

# 软件体系结构文档

## 文档修改历史

修改人员	日期	修改原因	版本号
苏致成	2022.06.29	添加进程图	v0.0.0
郁博文	2022.07.03	添加逻辑包图、逻辑视角图	v0.0.1
马宇森	2022.07.04	添加开发包图	v0.0.2
石睿琪	2022.07.04	添加模块视图	v0.0.3
郁博文	2022.07.04	Finance 模块相关图及说明	v0.0.4
苏致成	2022.07.05	添加部署图	v0.0.5
马宇森	2022.07.05	添加 business 相关模块说明	v0.0.6
石睿琪	2022.07.05	添加模块视图相关说明	v0.0.7
苏致成	2022.07.06	添加业务逻辑层 Staffbl 模块的接口规范	v0.0.8
苏致成	2022.07.06	添加业务逻辑层 Salarybl 模块的接口规范	v0.0.9
马宇森	2022.07.06	修改经营历程相关	v0.0.10

## 1. 引言

### 1.1 编制目的

本报告详细完成对 ERP 管理系统的概要设计，从而达到指导详细设计和开发的目的，同时实现和测试人员以及用户的沟通。

本报告面向开发人员、测试人员和用户编写，是了解系统的导航。

### 1.2 词汇表

词汇名称	词汇含义	备注
ERP	进销存管理系统	无

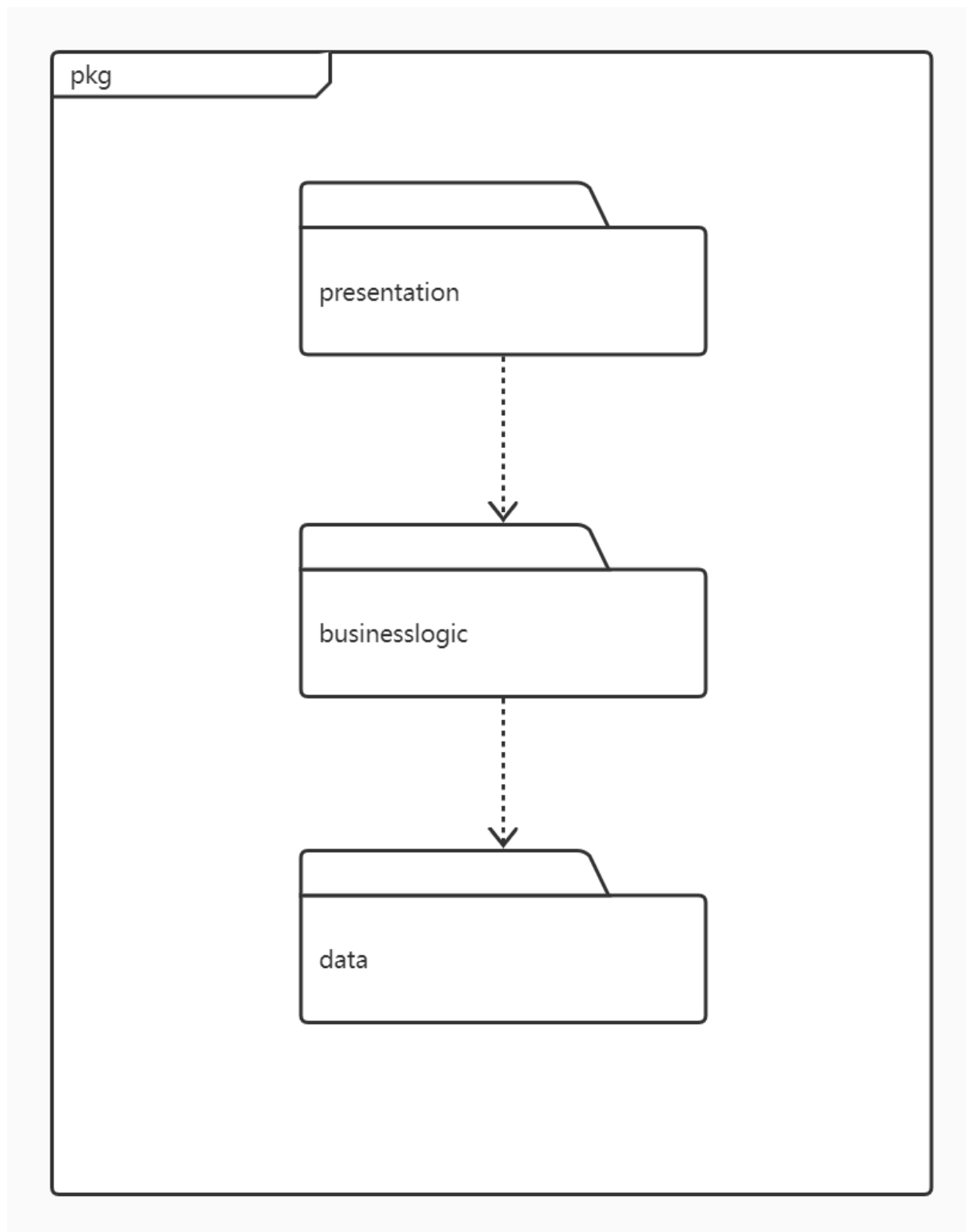
## 1.3 参考资料

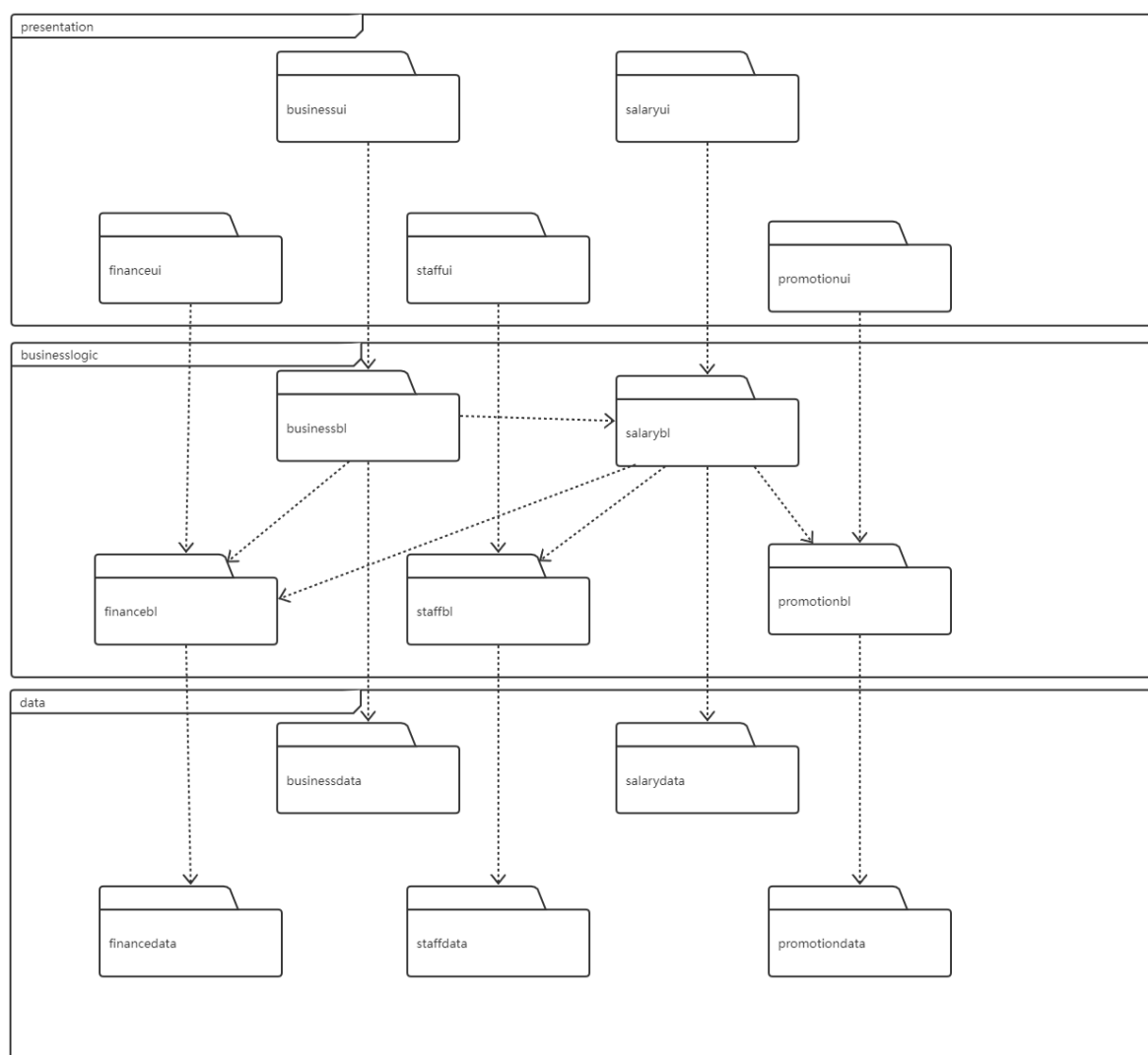
- 《软件工程与计算卷二》、《软件开发的技术基础》
- UML 2.0 语法
- IEEE 标准

## 2. 产品概述

参考 ERP 系统用例文档和 ERP 需求规格说明文档中对产品的概括描述。

### 3. 逻辑视图





## 4. 组合视图

### 4.1 开发包图

表示软件组件在开发时环境中的静态组织：

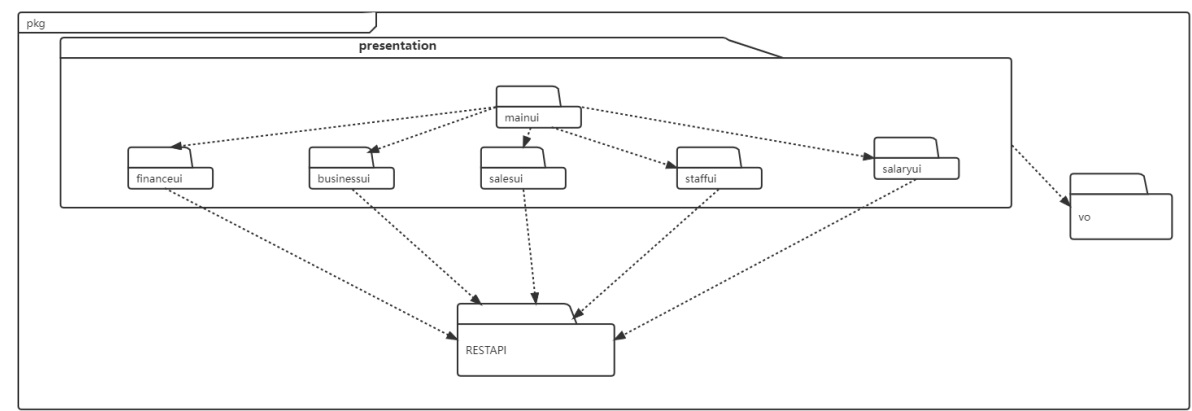
- 描述开发包以及相互间的依赖
- 绘制开发包图

开发（物理）包	依赖的其他开发包
FinanceUI	FinanceController, CustomerController, vo, vue 组件库

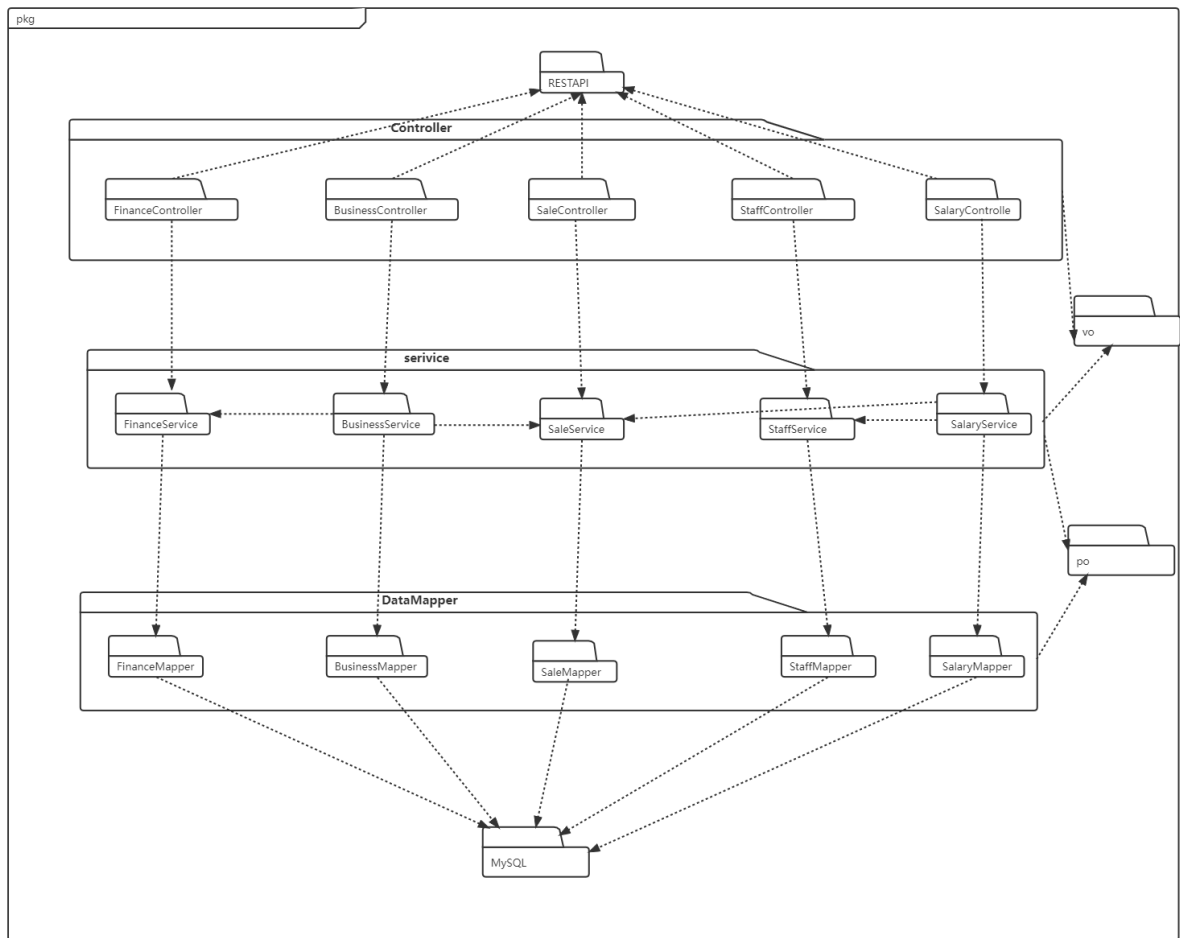
BusinessUI	BusinessController,SaleController,PurchaseController,FinanceController,SalaryController,vo,vue 组件库
StaffUI	StaffController,PostController,PunchInController,vo,vue 组件库
SalaryUI	SalaryController,vo,vue 组件库
SaleUI	SaleController,vo,vue 组件库
FinanceController	FinanceService,vo
BusinessController	BusinessService,vo
StaffController	StaffService,vo
PunchInController	PunchInService,vo
SalaryController	SalaryService,vo
SaleController	SaleService,vo
FinanceService	CompanyAccountMapper,PaySheetMapper,ReceiveSheetMapper,po,vo
BusinessService	SaleDetailMapper,PaySheetMapper,ReceiveSheetMapper,po,vo
StaffService	StaffMapper,PostMapper,po,vo
PunchInService	PunchInMapper,po,vo
SalaryService	SalarySheetMapper,po,vo
SaleService	PromotionStrategyOnLevel,po,vo PromotionStrategyOnAmount,po,vo
SaleDetailMapper	Datausability,po
PaySheetMapper	Datausability,po
ReceiveSheetMapper	Datausability,po
CompanyAccountMapper	Datausability,po
StaffMapper	Datausability,po

PostMapper	Datausability,po
PunchInMapper	Datausability,po
SalarySheetMapper	Datausability,po
VO	
PO	
vue 组件库	

ERP 系统客户端开发包图如下图所示

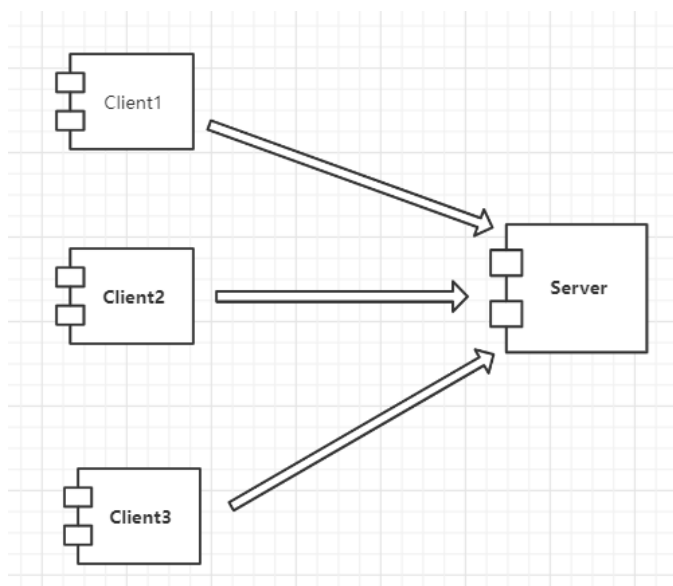


ERP 系统服务器端开发包图如下图所示



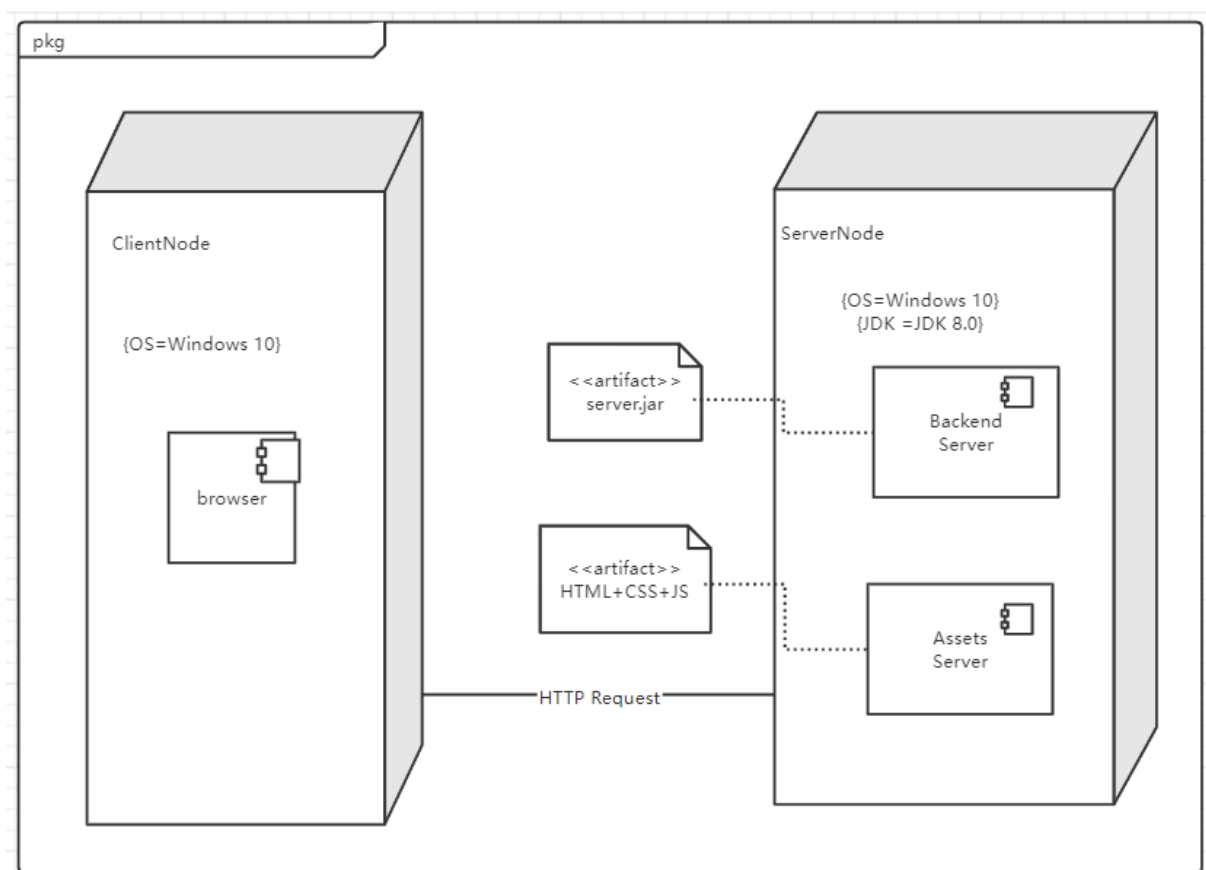
## 4.2 运行时进程

- 示意图



## 4.3 物理部署

- 处理如何将软件组件映射到硬件基础设施
- 部署图



## 5. 架构设计

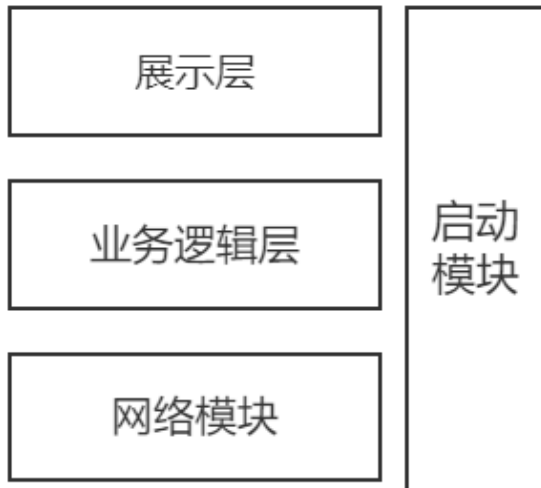
- 描述功能分解和如何在不同的层中安排软件模块
  - 描述架构中的对象，包含架构图
  - 描述组件接口信息
    - 包括：语法、前置消息、后置消息

### 5.1 模块职责

客户端模块和服务端模块视图分别如下两张图所示（依次对应上下两张）。

- 客户端模块视图





● 服务器端模块视图



● 下表为客户端各层的职责

层	职责
启动模块	负责初始化网络通信机制，启动用户界面
展示层	基于窗口化界面的灯具企业经营管理系统的界面
业务逻辑层	对用户界面的行为做出响应、进行业务逻辑处理并与服务端交互
网络模块	发起 HTTP 请求

● 下表为服务端各层职责

层	职责
启动模块	负责初始化网络通信机制，启动用户界面
网络模块	接受 HTTP 请求
数据层	负责数据的持久化和访问接口

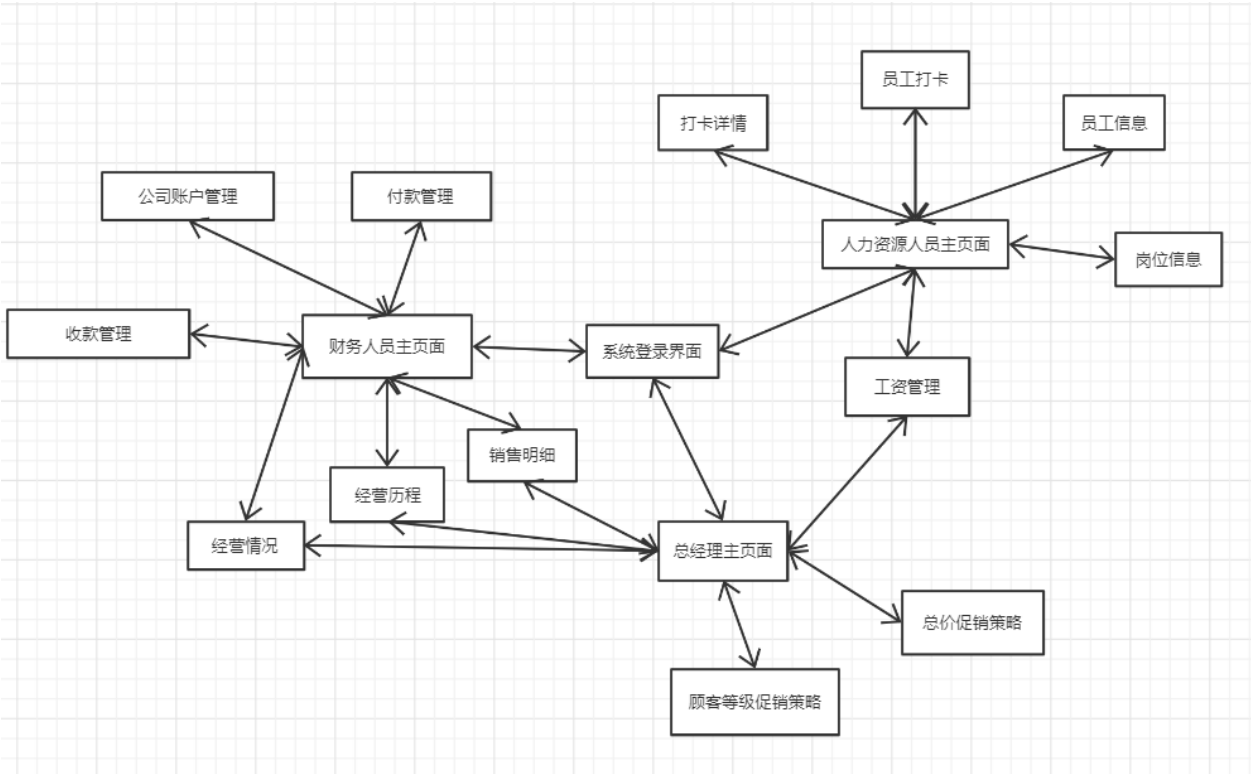
每一层只使用下方直接接触的层。层与层之间仅仅通过接口的调用来完成。层之间的接口如下表所示：

接口	服务调用方	服务提供方
BusinessController CompanyAccountController PayController RecieveController CategoryController CustomerController ProductController PunchInController PurchaseController PurchaseReturnsController SalaryController SaleController SaleReturnsController StaffController TestEnumController UserController WarehouseController	客户端展示层	客户端业务逻辑层
SalaryService PayStrategyService PunchInService StaffService CategoryService CustomerService ProductService PromotionStrategyService PurchaseReturnService PurchaseService SaleReturnsService SaleService SheetService UserService WarehouseService	客户端业务逻辑层	服务端数据层

## 5.2 用户界面层分解

注意，以下描述均只涉及 Lab7。

根据需求，系统存在 14 个用户界面：公司账户管理界面、收款管理界面、付款管理界面、销售明细界面、经营历程界面、经营情况界面、员工打卡界面、打卡详情界面、员工信息界面、岗位信息界面、工资管理界面、顾客等级促销策略界面、总价促销策略界面。界面跳转如下图所示。



### 5.2.1 职责

模块	职责
公司账户管理	负责管理公司的银行账户
收款管理	负责管理公司的收款情况
付款管理	负责管理公司的付款情况
销售明细表	负责展示商品的销售情况
经营历程表	负责展示经营过程中的单据
经营情况表	负责展示公司的经营情况
促销策略管理	负责管理经营的促销策略
员工管理	负责管理员工的详细信息

工资管理	负责管理工资单的制定情况
促销策略管理	负责管理促销策略的详细信息

## 5.2.2 接口规范

### ● 账户管理模块

接口名称	请求方法	路径	内容
companyAccountCreate	POST	/api/company-account/create	增加账户
companyAccountUpdate	POST	/api/company-account/queryAll	查询所有账户
companyAccountDelete	GET	/api/company-account/update	编辑账户
companyAccountQueryAll	GET	/api/company-account/delete	删除账户

### ● 员工管理模块

接口名称	请求方法	路径	内容
getPostInfo	GET	/api/staff/query-all-post	获取岗位
PostDelete	GET	/api/staff/post-delete	删除岗位
PostUpdate	POST	/api/staff/post-update	编辑岗位
PostCreate	POST	/api/staff/post-create	创建岗位
getStaffInfo	GET	/api/staff/query-all-staff	获取员工
StaffDelete	GET	/api/staff/staff-delete	删除员工
StaffUpdate	POST	/api/staff/staff-update	编辑员工
StaffCreate	POST	/api/staff/staff-create	创建员工
punchin	GET	/api/punch-in/punch-in	员工打卡
getPunchInInfo	GET	/api/punch-in/query-all	查看所有员工打卡
getPunchContent	GET	/api/punch-in/query-one	查看某一员工打卡

### ● 工资管理模块

接口名称	请求方法	路径	内容
generateAllSalary	GET	/api/salary/sheet-make	生成工资单
getAllSalary	GET	/api/salary/sheet-show	获取工资单
firstApproval	GET	/api/salary/first-approval	一级审批

secondApproval	GET	/api/salary/second-approval	二级审批
generateAllAnnual	GET	/api/salary/annual	制定年终奖

## ● 付款管理模块

接口名称	请求方法	路径	内容
paySheetCreate	POST	/api/pay/sheet-MAKE	添加付款单
paySheetApproval	GET	/api/pay/approval	审批付款单
paySheetQueryAll	GET	/api/pay/sheet-show	查看付款单

## ● 收款管理模块

接口名称	请求方法	路径	内容
receiveSheetCreate	POST	/api/receive/sheet-make	添加收款单
receiveSheetApproval	GET	/api/receive/approval	审批收款单
receiveSheetQueryAll	GET	/api/receive/sheet-show	查看收款单

## ● 经营情况表模块

接口名称	请求方法	路径	内容
getBusinessSituation	GET	/api/business/business-situation	获取经营情况表

## ● 经营历程表模块

接口名称	请求方法	路径	内容
getBusinessHistory	GET	/api/business/business-history	获取工资单

## ● 销售明细模块

接口名称	请求方法	路径	内容
getSaleDetail	GET	/api/businesssale-detail	获取销售明细
exportAsExcel	GET	/api/business/sale-detail/excel	导出为表格

## ● 促销策略模块

接口名称	请求方法	路径	内容
promotionStartegyOnLevelCreate	POST	/api/sale/PromotionStrategy/OnLevel/create	添加用户等级的促销策略
promotionStartegyOnLevelGetAll	GET	/api/sale/PromotionStrategy/OnLevel/getall	获取用户等级的促销策略
promotionStrategyOnLevelUpdate	POST	/api/sale/PromotionStrategy/OnLevel/update	编辑用户等级的促销策略
promotionStrategyOnLevelDelete	GET	/api/sale/PromotionStrategy/OnLevel/delete	删除用户等级的促销策略
promotionstrategyOnAmountCreate	POST	/api/sale/PromotionStrategy/OnAmount/create	添加关于总价的促销策略
promotionstrategyOnAmountGetAll	GET	/api/sale/PromotionStrategy/OnAmount/getall	获取关于总价的促销策略
promotionstrategyOnAmountUpdate	POST	/api/sale/PromotionStrategy/OnAmount/update	编辑关于总价的促销策略
promotionstrategyOnAmountDelete	GET	/api/sale/PromotionStrategy/OnAmount/delete	删除关于总价的促销策略

### 5.2.3 用户界面模块设计原理

用户界面利用Vue.js 框架实现。

## 5.3 业务逻辑层分解

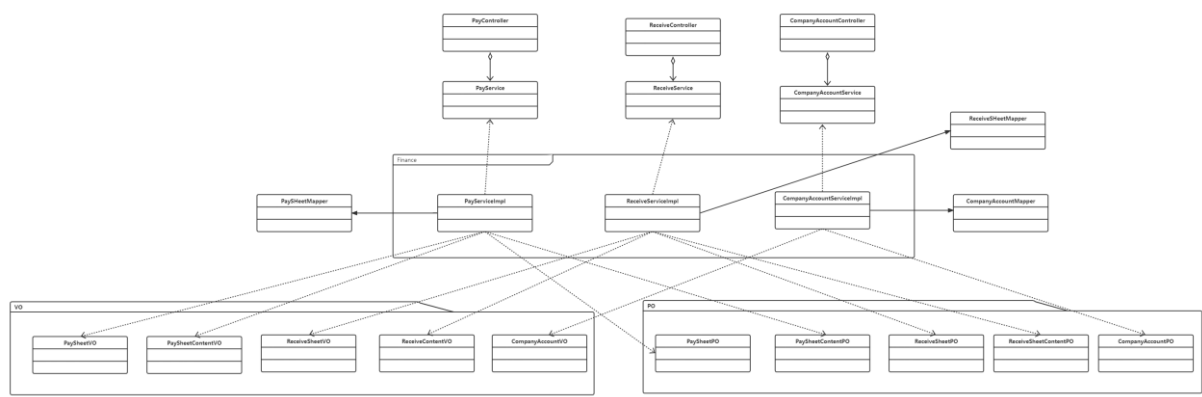
### 5.3.1 职责

模块	职责
Staffbl	负责员工信息管理的业务 负责岗位信息管理的业务 负责打卡的业务
Salarybl	负责制定工资的业务

	负责制定年终奖的业务
Financebl	负责制定付款单的业务 负责制定收款单的业务
Businessbl	负责查看经营情况表的业务 负责查看销售明细表的业务 负责查看经营历程表的业务
Promotionbl	负责制定促销策略的业务 负责查看促销策略的业务

5.3.2 接口规范

Finacebl 模块的关键类图



Finacebl 模块的接口规范

接口名字	语法	前置条件	后置条件
PayController.makePayOrder	makePayOrder(UserVO userVO, PaySheetVO paySheetVO)	无	将付款单信息传递给 PayService
PayController.showSheetByState	showSheetByState(PaySheetState state)	需查找的单据已被建立	从 PayService 获取对应状态的单据

PayController.approval	approval(String paySheetId, PaySheetState state)	需审批的单据已被建立	将审批信息传递给 PayService
PayController.findBySheetId	findBySheetId(String id)	需查找的单据已被建立	从 PayService 获取对应 ID 的单据
ReceiveController.makeReceiveOrder	makeReceiveOrder(UserVO userVO, ReceiveSheetVO receiveSheetVO)	无	将收款单信息传递给 ReceiveService
ReceiveController.showSheetByState	showSheetByState(ReceiveSheetState state)	需查找的单据已被建立	从 ReceiveService 获取对应状态的单据
ReceiveController.approval	approval(String receiveSheetId, ReceiveSheetState state)	需审批的单据已被建立	将审批信息传递给 ReceiveService
ReceiveController.findBySheetId	findBySheetId(String id)	需查找的单据已被建立	从 ReceiveService 获取对应 ID 的单据
CompanyAccountController.addCompanyAccount	addCompanyAccount(CompanyAccountVO companyAccountVO)	无	将账户信息传给 CompanyAccountService
CompanyAccountController.updateCompanyAccount	updateCompanyAccount(CompanyAccountVO companyAccountVO)	更新的账户已经存在	将更新的账户信息传给 CompanyAccountService

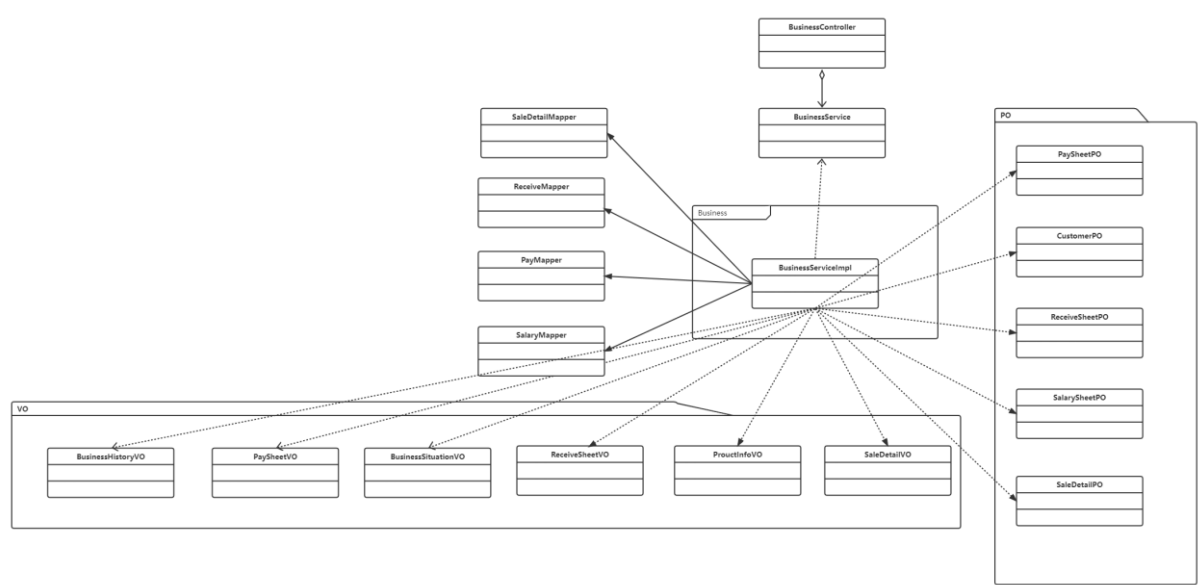


CompanyAccountController.deleteCompanyAccount	deleteCompanyAccount(Integer id)	删除的账户已经存在	CompanyAccountService 删除对应账户
CompanyAccountController.queryAllCompanyAccounts	queryAllCompanyAccounts()	无	从 CompanyAccountService 获取所有账户
CompanyAccountController.findCompanyAccountByName	findCompanyAccountByName(String name)	name 对应的账户存在	从 CompanyAccountService 中获取对应名称的账户

## Finacebl 模块需要的服务

接口名字	服务
PayService.makePaySheet	生成一个付款单据
PayService.getPaySheetByState	根据单据状态获取对应的付款单据
PayService.approval	审批单据
PayService.getPaySheetById	根据单据 ID 获取对应的付款单据
ReceiveService.makeReceiveSheet	生成一个收款单据
getReceiveSheetByState	根据单据状态获取对应的收款单据
ReceiveService.approval	审批单据
ReceiveService.getReceiveSheetById	根据单据 ID 获取对应的收款单据
CompanyAccountService.addCompanyAccount	增加公司账户
CompanyAccountService.updateCompanyAccount	更新公司账户
CompanyAccountService.deleteCompanyAccount	删除公司账户
CompanyAccountService.queryAllCompanyAccounts	查询所有公司账户
CompanyAccountService.findOneByName	根据名字查询公司账户

Businessbl 模块的关键类图



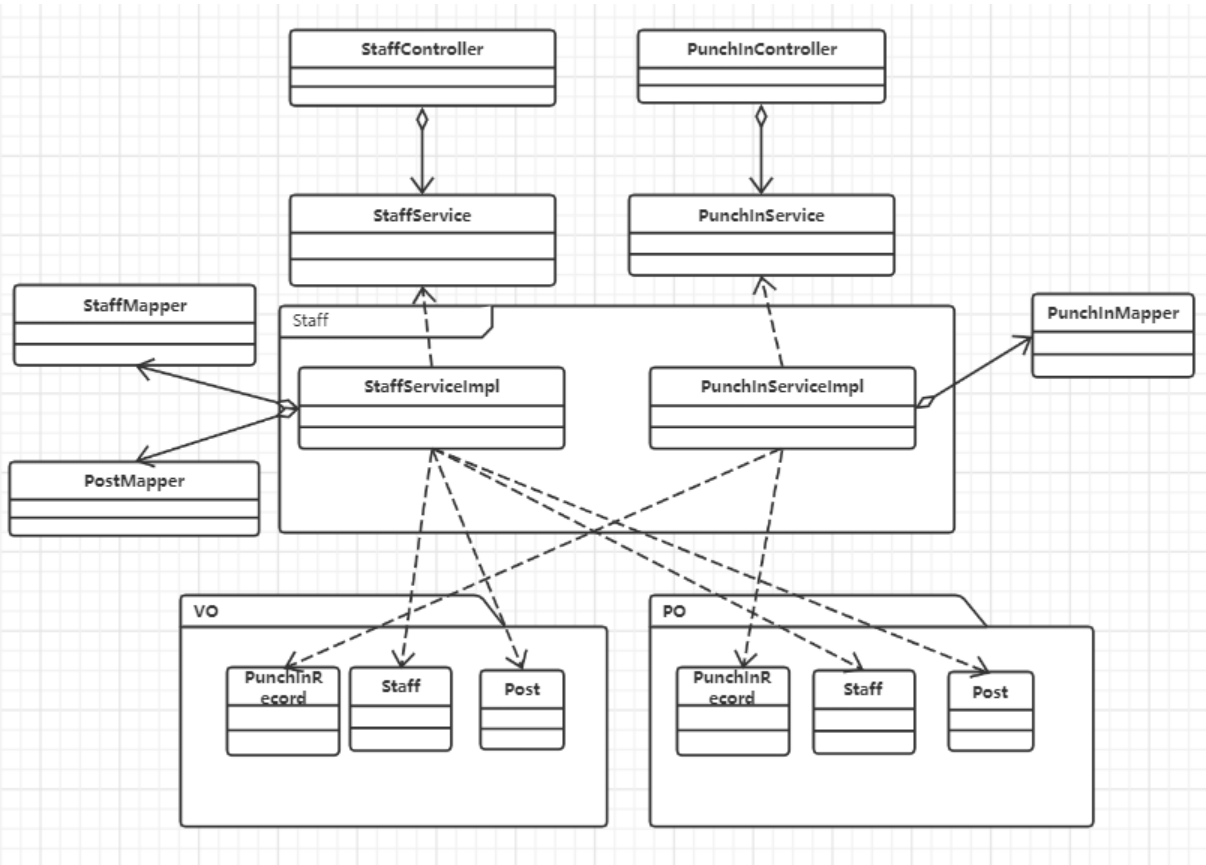
Businessbl 模块的接口规范

接口名字	语法	前置条件	后置条件
BusinessController.ge tBusinessSituation	getBusinessSituation( Date fromDate, Date toDate)	无	返回对应的经营情况 表
BusinessController.ge tSaleDetail	getSaleDetail()	无	返回所有销售明细
BusinessController.ge tBusinessSituation	getBusinessSituation	无	返回所有单据

Businessbl 模块需要的服务

服务名	服务
BusinessService.getBusinessSituation	根据给定日期获得这段时间内的经营情况
BusinessService.getSaleDetail	获得销售细节
BusinessService.getBusinessSituation	获得所有单据

Staffbl 模块的关键类图



Staffbl 模块的接口规范

接口名字	语法	前置条件	后置条件
------	----	------	------

PunchInController.punchIn	PunchIn(UserVO user)	需员工信息已录入	将打卡信息传递给 PunchInService
PunchInController.find AllPunchInRecordBy Period	findAllPunchInRecord ByPeriod(String beginDateStr,String endDateStr)	需时间格式正确（初 始时间小于结束时 间）	从 PunchInService 获 取所有的打卡信息
PunchInController.fin dPunchInRecordCont entByStaffAndPeriod	findPunchInRecordC ontentByStaffAndPeri od(int staffId, String beginDateStr, String endDateStr)	需时间格式正确（初 始时间小于结束时 间）	从 PunchInService 获取某一员工的打卡 信息
PunchInController.co untByIdAndPeriod	getPaySheetById(int id, Date t1,Date t2)	无	从 PunchInService 获 取打卡次数
StaffController.addSt aff	addStaff(StaffVO staffVO)	需填写的岗位 ID 已 被创建	将员工信息传递给 StaffService
StaffController.query All	queryAll()	无	从 StaffService 获取 对应状态的员工信息
StaffController.query AllPost	queryAllPost()	无	从 PostService 获取 对应状态的岗位信息
StaffController.delete Staff	deleteStaff(int staffId)	删除的员工已经存在	StaffService 删除对应 员工信息

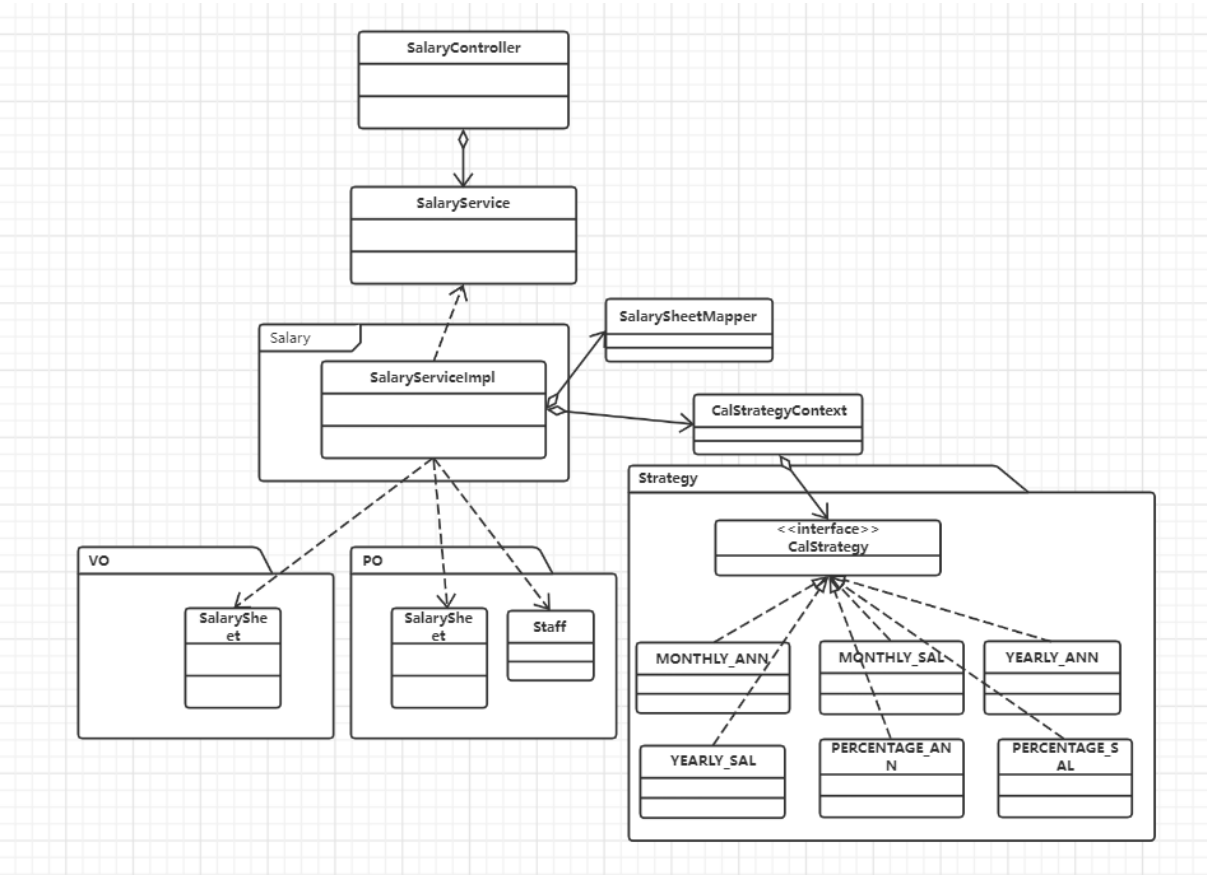
StaffController.deletePost	deletePost(int postId)	删除的岗位是否有员工与之相关联	StaffService 删除对应岗位信息
StaffController.updatePost	updatePost(PostVO postVO)	更新的岗位已经存在	PostService 更新对应岗位
StaffController.updateStaff	updateStaff(StaffVO staffVO)	更新的员工已经存在	StaffService 更新对应员工信息
StaffController.addPost	addPost(PostVO postVO)	无	将岗位信息传递给 PostService

### Staffbl 模块需要的服务

服务名	服务
PunchInService.punchIn	员工打卡
PunchInService.findAllPunchInRecordByPeriod	根据时间段查找打卡记录
PunchInService.findPunchInRecordContentByStaffAndPeriod	根据时间段和员工 ID 查找打卡记录
PunchInService.countByIdAndPeriod	根据时间段和员工 ID 查找打卡次数
StaffService.addStaff	新增员工
StaffService.queryAll	查询所有员工
StaffService.queryAllPost	查询所有岗位
StaffService.deleteStaff	删除员工
StaffService.deletePost	删除岗位
StaffService.updatePost	更新岗位

StaffService.updateStaff	更新员工
StaffService.addPost	新增岗位
StaffService.findByStaffId	根据员工 ID 查找员工
StaffService.findCalSalaryMethod	根据员工 ID 查找薪资计算方法
StaffService.findBasicSalary	根据员工 ID 查找基本薪资

Salarybl 模块的关键类图



Salarybl 模块的接口规范

接口名字	语法	前置条件	后置条件
------	----	------	------

SalaryController.makeSalary	makeSalary(UserVO userVO)	无	将打卡信息传递给 SalarySheetService
SalaryController.makeAnnual	makeAnnual(UserVO userVO)	年终奖未被制定	将薪资信息传递给 SalarySheetService
SalaryController.showSheetByState	getSalarySheetByState(SalarySheetState state)	无	从 SalarySheetService 获取工资记录
SalaryController.firstApproval	approval(String salarySheetId, SalarySheetState state)	需审批的单据已被建立	将审批信息传递给 SalarySheetService
SalaryController.secondApproval	approval(String salarySheetId, SalarySheetState state)	需审批的单据已被建立	将审批信息传递给 SalarySheetService
SalaryController.findBySheetId	findBySheetId(String id)	需查找的单据已被建立	从 SalarySheetService 获取对应 ID 的单据

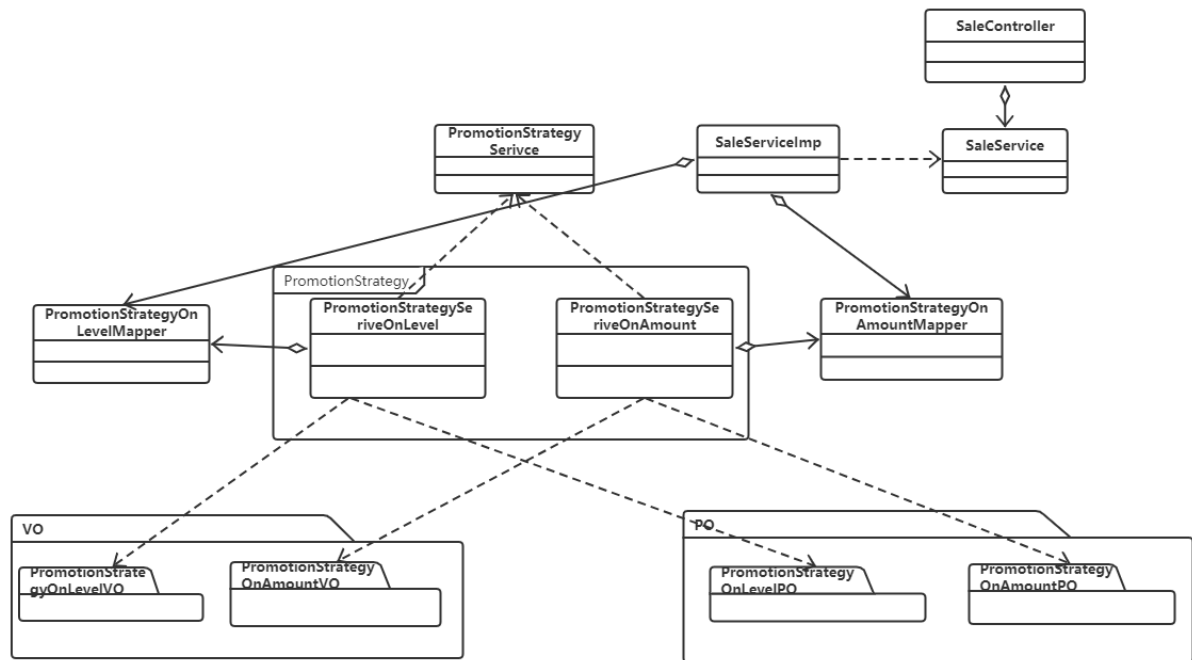
### Salarybl 模块需要的服务

服务名	服务
SalaryService.makeSalarySheet	制定工资单
SalaryService.generateSalary	生成工资条目
SalaryService.generateAnnual	生成年终奖条目
SalaryService.getSalarySheetByState	根据审批状态获取工资单
SalaryService.approval	审批工资单

SalaryService.getSalarySheetById

根据工资单 ID 获取工资单

## Promotionbl 模块的关键类图



## Promotionbl 模块的接口规范

接口名字	语法	前置条件	后置条件
SaleController.makePromotionStrategyOnLevel	makePromotionStrategyOnLevel(PromotionStrategyOnLevelVO promotionStrategyOnLevelVO)	无	将促销策略信息传递给 promotionStrategyOnLevel
SaleController.getAllPromotionStrategyOnLevel	getAllPromotionStrategyOnLevel()	无	从 promotionStrategyOnLevel 获取促销策略信息



SaleController.updatePromotionStrategyOnLevel	updatePromotionStrategyOnLevel(PromotionStrategyOnLevelVO promotionStrategyOnLevelVO)	需要编辑的策略已被建立	将促销策略信息传递给 promotionStrategyOnLevel
SaleController.deletePromotionStrategyOnLevel	deletePromotionStrategyOnLevel(Integer id)	需要删除的策略已被建立	从 promotionStrategyOnLevel 删除促销策略信息
SaleController.makePromotionStrategyOnAmount	makePromotionStrategyOnAmount(PromotionStrategyOnAmountVO promotionStrategyOnLevelVO)	无	将促销策略信息传递给 promotionStrategyOnAmount
SaleController.getAllPromotionStrategyOnAmount	getAllPromotionStrategyOnAmount()	无	从 promotionStrategyOnAmount 获取促销策略信息
SaleController.updatePromotionStrategyOnAmount	updatePromotionStrategyOnAmount(PromotionStrategyOnAmountVO promotionStrategyOnAmountVO)	需要编辑的策略已被建立	将促销策略信息传递给 promotionStrategyOnAmount
SaleController.deletePromotionStrategyOnAmount	deletePromotionStrategyOnAmount(Integer id)	需要删除的策略已被建立	从 promotionStrategyOnAmount 删除促销策略信息

## Promotionbl 模块需要的服务

服务名	服务
SaleService.makePromotionStrategyOnLevel	制定关于等级的促销策略
SaleService.getAllPromotionStrategyOnLevel	获取所有的关于等级的促销策略
SaleService.updatePromotionStrategyOnLeve	更新关于等级的促销策略

I	
SaleService.deletePromotionStrategyOnLevel	删除对应的等级促销策略
SaleService.makePromotionStrategyOnAmount	制定关于总价的促销策略
SaleService.getAllPromotionStrategyOnAmount	获取所有的关于总价的促销策略
SaleService.updatePromotionStrategyOnAmount	更新关于总价的促销策略
SaleService.deletePromotionStrategyOnAmount	删除对应的总价促销策略

## 5.4 数据层分解

### 5.4.1 职责

模块	职责
PaySheetMapper	持久化数据库的接口。提供付款单的集体载入、集体保存、增、删、改、查服务。
ReceiveSheetMapper	持久化数据库的接口。提供收款单的集体载入、集体保存、增、删、改、查服务。
SalarySheetMapper	持久化数据库的接口。提供工资单的集体载入、集体保存、增、删、改、查服务。
CompanyAccountMapper	持久化数据库的接口。提供公司账户的增、删、改、查服务。
SaleDetailMapper	持久化数据库的接口。提供销售明细的查找服务。
StaffMapper	持久化数据库的接口。提供员工信息的增、删、改、查服务。
PostMapper	持久化数据库的接口。提供岗位信息的增、删、改、查服务。
PunchInMapper	持久化数据库的接口。提供打卡信息的增、查服务。
PromotionStrategyOnLevelMapper	持久化数据库接口。提供等级促销策略信息的增、删、改、查服务。

PromotionStrategyOnAmountMapper	持久化数据库接口。提供总价促销策略信息的增、删、改、查服务。
---------------------------------	--------------------------------

## 5.4.2 接口规范

接口名称	语法	前置条件	后置条件
PaySheetMapper.saveSheet	saveSheet(PaySheetPO toSave)	无	在数据库中增添一条 PO 记录
PaySheetMapper.saveBatchSheetContent	saveBatchSheetContent(List<PaySheetContentPO> paySheetContent)	无	在数据库中增添一条 PO 记录
PaySheetMapper.findAllSheet	findAllSheet()	无	返回对应 PO 结果
PaySheetMapper.findSheetById	findSheetById(String id)	数据库中存在相同 ID 的 PO	返回对应 PO 结果
PaySheetMapper.findContentBySheetId	findContentBySheetId(String sheetId)	数据库中存在相同 ID 的 PO	返回对应 PO 结果
PaySheetMapper.findAllSheetByState	findAllSheetByState(PaySheetState state)	无	返回对应 PO 结果
PaySheetMapper.updateSheetState	updateSheetState(String sheetId, PaySheetState state)	数据库中存在相同状态的 PO	更新一个 PO
PaySheetMapper.updateSheetStateOnPrev	updateSheetStateOnPrev(String sheetId, PaySheetState prev, PaySheetState state)	数据库中存在相同状态的 PO	更新一个 PO
PaySheetMapper.getLatestSheet	getLatestSheet()	无	返回对应 PO 结果
PaySheetMapper.findAllSheetByDate	findAllSheetByDate(Date fromDate,	无	返回对应 PO 结果

	Date toDate)		
ReceiveSheetMapper.saveSheet	saveSheet(ReceiveSheetPO toSave)	无	在数据库中增添一条 PO 记录
ReceiveSheetMapper.saveBatchSheetContent	saveBatchSheetContent(List<ReceiveSheetContentPO> receiveSheetContent)	无	在数据库中增添一条 PO 记录
ReceiveSheetMapper.findAllSheet	findAllSheet()	无	返回对应 PO 结果
ReceiveSheetMapper.findSheetById	findSheetById(String id)	数据库中存在相同 ID 的 PO	返回对应 PO 结果
ReceiveSheetMapper.findContentBySheetId	findContentBySheetId(String sheetId)	数据库中存在相同 ID 的 PO	返回对应 PO 结果
ReceiveSheetMapper.findAllSheetByState	findAllSheetByState(ReceiveSheetState state)	无	返回对应 PO 结果
ReceiveSheetMapper.updateSheetState	updateSheetState(String sheetId, ReceiveSheetState state)	数据库中存在相同状态的 PO	更新一个 PO
ReceiveSheetMapper.updateSheetStateOnPrev	updateSheetStateOnPrev(String sheetId, ReceiveSheetState prev, ReceiveSheetState state)	数据库中存在相同状态的 PO	更新一个 PO
ReceiveSheetMapper.getLatestSheet	getLatestSheet()	无	返回对应 PO 结果
ReceiveSheetMapper.findAllSheetByDate	findAllSheetByDate(Date fromDate, Date toDate)	无	返回对应 PO 结果

CompanyAccountMapper.addCompanyAccount	void addCompanyAccount(CompanyAccountPO companyAccountPO)	无	增加一个 po
CompanyAccountMapper.updateCompanyAccount	int updateCompanyAccount(CompanyAccountPO companyAccountPO)	数据库中存在对应 PO	更新一个 po
CompanyAccountMapper.deleteCompanyAccount	int deleteCompanyAccount(Integer id)	数据库中存在对应 id 的 po	删除一个 po
CompanyAccountMapper.queryAllAccounts	List<CompanyAccountPO> queryAllAccounts()	无	返回所有 po
CompanyAccountMapper.findOneByName	CompanyAccountPO findOneByName(String name)	数据库中存在对应名称的 po	返回对应 po
CompanyAccountMapper.findOneById	CompanyAccountPO findOneById(int id)	数据库中存在对应 id 的 po	返回对应 po
SaleDetailMapper.getSaleDetail	List<SaleDetailPO> getSailDetail()	无	返回所有 po
StaffMapper.createStaff	int createStaff(StaffPO staffPO)	无	增加一个 po
StaffMapper.findByName	StaffPO findByName(String staffName)	数据库中存在对应 po	返回对应 po
StaffMapper.findById	StaffPO findById(int id)	数据库中存在对应 po	返回对应 po
StaffMapper.queryAll	List<StaffPO> queryAll()	数据库中存在对应 po	返回对应 po
StaffMapper.findPost	String findPost(int id)	数据库中存在对应 po	返回对应字段
StaffMapper.findBasicSalary	int findBasicSalary(int id)	数据库中存在对应 po	返回对应字段
StaffMapper.findCalSalaryMethod	int findBasicSalary(int id)	数据库中存在对应 po	返回对应字段
StaffMapper.checkIf	String	数据库中存在对应	返回对应字段

AnEmptyPost	findCalSalaryMethod(int id)	po	
StaffMapper.deleteStaff	void deleteStaff(int postId)	数据库中存在对应 po	删除一个 po
PostMapper.queryAllPost	List<PostPO> queryAllPost()	数据库中存在对应 po	返回对应 po
PostMapper.findIfHasPost	int findIfHasPost(int postId)	无	返回对应字段
PostMapper.deletePost	void deletePost(int postId)	数据库中存在对应 po	返回对应 po
PostMapper.updatePost	int updatePost(PostPO postPO)	数据库中存在对应 po	更新一个 po
PostMapper.addPost	void addPost(int postId)	无	增加一个 po
PunchInMapper.punchIn	int punchIn(int staffId, String staffName, Date punchInTime)	无	增加一个 po
PunchInMapper.checkPunchIn	int checkPunchIn(int staffId, Date punchInTime)	数据库中存在对应 po	返回对应字段
PunchInMapper.findByIdAndPeriod	List<PunchInRecordContentPO> findByIdAndPeriod(int staffId, Date t1, Date t2)	数据库中存在对应 po	返回对应 po
PunchInMapper.countByIdAndPeriod	int countByIdAndPeriod(int staffId, Date t1, Date t2)	数据库中存在对应 po	返回对应字段
PunchInMapper.findAllRecord	List<PunchInRecordPO> findAllRecord(Date t1, Date t2)	数据库中存在对应 po	返回对应 po
SalarySheetMapper.saveSheet	int saveSheet(SalarySheetPO toSave)	无	增加一个 po
SalarySheetMapper.findIfSalaryHasBeenMade	int findIfSalaryHasBeenMade(int staffId, Date	数据库中存在对应 po	返回对应字段

	d1,Date d2)		
SalarySheetMapper.findIfSalaryHasBeenMadeGM	int findIfSalaryHasBeenMadeGM(int staffId,int year)	数据库中是否存在对应 po	返回对应字段
SalarySheetMapper.findAllSheet	List<SalarySheetPO> findAllSheet()	数据库中是否存在对应 po	返回对应 po
SalarySheetMapper.findSheetById	SalarySheetPO findSheetById(String id)	数据库中是否存在对应 po	返回对应 po
SalarySheetMapper.findSheetByState	List<SalarySheetPO> findAllSheetByState(SalarySheetState state)	数据库中是否存在对应 po	返回对应 po
SalarySheetMapper.updateSheetState	int updateSheetState(String sheetId,SalarySheetState state)	数据库中是否存在对应 po	返回对应字段
SalarySheetMapper.updateSheetStateOnPrev	int updateSheetStateOnPrev(String sheetId,SalarySheetState prev,SalarySheetState state)	数据库中是否存在对应 po	返回对应字段
SalarySheetMapper.getLatestSheet	SalarySheetPO getLatestSheet()	数据库中是否存在对应 po	返回对应 po
SalarySheetMapper.findAllSheetByDate	List<SalarySheetPO> findAllSheetByDate(Date fromDate,Date toDate)	数据库中是否存在对应 po	返回对应 po
SalarySheetMapper.findAnnual	SalarySheetPO findAnnual(int staffId,int year)	数据库中是否存在对应 po	返回对应 po
PromotionStrategyOnLevelMapper.setPromotionStrategyOnLevel	setPromotionStrategyOnLevel(PromotionStrategyOnLevelPO promotionStrategyOnLevelPO)	无	在数据库中增添一条 PO 记录
PromotionStrategyOnLevelMapper.getAllPromotionStrategyOnLevel	List<PromotionStrategyOnLevelPO> getAllPromotionStrat	无	返回对应 PO 结果

nLevel	egyOnLevel()		
PromotionStrategyOnLevelMapper.updateByld	int updateByld(PromotionStrategyOnLevelPO promotionStrategyOnLevelPO);	数据库中存在对应 PO	更新一个 PO
PromotionStrategyOnLevelMapper.deleteByld	int deleteByld(Integer id)	数据库中存在对应 PO	删除一个 PO
PromotionStrategyOnAmountMapper.setPromotionStrategyOnAmount	int setPromotionStrategyOnAmount(PromotionStrategyOnAmountPO promotionStrategyOnAmountPO)	无	在数据库中增添一条 PO 记录
PromotionStrategyOnAmountMapper.getAllPromotionStrategyOnAmount	List<PromotionStrategyOnAmountPO> getAllPromotionStrategyOnAmount()	无	返回对应 PO 结果
PromotionStrategyOnAmountMapper.updateByld	int updateByld(PromotionStrategyOnAmountPO promotionStrategyOnAmountPO)	数据库中存在对应 PO	更新一个 PO



PromotionStrategyOnAmountMapper.deleteByld	int deleteByld(Integer id)	数据库中存在对应 PO	删除一个 PO
--------------------------------------------	----------------------------	-------------	---------

## 6. 信息视角

PaySheetPO

```
package com.nju.edu.erp.model.po.pay;

import java.math.BigDecimal;

import com.nju.edu.erp.enums.sheetState.PaySheetState;

import com.nju.edu.erp.enums.sheetState.PurchaseSheetState;

import lombok.AllArgsConstructor;

import lombok.Builder;

import lombok.Data;

import lombok.NoArgsConstructor;

import java.math.BigDecimal;

import java.util.Date;

@Data

@AllArgsConstructor

@NoArgsConstructor

@Builder

public class PaySheetPO {

    /**

     * 付款单据号

     */

    private String id;

    /**

     * 银行账户

     */

    private String bankAccount;
```

PaySheetContentPO

```
package com.nju.edu.erp.model.po.pay;

import java.math.BigDecimal;

import com.nju.edu.erp.enums.sheetState.PurchaseSheetState;

import lombok.AllArgsConstructor;

import lombok.Builder;

import lombok.Data;

import lombok.NoArgsConstructor;

import java.math.BigDecimal;

import java.util.Date;

@Data

@AllArgsConstructor

@NoArgsConstructor

@Builder

public class PaySheetContentPO {

    /**

     * 条目清单编号

     */

    private Integer id;

    /**

     * 相关付款单据号

     */

    private String paySheetId;
```

ReceiveSheetPO

```
package com.nju.edu.erp.model.po.receive;

import java.math.BigDecimal;

import com.nju.edu.erp.enums.sheetState.PurchaseSheetState;
import com.nju.edu.erp.enums.sheetState.ReceiveSheetState;
import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;

import java.math.BigDecimal;
import java.util.Date;

@Data
@AllArgsConstructor
@NoArgsConstructor
@Builder

public class ReceiveSheetPO {

    /**
     * 收款单据编号
     */
    private String id;

    /**
     * 客户名称
     */
}
```

ReceiveSheetContentPO



```
package com.nju.edu.erp.model.po.receive;

import com.nju.edu.erp.enums.sheetState.PurchaseSheetState;

import lombok.AllArgsConstructor;

import lombok.Builder;

import lombok.Data;

import lombok.NoArgsConstructor;

import java.math.BigDecimal;

import java.util.Date;

@Data

@AllArgsConstructor

@NoArgsConstructor

@Builder

public class ReceiveSheetContentPO {

    /**

     * 转账列表编号

     */

    private Integer id;

    /**

     * 相关收款单 id

     */

    String receiveSheetId;

    /**
```

CompanyAccountPO

```
package com.nju.edu.erp.model.po;
```

```
import lombok.AllArgsConstructor;
```

```
import lombok.Builder;
```

```
import lombok.Data;
```

```
import lombok.NoArgsConstructor;
```

```
import java.math.BigDecimal;
```

```
@Data
```

```
@AllArgsConstructor
```

```
@NoArgsConstructor
```

```
@Builder
```

```
public class CompanyAccountPO {
```

```
    /**
```

```
     * 账户 ID
```

```
    */
```

```
    private Integer id;
```

```
    /**
```

```
     * 账户名称
```

```
    */
```

```
    private String name;
```

**SaleDetailPO**

```
package com.nju.edu.erp.model.po;
```

```
import lombok.AllArgsConstructor;
```

```
import lombok.Builder;
```

```
import lombok.Data;
```

```
import lombok.NoArgsConstructor;
```

```
import java.math.BigDecimal;
```

```
import java.util.Date;
```

```
@Data
```

```
@AllArgsConstructor
```

```
@NoArgsConstructor
```

```
@Builder
```

```
public class SaleDetailPO {
```

```
    /**
```

```
     * 销售/退货时间
```

```
    */
```

```
    private Date createTime;
```

```
    /**
```

```
     * 商品 id
```

```
    */
```

```
    private String pid;
```

```
    /**
```

```
     * 客户 id
```

SalarySheetPO

```
package com.nju.edu.erp.model.po.salary;

import java.math.BigDecimal;

import com.nju.edu.erp.enums.sheetState.SalarySheetState;

import lombok.AllArgsConstructor;

import lombok.Builder;

import lombok.Data;

import lombok.NoArgsConstructor;

import java.util.Date;

@Data

@AllArgsConstructor

@NoArgsConstructor

@Builder

public class SalarySheetPO {

    /**

     * 工资单据号

     */

    private String id ;

    /**

     * 员工 id

     */

    private Integer staffId;

    /**
```

**PunchInRecordPO**



```
package com.nju.edu.erp.model.po.punchIn;
```

```
import lombok.AllArgsConstructor;
```

```
import lombok.Builder;
```

```
import lombok.Data;
```

```
import lombok.NoArgsConstructor;
```

```
import java.util.Date;
```

```
@Data
```

```
@AllArgsConstructor
```

```
@NoArgsConstructor
```

```
@Builder
```

```
public class PunchInRecordPO {
```

```
    //打卡条目记录表
```

```
    /**
```

```
     * 员工 id
```

```
     */
```

```
    private int staffId;
```

```
    /**
```

```
     * 员工姓名
```

```
     */
```

```
    private String staffName;
```

PunchInRecordContentPO

```
package com.nju.edu.erp.model.po.punchIn;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;

import java.util.Date;

@Data
@AllArgsConstructor
@NoArgsConstructor
@Builder

public class PunchInRecordContentPO {

    //打卡记录表

    /**
     * 员工 id
     */
    private int staffId;

    /**
     * 员工姓名
     */
    private String staffName;
```

PostPO

```
package com.nju.edu.erp.model.po;
```

```
import lombok.AllArgsConstructor;
```

```
import lombok.Builder;
```

```
import lombok.Data;
```

```
import lombok.NoArgsConstructor;
```

```
import java.util.Date;
```

```
@Data
```

```
@AllArgsConstructor
```

```
@NoArgsConstructor
```

```
@Builder
```

```
public class PostPO {
```

```
    /**
```

```
     * 岗位 id
```

```
    */
```

```
    private int id;
```

```
    /**
```

```
     * 岗位名称
```

```
    */
```

```
    private String postName;
```

**StaffPO**

```
package com.nju.edu.erp.model.po;
```

```
import lombok.AllArgsConstructor;
```

```
import lombok.Data;
```

```
import lombok.NoArgsConstructor;
```

```
import java.util.Date;
```

```
@Data
```

```
@AllArgsConstructor
```

```
@NoArgsConstructor
```

```
public class StaffPO {
```

```
    /**
```

```
     * id
```

```
     */
```

```
    private int id;
```

```
    /**
```

```
     * 姓名
```

```
     */
```

```
    private String name;
```

```
    /**
```

```
     * 性别
```

```
     */
```





PromotionStrategyOnLevelPO

```
package com.nju.edu.erp.model.po;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;
import java.math.BigDecimal;
import java.util.Date;

@Data
@AllArgsConstructor
@NoArgsConstructor
@Builder

public class PromotionStrategyOnLevelPO {

    /**
     * id
     */
    private Integer id;

    /**
     * 创建时间
     */
    private Date create_time;

    /**
```



PromotionStrategyOnAmountPO

```
package com.nju.edu.erp.model.po;

import lombok.AllArgsConstructor;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;

import java.math.BigDecimal;
import java.util.Date;

@Data
@AllArgsConstructor
@NoArgsConstructor
@Builder

public class PromotionStrategyOnAmountPO {

    /**
     * 编号
     */
    private Integer id;

    /**
     * 总价区间最低价
     */
    private Integer min_amount;
```

