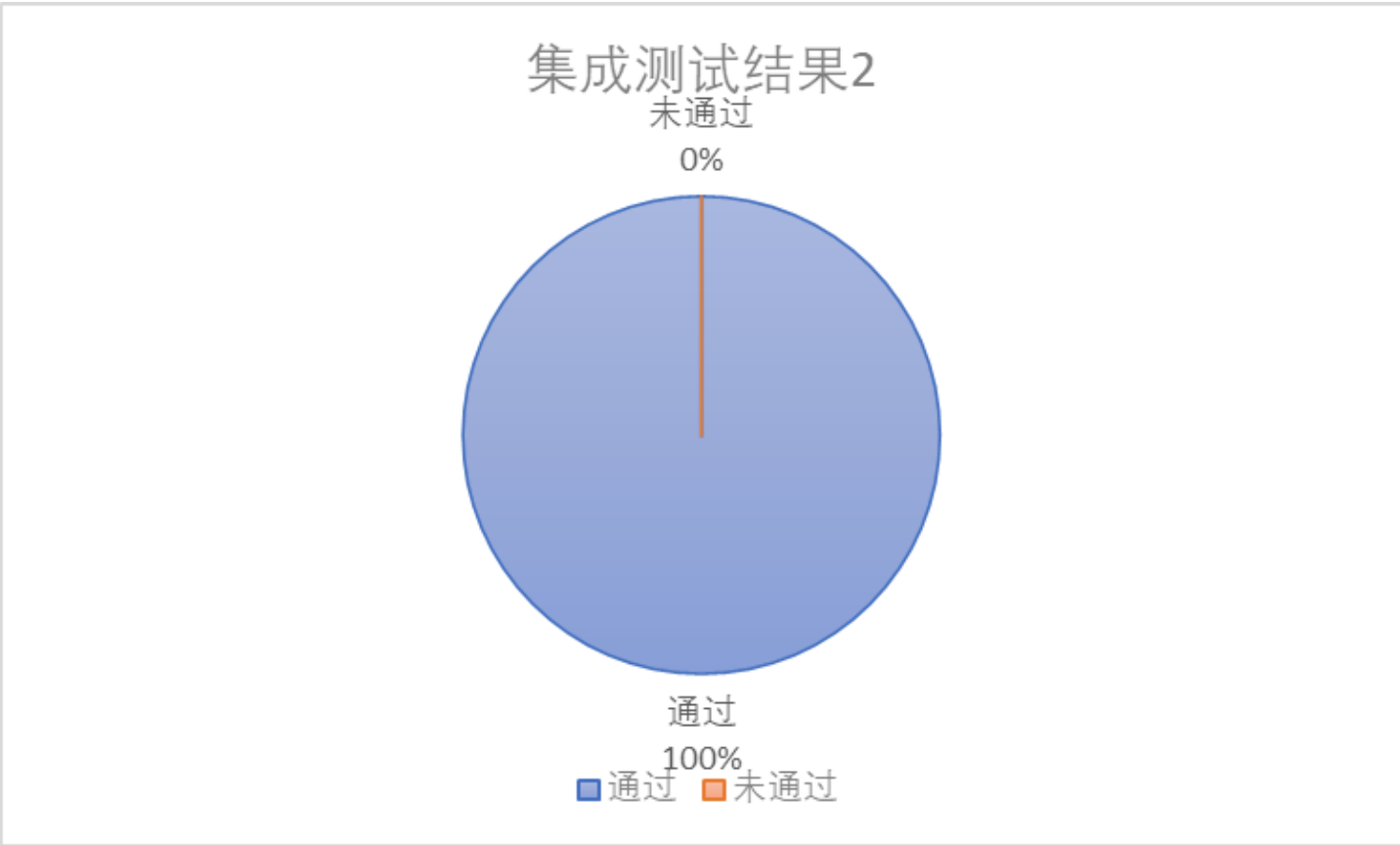
Column1	id	expect	real	result
	0 SG_Int_001_001	{ "COMMODITY_ID": "100001", "COMMODITY_NAME": "God of War: Ragnarok", "PRICE": 5 }	{ "COMMODITY_ID": "100001", "COMMODITY_NAME": "God of War: Ragnarok", "PRICE": 5 }	通过测试
	1 SG_Int_001_002	{ "STATUS": -1 }	{ "STATUS": -1 }	通过测试
	2 SG_Int_002_001	{ "STATUS": 1 }	{ "STATUS": -1, "REASON": "您已经拥有该游戏" }	未通过测试
	3 SG_Int_002_002	{ "STATUS": -1, "REASON": "您已经拥有该游戏" }	{ "STATUS": -1, "REASON": "您已经拥有该游戏" }	通过测试
	4 SG_Int_003_001	{ "STATUS": 1 }	{ "STATUS": -1, "REASON": "您已经评测过该游戏" }	未通过测试
	5 SG_Int_003_002	{ "STATUS": -1, "REASON": "您已经评测过该游戏" }	{ "STATUS": -1, "REASON": "您已经评测过该游戏" }	通过测试
	6 SG_Int_004_001	0	0	通过测试
	7 SG_Int_004_002	1	1	通过测试
	8 SG_Int_004_003	2	2	通过测试
	9 SG_Int_004_004	-1	-1	通过测试
	10 SG_Int_005_001	{ "COMMODITY_ID": "100001", "COMMODITY_NAME": "God of War: Ragnarok", "PRICE": 5 }	{ "COMMODITY_ID": "100001", "COMMODITY_NAME": "God of War: Ragnarok", "PRICE": 5 }	通过测试
	11 SG_Int_005_002	{ "STATUS": -1 }	{ "STATUS": -1 }	通过测试
	12 SG_Int_006_001	[{ "COMMODITY_ID": "100016", "COMMODITY_NAME": "原神", "PRICE": 648.00, "PUBLISH_T": 1 }]	[{ "COMMODITY_ID": "100016", "COMMODITY_NAME": "原神", "PRICE": 648.00, "PUBLISH_T": 1 }]	通过测试
	13 SG_Int_006_002	[]	[]	通过测试
	14 SG_Int_007_001	{ "STATUS": 1 }	{ "STATUS": 1 }	通过测试
	15 SG_Int_007_002	{ "STATUS": -1, "REASON": "您的购物车当中已经有该游戏" }	{ "STATUS": -1, "REASON": "您的购物车当中已经有该游戏" }	通过测试
	16 SG_Int_008_001	true	true	通过测试
	17 SG_Int_008_002	false	false	通过测试
	18 SG_Int_009_001	{ "STATUS": 1 }	{ "STATUS": 1 }	通过测试
	19 SG_Int_009_002	{ "STATUS": -1, "REASON": "您的愿望单当中已经有该游戏" }	{ "STATUS": -1, "REASON": "您的愿望单当中已经有该游戏" }	通过测试
	20 SG_Int_010_001	true	true	通过测试
	21 SG_Int_010_002	false	false	通过测试



共测试了22个用例，其中20个通过测试，2个未通过测试，通过率为91%。这次集成测试，我们发现了2个缺陷，分别为SG_Int_002_001、SG_Int_003_001对应的缺陷，并对他们进行了缺陷跟踪。

4.1.2 第二次测试：

Column1	id	expect	real	result
0	SG_Int_001_001	{ "COMMODITY_ID": "100001", "COMMODITY_NAME": "God of War: Ragnarok", "PRICE": 56 }	{ "COMMODITY_ID": "100001", "COMMODITY_NAME": "God of War: Ragnarok", "PRICE": 56 }	通过测试
1	SG_Int_001_002	{ "STATUS": -1 }	{ "STATUS": -1 }	通过测试
2	SG_Int_002_001	{ "STATUS": 1 }	{ "STATUS": 1 }	通过测试
3	SG_Int_002_002	{ "STATUS": -1, "REASON": "您已经拥有该游戏" }	{ "STATUS": -1, "REASON": "您已经拥有该游戏" }	通过测试
4	SG_Int_003_001	{ "STATUS": 1 }	{ "STATUS": 1 }	通过测试
5	SG_Int_003_002	{ "STATUS": -1, "REASON": "您已经评测过该游戏" }	{ "STATUS": -1, "REASON": "您已经评测过该游戏" }	通过测试
6	SG_Int_004_001	0	0	通过测试
7	SG_Int_004_002	1	1	通过测试
8	SG_Int_004_003	2	2	通过测试
9	SG_Int_004_004	-1	-1	通过测试
10	SG_Int_005_001	{ "COMMODITY_ID": "100001", "COMMODITY_NAME": "God of War: Ragnarok", "PRICE": 56 }	{ "COMMODITY_ID": "100001", "COMMODITY_NAME": "God of War: Ragnarok", "PRICE": 56 }	通过测试
11	SG_Int_005_002	{ "STATUS": -1 }	{ "STATUS": -1 }	通过测试
12	SG_Int_006_001	[{ "COMMODITY_ID": "100016", "COMMODITY_NAME": "原神", "PRICE": 648.00, "PUBLISH_T": 1677177600 }]	[{ "COMMODITY_ID": "100016", "COMMODITY_NAME": "原神", "PRICE": 648.00, "PUBLISH_T": 1677177600 }]	通过测试
13	SG_Int_006_002	[]	[]	通过测试
14	SG_Int_007_001	{ "STATUS": 1 }	{ "STATUS": 1 }	通过测试
15	SG_Int_007_002	{ "STATUS": -1, "REASON": "您的购物车当中已经有该游戏" }	{ "STATUS": -1, "REASON": "您的购物车当中已经有该游戏" }	通过测试
16	SG_Int_008_001	true	true	通过测试
17	SG_Int_008_002	false	false	通过测试
18	SG_Int_009_001	{ "STATUS": 1 }	{ "STATUS": 1 }	通过测试
19	SG_Int_009_002	{ "STATUS": -1, "REASON": "您的愿望单当中已经有该游戏" }	{ "STATUS": -1, "REASON": "您的愿望单当中已经有该游戏" }	通过测试
20	SG_Int_010_001	true	true	通过测试
21	SG_Int_010_002	false	false	通过测试



共测试了22个用例，其中22个通过测试，0个未通过测试，通过率为100%。设计的测试用例全部通过测试，达到了测试的验收标准之一。

4.1.3 Apifox自动化测试报告

apifox可以导出html形式的测试报告

详细的测试报告见文件"集成测试_Apifox1.html"。

Apifox 报告

测试场景	StarGames集成测试	
运行时间	2023-06-14 21:08:12	
运行工具	Apifox v2.2.40	
	总数	失败数
循环数	1	0
HTTP 接口请求数	22	0
断言数	0	0
总耗时	3.62s	
总返回数据	569B	
平均接口请求耗时	25ms	
通过率	100.00%	
失败率	0.00%	
未测率	0.00%	

接口请求情况

▼ StarGames集成测试

AddToGameLibrary(AddToGameLibrary)	
Method	GET
URL	http://124.70.163.146:8080/api/v1/GameLibService/AddToGameLibrary?BuyerID=10001&CommodityID=100010
平均耗时	30ms
平均返回数据	49B

4.2 子系统间集成测试结果

共测试了 9 个测试用例,成功: 9 失败: 0

可以看到，所有测试用例都通过。

	🔍 C1 ⚙	🔍 C2 ⚙	🔍 C3 ⚙	🔍 C4 ⚙	🔍 C5 ⚙
1	<null>	id	expect	real	result
2	0	SG_Int_011_001	{"PUBLISHER_ID":"10006","PUBLISH...	{"PUBLISHER_ID":"10006","PUBLISHER_NAME":"TJGames","S...	通过测试
3	1	SG_Int_011_002	{"PUBLISHER_ID":"10007","PUBLISH...	{"PUBLISHER_ID":"10007","PUBLISHER_NAME":"Microsoft",...	通过测试
4	2	SG_Int_012_001	[{"GENRE_ID":1,"GENRE_NAME":"角色扮...	[{"GENRE_ID":1,"GENRE_NAME":"角色扮演类"}, {"GENRE_ID":2,	通过测试
5	3	SG_Int_013_001	{"COMMODITY_ID":"100001","COMMODO...	{"COMMODITY_ID":"100001","COMMODITY_NAME":"God of War...	通过测试
6	4	SG_Int_013_002	{"COMMODITY_ID":"100002","COMMODO...	{"COMMODITY_ID":"100002","COMMODITY_NAME":"Elden Ring...	通过测试
7	5	SG_Int_013_003	{"COMMODITY_ID":"100003","COMMODO...	{"COMMODITY_ID":"100003","COMMODITY_NAME":"地平线2: 禁忌之...	通过测试
8	6	SG_Int_014_001	{"USER_ID":"10001","USER_PHONE":...	{"USER_ID":"10001","USER_PHONE":"1310636528","BUYER_N...	通过测试
9	7	SG_Int_014_002	{"USER_ID":"10002","USER_PHONE":...	{"USER_ID":"10002","USER_PHONE":"222","BUYER_NAME":"M...	通过测试
10	8	SG_Int_014_003	{"USER_ID":"10003","USER_PHONE":...	{"USER_ID":"10003","USER_PHONE":"333","BUYER_NAME":"x...	通过测试