# 云计算 1607C 田常平

# JAVA工程师

## 项目介绍

|  |
| --- |
| 我们当时主要做的是一个融信网p2p平台，平台是由借贷双方自由竞价，撮合成交。投资人获取利息收益。贷款人到期偿还本金。平台不吸储不放贷，仅收取手续费，比如提现手续费、还款利息管理费等。通过该平台，让用户更安全同时也更快的进行资金的借款和还款。系统主要包括用户开户、登录、个人中心、我要借款、我要理财，用户充值、提现功能。主要采用dubbo+zookeeper进行服务之间的调用。数据库采用的是mysql。数据访问层采用mybatis，控制层采用的是springmvc，安全登录采用springcecurity，使用JWT解决用户跨域访问问题。前端技术采用的是freemarker模板引擎以及jQuary+bootstrap技术。 |

## **功能**

|  |
| --- |
| 用户开户 当用户完成平台注册之后，使用httpclient调用存管银行接口，会让用户在存管银行进行实名认证，需要用户填写要绑定的银行卡，真实姓名等相关信息。存管银行会为用户开通存管账户。之后的用户借款、投资、充值、提现都是通过存管银行进行资金流转。 我要借款 首先判断借款人在借款申请前基本资料里的必填字段是否齐全，如不齐全，提示需要补充的信息，并跳转至基本资料中进行补充。新建时，需要提交借款信息：借款金额，借款用途，借款期限，还款方式（等额本息和等额本金）等。用户提交借款申请成功后，提示工作人员将在1~3个工作日内审核并联系。工作人员审核通过后，在后台进行借款信息上传，发布。 我要充值 用户通过点击我要充值，进入充值页面，页面分为快捷支付，支付宝充值以及跨行转账，当时快捷充值金额0.15%收取手续费，当时考虑到成本问题，将快捷支付放到了最后一个。当用户点击某一个充值方式时，比如说支付宝充值，回跳到支付宝充值页面，当时做的是一个二维码支付，通过手机操作，填写绑定的银行卡以及要充值的金额。因为我们在平台开户的时候实际上已经在银行的存管系统进行注册以及实名认证，存管银行会相应的为用户开通账户，我们一般称他为托管账户，而我们在平台上的账户我们一般叫做虚拟账户，当我们通过支付宝或者其他的充值方式充值成功后，支付宝会发送反馈信息。支付宝会与存管银行对接，将用户的托管账户余额增加，同时平台会实时监听存管银行的接口，当监听到用户的账户余额增加，平台就会将用户对应的虚拟账户余额增加，这样就完成了充值资金的同步。 我要提现 用户点击我要提现时，会跳转到提现列表，列表上有用户的真实姓名，提现金额，实际到账金额，用户的手续费，用户完成提现申请后，后台会根据用户的提现申请进行审核，首先当前用户不能有多个提现申请，还有就是当前实际提现金额不能大于账户的可用余额，当审核通过后，会提示平台审核成功，跳转到存管银行的提现页面，页面上同样也有用户的真实姓名，提现金额，实际到账金额，用户的手续费以及银行卡号等。提现成功后，存管银行会将用户的托管账户钱直接打到用户提供的银行卡上，平台实时监控用户账户，通过存管银行反馈消息，修改用户在平台上的虚拟账户。 风控资料上传（不是自己负责的） 这个模块不是我做的，当时做借款模块时，也涉及到了对风控资料的判断。首先，借款之前完善个人资料，以方便平台进行审核，主要包括个人信息、婚姻状况（已婚、未婚）、子女状况、月收入范围，住房相关等信息。一旦实名认证通过，不能更改信息。所以要谨慎填写。  资料上传：本人身份证原件（正、反两面照片）、个人信用报告、最近连续3个月收入证明、房产证明、住址证明、车产证明、结婚证明、学历证明、个人头像。【注：资料上传时选择下拉列表内容，鼠标触发时间展示提示信息，资料上传内容参见excel.】，上传图片使用的是uploadify.js插件上传。 我要还款（不是自己负责的） 还款模块：借款成功之后，会生成还款对象和回款对象。借款人根据还款对象来分期，首先判断当前用户账户余额是否大于还款金额，小于则提醒用户账户余额不足，去充值界面充值，假设账户金额大于还款金额，点击还款还款时需要更新几个表的数据，比如投资人账户可用余额增加，代收本金减少，生成账户成功收款流水，若是最后一期，则还款金额应该等于总金额减已经还款的金额，同时还款人的还款状态也要取消等。 |

## 开发中遇到的技术难点以及解决办法

|  |
| --- |
| 首页借款信息是怎么展示的 |
| 我们在项目中首页展示投标信息时用到了redis。因为投标信息比较多，属于热数据，所以我们当时考虑使用redis来存储。key是string类型的bidrequest字符串，value是一个list集合存储的是所有标的信息。通过redisTemplate.opsForValue().set()方法存储。当用户投标信息审核成功，准备发布时，会将投标信息存储到redis里面。以便于在前台展示。为了保证数据的一致性，每次对投标信息改变时都要同步到mysql和redis中。 |
| **用户的安全登录时怎么实现的** |
| 我们使用springsecurity实现用户的安全登录。编写配置文件spring-security里面。我记得有个http的标签，可以对静态资源放行，并设置登陆成功的跳转页面，默认的登录页面。当用户发送登录请求时，通过配置文件。我们自定义一个认证类，实现UserDetailsService接口。重写loadUserByUserName方法，然后创建角色列表，将在配置文件里面配置的角色放到列表里面，根据用户名去数据库查询是否存在该用户，如果存在将用户的密码与输入的密码进行比对，返回userDetails。这样就完成了安全登录。 |
| 验证码的消息是怎么获取到的 |
| 我们把用户的手机号，随机生成的四位验证码和在阿里云注册时的签名、模板等参数存到map里面，通过rabbittemplate存到RabbitMQ里面，消费者通过RabbitListener注解监听生产者，一旦监听到消息，获得到参数，就调用阿里大于的短信接口，完成发送短信的功能。 |

## 项目技术亮点分析

|  |
| --- |
| 手机发送验证码 当用户点击我要注册时，会跳转到注册页面，页面上需要输入用户的手机号，获取验证码，我们使用了阿里大于和云通讯两个短信平台来实现发送验证码。就比如说阿里大于，我们调用阿里大于接口，传递手机号，随机的四位验证码，和在阿里云注册时的签名、模板等参数，来完成验证码的发送。为了防止用户多次点击获取验证码，我们把验证码以string类型形式存储到redis中，key是用户的手机号，value是获取的验证码。存储redis有四个参数，key、value、过期时间和时间类型（小时、分钟、秒等）我们当时设置的验证码过期时间为60秒。在60秒之内用户不可以对同一个手机号重复发送验证码。当用户输入验证码之后，会向后台发送请求，判断验证码是否正确。因为当前验证码存到了redis里面，所以我们从redis里面获取验证码，因为设置了过期时间，所以也能防止用户输入过期的验证码。 JWT token验证 为了保证接口的安全性，我们使用JWT实现跨域访问，当用户登录成功后，后台生成一个token，就是一个String字符串，这个token包含一些信息，比如当前用户的唯一标识id，用户的手机号，Token的签发时间以及token的过期时间。过期时间我们设置为7200秒。  将这个登陆成功状态信息和token一起返回给前台。前台获取到这个token之后，每次请求接口时，都在请求头携带者token。我们在后台通过拦截器拦截到前台的请求，解析token，如果解析成功，将token重新放到request里面，当调用后台接口是，会首先判断request里面有没有token，如果token存在，可以接做接下来的操作，如果不存在，则向前台响应对应的状态码以及错误提示。 |

## 项目答辩中亮点简历审核评定标准

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **考核内容** | **简历样式是否规范** | **项目内容是否完整** | **项目中是否有亮点** | **技术难点解决办法是否高效** | **存在问题** |
| **项目经理 （评定）** |  |  |  |  |  |
| **就业处干事（评定）** |  |  |  |  |  |
| 简历是否合格 | | | | 合格 | 不合格 |

## **项目答辩成绩评定标准**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **考核内容** | **亮点简历与项目描述是否匹配** | **项目流程是否清晰** | **学生讲解是否流畅** | **项目介绍是否到位** | **技术亮点考核是否合格** | **存在问题** |
| **项目经理 （评定）** |  |  |  |  |  |  |
| **就业处干事（评定）** |  |  |  |  |  |  |
| 项目答辩是否合格 | | | | 合格 | 不合格 |  |