roombl职责：实现与更新空房信息，更新入住和退房信息相关的服务

roomDAO职责：负责保存酒店的空房信息和入住退房信息，供读取修改写入

strategybl职责：实现与更新酒店促销策略和网站营销策略相关的服务

strategyDAO职责：负责保存酒店促销策略和网站营销策略信息，供读取修改写入

1. roombl模块的接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| RoomInfoService.getAvailableRoomNum | 语法 | public int getAvailableRoomNum(String address, Enum<RoomType> roomType) |
| 前置条件 | 无 |
| 后置条件 | 返回对应房间类型的空房数量 |
| RoomInfoService.isTimeAvailable | 语法 | public boolean isTimeAvailable (String addresss, Enum<RoomType> roomType, Date beginDate, Date finishDate) |
| 前置条件 | 无 |
| 后置条件 | 返回用户预定时间段有无用户要求的房型剩余 |
| RoomInfoService.updateSpareRoom | 语法 | public boolean updateSpareRoom (String address, RoomVO roomvo) |
| 前置条件 | 每次更新可用客房后 |
| 后置条件 | 返回是否更新空房信息成功 |
| RoomInfoService.reduceSpareRoom | 语法 | public boolean reduceRoom(String address, int change, Enum<RoomType> roomType) |
| 前置条件 | （1）用户生成订单时  （2）线下入住并更新入住信息时  （3）异常订单通过酒店工作人员手动延迟入住变为已执行订单时 |
| 后置条件 | 返回是否减少空房成功 |
| RoomInfoService.AddSpareRoom | 语法 | public boolean addRoom(String address, int change, Enum<RoomType> roomType) |
| 前置条件 | （1）更新退房信息时  （2）当正常订单变成异常订单时  （3）用户撤销订单时 |
| 后置条件 | 返回是否增加空房成功 |
| BrowseSpareRoomService.getRoomInfoList | 语法 | public ArrayList<RoomVO> getRoomInfoList (String address) |
| 前置条件 | 启动一个浏览空房信息的请求 |
| 后置条件 | 返回空房信息列表 |
| UpdateCheckInService.getCheckInList | 语法 | public ArrayList<RoomVO> getCheckInList(String address) |
| 前置条件 | 启动一个更新入住信息请求 |
| 后置条件 | 返回入住信息列表 |
| UpdateCheckInService.searchCheckInInfo | 语法 | public RoomVO searchCheckInInfo(String address ,Date time) |
| 前置条件 | 无 |
| 后置条件 | 按入住时间进行查找返回相应的入住信息 |
| UpdateCheckInService.searchCheckInInfo | 语法 | public RoomVO searchCheckInInfo (String address , Enum<RoomType> roomType) |
| 前置条件 | 无 |
| 后置条件 | 按房间类型进行查找返回相应的入住信息 |
| UpdateCheckInService.AddCheckIn | 语法 | public boolean AddCheckIn(String address, RoomVO checkIn) |
| 前置条件 | 启动增加入住信息请求，checkIn符合规范 |
| 后置条件 | 返回是否更新成功 |
| UpdateCheckInService.modifyCheckIn | 语法 | public boolean ModifyCheckIn(String address, RoomVO checkIn) |
| 前置条件 | checkIn符合规范 |
| 后置条件 | 启动修改入住信息请求，返回是否修改成功 |
| UpdateCheckInService.delCheckIn | 语法 | public boolean delCheckIn(String address, RoomVO checkIn) |
| 前置条件 | 启动删除入住信息请求，checkIn在入住信息列表种已存在 |
| 后置条件 | 返回是否删除成功 |
| UpdateCheckInService.validCheckIn | 语法 | public boolean validCheckIn(String address, RoomVO checkIn) |
| 前置条件 | 无 |
| 后置条件 | 返回checkIn的信息是否符合规范 |
| UpdateCheckOutService.getCheckOutList | 语法 | public ArrayList<RoomVO> getCheckOutList(String address) |
| 前置条件 | 启动一个更新退房信息请求 |
| 后置条件 | 返回退房信息列表 |
| UpdateCheckOutService.searchCheckOutInfo | 语法 | public RoomPO getCheckOutInfo(String address, Date time) |
| 前置条件 | 无 |
| 后置条件 | 按实际离开时间进行查找返回相应的退房信息 |
| UpdateCheckOutService.searchCheckOutInfo | 语法 | public RoomPO getCheckOutInfo(String address, Enum<RoomType> roomType) |
| 前置条件 | 无 |
| 后置条件 | 按房间类型进行查找返回相应的退房信息 |
| UpdateCheckOutService.addCheckOut | 语法 | public boolean AddCheckOut(String address, RoomVO roomvo) |
| 前置条件 | 启动增加退房信息请求，checkOut符合规范 |
| 后置条件 | 返回是否更新成功 |
| UpdateCheckOutService.modifyCheckOut | 语法 | public boolean ModifyCheckOut(String address, RoomVO roomvo) |
| 前置条件 | 启动修改退房信息请求，checkOut符合规范 |
| 后置条件 | 返回是否修改成功 |
| UpdateCheckOutService.delCheckOut | 语法 | public boolean delCheckOut (String address, RoomVO roomvo) |
| 前置条件 | 启动删除退房信息请求，checkOut在入住信息列表种已存在 |
| 后置条件 | 返回是否删除成功 |
| UpdateCheckOutService.validCheckOut | 语法 | public boolean validCheckOut(String address, RoomVO checkOut) |
| 前置条件 | 无 |
| 后置条件 | 返回checkOut的信息是否符合规范 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| RoomDAO.getSpareRoomInfoList(String address) | 得到空房信息持久化对象列表 | |
| RoomDAO.getSpareRoomInfo(String address, Enum<RoomType> roomType) | 得到对应房间类型的空房信息持久化对象 | |
| RoomDAO.getCheckInInfoList(String address) | 得到入住信息持久化对象列表 | |
| RoomDAO.getCheckInInfo(String address, String time) | 得到对应入住时间的入住信息持久化对象 | |
| RoomDAO.getCheckInInfo(String address, Enum<RoomType> roomType) | 得到对应房间类型的入住信息持久化对象 | |
| RoomDAO.getCheckOutInfoList(String address) | 得到退房信息持久化对象列表 | |
| RoomDAO.getCheckOutInfo(String address, String time) | 得到对应实际离开时间的退房信息持久化对象 | |
| RoomDAO.getCheckOutInfo(String address, Enum<RoomType> roomType) | 得到对应房间类型的退房信息持久化对象 | |
| RoomDAO.find(long id) | 根据ID查找单一持久化对象RoomPO | |
| RoomDAO.update(RoomPO po) | 更新单一持久化对象 | |
| RoomDAO.insert(RoomPO po) | 在数据库插入Room PO对象 | |
| RoomDAO.delete(RoomPO po) | 在数据库删除某Room PO对象 | |
| DatabaseFactory.getRoomDatabase | 得到Room数据库的服务的引用 | |

1. roomDAO模块的接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| RoomDAO.getSpareRoomInfoList | 语法 | public ArrayList<RoomPO> getSpareRoomInfoList(String address) |
| 前置条件 | 无 |
| 后置条件 | 返回空房列表 |
| RoomDAO.getSpareRoomInfo | 语法 | public RoomPO getSpareRoomInfo(String address , Enum<RoomType> roomType) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按房间类型进行查找返回相应的空房信息 |
| RoomDAO.getCheckInInfoList | 语法 | public ArrayList<RoomPO> getCheckInInfoList(String address) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回入住信息列表 |
| RoomDAO.getCheckInInfo | 语法 | public RoomPO getCheckInInfo(String address , String time) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按入住时间进行查找返回相应的入住信息 |
| RoomDAO.getCheckInInfo | 语法 | public RoomPO getCheckInInfo (String address, Enum<RoomType> roomType) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按房间类型进行查找返回相应的入住信息 |
| RoomDAO.getCheckOutInfoList | 语法 | public ArrayList<RoomPO> getCheckOutInfoList(String address) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回退房信息列表 |
| RoomDAO.getCheckOutInfo | 语法 | public RoomPO getCheckOutInfo(String address, String time) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按实际离开时间进行查找返回相应的退房信息 |
| RoomDAO.getCheckOutInfo | 语法 | public RoomPO getCheckOutInfo(String address, Enum<RoomType> roomType) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按房间类型进行查找返回相应的退房信息 |
| RoomDAO.find | 语法 | public RoomPO find(long id) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的RoomVO结果 |
| RoomDAO.update | 语法 | public void update(RoomPO 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 |
| RoomDAO.insert | 语法 | public void insert(RoomPO po) throws RemoteException |
| 前置条件 | 同样ID的po在Mapper中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| RoomDAO.delete | 语法 | public void delete(RoomPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样ID的po |
| 后置条件 | 删除一个po |
| RoomDAO.init | 语法 | public void init(RoomPO po) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 持久化数据存储 |
| RoomDAO.finish | 语法 | public void finish(RoomPO po) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 结束持久化数据库的使用 |

1. strategybl模块的接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| StrategyInfoService.getAvailble PromotionName | 语法 | public String getAvailblePromotionName (OrderVO order) |
| 前置条件 | 无 |
| 后置条件 | 返回某订单能享受的唯一酒店促销策略折扣名称，如果没有，则返回null |
| StrategyInfoService.getAvailbleMarketStrategyName | 语法 | public String getAvailbleMarketStrategyName (OrderVO order) |
| 前置条件 | 无 |
| 后置条件 | 返回某订单能享受的唯一网站营销策略折扣名称，如果没有，则返回null |
| StrategyInfoService. getBestDiscount | 语法 | public int getBestDiscount(OrderVO order) |
| 前置条件 | 无 |
| 后置条件 | 返回某订单的最终的折扣百分比 |
| UpdateStrategyService.getStrategyList | 语法 | public ArrayList<StrategyVO> getStrategyList(String address, Enum<StrategyType> StrategyType) |
| 前置条件 | 启动一个制定策略的请求 |
| 后置条件 | 返回某种策略类型的折扣列表 |
| UpdateStrategyService.searchStrategyInfo | 语法 | public StrategyVO getStrategyInfo(String address, String name) |
| 前置条件 | 无 |
| 后置条件 | 返回某个折扣名称的具体折扣信息 |
| UpdateStrategyService.add | 语法 | public boolean add(String address, StrategyVO strategy) |
| 前置条件 | 该StrategyVO符合规范，通过Valid方法 |
| 后置条件 | 返回是否制定策略成功 |
| UpdateStrategyService.modify | 语法 | public boolean modify(String address, StrategyVO strategy) |
| 前置条件 | 该StrategyVO符合规范，通过Valid方法 |
| 后置条件 | 返回是否修改策略信息成功 |
| UpdateStrategyService.del | 语法 | public boolean delete(String address, StrategyVO strategy) |
| 前置条件 | 该StrategyVO已存在 |
| 后置条件 | 返回是否删除成功 |
| UpdateStrategyService.Valid | 语法 | public boolean Valid(String address, StrategyVO strategy) |
| 前置条件 | 无 |
| 后置条件 | 返回该策略信息是否符合规范 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| StrategyDAO.getStrategyList(String address, Enum<StrategyType> StrategyType) | 得到策略折扣持久化对象列表 | |
| StrategyDAO.getStrategyInfo(String address, String strategyName) | 得到对应折扣名称的折扣策略持久化对象 | |
| StrategyDAO.find(long id) | 根据ID查找单一持久化对象StrategyPO | |
| StrategyDAO.update(StrategyPO po) | 更新单一持久化对象 | |
| StrategyDAO.insert(StrategyPO po) | 在数据库插入StrategyPO对象 | |
| StrategyDAO.delete(StrategyPO po) | 在数据库删除某StrategyPO对象 | |
| DatabaseFactory.getStrategyDatabase | 得到Strategy数据库的服务的引用 | |
| HotelInfoService.getHotelBrifInfo(String address) | 得到酒店的简要信息(商圈名称) | |
| User.getRegularVipInfo(String ID) | 得到普通会员信息 | |
| User.getEnterpriseVipInfo(String ID) | 得到企业会员信息 | |

1. strategyDAO模块的接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| StrategyDAO.getStrategyList | 语法 | public ArrayList<StrategyPO> getStrategyList(String address, Enum<StrategyType> StrategyType) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 返回某种策略折扣列表 |
| StrategyDAO.getStrategyInfo | 语法 | public StrategyPO getMarketStrategyInfo(String address, String strategyName) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的策略折扣 |
| StrategyDAO.find | 语法 | public StrategyPO find(long id) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 按ID进行查找返回相应的StrategyVO结果 |
| StrategyDAO.update | 语法 | public void update(StrategyPO 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 |
| StrategyDAO.insert | 语法 | public void insert(StrategyPO po) throws RemoteException |
| 前置条件 | 同样ID的po在Mapper中不存在 |
| 后置条件 | 在数据库中增加一个po记录 |
| StrategyDAO.delete | 语法 | public void delete(StrategyPO po) throws RemoteException |
| 前置条件 | 在数据库中存在同样ID的po |
| 后置条件 | 删除一个po |
| StrategyDAO..init | 语法 | public void init(StrategyPO po) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 持久化数据存储 |
| StrategyDAO.finish | 语法 | public void finish(StrategyPO po) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 无结束持久化数据库的使用 |