5.4 数据层的分解

5.4.1 数据层模块的职责

数据层模块的职责如表所示。

数据层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| UserDataSer | 持久化数据库的接口，提供用户数据增删改查等服务 |
| UserDataSerImpl | 基于Mysql数据库的持久化数据库的接口，实现UserDataSer的服务 |
| TransitDataSer | 持久化数据库的接口，提供更新物流信息等服务 |
| TransitDataSerImpl | 基于Mysql数据库的持久化数据库的接口，实现TransitDataSerImpl的服务 |
| OrderNewDataSer | 持久化数据库的接口，提供在数据库新建单据等服务 |
| OrderNewDataSerImpl | 基于Mysql数据库的持久化数据库接口，实现OrderNewDataSer的服务 |
| OrderApproveDataSer | 持久化数据库的接口，提供导入单据信息和保持单据审批状态等服务 |
| OrderApproveDataSerImpl | 基于Mysql数据库的持久化数据接口，实现OrderApproveDataSer的服务 |
| AccountDataSer | 持久化数据库接口，提供账户数据的增删改查和期初建账服务 |
| AccountDataSerImpl | 基于Mysql数据库的持久化数据接口，实现AccountDataSer的服务 |
| CityDataSer | 持久化数据库接口，提供城市数据的增删改查服务 |
| CityDataSerImpl | 基于Mysql数据库的持久化数据接口，实现CityDataSer的服务 |
| WorkOrgManSer | 持久化数据库接口，提供人员机构数据的增删改查服务 |
| WorkOrgManSerImpl | 基于Mysql数据库的持久化数据接口，实现WorkOrgManSer的服务 |
| StockDataSer | 持久化数据库接口，提供库存数据的增删改查服务 |
| StockDataSerImpl | 基于Mysql数据库的持久化数据接口，实现StockDataSer的服务 |
| CheckDataSer | 持久化数据库接口，提供统计报表和日志记录的载入服务 |
| CheckDataSerImpl | 基于Mysql数据库的持久化数据接口，实现CheckDataSer的服务 |

5.4.2 数据层模块的接口规范

数据层模块的接口如表所示。

数据层模块的接口规范

UserDataSer的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| UserDataSer.findUser | 语法 | public UserPO find(String id) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的UserPO结果 |
| UserDataSer.addUser | 语法 | public boolean addUser(UserPO user) throws RemoteException |
| 前置条件 | 同样ID的po在数据库中不存在 |
| 后置条件 | 在数据库中增加一个PO记录 |
| UserDataSer.deleteUser | 语法 | public boolean deleteUser(String id) throws RemoteException |
| 前置条件 | 数据库中存在同样ID的用户 |
| 后置条件 | 在数据库中删除一个PO记录 |
| UserDataSer.reviseUser | 语法 | public boolean reviseUser(UserPO user) throws RemoteException |
| 前置条件 | 在数据库中存在同样ID的用户 |
| 后置条件 | 在数据库中修改该用户的数据 |
| UserDataSer.list | 语法 | public ArrayList<UserPO> list() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回数据库中用户数据的列表 |

TransitInfoDataSer的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| TransitInfoDataSer.find | 语法 | public orderPO find(String barcode) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按barcode进行查找返回相应的orderPO结果 |
| TransitInfoDataSer.findTransit | 语法 | public ArrayList<String> findTransit(String barcode) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按barcode进行查找返回相应订单的物流轨迹和货运信息 |

CityDataSer的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CityDataSer.add | 语法 | public boolean add(CityPO city) throws RemoteException |
| 前置条件 | 该city在数据库中不存在 |
| 后置条件 | 在数据库中增加一个city记录 |
| CityDataSer.revise | 语法 | Public Boolean revise(CityPO city) throws RemoteException |
| 前置条件 | 该city在数据库中已存在 |
| 后置条件 | 在数据库中更新该city |
| CityDataSer.reviseCities | 语法 | Public Boolean reviseCities(CityPO cities) throws RemoteException |
| 前置条件 | 该city在数据库中已存在 |
| 后置条件 | 改变城市之间距离和价格 |
| CityDataSer.getCities | 语法 | Public ArrayList<CityPO> getCities() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回城市对象列表 |

AccountDataSer的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| AccountDataSer.findAccount | 语法 | public ArrayList<AccountPO> findAccount(String keyword) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按关键字查找返回符合条件的账户列表 |
| AccountDataSer.addAccount | 语法 | public boolean addAccount(AccountPO account) throws RemoteException |
| 前置条件 | 同样账户名的账户在数据库中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| AccountDataSer.deleteAccount | 语法 | public boolean deleteAccount(String id) throws RemoteException |
| 前置条件 | 该id对应的账户在数据库中存在 |
| 后置条件 | 在数据库中删除一个po记录 |
| AccountDataSer.reviseAccount | 语法 | public boolean reviseAccount(AccountPO po) throws RemoteException |
| 前置条件 | 该账户在数据库中存在 |
| 后置条件 | 在数据库中更新该账户信息 |
| AccountDataSer.checkAccount | 语法 | public AccountPO checkAccount(String id) throws RemoteException |
| 前置条件 | 该id对应的账户在数据库中存在 |
| 后置条件 | 返回该id对应的账户信息 |
| AccountDataSer.findInitInfo | 语法 | public AcountPO findInitInfo() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回该id对应的期初信息 |
| AccountDataSer.addCount | 语法 | public boolean addCount(CountPO count) throws RemoteException |
| 前置条件 | 同样账户名的账户在数据库中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| AccountDataSer.findOrg | 语法 | public ArrayList<OrgPO> findOrg() throws RemoteException |
| 前置条件 |  |
| 后置条件 | 在数据库中获取并返回机构信息 |
| AccountDataSer.getWorkers | 语法 | Public ArrayList<WorkPO> getWorkers(String id)throws RemoteException |
| 前置条件 | 账户名的账户在数据库中存在 |
| 后置条件 | 在数据库中获取并返回人员信息 |
| AccountDataSer.getVans | 语法 | public ArrayList<VanPO> getVans(String id) throws RemoteException |
| 前置条件 |  |
| 后置条件 | 在数据库中获取并返回货车信息 |
| AccountDataSer.getStock | 语法 | Public ArrayList<StockPO> getStock(String id)throws RemoteException |
| 前置条件 |  |
| 后置条件 | 在数据库中获取并返回库存信息 |
| AccountDataSer.getAccount | 语法 | public ArrayList<AccountPO> getAccount(String id) throws RemoteException |
| 前置条件 |  |
| 后置条件 | 在数据库中获取并返回账户信息 |

OrderNewDataSer的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderNewDataSer.CenterLoadingOrderDSer.create | 语法 | public boolean create(CenterLoadingPO order) throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中新增一个CenterLoadingOrder记录 |
| OrderNewDataSer. CenterLoadingOrderDSer.getHalls | 语法 | public ArrayList<String> getHalls () throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中获取并返回营业厅信息 |
| OrderNewDataSer.ArriveOrderDSer.create | 语法 | public boolean create (ArrivePO order) throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中新增一个ArriveOrder记录 |
| OrderNewDataSer.ArriveOrderDSer.getCities | 语法 | public ArrayList<String> getCities () throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中获取并返回城市信息 |
| OrderNewDataSer.ExpressOrderDSer.create | 语法 | public boolean create (ExpressOrderPO order) throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中新增一个Order记录 |
| OrderNewDataSer.HallLoadingOrderDSer.create | 语法 | public boolean create (HallLoadingOrderPO order) throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中新增一个Order记录 |
| OrderNewDataSer. HallLoadingOrderDSer.getOrgs | 语法 | public ArrayList<String> getOrgs () throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中获取并返回机构信息 |
| OrderNewDataSer. HallLoadingOrderDSer.getVans | 语法 | public ArrayList<String> getVans() throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中获取并返回货车信息 |
| OrderNewDataSer.IncomeOrderDSer.create | 语法 | public boolean create (IncomeOrderPO order) throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中新增一个Order记录 |
| OrderNewDataSer.IncomeOrderDSer.getCouriers | 语法 | public ArrayList<String> getCouriers () throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中获取并返回快递员信息 |
| OrderNewDataSer.PaymentOrderDSer.create | 语法 | public boolean create (PaymentOrderPO order) throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中新增一个Order记录 |
| OrderNewDataSer.ReceiveOrderDSer.create | 语法 | public boolean create (ReceiveOrderPO order) throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中新增一个Order记录 |
| OrderNewDataSer.ReceiveOrderDSer.getExpress | 语法 | public ExpressPO getExpress (String id) throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中获取并返回物流信息 |
| OrderNewDataSer.RecipientOrderDSer.create | 语法 | public boolean create (RecipientOrderPO order) throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中新增一个Order记录 |
| OrderNewDataSer. RecipientOrderDSer.getCities | 语法 | public ArrayList<String> getCities () throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中获取并返回城市信息 |
| OrderNewDataSer.SendOrderDSer.create | 语法 | public boolean create (SendOrderPO order) throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中新增一个Order记录 |
| OrderNewDataSer.SendOrderDSer.getCouriers | 语法 | public ArrayList<String> getCouriers () throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中获取并返回快递员信息 |
| OrderNewDataSer.StockInOrderDSer.create | 语法 | public boolean create (StockInOrderPO order) throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中新增一个Order记录 |
| OrderNewDataSer.StockInOrderDSer.getCities | 语法 | public ArrayList<String> getCities () throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中获取并返回城市信息 |
| OrderNewDataSer.StockOutOrderDSer.create | 语法 | public boolean create (StockOutOrderPO order) throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中新增一个Order记录 |
| OrderNewDataSer.StockOutOrderDSer.getCities | 语法 | public ArrayList<String> getCities () throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中获取并返回城市信息 |
| OrderNewDataSer.TransitOrderDSer.create | 语法 | public boolean create (TransitOrderPO order) throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中新增一个Order记录 |
| OrderNewDataSer.TransitOrderDSer.getCities | 语法 | public ArrayList<String> getCities () throws RemoteException |
| 前置条件 | 单据已生成，且信息完整 |
| 后置条件 | 在数据库中获取并返回城市信息 |

workOrgManDataSer的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| WorkOrgManDataSer.findDriver | 语法 | public ArrayList<workPO> findDriver(String keyword) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按关键词查找并返回相应 DriverPO结果 |
| WorkOrgManDataSer.addDriver | 语法 | public boolean addDriver (DriverPO po) throws RemoteException |
| 前置条件 | 新人员信息已输入完全并确认 |
| 后置条件 | 在数据库中增加一个PO记录 |
| WorkOrgManDataSer.deleteDriver | 语法 | public boolean deleteDriver (String id) throws RemoteException |
| 前置条件 | 数据库中存在相应ID的司机 |
| 后置条件 | 在数据库中删除相应的司机 |
| WorkOrgManDataSer.reviseDriver | 语法 | public boolean reviseDriver(DriverPO) throws RemoteException |
| 前置条件 | 数据库中存在相应ID的司机 |
| 后置条件 | 在数据库中更新相应ID司机的信息 |
| WorkOrgManDataSer.checkDriver | 语法 | public WorkPO checkDriver(String id) throws RemoteException |
| 前置条件 | 数据库中存在相应ID的司机 |
| 后置条件 | 返回相应司机的信息 |
| WorkOrgManDataSer.findOrg | 语法 | public ArrayList<OrgPO> findOrg(String keywords) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按关键词查找并返回相应的OrgPO结果 |
| WorkOrgManDataSer.addOrg | 语法 | public boolean addOrg(OrgPO po) throws RemoteException |
| 前置条件 | 新机构信息已输入完全并确认 |
| 后置条件 | 在数据库中增加一个PO记录 |
| WorkOrgManDataSer.deleteOrg | 语法 | public boolean deleteOrg(String id) throws RemoteException |
| 前置条件 | 数据库中存在相应ID的机构 |
| 后置条件 | 在数据库中删除相应机构的信息 |
| WorkOrgManDataSer.reviseOrg | 语法 | public boolean reviseOrg(OrgPO) throws RemoteException |
| 前置条件 | 数据库中存在相应ID的机构 |
| 后置条件 | 在数据库中更新相应机构的信息 |
| WorkOrgManDataSer.checkOrg | 语法 | public OrgPO checkOrg(String id) throws RemoteException |
| 前置条件 | 数据库中存在相应ID的机构 |
| 后置条件 | 返回该机构的信息 |
| WorkOrgManDataSer.findVan | 语法 | public ArrayList<VanPO> findVan(String keyword) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按关键词查找并返回相应的VanPO结果 |
| WorkOrgManDataSer.addVan | 语法 | public boolean addVan(VanPO Van) throws RemoteException |
| 前置条件 | 新货车信息已输入完全并确认 |
| 后置条件 | 在数据库中增加一个PO记录 |
| WorkOrgManDataSer.deleteVan | 语法 | public boolean deleteVan(String id) throws RemoteException |
| 前置条件 | 数据库中存在相应ID的货车 |
| 后置条件 | 在数据库中删除相应货车的信息 |
| WorkOrgManDataSer.reviseVan | 语法 | public boolean reviseOrg(VanPO Van) throws RemoteException |
| 前置条件 | 数据库中存在相应ID的货车 |
| 后置条件 | 在数据库中更新相应机构的信息 |
| WorkOrgManDataSer.checkVan | 语法 | public VanPO checkVan(String id) throws RemoteException |
| 前置条件 | 数据库中存在相应ID的货车 |
| 后置条件 | 返回该货车的信息 |
| WorkOrgManDataSer.getWage | 语法 | public WagePO getWage(String id) throws RemoteException |
| 前置条件 | 信息已输入完全并确认 |
| 后置条件 | 从数据库中获取相应人的薪水 |
| WorkOrgManDataSer.reviseWage | 语法 | public boolean reviseWage (WagePO) throws RemoteException |
| 前置条件 | 数据库中存在相应ID的机构 |
| 后置条件 | 在数据库中更新相应的薪水信息 |
| WorkOrgManDataSer.findWork | 语法 | public ArrayList<WorkPO> findWork(String keyword) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按关键词查找并返回相应的WorkPO结果 |
| WorkOrgManDataSer.addWork | 语法 | public boolean addWork(WorkPO work) throws RemoteException |
| 前置条件 | 新工人信息已输入完全并确认 |
| 后置条件 | 在数据库中增加一个PO记录 |
| WorkOrgManDataSer.deleteWork | 语法 | public boolean deleteWork(String id) throws RemoteException |
| 前置条件 | 数据库中存在相应ID的工人 |
| 后置条件 | 在数据库中删除相应工人的信息 |
| WorkOrgManDataSer.reviseWork | 语法 | public boolean reviseWork(WorkPO work) throws RemoteException |
| 前置条件 | 数据库中存在相应ID的工人 |
| 后置条件 | 在数据库中更新相应工人的信息 |
| WorkOrgManDataSer.checkWork | 语法 | public WorkPO checkWork(String id) throws RemoteException |
| 前置条件 | 数据库中存在相应ID的工人 |
| 后置条件 | 返回该工人的信息 |

StockDataSer的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockDataSer.getWarning | 语法 | public ArrayList<Integer> getWarning(String id) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回数据库中各个区的警戒线比例 |
| StockDataSer.setWarning | 语法 | public boolean setWarning(int w,int part,String id) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 设置相应区的警戒线比例为w |
| StockDataSer.getInOut | 语法 | public ArrayList<OrderPO> getInOut(Date start, Date end,String id) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回数据库中开始时间和结束时间之间的收款单和付款单列表 |
| StockDataSer.getStock | 语法 | public StockPO getStock(String id)throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回数据库中库存的情况，包括各个架上的货物 |
| StockDataSer.getShelves | 语法 | public ArrayList<ShelfPO> getShelves(String id)throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回数据库中所有架的信息 |
| StockDataSer.addShelf | 语法 | public boolean addShelf(ShelfPO shelf) throws RemoteException |
| 前置条件 | 数据库中不含有同样ID的架 |
| 后置条件 | 在数据库中增加一个架的信息 |
| StockDataSer.deleteShelf | 语法 | public boolean deleteShelf(String id) throws RemoteException |
| 前置条件 | 数据库中含有同样ID的架 |
| 后置条件 | 在数据库中删除相应的架 |
| StockDataSer.reviseShelf | 语法 | public boolean reviseShelf(ShelfPO shelf) throws RemoteException |
| 前置条件 | 数据库中含有同样ID的架 |
| 后置条件 | 在数据库中修改相应的架 |
| StockDataSer.getShelf | 语法 | public ShelfPO getShelf(String id) throws RemoteException |
| 前置条件 | 数据库中含有同样ID的架 |
| 后置条件 | 返回数据库中相应架的信息 |
| StockDataSer.setPoint | 语法 | public boolean setPoint(String id, Date date) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 在数据库中增加一个盘点截止点 |

OrderApproveDataSer模块接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderApproveDataSer.update | 语法 | public boolean update(ArrayList<OrderPO> po)throws RemoteException |
| 前置条件 | Order存在于数据库中 |
| 后置条件 | 在数据库中更新订单审批状态 |
| OrderApproveDataSer.getOrder | 语法 | public OrderPO getOrder(String id) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 从数据库获得并返回相应的单据信息 |
| OrderApproveDataSer.getOrders | 语法 | public ArrayList<OrderPO> getOrder(String id) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 从数据库获得并返回相应的单据信息 |
| OrderApproveDataSer.getExamine | 语法 | public ArrayList<OrderPO> getExamine (String id) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 从数据库获得并返回相应的单据审批信息 |

CheckDataSer模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| CheckDataSer.seeDiary | 语法 | public ArrayList<DiaryPO> seeDiary(Date start,Date end) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 根据输入的开始和结束日期从数据库中获得日志记录，返回相应的数据 |
| CheckDataSer.findCollect | 语法 | public ArrayList<IncomePO> findCollect(Date start,Date end) throw RemoteException |
| 前置条件 | 无 |
| 后置条件 | 根据输入的开始和结束日期从数据库收款单记录，返回相应的数据 |
| CheckDataSer.getCollectRec | 语法 | public ArrayList<IncomePO> getCollectRec(Date date,String id) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 根据时间和营业厅名，从数据库中获取并返回该营业厅的所有收款单记录 |
| CheckDataSer.getHall() | 语法 | public ArrayList<HallPO> getHall() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 从数据库中获取并返回所有的营业厅编号和名称 |