### 业务逻辑模块的接口模范

1. **logIn模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| logInBL.guestSignUp | 语法 | public GuestVO guestSignUp (UserVO guestVO) |
| 前置条件 | 客户信息符合规范要求 |
| 后置条件 | 系统返回是否成功添加客户；是则进入系统的客户主界面，否则提示添加失败 |
| logInBL.logIn | 语法 | public UserType logIn (String userID, String password) |
| 前置条件 | 用户输入的用户编号和密码符合规范 |
| 后置条件 | 用户被认证并进入其相应的主界面 |
| logInBL.logOut | 语法 | public void logOut (String webManager) |
| 前置条件 | 无 |
| 后置条件 | 系统退出，进入登录主界面 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| userBL.add(UserVo newUserVo, UserType userType) | 新增该用户并持久化保存该用户的数据 | |
| userBL.getLogInfo(String userID, UserType userType) | 返回与该ID对应的user密码 | |

1. **order模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| orderBL.getAllOrderGenerals | 语法 | public Iterator<OrderGeneralVO> getAllOrderGenerals(String userID, UserType userType) |
| 前置条件 | 客户／酒店工作人员已经登录 |
| 后置条件 | 无 |
| orderBL.getSpecialOrderGenerals | 语法 | public Iterator<OrderGeneralVO> getSpecialOrderGenerals(String userID, UserType userType, OrderState orderState) |
| 前置条件 | 客户／酒店工作人员已经登录 |
| 后置条件 | 无 |
| orderBL.getOrderDetail | 语法 | public OrderVO getOrderDetail(String orderID) |
| 前置条件 | 客户／酒店工作人员／网站营销人员已经登录 |
| 后置条件 | 无 |
| orderBL.getEvaluations | 语法 | public Iterator<HotelEvaluationVO> getEvaluations(String hotelID) |
| 前置条件 | 客户／酒店工作人员已经登录 |
| 后置条件 | 无 |
| orderBL.getCalculatedPrice | 语法 | public int getCalculatedPrice (PreOrderVO preOrderVO) |
| 前置条件 | 客户已经登录，填写订单的内容符合规则且完整 |
| 后置条件 | 系统生成未执行订单并持久化增加该客户的订单数据 |
| orderBL.createOrder | 语法 | public ResultMessage createOrder(OrderVO orderVO) |
| 前置条件 | 客户已经登录，填写订单的内容符合规则且完整 |
| 后置条件 | 系统生成未执行订单并持久化增加该客户的订单数据 |
| orderBL.undoNormalOrder | 语法 | public ResultMessage undoNormalOrder(String orderID) |
| 前置条件 | 客户已经登录，订单状态为未执行 |
| 后置条件 | 系统更改状态为已执行订单并持久化保存该客户的订单数据 |
| orderBL.addEvaluation | 语法 | public ResultMessage addEvaluation(GuestEvaluationVO evaluationVO) |
| 前置条件 | 客户已经登录，订单状态为未执行 |
| 后置条件 | 系统更改状态为已执行订单并持久化保存该客户的订单数据 |
| orderBL.getAllGuestCommentOrderGeneral | 语法 | public Iterator<OrderGeneralVO> getAllGuestCommentOrderGeneral(String guestID, boolean hasCommented) |
| 前置条件 | 客户已经登录 |
| 后置条件 | 无 |
| orderBL.getMyOrdersOfThisHotel | 语法 | public Iterator<OrderGeneralVO> getMyOrdersOfThisHotel(String guestID, String hotelID) |
| 前置条件 | 客户已经登录 |
| 后置条件 | 无 |
| orderBL.updateCheckIn | 语法 | public ResultMessage updateCheckIn(CheckInVO checkInVO) |
| 前置条件 | 酒店工作人员已经登录 |
| 后置条件 | 系统更改此订单状态为已执行订单并持久化保存该客户的订单数据 |
| orderBL.updateCheckOut | 语法 | public ResultMessage updateCheckOut(CheckOutVO checkOutVO) |
| 前置条件 | 酒店工作人员已经登录 |
| 后置条件 | 系统更改此订单状态和酒店剩余房间数量，并持久化保存该客户的订单数据 |
| orderBL.getAllHotelCheckOutOrderGeneral | 语法 | public Iterator<OrderGeneralVO> getAllHotelCheckOutOrderGeneral(String hotelID, boolean hasCheckOut) |
| 前置条件 | 酒店工作人员已经登录 |
| 后置条件 | 无 |
| orderBL.undoAbnormalOrder | 语法 | public ResultMessage undoAbnormalOrder(String orderID, double percent) |
| 前置条件 | 网站营销人员已登录，订单状态为异常 |
| 后置条件 | 系统更改订单状态为已撤销订单并持久化保存该客户的订单数据 |
| orderBL.getAllAbnormalOrderGeneral | 语法 | public List<OrderGeneralVO> getAllAbnormalOrderGeneral(LocalDate date) |
| 前置条件 | 网站营销人员已经登录 |
| 后置条件 | 无 |
| orderBL.getAllUnexecutedOrderGeneral | 语法 | public List<OrderGeneralVO> getAllUnexecutedOrderGeneral(LocalDate date) |
| 前置条件 | 网站营销人员已经登录 |
| 后置条件 | 无 |
| orderBL.getBookedHotel | 语法 | public List<String> getBookedHotels(String guestID) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| orderBL.getOrderState | 语法 | public OrderState getOrderState(String guestID, String hotelID) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| orderDataService.getAllGuestOrderGeneral (String guestID) | 根据客户编号查找客户所有订单概况的持久化对象 | |
| orderDataService.getAllHotelOrderGeneral (String hotelID) | 根据酒店编号查找酒店所有订单概况的持久化对象 | |
| orderDataService.getOrderDetail(String orderID) | 根据订单编号查找订单详情的持久化对象 | |
| orderDataService.getEvaluations(String hotelID) | 根据酒店编号查找此酒店评价信息的持久化对象 | |
| orderDataService.createOrder(orderPO OrderPO) | 根据提供的po增加订单的持久化对象 | |
| orderDataService.undoNormalOrder(String orderID) | 根据订单编号修改单一订单持久化对象 | |
| orderDataService.addEvaluation(GuestEvaluationPO guestEvaluationPO) | 修改单一持久化对象订单的评价信息 | |
| orderDataService.updateCheckIn(CheckInPO checkInPO) | 修改单一持久化对象订单的入住信息 | |
| orderDataService.updateCheckOut(CheckOutPO checkOutPO) | 修改单一持久化对象订单的退房信息 | |
| orderDataService.UndoAbnormalOrder(String orderID, double percent) | 根据订单编号和撤销比例修改单一订单持久化对象 | |
| orderDataService.getAllAbnormalOrderGeneral (LocalDate date) | 根据时间查找持久化对象订单的集合 | |
| orderDataService.getAllUnexecutedOrderGeneral (LocalDate date) | 根据时间查找持久化对象订单的集合 | |
| creditBL.addCreditRecord(CreditVO creditVO) | 新增信用记录 | |
| hotelBL.checkIn(String hotelID, RoomType roomType, int roomNum) | 修改酒店剩余房间信息 | |
| hotelBL.checkOut(String hotelID, RoomType roomType, int roomNum) | 修改酒店剩余房间信息 | |
| hotelBL.updateRemainRoomNumForUndoOrder(String hotelID, RoomType roomType, int roomNum) | 更新订单撤销后的酒店剩余房间数 | |
| hotelBL.scoreUpdate(String hotelID,double score) | 更新酒店评分 | |
| promotionBL.getDiscountInSpan(PreOrderVO preOrder) | 根据现有订单状况获取此订单的折扣 | |
| userBL.getSingle(String userID) | 获取客户的单一持久化对象 | |

1. **credit模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| creditBL.charge | 语法 | public double charge(String guestID, double creditNum) |
| 前置条件 | 网站营销人员被识别和授权，并线下已知客户编号和欲充值的的信用值 |
| 后置条件 | 系统为客户信用值充值 |
| creditBL.getAllCreditDetail | 语法 | public Iterator<CreditVO> getAllCreditDetail(final String guestID) |
| 前置条件 | 客户已被识别和授权 |
| 后置条件 | 无 |
| creditBL.getCreditOfOneOrder | 语法 | public Iterator<CreditVO> getCreditOfOneOrder(String orderID) |
| 前置条件 | 无 |
| 后置条件 | 无 |
| creditBL.addCreditRecord | 语法 | public ResultMessage addCreditRecord(CreditVO creditVO) |
| 前置条件 | 用户已被识别和授权 |
| 后置条件 | 在信用中添加此条记录 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| creditDataService.getAllCreditDetail(final String guestID) | 根据客户编号获得单个客户的所有信用记录的持久化对象 | |
| creditDataService.getCreditOfOneOrder(String orderID) | 获得一个订单的所有信用记录持久化对象 | |
| creditDataService.addCredit(CreditVO creditVO) | 为客户添加信用记录 | |
| userBL.modifyCredit(String guestID, double creditNum) | 修改客户的信用信息 | |
| UserBL.getSingle(String userID) | 根据客户编号查看客户持久化对象 | |

## 数据层的分解

数据层主要给业务逻辑层提供数据访问服务，包括对于持久化数据的增、删、改、查。例如，AccountBusinessService的服务主要由AccountDataService提供。由于持久化数据的保存可能存在多种形式：Txt文件、序列化文件、数据库等，所示抽象了数据服务。以AccountDataService为例，数据层模块的具体描述下图所示：

### 数据层模块的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| orderData | 负责保存订单信息，进行增、改、查的操作 |
| hotelData | 负责保存酒店信息，进行增、改、查的操作 |
| promotionData | 负责保存促销策略信息，进行增、删、改、查的操作 |
| marketData | 负责保存会员等级制度信息，进行改、查的操作 |
| guestData | 负责保存客户信息，进行增、改、查的操作 |
| hotelWorkerData | 负责保存酒店工作人员信息，进行增、改、查的操作 |
| webMarketerData | 负责保存网站营销人员信息，进行改、查的操作 |
| webManagerData | 负责保存网站管理人员信息，进行改、查的操作 |

### 数据层模块的接口规范

1. **orderData模块的接口规范**

|  |  |  |
| --- | --- | --- |
| orderDataService.createOrder | 语法 | public ResultMessage createOrder (OrderPO order) |
| 前置条件 | 客户已被认证和授权 |
| 后置条件 | 系统生成未执行订单并持久化增加该客户的订单数据 |
| orderDataService.executeOrder | 语法 | public ResultMessage executeOrder (String orderID) |
| 前置条件 | 酒店工作人员已被认证和授权 |
| 后置条件 | 系统更改状态为已执行订单并持久化保存该客户的订单数据 |
| orderDataService.undoAbnormalOrder | 语法 | public ResultMessage undoAbnormalOrder (String orderID, double percent) |
| 前置条件 | 网站营销人员已被认证和授权 |
| 后置条件 | 系统更改状态为已撤销订单、恢复一定比例的信用值并持久化保存该客户的订单数据 |
| orderDataService.undoNormalOrder | 语法 | public ResultMessage undoNormalOrder (String orderID) |
| 前置条件 | 客户已被认证和授权，订单状态为未执行 |
| 后置条件 | 系统更改状态为已撤销订单并持久化保存该客户的订单数据 |
| orderDataService.getOrderDetails | 语法 | public OrderPO getOrderDetail(String orderID) |
| 前置条件 | 用户已被认证和授权 |
| 后置条件 | 无 |
| orderDataService.getAllGuestOrderGeneral | 语法 | public List<OrderGeneralPO> getAllGuestOrderGeneral (String guestID) |
| 前置条件 | 客户已被认证和授权 |
| 后置条件 | 无 |
| orderDataService.getAllHotelOrderGeneral | 语法 | public List<OrderGeneralPO> getAllHotelOrderGeneral (String orderID) |
| 前置条件 | 酒店工作人员已被认证和授权 |
| 后置条件 | 无 |
| orderDataService.getAllAbnormalOrderGeneral | 语法 | public List<OrderGeneralPO> getAllAbnormalOrderGeneral (LocalDate date) |
| 前置条件 | 网站营销人员已被认证和授权 |
| 后置条件 | 无 |
| orderDataService.getAllUnexecutedOrderGeneral | 语法 | public List<OrderGeneralPO> getAllUnexecutedOrderGeneral (LocalDate date) |
| 前置条件 | 网站营销人员已被认证和授权 |
| 后置条件 | 无 |
| orderDataService.updateCheckIn | 语法 | ResultMessage updateCheckIn(CheckInPO checkInPO) |
| 前置条件 | 网站营销人员已被认证和授权 |
| 后置条件 | 无 |
| orderDataService.updateCheckOut | 语法 | ResultMessage updateCheckOut(CheckOutPO checkOutPO) |
| 前置条件 | 网站营销人员已被认证和授权 |
| 后置条件 | 无 |
| orderDataService.addEvaluation | 语法 | ResultMessage addEvaluation(GuestEvaluationPO guestEvaluationPO) |
| 前置条件 | 网站营销人员已被认证和授权 |
| 后置条件 | 无 |
| orderDataService.getEvaluations | 语法 | List<HotelEvaluationPO> getEvaluations(String hotelID) |
| 前置条件 | 网站营销人员已被认证和授权 |
| 后置条件 | 无 |
| orderDataService.getBookedHotels | 语法 | List<String> getBookedHotels(String guestID) |
| 前置条件 | 网站营销人员已被认证和授权 |
| 后置条件 | 无 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| dataHelper.add(OrderPO orderPO) | 新增订单的持久化对象 | |
| dataHelper.setState(String orderID, OrderState state) | 根据订单编号修改订单持久化对象的状态 | |
| dataHelper.getSingleOrder(String orderID) | 根据客户编号查找客户单个订单的持久化对象 | |
| dataHelper.getAllOrderOfGuest(String guestID) | 根据客户编号查找客户的所有订单持久化对象 | |
| dataHelper.getAllOrderOfHotel(String hotelID) | 根据酒店编号查找酒店的所有订单持久化对象 | |
| dataHelper.getUnexecuted() | 查找所有未执行订单的持久化对象 | |
| dataHelper.getAbnormal() | 查找所有异常订单的持久化对象 | |
| dataHelper.setCheckIn(String orderID, String roomNumber, LocalDateTime checkInTime, LocalDateTime expectLeaveTime) | 根据订单编号修改单个订单持久化对象的入住信息 | |
| dataHelper.setCheckOut(String orderID, LocalDateTime checkOutTime) | 根据订单编号修改单个订单持久化对象的退房信息 | |
| dataHelper.setEvaluation(String orderID, double score, String comment) | 根据订单编号修改单个订单持久化对象的评论信息 | |
| dataHelper.setHasCommentBool(String orderID) | 根据订单编号修改单个订单持久化对象的评价特征值 | |

1. **creditData模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| creditDataService.getAllCreditDetail | 语法 | public List<CreditPO> getAllCreditDetail(String guestID) |
| 前置条件 | 客户已被认证和授权 |
| 后置条件 | 无 |
| creditDataService.addCredit | 语法 | public ResultMessage addCredit(CreditPO creditPO) |
| 前置条件 | 无 |
| 后置条件 | 添加一条信用记录 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| dataHelper.getAllCreditDetail(String guestID) | 获取该客户的所有订单记录 | |
| dataHelper.addCredit(CreditPO creditPO) | 新增信用记录 | |