

软件测试计划

Tomato

2018 年 1 月

1 引言

对于后端我们做了详尽的测试，包括 Controller 正确性的测试，Service 的测试和底层数据库的测试。

1.1 系统概述

整个 test 文件包含三种 test，分别是 Controller 的 test，Model 的 test 和 Service 的 test。Controller 中的 test 又分为 Admin 的 test 和 Client 的 test。由于没有找到微信 websocket 的 test 方法，所以微信的 websocket 没有写 test。

1.2 文档概述

本文档将介绍测试环境，测试计划以及测试进度。

1.3 与其他计划的关系

测试是在接口实现中逐步完善的。与接口的计划相辅相成，可以说是写一个接口就写了一个 test。

2 软件测试环境

2.1 后端测试

软件项 我们所用的语言是 Java，使用的框架是 Spring Boot，所有的测试都是搭建在这个框架之上的，写测试的方法也是使用了这个框架集成的 test annotations。

参与组织 Tomato 后端开发组。

人员 张慧盟、宋世虹。

要执行的测试 如上所述，需要测试 Controller, Model 和 Service。

3 计划

4 总体设计

4.1 测试过程

4.2 Controller

Admin

- AdminLoginControllerTest: 测试了两种情况，Admin 登陆成功和登录失败。
- AuctionControllerTest: 测试了 Auction 的接口，包括当 get auction 的时候应该返回什么，如果发起了错误的 get auction 操作时会返回什么，post auction 的时候 Service 的更新，post auction 错误的时候返回什么等等。
- ChangePasswordControllerTest: 测试了如果 password 输入正确的更新准则和输入错误时的报错。
- CompetitionCreatorControllerTest: 测试了创建比赛，更新比赛和获得比赛具体信息时的返回值，以及上述情况 request 出错时的返回值。
- CompetitionInformationControllerTest: 测试了请求比赛信息时返回信息的正确性，和请求失败的返回值。
- CompetitionStatusAdminControllerTest: 测试了不同比赛状态时的返回值。
- DeleteCompetitionControllerTest: 测试了删除比赛时：正常删除、没有比赛和获得全部比赛出错的情况。
- GetAllCompetitionControllerTest: 测试了返回全部比赛时：正常返回、没有比赛、获得全部比赛出错的情况。
- UpdateCompetitionStatusControllerTest: 测试了更新比赛状态时的返回值：正常返回、没有比赛、后端没法 update 比赛状态、非法状态等情况。

Client

- BuyMaterialControllerTest: 测试了买东西时：买材料和买机器的正在交易、已经交易完成和交易取消的情况。
- ClientInfoControllerTest: 测试了获取和更新 team 信息时成功和失败时的情况。
- ClientLoginControllerTest: 测试了登陆成功和失败的情况。
- ClientPropertyControllerTest: 测试了 team 获取 property 时成功、失败的情况，测试了 team 生产的时候成功和失败的情况，
- CompetitionStatusControllerTest: 测试了更新比赛状态时 Client 的返回情况。
- GetAllUsersControllerTest: 测试了获取所有 team 的成功和失败的情况。
- GetProduceHistoryControllerTest: 测试了获取生产历史的成功和失败（没有 team，没有 produce 记录）的情况。
- GetTradeHistoryControllerTest: 测试了获取交易历史的成功和失败（没有 team，没有 produce 记录）的情况。
- ListenPropertyControllerTest: 测试了在 websocket 转发中监听 produce 后资产的改变。
- SellMaterialControllerTest: 测试了卖资产时成功的情况，包括各种材料和机器。
- SendToSellerControllerTest: 测试了将售卖的账单转给买方和卖方的同 team，主要测试了不同账单状态的回复情况。
- UndoTradeControllerTest: 测试了撤销交易的成功情况，主要测试了发给买方和卖方的账单。

4.3 Service

- admin.CompetitionServiceTest: 测试了创建比赛、查找比赛、更新比赛、删除比赛的接口，测试了创建队伍、更新队伍、查找队伍的接口，测试了创建机器、查找机器、更新机器的接口，测试了获得所有 Produce 信息和交易信息的接口，测试了更新 Round 的接口。
- admin.LoginServiceTest: 测试了创建 Admin、更新 Admin、查找 Admin 和删除 Admin 的接口。

- user.CompetitionServiceTest: 测试了查找 Competition 和更新 Competition 的成功和失败的情况。
- user.LoginServiceTest: 测试了查找 team、更新 team、创建 team 和删除 team 的 Service 的成功和失败情况。
- user.OrderServiceTest: 测试了创建 Produce、查找 Produce、更新 Produce、删除 Produce 的接口, 测试了创建 Trade、查找 Trade、更新 Trade、删除 Trade 的接口。

4.4 Model

- AdminTest: 测试了 Admin 表的各 field。
- CompetitionTest: 测试了 Competition 表的各 field, 包括 Round。
- MachineTest: 测试了 Machine 表的各 field。
- MaterialTest: 测试了 Material 的 struct 是否正确。
- ProduceTest: 测试了 Produce 表的各 field。
- TeamTest: 测试了 Team 表的各 field。
- TradeTest: 测试了 Trade 表的各 field。

5 计划执行的测试

5.1 被测试项

Tomato 的整个后端。

6 测试进度表

如下图所示。

▼	server (com.java.asdan.tomato)	18s 346ms
▶	ServerApplicationTest	1ms
▶	AdminLoginControllerTest	4s 76ms
▶	AuctionControllerTest	1s 264ms
▶	ChangePasswordControllerTest	122ms
▶	CompetitionCreatorControllerTest	1s 12ms
▶	CompetitionInformationControllerTest	1s 621ms
▶	CompetitionStatusAdminControllerTest	1s 691ms
▶	DeleteCompetitionControllerTest	134ms
▶	GetAllCompetitionControllerTest	135ms
▶	UpdateCompetitionStatusControllerTest	353ms
▶	AvatarControllerTest	169ms
▶	BuyMaterialControllerTest	1s 668ms
▶	ClientInfoControllerTest	151ms
▶	ClientLoginControllerTest	177ms
▶	ClientPropertyControllerTest	770ms
▶	CompetitionStatusControllerTest	747ms
▶	GetAllUsersControllerTest	56ms
▶	GetProduceHistoryControllerTest	236ms
▶	GetTradeHistoryControllerTest	194ms
▶	ListenPropertyControllerTest	655ms
▶	SellMaterialControllerTest	785ms
▶	SendToSellerControllerTest	1s 310ms
▶	UndoTradeControllerTest	499ms
▶	WebSocketPushHandlerTest	0ms
▶	AdminTest	0ms
▶	CompetitionTest	0ms
▶	MachineTest	0ms
▶	MaterialTest	0ms
▶	ProduceTest	0ms
▶	TeamTest	3ms
▶	TradeTest	2ms
▶	CompetitionServiceTest	105ms
▶	LoginServiceTest	58ms
▶	CompetitionServiceTest	36ms
▶	LoginServiceTest	171ms
▶	OrderServiceTest	145ms
▶	CustomErrorTypeTest	0ms

7 需求的可追踪性

本测试计划覆盖了所有后端（除微信 WebSocket 外）的需求（接口）。

8 评价

8.1 评价准则

Coverage。

8.2 数据处理

数据处理由 IntelliJ 自动完成，IntelliJ 将为我们生成完整的 Coverage 报告。

8.3 结论

本测试计划十分合理。