模块：Strategybl 职责：负责对应于制定促销策略界面及显示策略列表和策略信息界面所需要的服务

接口规范：

|  |  |  |
| --- | --- | --- |
| 提供的服务(供接口) | | |
| Strategy.creat | 语法 | Public resultMessage creat(StrategyVO vo); |
| 前置条件 | 工作人员已登录和授权 |
| 后置条件 | 返回相应结果，包括成功、失败和失败原因（如时间不匹配） |
| Strategy.delete | 语法 | Public void delete(long strategyID); |
| 前置条件 | 工作人员已登录和授权 |
| 后置条件 | 删除对应数据库里的信息 |
| Strategy.change | 语法 | Public resultMessage change(StrategyVO vo); |
| 前置条件 | 工作人员已登录和授权 |
| 后置条件 | 返回相应结果，包括成功、失败和失败原因（如时间不匹配），如果成功则更改对应数据库里的信息 |
| Strategy.getInfo | 语法 | Public SrategyVO(long strategyID); |
| 前置条件 | 工作人员已登录和授权 |
| 后置条件 | 返回对应数据库里的信息 |
| Strategy.getStrategyList | 语法 | Public List<StrategyVO>() |
| 前置条件 | 工作人员已登录和授权 |
| 后置条件 | 返回策略信息 |
| 需要的服务(需接口) | | |
| DatabaseFactory.getStrategyDatabase | 得到Strategy数据库服务的引用 | |
| Hotel.getInfo | 得到酒店信息 | |
| Member.getInfo(long memberID) | 得到对应的会员信息 | |
| StrategyDataService.creat(StrategyPO po) | 在数据库里创建一条策略记录 | |
| StrategyDataService.delete(StrategyPO po) | 在数据库里删除一条策略记录 | |
| StrategyDataService.change(StategyPO po) | 在数据库里更改一条策略记录 | |
| StrategyDataService.getStrategys() | 返回数据库里的所有策略记录 | |

模块:Creditbl职责：负责对应于信用充值和查看记录等界面所需要的服务

接口规范：

|  |  |  |
| --- | --- | --- |
| 提供的服务(供接口) | | |
| Credit.getInfo | 语法 | Public List<CreditVO> getInfo(long userID); |
| 前置条件 | 客户已登录 |
| 后置条件 | 返回信用记录 |
| Credit.recharge | 语法 | Public ResultMessage recharge(int value); |
| 前置条件 | 网站营销人员已登录和授权 |
| 后置条件 | 存储并返回消息（成功或失败） |
| Credit.add | 语法 | Public void add(userID id,int value); |
| 前置条件 | 订单已执行或者酒店工作人员补登记订单或者网站工作人员撤销异常订单 |
| 后置条件 | 信用值增加并存储相应的结果 |
| Credit.decrease | 语法 | Public void decrease(userID id,int value); |
| 前置条件 | 订单由未执行状态变为异常状态 |
| 后置条件 | 信用值减少并存储相应的结果 |
| 需要的服务(需接口) | | |
| DatabaseFactory.getCreditDatabase | 得到Credit数据库服务的引用 | |
| CreditDataService.add(long userID,int value) | 增加数据库中的相应信用值 | |
| CreditDataService.decrease(long userID,int value) | 减少数据库中的相应信用值 | |
| CreditDataService.insert(CreditPO po) | 在数据库中插入CreditPO对象 | |

DATA：StrategyDataService

|  |  |  |
| --- | --- | --- |
| 提供的接口 | | |
| StrategyDataService.creat | 语法 | Public void creat(StrategyPO po) throws RemoteException |
| 前置条件 | 同样ID的po在数据库中不存在 |
| 后置条件 | 在数据库中创建一条PO记录 |
| StrategyDataService.delete | 语法 | Public void delete(long strategyID) throws RemoteException |
| 前置条件 | 同样ID的PO在数据库中存在 |
| 后置条件 | 在数据库中删除这条PO数据 |
| StrategyDataService.change | 语法 | Public void change(StrategyPO po) throws RemoteException |
| 前置条件 | 同样ID的PO在数据库中存在 |
| 后置条件 | 更改数据库中的相应PO数据 |
| StrategyDataService.getStrategys | 语法 | Public StrategyPO getStrategys() throws RemoteException |
| 前置条件 | 数据库中存在PO对象 |
| 后置条件 | 返回数据库中所有数据 |

CreditDataService

|  |  |  |
| --- | --- | --- |
| 提供的接口 | | |
| CreditDataService.add | 语法 | Public void add(UserID id,int value) throws RemoteException |
| 前置条件 | 同样ID的PO在数据库中存在 |
| 后置条件 | 增加数据库中的相应信用值 |
| CreditDataService.decrease | 语法 | Public void decrease(UserID id,int value) throws RemoteException |
| 前置条件 | 同样ID的PO在数据库中存在 |
| 后置条件 | 减少数据库中的相应信用值 |
| CreditDataService.insert | 语法 | Public void insert(CreditPO po)throws RemoteException |
| 前置条件 | 同样ID的PO在数据库中不存在 |
| 后置条件 | 增加一条PO记录 |

StrategyPO 定义：

Public class StrategyPO implement Serializable{

Long id;

Date date;

Long memberID;

Date memberBirth;

…

构造函数

Get/set

}

CreditPO 定义：

Public class CreditPO implement Serializable{

Long id;

Int value;

Long userID;

…

构造函数

Get/set

}