

测试文档

修改者	修改日期	修改原因	版本
苏致成	2022.07.03	增加测试背景、参考文献	v0.0.1
郁博文	2022.07.06	添加了 finance 模块相关测试	v0.0.2
马宇森	2022.07.06	添加了 business 模块，finance 模块相关测试	v0.0.3
石睿琪	2022.07.06	添加了 sale 模块的相关测试	v0.0.4

1. 测试背景

编写此报告的目的在于清楚地阐述 ERP 项目中与测试相关的所有内容。包括背景、人员、平台、内容、结果。

1.1 目的

为了更好地辅助开发、保证产品的正确性安全性、提高用户信赖程度而进行明确的、有目的的、有效率的测试。

1.2 参考文献

[1]《软件工程与计算 II——软件开发的技术基础》

2. 测试平台

后端本地测试平台为：MacOS、Windows、JDK1.8、JUnit、MySQL 数据库（单元测试、集成测试）。

3. 测试概述

项目采取自底向上集成的方式，主要进行的是驱动测试的开发，因而底部组件的开发可以同时进行。采取的技术有路径测试、随机测试、功能测试等。

4. 后端测试

4.1 DAO 层测试内容

DAO（Data Access Object）层主要进行的是单元测试。

- 测试的主要目的：检验 DAO 层代码能够完成数据存取的功能。

PaySheetDaoTest

测试名称	输入内容	预期结果	测试结果
saveSheetTest	用户信息，付款单据信息	返回的影响行数为一行	通过
saveBatchSheetContentTest	条目清单	返回的影响行数超过一行	通过
findAllSheetTest	无	返回全部的单据	通过
findSheetByIdTest	单据 ID	返回的单据结果不为空	通过
findContentBySheetIdTest	单据 ID	返回的条目清单结果不为空	通过
findAllSheetByStateTest	单据状态	返回的单据结果不为空	通过
updateSheetStateTest	单据 ID，单据状态	更新单据状态成功	通过
updateSheetStateOnPrevTest	单据 ID，单据目前状态、单据更新状态	更新单据状态成功	通过

测试结果截图

✓ 测试结果	788毫秒
✓ PaySheetDaoTest	788毫秒
✓ findSheetByIdTest()	710毫秒
✓ findAllSheetTest()	9毫秒
✓ saveBatchSheetContentTest()	27毫秒
✓ saveSheetTest()	11毫秒
✓ findContentBySheetIdTest()	7毫秒
✓ findAllSheetByStateTest()	7毫秒
✓ updateSheetStateOnPrevTest()	9毫秒
✓ updateSheetStateTest()	8毫秒

ReceiveSheetDaoTest

测试名称	输入内容	预期结果	测试结果
saveSheetTest	用户信息, 收款单据信息	返回的影响行数为一行	通过
saveBatchSheetContentTest	转账列表	返回的影响行数超过一行	通过
findAllSheetTest	无	返回全部的单据	通过
findSheetByIdTest	单据 ID	返回的单据结果不为空	通过
findContentBySheetIdTest	单据 ID	返回的条目清单结果不为空	通过
findAllSheetByStateTest	单据状态	返回的单据结果不为空	通过
updateSheetStateTest	单据 ID, 单据状态	更新单据状态成功	通过
updateSheetStateOnPrevTest	单据 ID, 单据目前状态、单据更新状态	更新单据状态成功	通过

测试结果截图

✓ 测试结果	746毫秒
✓ ReceiveSheetDaoTest	746毫秒
✓ findSheetByIdTest()	669毫秒
✓ findAllSheetTest()	9毫秒
✓ saveBatchSheetContentTest()	28毫秒
✓ saveSheetTest()	9毫秒
✓ findContentBySheetIdTest()	7毫秒
✓ findAllSheetByStateTest()	8毫秒
✓ updateSheetStateOnPrevTest()	8毫秒
✓ updateSheetStateTest()	8毫秒

CompanyAccountDaoTest

测试名称	输入内容	预期结果	测试结果
addCompanyAccountTest	新增的账户信息	返回的影响行数不为0	通过
updateCompanyAccountTest	编辑后的账户信息	返回的影响行数不为0	通过
deleteCompanyAccountTest	账户 id	返回的影响行数不为0	通过
queryAllAccountsTest	无	返回所有账户	通过
findOneByNameTest	账户名称	返回对应账户	通过
findOneByIdTest	账户 id	返回对应账户	通过

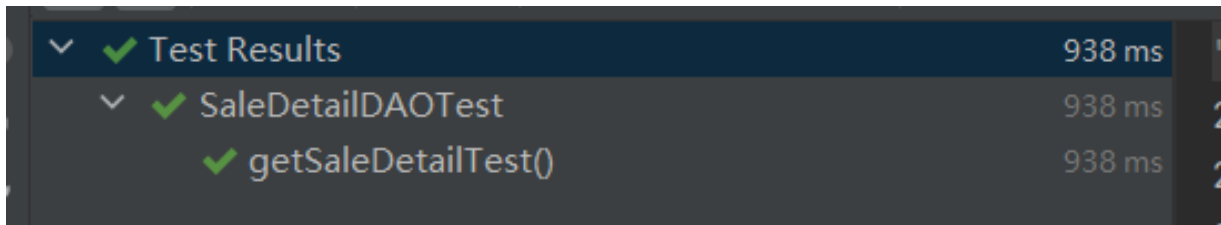
测试结果截图

✓ Test Results	1 sec 16 ms
✓ CompanyAccountDAOTest	1 sec 16 ms
✓ deleteCompanyAccountTest()	908 ms
✓ findOneByIdTest()	46 ms
✓ findOneByNameTest()	
✓ addCompanyAccountTest()	15 ms
✓ updateCompanyAccountTest()	32 ms
✓ queryAllAccountsTest()	15 ms

SaleDetailDAOTest

测试名称	输入内容	预期结果	测试结果
getSaleDetail	无	返回 List 不为空	通过

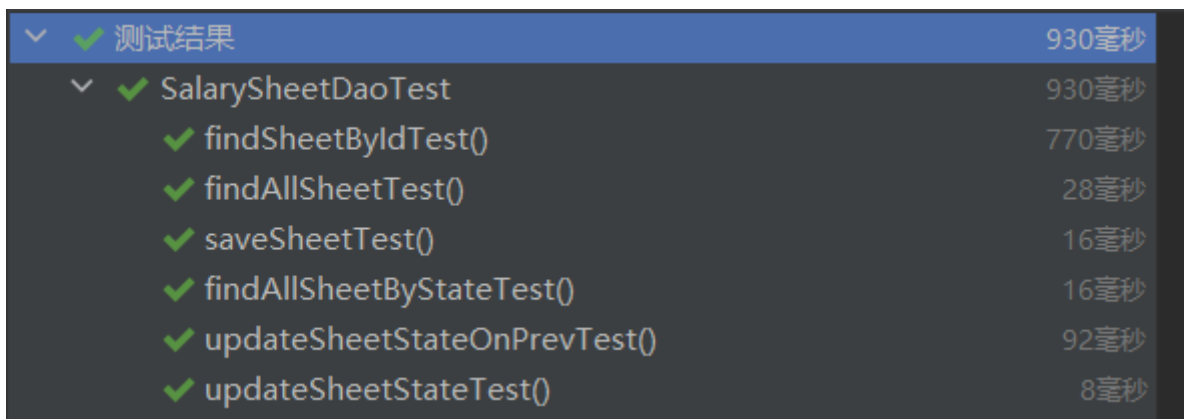
测试结果截图



SalarySheetDaoTest

测试名称	输入内容	预期结果	测试结果
saveSheetTest	用户信息, 收款单据信息	返回的影响行数为一行	通过
findAllSheetTest	无	返回全部的单据	通过
findSheetByIdTest	单据 ID	返回的单据结果不为空	通过
findAllSheetByStateTest	单据状态	返回的单据结果不为空	通过
updateSheetStateTest	单据 ID, 单据状态	更新单据状态成功	通过
updateSheetStateOnPrevTest	单据 ID, 单据目前状态、单据更新状态	更新单据状态成功	通过

测试结果截图



StaffDaoTest

测试名称	输入内容	预期结果	测试结果
createStaffTest	StaffPO 对象	返回的影响行数为 1	通过
updateStaff_not_foundTest	StaffPO 对象	返回的影响行数为 0	通过
updateStaffTest	StaffPO 对象	返回的影响行数为 1	通过
findByStaffIdTest	员工 ID	返回的对象非空	通过
queryAllTest	无	返回的对象非空且大小为期望值	通过
findBasicSalaryTest	员工 ID	返回值为期望值	通过
findCalSalaryMethodTest	员工 ID	返回值为期望值	通过
checkIfAnEmptyPostTest	岗位 ID	返回值为 0	通过
checkIfAnEmptyPost_not_emptyTest	岗位 ID	返回值为期望值	通过
deleteStaffTest	员工 ID	返回值为空	通过

测试结果截图

Test Results	1 sec 326 ms
StaffDaoTest	1 sec 326 ms
updateStaff_not_foundTest()	1 sec 143 ms
deleteStaffTest()	60 ms
checkIfAnEmptyPostTest()	17 ms
updateStaffTest()	19 ms
findByStaffIdTest()	14 ms
checkIfAnEmptyPost_not_emptyTest()	11 ms
findCalSalaryMethodTest()	15 ms
findBasicSalaryTest()	14 ms
createStaffTest()	17 ms
queryAllTest()	16 ms

PostDaoTest

测试名称	输入内容	预期结果	测试结果
queryAllPostTest	无	返回的 PO 列表的大小为期望值	通过
findByPostIdTest	岗位 ID	返回值非空	通过
findIfHasPostTest	岗位 ID	返回值为 1	通过
findIfHasPost_noTest	岗位 ID	返回值为 0	通过
deletePostTest	岗位 ID	再次查询则返回值为 0	通过
updatePost_not_foundTest	PostPO 对象	返回的影响行数为 0	通过
updatePostTest	PostPO 对象	返回的影响行数为 0	通过
addPostTest	PostPO 对象	再次查询则返回值为期望值	通过

测试结果截图

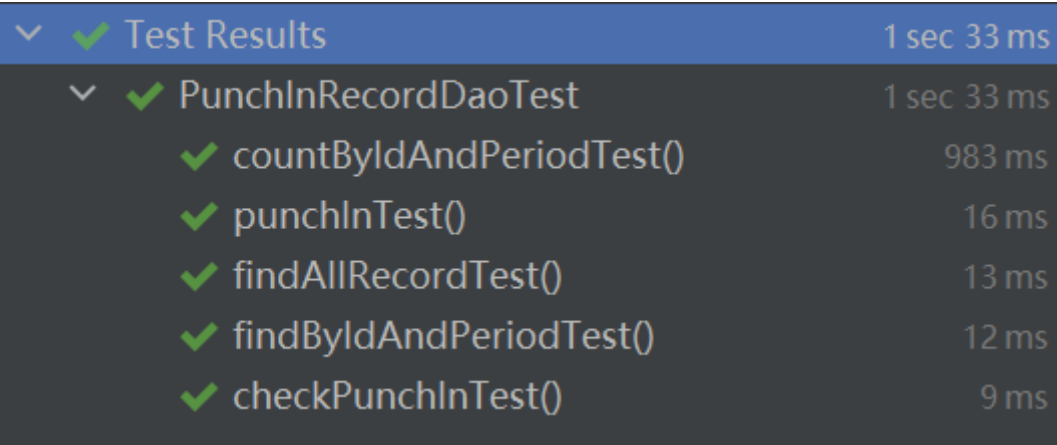
✓ Test Results	1 sec 55 ms
✓ PostDaoTest	1 sec 55 ms
✓ findIfHasPost_noTest()	909 ms
✓ deletePostTest()	22 ms
✓ findIfHasPostTest()	10 ms
✓ addPostTest()	26 ms
✓ findByPostIdTest()	11 ms
✓ queryAllPostTest()	18 ms
✓ updatePost_not_foundTest()	37 ms
✓ updatePostTest()	22 ms

PunchInRecordDaoTest

测试名称	输入内容	预期结果	测试结果
punchInTest	员工 ID, 员工姓名, 时间	返回的影响行数为 1	通过
checkPunchInTest	员工 ID, 时间	返回的影响行数为 0	通过
findByIdAndPeriodTe	员工 ID, 时间 1, 时	返回的值的的大小为期望值	通过

st	间 2	望值	
countByldAndPeriod Test	员工 ID, 时间 1, 时 间 2	返回的值为期望值	通过
findAllRecordTest	时间 1, 时间 2	返回的值为期望值	通过

测试结果截图



Test Results	1 sec 33 ms
PunchInRecordDaoTest	1 sec 33 ms
countByldAndPeriodTest()	983 ms
punchInTest()	16 ms
findAllRecordTest()	13 ms
findByldAndPeriodTest()	12 ms
checkPunchInTest()	9 ms

PromotionStrategyOnAmountDaoTest

测试名称	输入内容	预期结果	测试结果
setPromotionStrategyOnAmountTest	总价促销策略的信息	返回影响行数为一行	通过
getVoucherByAmountTest	总价金额	返回根据总价获得的代金券金额	通过
getDiscountByAmountTest	总价金额	返回根据总价获得的折扣	通过
getAllPromotionStrategyOnAmountTest	无	返回全部的总价促销策略	通过
updateByldTest	总价促销策略的修改信息	返回影响行数为一行	通过
findByldTest	总价促销策略编号	返回查询结果不为空	通过
getStrategyOnOneAmountTest	总价金额	返回查询结果不为空	通过

测试结果截图


```
✓ PromotionStrategyOnAmountDaoT 803 ms
  ✓ getDiscountByAmountTest() 686 ms
  ✓ getVoucherByAmountTest() 9 ms
  ✓ deleteByldTest() 7 ms
  ✓ getAllPromotionStrategyOnAmoi 14 ms
  ✓ getStrategyOnOneAmountTest() 11 ms
  ✓ findByldTest() 11 ms
  ✓ setPromotionStrategyOnAmount 21 ms
  ✓ updateByldTest() 44 ms
```

PromotionStrategyOnLevelDaoTest

测试名称	输入内容	预期结果	测试结果
setPromotionStrategyOnLevelTest	等级促销策略的信息	返回影响行数为一行	通过
getVoucherByLevelTest	等级	返回根据等级获得的代金券金额	通过
getDiscountByLevelTest	等级	返回根据等级获得的折扣	通过
getAllPromotionStrategyOnLevelTest	无	返回全部的等级促销策略	通过
updateByldTest	等级促销策略的修改信息	返回影响行数为一行	通过
findByldTest	等级促销策略编号	返回查询结果不为空	通过
getStrategyOnOneLevelTest	等级	返回查询结果不为空	通过

测试结果截图

```
✓ PromotionStrategyOnLevelDaoTest (com.l 853 ms)
  ✓ getVoucherByLevelTest() 752 ms
  ✓ getStrategyOnOneLevelTest() 14 ms
  ✓ deleteByldTest() 9 ms
  ✓ setPromotionStrategyOnLevelTest() 14 ms
  ✓ getDiscountByLevelTest() 9 ms
  ✓ findByldTest() 10 ms
  ✓ getAllPromotionStrategyOnLevelTest() 11 ms
  ✓ updateByldTest() 34 ms
```

4.2 Service 层测试内容

PayServiceTest

测试名称	输入内容	预期结果	测试结果
makePaySheetTest	用户信息, 付款单据信息	生成的单据信息正确	通过
getPaySheetByStateTest	单据状态	返回的单据结果不为空	通过
approval_exception_1Test	单据 ID, 单据状态	非法的审批操作, 抛出运行时错误	通过
approval_exception_2Test	单据 ID, 单据状态	非法的审批操作, 抛出运行时错误	通过
approval_failedTest	单据 ID, 单据状态	审批后的单据状态正确	通过
approval_successTest	单据 ID, 单据状态	审批后的单据状态正确	通过

测试结果截图

✓ 测试结果	832毫秒
✓ PayServiceTest	832毫秒
✓ approval_exception_1Test()	723毫秒
✓ approval_exception_2Test()	10毫秒
✓ approval_successTest()	46毫秒
✓ makePaySheetTest()	31毫秒
✓ approval_failedTest()	13毫秒
✓ getPaySheetByStateTest()	9毫秒

ReceiveServiceTest

测试名称	输入内容	预期结果	测试结果
makeReceiveSheetTest	用户信息, 收款单据信息	生成的单据信息正确	通过
getReceiveSheetByStateTest	单据状态	返回的单据结果不为空	通过
approval_exception_1Test	单据 ID, 单据状态	非法的审批操作, 抛出运行时错误	通过
approval_exception_2Test	单据 ID, 单据状态	非法的审批操作, 抛出运行时错误	通过

approval_failedTest	单据 ID, 单据状态	审批后的单据状态正确	通过
approval_successTest	单据 ID, 单据状态	审批后的单据状态正确	通过

测试结果截图

✓ ✓ 测试结果	779毫秒
✓ ReceiveServiceTest	779毫秒
✓ approval_exception_1Test()	675毫秒
✓ approval_exception_2Test()	7毫秒
✓ approval_successTest()	42毫秒
✓ makeReceiveSheetTest()	33毫秒
✓ approval_failTest()	13毫秒
✓ getReceiveSheetByStateTest()	9毫秒

CompanyAccountServiceTest

测试名称	输入内容	预期结果	测试结果
addCompanyAccountTest	新增的账户信息	返回的影响行数不为0	通过
updateCompanyAccountTest	编辑后的账户信息	返回的影响行数不为0	通过
deleteCompanyAccountTest	账户 id	返回的影响行数不为0	通过
queryAllAccountsTest	无	返回所有账户	通过
findOneByNameTest	账户名称	返回对应账户	通过

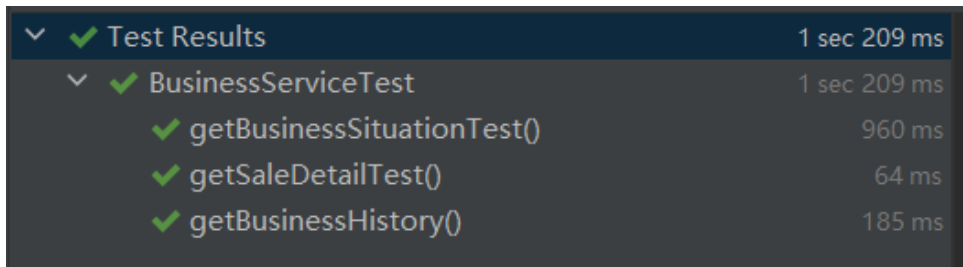
测试结果截图

✓ ✓ Test Results	970 ms
✓ CompanyAccountServiceTest	970 ms
✓ deleteCompanyAccountTest()	874 ms
✓ findOneByNameTest()	48 ms
✓ addCompanyAccountTest()	8 ms
✓ updateCompanyAccountTest()	24 ms
✓ queryAllAccountsTest()	16 ms

BusinessServiceTest

测试名称	输入内容	预期结果	测试结果
getSaleDetailTest	无	返回的 List 不为空	通过
getBusinessHistoryTest	无	返回的 VO 不为空且 VO 里的每个 List 都不为空	通过
getBusinessSituationTest	起始日期、终止日期	返回的经营情况数值正确	通过

测试结果截图



The screenshot shows a test results interface with a dark background. It displays a tree view of test results. The root node is 'Test Results' with a green checkmark and a duration of '1 sec 209 ms'. Under it is 'BusinessServiceTest' also with a green checkmark and '1 sec 209 ms'. Below 'BusinessServiceTest' are three sub-items, each with a green checkmark: 'getBusinessSituationTest()' (960 ms), 'getSaleDetailTest()' (64 ms), and 'getBusinessHistory()' (185 ms).

Test Results	1 sec 209 ms
BusinessServiceTest	1 sec 209 ms
getBusinessSituationTest()	960 ms
getSaleDetailTest()	64 ms
getBusinessHistory()	185 ms

StaffServiceTest

测试名称	输入内容	预期结果	测试结果
register_passTest	StaffVO 对象	增加员工成功, 状态正确	通过
register_phoneTest	StaffVO 对象	增加员工失败, 状态为手机格式不正确	通过
register_genderTest	StaffVO 对象	增加员工失败, 状态为性别格式不正确	通过
register_postTest	StaffVO 对象	增加员工失败, 状态为岗位信息不正确	通过
update_passTest	StaffVO 对象	更改员工成功, 状态正确	通过
update_phoneTest	StaffVO 对象	更改员工失败, 状态为手机格式不正确	通过
update_genderTest	StaffVO 对象	更改员工失败, 状态为性别格式不正确	通过
update_postTest	StaffVO 对象	更改员工失败, 状态为岗位信息不正确	通过
update_no_correspondingTest	StaffVO 对象	更改员工失败, 状态为无对应员工	通过
addPost_passTest	PostVO 对象	增加岗位成功, 状态	通过

		正确	
addPost_postNameTest	PostVO 对象	增加岗位失败，状态为岗位名称不正确	通过
addPost_salary_1Test	PostVO 对象	增加岗位失败，状态为岗位薪资不正确	通过
addPost_salary_2Test	PostVO 对象	增加岗位失败，状态为岗位薪资不正确	通过
addPost_calSalaryMethodTest	PostVO 对象	增加岗位失败，状态为薪资计算方法不正确	通过
addPost_levelTest	PostVO 对象	增加岗位失败，状态为等级不正确	通过
updatePost_passTest	PostVO 对象	更改岗位成功，状态正确	通过
updatePost_postNameTest	PostVO 对象	更改岗位失败，状态为岗位名称不正确	通过
updatePost_salary_1Test	PostVO 对象	更改岗位失败，状态为岗位薪资不正确	通过
updatePost_salary_2Test	PostVO 对象	更改岗位失败，状态为岗位薪资不正确	通过
updatePost_calSalaryMethodTest	PostVO 对象	更改岗位失败，状态为薪资计算方法不正确	通过
updatePost_levelTest	PostVO 对象	更改岗位失败，状态为等级不正确	通过
updatePost_no_correspondingTest	PostVO 对象	更改岗位失败，状态为无对应的岗位	通过
queryAllTest	无	返回对象非空且大小符合预期	通过

测试结果截图

✓ Test Results	1 sec 227 ms
✓ StaffServiceTest <ul style="list-style-type: none"> ✓ addPost_passTest() 950 ms ✓ updatePost_passTest() 59 ms ✓ update_phoneTest() 12 ms ✓ addPost_levelTest() 7 ms ✓ update_postTest() 13 ms ✓ updatePost_levelTest() 7 ms ✓ register_postTest() 10 ms ✓ updatePost_no_correspondingTest() 9 ms ✓ addPost_calSalaryMethodTest() 7 ms ✓ update_no_correspondingTest() 13 ms ✓ update_passTest() 24 ms ✓ queryAllTest() 23 ms ✓ addPost_salary_1Test() 8 ms ✓ addPost_salary_2Test() 10 ms ✓ updatePost_calSalaryMethodTest() 7 ms ✓ register_passTest() 18 ms ✓ register_phoneTest() 9 ms ✓ update_genderTest() 8 ms ✓ register_genderTest() 9 ms ✓ addPost_postNameTest() 7 ms ✓ updatePost_salary_1Test() 9 ms ✓ updatePost_salary_2Test() 8 ms 	1 sec 227 ms

SalaryServiceTest

测试名称	输入内容	预期结果	测试结果
getSalarySheetByStateTest	单据状态	返回的单据结果不为空	通过
approval_exception_1Test	单据 ID, 单据状态	非法的审批操作, 抛出运行时错误	通过
approval_exception_2Test	单据 ID, 单据状态	非法的审批操作, 抛出运行时错误	通过
approval_failedTest	单据 ID, 单据状态	审批后的单据状态正确	通过
approval_1Test	单据 ID, 单据状态	审批后的单据状态正确	通过
approval_2Test	单据 ID, 单据状态	审批后的单据状态正确	通过
generateSalaryTest	无	返回的列表大小符合预期	通过
generateAnnualTest	无	返回的列表大小符合预期	通过

strategy_MONTHLYTest	StaffPO	返回的数值符合预期	通过
strategy_PERCENTAGE_TEST	StaffPO	返回的数值符合预期	通过
strategy_YEARLYTest	StaffPO	返回的数值符合预期	通过
strategy_MONTHLY_ANNTTest	StaffPO	返回的数值符合预期	通过
strategy_PERCENTAGE_ANNTTest	StaffPO	返回的数值符合预期	通过
strategy_YEARLY_ANNTTest	StaffPO	返回的数值符合预期	通过

测试结果截图

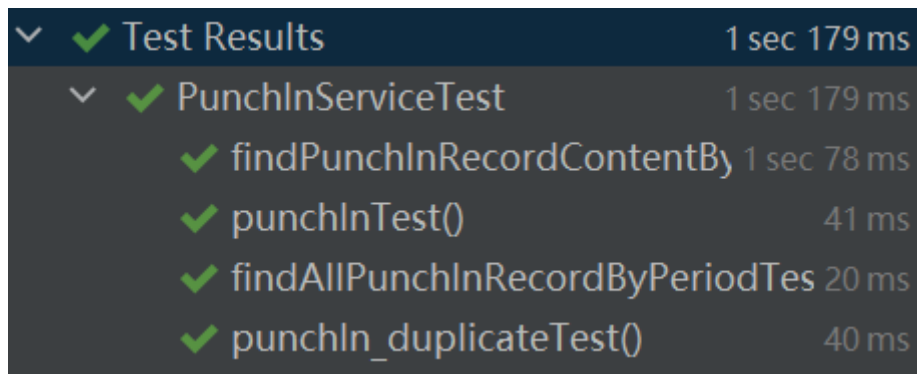
✓ 测试结果	1秒 40毫秒
✓ SalaryServiceTest	1秒 40毫秒
✓ generateSalaryTest()	798毫秒
✓ strategy_PERCENTAGE_ANNTTest()	15毫秒
✓ approval_exceptions_1Test()	10毫秒
✓ approval_exceptions_2Test()	8毫秒
✓ generateAnnualTest()	94毫秒
✓ approval_failedTest()	11毫秒
✓ strategy_YEARLY_ANNTTest()	9毫秒
✓ strategy_MONTHLY_ANNTTest()	9毫秒
✓ getSalarySheetByStateTest()	11毫秒
✓ strategy_PERCENTAGETest()	20毫秒
✓ approval_1Test()	9毫秒
✓ approval_2Test()	9毫秒
✓ makeSalarySheet_1Test()	12毫秒
✓ strategy_YEARLYTest()	8毫秒
✓ strategy_MONTHLYTest()	17毫秒

PunchInServiceTest

测试名称	输入内容	预期结果	测试结果
punchInTest	UserVO	打卡记录非空且记录大小为期望值	通过
punchIn_duplicateTest	UserVO	二次打卡失败，打卡记录非空且数值为期望值	通过

		望值	
findAllPunchInRecordByPeriodTest	时间字符串	打卡记录非空且大小为期望值	通过
findPunchInRecordContentByStaffAndPeriodTest	UserVO、时间字符串	打卡记录非空且大小为期望值	通过

测试结果截图



SaleServiceTest

测试名称	输入内容	预期结果	测试结果
queryAllPromotionStrategyOnLevel	无	返回等级促销策略的个数	通过
queryAllPromotionStrategyOnAmount	无	返回总价促销策略的个数	通过
update_onLevel	等级促销策略的修改内容	返回修改策略的编号	通过
update_onAmount	总价促销策略的修改内容	返回修改策略的编号	通过

测试结果截图

✓ SaleServiceTest (com.nju.edu.erp.service)	1 sec 121 ms
✓ getSaleSheetByStateTest()	774 ms
✓ approval_exceptions_1Test()	15 ms
✓ approval_exceptions_2Test()	12 ms
✓ approval_failedTest()	19 ms
✓ makeSaleSheetTest()	75 ms
✓ warehouseServiceTest()	15 ms
✓ update_onAmount()	37 ms
✓ queryAllPromotionStrategyOnLevel()	25 ms
✓ approval_1Test()	15 ms
✓ approval_2Test()	94 ms
✓ queryAllPromotionStrategyOnAmount()	12 ms
✓ findTransXByNameAndPeriodTest()	13 ms
✓ update_onLevel()	15 ms

4.3 Utils 包测试

TaxComputingTest

测试名称	输入内容	预期结果	测试结果
test_no_trapTest	数值	返回值符合预期	通过
test_over_no_trapTest	数值	返回值符合预期	通过
test_overTest	数值	返回值符合预期	通过
test_lowTest	数值	返回值符合预期	通过
test_bound_1Test	数值	返回值符合预期	通过
test_bound_2Test	数值	返回值符合预期	通过

测试结果截图

✓ Test Results	25 ms
✓ TaxComputingTest	25 ms
✓ test_bound_1Test()	23 ms
✓ test_bound_2Test()	
✓ test_lowTest()	1 ms
✓ test_overTest()	
✓ test_over_no_trapTest()	1 ms
✓ test_no_trapTest()	

4.4 后端测试结果与结论

所有代码均已根据测试结果进行调试、修改和优化。