[深圳google开发组-聚金融-后端方案设计](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089)

[转至元数据结尾](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#page-metadata-end)

* 由 [方逾阳](http://new.wiki.jnshu.com/display/~fangyuyang)创建, 最后修改于[六月 29, 2018](http://new.wiki.jnshu.com/pages/diffpagesbyversion.action?pageId=18581089&selectedPageVersions=11&selectedPageVersions=12)

[转至元数据起始](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#page-metadata-start)

[Edit Document](http://new.wiki.jnshu.com/plugins/servlet/confluence/editinword/18581089/attachments/%E8%81%9A%E9%87%91%E8%9E%8D%E5%90%8E%E7%AB%AF%E6%96%B9%E6%A1%88.docx)

深圳 google 开发组 - 聚金融 - 后端方案设计

[一， 架构与表格](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc25437)

[1 模块](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc30085)

[2 DB](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc30323)

[3 cache](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc7268)

[4 表格](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc912)

[4.1 产品交易历史记录表](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc2433)

[4.2 用户信息表](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc12804)

[4.3 当期本金交易表](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc873)

[4.4 债权列表](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc159)

[4.5 投资合同表](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc32321)

[4.6 产品信息表](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc2363)

[4.7 后台账号管理表](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc29084)

[4.8 角色管理表](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc7615)

[4.9 模块管理表](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc9720)

[4.10 鼎力推荐表](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc29202)

[4.11 消息管理表](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc32606)

[4.13 B anner 表](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc25308)

[4.14 意见反馈表](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc13744)

[4.15 节假日表](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc9983)

[二， 后台方案](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc12451)

[2.1 登录](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc27184)

[2.2 业务管理](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc11403)

[2.2.1 用户管理](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc291)

[2.2.2 债权管理](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc13905)

[2.2.3 产品管理](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc11285)

[2.3 运营管理](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc16463)

[2.3.1 Banner 图](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc17092)

[2.3.2 鼎力推荐](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc24298)

[2.3.3 消息管理](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc15639)

[2.3.4 意见反馈](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc29473)

[2.4 平台管理](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc26109)

[2.4.1 密码管理](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc31339)

[2.4.2 账号管理](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc12878)

[2.4.3 角色管理](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc13969)

[2.4.4 模块管理](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc17332)

[三， 前台方案](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc13684)

[3.1 注册登录并找回密码](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc8171)

[3.1.1 注册](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc16343)

[3.1.2 登录](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc9933)

[3.1.3 找回密码](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc12200)

[3.2 首页](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc26128)

[3.3 理财](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc29020)

[3.4 我的](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc8093)

[3.4.1 个人信息](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc20823)

[3.4.2 设置](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc16357)

[3.4.3 消息中心](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc32125)

[3.4.4 我的理财 - 续投](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc11170)

[3.4.5 交易记录](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc9307)

[3.4.6 银行卡](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc18664)

[3.4.7 意见反馈](http://new.wiki.jnshu.com/pages/viewpage.action?pageId=18581089#_Toc9867)

一， 架构与 表格

1 模块

1.1 spring-eureka：分为2个eureka，一个eureka最为高可用注册中心，一个eureka最为配置文件加载注册中心

1.2 spring-config-server：springcloud的配置文件加载服务，用来加载github上面的配置文件，作为服务提供给springcloud-config-client   
    1.3 spring-config-client：获取配置文件，启动后，主要业务逻辑写在 spring-config-client，也可以理解为spring-cloud-server-privator，spring-cloud的服务提供者     
    1.4 spring-feign：spring-cloud框架内部负载均   
    1.5 spring-zuul：spring-cloud的网关   
    1.6 spring-hystric：spring-cloud的熔断器   
    1.7 spring-bus：消息中心

2 DB

用到的数据库类型mysql

3 cache

cache：redis

4 表格

**4.1 产品交易历史记录表**

每次交易动作，回款动作，都要在这张表中记录， quartz 执行返款时也要执行这张表。

|  |  |  |  |
| --- | --- | --- | --- |
| **history\_product** | | | |
|  |
| 产品交易历史记录表 | | | |  |
|  |
| column | field | 备注 | 类型 |  |
|  |
| id | id |  | Long |  |
| 交易方式 | bank\_card |  | String |  |
|  |
| 产品 | product |  | String |  |
|  |
|  |
|  |
| 购买人 | buyer |  | String |  |
|  |
| 购买时间 | buy\_time |  | Long |  |
|  |
| 到期时间 | arrive\_time |  | Long |  |
|  |
| 交易动作 | transaction\_action | 利息 0/本息返1/本金返2/支付3预约续投4/续投中5 | int |  |
|  |
| 交易金额 | transaction\_amount |  | BigDecimal |  |
|  |
| 电话 | phone |  | String |  |

**4.2 用户信息表**

记录用户的信息，登录信息也在这张表中，状态冻结不能登录。

|  |  |  |  |
| --- | --- | --- | --- |
| **user** | | | |
|  |
| 用户信息表 | | | |  |
|  |
| column | field | 备注 | 类型 |  |
|  |
| id | id |  | Long |  |
|  |
| 电话 | phone |  | String |  |
|  |
| 密码 | password |  | String |  |
|  |
| 总资产 | total\_assets |  | BigDecimal |  |
|  |
| 身份证 | id\_card |  | String |  |
|  |
| 盐码 | salt |  | String |  |
|  |
| 账号 | account\_number |  | String |  |
|  |
| 真实姓名 | true\_name |  | String |  |
|  |
| 银行卡1 | bank\_card1 | 银行+id | String |  |
|  |
| 银行卡2 | bank\_card2 |  | String |  |
|  |
| 是否已买新手礼 | is\_novice\_gift |  | boolean |  |
|  |
| 状态 | status | 正常/冻结 | int |  |
|  |
| 累计收益 | accumul\_income |  | BigDecimal |  |
|  |
| 消息阅读记录 | infor\_record | String[] | String |  |
|  |
| 注册时间 | create\_at |  | Long |  |
| 更新时间 | update\_at |  | Long |  |

**4.3 当期本金交易表**

这张表记录当前有效的交易，用于处理返息。

|  |  |  |  |
| --- | --- | --- | --- |
| **trade** | | | |
|  |
| 当期本金交易表 | | | |  |
|  |
| column | field | 备注 | 类型 |  |
|  |
| id | id |  | Long |  |
|  |
| 投资合同id | trade\_id |  | String |  |
|  |
| 产品 | product |  | String |  |
|  |
| 购买人 | purchaser |  | String |  |
|  |
| 购买时间 | buy\_time |  | Long |  |
|  |
| 延续期限 | delay\_time |  | Long |  |
|  |
| 到期时间 | expire\_time |  | Long |  |
|  |
| 待批配金额 | to\_be\_matched |  | BigDecimal |  |
|  |
| 购买金额 | purchase\_amount |  | BigDecimal |  |
|  |
| 支付服务协议(投资合同id) | investment\_contract\_id | 待定 | String |  |
|  |
| 状态 | status | 返息中0/本息返1/准备返本金2/准备返本息3/准备返息4/预约续投中5/续投中6/ | Int |  |
|  |
| 已债权id | matched\_obl\_id | Sting[] | String |  |
|  |
| 年收益率 | expected\_rate |  | BigDecimal |  |
|  |

**4.4 债权列表**

线下的债权列表，用来记录债权，以及债权匹配，状态表示匹配的状态。

|  |  |  |  |
| --- | --- | --- | --- |
| **obligatory\_right** | | | |
|  |
| 债权列表 | | | |  |
|  |
| column | field | 备注 | 类型 |  |
|  |
| id | id |  | Long |  |
|  |
| 债权编号 | obl\_number |  | String |  |
|  |
| 企业名称 | company |  | String |  |
|  |
| 企业法人 | corporate |  | String |  |
|  |
| 手机号 | mobile |  | String |  |
|  |
| 身份证 | id\_card |  | String |  |
|  |
| 借款金额 | loan\_sum |  | BigDecimal |  |
|  |
| 借款期限 | loan\_term |  | Long |  |
|  |
| 借款日期 | loan\_time |  | Long |  |
|  |
| 还款日期 | repay\_time |  | Long |  |
|  |
| 投资合同ID | investment\_id | String[] | String |  |
|  |
| 投资金额 | investment\_amount | String[] | String |  |
|  |
| 理财期限 | financial\_period | String[] | String |  |
|  |
| 待匹配金额 | matching\_amount |  | BigDecimal |  |
|  |
| 状态 | status | 匹配中/已/未 | int |  |
|  |

**4.5 投资合同表**

用户的交易记录，表示投资合同，用来匹配债权。

|  |  |  |  |
| --- | --- | --- | --- |
| **investment\_contract** | | | |
|  |
| 投资合同表 | | | |  |
|  |
| column | field | 备注 | 类型 |  |
|  |
| id | id |  | Long |  |
|  |
| 真实姓名 | true\_name |  | String |  |
|  |
| 手机号 | phone\_number |  | String |  |
|  |
| 产品 | product |  | String |  |
|  |
| 债权id | obl\_right\_id | Sting[] | String |  |
|  |
| 投资金额 | invest\_amount |  | BigDecimal |  |
|  |
| 收益 | income |  | BigDecimal |  |
|  |
| 生效时间 | effective\_time |  | Long |  |
|  |
| 失效时间 | invalid\_time |  | Long |  |
|  |
| 状态 | status |  | int |  |
|  |

**4.6 产品信息表**

记录产品的详细信息。用户购买产品的信息从该表中查询，金融计算所需要的数据也行该表中查询

|  |  |  |  |
| --- | --- | --- | --- |
| **product** | | | |
|  |
| 产品信息表 | | | |  |
|  |
| column | field | 备注 | 类型 |  |
|  |
| id | id |  | Long |  |
|  |
| 产品Id | product\_id |  | String |  |
|  |
| 产品 | product\_name |  | String |  |
|  |
| 预期年利率 | expected\_rate |  | BigDecimal |  |
|  |
| 投资周期 | investment\_cycle |  | int |  |
|  |
| 起投金额 | origin\_money |  | BigDecimal |  |
|  |
| 最小增减额 | add\_money |  | BigDecimal |  |
|  |
| 产品介绍 | introduce |  | String |  |
|  |
| 每月是否返息 | return\_interest |  | boolean |  |
|  |
| 还款日期 | repayment\_date |  | Long |  |
|  |
| 返息日期 | return\_date |  | Long |  |
|  |
| 是否上架 | status |  | boolean |  |
|  |
| 更多信息 | more\_information |  | String |  |
| 创建人 | founder |  | String |  |
| 创建时间 | create\_at |  | Long |  |

**4.7 后台账号管理表**

后台管理人员的信息表格，后台人员有对应的角色，角色有对应权限

|  |  |  |  |
| --- | --- | --- | --- |
| **backstage\_account** | | | |
|  |
| 后台账号管理表 | | | |  |
|  |
| column | field | 备注 | 类型 |  |
|  |
| id | id |  | Long |  |
|  |
| 管理员 | admin | 用户名 | String |  |
|  |
| 账号 | account\_id |  | String |  |
|  |
| 密码 | password |  | String |  |
|  |
| 角色 | admin\_role |  | String |  |
|  |
| 创建人 | founder |  | String |  |
|  |
| 创建时间 | create\_at |  | Long |  |
| 更新人 | modifier |  | String |  |
| 更新时间 | update\_at |  | Long |  |
| 状态 | status |  | int |  |

**4.8 角色管理表**

角色创建有对应的权限，没种角色有对应的进入模块的权限

|  |  |  |  |
| --- | --- | --- | --- |
| **role\_manage** | | | |
|  |
| 角色管理表 | | | |  |
|  |
| column | field | 备注 | 类型 |  |
|  |
| id | id |  | Long |  |
|  |
| 角色id | role\_id |  | String |  |
|  |
| 角色 | role |  | String |  |
|  |
| 创建人 | founder |  | String |  |
|  |
| 创建时间 | create\_at |  | Long |  |
|  |
| 修改时间 | update\_at |  | Long |  |
|  |
| 跟新人 | modifier |  | String |  |
|  |
| 权限 | authority | int[] | String |  |
|  |

**4.9 模块管理表**

创建模块，每个模块就是对应的一种权限，进入模块需要权限。

|  |  |  |  |
| --- | --- | --- | --- |
| **m** **odule** **\_manage** | | | |
|  |
| 模块管理表 | | | |  |
|  |
| column | field | 备注 | 类型 |  |
|  |
| id | id |  | Long |  |
|  |
| 模块名 | module\_name |  | String |  |
|  |
| 父某块 | parent\_module |  | Long |  |
|  |
| 模块类型 | type |  | String |  |
|  |
| 更新时间 | update\_at |  | Long |  |
|  |
| 创建时间 | create\_at |  | Long |  |
|  |
| 创建人 | founder |  | String |  |
|  |
| 更新人 | modifier |  | String |  |
|  |
| url | module\_url |  | String |  |

**4.10 鼎力推荐表**

鼎力推荐表就是推荐对应的产品

|  |  |  |  |
| --- | --- | --- | --- |
| **recommend** | | | |
|  |
| 鼎力推荐表 | | | |  |
|  |
| column | field | 备注 | 类型 |  |
|  |
| id | id |  | Long |  |
|  |
| 推荐id | recommend\_id |  | String |  |
|  |
| 产品标题 | product |  | String |  |
|  |
| 图片 | recommend\_picture |  | String |  |
|  |
| 创建人 | founder |  | String |  |
|  |
| 创建时间 | create\_at |  | Long |  |
|  |
| 更新人 | modifier |  | String |  |
|  |
| 更新时间 | update\_at |  | Long |  |
|  |
| 消息状态 | status |  | int |  |
|  |

**4.11 消息管理表**

消息推送，记录时间状态，及时推送，状态直接推送，记录定时任务，就用 quartz 执行

|  |  |  |  |
| --- | --- | --- | --- |
| **information\_manage** | | | |
|  |
| 消息管理表 | | | |  |
|  |
| column | field | 备注 | 类型 |  |
|  |
| id | id |  | Long |  |
|  |
| 消息id | infor\_id |  | String |  |
|  |
| 消息标题 | infor\_title |  | String |  |
|  |
| 创建人 | founder |  | String |  |
|  |
| 创建时间 | create\_at |  | Long |  |
|  |
| 更新人 | modifier |  | String |  |
|  |
| 更新时间 | update\_at |  | Long |  |
|  |
| 消息状态 | infor\_status |  | Number |  |
|  |
| 发送时间 | send\_time |  | Long |  |
|  |
| 消息内容 | infor\_content |  | String |  |
|  |
| 图片 | infor\_picture |  | String |  |
|  |

**4.13 B anner 表**

对应权限的人直接添加 banner 图， banner 有对应产品记录产品。 url 就是对应产品详情。

|  |  |  |  |
| --- | --- | --- | --- |
| **banner** | | | |
|  |
| banner图表 | | | |  |
|  |
| column | field | 备注 | 类型 |  |
|  |
| id | id |  | Long |  |
|  |
| Banner图编号 | banner\_id |  | String |  |
|  |
| 标题 | title |  | String |  |
|  |
| url | url |  | String |  |
| 缩略图 | banner\_picture |  | String |  |
|  |
| 间隔 | interval |  | int |  |
|  |
| 创建人 | founder |  | String |  |
|  |
| 创建时间 | create\_at |  | Long |  |
|  |
| 更新人 | modifier |  | String |  |
|  |
| 更新时间 | update\_at |  | Long |  |
|  |
| 状态 | status |  | number |  |
|  |
| 推送时间 | push\_time |  | Long |  |
|  |

**4.14 意见反馈表**

用户提交意见，插入数据，回复就在回复内容添加内容。

|  |  |  |  |
| --- | --- | --- | --- |
| **suggestion** | | | |
|  |
| 意见反馈表 | | | |  |
|  |
| column | field | 备注 | 类型 |  |
|  |
| id | id |  | Long |  |
|  |
| 意见id | sugge\_id |  | String |  |
|  |
| 意见内容 | sugge\_content |  | String |  |
|  |
| 提交人 | adviser |  | String |  |
|  |
| 手机号 | phone |  | String |  |
|  |
| 提交时间 | submit\_time |  | Long |  |
|  |
| 回复人 | replier |  | String |  |
|  |
| 回复内容 | reply\_content |  | String |  |
|  |
| 回复状态 | status |  | int |  |
|  |
| 回复时间 | reply\_time |  | Long |  |
|  |

**4.15 节假日表**

银行节假日不上班，所有需要节假日

|  |  |  |  |
| --- | --- | --- | --- |
| **festival** | | | |
|  |
| 节假日表 | | | |  |
|  |
| column | field | 备注 | 类型 |  |
|  |
| id | id |  | Long |  |
|  |
| 日期 | day |  | Date |  |

二， 后台方案

2.1 登录

模块名称：登录模块

需求描述： 作为一个管理员，我想要登录后台账号，以便于使用账号权限进行管理。

相关方案：

后台是给管理人员进行对相应的用户管理，业务管理的地方，我们在设计用户登录时需要对应上用户的用户名与密码，在与数据库对应上后可以进入后台管理系统，但是对应的页面显示会根据对应用户名查出其角色进而查出对应权限而开放给他相应的管理页面，我们查询每个账户的对应角色，然后查询其权限可以进入的模块，向前端传送对应的模块。

2.2 业务管理

**2.2.1 用户管理**

模块名称：用户管理

需求描述： 作为一个管理员，我想要 管理当前的用户信息。

相关方案：

（ 1 ） 查询

查询方式可以选择任意四个字段：用户 ID ，真实姓名，手机号，状态来组合查询对应用户并显示在用户列表中。当对应的字段输入越多就查询越精确，如果查询不到，就返参告诉前端查不到对应数据，并且用户列表也为空。

（ 2 ） 列表显示

用户列表刚开始就查询数据库的用户列表所有数据并分页显示，然后操作过程有查看，冻结，解冻。

查看可以跳转去用户详情，解冻和冻结都是对用户的状态进行修改。

（ 3 ） 用户详情

    用户详情显示具体的对应的用户全部内容，在用户详情可以对手机号进行更改

    在总资产中也可显示交易记录，我们的总资产是记录下当前生效的用户的理财产品的总本金额，我们可以去查对应的当前本金生效表，

然后算下对应的用户的各个理财方案的本金和，同时将对应的个人在本金生效表所有的理财方案去数据库中查找。

   对于累计收益的金额是累计所有这个用户的交易记录后算出的利息金额。这个我们在每次进行当月利息返还时就更新，把所有月息累加起来。

    对于交易记录的话我们是去查找交易记录表把对应这个账户的涉及本金交易的记录查出，如果是续款的话我们是算成一个新的交易记录的，所以单独列出，交易时间就算当时起效时间。

    所以我们制定的每天零点程序需要一一遍历当前生效本金表的执行过程如下：

（ 1 ） 查看表中结束日期为今天的所有理财产品，然后判断他们的状态是本息返还，还是本金返还。则若为本息返还则需要对应计算所有利息再加上本金一次性返还给对应银行卡，若为本金返还则只返本金。

（ 2 ） 查看今天日期是否为 20 号，若为 20 号，则接着查找所有是在返息中的理财产品，决定在今天把所有需要在今天支付的月息返还

（ 3 ） 对于状态修改，因为我们是查看当前生效本金表的状态来处理的，所有当如果已经返还本金与本息则移动此交易记录去历史记录表，但是如果遇到节假日与周末无法返息或者返本金情况需要拖延的，需要将状态改成待 .. 中，具体情况具体再定，然后直到我们查出今天不再是节假日就将资金打出，再修改状态成已付，再对应操作。 有另一种方法就是直接算出几天后可以打款，若今天打不了就使用 quartz 定时到非节假日的第一天打款给用户。

（ 4 ） 当然我们这里的支付由于是在富友支付的支付环境，所以只能使用模拟支付，只在数据库修改金额，修改表格。

**2.2.2 债权管理**

模块名称：债权管理

需求描述： 1. 作为一名管理员，我想要对债权进行增删改查的统一管理，以便于线上债权记录与线下购买债权情况一致。

2. 作为一名管理员，我想要对债权进行手动匹配，以便于保证用户理财投资能够正常获得收益。

相关方案：

（ 1 ） 查询

   查询过程是使用四个字段：债权 ID ，企业名称，债务人，状态，只需要查询这任意四个字段的组合都可以查询对应想要的债权。

（ 2 ） 债权列表

    债权列表刚开始就查询数据库的债权列表所有数据并分页显示，然后操作过程有匹配，编辑，删除。注意债权列表对于编辑就是对其具体内容进行修改，当我们点击匹配时则可以跳转到债权匹配，在对应的表格中显示有该债权匹配的投资合同，投资金额，理财期限，我们设置 String 数组存储这三部分内容，

对于删除操作我们要注意到删除时会对所有的投资合同的待匹配金额进行修改，当删除这条字段时会将之前对这个投资合同的投资金额加回待批配金额，

（ 3 ） 债权匹配

        我们讨论下何为债权匹配，是因为我们需要将用户投资的理财产品资金由我们去购买对应的公司债权，将之间不等额的资金匹配成相等的资金，我们注意到之前存在待匹配，匹配中，已批配三种情况，而且我们需要遵循以下几个原则：

1. 一笔债权匹配可由多笔投资组合而成，但投资总额不得大于债权总额。同一债权与同一投资仅匹配一次。
2. 注意到待匹配金额在于所有投资合同匹配时可能存在配不平，得用更大的投资合同去配，这样存在匹配好合同后存在部分投资合同金额为被匹配债权，需要再进行下一步的将此合同放入匹配过程进行匹配
3. 优先推荐与借款期限相近的投资，其次是推荐到期日接近的投资，最后是推荐与借款金额接近的投资。
4. 当投资的理财期限早于债权的借款期限到期时，投资正常退出后，未完成债权以该退出日期进行新一轮推荐，优先推荐投资日期与退出日期相近的新投资。
5. 当投资的理财期限晚于债权的借款期限到期时，债权正常到期后，未完成投资以该到期日期进行新一轮推荐，正常参与其他债权匹配，不影响收益计算。

所以针对这几点当我们在用户详情点击匹配时就跳转去匹配列表。

我们要将下面推荐的投资合同金额去匹配上债权金额，先讨论如何实现去对投资合同的分拆匹配债权，我们在每个投资合同在匹配时使用的字段是待匹配金额，最开始待匹配金额是与合同金额一样，当被匹配完全后就待匹配金额为 0 ，如果没有被匹配完全，债权只能匹配其一部分则只减去已匹配的金额剩余的就是待匹配金额，然后将这个未匹配好的重新在第二次匹配时再根据合适的条件去调出这个待匹配金额还未为 0 的投资合同。

对于推荐的投资合同，我们推荐的实现方法 :  用债权的具体期限减去所有资金的延续期限，若为大于 0 的按照差值越小的越排前，当对于同个期限的则需要再按结束期限越接近债权的结束期限的排前面，加入这两个条件也有重复的再按照金额越大排前面，我们直接用 sql 语句来排序

        在我们点击匹配后就会跳转到匹配列表，显示的状态有已批配，匹配中，未匹配，匹配中的情况是因为债权金额大于匹配的投资合同金额。已批配则是金额已经满足则显示已批配。

       此外所有投资页面我们是遍历有效本金表显示出的合同。

**2.2.3 产品管理**

模块名称：产品推荐

需求描述： 1. 作为一名管理员，我想要对产品进行增删改查、上下架的统一管理，以便于根据业务情况调整产品信息。

2. 作为一名管理员，我想要对新手礼进行限购，以便于保证用户（真实身份）仅能购买一次新手礼。

相关方案：

（ 1 ） 查询

       产品查询方式有产品 ID ，产品名称，还款方式，创建人，

（ 2 ）产品列表

       产品管理显示的是我们的所有产品，产品状态有上架，下架，刚开始新增的产品都是下架的状态，即为 0 ，上架为 1 此时才可以被我们进行推送。

       用户买了新手礼后我们对每个都记录，这样当要推送给每个用户时就需要先找到这个字段判断是否买了，才决定推送。

2.3 运营管理

**2.3.1 Banner 图**

模块名称： banner 图

需求描述： 作为一名管理员，我想要对banner图进行增删改查和上下架的统一管理，以便于根据情况调整宣传内容。

相关方案：

       （ 1 ）查询

       查询方式有 banner 图 ID ， banner 图名称，创建人，更新人，

（ 2 ） Banner 列表

       Banner 图列表显示的是所有 banner 。

对于编辑页面显示的有 url 这个选项，代表跳转的页面地址，我们在保存这个 banner 图对象时， 1 ，需要保证标题输入汉字在 2-10 个，这个前端判断，但是不能重名，所以得先去查一遍是否有相同的字段，若无才可以放进去数据库

此外对于更新时间与创建时间都应该记下当时时间

**2.3.2 鼎力推荐**

模块名称：鼎力推荐

需求描述： 作为一名管理员，我想要对鼎力推荐进行增删改查和上下架的统一管理，以便于宣传最新最火的产品。

相关方案：

       （ 1 ）查询

        查询字段采用 Banner 图 ID ， Banner 图名称，创建人，更新人。

（ 2 ）推荐列表

推荐列表显示的其实就是所有已经上架的产品，然后我们要注意到上架的应该至少得有两个项目，一个为新手礼和其他类型产品。如果上架多个的话，我们随机推送一个其他类型产品和新手礼给前端。我们在编辑时只能对其图片，标题进行修改。

**2.3.3** **消息管理**

模块名称：消息管理

需求描述： 1. 作为一名管理员，我想要对消息进行增删改查的统一管理，以便于根据运营情况调整消息内容。

2. 作为一名管理员，我想要对用户定时推送消息，以便于提前准备消息文案，更好地配合活动策划和运营

相关方案：

（ 1 ）这里的消息是群发消息，可以通过类型，创建人，消息标题查找

（ 2 ）消息类型分为通知类和应用类，表示推送位置的不同，状态有已推送，待推送，在推送时间之前都为待推送。实现方法：当后台计算时间到发送时间时就将推送状态更改为已推送，然后在每个用户的个人信息表中的阅读记录中记下为“未阅读”状态，推送定时消息我们使用 quartz 来进行。

实现方法： 消息表中2个字段一个是状态，一个推送时间。在插入数据的时候有2个选项，如果是 当时就把 状态直接设置 为 推送，用户就能直接查询到，如果是定时任务，就插入推送时间，同时quartz任务启动定时任务执行，用户在get消息的时候要满足推送条件的消息才能获取到，quartz执行定时任务，修改消息的状态，改为推送，用户就能在对应时间获取到消息。

（ 3 ）然后我们在编辑界面中可以附加宣传图，所以数据库中也得有相应的保存图片的地方。

**2.3.4 意见反馈**

模块名称：意见反馈

需求描述： 作为一名管理员，我想要对用户反馈的建议意见进行增删改查和回复的统一管理，以便于提升APP的用户体验，方便用户使用操作

相关方案：

       （ 1 ）查询

        查询字段有意见 ID ，关键词，提交人，手机号，对于关键词我们设置为模糊搜索，只需要对相关字段查找即可。

       （ 2 ）意见列表

         意见列表里的所有意见都是当客户发送了相应的意见后会在数据库加上这条意见，这条意见有两个状态，已处理和未处理，在遍历表时应该先显示未处理的数据，然后回复过的都改变他的状态，标记状态为已回复，不能进行操作。至于还未恢复的可以点击回复书写回复意见，回复则可以存入数据库中。

2.4 平台管理

**2.4.1 密码管理**

模块名称：密码管理

需求描述： 作为一个管理员，我想要修改后台账号密码，以便于保证后台账号的安全性

相关方案：

输入密码时当输入的旧密码与找出的用户密码不一致时就返参告诉前端密码错误，如果正确的话就可以直接修改密码。所以逻辑点击保存时先查找密码，得到旧密码进行比对再进行下一步操作。但是如果是管理员可以直接修改。

**2.4.2 账号管理**

模块名称：账号管理

需求描述： 作为一个管理员，我想要可以对所有的账号角色进行增删改查的操作，以便我对不同账号的角色管理。

相关方案：

账号管理的创建者有严格限制，需要是至少管理员级别的成员才能设置，所以我们待会在角色管理这一块将所有的角色所拥有的权限进行限制，对于账号的状态，我们有启用禁用，只有更高级别的人员才能进行操作，而级别优先级如下：

运营 = 客服 < 管理员《超级管理员，所以在进行禁用操作时会先根据具体人员的级别，然后去查找所要删除的人员的级别是否比他低，如果低则可以编辑，而且创建也是得级别更高才可以。此外在编辑页面中能够进行编辑与新增的都只有管理员级别的人才能操作。

所以具体实现情况可以在初步遍历显示账号列表时，就记下对应的账号的等级放入缓存中，然后每次去比较账号的等级时就可以直接去缓存调用账号的等级比较。

**2.4.3 角色管理**

模块名称：角色管理

需求描述： 作为一名管理员，我想要对所有的角色权限进行增删改查的统一管理，以便我对不同角色的权限分配。

相关方案：

（ 1 ） 查询

   查询条件有角色 ID ，角色名，创建人，更新人

（ 2 ） 角色列表

   角色是我们赋予对后台管理的职权的设定，对于所有个角色，分为超级管理员，管理员，和非管理员类，我们赋予的对各个模块的职权：

   超级管理员：所有模块都可以操作

   管理员：删除修改除了管理员级别的其他账号和角色

   非管理员：不允许删除账号。

        实现方式： 使用一个 int[] 数组存放对于每个位置的权限管理，有权限的位置放置 1 ，无权限仍为 0 。

**2.4.4 模块管理**

模块名称：模块管理

需求描述： 作为一名管理员，我想要对所有的模块内容进行增删改查的统一管理，以便我对不同模块的内容设置

相关方案：

（ 1 ） 查询

   模块查询具体的查询字段有模块 ID ，模块名，父模块，创建人，之间可以随机组合配合查询

（ 2 ） 模块管理

   模块是我们对程序的管理登记的地方。例如我们当前所有的模块都得去这个模块管理登记一下，增加程序的扩展性。

我们记下模块的 id ，和对应的父模块 id ，和模块 url ，初步设置成如下：

我们为了便于解决问题，设置如 id 为 231 的密码管理，父模块记为 23 ，即后台管理，密码修改设为 2311 ，父模块既是密码管理 231 。

三， 前台方案

3.1 注册登录并找回密码

**3.1.1 注册**

模块名称：个人信息

需求描述： 作为一名用户，我想要在APP上更换绑定手机，以便于保证账号和资产的安全性。

相关方案：

      使用手机号进行注册，当点击获取短信验证码时，我们后台根据此手机号调用阿里云 API 的短信接口随机生成验证 code ，放入缓存中，开始计时并把时间放入缓存，发送给用户，然后用户根据验证码来输入验证码，点击下一步后就将该短信验证码发送给我们后台去调用缓存中的 code 比对，相同则可以注册成功，将该手机号放置数据库中。各种情况处理如下：

<1> 如果号码输入导致阿里云无法发送，就返回号码输入错误的返参提示前端

<2> 我们在输入号码点击获取验证码时会先去查找数据库的手机号码是否有相同的，只有不相同才会调用 API ，如果相同则返回有重复手机号的返参

<3> 获取验证码过程可能存在手机信号延迟，或者输入过慢，或者号码忘记，可以在 60s 后重新输入，这时再更新缓存的 code 值，时间。

<4> 如果点击下一步的时候，我们把缓存的时间取出在用当前时间减去发现大于 15\*60\*1000 （ 15 分钟），就返回密码失效的参数，需要重新发送验证码并更新对应这号码的验证码。

<5> 每个手机号码可以在每天获取二十次验证码，这种情况我们设置对应的缓存，将其今天使用次数，对应的号码都记录到缓存中，然后统计达到 20 次后拒绝发送给此号码验证码，并返参，第二天的零点开始刷新这部分号码缓存

<6> 如果验证码与自己的缓存 code 不一致时，会统计下对应的错误次数，放置到缓存中，然后第三次错误就去调出我们的图形验证码函数，图形验证码是由随机的四位字母，数字组成，当看不清楚时点击刷新可以直接去调用图形验证码函数发送新的图形验证码过来。同时这个图形验证码也要放置到对应手机号的缓存中这样可以进行比对，我们把图片放到云上，只发送 url 给前端。

**3.1.2 登录**

      登录时当输入手机号与密码后，传到数据库进行查找此手机号码然后再比对密码是否一致如果是则允许该用户登录。情况说明：

      <1> 如果发现数据库该用户是被冻结则拒绝此号码登录，返对应参数

      <2> 如果查找格式正确的手机号在数据库中无，那么就返回对应参数，提示去注册

**3.1.3 找回密码**

      对于当已经无法记住原来密码的需要重新设置密码，需要使用手机号去重新发送短信验证码。这部分的实现情况实际上与注册还是很相似的，不再多加叙述。

3.2 首页

      首页显示的是 banner 图，鼎力推荐。当初始进来都是这个页面，非登录也是进这个页面。我们在用户登录时同时记录下该用户是否为新手，若为新手时就请求后产生的鼎力推荐图片应该是显示新手礼的图片（非登录也是显示这个鼎力推荐），若为老手则应该显示我们随机推送的其他产品，实现方式是前端请求，我们判断其为新手与否来决定传哪个产品给前端显示。

3.3 理财

     理财列表显示前五条理财产品，注意新手礼都只用一次，所以得根据用户是否购买了新手礼来判断是否要显示新手礼，然后注意理财列表只显示前五条数据，所以查询时每刷新一次都可以要去查后五条数据。

实现方法：我们记下刷新页面次数，然后使用分页法，每次发 5 个数据，

当点击对应的产品时就进入对应的产品页面

我们把产品分成先息后本和本息返还两种，目前先息后本的有聚年增，聚年盈；本息返还的有新手礼，聚月赚，聚月增。本息返还就以聚月赚来举例，先息后本的则举下聚年盈。

（ 1 ）聚月赚

本息返还都是等到还款日期再全部返还本金加利息，我们注意到我们是单利产品，聚月赚的年收益率是 7% ， 3 个月后才返本息，所以计算方式是本金 \* （ 1+7%/360\*90 ），但是对于程序的实现我们是通过在一天的零点时查看当前本金记录表看结束日期是否是几天，如果是的话就进行本息返还到默认的第一张银行卡中，然后将该理财产品移入到产品历史交易记录表中。

然后注意到返息时间是需要根据银行的工作日来判断的，而这时得参考是否日期为周末还是节假日，我们周末的判断是采用 JAVA 自带的工具类来使用的，而节假日的判断则是将节假日输入数据库做个节假日表，只要去查找当天是否节假日则可以知道得延期返款。

至于购买过程我们输入购买金额后，我们默认是使用第一张绑定的银行卡，当然这个过程得判断是否第一张卡是否存在，然后也可以选卡。

我们这里因为使用的是富友支付的测试环境，使用提供的网关支付来进行测试，所以需要跳转到富友支付的提供的页面来再支付，所以我们刚开始使用的页面填写的先存入缓存，等在富友支付支付成功后前端返回 code 码给我们，再将金额存入数据库。

<1> 如果购买成功的话今天购买时间不能算，需要起效时间为第二天，则结束时间就应该是 T+1

（ 2 ）聚年盈

对于聚年盈这些先息后本的理财产品则是每个月的 20 号固定返息，我们这个产品年收益率是 9% ，那么这种情况的话：

在第一个月如果是在 20 号之前买的产品，则返还第一个月的月息是本金 \*(20- 当前日期 )\*9%/360 ，如果是在 20 号后则是本金 \* （ 50- 当前日期） \*9%/360 ，之后的每个月都是本金 \*30\*9%/360 。

在最后一个月的时候如果是在 20 号之前买的则还本金时需要再加上本金 \*(10+ 当前日期 )\*9%/360 ，如果在 20 号之后买的则可以直接 20 号先返回月息再返本金。

3.4 我的

     在我的页面有显示我的理财的地方，需要在登录后将该用户的总资产显示出来。当然未登录会直接跳转去登录界面。

需求描述： 3 作为一名用户，我想要获得最新消息的及时提示，以便于不会错过知道最新消息内容。

相关方案：此页面显示时必须自动去查找个人信息表的阅读记录，统计多少还是未读的信息，然后显示出来

实现方法如下： 先去查找个人信息表中的阅读记录字段，按照最新的信息排序，为此我们对阅读记录这个字段设计的是 string 数组，需要在字段中加上消息的名称，时间，阅读状态，哪个表，对应的表中 id 。查出这条字段然后切段，统计有多少是未读的消息。

第二种实现方法就是新建个阅读过的记录表，放置的字段是对应的消息 id ，阅读人 id ，只要统计阅读人 id 读过的消息次数，再比较消息反馈表和消息总表此人的消息总数即可知道多少未阅读。

**3.4.1 个人信息**

     个人信息要去查询个人信息表展示当前用户的用户名（手机号），是否实名认证，

（ 1 ） 实名认证

模块名称：银行卡 / 实名认证

需求描述： 作为一名打算购买理财的用户，我想要可以绑定银行卡，以便于使用常用的银行卡进行支付 ，也 想要在APP上进行实名认证，以便于提高账号和资产的安全性。

相关方案：实名认证需要去调用身份证 API ，银行卡 API ，在输入名字，身份证，卡号后，点击下一步，会判断以下几种情况

<1>我们测试环境只能实现对应身份证是否有效和籍贯，所以第二种想法就是调用其他接口去实名认证。

（ 2 ） 更换手机

  更换手机需要输入手机验证码这个验证过程跟之前的注册时输入手机验证码基本相同。

在验证码输入成功后进入更换手机页面，这个验证码方式其实也是一样的， 只是这里要注意同一手机号一天可以收到 50 次验证码。

**3.4.2 设置**

（ 1 ） 修改密码

模块名称：通用设置

需求描述： 作为一名用户，我想要修改个人账号的密码，以便于保证个人账号和资产的安全性。

相关方案：

修改密码时需要先输入原来的密码，当输入新密码和确认密码都正确时，就可以去查该用户的密码是否正确然后如果正确就可以修改新密码，如果不正确还得返参，提醒输入正确旧密码，当输入成功后就可以去将新密码更改了旧密码。

（ 2 ） 版本更新

模块名称：通用设置

需求描述： 作为一名用户，我想要获取最新的版本更新，以便于体验最新的功能和服务。

相关方案：

         版本更新是我们前后台进行版本迭代时进行的过程，提醒用户去更新最新的程序。操作过程我们只能是在更新时不显示页面，等更新好再用。

（ 3 ） 退出登录

模块名称：通用设置

需求描述： 作为一名用户，我想要退出登录状态，以便于保护自己的个人隐私

相关方案：

         当我们用户开启登录进程序时，这时我们使用 cookie 记录下用户名，和当前时间，我们免登时间为 5 小时，超出后 token 就需要用户去使用密码登录了，但是如果用户开启程序就会有个状态，只要一直开启程序就一直保持着免登状态。如果点击退出登录则会使该 Cookie 为 null

**3.4.3 消息中心**

模块名称：消息中心

需求描述： 1 作为一名用户，我想要查看所有接收到的消息，以便于查收和管理所有接收到的消息。

相关方案： 消息页面显示的是个人的意见反馈，以及统一的消息推送，所以在点击该页面，去查找个人消息表的阅读记录字段。需要显示时间，阅读与否，消息名。

需求描述： 2 作为一名用户，我想要查看消息详情，以便于了解更详细的消息内容

相关方案：当点击信息时才去调用反馈表或者消息列表的消息，若是未阅读的点击时就会先修改阅读记录为已阅读后再去查询。

**3.4.4 我的理财 - 续投**

模块名称：我的理财

需求描述： 1 作为一名用户，我想要知道账户资产的变化情况，以便于对账户资产进行合理配置。

          2.作为一名用户，我想要理财产品到期后可以进行续投，以便于我可以继续享受收益。

          3.作为一名用户，我想要查看签署的投资合同，以便于能够维护自己的投资权益。

相关方案：这里划分为已投资，已续投，已退出三种选项，已投资是当前购买的理财还在生效的产品，已续投是之前购买的产品选择在投资时间结束时接着续买该产品，已退出是已经超过购买时间并且将所有本金已退回的理财产品。具体实现方法如下：

         我们去查历史交易记录表里所有对应该账户的本金交易

（ 1 ） 已投资是记录表中交易动作为支付状态的，当点击预约续投时就会在记录表中的交易动作加上预约续投这个动作，等到我们后台查到到期时间是今天时，就将支付状态改成返息（根据具体项目来定），将过期状态记成已过期，然后就产生一条新纪录标记为续投中。如果未点击预约续投则到期后标记交易动作未本金返还 / 本息返还，将记录标记过期。此外，当将取消续投按钮点下，会将对应在该产品的交易动作的预约续投消除掉。

（ 2 ） 已续投是记录表中记录为续投中的记录，如果点击预约续投可以进行下一轮的续投，这时在交易动作中增加预约续投即可，当然点击取消续投也可以把对应字段删除。

（ 3 ） 已退出是显示记录为过期的且交易动作为本息 / 本金返还的交易记录。

（ 4 ） 点击投资合同去查找对应的用户信息表从而生成投资合同。

**3.4.5 交易记录**

模块名称：交易记录

需求描述： 作为一名用户，我想要查看每笔资金的交易记录，以便于知道每笔交易的内容和结果 。

相关方案：交易记录记录的是所有这个账号的回收款的所有记录包括月息，付款，收回本息。这个实现只要去历史交易表查找这个用户的所有历史交易记录，然后按照时间逆序排好，根据相应的分类写好返回给前端的参数。

**3.4.6 银行卡**

        （ 1 ）银行卡在实名验证时就已经进行绑定银行卡操作，若为实名认证则得去先进行实名认证页面，若已经实名认证，直接查看到该用户的实名认证字段就跳转到添加银行卡页面，这里是添加第二张银行卡，然后再手机号验证这部分，跟之前的手机号验证码验证部分都一样，只不过通过手机号可以一天收到 50 条验证短信。

        （ 2 ）解绑

模块名称：银行卡

需求描述： 作为一名已经绑卡的用户，我想要解绑现在绑定的银行卡，以便于更换绑定新的银行卡

相关方案：解绑可以解绑任意一张银行卡，只要确认好手机号即可。

**3.4.7 意见反馈**

模块名称：消息中心

需求描述： 作为一名用户，我想要反馈使用时的感受和建议，以便于改善使用时的不周之处 。

相关方案：意见反馈这一部分是我们对一些意见的提出，当编辑好点击发送时，我们会将该意见记录到对应该用户的意见记录表中，同时记下数据库收到该意见的时间。