基本框架说明文档

版本1.0.1

**修订历史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 作者 | 修订章节 | 修订原因 | 修订日期 |
| 1.0.0 | 朱金龙 | 初稿 | 初建 | 2017-06-05 |
| 1.0.1 | 余洋智 | 修订 | 骨架升级 | 2018-07-12 |

## 服务端代码框架结构



**1.1.1接口展示层（facade）**

定义对外提供服务的接口、请求参数对象、应答参数对象，及对外公开的常量、变量和枚举类型。

所有对外服务都必须在该层定义，以后计划使用dubbo来提供服务。

**1.1.2服务接口提供层（web）**

对外提供服务，现阶段使用http方式提供服务接口，接口间调用统一使用json做参数传递。

除了本系统内部调用可在web直接调用service层，对外的服务都只能调用biz层提供的服务

**1.1.3接口实现层（biz）**

对façade定义的接口做业务实现，包括幂等，参数校验，基本的业务规则流程控制和校验等。

impl:调用action，统一处理返回并转换为对应的response对象

action: 业务控制，调用service，下游异常统一转换为BizException返回。

**1.1.4业务处理层（service）**

业务逻辑处理，控制业务对象的状态流转以及事务管理。

repository:数据库操作

bussiness:业务一致性控制

consume: intergration层调用

**1.1.5数据库操作层（dal）**

数据库的持久化操作。

**1.1.6外部接口调用层（integration）**

集成不同后端系统（其他系统）提供的服务接口，提供统一的服务接口供系统内部业务处理层调用。

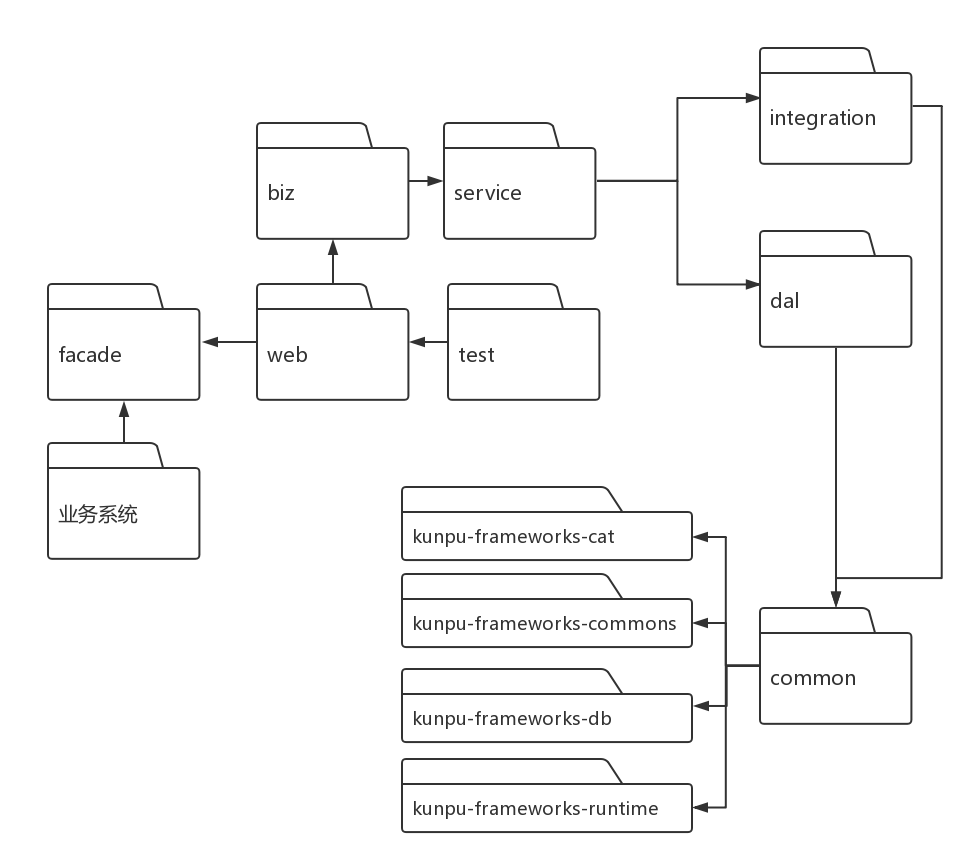
**1.1.7通用工具类层（common）**

提供本系统特有的常用类和参数，包括配置项，枚举，通用工具类以及通用的请求和返回对象等。

**1.1.8单元测试集合层（test）**

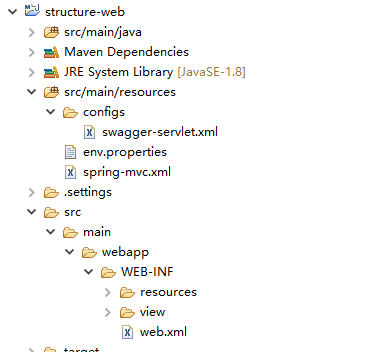
系统单元测试类的集合，方便后续测试。

**1.1.9 包图**

****

## 1.1.10配置视图

**web层：**



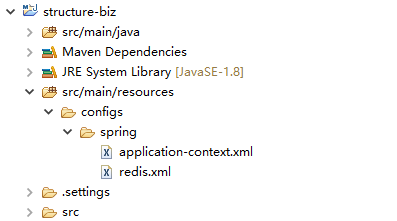
web.xml ：J2EE标准WEB描述文件

spring-mvc.xml：MVC配置文件

swagger-servlet.xml：swagger服务配置文件

env.properties:常规服务配置项文件

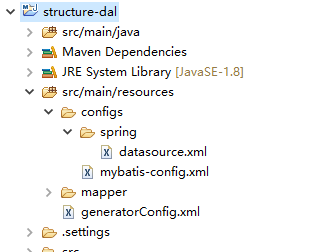
**biz层：**



redis.xml：redis配置

application-context.xml：服务相关配置项

**dal层：**



datasource.xml：数据库配置

mybatis-config.xml：mybatis配置

generatorConfig.xml：mybatis自动生成DO对象和mapping文件配置

说明：

在接收下游系统异常时，若异常或者返回对象为空（包括返回码为空），直接向上抛出异常

否则判断是否返回成功（即返回码为000000），否则把对应的返回码和返回描述直接用异常向上抛出

因为façade要单独对外发布，所以依赖的引用都要在façade的pom.xml中单独指定版本，由于插件的原因，在自动生成项目后请删除façade的pom.xml文件中的以下部分

<parent>

<artifactId>${系统名}</artifactId>

<groupId>com.kunpu</groupId>

<version>0.0.1-SNAPSHOT</version>

</parent>

## 服务运行目录结构

**日志目录**：

/

/data/

/data/logs/

/data/logs/${系统名}/

/data/logs/${系统名}/${包}/

/data/logs/${系统名}/${包}/${系统名}-${包}.log

/data/logs/${系统名}/${包}/${系统名}-${包}-yyyyMMdd.${自动生成序号}.gz

注：日志文件只保留三十天内的，跨日或者超过10M会自动打包，创建新的日志文件，详见logback.xml

注：日志文件只保留三十天内的，跨日或者超过10M会自动打包，创建新的日志文件，详见logback.xml

**tomcat部署目录**：

**/**

/opt/

/opt/services/

/opt/services/${系统名}/

/opt/services/${系统名}/tomcat/

**war包目录**：

**/**

/opt/

/opt/apps/

/opt/apps/${系统名}/

/opt/apps/${系统名}/${系统名}.war

**配置文件目录**：

**/**

/opt/

/opt/configs/

/opt/configs/${系统名}/

/opt/configs/${系统名}/env.properties

/opt/configs/${系统名}/logback.xml

/opt/configs/${系统名}/redis.properties

/opt/configs/${系统名}/datasource.properties

/opt/configs/${系统名}/其他中间件的配置项外化

## 加入基础组件

**1.3.1锁**

**1.3.1.1 基于数据库的锁**

com.kunpu.archetest.service.bussiness.TaskLockBusService

**1.3.1.2 基于redis锁**

com.kunpu.frameworks.distributed.lock.RedisLockService

**1.3.2序列号**

**1.3.2.1 基于数据库的序列号**

com.kunpu.structure.service.repository.SequenceService

**1.3.2.2 基于redis的序列号**

com.kunpu.structure.service.bussiness.SequenceBusService

**1.3.2消息处理**

**1.3.3 mq**

**1.3.3.1 生成者**

1. 外化配置文件/opt/configs/demo-provider/rabbit.properties内容

#集群地址

rabbit.addresses=10.16.3.103:5672,10.16.3.104:5672,10.16.3.105:5672

#各系统的生产者独立配置

demo-provider.rabbit.virtual-host=demo

demo-provider.rabbit.username=demo.user

demo-provider.rabbit.password=demo

2.单独系统消费者spring配置文件\configs\spring\rabbit.demo.product.xml



3.Java 发送代码



该文件举了三个例子的发送消息对象,全部通过JSON转换提交,可以在RabbitMQ的后台管理系统中查看具体的消息内容.

**1.3.3.2 消费者**

1.外化配置文件/opt/configs/demo-consumer/rabbit.properties内容

#集群地址

rabbit.addresses=10.16.3.103:5672,10.16.3.104:5672,10.16.3.105:5672

#各系统的消费者独立配置

demo-consumer.rabbit.virtual-host=/

demo-consumer.rabbit.username=centos

demo-consumer.rabbit.password=kunpu2017

//任务线程池

demo-consumer.rabbit.pool-size=5-10

#任务队列容量

demo-consumer.rabbit.queue-capacity=128

2.单独系统消费者spring配置文件\configs\spring\rabbit.demo.consumer.xml



3.java接收代码



**1.3.4配置中心**

**1.3.4.1 配置文件**



**1.3.4.1 配置中心管理地址**

**<http://test.disconf.com/login.html>**

如下图：

