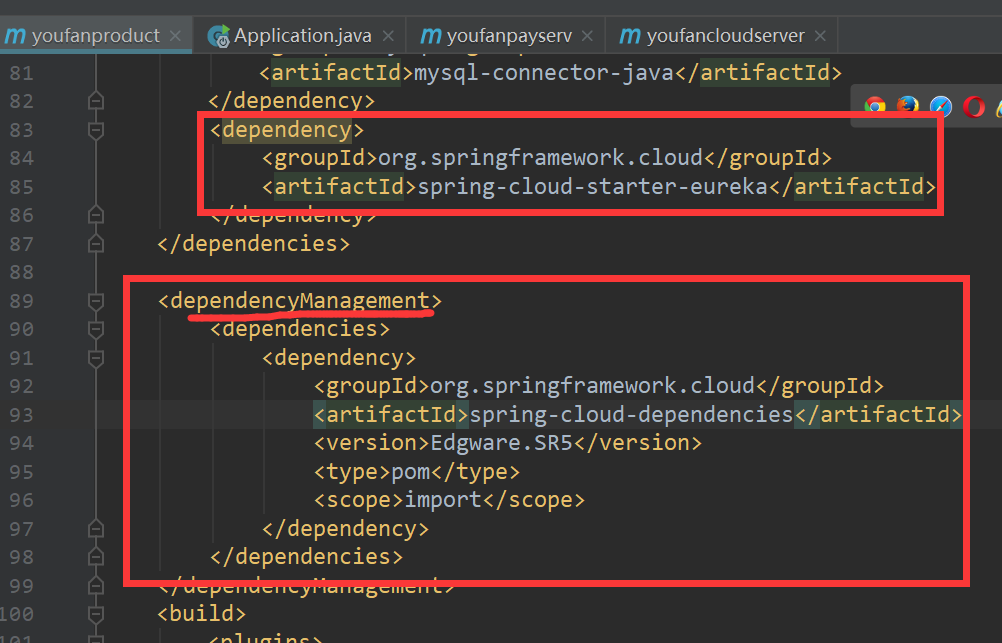
1. EUREKA
   1. 要用eureka客户端，POM中必须引入cloud的依赖管理



1. Feign调用
   1. pom引入cloud,eureka,feign
   2. service加入@FeignClient面向接口编程
   3. service加入需要引用的restful地址，要写成与提供者地址，方法要一样。service这里如果有requestparame, 需要指定value,因为不是controller，不能自动识别。

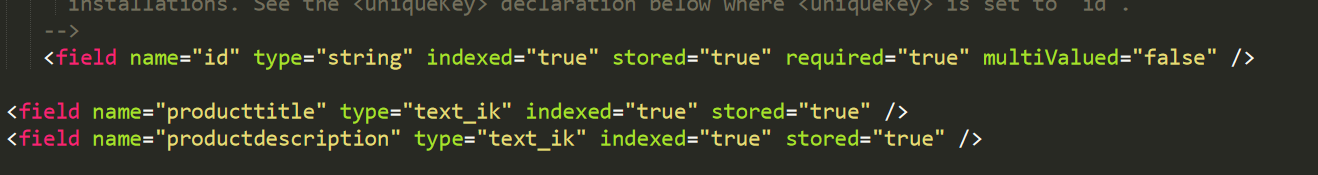
如果是@RequestBody就直接写@RequestBody，后不用跟东西。

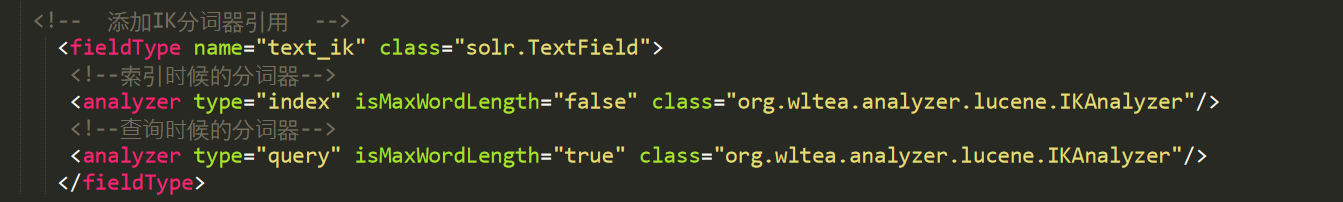




* 1. 调用都微服务的启动类上加上@EnableDiscoveryClient和@EnableFeignClients

1. Sorl
   1. /usr/local/solr-4.10.4/example/solr/collection1/conf下改schema.xml，加上需要查询的filed.并且添加分词器





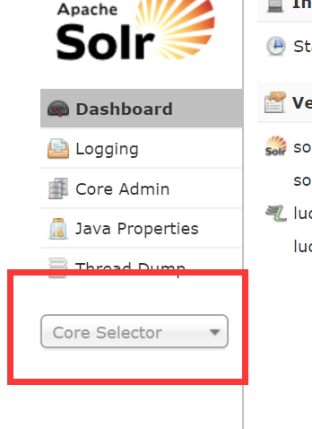
* 1. 到/usr/local/solr-4.10.4/example目录下执行java -jar start.jar
  2. 分词
     1. 需要IKAnalyzer.cfg.xml，IKAnalyzer2012FF\_u1.jar，sougou.dic，stopword.dic

把IKAnalyzer2012FF\_u1.jar放到solr-4.10.4\example\webapps\solr.war里的

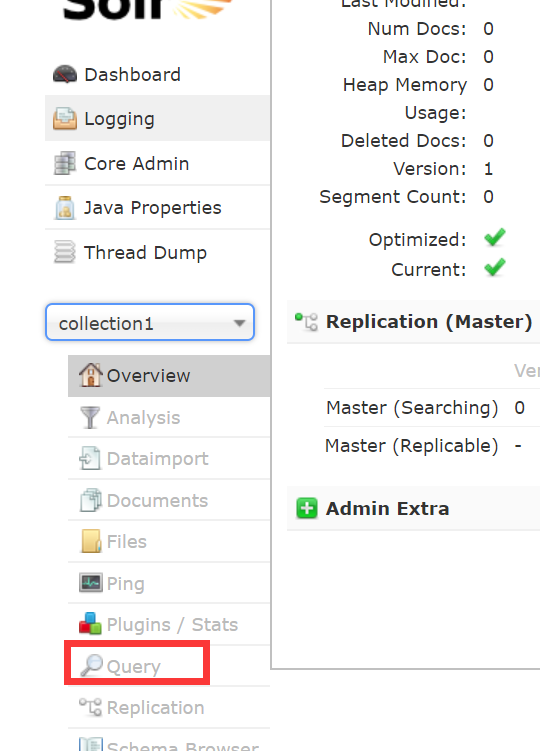
solr-4.10.4\example\solr-webapp\WEB-INF\lib下，把配置文件和词典放到solr-4.10.4\example\webapps\solr.war里的

solr-4.10.4\example\solr-webapp\WEB-INF下

* + 1. 重新java -jar start.jar
  1. 访问http://192.168.0.113:8983/，然后点连接
  2. 选择一个项目，点击



* 1. 然后就可以查询了



* 1. java调用
     1. model类上加上filed注解
     2. [sorlUtil.java](SolrUtil.java)
  2. 整合springboot
     1. 添加POM依赖
     2. 同样MODEL需要加上field注解
     3. util中把SolrClient当作参数传递进去
     4. service里把SolrClient给autowire进来，然后用就可以了

1. 微信支付
   1. 网页授权
      1. 引入POM

commons-codec,com.squareup.retrofit2,com.thoughtworks.xstream,xstream,com.squareup.retrofit2,simpleframework,lombok

* + 1. 根据https://mp.weixin.qq.com/wiki?t=resource/res\_main&id=mp1421140842提供的步骤来授权
       1. 先授权，开发者工具--》公众平台测试账号





* + - 1. 然后用测试者账号的appid和secret来测试

appID

wx8294815eedc65ddc

appsecret

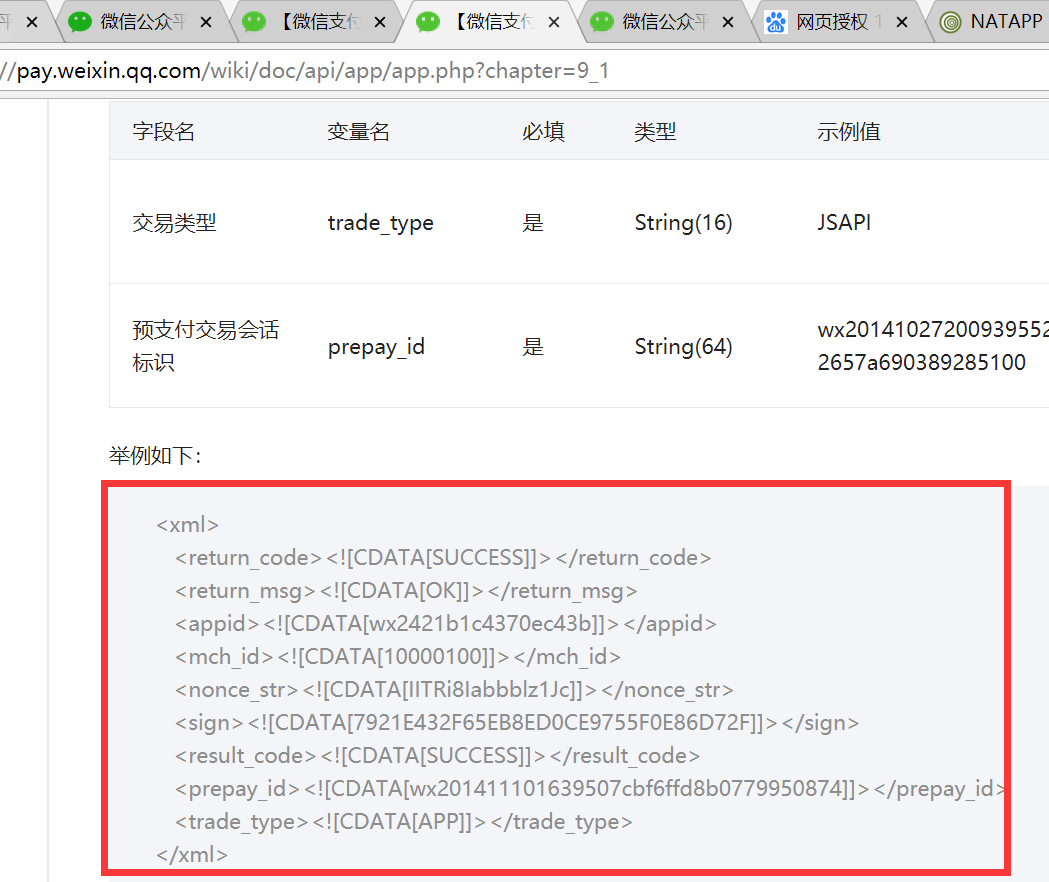
87bd3a28e850daa8eff102d7ebd83f78

@RequestMapping("/auth")  
 public void auth(@RequestParam("code") String code){  
 *log*.info("进入auth方法");  
 *log*.info("code={}",code);  
// https://open.weixin.qq.com/connect/oauth2/authorize?appid=wx8294815eedc65ddc&redirect\_uri=http://kejiawei.nat300.top/auth&response\_type=code&scope=snsapi\_base#wechat\_redirect  
 String url="https://api.weixin.qq.com/sns/oauth2/access\_token?appid=wx8294815eedc65ddc&secret=87bd3a28e850daa8eff102d7ebd83f78&code="+code+"&grant\_type=authorization\_code";  
 RestTemplate restTemplate=new RestTemplate();  
 String response = restTemplate.getForObject(url, String.class);  
 *log*.info("response="+response);  
 JSONObject object=JSONObject.*parseObject*(response);  
 String openid = object.getString("openid");  
 *log*.info(openid);  
 }

* 1. 支付
     1. 文档

<https://pay.weixin.qq.com/wiki/doc/api/app/app.php?chapter=9_1>

目的是构建出下面这个XML文档



1. 单点登录CAS
   1. 下载CAS

<https://github.com/apereo/cas-overlay-template>

选择4.2版本，下载

* 1. IDEA里 FILE-NEW-Project from existing source->导入进去，让IDEA自动下载依赖包
  2. 对etc/cas.properties文件进行修改，端口改一下。比如9080。然后把这个文件COPY到其它地方
  3. 对WEB-INF/spring-configuration/propertyFileConfigurer.xml进行修改

修改<util:properties id="casProperties" location="file:/etc/cas/cas.properties" />

location为刚才放的绝对路径

* 1. 执行package命令进行打包
  2. 把打好的WAR包放到TOMCAT下，确保TOMCAT端口和cas.properties里的端口一致,比如9080
  3. 启动tomcat通过http://localhost:9080/cas进行访问 用户名 casuser,密码Mellon。登录进来说明成功了，由于现在的用户名和密码是写死的，那我们下面来做从数据库来配置用户名和密码。
  4. 加入三个JAR包到项目的webapp/lib下

cas-server-support-jdbc-4.2.7.jar,c3p0-0.9.5.2,mysql-connector-java-5.1.32

* 1. 修改编译后的文件

webapps\cas\WEB-INF\deployerConfigContext.xml

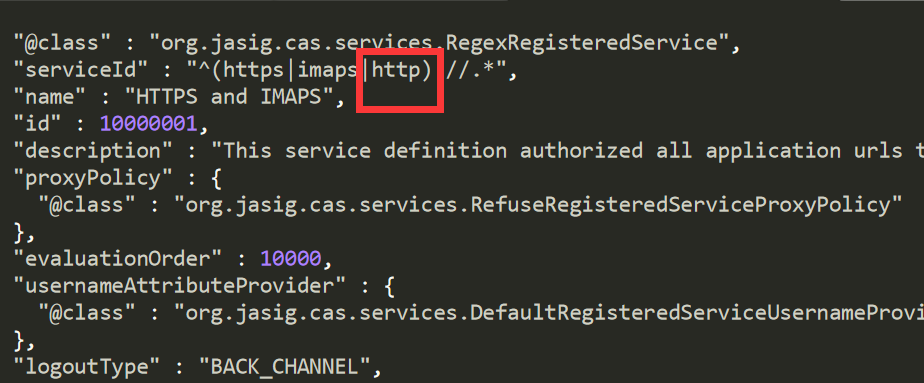
COPY这个连接的文件过去 <deployerConfigContext.xml>

* 1. 修改单独拷贝出来的cas.properties，最后面加入一行。后面是查询SQL，根据数据库自行修改

cas.jdbc.authn.query.sql=select passwordencrypt from user where name=?

* 1. 让cas支持http

\webapps\cas\WEB-INF\classes\services\HTTPSandIMAPS-10000001.json修改



下面是客户端与springboot整合

* 1. 添加POM依赖

<dependency>

<groupId>net.unicon.cas</groupId>

<artifactId>cas-client-autoconfig-support</artifactId>

<version>1.5.0-GA</version>

</dependency>

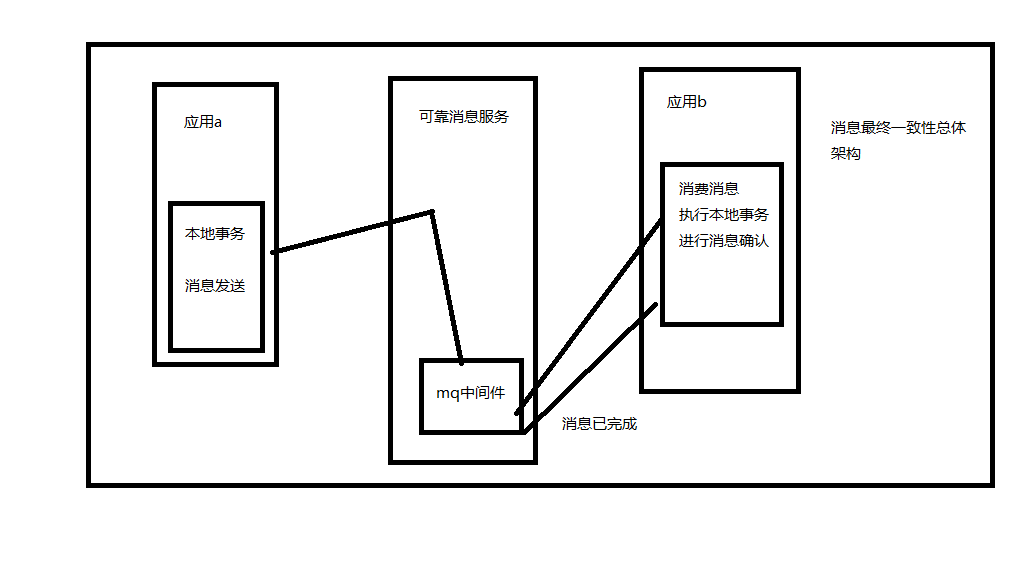
* 1. 配置文件中配置cas

cas.server-url-prefix=http://localhost:9080/cas  
cas.server-login-url=http://localhost:9080/cas/login  
cas.client-host-url=http://localhost:8087  
cas.validation-type=*cas*

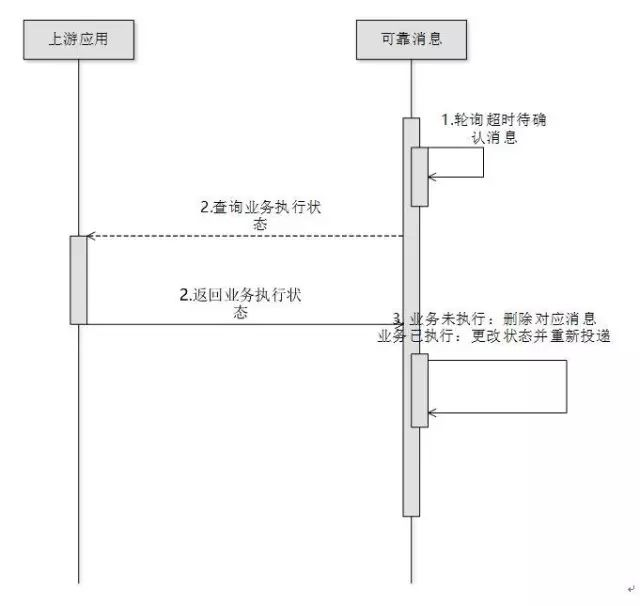
* 1. 客户端调用CAS

@Configuration  
@EnableCasClient  
public class CasConfigure extends CasClientConfigurerAdapter {  
@Override  
public void configureAuthenticationFilter(FilterRegistrationBean authenticationFilter) {  
 super.configureAuthenticationFilter(authenticationFilter);  
 authenticationFilter.getInitParameters().put("authenticationRedirectStrategyClass","com.patterncat.CustomAuthRedirectStrategy");  
 }  
}

1. 分布式事务 可靠消息消息一致性
   1. 整体架构

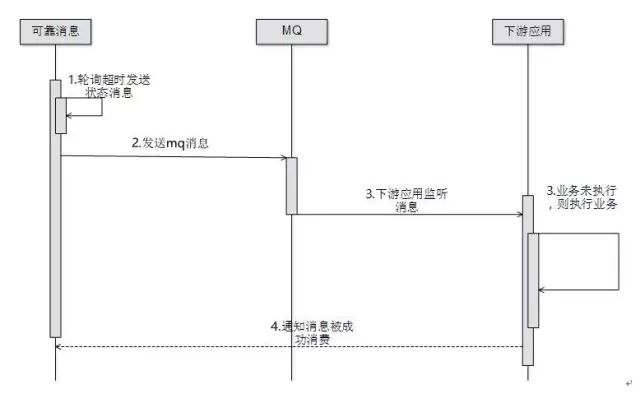


* 1. 异常处理
     1. 消息状态确认(可靠消息服务定时监听消息的状态，如果存在状态为待确认并且超时的消息，则表示上游应用和可靠消息交互出现异常)



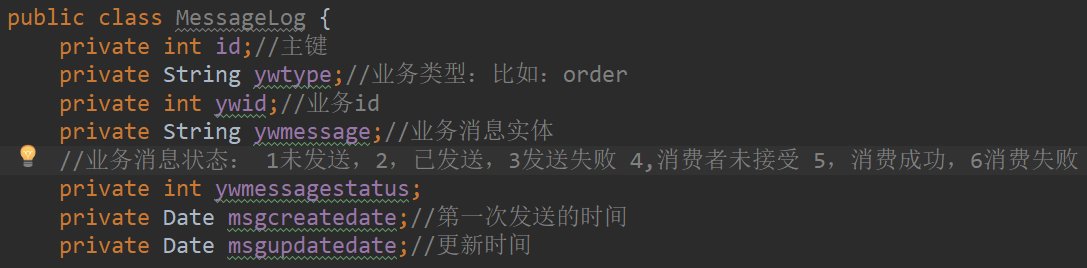
* + 1. 消息重发(消息已发送则表示上游应用已经执行，接下来则确保下游应用也能正常执行)

可靠消息服务发现可靠消息服务中存在消息状态为已发送并且超时的消息，则表示可靠消息服务和下游应用中存在异常的步骤，无论哪个步骤出现异常，可靠消息服务都将此消息重新投递到 MQ 组件中供下游应用监听

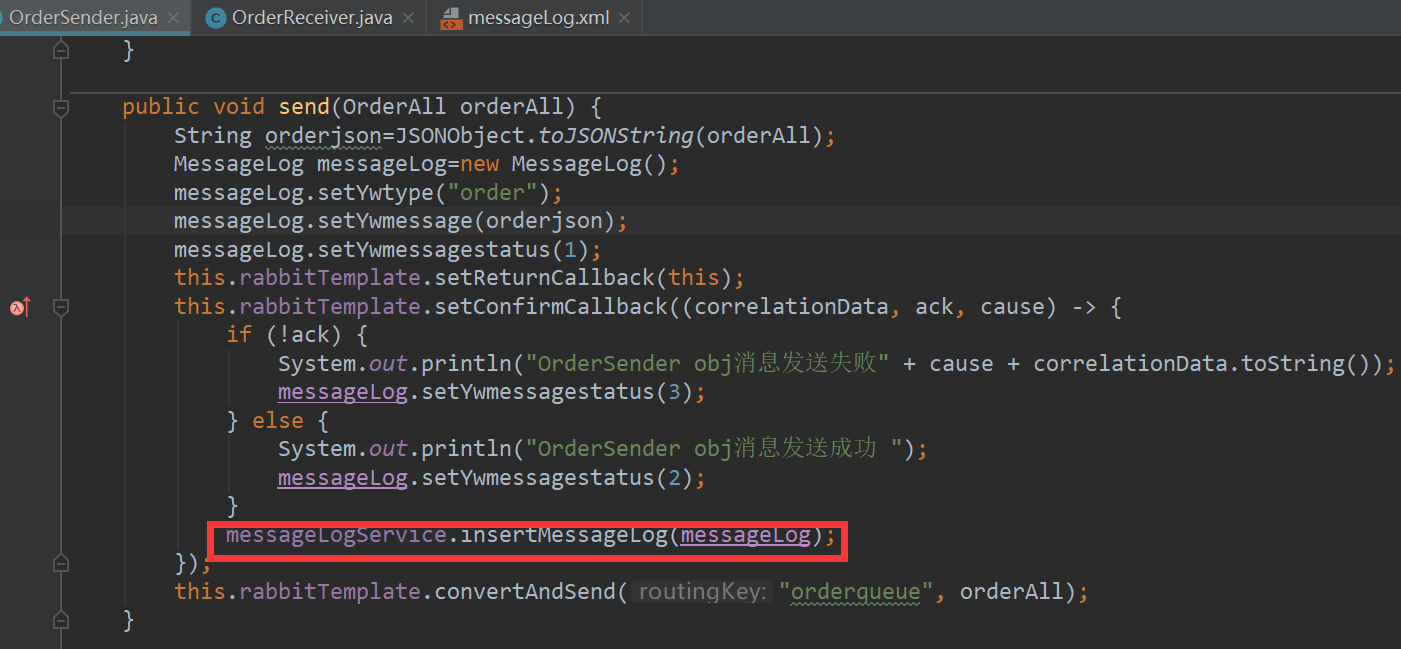


* 1. 代码实现

可靠消息可以新建一张表，来保存状态。



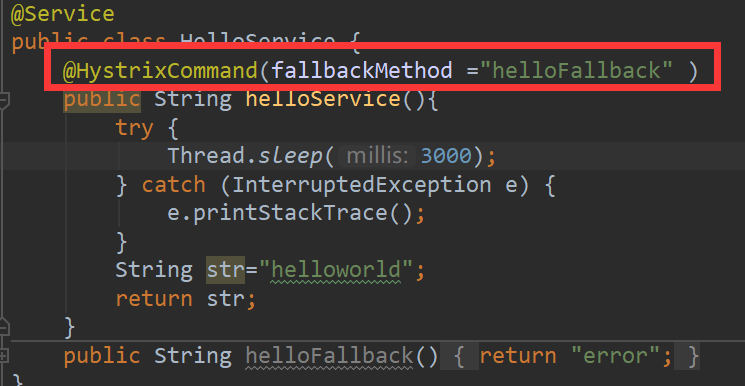
消息发送端：



消息接收端



1. Hystrix
   1. 引入spring-cloud-starter-hystrix依赖
   2. 主启动类加上@EnableCircuitBreaker注解
   3. service



1. ribbon负载均衡
   1. 首先要有两个同样的服务
   2. 再加个客户端调用这两个服务，客户端通过原有的地址去访问，比如url="http://YOUFANPRODUCTTYPE/listproducttype";
   3. 把原来调用的微服务地址这个客户端，比如把@FeignClient(value = "youfanproducttype")改成@FeignClient(value = "youfanrobbinserver")
2. DRUID数据库连接池

目前最好的连接池，为监控而生的连接池

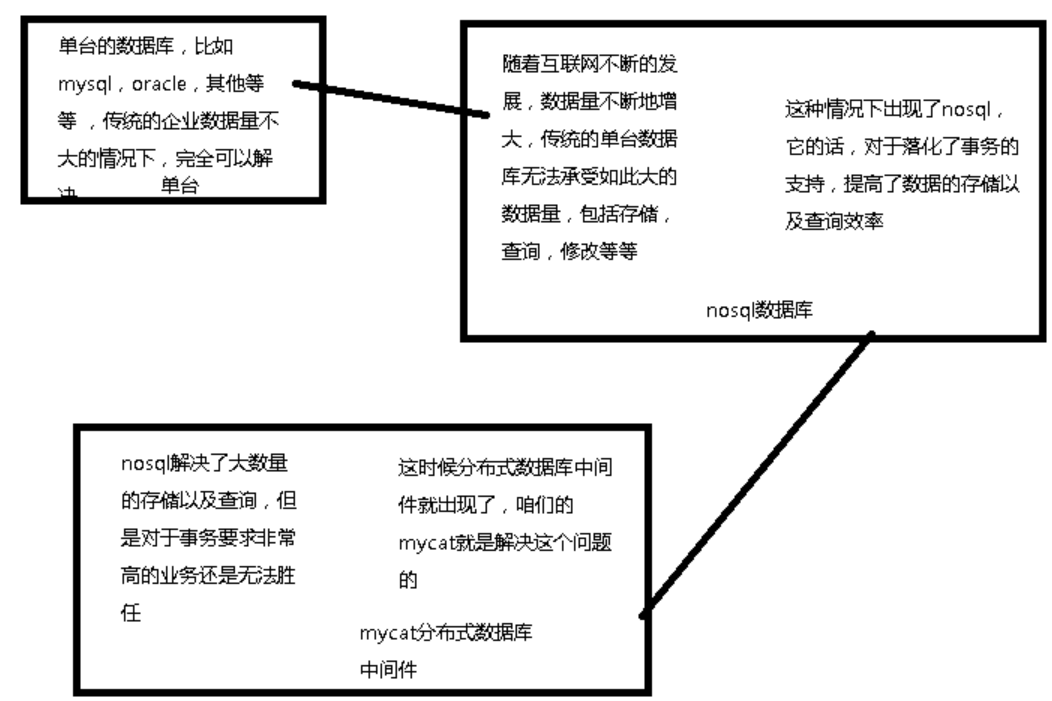
* 1. 加入pom

<dependency>  
 <groupId>com.alibaba</groupId>  
 <artifactId>druid</artifactId>  
 <version>1.0.31</version>  
</dependency>

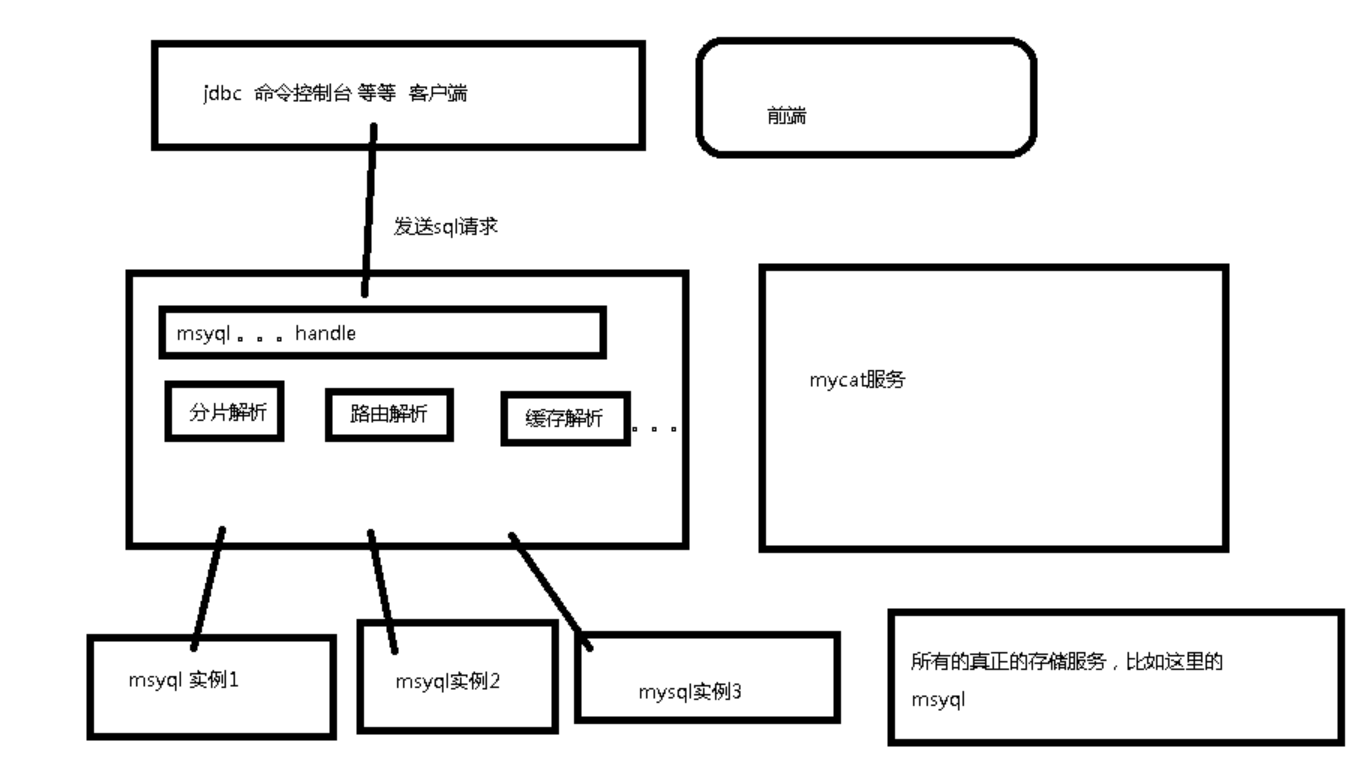
<dependency>  
 <groupId>org.mybatis.spring.boot</groupId>  
 <artifactId>mybatis-spring-boot-starter</artifactId>  
 <version>1.3.0</version>  
</dependency>

* 1. springboot默认不支持druid,所以需要写一个配置文件[DruidConfiguration](DruidConfiguration.java)
  2. 如果启动类里有 datasource相关的都可以删除掉了，不需要了
  3. 访问druid 如http://localhost:8082/druid

1. MyCat
   1. 数据库演进



* 1. mycat架构原理



* 1. 分片：

垂直切片（将不同的表切分到不同的数据库中水平切片）和水平切片（将同一种表按照某种条件切分到不同的数据库中（这种方式最常用）

* 1. 配置

mycat主要有3个配置文件，rule.xml(定义了分片规则),schema.xml（该文件是MyCat中重要的配置文件之一，管理着MyCat的逻辑库、表、分片规则、DataNode以及DataSource）和server.xml，这主要对这3个文件作讲解



* 1. 配置mysql并加用户

UPDATE USER SET HOST='%' WHERE USER='root';

FLUSH PRIVILEGES;

CREATE USER 'mycat'@'%' IDENTIFIED BY 'root';

GRANT ALL PRIVILEGES ON \*.\* TO 'mycat'@'%' IDENTIFIED BY '123456';

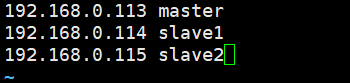
* 1. 安装mycat
     1. tar -zxvf mycat
     2. vim /etc/profile
     3. source /etc/profile
     4. 创建用户groupadd mycat

useradd -g mycat mycat

设置密码 passwd mycat

123456

* + 1. 配置mycat配置文件 vi /etc/my.cnf加入lower\_case\_table\_names = 1
    2. 备份schema.xml cd/usr/local/mycat/conf cp schema.xml schema.xml.cp
    3. vi /etc/hosts



* + 1. 配置schema.xml

<?xml version="1.0"?>

<!DOCTYPE mycat:schema SYSTEM "schema.dtd">

<mycat:schema xmlns:mycat="http://org.opencloudb/">

<schema name="youfanshop" checkSQLschema="false" sqlMaxLimit="100">

<table name="employee" primaryKey="ID" dataNode="dn1,dn2,dn3" rule="sharding-by-intfile"/>

</schema>

<!--数据库节点dn1,对应的主机c1，对应的数据库db1-->

<dataNode name="dn1" dataHost="master" database="db1"/>

<dataNode name="dn2" dataHost="slave1" database="db2"/>

<dataNode name="dn3" dataHost="slave2" database="db3"/>

<!--主机c1-->

<dataHost name="master" maxCon="1000" minCon="10" balance="0" writeType="0" dbType="mysql" dbDriver="native">

<heartbeat>select user()</heartbeat>

<!--mysql数据库的连接串-->

<writeHost host="hostM1" url="master:3306" user="mycat" password="123456">

</writeHost>

</dataHost>

<!--主机c2-->

<dataHost name="slave1" maxCon="1000" minCon="10" balance="0" writeType="0" dbType="mysql" dbDriver="native">

<heartbeat>select user()</heartbeat>

<!--mysql数据库的连接串-->

<writeHost host="hostM2" url="master:3306" user="mycat" password="123456">

</writeHost>

</dataHost>

<!--主机c3-->

<dataHost name="slave2" maxCon="1000" minCon="10" balance="0" writeType="0" dbType="mysql" dbDriver="native">

<heartbeat>select user()</heartbeat>

<!--mysql数据库的连接串-->

<writeHost host="hostM3" url="master:3306" user="mycat" password="123456">

</writeHost>

</dataHost>

</mycat:schema>

* + 1. 配置server.xml

删除无用的用户

添加可用的用户

<user name=”mycat”>

<property name=”password”>123456</property>

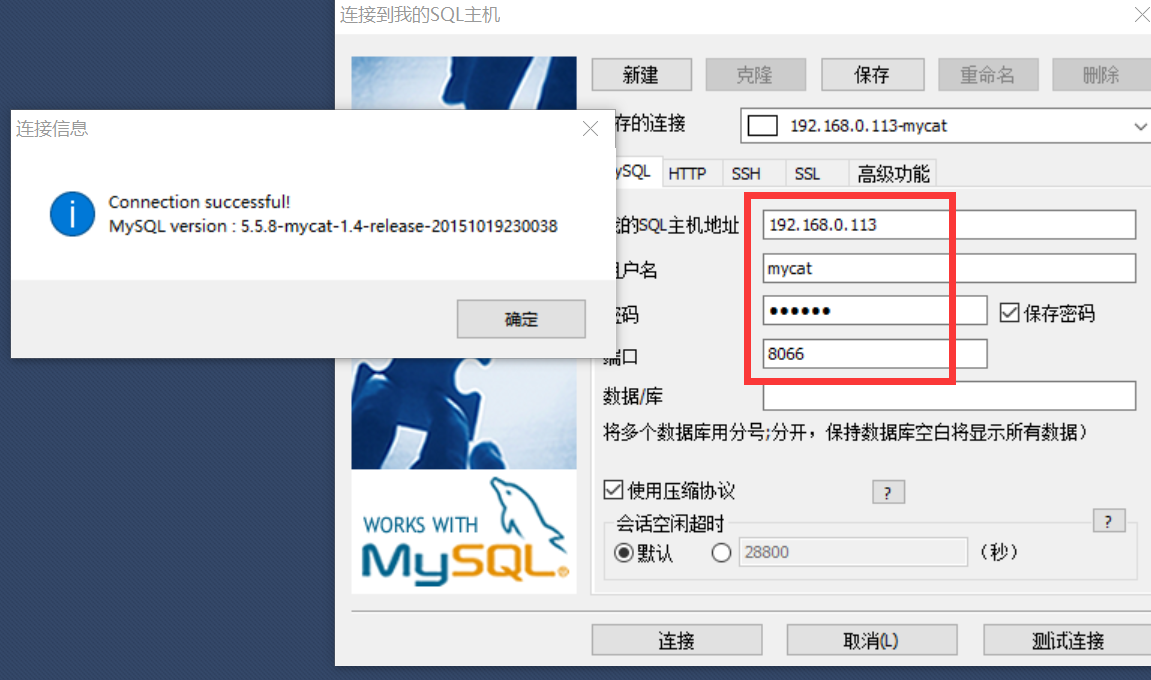
<property name=”schemas”>youfanshop</property>

</user>

* + 1. 修改/usr/local/mycat/conf/partition-hash-int.txt文件，在下面添加10020=2,原本默认的是分两个就是10000和10010，现在我们三个就要分三个分类ID了，添加一即可。
    2. 分别三台机器上创建db1,db2,db3三个数据库，每个数据库中创建表

CREATE TABLE employee (id NOT NULL PRIMARY KEY,NAME VARCHAR(1000),sharding\_id INT NOT NULL)

* + 1. 启动mycat: mycat start
    2. 检查端口号，是否启动成功ss -lnpt|grep 8066
    3. 用mysql客户端连接mycat,在表里尝试增加数据，发面数据能成功分配到其它地方。
    4. 如果有错误，more /usr/local/mycat/logs/wrapper.log



* 1. [mycat分片算法](分片算法.docx)
  2. ID自增

sequence\_conf.properties需要配置