中证报价APP服务端升级改造总结

# 框架

框架根据业务场景而定，原有主要场景为业务以及数据透传，类似于一个前置系统，前置系统应合理使用缓存，一是提高系统的响应速度，二是减少数据库的压力；对访问进行控制，尽量减轻对核心业务的访问压力；数据与页面分离，对开发速度进行提升。

## 当前框架

主体是springmvc+spring4.3.2.RELEASE+mybatis3.4.0+activemq5.10.0；

数据库使用mysql；

日志实现为日志：slf4j1.7.21+logback1.1.7；

单元测试使用junit4+spring-test；

缓存jedis 2.8.1

# 改造点

## 整理pom

升级spring版本为4.3.2.RELEASE

添加logback以及转换桥(slf4j1.7.21+logback1.1.7)

升级 mybatis版本与spring-mybatis(mybatis3.4.0+spring-mybatis1.3.0)

完善httpclient(写全)(httpclient4.5.2 + httpcore 4.4.5)

剔除无效的log4j包(主要是极光推送中不同版本以及重复引用)

将activemq-all包拆解(自带slf4j的实现，与logback冲突)，

使用activemq-core（5.7.0）+spring-jms+activemq-spring（5.10.0）进行替换

添加版本打包

添加不同环境打包local/test/prod

## 接入mq

activemq all包的拆解，修改成spring的jms，便于扩展

封装成基础服务(写父类抽象类，公共方法，子类继承并实现父类的抽象方法)

## 切换log框架

引入logback包

定义logback.xml

分目录、级别输出:报价请求接收 本地请求响应 全部log

整理所有的输出，尤其是直接System.out.println

冗余调试log

## Httpclient切换连接池

切换连接池（使用类已经引入，需要接入测试一下）

统计响应请求log

请求响应优化方案（封装）

请求头、请求参数、响应结果封装,力图优化接口开发成本

封装成service，减小每次请求需要new对象的成本（先写成静态实现，最后再改service）

## 基础类封装与改造

基础service中，

抽取改造公共方法 转化json，组装公共响应报文等

添加BaseAction，

改造，抽取公共方法(公共参数的封装)

转化json 异常处理（先不改），session判断，请求参数封装等

## 请求头user-agent与di验证

公共包头描述：

手机操作系统\_版本/客户端名称\_平台\_pad(orPhone)\_客户端版本号/手机型号

user-agent=Android\_4.0/TDRMPC\_Pad(orPhone)\_clientVersion/LG778;

举例如下：

User-Agent

Android\_4.4.2/OTCTOP\_Phone\_0.0.0.1/HUAWEI P7-L09

di

f8:01:13:fe:22:18/F8:01:13:FE:0F:58/ARMv7 Processor rev 1 (v7l)/ffffffff-8812-a3b3-d8f9-a2420033c587/1080

请求规则（已经引入，没有配置）

拦截器验证(可配置)

改造控制层（默认是拦截还是不拦截，衡量一下修改的个数，前端接口与后台管理不同）

添加拦截器到spring-mvc.xml中

## redis接入

封装redis的基础类

测试类中测试

## 异常封装

封装码封装成配置文件

异常码读入工具类(单例模式) 文件—>内存(类map)—>使用

异常码过滤与转译 工具类中实现方法

引入到控制层以及异常输出

## MYSQL文件资源化

将mapper文件配置到资源目录

查看sql以及优化

## 定时器优化

引入spring的线程池管理包spring-task/quartz

写入公共类，封装成服务

## 代码走查，将代码注释补齐

冗余log

方法实现

## 统一接口返回格式

处理报价返回的异常格式

接口异常，参数异常 统一返回格式

## Mybatis 改造

修改mybatis 写法。使用spring管理mybatis，简化开发

## 参数 配置到数据库

测试和生产不同的参数，方便切换调试，减少出错

# 已修复Bug

## 整理pom