**TransportData的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| TransportDataService  .addCommodity | 语法 | public boolean addCommodity(CommodityPo commodity) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| TransportDataService  .findCommodity | 语法 | Public CommodityPo findCommodity(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id进行查找返回相应的CommodityPo结果 |
| TransportDataService  .deleteCommodity | 语法 | Public boolean deleteCommodity(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id查找到相应的CommodityPo结果并删除 |
| TransportDataService  .updateCommodity | 语法 | Public Boolean updateCommodity(CommodityPo commodity) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 查找到相应的CommodityPo并用传入的commodity替换 |
| TransportDataService  .addSend | 语法 | public boolean addSend(SendPo send) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| TransportDataService  .deleteSend | 语法 | Public boolean deleteSend(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id查找到相应的LoadPo结果并删除 |
| TransportDataService  .addLoad | 语法 | public boolean addLoad(LoadPo load) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| TransportDataService  .deleteLoad | 语法 | Public boolean deleteLoad(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id查找到相应的LoadPo结果并删除 |
| TransportDataService  .addArrival | 语法 | public boolean addArrival(ArrivalPo arrival) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| TransportDataService  .deleteArrival | 语法 | Public boolean deleteArrival(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id查找到相应的ArrivalPo结果并删除 |
| TransportDataService  .addRecieve | 语法 | public boolean addRecieve(RecievePo Recieve) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| TransportDataService  .deleteRecieve | 语法 | Public boolean deleteRecieve(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id查找到相应的RecievePo结果并删除 |
| TransportDataService  .addVehicle | 语法 | public boolean addVehicle(VehiclePo vehicle) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| TransportDataService  .findVehicle | 语法 | Public VehiclePo findVehicle(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id进行查找返回相应的VehiclePo结果 |
| TransportDataService  .deleteVehicle | 语法 | Public boolean deleteVehicle(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id查找到相应的vehiclePo结果并删除 |
| TransportDataService  .updateVehicle | 语法 | Public Boolean updateVehicle(VehiclePo vehicle) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 查找到相应的vehiclePo并用传入的vehicle替换 |
| TransportDataService  .addDriver | 语法 | public boolean addDriver(DriverPo driver) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| TransportDataService  .findDriver | 语法 | Public DriverPo findDriver(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id进行查找返回相应的driverPo结果 |
| TransportDataService  .deleteDriver | 语法 | Public boolean deleteDriver(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id查找到相应的driverPo结果并删除 |
| TransportDataService  .updateDriver | 语法 | Public Boolean updateDriver(DriverPo driver) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 查找到相应的driverPo并用传入的driver替换 |
| TransportDataService  .addRecieve | 语法 | public boolean addRecieve(RecievePo recieve) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| TransportDataService  .deleteRecieve | 语法 | Public boolean deleteRecieve(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id查找到相应的recievePo结果并删除 |

**WarehouseData的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| WarehouseDataService  .addWarehouse | 语法 | public boolean addWarehouse(WarehousePo warehouse) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| WarehouseDataService  .findWarehouse | 语法 | Public WarehousePo findWarehouse(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id进行查找返回相应的warehousePo结果 |
| WarehouseDataService  .deleteWarehouse | 语法 | Public boolean deleteWarehouse(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id查找到相应的warehousePo结果并删除 |
| WarehouseDataService  .updateWarehouse | 语法 | Public Boolean updateWarehouse(WarehousePo warehouse) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 查找到相应的WarehousePo并用传入的Warehouse替换 |
| WarehouseDataService  .addCheckin | 语法 | public boolean addCheckin(CheckinPo checkin) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| WarehouseDataService  .findCheckin | 语法 | Public CheckinPo findCheckin(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id进行查找返回相应的CheckinPo结果 |
| WarehouseDataService  .deleteCheckin | 语法 | Public boolean deleteCheckin(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id查找到相应的CheckinPo结果并删除 |
| WarehouseDataService  .updateCheckin | 语法 | Public Boolean updateCheckin(CheckinPo Checkin) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 查找到相应的CheckinPo并用传入的Checkin替换 |
| WarehouseDataService  .addCheckout | 语法 | public boolean addCheckout(CheckoutPo checkout) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| WarehouseDataService  .findCheckout | 语法 | Public CheckoutPo findCheckout(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id进行查找返回相应的CheckoutPo结果 |
| WarehouseDataService  .deleteCheckout | 语法 | Public boolean deleteCheckout(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id查找到相应的CheckoutPo结果并删除 |
| WarehouseDataService  .updateCheckout | 语法 | Public Boolean updateCheckout(CheckoutPo Checkout) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 查找到相应的CheckoutPo并用传入的Checkout替换 |

**UserData的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| UserDataService  .addUser | 语法 | public boolean addUser(UserPo user) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| UserDataService  .findUser | 语法 | Public UserPo findUser(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id进行查找返回相应的UserPo结果 |
| UserDataService  .deleteUser | 语法 | Public boolean deleteUser(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id查找到相应的UserPo结果并删除 |
| UserDataService  .updateUser | 语法 | Public Boolean updateUser(UserPo user) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 查找到相应的UserPo并用传入的user替换 |

**PersonnelData的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| PersonnelDataService  .addPersonnel | 语法 | public boolean addPersonnel(PersonnelPo personnel) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| PersonnelDataService  .findPersonnel | 语法 | Public PersonnelPo findPersonnel(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id进行查找返回相应的PersonnelPo结果 |
| PersonnelDataService  .deletePersonnel | 语法 | Public boolean deletePersonnel(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id查找到相应的PersonnelPo结果并删除 |
| PersonnelDataService  .updatePersonnel | 语法 | Public Boolean updatePersonnel(PersonnelPo personnel) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 查找到相应的PersonnelPo并用传入的Personnel替换 |

**DepartmentData的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| DepartmentDataService  .addDepartment | 语法 | public boolean addDepartment(DepartmentPo Department) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| DepartmentDataService  .findDepartment | 语法 | Public DepartmentPo findDepartment(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id进行查找返回相应的DepartmentPo结果 |
| DepartmentDataService  .deleteDepartment | 语法 | Public boolean deleteDepartment(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id查找到相应的DepartmentPo结果并删除 |
| DepartmentDataService  .updateDepartment | 语法 | Public Boolean updateDepartment(DepartmentPo Department) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 查找到相应的DepartmentPo并用传入的Department替换 |
| DepartmentDataService  .addCity | 语法 | public boolean addCity(CityPo City) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| DepartmentDataService  .findCity | 语法 | Public CityPo findCity(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id进行查找返回相应的CityPo结果 |
| DepartmentDataService  .deleteCity | 语法 | Public boolean deleteCity(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id查找到相应的CityPo结果并删除 |
| DepartmentDataService  .updateCity | 语法 | Public Boolean updateCity(CityPo City) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 查找到相应的CityPo并用传入的City替换 |

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| LogDataService  .addLog | 语法 | public boolean addOperation(OperationPo Operation) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| LogDataService  .findLog | 语法 | Public List<OperationPo> findOperation(Calendar begin, Calendar end) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id进行查找返回相应的OperationPo结果 |

**LogData的接口规范**

**FinanceData的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| FinanceDataService  .addReceipt | 语法 | public boolean addReceipt(ReceiptPo Receipt) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| FinanceDataService  .findReceipt | 语法 | Public List<ReceiptPo> findReceipt(Calendar date) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id进行查找返回相应的ReceiptPo结果 |
| FinanceDataService  .deleteReceipt | 语法 | Public boolean deleteReceipt(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id查找到相应的ReceiptPo结果并删除 |
| FinanceDataService  .updateReceipt | 语法 | Public Boolean updateReceipt(ReceiptPo Receipt) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 查找到相应的ReceiptPo并用传入的Receipt替换 |
| FinanceDataService  .addPayment | 语法 | public boolean addPayment(PaymentPo payment) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| FinanceDataService  .findPayment | 语法 | Public List<PaymentPo> findPayment(Calendar date) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id进行查找返回相应的PaymentPo结果 |
| FinanceDataService  .deletePayment | 语法 | Public boolean deletePayment(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id查找到相应的PaymentPo结果并删除 |
| FinanceDataService  .updatePayment | 语法 | Public Boolean updatePayment(PaymentPo Payment) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 查找到相应的PaymentPo并用传入的Payment替换 |
| FinanceDataService  .addFreight | 语法 | public boolean addFreight(FreightPo Freight) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| FinanceDataService  .findFreight | 语法 | Public List<FreightPo> findFreight(Calendar date throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id进行查找返回相应的FreightPo结果 |
| FinanceDataService  .deleteFreight | 语法 | Public boolean deleteFreight(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id查找到相应的FreightPo结果并删除 |
| FinanceDataService  .updateFreight | 语法 | Public Boolean updateFreight(FreightPo Freight) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 查找到相应的FreightPo并用传入的Freight替换 |
| FinanceDataService  .addEarning | 语法 | public boolean addEarning(EarningPo Earning) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| FinanceDataService  .findEarning | 语法 | Public List<EarningPo> findEarning(Calendar date) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按date进行查找返回相应的EarningPo结果 |
| FinanceDataService  .deleteEarning | 语法 | Public boolean deleteEarning(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id查找到相应的EarningPo结果并删除 |
| FinanceDataService  .updateEarning | 语法 | Public Boolean updateEarning(EarningPo Earning) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 查找到相应的EarningPo并用传入的Earning替换 |
| FinanceDataService  .addAccount | 语法 | public boolean addAccount(AccountPo Account) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| FinanceDataService  .findAccount | 语法 | Public AccountPo findAccount(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id进行查找返回相应的AccountPo结果 |
| FinanceDataService  .deleteAccount | 语法 | Public boolean deleteAccount(long id) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id查找到相应的AccountPo结果并删除 |
| FinanceDataService  .updateAccount | 语法 | Public Boolean updateAccount(AccountPo Account) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 查找到相应的AccountPo并用传入的Account替换 |
| FinanceDataService  .addInitialInfo | 语法 | public boolean addInitialInfo(InitialInfoPo InitialInfo) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| FinanceDataService  .addPriceStrategy | 语法 | public boolean addPriceStrategy(PriceStrategyPo PriceStrategy) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| FinanceDataService  .findPriceStrategy | 语法 | Public PriceStrategyPo findPriceStrategy() throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 返回相应的PriceStrategyPo结果（唯一） |
| FinanceDataService  .deletePriceStrategy | 语法 | Public boolean deletePriceStrategy() throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 删除相应的PriceStrategyPO（唯一） |
| FinanceDataService  .updatePriceStrategy | 语法 | Public Boolean updatePriceStrategy(PriceStrategyPo PriceStrategy) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 用传入的PriceStrategy替换原PriceStrategyPO |
| FinanceDataService  .addSalaryStrategy | 语法 | public boolean addSalaryStrategy(SalaryStrategyPo SalaryStrategy) throws RemoteException |
| 前置条件 | 同样的Po在记录中不存在 |
| 后置条件 | 在记录中增加一条Po |
| FinanceDataService  .findSalaryStrategy | 语法 | Public SalaryStrategyPo findSalaryStrategy() throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 返回相应的SalaryStrategyPo结果（唯一） |
| FinanceDataService  .deleteSalaryStrategy | 语法 | Public boolean deleteSalaryStrategy() throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 删除相应的SalaryStrategyPO（唯一） |
| FinanceDataService  .updateSalaryStrategy | 语法 | Public Boolean updateSalaryStrategy(SalaryStrategyPo SalaryStrategy) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 用传入的SalaryStrategy替换原StrategyPO |

**ListData的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ListDataService  .findList | 语法 | Public List<ListPo> findList(Calendar begin，Calendar end) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 按id进行查找返回相应的ListPo结果 |
| ListDataService  .updateList | 语法 | Public Boolean updateList(long id，State state) throws RemoteException |
| 前置条件 | 该Po在记录中要存在 |
| 后置条件 | 查找到相应的ListPo并用传入的List替换 |