#### userBL模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| User.getAll | 语法 | Public List<UserVO> getAll(UserType userType) |
| 前置条件 | 无 |
| 后置条件 | 系统返回指定用户类型的所有用户信息 |
| User.getSingle | 语法 | Public UserVO getSingle(String userID，UserType userType) |
| 前置条件 | userID.Length=对应用户类型的ID长度 |
| 后置条件 | 系统返回与该ID对应的user信息 |
| User.add | 语法 | Public ResultMessage add(UserVO newUserVO) |
| 前置条件 | 输入的userID不存在，password和所有信息符合规范 |
| 后置条件 | 系统新增该用户并持久化保存该用户的数据 |
| User.addHotel | 语法 | Public ResultMessage addHotel(HotelVO newHotelVO，String hotelID) |
| 前置条件 | 输入的hotelID不存在,所有酒店信息符合规范 |
| 后置条件 | 系统新增该酒店并持久化保存该酒店的数据 |
| User.modify | 语法 | Public ResultMessage modify(UserVO userVO) |
| 前置条件 | 用户信息符合规则且齐全 |
| 后置条件 | 系统修改该用户信息并持久化保存该用户数据 |
| User.modifyCredit | 语法 | Public ResultMessage modifyCredit (float creditNum) |
| 前置条件 | 网站营销人员为客户信用充值 |
| 后置条件 | 系统修改该客户信用值并持久化保存该用户数据 |
| User.getLoginInfo | 语法 | Public ResultMessage getLoginInfo(String userID，UserType userType) |
| 前置条件 | userID.Length=对应用户类型的ID长度 |
| 后置条件 | 系统返回与该用户ID对应的密码  如果ID不存在，则系统返回该ID对应用户不存在 |
|  | | |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| guestDataService.  add(GuestPO newGuestPO) | 根据提供的newGuestPO增加客户持久化对象 | |
| guestDataService.  getSingle (String userID) | 根据userID查找客户持久化对象 | |
| guestDataService.modify  (GuestPO guestPO) | 修改单一客户持久化对象 | |
| guestDataService.modifyMember  (MemberPO memberPO) | 修改单一会员持久化对象 | |
| guestDataService.init() | 清空所有客户持久化对象 | |
| hotelWorkerDataService.  add(HotelWorkerPO newHotelWorkerPO) | 根据提供的newHotelWorkerPO增加酒店工作人员持久化对象 | |
| hotelWorkerDataService.  getSingle (String userID) | 根据userID查找酒店工作人员持久化对象 | |
| hotelWorkerDataService.modify  (HotelPO hotelPO) | 修改单一酒店工作人员持久化对象 | |
| hotelWorkerDataService.init() | 清空所有酒店工作人员持久化对象 | |
| webMarkterDataService.  add(WebMarkterPO newWebMarkterPO) | 根据提供的newWebMarkterPO增加网站营销人员持久化对象 | |
| webMarkterDataService.  getSingle (String userID) | 根据userID查找网站营销人员持久化对象 | |
| webMarkterDataService.modify  (WebMarkterPO webMarkterPO) | 修改单一网站营销人员持久化对象 | |
| webMarkterDataService.init() | 清空所有网站营销人员持久化对象 | |
| webManagerDataService.  add(WebManagerPO newWebManagerPO) | 根据提供的newWebManagerPO增加网站管理人员持久化对象 | |
| webManagerDataService.  getSingle (String userID) | 根据userID查找网站管理人员持久化对象 | |
| webManagerDataService.modify  (WebManagerPO webManagerPO) | 修改单一网站管理人员持久化对象 | |
| webManagerDataService.init() | 清空所有网站管理人员持久化对象 | |
| Hotel.add | 新增该酒店并持久化保存该酒店的数据 | |

##### memberBL模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Memeber.add | 语法 | Public ResultMessage add(MemberVO memberVO) |
| 前置条件 | 会员信息符合规则且齐全 |
| 后置条件 | 系统根据已指定会员类型持久化为该客户增加会员数据 |
| Member.modify | 语法 | Public ResultMessage modify(MemberVO memberVO) |
| 前置条件 | 会员信息符合规则且齐全 |
| 后置条件 | 系统修改该会员并持久化保存该客户的会员数据 |
| Member.getMemberInfo | 语法 | Public MemberVO getMemberInfo(String userID,MemberType memberType) |
| 前置条件 | userID.Length=对应用户类型的ID长度 |
| 后置条件 | 系统返回与该userID对应的指定会员类型 |
| Member.isMember | 语法 | public boolean isMember(String userID，MemberType memberType) |
| 前置条件 | userID.Length=对应用户类型的ID长度 |
| 后置条件 | 返回该userID的用户是否为会员 |
| Member.getMemberType | 语法 | public MemberType getMemberType(String userID, MemberType memberType) |
| 前置条件 | userID.Length=对应用户类型的ID长度 |
| 后置条件 | 系统返回该userID的会员类型 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| GuestDataService.  ModifyMember(MemberPO memberPO) | 根据提供的memberPO修改单一持久化客户对象的信用值 | |
| GuestDataService.  getSingle (String userID) | 根据ID查找客户持久化对象 | |