**百威数据查询系统【中间件】**

**一、项目概述：**

根据公司需求，开发一款轻量级，界面友好，灵活的，方便移动查询的系统；并且可根据需要，调整项目架构，实现与第三方接口的对接，实现中间件功能。

**二、模块划分：**

根据日常公司运行规律，分成商品，订单，库存，出库发货，客户，资金6大模块；并增加仪表盘模块，可以提供统一直观的数据展示。

**三、设计思想：**

mvc设计思想

A表现层：html5+css+Jqury+ajax（局部刷新，提供更好的体验）；

B控制层：springmvc；

C业务层：service组件，+redis缓存组件；

D持久层：dao组件,使用mybatis框架+pageHelper组件；

**四、技术架构：**

开发环境：window7+tomcat8.5+sqlserver2008

开发工具：eclipse

开发技术：

Java:开发核心技术；

jquery：简化js（$对象和api）；

html+css+js：提供静动态前端页面效果；

ajax：局部处理页面，提升用户体验度；

spring：（iop、aop）管理相关组件；

iop：控制反转；负责管理Controller/service/dao,维护它们之间的关系；

aop：面向切面编程，可以不修改原有代码，给系统增加新功能；

mybatis：数据库操作核心技术；

pagehelper：查询大量数据时，实现分页展示的技术；

redis：数据缓存技术，提高数据响应速度；

Httpclient : 实现中间件数据上传的功能；

版本控制工具：git

**五、整体规范：**

所有的请求使用ajax方式访问

前端页面采用HTML

请求结果进行json响应【此种格式，既可以实现中间件数据传输，可以作为独立web项目的数据格式】

｛status: , "msg": ,"data": ｝

**六、数据库构建：**

数据库类型：sqlserver 2008R2

**七、详细介绍：**

整个项目，使用spring+springmvc+mybatis框架，实现功能的集成，和可拓展性；

**（一）软件最初构想：只开发后端系统，对接应订货、商之翼的系统后台，作为中间件使用；**

SO：

**1持久层：【dao层+mapper映射文件】**

【旨在获取数据，用最简介的代码，实现数据增删改查，不涉及具体的业务逻辑】：

**A**在整个框架搭建好之后，选择mybatis作为持久层的主要技术；

原因在于，相对hibernate技术，mybatis更加灵活，可以自主选择sql语句，提高增删改查的性能；

**B**通过spring，集成mybatis；在项目文件conf中添加配置文件mybatis.xml；

**C**添加pagehelper组件。

pagehelper是和mybaits配套使用的一个功能组件；可以通过简单的配置，以及少量代码的添加，配合前端实现分页查询；

**D** 数据包：

使用Map封装；map是集合的一种，具有键值对特效，一个key对应一个value，并且容易实现扩容；基础的数据分装，都使用map；

之所以抛弃前期使用的Bean，因为每增加一个新的查询，都需要创建一个新的类来封装数据，还得保证每个类【bean】的属性与数据库中表的属性名一致，这样就产生了更多的代码以及可能产生的资源消耗；一旦数据量多起来后，将会影响到系统的整体效率；

**2业务层：【service层】**【旨在对数据做逻辑处理；】

**A service层**分成2成：

service接口层：对应相应的dao层，提出需要实现的功能；

service实现层：实现具体的业务处理，封装业务逻辑；

1实现统一的封装格式： ｛status: , "msg": ,"data": ｝

util包中建立一个工具类：ResultNote<T>.class；用此类封装任意类型的数据；

2 中间件：

直接在service层中对数据进行处理，通过JSON，将dao层获取到的数据转为JSON格式的数据；

通过httpclient 发送到指定的url；

通过java自带的timer，实现定时上传数据；

**B【aspect】切面技术：**

通过切面技术，可以给全部的service层，增加新的功能；

A logger 日志功能；将系统发生的异常，统一写入硬盘中，以便后期的维护；

B 统计系统性能；

C 动态事务处理器

@Transactional：相当于给service层套上try…catch ，可以捕捉异常，回滚事务等

**C【redis】数据缓存**，将数据存到内存中，减少sql查询次数，提高查询效能：

A 为了提高系统响应速度，在业务层中集合redis的功能；

B 将查询、修改的数据，先保存至redis中；通过时间设置，如间隔2分钟，再从redis中取出发送到前端；

C 面对客户的多次相同查询 ，可以直接调用redis；

**3控制层：**

springmvc：用来接收前台传值，调用service层和持久层，返回数据再通过SpringMVC把数据返回前台；

**（二）增加前端的展示页：**

**1 前端代码：**

1 下载免费前端模版，经过改装，可以快速拥有需要的效果；

2 技术核心：jquery，ajax，js，css，html

3 通过js代码，将所有页面中的公共的部分，如页面中head，foot，navigation 提取成为共用代码；便于维护；

4 所有页面，通过ajax，实现动态展示效果；提高用户体验度。

**2 控制层【controller层】**

1 只负责数据的转换工作，将service层处理好的数据，反馈给前端；

2 转换：@ResponseBody 将map数据，转为json数据。

**Linghao**

**2019-6-4**

备注：

项目源码中，也做必要的文字说明。