# 云计算 1607C 刘长根

# JAVA工程师

1. 项目介绍()

|  |
| --- |
| 小飞额信用贷是个人对个人借款的网贷系统，它的模式为网络信贷公司提供平台，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。项目分为前台系统、运营商后台管理系统。  项目分为前台系统、运营商后台管理系统。前台系统有：登录模块、注册模块、首页模块、项目投资管理模块、账户模块、借贷项目模块、资产详情模块、投标模块。运营商后台管理系统：用户管理模块、安全管理模块、审核模块、报表管理模块。项目使用的框架是Spring、SpringMVC、和MyBatis；Dubbo+Zookeeper作为服务调度和注册中心,数据库使用MySQL对用户的信息进行存储，使用Dubbo和Zookeeper实现服务之间的调用，前端使用jQuery、FreeMaker和Bootstrap对页面的展示，使用Redis对数据进行缓存处理。 |

1. 功能模块介绍

|  |
| --- |
| 1. 注册模块：   首先通过username查询username在数据库中的数量count,对返回的数据进行判断；如果count<=0时，则表示用户不存在，创建用户信息对象，对这个对象进行赋值，创建新用户,并且返回新增主键，并保存这个对象和对象的状态，进行用户的添加，为了考虑安全性，则对密码进行MD5加密，同时根据新增主键关联账户初始化信息；否则,就抛一个AdException异常，并提示“用户已存在”.  在前台使用jquery中的validite的验证插件对用户名和密码的长度进行验证，遵循rule规则进行验证，然后用message属性对验证信息进行反馈。  另外，在进行用户注册时，用户开户信息是和用户注册时一起创建的，在开户时调用第三方的接口，第三方返回给用户需要填写的账户信息表，填写完返回数据给第三方，进行开户。   1. 登录模块：   登录模块我们使用的是springsecurity技术实现用户的安全登录。在登陆前，首先查询用户名和密码是否存在，如果返回的对象不为null，则把对象放到封装类的session中，否则返回错误信息的状态码，然后在web.xml中配置一个“管理请求内容的监听器”，对session的状态进行监听，登陆成功后跳转到个人中心页面。   1. 借款模块：   首先获取当前用户的id,查询出当前的用户信息，判断用户是否具有借款资格，如果没有则进行对用户基本信息、实名认证、风控资料的认证、视频认证，如果返回的是false,给出错误信息的提示；返回的true时，把最小借款金额和最小投标金额放置module中，然后从session中获取用户的id，查询账户信息，使用hashcode值校验账户是否异常，如果异常，进行错误信息的提示；如果合法则跳转到借款页面，对借款信息进行填写，最后发布标的。   1. 借款审核模块：   根据用户id查询当前用户的申请表（申请表联查用户信息表，获取申请人的信息），判断申请是否是处于发布状态，如果是true,创建一个审核对象，对其属性进行赋值，判断前台传来的状态值是否是审核通过的状态，审核通过，通过状态码变成招标中的状态。如果返回的是false,审核状态改变成拒绝状态。   1. 满标审核模块：   首先判断当前借款是否存在，判断当前标是否处于满标一审的状态，如果满足条件时，创建审核历史对象，修改风控意见，如果审核成功，设置状态为满标二审；如果审核拒绝，将借款状态修改为满审拒绝状态，将投资者的钱退回给投资者将投资者的钱退回给投资者,并记录退款账户的流水。进入满标二审，首先判断当前标是否处于满标二审的状态，修改借款对象的状态为还款中；对借款人收款的操作：借款人可用余额增加，生成账户流水记录，同时修改待还本息和修改剩余信用额度，移除借款人在借款中的状态，借款人支付手续费，借款人可用余额减少，生成借款成功支付手续费账户流水记录；另外，平台收取手续费，平台账户可用余额增加，生成平账户生成借款手续费流水；最后遍历投标列表，获取投资者账户，投资者账户冻结金额减少，生成成功投标的账户流水，计算每一次投标的待收利息和本金。 |
|  |

1. 开发中遇到的技术难点以及解决办法

|  |
| --- |
| 难点1：springsecurity是如何实现安全登录和权限认证的？  解决方法1：  首先通过Http请求，自定义url和所需权限的对应关系，在springsecurity配置文件中实现Http请求的权限验证检查用户登录授权阶段获取到的用户权限是否可以访问该url,通过调用AccessDecisonManager的相应接口完成权限认证,配置vote()方法来进行投票(成功返回1，失败返回-1)，根据投票结果及一定的策略决定该用户是否有权限访问该url，只要有一个投反对票，即vote（）方法失败返回-1，则拒绝访问url,否则通过权限验证；另外用户发送登录请求时，通过配置文件，我们自定义一个登录认证类，实现UserDetailsService接口，重写loadUserByUserName方法，数据库中根据用户名查询对应的用户信息，返回对应的实体对象，如果存在对用户的密码与输入的密码进行比对，返回userDetails，这样就实现了安全登录。  难点2：项目中使用的是MySQL，MySQL是如何进行优化的？  解决方法2：  数据库优化：数据库优化吧我觉应该从硬盘、内存和网络带宽考虑，提高硬盘的读写速度，增大带宽提高吞吐量，增大服务器内存，可以采用读写分离，降低单台数据库的访问压力，查询的时候控制数据量的大小，返回更少数据，减少交互次数，减少cpu及内存的开销。  Sql语句优化：如果一个表中数据量过大我们可以采用横切割，如果一个表中字段过多，我们可以采用纵切割，适度冗余减少表关联查询，避免过多的联查，设计合理的表关系，适当建立索引，但是不能每个字段都建立索引，因为索引也要占用一定的物理存储空间，而且索引也需要动态维护，增加索引虽然可以提高查询速度，但是如果索引过多就会降低增删改的速度，写sql的时候也要注意，尽可能不要写一些让索引失效的sql，例如：索引的字段不能为空，不能进项模糊搜索，不能进项逻辑运算，不能使用函数，给索引查询的值应是已知数据，不能是未知字段值。  难点3：该项目是分布式项目，你是如何解决分布式事务的？  解决方法3：  1.两阶段提交方案/XA方案  事务管理器主要负责协调多个数据库得事务，去执行各个的本地事务，然后就可以正式提交整个事务了。如果有一个数据库不同意得话，就需要回滚。这个方案他适合域单体应用，会连接多个数据源，会跨库操作，会严重依赖于数据库层面操作事务，操作效率低，不合适高并发得场景，如果去实现的话，可以使用spring+JTA。  2.TCC（补偿事务）方案（具有强一致性）  T:TRY :对各个服务得资源做确认，及对资源的锁定或者保留  C:Confirm：各个服务实际得操作  C:Cancel：只要有其中一个服务得业务方法出现出错，需要补偿，即将已经执行成功得操作回滚  比如：银行转账  T:将俩个银行账户中得资金冻结。  C:实际进行转账，A 减钱 B加钱  C:如果有任何一个银行得操作失败，需要回滚进行补偿，比如 A已经扣减完成了，B在增加资金得时候失败了，此时就把A的钱加回去。  这个方案在回滚操作时严重依赖于开发人员的代码进行回滚或者补偿的，代价比较大，使用的场景是有限的。一般是要求数据一致性特别高的时候，才采用这种方案，核心中的核心业务。  3.本地消息表:  （1）A系统在自己的本地事务操作的同时，会插入一条数据到消息表;  （2）接着A系统会把这个数据发送到MQ,A系统就是消息的生产者;  （3）B系统接收到消息（B系统是个消息的消费者），在自己的本地事务中，往自己的本地消息表中插入一个数据，同时执行自己的业务如果这个消息被处理过了，那么这个事务就需要回滚;  （4）B系统执行成功后，会更新自己的本地消息表及A系统的本地消息表的状态;  （5）如果B系统处理失败了，首先不会更新自己的本地消息表，A系统会自己定时扫描自己的消息表，如果发现数据没有改变成功的状态，则会往MQ中再次发送该消息，让B系统再次处理;  （6）这个方案保证的是消息的最终一致性，哪怕B失败无数次，A永远不放弃，一直发送消息，直到B成功;  这个方案严重依赖于数据库的消息表来管理事务的，不合适于高并发的场景，实际使用少。  4.可靠消息得最终一致性:  （1）A系统先发送prepared消息到MQ,如果消息发送失败，就取消操作;  （2）如果发送成功了，MQ会给A发送回调消息，那么A就处理自己的本地事务;如果失败了，告诉MQ进行回滚;  （3）A系统给MQ发送确认消息，此时B系统就会接收到这个消息，然后执行本地事务;  （4）MQ会自动定时轮询所有prepared消息回调你的接口，MQ会定期扫描消息集群中的事务消息，这时候发现了Prepared消息，它会向消息发送者确认，所以生产者需要实现一个check接口，MQ会根据发送端设置的策略来决定是回滚还是继续发送确认消息。这样就保证了消息发送与本地事务同时成功或同时失败。 |
|  |

1. 项目技术亮点分析

|  |
| --- |
| 1. 项目框架：Dubbo+Zookeeper作为服务调度和注册中心。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，以及SOA服务治理方案。最大的特点是按照分层的方式来架构，使用这种方式可以使各个层之间解耦合，Dubbo基于Zookeeper实现数据发布订阅、负载均衡等功能，Dobbo节点服务提供者和服务消费者，服务提供者将自己提供的服务注册到 ZooKeeper 中心，服务的消费者在进行服务调用的时候先到 ZooKeeper 中查找服务，获取到服务生产者的详细信息之后，再去调用服务生产者的内容与数据。   （2）使用Redis缓存数据：在项目的个人中心展示投标信息时我们考虑到高并发的问题，使用数据库存储数据的时候，如果业务量过大时，会造成数据库较大的压力，因此我们考虑使用Redis来对数据进行缓存。Redis是key和valued的方式进行存储数据的，而且是按照原子性进行操作的，同时**redis会周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件保证数据的持久化,**key是string类型的字符串，value是用list集合存储投标的信息，通过注入redisTemplate调用opsForValue().set()方法存储数据。运营商后台对投标信息进行审核，如果审核通过，则把投标信息存储到redis里面，前台从redis中获取数据，在个人中心页面进行展示。 |
|  |

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 |  |