逻辑层（userbl）模块的接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| UserService.login | 语法 | public boolean login(long ID, String password) |
| 前置条件 | ID和password符合输入规范 |
| 后置条件 | 查找是否存在相应的ID，根据输入的password返回输入的结果 |
| UserInfoService.getUserInfo | 语法 | public UserVO getUserInfo(long ID) |
| 前置条件 | 用户或网站管理人员查看用户信息 |
| 后置条件 | 返回该用户信息 |
| UserInfoService.modifyUserInfo | 语法 | public boolean modifyUserInfo(UserVO user) |
| 前置条件 | 客户或网站管理人员修改用户信息 |
| 后置条件 | 返回是否修改成功 |
| SignVipService.signRegularVip | 语法 | public boolean signRegularVip(UserVO regularVip) |
| 前置条件 | 客户申请普通会员 |
| 后置条件 | 返回是否申请成功 |
| SignVipService.signEnterpriseVip | 语法 | public boolean signEnterpriseVip(UserVO EnterpriseVip) |
| 前置条件 | 客户申请企业会员 |
| 后置条件 | 返回是否申请成功 |
| ClientCreditService.queryCredit | 语法 | public UserVO queryCredit(long ID) |
| 前置条件 | ID符合输入规范 |
| 后置条件 | 返回该客户信用值及信用值变化记录 |
| ClientCreditService.addCreditValue | 语法 | public boolean addCreditValue(long ID,int creditAdded) |
| 前置条件 | ID和creditAdded符合输入规范 |
| 后置条件 | 返回是否充值成功 |
| UserService.add | 语法 | public boolean add(UserVO user) |
| 前置条件 | 用户注册账号或网站管理人员添加用户账号 |
| 后置条件 | 返回是否注册或添加成功 |
| UserService.del | 语法 | public boolean del(UserVO user) |
| 前置条件 | 用户注销账号或网站管理人员删除用户账号 |
| 后置条件 | 返回是否注销或删除成功 |
| UserService.insert | 语法 | public boolean insert(UserVO staff) |
| 前置条件 | 网站管理人员添加酒店工作人员账号 |
| 后置条件 | 返回是否注册或添加成功 |
| ClientCreditService.getCreditValue | 语法 | public int getCreditValue(long ID) |
| 前置条件 | ID符合输入规范 |
| 后置条件 | 返回客户的当前信用值 |
| ClientCreditService.changeCreditValue | 语法 | public boolean changeCreditValue(long ID,int num) |
| 前置条件 | ID和num符合输入规范 |
| 后置条件 | 返回是否更改信用值成功 |
| UserInfoService.getRegularVipInfo | 语法 | public UserVO getGegularVipInfo(long ID) |
| 前置条件 | ID符合输入规范 |
| 后置条件 | 返回普通会员的信息 |
| UserInfoService.getEnterpriseVipInfo | 语法 | public UserVO getEnterpriseInfo(long ID) |
| 前置条件 | ID符合输入规范 |
| 后置条件 | 返回企业会员的信息 |
| **需要的服务（需接口）** | | |
| 服务名 | 服务 | |
| userDAO.getUserInfo(long ID) | 得到ID对应的用户信息 | |
| userDAO.queryCredit（long ID) | 得到ID对应的用户信用值及信用值变化记录 | |
| UserDAO.getCreditValue(long ID) | 得到ID对应的用户的当前信用值 | |
| userDAO.insert(UserPO po) | 插入单一持久化对象 | |
| userDAO.del(UserPO po) | 删除单一持久化对象 | |
| userDAO.update(UserPO po) | 更新单一持久化对象 | |
| userDAO.finish() | 结束持久化数据库的使用 | |
| DatabaseFactory.getUserDatabase | 得到user数据库的服务的引用 | |

数据层（userDAO）模块的接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| userDAO.getUserInfo | 语法 | public UserPO getUserInfo(long ID) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 查找是否存在相应的ID，根据输入的password返回输入的结果 |
| userDAO.queryCredit | 语法 | public UserPO queryCredit(long ID) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 查找是否存在相应的ID，存在则返回相应的PO |
| userDAO.getCreditValue | 语法 | public int getCreditValue(long ID) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 查找是否存在相应的ID，存在则返回相应的信用值 |
| userDAO.insert | 语法 | public void insert(UserPO po) throws RemoteException |
| 前置条件 | 在数据库中不存在同样ID的po |
| 后置条件 | 在数据库中增加一个po记录 |
| userDAO.del | 语法 | public void delete(UserPO po) throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 删除一个po |
| userDAO.update | 语法 | public void update(UserPO 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 |
| userDAO.finish | 语法 | public void finish() throws RemoteException |
| 前置条件 | 无 |
| 后置条件 | 结束持久化数据库的使用 |