Online Bank Design Scheme

# 业务选择

网上银行设计业务体系较为复杂，比较适合使用分布式系统，目前系统设计三块业务。

1. 查询、转账：转账可为跨行转账
2. 借贷服务：借贷涉及模块、风控（校验，调用接口查看征信信息，或者本行校验，是否有未完成借贷服务）
3. 理财产品：包括行内的以及代理的。

目前仅仅涉及三块内容。

# 技术选型

分布式系统，微服务架构，选择spring-cloud。包括服务治理、熔断、声明式调用、restful调用、消息驱动、网关、鉴权中心、分布式服务跟踪，分布式配置中心。

# 系统架构设计

系统：

OnlineBank-geteway:api网关 spring-cloud-zuul

OnlineBank-sso：单点登录 –前期、只做seesion管理、后期、做角色权限管理。

-----OnlineBank-exceptionHandler:统一错误处理，消息驱动

-----OnlineBank-router:路由yue

OnlineBank-customer-search:查询余额（接口调用频繁）

OnlineBank- customer-handler-web:用户操作、转账、借贷、理财、支付

onlineBank-manage-transfer:转账后台服务-区分跨行和本行（本行直接操作数据库，跨行的调用其他银行提供的接口）

onlineBank-manage-outward-pay:转账接口，这个以消息队列的形式接收跨行转账的消息

转账逻辑：转账操作-🡪后台校验是否本行-🡪本行直接行内进行账户加减操作—>非本行，调用其他银行接口-🡪其他银行进行账户加操作，并且记录下操作银行，操作账户，操作金额—>返回给本行—>操作成功，则进行账户减操作，并且发送一条消息给outward-pay系统—>outward-pay系统将消息归类，统一进行转账操作（按不同的银行进行转账），并记录表操作记录

OnlineBank-manage-loans:借贷后台服务，用户操作调用该系统接口（1、接口返回借贷资质，根据风控中心返回的数据来判断该用户的额度）

OnlineBank-manage-products:理财产品，行内

OnlineBank-manage-out-products:理财产品外部代理（接口调用restTemplate,有自己的表，方便生成季度报表）

理财产品逻辑：客户选择理财产品—>后台评测理财产品的来源方—>本行

onlineBank-manage-message-push:消息推送中心(websockt、feign、netty)

onlineBank-manage-message-web:消息中心，编辑消息，统一在消息推送推送，定时消息和即时消息

onlineBank-manage-job:定时任务中心(zookeeper选举)

onlineBank-manage-pay:支付中心，小额支付中心（类似于支付宝的付款）

onlineBank-manage-risk:风控中心（借贷调用风控中心-风控返回结果-根据结果判断是否放款操作-是->放款）

onlineBank-manage-loans-pay:放款中心，借贷后台

借贷操作：用户->资质（先查表，看用户是否已经经过资质评测，如果没有，则进行风控资质评测，和后面的风控调用同一接口，验证：1、当前用户是否有其他未还贷款。2、当前用户是否有过逾期记录）->借款操作->风控（验证：1、当前用户是否有其他未还贷款。2、当前用户是否有过逾期记录。）-）是否放款->评测通过则调用放贷中心

onlineBank-manage-common:公共类

onlineBank-restful-commom-web:公共框架中心，包括请求进入的logger日志，操作日志记录操作

onlineBank-restful-common-service

技术汇总：

Springboot:1.5.9

Springcloud—eurake、feign、ribben、分布式配置中心、stream、zuul、熔断

Zookeeper:选举、分布式锁

Redis:分布式锁：redisTeplate、springboot的redis仓库

Mongo:报表、理财产品报表

Netty:sso和message

Websocket:message消息推送

Shiro—security—smart:三个

Mybatis

rabbitMQ:

kafka:

fastjson:

Quartz：定时任务

数据连接池：druid

日志管理：slf4j、logback按时间拆分日志

前端：vue.js（简单的操作）