# 数据库设计

## 用户注册信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名** | 用户注册信息表 | | **描述** | 记录用户注册的信息（请用navicat随便添加一条数据上去，确保Fuid=100000000即可。） |
| User\_db.t\_user\_info | |
| **主键** | | Fuid | | |
| **索引** | | 唯一索引Fuser\_id  Fmobile  Femail  Fmodify\_time | | |
| **字段名** | | **类型** | | **说明** |
| Fuid | | bigint(auto\_increment) | | 用户内部ID, 从100000000起增 |
| Fuser\_id | | varchar(128) | | 注册的用户名 |
| Ftrue\_name | | varchar(128) | | 用户真实姓名 |
| Fopenid | | varchar(64) | | 第三方的帐号,如微信的openid或QQ号 |
| Fchannel\_id | | varchar(64) | | 渠道信息  1 PC注册  2 IOS APP注册  3 Android APP注册 |
| Fpwd | | varchar(64) | | 用户注册时的登陆密码（hash后保存） |
| Fpay\_pwd | | varchar(64) | | 用户的支付操作密码（hash后保存） |
| Fsex | | smallint | | 性别  1男  2 女 |
| Fmobile | | varchar(32) | | 手机号（加密保存） |
| Femail | | varchar(64) | | 邮箱地址（加密保存） |
| Fphone | | varchar(32) | | 固定电话，默认为空 |
| Fauth\_id | | bigint | | 实名认证ID |
| Frisk\_type | | smallint | | 用户风险评测级别 |
| Frisk\_time | | datetime | | 用户风险评测时间 |
| Fstate | | smallint | | 用户状态  1正常  2冻结  3注销 |
| Flstate | | smallint | | 物理逻辑状态（logical state）  1有效  2无效 |
| Fip | | varchar(32) | | 用户注册时的IP |
| Fuser\_seed | | varchar(64) | | 用户签名的种子信息 |
| Fsign | | varchar(64) | | 数据签名 |
| Fcreate\_time | | datetime | | 创建时间 |
| Fmodify\_time | | datetime | | 最后修改时间 |
| Flogin\_ip | | varchar(32) | | 用户最后登陆IP |
| Flogin\_time | | datetime | | 用户最后登陆时间 |
| Fpwd\_mtime | | datetime | | 用户密码最后修改时间 |
| Fpwd\_ip | | varchar(32) | | 用户密码最后修改IP |
| Fpaypwd\_mtime | | datetime | | 用户支付密码最后修改时间 |
| Fpaypwd\_ip | | varchar(32) | | 用户支付密码最后修改IP |
| Fquestion1 | | varchar(255) | | 用户问题1（加密保存） |
| Fanswer1 | | varchar(255) | | 用户答案1（hash后保存） |
| Fmemo | | varchar(255) | | 备注字段 |

## 用户实名认证信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名** | 用户实名认证信息表 | | **描述** | 记录用户的实名认证信息 |
| user\_db.t\_user\_auth\_info | |
| **主键** | | Fpk\_id | | |
| **索引** | | 唯一索引Fcre\_id+ Fcre\_type+ Ftrue\_name  Fmodify\_time | | |
| **字段名** | | **类型** | | **说明** |
| Fpk\_id | | bigint(auto\_increment) | |  |
| Ftrue\_name | | varchar(128) | | 用户真实姓名 |
| Fsex | | smallint | | 性别  1男  2 女 |
| Fmobile | | varchar(32) | | 手机号（加密保存） |
| Fcre\_type | | smallint | | 证件类型  1身份证 |
| Fcre\_id | | varchar(255) | | 证件号，3DES加密保存  证件为身份证时，保存18位的身份证 |
| Fuid | | bigint | | 实名认证时用户ID |
| Fauth\_type | | smallint | | 实名认证方式  1公安部实名---1  2银行打款实名认证  3第三方支付实名认证 |
| Fstate | | smallint | | 状态  1实名  2认证  3实名认证 |
| Flstate | | smallint | | 物理逻辑状态（logical state）  1有效  2无效 |
| Fip | | varchar(32) | | 用户注册时的IP |
| Fsign | | varchar(64) | | 数据签名 |
| Fcreate\_time | | datetime | | 创建时间 |
| Fmodify\_time | | datetime | | 最后修改时间 |
| Fmemo | | varchar(255) | | 备注字段 |
| Fstandby1 | | int | | 保留字段（暂未启用） |
| Fstandby2 | | bigint | | 保留字段（暂未启用） |
| Fstandby3 | | varchar(64) | | 保留字段（暂未启用） |
| Fstandby4 | | varchar(255) | | 保留字段（暂未启用） |
| Fstandby5 | | datetime | | 保留字段（暂未启用） |
|  | |  | |  |

## 用户银行卡绑定表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表名** | 注册用户绑定银行卡信息表 | | **描述** | 记录用户绑定的银行卡信息  按uid分库分表 |
| user\_db\_xx.t\_bind\_card\_x | |
| **主键** | | Fpkid | | |
| **索引** | | 唯一索引Fbind\_no  Fuid+Fcard\_tail | | |
| **字段名** | | **类型** | | **说明** |
| Fpkid | | bigint(auto\_increment) | | 自增主键 |
| Fbind\_no | | varchar(64) | | 绑定序号，6位时间+5位序号+uid |
| Fprotocol\_no | | varchar(64) | | 绑定协议号 |
| Fuid | | long | | 用户UID |
| Fmobile | | varchar(32) | | 绑定的银行卡手机号（3DES加密保存） |
| Fcre\_type | | smaillint | | 银行对应的证件类型 |
| Fcre\_id | | varchar(64) | | 证件ID（3DES加密保存） |
| Fuser\_name | | varchar(64) | | 持卡人姓名 |
| Fcardid | | varchar(64) | | 卡号（3DES加密保存） |
| Fcard\_tail | | varchar(10) | | 卡尾号，后4位 |
| Fcard\_flag | | smallint | | 卡标记  1主卡  2副卡 |
| Fbind\_type | | int | | 绑定类型  1借记卡 |
| Fbank\_type | | int | | 银行类型 |
| Fbank\_name | | varchar(64) | | 银行名称/开户行名称 |
| Fbank\_area | | varchar(64) | | 开户地区，省级 |
| Fbank\_city | | varchar(64) | | 开户城市 |
| Fbind\_time | | datetime | | 绑定完成时间 |
| Funbind\_time | | datetime | | 解绑时间 |
| Fstate | | smallint | | 账户状态  1初始  2绑定成功  3取消绑定 |
| Flstate | | smallint | | 物理状态  1正常  2无效 |
|  | |  | |  |
| Fcreate\_time | | datetime | | 创建时间 |
| Fmodify\_time | | datetime | | 最后修改时间 |
| Fsign | | varchar(32) | | 签名字段 |
| Fstandby1 | | int | | 预留字段 |
| Fstandby2 | | bigint | | 预留字段 |
| Fstandby3 | | decimal(18,2) | | 预留字段 |
| Fstandby4 | | datetime | | 预留字段 |
| Fstandby5 | | varchar(255) | | 预留字段 |
|  | |  | |  |

# 模块设计

## 发送短信验证码接口

请求名称：sendMCode（发送短信验证码）

接口说明：发送短信验证码，会将短信验证码存到sms\_db\_xx.t\_mvcode\_info\_x

请求方式：HTTP POST &soap GET/POST

接口地址：http://120.24.235.105:9010/sms-service-war-1.0/ws/smsFacade.ws?wsdl

输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| client\_ip | String | 必填 | ip地址，可以随便填（这个是客户端ip地址） |
| tmpl\_id | String | 必填 | 短信模板（默认为：1） |
| mobile | String | 必填 | 手机号码 |

输出参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| retCode | String | 必填 | 返回的代码，0-成功，其他失败 |
| retInfo | String | 必填 | 返回的信息，ok表是成功，其他失败 |

Ps：

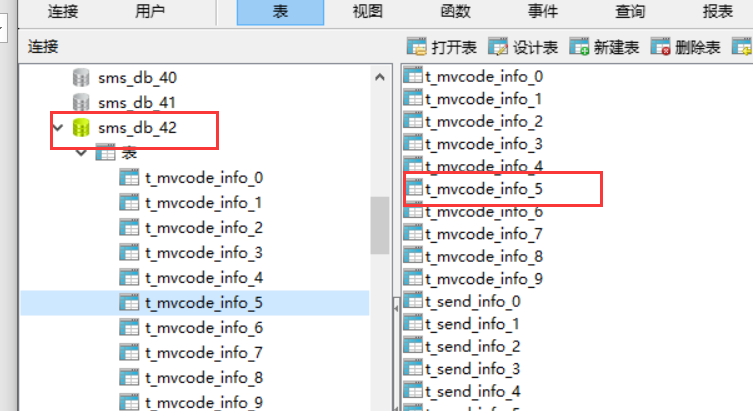
1.接口中的其他参数可以不填，将参数的”?”去掉即可

2.短信的存储是手机号码的最后2位决定在哪个库，倒数第三位决定在哪个表。

例如：

手机号：139xxxx6542

那么短信会存储在sms\_db\_42库，t\_mvcode\_info\_5表里面



## 注册接口

请求名称：userRegister（用户注册）

接口说明：用户注册接口，将用户的信息保存到user\_db.t\_user\_info表。

请求方式：HTTP POST &soap GET/POST

接口地址：

http://120.24.235.105:9010/finance-user\_info-war-1.0/ws/financeUserInfoFacade.ws?wsdl

输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| verify\_code | String | 必填 | 验证码(短信) |
| user\_id | String | 必填 | 注册的用户名 |
| channel\_id | String | 必填 | 渠道信息 1,2,3 |
| pwd | String | 必填 | 登陆密码 |
| mobile | String | 必填 | 手机号 |
| ip | String | 必填 | Ip地址，可以随便填 |

输出参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| retCode | String | 必填 | 返回的代码，0-成功，其他失败 |
| retInfo | String | 必填 | 返回的信息，ok表是成功，其他失败 |

内部逻辑

1. 校验入参参数据是否为空值与并对数据格式进行校验（手机号码）。
2. 根据传递的用户名查询用户的注册信息：用户注册信息已存在，抛出异常。
3. 检查短信验证码，传递用户的手机号及用户输入的验证码。

1）验证码与手机号码关系对应

2）验证码未超时

1. 注册时将用户的关键数据进行加密处理：用户的登录密码/手机号加密，然后将用户的注册信息保存到数据库表user\_db.t\_user\_info。
2. 验证影响数据库的行数是否等于1。

Ps：

1.其他参数可以不填，将”?”去掉即可

## 实名认证

接口名称：verifiedUserAuth（实名认证）

接口说明：实名认证接口，将用户的信息保存到user\_db. t\_user\_auth\_info表。

请求方式：HTTP POST &soap GET/POST

接口地址：

http://120.24.235.105:9010/finance-user\_info-war-1.0/ws/financeUserInfoFacade.ws?wsdl

输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| uid | long | 必填 | 用户Uid |
| true\_name | String | 必填 | 真实姓名 |
| cre\_id | String | 必填 | 身份证号 |

输出参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| retCode | String | 必填 | 返回的代码，0-成功，其他失败 |
| retInfo | String | 必填 | 返回的信息，ok表是成功，其他失败 |

内部逻辑

1. 校验入参用户的真实姓名，身份证号是否为空以及格式是否正确。
2. 用户注册后才能进行实名认证
3. 实名认证成功后，身份证号需要加密存入实名认证表中。
4. 验证数据库影响行数是否为1。

## 绑定银行卡====暂时不做

接口名称：bindBankCard（绑定银行卡）

接口说明：实名认证接口，将用户的信息保存到user\_db. t\_ban\_card\_x表。

请求方式：HTTP POST &soap GET/POST

接口地址：

http://120.24.235.105:9010/finance-user\_info-war-1.0/ws/financeUserInfoFacade.ws?wsdl

输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| uid | long | 必填 | 用户Uid |
| pay\_pwd | String | 必填 | 支付密码 |
| mobile | String | 必填 | 绑定的银行卡手机号（3DES加密保存） |
| cre\_id | String | 必填 | 证件ID（3DES加密保存） |
| user\_name | String | 必填 | 持卡人姓名 |
| cardid | String | 必填 | 卡号（3DES加密保存） |
| bank\_type | int | 必填 | 银行类型1001 |
| bank\_name | String | 必填 | 银行名称/开户行名称 |
| bank\_area | String | 非必填 | 开户地区，省级 |
| bank\_city | String | 非必填 | 开户城市 |

输出参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| retCode | String | 必填 | 返回的代码，0-成功，其他失败 |
| retInfo | String | 必填 | 返回的信息，ok表上成功，其他失败 |

内部逻辑

1. 校验入参的参数，主要是判断字段空值判断和格式判断，不满足则抛异常。
2. 根据传入对象参数用户ID(uid)，检查当前用户的实名认证状态，必须是已经实名认证的才能绑定。
3. 验证用户的证件ID及姓名是否与实名中的一致。

数据存储：

uid的最后2位决定在哪个库，倒数第三位决定在哪个表。

# 数据库连接信息

**Mysql数据库相关登录信息：**

120.24.235.105:3306

python

python666

1. soapui能分别跑通 发送短信验证码、注册、实名认证 ===必须完成
2. 难度升级，短信验证码、注册、实名认证接口串起来跑，流程

Jmeter， ====不做强制要求