**产品服务模块设计**

**修订历史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本号** | **作者** | **修订章节** | **修订原因** | **修订日期** |
| 1.0 | 陈国利 | 创建 |  | 2014-09-11 |

**目 录**

[1 概述 4](#_Toc398273989)

[1.1 名词解释 4](#_Toc398273990)

[1.2 参考资料 4](#_Toc398273991)

[2 详细设计 4](#_Toc398273992)

[2.1 流程概述 4](#_Toc398273993)

[2.2 产品服务模块流程图 5](#_Toc398273994)

[2.2.1 处理产品申购买卖模块请求（子模块1） 5](#_Toc398273995)

[2.2.2 处理账户接口模块反馈（子模块2） 7](#_Toc398273996)

[2.2.3 处理交易接口模块反馈（子模块3） 8](#_Toc398273997)

[2.2.4 信息查询（子模块4） 9](#_Toc398273998)

[2.3 系统容错设计 9](#_Toc398273999)

[2.3.1 容错方案 9](#_Toc398274000)

[2.3.2 出错信息 9](#_Toc398274001)

[2.3.3 补救措施 10](#_Toc398274002)

[3 接口设计 10](#_Toc398274003)

[3.1 外部接口设计 10](#_Toc398274004)

[3.1.1 处理产品申购买卖模块请求 10](#_Toc398274005)

[3.1.2 处理账户接口模块反馈 10](#_Toc398274006)

[3.1.3 处理交易接口模块反馈（脚本） 11](#_Toc398274007)

[3.1.4 信息查询 11](#_Toc398274008)

[3.2 内部接口设计 11](#_Toc398274009)

[3.2.1 处理产品申购买卖模块请求 11](#_Toc398274010)

[3.2.2 处理账户接口模块反馈 11](#_Toc398274011)

[3.2.3 处理交易接口模块反馈（脚本） 12](#_Toc398274012)

[3.2.4 信息查询 13](#_Toc398274013)

[3.3 依赖外部接口设计 13](#_Toc398274014)

[3.3.1 处理产品申购买卖模块请求 13](#_Toc398274015)

[3.3.2 处理账户接口模块反馈 13](#_Toc398274016)

[3.3.3 处理交易接口模块反馈（脚本） 13](#_Toc398274017)

[3.3.4 信息查询 14](#_Toc398274018)

[4 非功能需求的设计 14](#_Toc398274019)

[5 模块间服务依赖 14](#_Toc398274020)

[5.1 依赖的外部系统 14](#_Toc398274021)

[5.1.1 环境依赖 14](#_Toc398274022)

[5.1.2 功能依赖 14](#_Toc398274023)

[5.2 被外部系统所依赖的 15](#_Toc398274024)

# 概述

## 名词解释

|  |  |  |  |
| --- | --- | --- | --- |
| **名词** | **英文名** | **说明** | **举例** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 参考资料

模块框架构略图

# 详细设计

## 流程概述

产品服务模块包含三个主要流程：处理产品申购买卖模块请求（子模块1）、处理账户接口模块反馈（子模块2）、处理交易接口模块反馈（子模块3）、信息查询（子模块4）。

1. 子模块1接收产品申购买卖模块的申购请求。
   1. 申购：创建订单，向账户接口模块发出交易申请。
   2. 卖出：向交易接口模块发出卖出请求。
2. 子模块2接收账户接口模块的交易结果通知，根据通知结果做相应的后续处理。
   1. 成功：检测开户条件，更新plan信息，向交易接口模块发出开户/买入的交易请求。
   2. 失败：更新plan为失败，退出。
3. 子模块3接收子模块2中成功的plan信息（id），检测开户/买入的结果并做相应的后续处理。
   1. 成功：更新plan为成功，向账户接口模块发起确认交易请求。
   2. 失败：更新plan为失败，向账户接口模块发起取消交易请求。
   3. 未知：等待一段时间后重新检测。

4、子模块4信息查询。信息查询包括订单基本信息、订单实时信息、锁定页信息（第三屏）、结算页信息（第四屏）。

## 产品服务模块流程图

### 产品申购买卖模块

#### 申购



#### 申购支付和开户



#### 卖出



### 账户接口模块反馈

#### 下订支付反馈



#### 确认/取消支付反馈

 

### 交易接口模块反馈



### 手动取消交易



### 信息查询

 

 

## 系统容错设计

### 容错方案

在事务执行过程中，对可能出错的地方引入补偿接入点，如果长时间未接到通知就再次发送请求，自动完成再一次的执行。

### 出错信息

补偿机制根据订单当前所处状态进行相应的补偿操作。

1. 下订支付失败。
   1. 补偿点1.1 - 订单长时间未做下订支付。
   2. 补偿点1.2 - 长时间未收到下订支付结果。
   3. 补偿点1.3 - 订单下订支付失败（异常）。
2. 开户失败。
   1. 补偿点2.1 - 订单下订支付成功，长时间未做产品开户操作。
   2. 补偿点2.2 - 产品开户成功，长时间未收到实盘开户结果。
   3. 补偿点2.3 - 产品开户失败，长时间未做取消交易请求。
3. 确认/取消支付失败。
   1. 补偿点3.1 - 长时间未收到确认交易支付结果。
   2. 补偿点3.2 - 确认交易支付失败（异常）。
   3. 补偿点3.3 - 长时间未收到取消交易支付结果。
   4. 补偿点3.4 - 取消交易支付失败（异常）。

### 补偿方案流程



# 接口设计

## 外部接口设计

### 产品申购买卖模块

/\*\*

\* 接收产品发出的申购请求

\* **@param** array $params 创建订单所需参数数组

\* eg: uid,rule\_type,stock\_code,stock\_name,stock\_amount,

\* zs\_types,buy\_type,price,account\_type,phj\_config

\* **@param** string $proName 产品类型

\* **@return** boolean 账户接口接收结果

\*/

**public function** applyRequest($params, $proName) {}

/\*\*

\* 接收产品发出的卖出请求

\* **@param** array $params 卖出所需参数数组

\* eg: stocker\_user\_name,price,must\_deal

\* **@param** string $proName 产品类型

\* **@return** boolean 交易接口接收结果

\*/

**public function** sellRequest() {params, $proName}

### 账户接口模块反馈

/\*\*

\* 账户接口模块处理后回调产品服务模块处理

\* **@param** json-string $json 账户接口处理结果

\* orders:订单号 status:处理结果 code:错误代码

\* **@return** json-string 产品服务处理结果

\*/

**public function** accountCallback($json=’’){}

### 交易接口模块反馈（脚本）

### 信息查询

/\*\*

\* 查询订单基本信息

\* **@param** string $idStr 订单id集合字符串

\* **@param** int $uid 用户id

\* **@param** string $proName 产品类型

\* **@return** array 结果数组

\*/

**public function** getOrderInfoRequest($idStr, $uid, $proName){}

/\*\*

\* 获取订单的实时信息

\* **@param** string $usernames 订单username数组拼接的字符串

\* **@param** int $uid 用户id

\* **@param** string $proName 产品类型

\* **@return** array 结果数组

\*/

**public function** getRealtimeRecords($usernames, $uid, $proName){}

/\*\*

\* 获取锁定页信息（第三屏）

\* **@param** string $proName 产品类型

\* **@param** int $uid 用户id

\* **@param** int $buyType 买入方向

\* **@param** int $startId 查询起始id

\* **@param** int $limit 查询记录数

\* **@return** array 结果数组

\*/

**public function** getTradeOrders($proName, $uid, $buyType, $startId, $limit){}

/\*\*

\* 获取结算页信息（第四屏）

\* **@param** string $proName 产品类型

\* **@param** int $uid 用户id

\* **@param** int $buyType 买入方向

\* **@param** int $startId 查询起始id

\* **@param** int $limit 查询记录数

\* **@return** array 结果数组

\*/

**public function** getCheckoutOrders($proName, $uid, $buyType, $startId, $limit){}

## 内部接口设计

### 产品申购买卖模块

/\*\*

\* 创建订单

\* **@param** array $params 创建订单所需参数数组

\* eg: uid,rule\_type,stock\_code,stock\_name,stock\_amount,

\* zs\_types,buy\_type,price,account\_type,phj\_config

\* **@param** string $proName 产品类型

\* **@return** boolean 创建结果

\*/

**public function** createPlan($params, $proName) {}

/\*\*

\* 更新订单字段（单一）

\* **@param** int $objectId 订单id

\* **@param** int $field 目标字段（注意限定field范围）

\* **@param** int $value 目标字段值

\* **@param** string $proName 产品类型

\* **@return** boolean 更新结果

\*/

**public function** updatePlanField($objectId, $field, $value, $proName) {}

/\*\*

\* 创建订单状态机（初始状态为创建成功）

\* **@param** int $objectId 订单id

\* **@param** string $proName 产品类型

\* **@return** boolean 创建结果

\*/

**public function** createPlanState ($objectId, $proName) {}

/\*\*

\* 更新状态机

\* **@param** int $objectId 订单id

\* **@param** string $state 目标状态

\* **@param** string $proName 产品类型

\* **@return** boolean 更新结果

\*/

**public function** updatePlanState ($objectId, $state, $proName) {}

### 账户接口模块反馈

/\*\*

\* 是否满足开户条件检测

\* **@param** string $orderIdStr 目标订单id数组拼接的字符串

\* **@param** string $proName 产品类型

\* **@return** boolean 检测结果

\*/

**public function** checkCreateCondition($orderIdStr, $proName){}

/\*\*

\* 开户接口

\* **@param** int $uid 用户id

\* **@param** int $objectId 订单id

\* **@param** string $username

\* **@param** int $stockCode 股票代码

\* **@param** int $stockAmount 买入数量

\* **@param** string $proName 产品类型

\* **@return** int 开户结果标识成功/失败/重复

\*/

**public function** createStocker($uid, $id, $username, $stockCode, $stockAmount, $proName){}

### 交易接口模块反馈（脚本）

/\*\*

\* 检测订单开户结果

\* **@param** int $objectId 订单id

\* **@param** string $proName 产品类型

\* **@return** int 检测结果标识

\*/

**public function** check($objectId, $proName){}

/\*\*

\* 开户成功处理流程

\* **@param** int $objectId 订单id

\* **@param** array $params 处理所需其他参数数组

\* **@param** string $proName 产品类型

\* **@return** void

\*/

**public function** successHandler($objectId, $params, $proName){}

/\*\*

\* 开户失败处理流程

\* **@param** int $objectId 订单id

\* **@param** array $params 处理所需其他参数数组

\* **@param** string $proName 产品类型

\* **@return** void

\*/

**public function** failHandler($objectId, $params, $proName){}

/\*\*

\* 更新plan信息组合

\* **@param** int $objectId

\* **@param** array $params 待更新数据键值对

\* **@param** string $proName 产品类型

\* **@return** boolean 更新结果

\*/

**public function** updatePlan($objectId, $params, $proName){}

### 信息查询

/\*\*

\* 查询订单基本信息

\* **@param** string $idStr 订单id数据拼接的字符串

\* **@param** int $uid 用户id

\* **@param** string $proName 产品类型

\* **@return** array 结果数组

\*/

**public function** getOrderInfo($idStr, $uid, $proName){}

### 补偿

/\*\*

\* 获取需要补偿订单

\* **@return** void

\*/

**public function** getCompensionOrder(){}

## 依赖外部接口设计

### 处理产品申购买卖模块请求

1. 账户接口模块交易请求接口（由账户接口模块提供）
2. 交易接口模块卖出请求接口（由交易接口模块提供）

### 处理账户接口模块反馈

1. 获取操盘服务组名（由交易接口模块提供）
2. 交易接口模块开户/买入接口（由交易接口模块提供）
3. 账户接口模块取消交易请求接口（由账户接口模块提供）

### 处理交易接口模块反馈（脚本）

1. 查询订单交易状态（由交易接口模块提供）
2. 账户接口模块确认交易请求接口（由账户接口模块提供）
3. 账户接口模块取消交易请求接口（由账户接口模块提供）

### 信息查询

1、查询实时信息（由交易接口模块提供）

# 非功能需求的设计

1. 差错率。保证申购买入过程的成功率和响应速度。成功率保证在95%以上。
2. 性能。保证大并发量下同一请求的幂等性和承受力。

要约单的交易量比较大，经查询当前实际数据量最高可达到120单每分钟。按照业务扩展需求，数据量按扩大10倍计算，同时金额减少造成数据量再扩大10倍计算。120\*10\*10=12000单/分钟。保证系统的性能在此基础的1.5倍则需要12000\*1.5=18000单/分钟。

1. 补偿速度。保证异常情况下的补偿机制处理速度。补偿机制启动时间根据不同模块功能确定。

# 数据库设计

## 数据库设计

### 产品订单

### 产品订单状态机

## 表说明

### 产品订单

记录产品订单的相关信息，包括金额，股票，数量等。

### 订单状态机：记录订单操作过程中所处的状态。

1. 订单创建。1001成功。
2. 下订交易（申购冻结支付）。2000下订中，2001成功，2002失败。
3. 产品开户（取组名/开户请求）。3001成功。3002失败。
4. 实盘开户。4001成功，4002失败。
5. 确认交易（申购解冻转账）。5000确认中，5001成功，5002失败。
6. 取消交易（申购取消解冻）。6000取消中，6001成功，6002失败。
7. 结算交易。7000结算中，7001成功，7002失败。（清结算模块）
8. 重复结算交易。8000结算中，8001成功，8002失败。（清结算模块）

# 模块间服务依赖

## 依赖的外部系统

### 环境依赖

1. Php开发语言5.4以上版本。
2. Yaf开发框架支持。
3. Mysql支持。
4. Gearman job分发服务端。
5. Ngnix或apache的web支持。

### 功能依赖

产品服务模块服务于产品、交易、账户三个模块间。依赖的外部模块有：

1. 账户接口模块，。产品服务模块需要请求账户接口模块做交易处理，并获取交易结果。
2. 交易接口模块。
   1. 向交易接口模块发出开户/买入请求，并获取结果。
   2. 向交易接口模块发出查询请求，并获取结果。

## 被外部系统所依赖的

产品服务模块为产品申购买卖模块、账户接口模块提供服务。

1. 供产品申购模块。
   1. 提供产品的申购买卖接口。
   2. 提供信息查询接口。
2. 供账户接口模块。
   1. 提供账户接口反馈回调接口。