模块：Userbl 职责：负责对应于登录界面及相应用户数据更改界面等所需要的服务

接口规范：

|  |  |  |
| --- | --- | --- |
| 提供的服务(供接口) | | |
| User.login | 语法 | Public ResultMessage login(String username,String password); |
| 前置条件 | Username和password输入符合规范 |
| 后置条件 | 查找是否有相对应的user,根据输入的password返回登录验证的结果 |
| User.changeInfo | 语法 | Public ResultMessage changeInfo(UserVO vo); |
| 前置条件 | 用户已登录（即userId不为空） |
| 后置条件 | 根据userid更改对应user的信息 |
| User.register | 语法 | Public ResultMessage register(String username,String password); |
| 前置条件 | 用户未登陆 |
| 后置条件 | 先根据username查询是否注册过，若无，则经过加密后将信息写入数据库 |
| 需要的服务(需接口) | | |
| DatabaseFactory.getUserDatabase | 得到User数据库服务的引用 | |
| UserDataService.insert(UserPO po) | 在数据库中插入po对象 | |
| UserDataService.getInfo(long userid) | 得到user的基本信息 | |
| UserDataService.changeInfo(UserPO po) | 更改user的基本信息 | |
| Member.getInfo(long memberid) | 得到对应的会员信息 | |

模块：Browsebl 职责：提供浏览各类信息的界面的服务

接口规范：

|  |  |  |
| --- | --- | --- |
| 提供的服务(供接口) | | |
| Browse.getOrdersInfo | 语法 | Public List<Order> getOrdersInfo(long userid,int method); |
| 前置条件 | 用户已登录，且这里用户为客户 |
| 后置条件 | 根据对应的method 查找对应类型的对应用户订单信息，其中0为所有，1为未执行，2为已执行，3为已撤销，4为异常 |
| Browse.getHotelOrdersInfo | 语法 | Public List<Order> getHotelOrdersInfo(long hotelid,int method); |
| 前置条件 | 酒店管理人员已登录 |
| 后置条件 | 根据对应的method 查找对应类型的对应酒店订单信息，其中0为所有，1为未执行，2为已执行，3为已撤销，4为异常 |
| Browse.getWEBOrdersInfo | 语法 | Public List<Order> getWEBOrdersInfo(); |
| 前置条件 | 网站营销人员已登录 |
| 后置条件 | 根据对应的method 查找对应类型的当天当时间异常订单信息 |
| Browse.getCreditInfo(long userid) | 语法 | Public List<String> getCreditInfo(long userid); |
| 前置条件 | 客户已登录 |
| 后置条件 | 根据对应的user查找这个user的所有的信用纪录 |
| Browse.getHotels | 语法 | Public List<RoughHotelInfo> getHotels(Map<field,value>); |
| 前置条件 | 无 |
| 后置条件 | 根据map表里的字段与值的限制，查找相应的酒店列表信息，对数据处理后返回相应粗略酒店信息 |
| Browse.getHotel | 语法 | Public Map<field,value> getHotel(long hotelid); |
| 前置条件 | 无 |
| 后置条件 | 根据对应的id,查询相应酒店的详细信息，处理后返回 |
| Browse.getStrategyInfo | 语法 | Public String getStrategyInfo(int orderid); |
| 前置条件 | 用户选择一个订单 |
| 后置条件 | 根据对应订单，查看该订单生效的策略信息 |
| Browse.getMemberInfo | 语法 | Public String getMemberInfo(id userid); |
| 前置条件 | 用户已登录 |
| 后置条件 | 根据对应的id，查询相应用户的会员信息并返回 |
| 需要的服务(需接口) | | |
| DatabaseFactory.getBrowseDatabase | 得到Browse数据库服务的引用 | |
| BrowseDataService.insert(BrowsePO po) | 在Browse数据库中插入po（搜索记录） | |
| BrowseDateService.getInfo(userid) | 返回一个用户的浏览/搜索记录 | |
| BrowseDataService.clear(userid) | 晴空一个用户的浏览搜索记录 | |
| Hotel.getHotels(Map<field,value>) | 得到基于map规则的酒店信息列表 | |
| Hotel.getInfo(long Hotelid) | 得到hotel的详细信息 | |
| Credit.getInfo(userID) | 得到user的信用信息 | |
| Order.getuserInfo(int userid,int method) | 得到user的对应method的对应类别的订单信息 | |
| Order.getHotelInfo(int hotelid,int method) | 得到hotel的对应method的对应类别的订单信息 | |
| Order.getWEBInfo() | 得到当天当前时间前所有的异常订单信息 | |
| Member.getInfo(userid) | 得到对应的会员信息 | |
| Strategy.getInfo(orderid) | 得到对应订单的策略信息 | |
| Info.add(map<fiield,value>); | 插入 | |

模块：Order 职责：负责对应于关于订单的所有操作的服务

接口规范：

|  |  |  |
| --- | --- | --- |
| 提供的服务(供接口) | | |
| Order.getuserInfo | 语法 | Public List<Order> getuserInfo(int userid,int method); |
| 前置条件 | 用户已登录，且这里用户为客户 |
| 后置条件 | 根据对应的method 查找对应类型的对应用户订单信息，其中0为所有，1为未执行，2为已执行，3为已撤销，4为异常 |
| Order.getHotelInfo | 语法 | Public List<Order> getHotelOrdersInfo(long hotelid,int method); |
| 前置条件 | 酒店管理人员已登录 |
| 后置条件 | 根据对应的method 查找对应类型的对应酒店订单信息，其中0为所有，1为未执行，2为已执行，3为已撤销，4为异常 |
| Order.getWEBInfo | 语法 | Public List<Order> getWEBOrdersInfo(); |
| 前置条件 | 网站营销人员已登录 |
| 后置条件 | 根据对应的method 查找对应类型的当天当时间异常订单信息 |
| Order.create | 语法 | Public ResultSet create(map<field,value>); |
| 前置条件 | 客户已登录 |
| 后置条件 | 根据map中field字段与value注入相关信息，返回创建消息 |
| Order.userCancel | 语法 | Public ResultSet userCancel(long orderid); |
| 前置条件 | 用户已登录 |
| 后置条件 | 所orderid所对应订单为未执行订单，撤销该订单 |
| Order.comment | 语法 | Public ResultSet comment(long orderid,int score,String comment); |
| 前置条件 | 客户已登录 |
| 后置条件 | 若订单已执行，则写入相关评价并返回结果 |
| Order.execute | 语法 | Public ResultSet(long orderid); |
| 前置条件 | 酒店工作人员已登录 |
| 后置条件 | 根据对应订单，判断订单状态，执行或补登记执行订单，并作出信用值的相应改变 |
| Order.WEBCancel | 语法 | Public ResultSet WEBCancel(long orderid); |
| 前置条件 | 网站营销人员已登录且线下已判断订单可撤销 |
| 后置条件 | 根据对应的订单号，撤销订单，并返回状态 |
| Order.getTotal() | 语法 | Public double getTotal(long orderid); |
| 前置条件 | 无 |
| 后置条件 | 计算订单总价，并返回 |
| 需要的服务(需接口) | | |
| DatabaseFactory.getOrderDatabase | 得到Order数据库服务的引用 | |
| OrderDataService.insert(OrderPO po) | 在Order数据库中插入po | |
| OrderDataService.changeStatus(OrderPO po); | 改变订单的状态 | |
| OrderDataService.getInfo(long orderid); | 获取订单的信息 | |
| Hotel.getRooms(long hotelid) | 得到该hotel的房间信息 | |
| Hotel.lockroom(long hotelid,map<field,value>) | 锁定该酒店的对应field的房间 | |
| Credit.change(long userid,int value); | 改变该user的信用值信息 | |
| Comment.add(long hotelid,int score,String comment) | 评论酒店并记录 | |
| Member.getInfo(userid) | 得到对应的会员信息 | |
| Strategy.getInfo(orderid) | 得到对应订单的策略信息 | |
| Strategy.getAvaliableInfo(map<field,value>) | 得到当前限制条件下可用的策略信息 | |

//下面不是的

|  |  |
| --- | --- |
| HotelDataService.getHotels(Map<field,value>) | 得到基于map规则的酒店信息列表 |
| HotelDataService.getInfo(long Hotelid) | 得到hotel的详细信息 |
| CreditDataService.getInfo(userID) | 得到user的信用信息 |
| OrderDataService.getInfo(int method) | 更改user的基本信息 |
| MemberDataService.getInfo(UserPO po) | 得到对应的会员信息 |

DATA：userDataService

|  |  |  |
| --- | --- | --- |
| 提供的接口 | | |
| UserDataService.insert | 语法 | Public void insert(UserPO po) throws RemoteException |
| 前置条件 | 同样ID的po在数据库中不存在 |
| 后置条件 | 在数据库中增加一条PO记录 |
| UserDataService.changeInfo | 语法 | Public void update(UserPO po) throws RemoteException |
| 前置条件 | 同样ID的PO在数据库中存在 |
| 后置条件 | 更新这条PO |
| UserDataService.getInfo | 语法 | Public UserPO getInfo(long userID) throws RemoteException |
| 前置条件 | 该ID的po在数据库中存在 |
| 后置条件 | 查询并返回这条PO |

BrowseDataService

|  |  |  |
| --- | --- | --- |
| 提供的接口 | | |
| BrowseDataService.insert | 语法 | Public void insert(BrowsePO po) throws RemoteException |
| 前置条件 | 同样ID的po在数据库中不存在 |
| 后置条件 | 在数据库中增加一条PO记录 |
| BrowseDataService.getInfo | 语法 | Public List<BrowsePO> getInfo(long userID) throws RemoteException |
| 前置条件 | 该ID至少有一条po在数据库中存在 |
| 后置条件 | 查询并返回该ID的所有PO |
| BrowseDataService.clear | 语法 | Public void clear(long userid) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 清空数据库中该ID的所有PO |

OrderDataService

|  |  |  |
| --- | --- | --- |
| OrderDataService.insert | 语法 | Public void insert(OrderPO po) throws RemoteException |
| 前置条件 | 同样ID的po在数据库中不存在 |
| 后置条件 | 在数据库中增加一条PO记录 |
| OrderDataService.changeInfo | 语法 | Public void changeInfo(OrderPO po) throws RemoteException |
| 前置条件 | 同样ID的PO在数据库中存在 |
| 后置条件 | 更新这条PO |
| OrderDataService.getInfo | 语法 | Public OrderPO getInfo(long OrderID) throws RemoteException |
| 前置条件 | 该ID的po在数据库中存在 |
| 后置条件 | 查询并返回这条PO |

UserPO 定义：

Public class UserPO implement Serializable {

long id;

String username;

String password;

Role role;

Public UserPO(int id,String username,Stirng password,Role role){

This.id = id;

This.username = username;

This.password = password;

This.role = role;

}

…

Get/set

}

OrderPO 定义：

Public class orderPO implement Serializable {

Long id;

Status status;

String begintime;

String endtime;

String actualBeginTime;

Int expectedpeople;

Boolean isHaveChild;

Hotelid hotelid;

Int[] roomnumber;

…

构造函数

Get/set

}

BrowsePO 定义：

Public class BrowsePO implement Serializable {

Int browseID;

Int userID;

Int toUserID;

Time time;

构造函数

Get/set

}