逻辑层（orderbl）模块的接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| OrderInfo.getAllOrders | 语法 | public ArrayList<orderVO> getAllOrders(long ID); |
| 前置条件 | ID有效 |
| 后置条件 | 返回对应客户的所有订单的VO，否则返回null |
| OrderInfo.isReserved | 语法 | public boolean isReserved(long ID, String address); |
| 前置条件 | ID和address有效 |
| 后置条件 | 返回该酒店是否被该用户预定过 |
| OrderInfo.getCommentableOrderList | 语法 | public ArrayList<OrderVO> getCommentableOrderList (long ID); |
| 前置条件 | ID有效 |
| 后置条件 | 返回当前用户可评价酒店对应的订单列表 |
| OrderInfo.getOrderList | 语法 | getOrderList(long ID, String address); |
| 前置条件 | ID和address有效 |
| 后置条件 | 返回该用户在该酒店的所有订单 |
| Order.getSingleOrder | 语法 | public orderVO getSingleOrder(String address, String orderID); |
| 前置条件 | 酒店工作人员输入订单号 |
| 后置条件 | 检测订单号是否存在，如果存在，则返回对应订单的VO，否则返回null |
| Order.getHotelOrders | 语法 | public ArrayList<orderVO> getHotelOrders (String address, Enum orderType); |
| 前置条件 | 酒店工作人员启动浏览订单流程 |
| 后置条件 | 按orderType的值进行查找并按生成时间顺序返回相应的订单列表 |
| Order.getUserOrders | 语法 | public ArrayList<orderVO> getUserOrders (long ID, Enum orderType); |
| 前置条件 | 客户启动浏览订单流程 |
| 后置条件 | 按orderType的值进行查找并按生成时间顺序返回相应的订单列表 |
| Order.getAbnormalOrdersByDate | 语法 | public ArrayList<orderVO> getAbnormalOrdersByDate (Date date); |
| 前置条件 | 网站营销人员发起浏览每日异常订单情况操作 |
| 后置条件 | 返回所查询日期当天所有异常订单列表 |
| Order.initNewOrder | 语法 | public OrderVO initNewOrder(long ID, String address); |
| 前置条件 | 客户确认即将生成订单的酒店信息 |
| 后置条件 | 初始化订单信息，返回一个被初始化的OrderVO |
| Order.checkNewOrder | 语法 | public ResultMessage checkNewOrder(OrderVO vo); |
| 前置条件 | 客户确认生成订单 |
| 后置条件 | 检查订单要求是否可以得到满足，格式是否正确，并返回结果 |
| Order.addNewOrder | 语法 | public ResultMessage addNewOrder(OrderVO vo); |
| 前置条件 | 订单要求可以得到满足 |
| 后置条件 | 增加新的订单 |
| Order.getPrice | 语法 | public unsigned int getPrice(OrderVO vo); |
| 前置条件 | 用户已填写房间类型和房间数量 |
| 后置条件 | 计算并返回订单价格 |
| Order.setOrderState | 语法 | public boolean setOrderState(Enum orderstate, OrderVO vo); |
| 前置条件 | 客户选择撤销订单或订单已执行或需要管理异常订单执行情况 |
| 后置条件 | 将当前订单状态置为orderstate所示，并更新数据库 |
| Order.isTooLate | 语法 | public boolean isTooLate(OrderVO vo); |
| 前置条件 | 客户发起撤销订单动作 |
| 后置条件 | 计算此次撤销操作是否需要扣除信用值 |
| Order.getOrderDone | 语法 | public boolean getOrderDone(OrderVO vo); |
| 前置条件 | 酒店工作人员确定执行订单 |
| 后置条件 | 更新当前订单状态为已执行订单，并为客户增加与订单金额等值的信用值 |
| Order.withdrawOrder | 语法 | public boolean withdrawOrder(OrderVO vo); |
| 前置条件 | 客户确认撤销订单 |
| 后置条件 | （扣除信用值），系统将订单置为撤销状态，记录撤销时间，更新空房列表 |
| Order.getLatestDoneTime | 语法 | public Time getLatestDoneTime(OrderVO vo); |
| 前置条件 | 客户发起撤销订单动作 |
| 后置条件 | 返回当前订单的最晚执行时间 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| OrderDAO.getUserOrders(long ID, Enum orderType) | 根据orderType的值得到当前用户的对应订单持久化对象列表 | |
| OrderDAO.insert(OrderPO po) | 插入单一持久化对象 | |
| OrderDAO.delete(OrderPO po) | 删除单一持久化对象 | |
| OrderDAO.update(OrderPO po) | 更新单一持久化对象 | |
| OrderDAO.finish() | 结束持久化数据库的使用 | |
| OrderDAO.searchHotelOrdersByType(String address, Enum orderType) | 得到按orderType的值进行查找并按时间顺序返回该酒店的相应订单列表 | |
| OrderDAO.searchAbnormalsByDate(Date date) | 得到查询日期当天所有异常订单列表 | |
| OrderDAO.getSingleOrder(String address, String orderID) | 检测该酒店的订单号是否存在，如果存在，则返回对应订单的VO，否则返回null | |
| OrderDAO.isReserved(long ID, String address); | 返回该酒店是否被该用户预定过 | |
| OrderDAO.getCommentableOrderList (long ID); | 返回当前用户可评价酒店对应的订单列表 | |
| DatabaseFactory.getOrderDatabase | 得到Order数据库服务的引用 | |
| RoomInfoService.getAvailableRoomNum(String address, Enum roomType) | 获取当前酒店roomType房型的空房数量 | |
| RoomInfoService.isTimeAvailable (String address, Enum roomType, Date beginDate, Date finishDate) | 查询用户预定时间段有无用户要求的房型剩余 | |
| RoomInfoService.addSpareRoom(String address, int change, Enum roomType) | 更新（增加）当前酒店被预定房型的可用房间数据 | |
| RoomInfoService.reduceSpareRoom(String address , int change, Enum roomType) | 更新（减少）当前酒店被预定房型的可用房间数据 | |
| Hotel.getHotelBriefInfo(String address) | 得到当前酒店的简要信息 | |
| Hotel.getHotelDetails(String address) | 得到当前酒店的详细信息 | |
| User.getCreditValue(long ID) | 得到当前登陆客户的信用值 | |
| User.changeCreditValue(long ID, int num) | 根据num(可正可负)的值，增加或减少当前客户的信用值 | |
| Strategy.getAvailblePromotionName(OrderVO vo) | 得到当前最优酒店优惠策略的名称 | |
| Strategy.getAvailbleMarketStrategyName(OrderVO vo) | 得到当前最优网站优惠策略的名称 | |
| Strategy.getBestDiscount(OrderVO vo) | 得到当前最终的最优的优惠折扣 | |

数据层（orderDAO）模块的接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| orderDAO.getUserOrders | 语法 | public ArrayList<OrderPO> getUserOrders(long ID, Enum orderType) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 按orderType的值进行查找返回相应的OrderPO列表（全部订单，未执行订单，已执行订单，异常订单和已撤销订单4种） |
| orderDAO.getSingleOrder | 语法 | public OrderPO getSingleOrder(String address, String orderID); |
| 前置条件 | 无 |
| 后置条件 | 检测该酒店的订单号是否存在，如果存在，则返回对应订单的VO，否则返回null |
| orderDAO.isReserved | 语法 | public boolean isReserved(long ID, String address); |
| 前置条件 | 无 |
| 后置条件 | 返回该酒店是否被该用户预定过 |
| orderDAO. getCommentableOrderList | 语法 | public ArrayList<OrderPO> getCommentableOrderList (long ID); |
| 前置条件 | 无 |
| 后置条件 | 返回当前用户可评价酒店对应的订单列表 |
| OrderDAO.insert | 语法 | public void insert(OrderPO po) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 在数据库中增加一个po记录 |
| OrderDAO.delete | 语法 | public void delete(OrderPO po) throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 删除一个po |
| OrderDAO.update | 语法 | public void update(OrderPO po) throws RemoteException; |
| 前置条件 | 在数据库中存在同样ID的po |
| 后置条件 | 更新一个po |
| OrderDAO.finish | 语法 | public void finish() throws RemoteException; |
| 前置条件 | 无 |
| 后置条件 | 结束持久化数据库的使用 |
| OrderDAO.searchAbnormalsByDate | 语法 | public ArrayList<orderPO> searchAbnormalsByDate (Date date); |
| 前置条件 | 无 |
| 后置条件 | 返回所查询日期当天所有异常订单列表，若无异常订单，则返回null（数据库中所有的） |
| OrderDAO.searchHotelOrdersByType | 语法 | public ArrayList<OrderPO> searchHotelOrdersByType(String address, Enum orderType) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按orderType的值进行查找并按生成时间顺序返回该酒店的相应的订单列表，若无对应类型，则返回null |