

pr



交接文档(1).html
44.5KB

关于产险的微服务构成

架构简述：

微服务采取的是本地环境和测试环境共用一套注册中心，本地环境不注册到注册中心，服务默认路由是到测试环境，可以通过配置路由到本地：

网关的配置（参考）：

```
pbase-admin-web.ribbon.listOfServers=http://localhost:8763
```

```
pbase-admin-
```

```
web.ribbon.NIWSserverListClassName=com.chinacri.cloud.custom.netflix.loadbalancer.CloudConfigurationBasedServerList
```

应用之间的配置（参考）：

```
-Dcom.chinacri.cloud.service.CodeService=dubbo://localhost:20881?version=1.0.1-SNAPSHOT
```

网络入口：

F5 交由网络组和运维组进行处理

前端

本地开发

环境：node

测试

环境：nginx 服务器地址：192.168.13.40（用户名和密码 咨询运维）

静态文件存放目录：/mwbase/appsystem

注意：本地和测试的js私服服务器：192.168.13.60（用户名和密码 咨询运维）

私服网址：http://192.168.13.60:8081/#admin/repository/repositories:npm-public

构建服务器地址：http://192.168.26.53:8080/（admin/123456）

生产

交由运维进行维护

后端

本地开发

环境：java 1.8、maven >3.5

项目说明：

对于基础功能的扩展

分组地址: <http://gitlab.sino-life.com/groups/chinacri-pbase-extend>

1. 针对于controller层的通用封装: <http://gitlab.sino-life.com/chinacri-pbase-extend/chinacri-pbase-webapp>
2. 针对于服务层的通用封装: <http://gitlab.sino-life.com/chinacri-pbase-extend/chinacri-pbase-service>
3. 关于dubbo的bug和功能适应: <http://gitlab.sino-life.com/chinacri-pbase-extend/chinacri-pbase-dubbo>
4. 关于redis的通用封装: <http://gitlab.sino-life.com/chinacri-pbase-extend/chinacri-pbase-cache-redis>
5. 针对于整个架构所用到的类和功能封装: <http://gitlab.sino-life.com/chinacri-pbase-extend/chinacri-pbase-common>

6. 关于nacos的功能封装: <http://gitlab.sino-life.com/chinacri-pbase-extend/chinacri-pbase-nacos-extend>

对于技术规范以及公司的技术沉淀

分组地址: <http://gitlab.sino-life.com/groups/open-source>

1. 网关的入口: <http://gitlab.sino-life.com/open-source/chinacri-pbase-gateway>
2. oracle的代码生成器: <http://gitlab.sino-life.com/open-source/chinacri-pbase-generator-oracle>
3. mysql的代码生成器: <http://gitlab.sino-life.com/open-source/chinacri-pbase-generator-mysql>
4. 整个的代码规范以及相关的技术文档: <http://gitlab.sino-life.com/open-source/technicalSpecifications>
5. 后端的脚手架: <http://gitlab.sino-life.com/open-source/chinacri-pxxx-demo>

基础功能的开发 (用户菜单权限)

分组地址: <http://gitlab.sino-life.com/groups/chinacri-pbase>

1. 整个微服务的版本管理: <http://gitlab.sino-life.com/chinacri-pbase/chinacri-pbase-parent>
2. 配置文件的应用: <http://gitlab.sino-life.com/chinacri-pbase/chinacri-pbase-config-server>
3. 存放配置文件的项目: <http://gitlab.sino-life.com/chinacri-pbase/chinacri-pbase-configurations>
4. 自动作业的项目: <http://gitlab.sino-life.com/chinacri-pbase/chinacri-pbase-job-admin>
5. 关于用户、角色和菜单的查询: <http://gitlab.sino-life.com/chinacri-pbase/chinacri-pbase-admin>
6. 对于用户进行授权: <http://gitlab.sino-life.com/chinacri-pbase/chinacri-pbase-auth>
7. 调用非微服务的应用: <http://gitlab.sino-life.com/chinacri-pbase/chinacri-pbase-internal>
8. 分布式下一编号的生成: <http://gitlab.sino-life.com/chinacri-pbase/chinacri-pbase-uniqueId>
9. 基础服务的前端: <http://gitlab.sino-life.com/chinacri-pbase/chinacri-pbase-web>
10. 访问外网项目一比如 (支付, 微信): <http://gitlab.sino-life.com/chinacri-pbase/chinacri-pbase-external>
11. IE8的前端兼容框架: <http://gitlab.sino-life.com/chinacri-pbase/nep-admin>

测试环境

环境: java 1.8 服务器地址: 192.168.26.57、192.168.26.52、192.168.26.110 (用户名和密码 咨询运维)

jar文件存放目录: /mwbase/appsystem

注意: 本地和测试的jar外网私服服务器: 192.168.13.60

内网私服服务器: 192.168.26.50 (用户名和密码 咨询运维)

外网私服网址: <http://192.168.13.60:8081/#admin/repository/repositories:maven-public>

内网私服网址: <http://192.168.26.50:8081/nexus/#welcome>

构建服务器地址: <http://192.168.26.53:8080/> (admin/123456)

生产

交由运维进行维护

问题

pbase

~~eureka 注册中心~~ <http://gitlab.sino-life.com/chinaeri-pbase/chinaeri-pbase-eureka-server.git>
~~sentinel~~ <http://gitlab.sino-life.com/chinaeri-pbase/chinaeri-pbase-sentinel-dashboard.git>

~~dubbo-admin~~ <http://gitlab.sino-life.com/chinaeri-pbase/chinaeri-pbase-dubbo-admin.git>
~~apollo~~ <http://gitlab.sino-life.com/chinaeri-pbase/chinaeri-pbase-apollo-config.git> ==
~~秒杀系统~~ <http://gitlab.sino-life.com/chinaeri-pbase/chinaeri-pbase-seckill.git>

拓展

~~<http://gitlab.sino-life.com/chinaeri-pbase-extend/chinaeri-pbase-reinforce.git>~~
~~<http://gitlab.sino-life.com/chinaeri-pbase-extend/chinaeri-pbase-cache-redis.git>~~
~~<http://gitlab.sino-life.com/chinaeri-pbase-extend/chinaeri-pbase-dubbo-swagger.git>~~
~~<http://gitlab.sino-life.com/chinaeri-pbase-extend/chinaeri-pbase-readiness-health-check.git>~~
~~<http://gitlab.sino-life.com/chinaeri-pbase-extend/chinaeri-pbase-health-check.git>~~
~~<http://gitlab.sino-life.com/chinaeri-pbase-extend/chinaeri-pbase-apollo-extend.git>~~
~~<http://gitlab.sino-life.com/chinaeri-pbase-extend/chinaeri-pbase-cache-redis-cluster.git>~~ ---
自动监控报警 <http://gitlab.sino-life.com/chinaeri-pbase-extend/chinaeri-pbase-prometheus-extend.git> --- 运维
~~<http://gitlab.sino-life.com/chinaeri-pbase-extend/chinaeri-pbase-under-tow-extend.git>~~

RPC与HTTP调用方式选择

SpringCloud 两种调用方式

一种是Ribbon+RestTemplate, 另外一种是Feign。@FeignClient

项目模块相关

<http://gitlab.sino-life.com/chinaeri-pweixin/chinaeri-pbase-record.git> ---华为云资源 账号 等

redis 两种模式? chinaeri-pbase-cache-redis chinaeri-pbase-cache-redis-cluster

redis sentinel

redis 持久化配置启用 rdb ? 生产配置 未设置db 本地管理客户端

-- 缓存一致性问题 ---- 用户token认证暂无, 缓存

非车 -- 更新数据库-- 删除缓存

```
    }  
    return qbBusinessTheircompanyResult;  
}  
  
@Override  
public void update(QbBusinessTheircompanyResult qbBusinessTheircompanyResult) {  
    QbBusinessTheircompany qbBusinessTheircompany = new QbBusinessTheircompany();  
    BeanUtils.copyProperties(qbBusinessTheircompanyResult, qbBusinessTheircompany);  
    qbBusinessTheircompany.setUpdatedDate(DateUtil.getTodayDate());  
    qbBusinessTheircompany.setUpdatedUser(super.getCurrentUserCode());  
    qbBusinessTheircompanyCommonBiz.updateSelectiveById(qbBusinessTheircompany);  
    redisUtil.delPFactorRedis(QbBusinessTheircompanyResult.class.getSimpleName(), qbBusinessTheircompanyResult.getCompanycode());  
}
```

auth ---- 主要是web 接入

测试环境k8s 机器 ---生产 k8s是否已上 =====>咨询运维

SpringCloud注册+dubbo注册 =====>融合

接口注册 =====>改应用注册

分布式下一编号 --- 美团的, 暂时使用的zookeeper方式, 数据库

hbase

pbase-admin RABC+JWT =====>Auth2

~~shiro、security~~

唐奇华

微信后台管理 (旧迁移)

Hbase 模块计划于安排

配置文件问题

common-sentinel

common-dubbo-nacos 区别

~~common-dubbo-spring-cloud~~

common-nacos nacos 服务?

common-rabbitmq ----非车使用--首付 创富

common-zookeeper ---- uuid

dubbo 通讯协议

二进制 dubbo 、 rmi

json,xml webservice soap

协议名称	实现描述	连接	使用场景
dubbo	传输: mina、netty、grizzly 序列化: dubbo、hessian2、java、json	dubbo缺省采用单一长连接和NIO异步通讯	1.传入传出参数数据包较小 2.消费者 比提供者多 3.常规远程服务方法调用 4.不适合传送大数据量的服务器文件、传视频
rmi	传输: java rmi 序列化: java 标准序列化	连接个数: 多连接 连接方式: 短连接 传输协议: TCP/IP 传输方式: BIO	1.常规RPC调用 2.与原RMI客户端互操作 3.可传文件 4.不支持防火墙穿透
hessian	传输: Servlet容器 序列化: hessian二进制序列化	连接个数: 多连接 连接方式: 短连接 传输协议: HTTP 传输方式: 同步传输	1.提供者比消费者多 2.可传文件 3.跨语言传输
http	传输: servlet容器 序列化: 表单序列化	连接个数: 多连接 连接方式: 短连接 传输协议: HTTP 传输方式: 同步传输	1.提供者多余消费者 2.数据包混合
webservice	传输: HTTP 序列化: SOAP文件序列化	连接个数: 多连接 连接方式: 短连接 传输协议: HTTP 传输方式: 同步传输	1.系统集成 2.跨语言调用
thrift	与thrift RPC实现集成, 并在基础上修改了报文头	长连接、NIO异步传输	

dubbo 序列化协议

支持hessian、java二进制序列化、json、SOAP文本序列化多种序列化协议。

hessian是其默认的序列化协议

dubbo 负责均衡

dubbo 调用流程