# 云计算 1607C 李强

# JAVA工程师

1. 项目介绍(按照实训二做的项目写)

|  |
| --- |
| 简单介绍一下你这个项目 我们当时主要做的是一个p2p金融平台，由借贷双方自由竞价，撮合成交。资金借出人获取利息收益，并承担风险；资金借入人到期偿还本金，网络信贷公司收取中介服务费。主要包括借款人和投资人之间借款和投资，还有通过后台管理人员审核完成相应业务。我们这个平台主要分为前台后台两个系统。前台系统主要包含注册登录，个人中心，借款资质申请，我要借款，充值，投标，提现等功能。后台系统主要包括用户管理、安全管理、审核管理、平台管理等，其中最重要的是审核管理，它是对借款人资质审核，以及对账户资金变动的相关审核，比如发标前审核、满标一审、满标二审等。技术架构采用的是前后端分离开发。技术栈主要采用ssm框架开发，持久层采用的是mysql和redis，数据访问层采用mybatis，控制层采用的是springmvc和springcecurity，前端技术采用的是freemarker和bootstrap技术。 |

1. 功能模块介绍(按照自己得实际项目写)

|  |
| --- |
| **你主体要负责那几个模块** 我主要负责满标一审、满标二审以及生成还款信息这三个模块。 能说下您主要负责这些模块的简单流程吗？3.1我要借款 如果我们的用户想要借款，首先会通过前台页面点击我要借款按钮跳转到我要借款首页，首先我们会判断用户是否登录，若是用户在登录的状态下跳转到我要借款首页，那么，我们会先判断用户是否满足我们的借款条件，借款条件有是否填写个人资料、是否进行实名认证、风控资料分数是否达到可借款分数，以及是否进行视频认证，如果都满足了这些申请条件，用户就可以申请借款，如果当前用户已经有一个借款申请还在申请流程中，那么就不能再次申请借款。填写完借款相关信息之后将数据传到我们的后台，比如借款金额、借款利息、借款期限、还款方式、最小投标等一些信息，那么我们的后台会再次判断借款人是否满足我们的借款条件和当前没有借款在审核流程中，然后再判断借款申请信息中的借款金额是否大于平台最小借款金额和是否小于剩余信用额度；借款利息是否在5%和20%之间；最小投标金额是否大于平台规定的最小投标金额。如果这些条件都满足的话，那么会生成一个借款对象保存到数据库中，然后再为当前借款人添加借款状态码，代表当前用户在借款中。这时我们的后台管理人员登录后台管理系统点击发标前审核就会出现申请借款的相关信息，后台管理人员通过审核借款人的申请信息进行审核通过或者审核拒绝操作。如果审核通过修改标的状态为招标中，在设置相应的风控意见，前台投标页面就会展示这个标的信息，如果审核拒绝，修改标的状态为发标前审核拒绝，然后再去掉用户借款状态码，这是在用户登录的状态下。若是没有登录，展现给用户的是个静态页面，点击申请借款没有任何效果，这就是借款发标前相关操作。 3.2投标 如果投资人想投资的话，在前台客户系统点击我要投资就会跳转到我要投资主页，在该页面可以看到正在招标中的招标信息，投资人可以点击查看按钮去查看该标的详细信息，包括借款金额、年化利率、可得利息、借款人信息、材料信息、还款情况等信息，投资人可以根据标的信息选择合适的标进行投标。如果投资人和该标的借款人是同一人，所以在标的详情页面展示的是撤销借款按钮而不是马上投标按钮，我们平台是不允许自己给自己投标的，当投资人想要投资的时候就可以在相应的投资框输入投资金额进行投标，我们平台规定每个投资人最多对一个标的投标次数不超过5次，并且对一个标的总投标金额不超过该标的总金额的20%，这也是为了让其他投资人进行投标并获得收益。投资人输入投标金额后前台会对投标金额进行校验，如果投标金额大于该标所需的金额或者小于最小投标金额，那么会有提示用户投标金额超过借款标金额或者投标金额小于最小投标金额等信息。在输入符合的投标金额后点击马上投标按钮，弹出提示框让投资人输入交易密码进行投资，之后我们的后台会对传入的投标金额进行判断，判断该标的借款状态是否在招标中，当前用户的账户金额余额是否大于等于投标金额，投标金额是否大于等于最小投标金额，投标金额是否小于等于借款剩余投标金额，还有当前用户不是借款的借款人，在满足这些条件后会生成一个投标对象保存到数据库当中，并且生成这次投标的一条投标流，对该次投标做一次投标记录，并修改借款相关信息。之后在判断该标是否已经投满，如果已经投满，我们将该标的借款状态改为满标一审状态，等待后台管理员的满一审核。 3.3满标一审 对于借款主要包括几个重要的对象：  bidrequest：借款信息  bid：投资人每次的投标记录  Accountflow：账户流水  比如说您想要在我这里借款，我是投资人，当我投资的金额达到您的借款金额时，就进入了满标一审的状态，在满标一审审核时，首先后台要对您的信息再次核对，是否存在当前借款信息，是否处于满标一审状态，如果不满足条件，前台提示用户不满足条件，进行资质审核  如果都没有问题，当管理员点击审核通过按钮时，借款对象(bidrequest)的state状态就变成了满标二审状态。同时生成审核历史对象。  当管理员点击审核拒绝时，说明该借款进入了满标拒绝的状态，需要退换投资人的钱，因为涉及到了钱的变动，我们在这里有一个账户的流水对象accountflow，这个对象包含当前流水金额，当前流水的状态以及流水的备注信息。用于记录退还投资人的投资的流水，同时该借款人的个人信息中的借款状态移除。借款人可以再次发起借款，投资人也可以再次投资。 3.4满标二审 当借款进入满标二审审核时，也要对信息进行审核，判断是否有该借款对象以及该标是否是满标二审状态，当后台管理员点击的是审核通过，此时借款人借款成功，那么我们需要修改借款状态为还款中、增加借款人的账户余额、增加借款人账户待还本息、减少借款人可用信用额度、移除借款人借款状态码、支付借款手续费，平台账户的可用余额增加，我们平台当时是对于不足6个月的借款，将会收取借款金额的2%作为服务费用，至于6个月及6个月到12个月的借款，收取的是4%的服务费用，生成平台账户收取手续费流水。对于投资人来说待收本金和待收利息增加，生成成功投资流水，同时我们考虑到后续借款人需要分期还款，所以在成功借款之后生成每一期还款对象(Paymentschedule)和回款对象：还款明细PaymentscheduleDetail：回款明细，还款对象是针对与借款人的，回款对象是针对投资人的，还款对象包含每一期借款人需要还款金额，而回款对象包含投资人每一笔投资生成的本金和利息，这样便于投资人查看每一笔资金的收益情况（生成还款和回款对象）。  当管理员点击审核拒绝时，说明该借款进入了满标拒绝的状态，需要退换投资人的钱，因为涉及到了钱的变动，我们在这里有一个账户的流水对象accountflow，这个对象包含当前流水金额，当前流水的状态以及流水的备注信息。用于记录退还投资人的投资的流水，同时该借款人的个人信息中的借款状态移除。借款人可以再次发起借款，投资人也可以再次投资。 **3.5还款** 借款人根据还款对象来分期还款，首先判断当前用户是否有还款能力，如果没有还款能力，就要提醒用户账户余额不足，去充值界面充值(充值界面是另外一个同事负责开发的)，  如果账户余额充足，要生成当前这一期还款成功的账户流水，并且平台收取利息管理费，我们平台当时费用是利息的5%，投资人账户的可用余额要增加，代收的本金要减少，也要生成账户成功收款流水，如果当前还款是最后一期，为保证金额的一致，最后一期需要还的金额等于总金额减去已经还款金额。同时借款人的还款状态也要取消。这样就完成了借款人的还款 |

1. 开发中遇到的技术难点以及解决办法

|  |
| --- |
| **难点1：**你们公司做的这个平台，投资者可以对一个借款一直投资吗？  解决方法：  我们当时在后台已经判断了，一个人对一个借款不能投标超过5次，同时投资金额不能查过借款的20%，多投资次数有严格限制，不能任意次数投资。  **难点2：**线上用的短信接口被攻击了，我们的用户就收到了很多条垃圾短信，然后就被用户投诉。  解决方法：  1.增加图形验证：即当用户进行“获取动态短信” 操作前，弹出图片验证码，要求用户输入验证码（通过后台随机生成图片\字符验证），然后在调用短信接口。  2.限制发送时长：建议采用限制重复发送动态短信的间隔时长，即当单个用户请求发送一次动态短信之后，服务器端限制只有在一定时长之后（此处一般为60秒），才能进行第二次动态短信请求。该功能可进一步保障用户体验，并避免包含手工攻击恶意发送垃圾验证短信。用户手机号发送到后台后，先在redis中判断手机号key是否存在，存在则不送。  3.单IP请求次数限制：对该用户在规定时间内如30分钟，访问次数是否超值做一个判断（如值为10次），使用redis.incr（key）自增判断实现，一旦用户请求次数（包括失败请求次数）超出设定的阈值，则暂停对该IP一段时间的请求;若情节特别严重，可以将IP加入黑名单，禁止该IP的访问请求。该措施能限制一个 IP地址的大量请求，避免攻击者通过同一个IP对大量用户进行攻击。  。。。。。。。。。。。 |

1. 项目技术亮点分析

|  |
| --- |
| **一．Dubbo+Zookeeper技术：**  **（1）Dubbo介绍：**  Dubbo是一个分布式服务框架，致力于提供高性能和透明化的RPC远程服务调用方案，SOA服务治理方案。简单的说，dubbo就是个服务框架，如果没有分布式的需求，其实是不需要用的，只有在分布式的时候，才有dubbo这样的分布式服务框架的需求，并且本质上是个服务调用，说白了就是个远程服务调用的分布式框架。告别Web Service模式中的wsdl,以服务者与消费者的方式在dubbo 上注册)。  其核心部分包含:  1.远程通讯，提供对多种基于长连接的NiO框架抽象封装，包括多种线程模型，序列化，以及“请求一响应”模式的信息交换方式。  2.集群容错: 提供基于接口方法的透明远程过程调用，包括多协议支持。以  及负载均衡，失败容错，地址路由，动态配置等集群支持。  3.自动发现:基于注册中心目录服务，使用服务消费能动态查找服务提供方,使地址透明,使用服务提供方可以平滑增加或减少服务器。  **（2）Zookeeper介绍：**  Zookeeper是一个分布式开源框架，提供了协调分布式应用的基本服务，它向外部应用暴露一组通用服务——分布式同步（Distributed Synchronization）、命名服务（Naming Service）、集群维护（Group Maintenance）等，简化分布式应用协调及其管理的难度，提供高性能的分布式服务。ZooKeeper本身可以以单机模式安装运行，不过它的长处在于通过分布式ZooKeeper集群（一个Leader，多个Follower），基于一定的策略来保证ZooKeeper集群的稳定性和可用性，从而实现分布式应用的可靠性。  **（3）Dubbo+Zookeeper工作原理：**  首先服务Container在启动时，它会将相关的服务注册到Zookeeper下，zookeeper注册中心具有服务的注册与发现的功能，本质上讲，他维护了一个服务的白名单，服务消费者在启动的时候,会向zookeeper注册中心订阅自己的服务，此时，注册中心返回服务提供者提供的地址列表给消费者.服务消费者，从提供者地址列表中，基于软负载均衡算法，选一台提供者进行调用，如果调用失败，再选另一台调用。所有提供者在zookeeper上创建的节点都是临时节点,利用的是临时节点的生命周期和客户端会话相关的特性,因此一旦提供者所在的机器出现故障导致该提供者无法对外提供服务时,该临时节点就会自动从zookeeper上删除,zookeeper会把最新的服务推送给消费者.消费者每次调用服务提供方是不经过zookeeper的,消费者只是从zookeeper那里获取服务提供方地址列表,存入内存中,所以当zookeeper宕机后,不会影响消费者调用服务提供者,影响的是zookeeper宕机之后,如果提供者有变动,增加或者减少,无法把最新的服务提供者地址列表推送给消费者。  服务消费者和提供者，在内存中累计调用次数和调用时间，定时每分钟发送一次统计数据到监控中心。  **（4）Zookeeper选举机制：**  假设有五台服务器组成的zookeeper集群,它们的id从1-5,同时它们都是最新启动的,也就是没有历史数据,在存放数据量这一点上,都是一样的.假设这些服务器依序启动：  1) 服务器1启动,此时只有它一台服务器启动了,它发出去的报没有任何响应,所以它的选举状态一直是LOOKING状态。  2) 服务器2启动,它与最开始启动的服务器1进行通信,互相交换自己的选举结果,由于两者都没有历史数据,所以id值较大的服务器2胜出,但是由于没有达到超 过半数以上的服务器都同意选举它(这个例子中的半数以上是3),所以服务器1,2还是继续保持LOOKING状态。  3) 服务器3启动,根据前面的理论分析,服务器3成为服务器1,2,3中的老大,而与上面不同的是,此时有三台服务器选举了它,所以它成为了这次选举的leader。  4) 服务器4启动,根据前面的分析,理论上服务器4应该是服务器1,2,3,4中最大的,但是由于前面已经有半数以上的服务器选举了服务器3,所以它只能接受当小弟的命了。  5) 服务器5启动,同4一样,当小弟。  **（5）项目中如何使用Dubbo+Zookeeper：**  要想实现dubboServer和dubboClient之间完成调用，首先服务端和客户端需要一个接口来公用。可以有两种方式：  1)接口写在一个应用中，服务端和客户端都对这个应用进行引用  2)更多的时候，两个应用是不太可能引用同一个应用。这个时候应该把接口以jar包的形式给Server和Client共同使用，然后，Server对jar包中的接口进行实现，并且向dubbo进行注册，而Client端调用jar包中的接口中的方法名，dubbo会去自动的寻找这个由Server端注册的方法的实现（如果使用zookeeper则通过spring的<dubbo:registry address=”\*\*”/>标签，将服务发布，客户端也使用该标签实现订阅）。  **二．redis技术：**  发送手机验证码：  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。  redis的数据类型主要有:String,hash,list,set,zset  redis持久化方式:RDB快照方式和AOF日志方式  RDB效率比较高，但是会丢失数据，存数据的格式是dump.rdb  AOF是把增删改操作记录下来，不会造成数据丢失，效率相对RDB较低  我们在项目中呢是两者结合使用的，服务器运行的时候使用AOF方式记录日志，一定时间点（定时任务）用RDB快照持久化，把此时间点之前的日志删除。如果服务器duwn机了。先根据RDB快照做恢复，再根据AOF方式做恢复。  （3）如何用redis做一个消息队列  可以用list，用lpush+rpop实现队列功能,lpush从左往右放，rpop是从右开始取并移除相对应的值 |
|  |

1. 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 |

1. 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 |  |