莱麦项目架构说明文档

编写人：韩东

时间：2017年04月06日

版本：V1.0

文档变更记录：

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本号 | 变更内容 | 负责人 |
| 2017-04-06 | V1.0 | 编写项目架构说明文档 | 韩东 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**软件以及环境：**

|  |  |  |  |
| --- | --- | --- | --- |
| 软件 | 类别 | 版本 | 备注 |
| Spring boot | 框架 | 1.4.3X |  |
| Jdk | SDK | 1.8.X |  |
| Tomcat | Web服务器 | 8.5.X | Spring boot内置 |
| Mysql | 数据库 | 5.X |  |
| Nginx | 反向代理工具 | 1.7.X |  |
| Centos | 服务器系统 | 6.5.X |  |
| Redis | 缓存框架 | 3.2.X |  |
| ActivityMQ | 消息队列 | 5.X |  |
| Log4J2 | 日志框架 | 2.X |  |
| Druid | 数据库连接池 | 1.0.28.X |  |
| String Data Jpa | ORM框架 | 1.10.X |  |
| Log4jJdbc | JDBC日志框架 | 1.1.16.X |  |
| Maven | 工程管理工具 | 3.3.9X |  |

备注：

X代表高于之前的版本。

文档说明：

该文档主要描述系统的架构，通过架构和文字描述，让开发者理解系统的整个架构。

架构概要说明：

架构基于spring boot框架，会逐渐从1.0迭代到3.0，都是使用maven的管理方式。

1.0主要是业务功能分离，功能之前采用代码依赖的方式；

2.0主要把各个组件之间解耦分离，使用多协议的方式访问服务组件，默认使用http协议，随后根据需要扩展更多的协议，比如web service等；

3.0主要是解决性能瓶颈，使用分布式集群的架构，基于dubbo框架和zookeeper框架或者spring cloud实现。

* 1.0架构图



**1.0架构说明：**

各个服务组件按照业务和功能拆分不同的组件，每个组件是一个独立的spring boot工程，使用代码依赖的方式引入。Web页面和服务器端，采用动静分离的方式，以http的方式调用服务端的接口。用户发出https请求以后，nginx会根据接口的地址，跳转到对应的服务组件，在完成登陆认证授权等一些列认证之后，调用相应的服务。根据业务和功能来划分一个模块，使用独立的spring boot，这些工程使用不同的端口号。最初在并发量比较小的情况下，这些服务组件可以部署在同一个机器上，随着业务和并发量增多，这些组件可以分别部署在不同的机器上，提高性能。

**代码依赖：**

公用使用的部分使用代码依赖的方式，比如bean、service、util、dao等；

**组件拆分：**

根据业务和功能模块进行拆分，比如商品、订单、支付、财务统计，资金清算、自动任务作业这些需要单独分离处理，做独立工程，降低系统之间的耦合度。

**消息队列的使用：**

类似于邮件、短信这些非实时非事务性的业务，采用消息队列的来处理，提高系统的性能。业务层使用jms把数据推送到消息队列组件，短信和邮件就是消费者，从消息队列中获取消息，完成消息的发送处理。

**第三方组件：**

微信、支付宝这些第三方接口，采用独立的maven java项目，用代码依赖的方式加入进来，业务参数不能直接写在第三方的组件里面，业务参数需要传入，第三方工具组件就是一个工具组件，以后可以应用于其他项目。

**日志：**

每个组件会有自己的日志文件，日志文件会被Logstash软件采集或者日志直接输出到Logstash，由Logstash推送到Elasticsearch搜索引擎，然后Kibana结合Elasticsearch搜索日志，分析处理日志。Kibana是日志分析器，Elasticsearch是数据源，Logstash是数据通道。

* 2.0架构：

架构描述：

2.0架构是基于1.0的架构演变的，主要是第三方服务不再使用代码依赖的方式，而且使用远程调用的方式来实现，默认实现http方式的调用，后期根据业务需要，可以使用其他的访问协议，比如web service等。

授权中心：

2.0架构中，增加了一个授权中心，在访问接口之前，需要通过授权中心的认证之后才可以使用服务。

2.0架构图



3.0架构

架构说明

3.0架构主要是分布式集群的方式，使用远程调用的方式，服务提供方

