## 数据层的分解

数据层的开发包图请参见体系结构描述文档图4。

数据层采用代理模式，一个接口Service对应两个实现类Impl和Proxy，方便期初建账，系统日志等的实现。

### AccountDataService

1. 模块概述

AccountDataService模块提供对系统账号的基本操作

1. 模块内类接口规范

AccountDataService接口规范如下表

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| AccountDataService. getAccountPO | 语法 | public AccountPO getAccountPO (String accountID) |
| 前置条件 | AccountID在数据中存在 |
| 后置条件 | 查找返回相应的AccountPO |
| AccountDataService.getAccountPOs | 语法 | public List<AccountPO> getAccountPOs() |
| 前置条件 | 无 |
| 后置条件 | 返回所有的AccountPO |
| AccountDataService.insert | 语法 | Public OperationMessage insert  (AccountPO po) |
| 前置条件 | 同样的po在数据中不存在，参数为插入对象 |
| 后置条件 | 增加一条po数据，返回结果信息 |
| AccountDataService.delete | 语法 | public OperationMessage delete  (String name) |
| 前置条件 | 同样的po在数据中存在,传入name |
| 后置条件 | 删除po，返回结果信息 |
| AccountDataService.update | 语法 | public OperationMessage update  (AccountPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po，返回结果信息 |
| AccountDataService.checkAccount | 语法 | Public boolean checkAccount(String id , String password) |
| 前置条件 | CheckID返回true |
| 后置条件 | 若ID与password匹配，则返回true，否则，返回false |
| AccountDataService.setAccount | 语法 | **public** OperationMessage setAccount(String id , **boolean** isOnline) |
| 前置条件 | 传入账户的ID和当前状态 |
| 后置条件 | 返回修改该账户状态的操作信息 |

### CacheDataService

(1)模块概述

CacheDataService模块提供对对数据缓存的一些支持

(2)模块内类接口规范

CacheDataService接口规范如下表

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CacheDataService. getLatestVersionID | 语法 | **public** **default** **long** getLatestVersionID() |
| 前置条件 | 无 |
| 后置条件 | 返回最新的期初版本号 |
| CacheDataService.getOperation | 语法 | **public** **default** List<Operation> getOperation(**long** localVersion) |
| 前置条件 | 传入当前的版本号 |
| 后置条件 | 返回操作的列表 |

### ChatRemindService

(1)模块概述

ChatRemindService模块提供对公司体系调整的操作（营业厅，中转中心，车辆的管理）

(2)模块内类接口规范

ChatRemindService接口规范如下表

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CacheDataService. getLatestVersionID | 语法 | **public** **default** **long** getLatestVersionID() |
| 前置条件 | 无 |
| 后置条件 | 返回最新的期初版本号 |
| CacheDataService.getOperation | 语法 | **public** **default** List<Operation> getOperation(**long** localVersion) |
| 前置条件 | 传入当前的版本号 |
| 后置条件 | 返回操作的列表 |

### CompanyDataService

1. 模块概述

CompanyDataService模块提供对公司体系调整的操作（营业厅，中转中心，车辆的管理）

1. 模块内类接口规范

CompanyDataService接口规范如下表

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| companyDataService.getHall | 语法 | public List<HallPO> getHall() |
| 前置条件 | 无 |
| 后置条件 | 返回公司所有营业厅的信息 |
| companyDataService.getHallByID | 语法 | **public** HallPO getHallByID(String ID) |
| 前置条件 | 传入营业厅编号 |
| 后置条件 | 返回该营业厅的信息 |
| companyDataService.newHallID | 语法 | public String newHallID(String centerID) |
| 前置条件 | 上属中转中心编号 |
| 后置条件 | 返回一个合法的新编号 |
| companyDataService.addHall | 语法 | public OperationMessage addHall(HallPO Hall) |
| 前置条件 | Hall为新营业厅的信息 |
| 后置条件 | 若新建成功，返回成功；否则，返回失败原因 |
| companyDataService.deleteHall | 语法 | public OperationMessage deleteHall(HallPO hall) |
| 前置条件 | hall为待删除营业厅信息；已经获取所有的营业厅信息 |
| 后置条件 | 若删除成功返回成功，否则返回失败原因 |
| companyDataService.modifyHall | 语法 | public OperationMessage modifyHall(HallPO hall) |
| 前置条件 | hall为修改后的营业厅信息 |
| 后置条件 | 若修改成功，返回成功；否则返回失败原因 |
| companyDataService.getCenter | 语法 | public List<CenterPO> getCenter() |
| 前置条件 | 无 |
| 后置条件 | 获取所有中转中心的信息 |
| companyDataService.getCenterByID | 语法 | **public** CenterPO getCenterByID(String ID) |
| 前置条件 | 传入中转中心编号 |
| 后置条件 | 返回该中转中心信息 |
| companyDataService.addCenter | 语法 | public OperationMessage addCenter(CenterPO center) |
| 前置条件 | center为新建中转中心的信息 |
| 后置条件 | 若新建成功，返回成功；否则返回失败原因 |
| companyDataService.deleteCenter | 语法 | public OperationMessage deleteCenter(CenterPO center) |
| 前置条件 | center为待删除中转中心信息 |
| 后置条件 | 若删除成功，返回成功，否则返回失败原因 |
| companyDataService.modifyCenter | 语法 | public OperationMessage modifyCenter(CenterPO center) |
| 前置条件 | center为待修改中转中心信息 |
| 后置条件 | 若修改成功，返回成功；否则，返回失败原因 |
| companyDataService.getCars | 语法 | **public** List<CarPO> getCars(String unitID) |
| 前置条件 | 传入营业厅编号 |
| 后置条件 | 返回该营业厅所有的车辆信息 |
| companyDataService.getCar | 语法 | public List<CarPO> getCar() |
| 前置条件 | 无 |
| 后置条件 | 返回公司车辆信息 |
| companyDataService.newCarID | 语法 | public String newCarID() |
| 前置条件 | 无 |
| 后置条件 | 返回一个合法的新编号 |
| companyDataService.addCar | 语法 | public OperationMessage addCar(CarPO car) |
| 前置条件 | car为新增车辆的信息 |
| 后置条件 | 若新建成功则返回成功，否则返回失败原因 |
| companyDataService.deleteCar | 语法 | public OperationMessage deleteCar(CarPO car) |
| 前置条件 | car为待删除的车辆信息 |
| 后置条件 | 若删除成功返回成功，否则返回失败原因 |
| companyDataService.modifyCar | 语法 | public OperationMessage modifyCar(CarPO car) |
| 前置条件 | car为修改后的车辆信息 |
| 后置条件 | 若修改成功则返回成功；否则返回失败原因 |
| companyDataService.availableCar | 语法 | **public** List<CarPO> availableCar(String unitID) |
| 前置条件 | 传入营业厅编号 |
| 后置条件 | 返回该营业厅当前可用的车辆列表 |

### ConfigurationDataService

1. 模块概述

ConfigurationDataService模块提供有关公司基础信息的操作的操作（常量，薪水策略等）

1. 模块内类接口规范

ConfigurationDataService接口规范如下表

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| configurationDataService.getCity2D | 语法 | **public** City2DPO getCity2D(String ID) |
| 前置条件 | 传入城市名称 |
| 后置条件 | 返回城市的信息 |
| configurationDataService.getAllCity2D | 语法 | **public** List<City2DPO> getAllCity2D() |
| 前置条件 | 无 |
| 后置条件 | 返回系统中所有城市信息 |
| configurationDataService.newCity2D | 语法 | **public** OperationMessage newCity2D(City2DPO po) |
| 前置条件 | 传入新的城市信息 |
| 后置条件 | 返回添加的结果 |
| configurationDataService.deleteCity2D | 语法 | **public** OperationMessage deleteCity2D(String name) |
| 前置条件 | 传入城市名称 |
| 后置条件 | 返回删除目标城市的操作信息 |
| configurationDataService.modifyCity2D | 语法 | **public** OperationMessage modifyCity2D(City2DPO po) |
| 前置条件 | 出入需要修改的城市信息 |
| 后置条件 | 返回修改城市信息的操作结果 |
| configurationDataService.getSalaryStrategy | 语法 | public List<SalaryStrategyPO> getSalaryStrategy () |
| 前置条件 | 无 |
| 后置条件 | 返回员工的薪水策略信息 |
| configurationDataService.newSalaryStrategy | 语法 | **public** OperationMessage newSalaryStrategy(List<SalaryStrategyPO> po) |
| 前置条件 | 传入新的薪水策略列表（适用于期初建账） |
| 后置条件 | 返回新建结果 |
| configurationDataService.modifySalaryStrategy | 语法 | public OperationMessage modifySalaryStrategy (SalaryStrategyPO SalaryStrategy) |
| 前置条件 | SalaryStrategy为修改的薪水策略 |
| 后置条件 | 若修改成功则返回成功；否则返回失败原因 |
| configurationDataService.getPack | 语法 | public List<PackPO>  getPack() |
| 前置条件 | 无 |
| 后置条件 | 返回包装费用信息 |
| configurationDataService.modifyPack | 语法 | public OperationMessage modifyPack(PackPO pack) |
| 前置条件 | pack为修改后的包装信息 |
| 后置条件 | 若修改成功返回成功；否则返回失败原因 |
| configurationDataService.getPrice | 语法 | public PricePO getPrice() |
| 前置条件 | 无 |
| 后置条件 | 返回当前快递价格 |
| configurationDataService.modifyPrice | 语法 | public OperationMessage modifyPrice(PricePO price) |
| 前置条件 | price为修改后价格信息 |
| 后置条件 | 若修改成功返回成功；否则返回失败原因 |
| configurationDataService.newPrice | 语法 | **public** OperationMessage modifyPrice(PricePO price) |
| 前置条件 | 传入新的价格对象（适用于期初建账） |
| 后置条件 | 返回新建结果 |
| configurationDataService.getProportion | 语法 | public ProportionPO getProportion () |
| 前置条件 | 无 |
| 后置条件 | 返回不同快递收费比例 |
| configurationDataService.modifyProportion | 语法 | public OperationMessage modifyProportion(ProportionPO proportion) |
| 前置条件 | proportion为修改后的比例 |
| 后置条件 | 若修改成功，返回成功；否则返回失败原因 |
| configurationDataService.newProportion | 语法 | **public** OperationMessage newProportion(ProportionPO po) |
| 前置条件 | 传入新的比例对象（适用于期初建账） |
| 后置条件 | 返回新建结果 |

### DeliverDataService

1. 模块概述

DeliverDataService模块提供货物派送的一系列操作（派件单）

1. 模块内类接口规范

DeliverDataService接口规范如下表

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| DeliverDataService.insert | 语法 | public OperationMessage insert  (DeliverPO po) |
| 前置条件 | 同样的po在数据中不存在，参数为插入对象 |
| 后置条件 | 增加一条po数据，返回结果信息 |
| DeliverDataService.getFormPO | 语法 | public DeliverPO getFormPO(String id) |
| 前置条件 | 参数为派件单号 |
| 后置条件 | 查找返回相应的结果 |
| DeliverDataService.delete | 语法 | public OperationMessage delete  (String id) |
| 前置条件 | 同样的po在数据中存在,传入id |
| 后置条件 | 删除po，返回结果信息 |
| DeliverDataService.update | 语法 | public OperationMessage update  (DeliverPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po，返回结果信息 |
| DeliverDataService.newID | 语法 | public String newID(String unitID) |
| 前置条件 | 传入机构编号 |
| 后置条件 | 返回一个新的派件单ID |
| DeliverDataService. clear | 语法 | public OperationMessage clear () |
| 前置条件 | 无 |
| 后置条件 | 清空DeliverPO数据 |
| DeliverDataService. getAll | 语法 | public List< DeliverPO >  getAll () |
| 前置条件 | 无 |
| 后置条件 | 返回所有派件单PO数据 |
| DeliverDataService.available | 语法 | **public** List<String> available(String HallID) |
| 前置条件 | 传入营业厅编号 |
| 后置条件 | 返回该厅当前带派送订单的清单 |
| DeliverDataService.searchAsPerson | 语法 | **public** List<String> searchAsPerson(String ID) |
| 前置条件 | 传入快递员编号 |
| 后置条件 | 返回该快递员的任务清单 |
| DeliverDataService.getHistory | 语法 | **public** List<DeliverPO> getHistory(String creatorID) |
| 前置条件 | 传入员工ID |
| 后置条件 | 返回该员工创建的该单列表 |

### ExamineService

1. 模块概述

ExamineService模块提供有关审批单据的功能

1. 模块内类接口规范

ExamineService接口规范如下表

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ExamineSubmitService. submit | 语法 | **public** OperationMessage submit(FormPO form) |
| 前置条件 | 传入要提交的表单信息 |
| 后置条件 | 返回操作结果 |
| ExamineManageService .getForms | 语法 | **public** List<FormPO> getForms() |
| 前置条件 | 无 |
| 后置条件 | 当前待审批的表单清单 |
| ExamineManageService .modifyForm | 语法 | **public** OperationMessage modifyForm(FormPO form) |
| 前置条件 | 传入待修改的表单信息 |
| 后置条件 | 返回操作结果 |
| ExamineManageService.passForm | 语法 | **public** OperationMessage passForm(List<FormPO> forms) |
| 前置条件 | 传入待通过的表单清单 |
| 后置条件 | 返回操作结果 |
| ExamineManageService.deleteForm | 语法 | **public** OperationMessage deleteForm(List<FormPO> forms) |
| 前置条件 | 传入待删除的表单清单 |
| 后置条件 | 返回操作结果 |
| ExamineManageService.getQueue | 语法 | **public** ExamineQueue getQueue() |
| 前置条件 | 无 |
| 后置条件 | 返回服务器端表单队列对象 |

### FinanceDataService

1. 模块概述

FinanceDataService模块提供有关财务支出审查的操作（收款单，付款单）

1. 模块内类接口规范

FinanceDataService接口规范如下表

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| PaymentDataService.insert | 语法 | public OperationMessage insert  (PaymentPO po) |
| 前置条件 | 同样的po在数据中不存在，参数为插入对象 |
| 后置条件 | 增加一条po数据，返回结果信息 |
| PaymentDataService.getFormPO | 语法 | public PaymentPO getFormPO(String id) |
| 前置条件 | 参数为付款单号 |
| 后置条件 | 查找返回相应的结果 |
| PaymentDataService.delete | 语法 | public OperationMessage delete  (String id) |
| 前置条件 | 同样的po在数据中存在,传入id |
| 后置条件 | 删除po，返回结果信息 |
| PaymentDataService.update | 语法 | public OperationMessage update  (PaymentPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po，返回结果信息 |
| PaymentDataService.newID | 语法 | public String newID(String unitID) |
| 前置条件 | 传入机构编号 |
| 后置条件 | 返回一个新的付款单ID |
| PaymentDataService. clear | 语法 | public OperationMessage clear () |
| 前置条件 | 无 |
| 后置条件 | 清空PaymentPO数据 |
| PaymentDataService. getAll | 语法 | public List< PaymentPO >  getAll () |
| 前置条件 | 无 |
| 后置条件 | 返回所有付款单单PO数据 |
| PaymentDataService.getByTime | 语法 | **public** List<PaymentPO> getByTime(Calendar start, Calendar end) |
| 前置条件 | 传入起始时间，截止时间 |
| 后置条件 | 返回区间段的付款单对象 |
| PaymentDataService.getHistory | 语法 | **public** List<PaymentPO> getHistory(String creatorID) |
| 前置条件 | 传入员工ID |
| 后置条件 | 返回该员工创建的该单列表 |
| BankAccountDataService.getByAccID | 语法 | **public** List<FormPO> getByAccID(String ID) |
| 前置条件 | 传入一个银行账户ID |
| 后置条件 | 返回与其有关的单据信息 |
| BankAccountDataService.getAll | 语法 | **public** List<BankAccountPO> getAll() |
| 前置条件 | 无 |
| 后置条件 | 返回全部银行账户信息 |
| BankAccountDataService.clear | 语法 | **public** OperationMessage clear() |
| 前置条件 | 无 |
| 后置条件 | 返回清空操作结果 |
| BankAccountDataService.updata | 语法 | **public** OperationMessage update(BankAccountPO po) |
| 前置条件 | 传入待修改的银行账户信息 |
| 后置条件 | 返回修改操作结果 |
| BankAccountDataService.delete | 语法 | **public** OperationMessage delete(String id) |
| 前置条件 | 传入银行账户id |
| 后置条件 | 返回删除操作结果 |
| BankAccountDataService.insert | 语法 | **public** OperationMessage insert(BankAccountPO po) |
| 前置条件 | 传入新的银行账户信息 |
| 后置条件 | 返回新建操作结果 |
| BankAccountDataService.getBankIDByName | 语法 | **public** String getBankIDByName(String bankName) |
| 前置条件 | 传入一个名称 |
| 后置条件 | 返回该名称对应的账户编号 |
| BankAccountDataService.getBankAccount | 语法 | **public** BankAccountPO getBankAccount(String bankID) |
| 前置条件 | 传入银行账户 |
| 后置条件 | 返回该账户的信息 |
| BankAccountDataService. getNewBankID | 语法 | public String getNewBankID() |
| 前置条件 | 无 |
| 后置条件 | 返回未使用过的银行账户ID号 |
| BankAccountDataService.  checkIsNameUsed | 语法 | public boolean  checkIsNameUsed(String name) |
| 前置条件 | 无 |
| 后置条件 | 检查传入的账户名是否已经在使用 |
| RevenueDataService.insert | 语法 | public OperationMessage insert  (RevenuePO po) |
| 前置条件 | 同样的po在数据中不存在，参数为插入对象 |
| 后置条件 | 增加一条po数据，返回结果信息 |
| RevenueDataService.getFormPO | 语法 | public RevenuePO getFormPO (String id) |
| 前置条件 | 参数为付款单号 |
| 后置条件 | 查找返回相应的结果 |
| RevenueDataService.delete | 语法 | public OperationMessage delete  (String id) |
| 前置条件 | 同样的po在数据中存在,传入id |
| 后置条件 | 删除po，返回结果信息 |
| RevenueDataService.update | 语法 | public OperationMessage update  (RevenuePO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po，返回结果信息 |
| RevenueDataService.newID | 语法 | public String newID(String unitID) |
| 前置条件 | 传入机构编号 |
| 后置条件 | 返回一个新的收款单ID |
| RevenueDataService. clear | 语法 | public OperationMessage clear () |
| 前置条件 | 无 |
| 后置条件 | 清空RevenuePO数据 |
| RevenueDataService. getAll | 语法 | public List< PaymentPO >  getAll () |
| 前置条件 | 无 |
| 后置条件 | 返回所有收款单单PO数据 |
| RevenueDataService.getByTime | 语法 | **public** List< RevenuePO > getByTime(Calendar start, Calendar end) |
| 前置条件 | 传入起始时间，截止时间 |
| 后置条件 | 返回区间段的收款单对象 |
| RevenueDataService,getByHallID | 语法 | **public** List<RevenuePO> getByHallID(String ID) |
| 前置条件 | 传入营业厅编号 |
| 后置条件 | 返回该营业厅产生的收款单清单 |
| RevenueDataService。getHistory | 语法 | **public** List<RevenuePO> getHistory(String creatorID) |
| 前置条件 | 传入员工ID |
| 后置条件 | 返回该员工创建的该单列表 |

### InitialDataService

1. 模块概述

InitialDataService模块提供有关初期建账的操作

1. 模块内类接口规范

InitialDataService接口规范如下表

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| InitialDataService.getInitialDataPO | 语法 | **public** InitialDataPO getInitialDataPO(String version) |
| 前置条件 | 传入版本号 |
| 后置条件 | 返回对应的期初信息 |
| InitialDataService.requestInitData | 语法 | **public** OperationMessage requestInitData(String staffID) |
| 前置条件 | 传入员工编号（申请人） |
| 后置条件 | 返回请求期初建账的操作结果 |
| InitialDataService.uploadInitialData | 语法 | **public** OperationMessage uploadInitialData(String staffID, InitialDataPO newData) |
| 前置条件 | 传入员工ID，新的期初信息 |
| 后置条件 | 返回载入结果 |
| InitialDataService.aboutInitData | 语法 | **public** OperationMessage abortInitData(String staffID) |
| 前置条件 | 传入员工ID |
| 后置条件 | 如果ID符合期初申请人ID，则终止建账  否则无效 |
| InitialDataService.getLatest\_version | 语法 | **public** String getLatest\_version(String staffID) |
| 前置条件 | 传入员工ID |
| 后置条件 | 返回最新的期初版本号 |

### MemberDataService

1. 模块概述

MenberDataService模块提供有关人员信息的操作

1. 模块内类接口规范

MemberDataService接口规范如下表

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| memberDataService.getStaff | 语法 | public List<StaffVO> getStaff(StaffTypeEnum StaffTypeEnum) |
| 前置条件 | StaffTypeEnum为员工类型的枚举 |
| 后置条件 | 根据StaffTypeEnum获取相应的所有员工信息 |
| memberDataService.getStaffByInstitution | 语法 | **public** List<PO> getStaffByInstitution(String institutionID) |
| 前置条件 | 传入机构ID |
| 后置条件 | 返回该机构下属所有员工信息 |
| memberDataService.getPerson | 语法 | **public** StaffPO getPerson(String ID) |
| 前置条件 | 传入员工ID |
| 后置条件 | 返回该员工信息 |
| memberDataService.modifyStaff | 语法 | public OperationMessage modifyStaff(StaffPO after) |
| 前置条件 | after为修改后的信息 |
| 后置条件 | 若修改成功返回成功，否则失败原因 |
| memberDataService.addStaff | 语法 | public OperationMessage addStaff(StaffPO staff) |
| 前置条件 | Staff为新员工的信息;已经获取所有员工信息 |
| 后置条件 | 若新建成功则返回成功；否则，返回失败原因 |
| memberDataService.dismissStaff | 语法 | public OperationMessage dismissStaff(StaffPO Staff) |
| 前置条件 | Staff为待解雇员工信息；已经获取员工信息列表 |
| 后置条件 | 若删除成功则返回成功；否则返回失败原因 |
| memberDataService.newStaffID | 语法 | public String newStaffID(StaffTypeEnum type,String unitID) |
| 前置条件 | type为新员工的工种,以及所属机构编号 |
| 后置条件 | 返回合法的员工ID |

### OrderDataService

1. 模块概述

OrderDataService模块提供有关人员信息的操作（订单）

1. 模块内类接口规范

OrderDataService接口规范如下表

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderDataService.getFormPO | 语法 | public OrderPO getFormPO(String id) |
| 前置条件 | 参数为快递单号 |
| 后置条件 | 查找返回相应的结果 |
| OrderDataService.getAll | 语法 | Public List<OrderPO> getAll() |
| 前置条件 | 无 |
| 后置条件 | 返回所有订单列表 |
| OrderDataService.insert | 语法 | public OperationMessage insert  (OrderPO po) |
| 前置条件 | 同样的po在数据中不存在，参数为插入对象 |
| 后置条件 | 增加一条po数据，返回结果信息 |
| OrderDataService.delete | 语法 | public OperationMessage delete  (String id) |
| 前置条件 | 同样的po在数据中存在,传入id |
| 后置条件 | 删除po，返回结果信息 |
| OrderDataService.update | 语法 | public OperationMessage update  (OrderPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po，返回结果信息 |
| OrderDataService.newID | 语法 | public String newID() |
| 前置条件 | 无 |
| 后置条件 | 返回一个新的订单号 |
| OrderDataService. clear | 语法 | public void clear () |
| 前置条件 | 无 |
| 后置条件 | 清空OrderPO数据 |
| OrderDataService.addFormID | 语法 | **public** OperationMessage addFormID(String orderID,String formID) |
| 前置条件 | 传入目标订单号，和有关的表单号 |
| 后置条件 | 将表单号加入订单的有关单据历史中，返回操作信息 |
| OrderDataService.setFinish | 语法 | **public** OperationMessage setFinished(String orderID) |
| 前置条件 | 传入订单ID |
| 后置条件 | 将该订单设为完成，返回操作结果 |
| OrderDataService.getHistory | 语法 | **public** List<OrderPO> getHistory(String creatorID) |
| 前置条件 | 传入员工ID |
| 后置条件 | 返回该员工创建的该单列表 |

### ReceiveDataService

1. 模块概述

ReceiveDataService模块提供货物到达的一系列操作（到达单）

1. 模块内类接口规范

ReceiveDataService接口规范如下表

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ReceiveDataService.insert | 语法 | public OperationMessage insert  (Delive ReceivePO rPO po) |
| 前置条件 | 同样的po在数据中不存在，参数为插入对象 |
| 后置条件 | 增加一条po数据，返回结果信息 |
| ReceiveDataService.getFormPO | 语法 | public ReceivePO getFormPO(String id) |
| 前置条件 | 参数为到达单号 |
| 后置条件 | 查找返回相应的结果 |
| ReceiveDataService.delete | 语法 | public OperationMessage delete  (String id) |
| 前置条件 | 同样的po在数据中存在,传入id |
| 后置条件 | 删除po，返回结果信息 |
| ReceiveDataService.update | 语法 | public OperationMessage update  (ReceivePO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po，返回结果信息 |
| ReceiveDataService.getHistory | 语法 | **public** List<ReceivePO> getHistory(String creatorID) |
| 前置条件 | 传入员工ID |
| 后置条件 | 返回该员工创建的所有该单列表 |
| ReceiveDataService.newID | 语法 | public String newID(String unitID) |
| 前置条件 | 传入机构ID |
| 后置条件 | 返回一个新的到达单ID |
| ReceiveDataService. clear | 语法 | public OperationMessage clear () |
| 前置条件 | 无 |
| 后置条件 | 清空ReceivePO数据 |
| ReceiveDataService. getAll | 语法 | public List< DeliverPO >  getAll () |
| 前置条件 | 无 |
| 后置条件 | 返回所有到达单PO数据 |

### StoreDataService

1. 模块概述

StoreDataService模块提供有关货物库存的一系列操作（入库单，出库单）

1. 模块内类接口规范

StoreDataService接口规范如下表

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StoreFormDataService.insertStoreInPO | 语法 | public OperationMessage insertStoreInPO  (StoreInPO po) |
| 前置条件 | 同样的po在数据中不存在，参数为插入对象 |
| 后置条件 | 增加一条po数据，返回结果信息 |
| StoreFormDataService.getStoreInPO | 语法 | public StoreInPO get StoreInPO (String id) |
| 前置条件 | 参数为入库单号 |
| 后置条件 | 查找返回相应的结果 |
| StoreFormDataService.deleteStoreInPO | 语法 | public OperationMessage deleteStoreInPO  (String id) |
| 前置条件 | 同样的po在数据中存在,传入id |
| 后置条件 | 删除po，返回结果信息 |
| StoreFormDataService.updateStoreInPO | 语法 | public OperationMessage updateStoreInPO  ( StoreInPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po，返回结果信息 |
| StoreFormDataService.newStoreInPO | 语法 | public String newStoreInPO (String unitID) |
| 前置条件 | 传入机构编号 |
| 后置条件 | 返回一个新的入库单编号 |
| StoreFormDataService. clearStoreInPO | 语法 | public void clearStoreInPO () |
| 前置条件 | 无 |
| 后置条件 | 清空 StoreInPO 数据 |
| StoreFormDataService.getHistoryIn | 语法 | **public** List< StoreInPO > getHistoryIn(String creatorID) |
| 前置条件 | 传入员工ID |
| 后置条件 | 返回该员工创建的所有该单列表 |
| StoreFormDataService.insertStoreOutPO | 语法 | public OperationMessage insertStoreOutPO  ( StoreOutPO po) |
| 前置条件 | 同样的po在数据中不存在，参数为插入对象 |
| 后置条件 | 增加一条po数据，返回结果信息 |
| StoreFormDataService.getStoreOutPO | 语法 | public StoreOutPO getStoreOutPO (String id) |
| 前置条件 | 参数为出库单号 |
| 后置条件 | 查找返回相应的结果 |
| StoreFormDataService.deleteStoreOutPO | 语法 | public OperationMessage deleteStoreOutPO  (String id) |
| 前置条件 | 同样的po在数据中存在,传入id |
| 后置条件 | 删除po，返回结果信息 |
| StoreFormDataService.updateStoreOutPO | 语法 | public OperationMessage updateStoreOutPO  ( StoreOutPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po，返回结果信息 |
| StoreFormDataService.newStoreOutPO | 语法 | public String new StoreOutPO (String unitID) |
| 前置条件 | 传入机构编号 |
| 后置条件 | 返回一个新的出库单编号 |
| StoreFormDataService. clearStoreOutPO | 语法 | public void clearStoreOutPO () |
| 前置条件 | 无 |
| 后置条件 | 清空StoreOutPO数据 |
| StoreFormDataService.getHistoryOut | 语法 | **public** List< StoreOutPO> getHistoryOut(String creatorID) |
| 前置条件 | 传入员工ID |
| 后置条件 | 返回该员工创建的所有该单列表 |
| StoreFormDataService . getAllStoreInPO | 语法 | public List< StoreInPO >  getAllStoreInPO ( String centerID,) |
| 前置条件 | 传入仓库编号 |
| 后置条件 | 返回所有入库信息 |
| StoreFormDataService . getAllStoreOutPO | 语法 | public List< StoreOutPO >  getAllStoreOutPO ( String centerID,) |
| 前置条件 | 传入仓库编号 |
| 后置条件 | 返回所有出库信息 |
| StoreFormDataService.getInOutInfo | 语法 | **public** List<FormPO> getInOutInfo( String centerID,Calendar start,Calendar end) |
| 前置条件 | 传入仓库编号,起始时间，截至时间 |
| 后置条件 | 返回一定时间内的出入库记录信息 |
| TackDataService.getTack | 语法 | **public** **int** getTack(String centerID) |
| 前置条件 | 传入仓库ID |
| 后置条件 | 返回当前盘点次数 |
| TackDataService.setTack | 语法 | **public** OperationMessage setTack(String centerID,String num) |
| 前置条件 | 传入仓库ID和要设置的盘点次数 |
| 后置条件 | 改变数据库中盘点次数，返回操作信息 |
| StoreModelDataService.setLocation | 语法 | **public** OperationMessage setLocation(String centerID,StoreLocation location) |
| 前置条件 | 传入要修改的位置信息 |
| 后置条件 | 返回修改结果 |
| StoreModelDataService.getModels | 语法 | **public** List<StoreModel> getModels() |
| 前置条件 | 无 |
| 后置条件 | 返回系统中所有的仓库信息 |
| StoreModelDataService.getArea | 语法 | **public** StoreArea getArea(String centerID,StoreAreaCode code) |
| 前置条件 | 传入仓库编号以及区域类型 |
| 后置条件 | 返回该区域信息 |
| StoreModelDataService.getModel | 语法 | **public** StoreModel getModel(String centerID) |
| 前置条件 | 传入仓库编号ID |
| 后置条件 | 获得该仓库的信息 |
| StoreModelDataService.newShelf | 语法 | **public** OperationMessage newShelf(String centerID,StoreAreaCode code, **int** row,**int** shelf) |
| 前置条件 | 传入新货架的位置信息 |
| 后置条件 | 返回新建的操作结果 |
| StoreModelDataService.removeShelf | 语法 | **public** OperationMessage removeShelf(String centerID,StoreAreaCode code, **int** row,**int** shelf) |
| 前置条件 | 传入要移除货架的位置信息 |
| 后置条件 | 返回删除的操作结果 |
| StoreModelDataService.moveShelf | 语法 | **public** OperationMessage moveShelf(String centerID,StoreAreaCode code\_now, **int** row\_now,  **int** shelf\_now, StoreAreaCode code, **int** row, **int** shelf) |
| 前置条件 | 传入货架新老位置的信息 |
| 后置条件 | 返回更改位置的操作结果 |
| StoreModelDataService.getLocation | 语法 | **public** String getLocation(String centerID,StoreAreaCode code, **int** row,  **int** shelf,**int** position) |
| 前置条件 | 传入某一位置信息 |
| 后置条件 | 返回这一位置上的订单号码 |

### TransportDataService

1. 模块概述

TransportDataService模块提供有关货物中转的一系列操作（中转单，装车单）

1. 模块内类接口规范

TransportDataService接口规范如下表

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CenterOutDataService.insert | 语法 | public OperationMessage insert  (CenterOutPO po) |
| 前置条件 | 同样的po在数据中不存在，参数为插入对象 |
| 后置条件 | 增加一条po数据，返回结果信息 |
| CenterOutDataService.getFormPO | 语法 | public TransportPO getFormPO (String id) |
| 前置条件 | 参数为中转单号 |
| 后置条件 | 查找返回相应的结果 |
| CenterOutDataService.delete | 语法 | public OperationMessage delete  (String id) |
| 前置条件 | 同样的po在数据中存在,传入id |
| 后置条件 | 删除po，返回结果信息 |
| CenterOutDataService.update | 语法 | public OperationMessage update  (CenterOutPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po，返回结果信息 |
| CenterOutDataService.newTransID | 语法 | public String newTransID(String unitID) |
| 前置条件 | 传入机构编号 |
| 后置条件 | 返回一个新的中转单编号 |
| CenterOutDataService. clear | 语法 | public void clear () |
| 前置条件 | 无 |
| 后置条件 | 清空CenterOutPO数据 |
| CenterOutDataService.getHistory | 语法 | **public** List<CenterOutPO> getHistory(String creatorID) |
| 前置条件 | 传入员工ID |
| 后置条件 | 返回该员工创建的所有该单列表 |
| CenterOutDataService. getAll | 语法 | public List< CenterOutPO >  getAll () |
| 前置条件 | 无 |
| 后置条件 | 返回所有装车单PO数据 |
| LoadDataService.insert | 语法 | public OperationMessage insert  (LoadPO po) |
| 前置条件 | 同样的po在数据中不存在，参数为插入对象 |
| 后置条件 | 增加一条po数据，返回结果信息 |
| LoadDataService.getFormPO | 语法 | public LoadPO getFormPO (String id) |
| 前置条件 | 参数为装车单号 |
| 后置条件 | 查找返回相应的结果 |
| LoadDataService.delete | 语法 | public OperationMessage delete  (String id) |
| 前置条件 | 同样的po在数据中存在,传入id |
| 后置条件 | 删除po，返回结果信息 |
| LoadDataService.update | 语法 | public OperationMessage update  (LoadPO po) |
| 前置条件 | 同样的po在数据中存在 |
| 后置条件 | 更新po，返回结果信息 |
| LoadDataService.newTransID | 语法 | public String newTransID(String unitID) |
| 前置条件 | 传入机构编号 |
| 后置条件 | 返回一个新的装车单编号 |
| LoadDataService. clear | 语法 | public void clear () |
| 前置条件 | 无 |
| 后置条件 | 清空LoadPO数据 |
| LoadDataService.getHistory | 语法 | **public** List< LoadPO > getHistory(String creatorID) |
| 前置条件 | 传入员工ID |
| 后置条件 | 返回该员工创建的所有该单列表 |
| LoadDataService. getAll | 语法 | public List< LoadPO >  getAll () |
| 前置条件 | 无 |
| 后置条件 | 返回所有装车单PO数据 |