天津大学

《软件工程综合实践》结课报告

项目测试文档

学　　院　智能与计算学部

专　　业　软件工程

年　　级　2022 级

姓　　名　邵维 王俊哲 聂哲浩 李亮克

学　　号　30222444(55 31 26 34)

2024 年 9 月 12 日

目　　录

第一章　测试部分

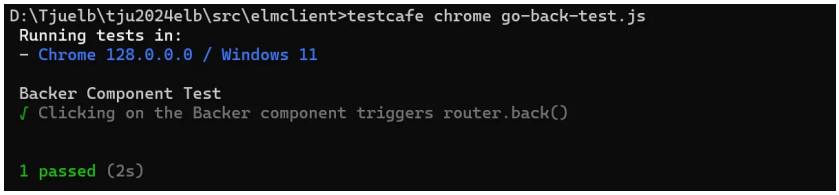
1.1　前端测试

1.1.1　集中测试

采用 testCafe 工具对页面和组件进行测试，该组件可以通过 npm 指令安装，同时支持 chrome 浏览器和 firefox 浏览器，他可以在这两个浏览器上对代码进行测试。测试前需要保证安装有 chrome 或者 firefox 浏览器。TestCafe 安装方法：

- npm install -g testcafe
- 安装时可以在 elmclient 目录下执行这个命令
- 然后就可以在 elmclient 目录下编写测试代码并通过执行
- testcafe 浏览器名测试代码文件名

运行测试代码。例如：testcafe chrome alert-test.js, 使用 chrome 浏览器运行 alert-test.js 文件, 首先测试 backer，测试成功：



1.2　后端测试

添加 h2 内嵌数据库和 mybatis-spring-boot-starter-test 依赖，来进行后端单元测试

1. 设置测试环境：
    - @SpringBootTest 注解：告诉 Spring Boot 测试框架这是一个 Spring Boot 应用的测试类，它会启动应用上下文。
    - @Autowired 注解：自动注入测试实例
    - @MockBean 注解：创建测试实例调用的模拟对象。在测试中不调用实际的实现，而是使用模拟对象来控制行为和返回值。
2. 测试数据准备
    - 以 BusinessService 的 ListBusinessByOrderTypeId 为例，创建三个 ‘Business‘ 对象，分别设置它们的 ‘orderTypeId‘ 和 ‘businessName‘ 属性。
    - 根据 ‘orderTypeId‘ 将 ‘Business‘ 对象分组到两个列表中，模拟数据库查询的结果。

3. 设置预期行为
    - 使用 Mockito 的 ‘when(...).thenReturn(...)‘ 方法设置模拟对象的预期行为。
    - 当 ‘listBusinessByOrderTypeId‘ 方法被调用时，根据传入的 ‘orderTypeId‘ 返回相应的商家列表。
4. 执行操作
    - 调用 ‘businessService.listBusinessByOrderTypeId‘ 方法两次，分别传入不同的 ‘orderTypeId‘，获取结果列表。
5. 验证结果
    - 使用 ‘assertNotNull‘ 验证返回的列表不为 ‘null‘。
    - 使用 ‘assertEquals‘ 验证返回的列表与预期的列表相等。
    - 使用 ‘assertNotEquals‘ 验证两个不同 ‘orderTypeId‘ 返回的列表不相等。
6. 验证交互
    - 使用 Mockito 的 ‘verify‘ 方法验证 ‘businessServiceImpl‘ 是否被正确调用了 ‘listBusinessByOrderTypeId‘ 方法，并且传入了正确的参数。
    - 这个测试类确保了 ‘BusinessService‘ 的 ‘listBusinessByOrderTypeId‘ 方法能够根据订单类型 ID 正确地返回商家列表，并且能够处理不同的输入。
    - 对于 controller 层，由于添加了 token 验证，拦截器会拦截所有不在排除路径中的，token 不符的 url，使用 @MockBean 注解 TokenInterceptor 以绕过拦截器，便于 controller 层测试