userLogic模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| loginLogic.login | 语法 | public ResultMessage login(String id,String password) |
| 前置条件 | id和password均输入,选择登陆 |
| 后置条件 | 查找是否存在相应的用户，根据输入的id和password返回结果和用户信息 |
| UserManLogic.newUser | 语法 | public Result newUser(UserVO user,String name) |
| 前置条件 | 输入所有用户信息后，确认新建 |
| 后置条件 | 将新用户信息加入数据库，返回加入结果 |
| UserManLogic.deleteUser | 语法 | public Result deleteUser(String id,String name ,int index) |
| 前置条件 | 选择了一个用户，选择删除 |
| 后置条件 | 在数据库中除去相应用户的信息，返回删除结果 |
| UserManLogic.reviseUser | 语法 | public Result reviseUser(UserVO user,intinde,String name) |
| 前置条件 | 修改了用户的信息，确认 |
| 后置条件 | 在数据库中修改相应用户的信息，返回修改结果 |
| UserManLogic.findUser | 语法 | public ResultMessage findUser(String keyword) |
| 前置条件 | 用户选择查看用户列表或关键词检索 |
| 后置条件 | 返回数据库中所有用户的列表或含关键词的用户列表，包含用户的账号、姓名和所属机构 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| UserManDSer.findUser(String keyword) | 得到User数据库的服务的引用 | |
| UserManDSer.addUser(UserPO po) | 在数据库中添加UserPO对象 | |
| UserManDSer.deleteUser(String id) | 在数据库中删除id所指向的对象 | |
| UserManDSer.reviseUser(UserPO po) | 在数据库中修改UserPO中id指向的对象的信息 | |
| LoginDSer.findUser(String id,String password) | 在数据库中查找返回对应id的PO信息 | |

transitInfo模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CustomersLogic.enterBarcodeCustom | 语法 | Public ResultMessage enterBarcodeCustom(String barcode); |
| 前置条件 | barcode符合输入规则 |
| 后置条件 | 查找是否存在相应的订单，据输入的barcode返回相应订单的物流信息和货运状态 |
| CourierLogic.enterBarcodeCourier | 语法 | Public ResultMessage enterBarcodeCourier(String barcode); |
| 前置条件 | barcode符合输入规则 |
| 后置条件 | 查找是否存在相应的订单，根据输入的barcode返回该订单所有信息 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| transitInfoDSer.find(String  barcode) | 根据barcode进行查找订单对象 | |
| transitInfoDSer.findTransit(String  barcode) | 根据barcode查找相应订单的物流信息和货运状态 | |

orderNewLogic模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ArriveOrderLogic.Create | 语法 | public Result Create(ArriveVO order) |
| 前置条件 | 到达单的各项信息已输入并确认 |
| 后置条件 | 将该新增到达单信息上传至数据库，返回上传是否成功 |
| ArriveOrderLogic.geLocalHallsAndAllCenter(String orgId) | 语法 | public ResultMessage geLocalHallsAndAllCenter(String orgId) |
| 前置条件 | 该机构id在数据中存在，用户打开到达单界面 |
| 后置条件 | 在数据库中查找所有城市，并返回城市名称列表 |
| ArriveOrderLogic.getArriveOrders(String org) | 语法 | public ResultMessage getArriveOrders(String org) |
| 前置条件 | 该机构id在数据中存在，用户打开到达单界面 |
| 后置条件 | 在数据库中查找所有，并返回到达单列表 |
| CenterLoadingLogic.Create | 语法 | public Result Create(CenterLoadingVO order) |
| 前置条件 | 中转中心装车单的各项信息已输入并确认 |
| 后置条件 | 将该新增中转中心装车单信息上传至数据库，返回上传是否成功 |
| CenterLoadingLogic.getHalls | 语法 | public ResultMessage getHalls() |
| 前置条件 | 用户输入中转中心装车单界面 |
| 后置条件 | 在数据库中查找所有营业厅，并返回营业厅名称列表 |
| CenterLoadingLogic.getPrice | 语法 | public String getPrice(String org1,String org2,List<String >expressList) |
| 前置条件 | 用户打开中转中心装车单界面 |
| 后置条件 | 在数据库中获取信息并计算运费 |
| CenterLoadingLogic.getNextId | 语法 | public ResultMessage getNextId(String id) |
| 前置条件 | 用户打开中转中心装车单界面 |
| 后置条件 | 在数据库中查找装车单并得到下一个订单号 |
| CenterLoadingLogic.getExpressList | 语法 | public ResultMessage getExpressList(String id) |
| 前置条件 | 用户打开中转中心装车单 |
| 后置条件 | 在数据库中获取快递单列表 |
| ExpressOrderLogic.Create | 语法 | public Result Create(ExpressVO order) |
| 前置条件 | 寄件单的各项信息已输入并确认 |
| 后置条件 | 将该寄件单信息上传至数据库，返回上传是否成功 |
| ExpressOrderLogic.getPrice | 语法 | public String getPrice (String city1,String city2,Express expressKind,Parse pakKind,double weight) |
| 前置条件 | 用户已经输入出发地到达地 |
| 后置条件 | 自动计算运费并返回 |
| ExpressOrderLogic.getTime | 语法 | public int getTime (String city1,String city2) |
| 前置条件 | 快递单已经填写城市 |
| 后置条件 | 自动计算到达时间 |
| HallLoadingOrderLogic.Create | 语法 | public Result Create(HallLoadingVO order) |
| 前置条件 | 营业厅装车单的各项信息已输入并确认 |
| 后置条件 | 将该新增营业厅装车单上传至数据库，返回上传是否成功 |
| HallLoadingOrderLogic.getLocalHallAndAllCenter | 语法 | public ResultMessage getLocalHallAndAllCenter (String orgId) |
| 前置条件 | 用户打开营业厅装车单界面 |
| 后置条件 | 查找数据库中所有到达地，返回到达地名称列表 |
| HallLoadingOrderLogic.getVans | 语法 | public ResultMessage getVan(String org) |
| 前置条件 | 用户打开营业厅装车单界面 |
| 后置条件 | 查找数据库中所有车辆，返回车辆代号列表 |
| HallLoadingOrderLogic.getPrice | 语法 | public String getPrice () |
| 前置条件 | 用户输入出发地到达地 |
| 后置条件 | 查找数据库，计算运费并返回 |
| HallLoadingOrderLogic.getNextId | 语法 | public ResultMessage getNextId (String orgId) |
| 前置条件 | 用户打开营业厅装车单 |
| 后置条件 | 处理数据计算下一个订单号 |
| HallLoadingOrderLogic.getUnloadExpress | 语法 | public ResultMessage getUnloadExpress (String org) |
| 前置条件 | 用户打开营业厅装车单界面 |
| 后置条件 | 得到所有未装载订单 |
| HallLoadingOrderLogic.isLegal | 语法 | public Boolean isLegal(String id) |
| 前置条件 | 用户输入装车单 |
| 后置条件 | 检查订单的非法性 |
| IncomeOrderLogic.Create | 语法 | public Result Create(IncomeVO order) |
| 前置条件 | 收款单的各项信息已输入并确认 |
| 后置条件 | 将该新增收款单信息上传至数据库，返回上传是否成功 |
| IncomeOrderLogic.getCouriers | 语法 | public ResultMessage getCouriers(String id) |
| 前置条件 | 该id对应的营业厅在数据库中存在，打开收款单界面 |
| 后置条件 | 根据id在数据库中查找对应营业厅并返回该营业厅快递员姓名列表 |
| IncomeOrderLogic.getAccount | 语法 | public ResultMessage getAccount () |
| 前置条件 | 打开收款单输入界面 |
| 后置条件 | 获取并返回账户列表 |
| IncomeOrderLogic.getNextId | 语法 | public ResultMessage getNextId (String orgid) |
| 前置条件 | 该机构id在数据库中存在，打开收款单界面 |
| 后置条件 | 获取并返回下一个单据编号 |
| IncomeOrderLogic.expressAvailable | 语法 | public ResultMessage expressAvailable (String orgid) |
| 前置条件 | 已填写快递单号 |
| 后置条件 | 快递单有效性检测 |
| PaymentLogic.Create | 语法 | public Result Create(PaymentVO order) |
| 前置条件 | 付款单的各项信息已输入并确认 |
| 后置条件 | 将该新增付款单信息上传至数据库，返回上传是否成功 |
| PaymentLogic.getAccount | 语法 | public ResultMessage getAccount () |
| 前置条件 | 开启付款单填写 |
| 后置条件 | 在数据库中查找并返回账户 |
| PaymentLogic.getNextId | 语法 | public ResultMessage getNextId () |
| 前置条件 | 开启一个新的付款单填写界面 |
| 后置条件 | 在数据库中获取信息并返回单据编号 |
| ReceiveOrderLogic.Create | 语法 | public Result Create(ReceiveVO order,String org,String name) |
| 前置条件 | 收件单的各项信息已输入并确认 |
| 后置条件 | 将该新增收件信息上传至数据库，返回上传是否成功 |
| ReceiveOrderLogic.getExpress | 语法 | public ResultMessage getExpress(String id) |
| 前置条件 | 该id对应的订单信息在数据库中存在 |
| 后置条件 | 根据id在数据库中查找对应的订单信息并返回 |
| RecipientOrderLogic.Create | 语法 | public Result Create(RecipientVO order) |
| 前置条件 | 接收单的各项信息已输入并确认 |
| 后置条件 | 将该新增接收单信息上传至数据库，返回上传是否成功 |
| RecipientOrderLogic.getLocalHallAndAllCenter | 语法 | public ResultMessage getLocalHallAndAllCenter (String id) |
| 前置条件 | 打开营业厅接收单 |
| 后置条件 | 从数据层获取所有的营业厅和中转中心 |
| RecipientOrderLogic.isLoadValid | 语法 | public boolean isLoadValid (String id) |
| 前置条件 | 输入装车单编号 |
| 后置条件 | 检测装车单编号是否存在并返回结果 |
| RecipientOrderLogic.expressList | 语法 | public List<String> expressList (String id) |
| 前置条件 | 打开营业厅接收单 |
| 后置条件 | 从数据库中获取中转单编号列表并返回 |
| RecipientOrderLogic.getNextId | 语法 | public ResultMessage getNextId (String orgId) |
| 前置条件 | 打开营业厅接收单 |
| 后置条件 | 从数据层获取信息得到接收单编号并返回 |
| RecipientOrderLogic.getComingLoadingOrder | 语法 | public ResultMessage getComingLoadingOrder (String org) |
| 前置条件 | 打开接收单创建界面 |
| 后置条件 | 从数据层获取信息并返回正在到来的装车单 |
| SendOrderLogic.Create | 语法 | public Result Create(SendOrderVO order) |
| 前置条件 | 派件单的各项信息已输入并确认 |
| 后置条件 | 将该新增派件信息上传至数据库，返回上传是否成功 |
| SendOrderLogic.getCouriers | 语法 | public ResultMessage getCouriers (String id) |
| 前置条件 | 打开派件单界面 |
| 后置条件 | 获取该机构的快递员并返回 |
| SendOrderLogic.getNextId | 语法 | public ResultMessage getNextId (String orgId) |
| 前置条件 | 打开派件单界面 |
| 后置条件 | 获取派件单编号并返回 |
| SendOrderLogic.getGoingExpress | 语法 | public ResultMessage getGoingExpress (String orgId) |
| 前置条件 | 打开派件单界面 |
| 后置条件 | 获取所有正在到来的中转单列表 |
| StockInOrderLogic.Create | 语法 | public Result Create(StockInOrderVO order) |
| 前置条件 | 入库单的各项信息已输入并确认 |
| 后置条件 | 将该新增入库信息上传至数据库，返回上传是否成功 |
| StockInOrderLogic.getOrgs | 语法 | public ResultMessage getOrgs () |
| 前置条件 | 打开入库单界面 |
| 后置条件 | 获取机构列表并返回 |
| StockInOrderLogic.getShelves | 语法 | public ResultMessage getShelves (String orgId) |
| 前置条件 | 打开入库单界面 |
| 后置条件 | 获取空余的架列表并返回 |
| StockInOrderLogic.isExpressValid | 语法 | public boolean isExpressValid (String id) |
| 前置条件 | 输入中转单编号 |
| 后置条件 | 检测并返回中转单编号是否合法 |
| StockInOrderLogic.getNextId | 语法 | public ResultMessage getNextId (String id) |
| 前置条件 | 打开入库单界面 |
| 后置条件 | 获取并返回订单编号 |
| StockInOrderLogic.getComingExpress | 语法 | public ResultMessage getComingExpress (String id) |
| 前置条件 | 打开入库单界面 |
| 后置条件 | 从数据库中获取正在到来的中转单 |
| StockInOrderLogic.getRow | 语法 | public Integer[] getRow (String shelf) |
| 前置条件 | 打开入库单界面 |
| 后置条件 | 从数据层获取空闲架信息并返回 |
| StockInOrderLogic.getPlace | 语法 | public Integer[] getPlace (String id,int row) |
| 前置条件 | 打开入库单界面 |
| 后置条件 | 从数据层获取空余位位号列表 |
| StockOutOrderLogic.Create | 语法 | public Result Create(StockoutOrderVO order) |
| 前置条件 | 出库单信息已填写并确认 |
| 后置条件 | 将该新增出库信息上传至数据库，返回上传是否成功 |
| StockOutOrderLogic.getOrgs | 语法 | public Result getOrgs () |
| 前置条件 | 打开出库单界面 |
| 后置条件 | 获取所有机构列表 |
| StockOutOrderLogic.isExpressValid | 语法 | public Result isExpressValid (String id) |
| 前置条件 | 填写快递单 |
| 后置条件 | 检查快递单是否有效并返回 |
| StockOutOrderLogic.isTransitValid | 语法 | public Result isTransitValid (String id) |
| 前置条件 | 填写中转单 |
| 后置条件 | 检查中转单是否有效并返回 |
| StockOutOrderLogic.getNextId | 语法 | public ResultMessage getNextId (String orgId) |
| 前置条件 | 打开出库单界面 |
| 后置条件 | 从数据层获取并返回订单编号 |
| StockOutOrderLogic.isConveyValid | 语法 | public Result isConveyValid (String id) |
| 前置条件 | 填写装车单 |
| 后置条件 | 检查并返回装车单的有效性 |
| StockOutOrderLogic.getGoingOrders | 语法 | public Result getGoingOrders (String org,String orgId) |
| 前置条件 | 打开出库单界面 |
| 后置条件 | 从数据层获取并返回正在到来的订单 |
| StockOutOrderLogic.getGoingExpress | 语法 | public Result getGoingExpress (String orderId,String orgId) |
| 前置条件 | 从数据层获取并返回正在到来的中转单 |
| 后置条件 | 将该新增出库信息上传至数据库，返回上传是否成功 |
| TransitOrderLogic.Create | 语法 | public Result Create(TransitOrderVO order) |
| 前置条件 | 中转单的各项信息已输入并确认 |
| 后置条件 | 将该新增中转单信息上传至数据库，返回上传是否成功 |
| TransitOrderLogic.getCenter | 语法 | public ResultMessage getCenters () |
| 前置条件 | 打开中转单 |
| 后置条件 | 获取中转中心列表并返回 |
| TransitOrderLogic.isExpressValid | 语法 | public int isExpressValid (List<String >expressList) |
| 前置条件 | 输入快递单编号 |
| 后置条件 | 检查并返回快递单是否有效 |
| TransitOrderLogic.getPrice | 语法 | public ResultMessage getPrice(String org1,String org2,int tran,List<String>expressList) |
| 前置条件 | 已经输入出发地到达地 |
| 后置条件 | 计算并返回运费 |
| TransitOrderLogic.getNextId | 语法 | public ResultMessage getNextId (TransitOrderVO order) |
| 前置条件 | 打开中转单界面 |
| 后置条件 | 得到并返回订单编号 |
| TransitOrderLogic.getGoingExpress | 语法 | public ResultMessage getGoingExpress (TransitOrderVO order) |
| 前置条件 | 打开中转单界面 |
| 后置条件 | 得到并返回正在到来的快递单列表 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| DatabaseFactory.getOrderData | 得到Order数据库的服务的引用 | |
| orderNewDSer. \*OrderDSer .create(\*PO order ) | 在数据库中添加对应订单\*PO对象 | |
| orderNewDSer. getOderSize(int partrId) | 根据区号在数据库中获取并返回当天单据数量 | |
| ReceiveOrderDSer. getExpress(String id) | 在数据库中获取并返回快递单PO | |
| SendOrderDSer.getCouriers() | 在数据库中获取并返回快递员信息 | |
| StockInOrderDSer.getShelves() | 在数据库中获取并返回架信息列表 | |
| StockInOrderDSer.addGood(GoodPO good) | 在数据库中添加货物PO的储存信息 | |
| StockOutOrderDSer.removeGood(String id) | 在数据库中删除货物PO并返回结果 | |
| UpdateTransitOrderDSer.addInfo(TransitInfoPO info) | 创建快递的物流信息并返回结果 | |
| UpdateTransitOrderDSer.update(TransitInfoPO info) | 在该快递物流信息中添加新的物流轨迹 | |
| UpdateTranStateDSer. updateTranState(UpdateTranState order) | 更新单据的物流状态并返回结果 | |
| UpdateTranStateDSer.update (LocationFind order) | 根据当前机构ID从数据库中获得所有订单列表并返回 | |

workOrgManLogic模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| OrgManLogic.addOrg | 语法 | | public Result add(OrgVO org,String name); |
| 前置条件 | | 用户输入了新机构的信息 |
| 后置条件 | | 将新增的机构信息加入数据库，返回加入结果 |
| OrgManLogic.deleteOrg | 语法 | | public Result deleteOrg(String id,String name) |
| 前置条件 | | 用户选择一个机构进行删除 |
| 后置条件 | | 根据机构名在数据库中删除该账户信息，返回删除结果 |
| OrgManLogic.reviseOrg | 语法 | | public Result reviseOrg(OrgVO org，String name) |
| 前置条件 | | 修改了机构信息并确认 |
| 后置条件 | | 更新数据库中该机构的信息，返回更新结果 |
| OrgManLogic.searchOrg | 语法 | | public ResultMessage searchOrg(String keywords) |
| 前置条件 | | 用户打开机构管理或检索机构并输入关键字 |
| 后置条件 | | 返回所有机构或根据该关键字查找符合条件的机构并返回 |
| OrgManLogic.getCities | 语法 | | public ResultMessage getCities () |
| 前置条件 | | 用户打开机构管理 |
| 后置条件 | | 返回城市间信息 |
| OrgManLogic.getId | 语法 | | public String getId (CityVO city,String kind) |
| 前置条件 | | 用户打开机构管理 |
| 后置条件 | | 从数据库中获取机构编号 |
| WorkManLogic.addWork | 语法 | | public Result addWork(WorkVO work); |
| 前置条件 | | 用户输入了新人员信息 |
| 后置条件 | | 将新增的人员信息加入数据库，返回加入结果 |
| WorkManLogic.deleteWork | 语法 | | public Result deleteWork(String id) |
| 前置条件 | | 用户选择一个人员进行删除 |
| 后置条件 | | 根据人员名在数据库中删除该人员信息，返回删除结果 |
| WorkManLogic.reviseWork | 语法 | | public Result reviseWork(WorkVO work) |
| 前置条件 | | 修改了人员信息并确认 |
| 后置条件 | | 更新数据库中该人员的信息，返回更新结果 |
| WorkManLogic.searchWork | 语法 | | public ResultMessage searchWork(String keywords) |
| 前置条件 | | 用户打开人员管理或输入关键字检索 |
| 后置条件 | | 返回全部人员列表或根据该关键字查找符合条件的人员并返回 |
| DriverManLogic.addDriver | 语法 | | public Result addDriver(DriverVO driver，String name) |
| 前置条件 | | 用户选择添加司机信息 |
| 后置条件 | | 在数据库中增加该司机信息，并返回添加结果 |
| DriverManLogic.deleteDriver | 语法 | | public Result deleteDriver(String id,String name) |
| 前置条件 | | 该id对应的司机信息在数据库中存在 |
| 后置条件 | | 根据id在数据库中删除该司机信息，并返回删除结果 |
| DriverManLogic.reviseDriver | 语法 | | public Result reviseDriver(DriverVO driver,String name) |
| 前置条件 | | 该司机在数据库中存在 |
| 后置条件 | | 在数据库中更新该司机信息，并返回更新结果 |
| DriverManLogic.SearchDriver | 语法 | | public ResultMessage SearchDriver (String hallid) |
| 前置条件 | | 该id对应的司机信息在数据库中存在 |
| 后置条件 | | 根据id在数据库中查找该司机信息并返回 |
| DriverManLogic.getNextDriver | 语法 | | public ResultMessage getNextDriver (String hallid) |
| 前置条件 | | 新建司机信息 |
| 后置条件 | | 在数据库中获得并返回下一个司机编号 |
| VanManLogic.addVan | 语法 | | public Result addVan (VanVO van) |
| 前置条件 | | 用户选择添加车辆信息 |
| 后置条件 | | 在数据库中增加该车辆信息，并返回添加结果 |
| VanManLogic.deleteVan | 语法 | | public Result deleteVan (String id) |
| 前置条件 | | 该id对应的车辆信息在数据库中存在 |
| 后置条件 | | 根据id在数据库中删除该车辆信息，并返回删除结果 |
| VanManLogic.reviseVan | 语法 | | public Result reviseVan (VanVO van) |
| 前置条件 | | 该车辆在数据库中存在 |
| 后置条件 | | 在数据库中更新该车辆信息，并返回更新结果 |
| VanManLogic.getNextId | 语法 | | public ResultMessage getNextId (String id) |
| 前置条件 | | 新增车辆信息 |
| 后置条件 | | 从数据库中获取并返回车辆id |
| VanManLogic.searchVan | 语法 | | public ResultMessage searchVan(String keywords) |
| 前置条件 | | 用户打开车辆管理界面或输入关键字检索 |
| 后置条件 | | 返回全部车辆列表或根据该关键字查找符合条件的车辆并返回车辆信息列表 |
| WageManLogic.setWage | 语法 | | public Result setWage (List<WageVO> wageList,int[]index,String name) |
| 前置条件 | | 用户选择设置薪水 |
| 后置条件 | | 在数据库中更新该薪水信息，并返回更新结果 |
| WageManLogic. searchWorks | 语法 | | public ResultMessage searchWorks (String keyword) |
| 前置条件 | | 逻辑层需要搜索相关人员信息 |
| 后置条件 | | 在数据库找到并根据关键字返回人员信息 |
| 需要的服务（需接口） | | | |
| 服务名 | | 服务 | |
| wageManDSer.searchWork(String keyword) | | 从数据库中找到含有该关键字的人员，返回数据 | |
| workManDSer.addWork(WorkPO po) | | 在数据库中新增一个人员的数据 | |
| workManDSer.deleteWork(String id) | | 在数据库中删除一个人员的数据 | |
| workManDSer.reviseWork(WorkPO po) | | 在数据库中修改一个人员的数据 | |
| workManDSer.findWork(String keyword) | | 从数据库中找到含有该关键字的人员，返回数据 | |
| wagekManDSer.updateWork(WorkPOwork) | | 在数据库中更新人员的薪资信息 | |
| DatabaseFactory.getWorkbase() | | 得到Work数据库的服务的引用 | |
| OrgManDSer.findOrg(String keywords) | | 在数据库中找到含有该关键字的机构，返回数据 | |
| OrgManDSer.addOrg(OrgPO po) | | 在数据库中新增一个机构的信息 | |
| OrgManDSer.deleteOrg(String id) | | 在数据库中删除一个机构的信息 | |
| OrgManDSer.reviseOrg(OrgPO po) | | 在数据库中修改一个机构的信息 | |
| DatabaseFactory.getOrgbase | | 得到org数据库的服务的引用 | |
| DriverManDSer.addDriver(DriverPO driver) | | 在数据库中新增一个司机的数据 | |
| DriverManDSer.deleteDriver (String id) | | 在数据库中删除一个司机的数据 | |
| DriverManDSer.reviseDriver(DriverPO driver) | | 在数据库中修改一个司机的数据 | |
| DriverManDSer.findDriver (String keyword) | | 在数据库中找到含有该关键字的司机，返回数据 | |
| DatabaseFactory.getDriverbase() | | 得到Driver数据库的服务的引用 | |
| VanManDSer.findVan(String keywords) | | 从数据库中找到含有该关键字的车辆，返回数据 | |
| VanManDSer.addVan (VanPO van) | | 在数据库中新增一个车辆的数据 | |
| VanManDSer.deleteVan (String id) | | 在数据库中删除一个车辆的数据 | |
| VanManDSer.reviseVan(VanPO van) | | 在数据库中修改一个车辆的数据 | |
| DatabaseFactory.getVanbase() | | 得到Van数据库的服务的引用 | |

cityLogic模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CityLogic.addCity | 语法 | public Result addCity(CityVO city); |
| 前置条件 | 增加一个城市 |
| 后置条件 | 将该城市信息加入数据库，返回添加结果 |
| CityLogic.reviseCity | 语法 | public Result reviseCity(CityVO city); |
| 前置条件 | 修改了一个城市信息 |
| 后置条件 | 更新数据库已有信息，返回修改结果 |
| CityLogic.reviseCities | 语法 | public Result reviseCities(CitiesVO city,int index,String name); |
| 前置条件 | 总经理已经选定了两个城市 |
| 后置条件 | 将两个城市间的距离和价格信息加入数据库或更新数据库中已有的距离与价格信息 |
| CityLogic.deleteCitiy | 语法 | public Result reviseCities(CitiesVO city,String name); |
| 前置条件 | 总经理已经选定了城市 |
| 后置条件 | 从数据库中删除城市信息及与城市有关的信息 |
| CityLogic.citiesList | 语法 | public Result citiesList (); |
| 前置条件 | 打开城市管理界面 |
| 后置条件 | 从数据库中获取并返回城市间信息列表 |
| City.cityList | 语法 | public ResultMessage cityList(); |
| 前置条件 | 需要获取城市对象 |
| 后置条件 | 查找数据库中所有城市并返回城市对象列表 |
| ConstantLogic.setConstant | 语法 | public Result setConstant (ConstantVO city,String opname); |
| 前置条件 | 选择确定设置常量 |
| 后置条件 | 更新数据库中已有常量信息 |
| ConstantLogic.getConstant | 语法 | public ResultMessage getConstant (); |
| 前置条件 | 选择确定设置常量 |
| 后置条件 | 从数据库中获取常量信息并返回 |
| LocLogic.getCities | 语法 | public ResultMessage getCities () |
| 前置条件 | 需要获取城市间信息 |
| 后置条件 | 从数据库中获得城市信息并返回 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| cityDSer.addCity(cityPO po) | 新增城市对象 | |
| cityDSer.reviseCity(CityPO city) | 更新城市对象信息 | |
| cityDSer.deleteCity(String cityname) | 在数据库中删除该city | |
| cityDSer.addCities(CitiesPO cities) | 新增城市间信息对象 | |
| CityDSer.deleteCities(CitiesPO cities) | 在数据库中删除该citiesPO信息 | |
| cityDSer.reviseCities(CityPO cities) | 更新城市之间距离和价格 | |
| CityDSer.getCitiesInfo () | 返回城市间信息列表 | |
| CityDSer.isCityUsed(String name,String id) | 检查城市间信息是否被使用并返回结果 | |
| ConstantDSer.getConstant | 从数据库中获取并返回常量信息 | |
| ConstantDSer.setConstant(ConstantPO constant) | 在数据库中更新常量信息 | |

accountLogic模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CountLogic.newCount | 语法 | public Boolean newCount(String name); |
| 前置条件 | 用户新建一套账 |
| 后置条件 | 将新增的一套账加入数据库，返回加入结果 |
| CountLogic.deleteCount | 语法 | public ResultMessage deleteCount (int index,String name); |
| 前置条件 | 用户选择删除期初信息 |
| 后置条件 | 在数据库中删除期初信息并返回结果 |
| CountLogic.getCount | 语法 | public ResultMessage getCount (); |
| 前置条件 | 用户打开期初信息查看界面 |
| 后置条件 | 返回期初信息 |
| AccountLogic.addAccount | 语法 | public Result addAccount(AccountVO account，String opname); |
| 前置条件 | 用户输入了新账户信息 |
| 后置条件 | 将新增的账户信息加入数据库，返回加入结果 |
| AccountLogic.deleteAccount | 语法 | public Result deleteAccount(int index,String opname) |
| 前置条件 | 用户选择一个账户进行删除 |
| 后置条件 | 根据账户名在数据库中删除该账户信息，返回删除结果 |
| AccountLogic.reviseAccount | 语法 | public Result reviseAccount(AccountVO account,int index,String opname) |
| 前置条件 | 修改了账户信息并确认 |
| 后置条件 | 更新数据库中该账户的信息，返回更新结果 |
| AccountLogic.searchAccount | 语法 | public ResultMessage searchAccount(String keywords) |
| 前置条件 | 用户打开账户管理界面或输入关键字检索 |
| 后置条件 | 返回全部账户列表并根据该关键字查找符合条件的账户并返回 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| accountDSer.findAccount(String keywords) | 在数据库中查找含有关键词的账户，返回数据 | |
| accountDSer.addAccount(AccountPO  po) | 在数据库中新增一个账户 | |
| accountDSer.deleteAccount(String id) | 在数据库中删除一个账户 | |
| accountDSer.reviseAccount(AccountPO po) | 在数据库中修改一个账户的信息 | |
| DatabaseFactory.getAccountbase() | 得到account数据库的服务的引用 | |
| CountDSer.addCount(countPO po) | 在数据库中新增一套账 | |
| CountDSer.getCount() | 根据id在数据库中查找期初信息，返回数据 | |
| CountDSer.deleteCount(int id) | 在数据库中删除id对应的账的信息 | |

stockLogic模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockWarningLogic.setWarning | 语法 | Public Result setWarning (ArrayList <Integer> warnings, String id ,String name) |
| 前置条件 | 用户输入警戒线比例和区并提交 |
| 后置条件 | 修改数据库中的警戒线比例，并返回修改是否成功 |
| StockWarningLogic.getWarning | 语法 | public ResultMessage getWarning(String id) |
| 前置条件 | 用户选择了库存报警 |
| 后置条件 | 在数据库中获得各区警戒线比例，返回值 |
| StockCheckWarnLogic.checkWarning | 语法 | public ResultMessage checkWarning(String id) |
| 前置条件 | 用户新建了入库单 |
| 后置条件 | 对比数据库中库存数/库存容量和警戒线比例，返回结果 |
| StockCheckNowLogic.checkNow | 语法 | public ResultMessage checkNow(String id) |
| 前置条件 | 用户选择了库存盘点 |
| 后置条件 | 从数据库中获取当前库存情况，返回结果 |
| StockCheckNowLogic.outputExcel | 语法 | public Result outputExcel(String location,String name) |
| 前置条件 | 用户选择导出库存情况和导出位置 |
| 后置条件 | 将生成的Excel导出到相应位置，返回导出是否成功 |
| StockDivideLogic.shelfList | 语法 | public ResultMessage shelfList(String id) |
| 前置条件 | 用户选择了库存分区 |
| 后置条件 | 读取数据库中的架信息，返回架信息 |
| StockDivideLogic.addShelf | 语法 | public Result addShelf(ShelfVO shelf，String name, String orgId) |
| 前置条件 | 用户输入了新建的架的所有信息，确认新建 |
| 后置条件 | 在数据库中添加新建的架的信息，返回是否新建成功 |
| StockDivideLogic.deleteShelf | 语法 | public Result deleteShelf(String id,String name,String orgId) |
| 前置条件 | 用户选择了一个架，选择删除 |
| 后置条件 | 在数据库中去除相应架的信息，返回是否删除成功 |
| StockDivideLogic.reviseShelf | 语法 | public Result reviseShelf(ShelfVO shelf) |
| 前置条件 | 用户修改了架信息，并确认 |
| 后置条件 | 在数据库中修改相应架的信息，返回是否修改成功 |
| StockDivideLogic.searchShelf | 语法 | public ResultMessage searchShelf(String id，String keyword ) |
| 前置条件 | 用户输入了关键词，选择搜索 |
| 后置条件 | 在数据库中找到含有关键词的架的信息，返回信息 |
| StockDivideLogic.getNextId | 语法 | public String getNextId (String orgId) |
| 前置条件 | 用户打开库存分区，获取空闲架的列表并返回 |
| 后置条件 | 在数据库中找到含有关键词的架的信息，返回信息 |
| StockCheckLogic.orderStock | 语法 | public ResultMessage checkStockIn(Calendar start, Calendar end,String id) |
| 前置条件 | 用户输入了开始和结束时间 |
| 后置条件 | 在数据库中按照开始时间和结束时间，查找所有出入库信息并按时间排序后返回 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| DatabaseFactory.getStockData() | 得到Stock数据库的服务的引用 | |
| StockWarningDSer.getWarning (String id) | 从数据库中获得ID相应仓库的警戒线比例 | |
| StockWarningDSer.setWarning(int w,Part part,String id) | 在数据库中设置ID相应仓库和相应区的警戒线比例 | |
| StockCheckDSer.getIn(Calendar start, Calendar end,String id) | 在数据库中获得ID相应仓库开始时间和结束时间之间的入库单 | |
| StockCheckDSer.getOut (Calendar start, Calendar end, String id) | 在数据库中获得ID相应仓库开始时间和结束时间之间的出库单 | |
| StockCheckNowDSer.getStock(String id) | 在数据库中获得ID相应仓库当前库存的情况 | |
| StockDivideDSer.getShelves(String id) | 在数据库中获得ID相应仓库当前架的情况 | |
| StockDivideDSer.addShelf(ShelfPO shelf) | 在数据库中ID相应仓库添加一个架的信息 | |
| StockDivideDSer.deleteShelf(String id) | 在数据库中ID相应仓库删除一个架的信息 | |
| StockDivideDSer.reviseShelf(ShelfPO po) | 在数据库中ID相应仓库修改一个架的信息 | |

checkLogic模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| DiaryLogic.seeDiary | 语法 | public ResultMessage seeDiary(Calendar start, Calendar end) |
| 前置条件 | 输入的日期符合输入规则 |
| 后置条件 | 根据输入的开始和结束日期，返回相应的日志记录 |
| BusiCircumLogic.getBusiCircum | 语法 | public ResultMessage getBusiCircum (Calendar start, Calendar end) |
| 前置条件 | 输入的日期符合输入规则 |
| 后置条件 | 根据输入的开始和结束日期，返回由入款单和收款单信息组成的经营情况表 |
| BusiCircumLogic.outputExcel | 语法 | public boolean outputExcel(String name, String location) |
| 前置条件 | 用户选择地址后确认导出经营情况表 |
| 后置条件 | 导出经营情况表到相应地址 |
| CostBeneLogic.getCostBene | 语法 | public ResultMessage getCostBene() |
| 前置条件 | 用户开启成本收益表查看 |
| 后置条件 | 从数据层获取并返回结果 |
| IncomeListLogic.seeIncomeList | 语法 | public ResultMessage seeIncomeList(Calendar date, String id) |
| 前置条件 | 输入了日期，选择了营业厅确认获取收款记录 |
| 后置条件 | 根据输入的日期，返回当日的收款单列表 |
| IncomeListLogic.total | 语法 | public ResultMessage total(ArrayList<IncomeVO> income) |
| 前置条件 | 用户选择合计收款 |
| 后置条件 | 返回合计的收款总额 |
| IncomeListLogic.getHall | 语法 | public ResultMessage getHall() |
| 前置条件 | 用户选择查看经营情况表 |
| 后置条件 | 返回所有营业厅的名称 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| DiaryDSer.seeDiary(Calendar start, Calendar end) | 根据开始和结束日期从数据库中获得日志记录，返回数据 | |
| IncomeListDSer.getCollectRec (Calendar date, String id) | 根据给的日期和营业厅编号从数据库中获得当日该营业厅收款记录，返回数据 | |
| BusiCircumDSer.findIncome(Calendar start，Calendar end) | 根据开始结束时段从数据库中获取并返回所有的收款记录 | |
| BusiCircumDSer.findPayment(Calendar start，Calendar end) | 根据开始结束时段从数据库中获取并返回所有的付款记录 | |
| CostBeneDSer.getIncomes | 从数据库中获取并返回所有的收款记录 | |
| CostBeneDSer.getPayments | 从数据库中获取并返回所有的付款记录 | |
| DatabaseFactory.getDiaryDatabase | 得到diary数据库的服务的引用 | |
| DatabaseFactory.getRecordDatabase | 得到record数据库的服务的引用 | |

orderApprove模块接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| orderApproveLogic.askExamine | 语法 | public ResultMessage askExamine() |
| 前置条件 | 用户请求审批单据 |
| 后置条件 | 返回待审批单据列表 |
| orderApproveSer.examine | 语法 | public Result examine(boolean approve, ArrayList <Integer> index,String name) |
| 前置条件 | 用户决定单据是否通过审批 |
| 后置条件 | 返回审批结果 |
| orderApproveLogic.chooseOrder | 语法 | public ResultMessage chooseOrder(String id, Order kind) |
| 前置条件 | 用户选择一个订单，查看详情 |
| 后置条件 | 系统返回订单详情 |
| orderResubmitLogic.getUnpassed | 语法 | public ResultMessage getUnpassed(String id) |
| 前置条件 | 用户登录后 |
| 后置条件 | 返回未通过单据的数量 |
| orderResubmitLogic.getOrdersDisplay | 语法 | public ResultMessage getOrdersDisplay () |
| 前置条件 | 用户选择查看未通过单据 |
| 后置条件 | 返回未通过单据列表 |
| orderResubmitLogic.getOrder | 语法 | public ResultMessage getOrder (int index) |
| 前置条件 | 用户选择一个订单，查看详情 |
| 后置条件 | 通过传入参数得到列表中单据在表中位置，返回单据的vo |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| orderApproveDSer.update(Boolean isPassed, ArrayList<String>id, Order kind) | 更新订单审批状态 | |
| orderApproveDSer.getOrder(String id,Order kind) | 从数据库中获取相应单据的信息，返回。 | |
| OrderApproveDSer.getUser(String userId) | 根据用户id从数据库获得并返回相应的用户信息 | |
| orderApproveDSer.getExamine(String id) | 从数据库中获取相应单据的审批信息，返回。 | |
| OrderResubmitDSer.getOrders(String userId) | 根据用户ID从数据库获得并返回相应的单据列表 | |
| DatabaseFactory.getOrderDatabase | 得到orders数据库的服务的引用 | |