1. **user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) |
| 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，UserType userType) |
| 前置条件 | 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) |
| 前置条件 | 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 (String gusetID, double 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.  getSingleGuest (String guestID) | 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.getAllGuest() | 获取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 HotelkWorker(String hotelWorkerID) | 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.getAllHotelWorker() | 获取所有酒店工作人员持久化对象 | |
| webMarketerDataService.  add(WebMarketerPO newWebMarketerPO) | 根据提供的newWebMarkterPO增加网站营销人员持久化对象 | |
| webMarkterDataService.  getSingleWebMarketer (String webMarketerID) | 根据userID查找网站营销人员持久化对象 | |
| webMarkterDataService.modify(WebMarketerPO webMarketerPO) | 修改单一网站营销人员持久化对象 | |
| webMarketerDataService.getAllWebMarketer() | 获取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.  getSingleWebManager (String webManagerID) | 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.getAllWebManager() | 获取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 | |

1. **market模块的接口规范**

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| market.getMemberFormulation | 语法 | Public List<MarketVO> getMemberFormulation () |
| 前置条件 | 无 |
| 后置条件 | 返回现有的会员等级制度信息 |
| market.setMemberFormulation | 语法 | Public ResultMessage setMemberFormulation (List<MarketVO> marketVOList) |
| 前置条件 | 网站营销人员输入会员等级制度 |
| 后置条件 | 系统记录此会员等级制度信息 |
| market.getLevel | 语法 | Public int getLevel （String userID) |
| 前置条件 | userID.Length=对应用户类型的ID长度 |
| 后置条件 | 系统根据客户当前信用值计算并返回当前会员等级 |
| market.getLevelName | 语法 | Public String getLevel （String userID) |
| 前置条件 | userID.Length=对应用户类型的ID长度 |
|  | 后置条件 | 系统根据客户当前信用值计算并返回当前会员等级的名字 |
| market.getMemberDiscout | 语法 | Public double getMemeberDiscout（String userID) |
| 前置条件 | userID.Length=对应用户类型的ID长度 |
| 后置条件 | 系统根据客户当前信用值计算用户当前会员等级，根据等级返回该用户享受的等级折扣 |
| 需要的接口（需接口） | | |
| 服务名 | **服务内容** | |
| marketDataService.getMemberFormulation（） | 返回现有的会员等级制度信息 | |
| marketDataService.setMemberFormulation(List<MarketVO> marketVOList) | 修改会员等级制度信息持久化对象，系统返回是否成功记录 | |