|  |  |
| --- | --- |
| 文档编号 | V1.0 |
| 文档负责人 |  |
| 文档名称 | XXX设计 |

概要设计

目录

[目录 2](#_Toc416429915)

[第1章 文档记录 4](#_Toc416429916)

[第2章 引言 5](#_Toc416429917)

[2.1 目的 5](#_Toc416429918)

[2.2 目标读者 5](#_Toc416429919)

[2.3 名词解释 5](#_Toc416429920)

[2.4 文档约定 5](#_Toc416429921)

[第3章 业务流程 6](#_Toc416429922)

[3.1 用户垫资使用信息同步流程 6](#_Toc416429923)

[3.2 对账单上传 6](#_Toc416429924)

[3.3 对账结果下载 6](#_Toc416429925)

[第4章 数据库设计 7](#_Toc416429926)

[4.1 商户信息表 7](#_Toc416429927)

[4.2 XXXX 8](#_Toc416429928)

[第5章 接口设计 10](#_Toc416429929)

[5.1 account\_server（账号管理服务） 10](#_Toc416429930)

[5.1.1 商户信息查询 10](#_Toc416429931)

[5.1.2 商户RSA密钥 10](#_Toc416429932)

[5.1.3 生成商户签名 11](#_Toc416429933)

[5.1.4 验证商户签名 11](#_Toc416429934)

[5.2 12](#_Toc416429935)

[第6章 安全设计 13](#_Toc416429936)

# 文档记录

# 引言

## 目的

## 目标读者

## 名词解释

## 文档约定

是否必填项说明

|  |  |  |
| --- | --- | --- |
| 选项 | 全称 | 说明 |
| R | Required | 必需 |
| C | Conditional | 有条件必填 |
| O | Optional | 可选项 |

# 业务流程

## 用户垫资使用信息同步流程



## XXX

## XXX

# 数据库设计

## 商户信息表

见《垫资系统数据库设计.doc》finance\_db.t\_sp\_config

## 用户注册信息表

见《垫资系统数据库设计.doc》finance\_db.t\_user\_info

# 接口设计

## 签名管理（使用已有模块）

提供签名相关接口

### 生成商户签名

接口说明

xxxx/genSpSign

输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| spid | string(16) | R | 商户号 |
| sign\_type | string(16) | R | 签名类型  MD5  RSA |
| text | string(1024) | R | 需要签名的内容 |

输出参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| sign | string | R |  |

内部逻辑

1. 检必填参数是否为空
2. 查询商户信息，判断商户是否存在，是否有相应的权限
3. sign\_type=RSA
4. 根据商户号和本地RSA密钥文件存放路径获得商户的RSA私钥
5. 将text内容使用私钥进行RSA加密
6. sign\_type=MD5
7. 将text内容拼接上“|$sp\_key”
8. 将生成的字串计算MD5值
9. 返回生成的签名结果

### 验证商户签名

接口说明

xxxx/checkSpSign

输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| spid | string | R | 商户号 |
| sign\_type | string | R | 签名类型  MD5  RSA |
| text | string | R | 需要签名的内容 |
| sp\_sign | string | R | 商户生成的签名 |

输出参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| check\_result | bool | R | 检查结果 |

内部逻辑

1. 检必填参数是否为空
2. 查询商户信息，判断商户是否存在，是否有相应的权限
3. sign\_type=RSA
4. 根据商户号和本地RSA密钥文件存放路径获得商户的RSA私钥
5. 将sp\_sign内容使用私钥进行RSA解密
6. 判断解密后的内容与text是否相同
7. sign\_type=MD5
8. 将text内容拼接上“|$sp\_key”
9. 将生成的字串计算MD5值
10. 判断生成的MD5与sp\_sign是否相同
11. 返回签名验证结果

## bill\_no（订单号生成）

### 生成订单号

#### 接口说明

xxxx/genbillno

#### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| list\_type | int | R | 类型  0仅生成10位单号  101转账单号  102提现单号  103充值单号  104冻结单号  105解冻单号 |
| appid | int | R | 应用ID |
| spid | string(16) | C | 商户号 |
|  |  |  |  |

#### 输出参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| billno | string | R | 返回生成提单号 |

#### 内部逻辑

原则：

9位序号自增生成：直接使用int型变量递增（6亿个单号），范围[0, 599999999]（最高位5以上的用于特殊场景使用，例如单号业务单号因appid配置错误出现重复时，修改最高位来保证单号不同不用批量重新生成单号），最大值599999999，超过最大值则从0开始。每个appid的自增变量不同，互不干扰。

每次生成1000个，存放在缓存在数组中，当使用完之后重新生成1000个到数组中。每生成一次将生成的最大值写入文件（每个appid一个文件，文件在读取和更新时需要对文件进行加锁），只有写入文件成功后生成的数字才能可用。

当服务或机器重启后从文件中读上取一次已使用的数字的最大值，从该值开始使用。

10位序号的规则：1位机器号+9位自增的序号，机器号从配置文件中读取，每台提供单号服务的机器机号不能相同，最多支持10台机器。

返回的单号规则：a、list\_type!=0，类3务类型+spid(10位)+8位日期+10位生成的序号；b、list\_type=0，直接返回10位生成的序号

1. 检查输入参数是否合法

appid必须为配置中的appid

list\_type必须为配置的类型

1. java线程加锁appid对应的自增序号
2. 获取已生成的自增序号（最长9位），如果已生成的序号用完，则重新生成一批
3. 线程解锁appid对应的自增序号
4. 生成序号（1位机器号+9位自增序号）
5. 按规则生成单号
6. 返回生成的单号

## finance\_service

### 用户注册

#### 接口说明

xxxx/ insertUserInfo

#### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| userName | string | R | 用户名 |
| trueName | string | R | 用户真实姓名 |
| sex | int | R | 性别 |
| mobile | string | R | 手机号码 |
| email | string | R | 邮箱 |
| userPwd | string | R | 用户密码 |
| userPayPassword | string |  | 用户的支付操作密码 |
| phone | string |  | 固定电话，默认为空 |
| creType | int |  | 证件类型 1身份证 |
| creId | string |  | 证件号 |
| state | int |  | 用户状态 1正常2冻结3注销 |
| logicState | int |  | 物理逻辑状态（logical state）1有效2无效 |
| regIp | string |  | 用户注册时的IP |
| userSeed | string |  | 用户种子信息 |
| sign | string |  | 数据签名 |
| createTime | datetime |  | 创建时间 |
| modifyTime | datetime |  | 最后修改时间 |
| loginIp | string |  | 用户最后登陆IP |
| loginTime | datetime |  | 用户最后登陆时间 |
| pwdModTime | datetime |  | 用户密码最后修改时间 |
| pwdModIp | string |  | 用户密码最后修改IP |
| payPwdModTime | datetime |  | 用户支付密码最后修改时间 |
| payPwdModIp | string |  | 用户支付密码最后修改IP |
| question1 | string |  | 用户问题1 |
| answer1 | string |  | 用户答案1 |
| memo | string |  | 备注 |

#### 输出参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| result | int |  | 返回结果,成功返回userId，失败返回0 |

#### 内部逻辑

保存用户信息

## finance\_web

### 用户注册登记

#### 接口说明

xxxx/Reg\_user

#### 输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| userName | string | R | 用户名 |
| trueName | string | R | 用户真实姓名 |
| sex | int | R | 性别 |
| mobile | string | R | 手机号码 |
| email | string | R | 邮箱 |
| userPwd | string | R | 用户密码 |

#### 输出参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| retcode | string |  | 结果编码，成功为0，其它为错误码 |
| retmsg | string |  | 结果消息，成功为OK，其他为错误信息 |

#### 内部逻辑

1.检测参数

2.检测用户是否已经注册过

3.完善用户信息

4.保存用户信息

# 安全设计

1. 敏感信息需要加密保存
2. 密码等信息需要不可逆hash保存