## 第三次作业

## 基本业务逻辑：

当有大量的用户同时提交申请参加健身课程的表时，直接进行数据库的操作会浪费很多时间，系统的消息队列也会很拥堵。所以我们将“返回页面”和“存入数据库”的两个操作进行了分离，让存储到数据库这个操作与返回界面异步提高系统的响应效率。

## 实现逻辑：

使用kafka作为消息的中间件，将提交上来的用户的信息作为消息的内容发布到information这个topic中去,返回界面，显示用户信息正在保存。

处理端监听“information”这个topic，听到信息后，将信息解码之后存储到数据库中去。

## 实现代码：

1. 引入spring-kafka依赖（pom.xml）

<dependency>

<groupId>org.apache.kafka</groupId>

<artifactId>kafka-streams</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.kafka</groupId>

<artifactId>spring-kafka</artifactId>

</dependency>

1. 配置spring-kafka（application.properties）

#Kafka Topic

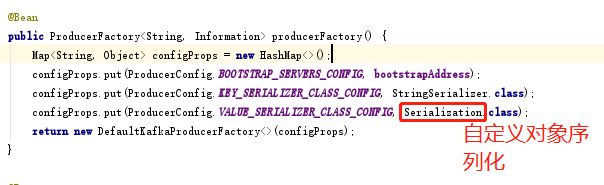
message.topic.name=information

spring.kafka.bootstrap-servers=localhost:9092

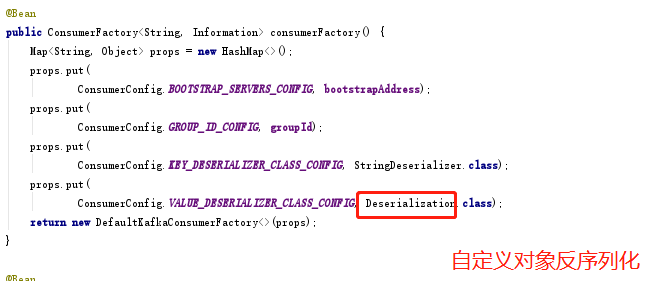
#Unique String which identifies which consumer group this consumer belongs to

spring.kafka.consumer.group-id=jcg-group

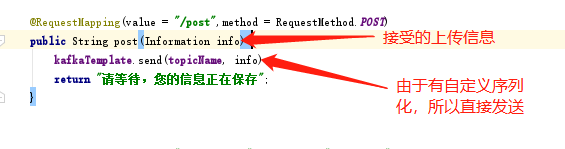
1. 配置Producer



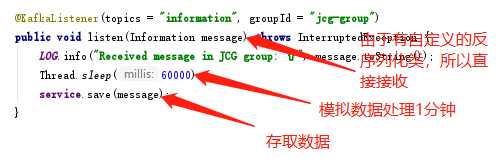
1. 配置Consumer



1. 发送消息，并返回界面

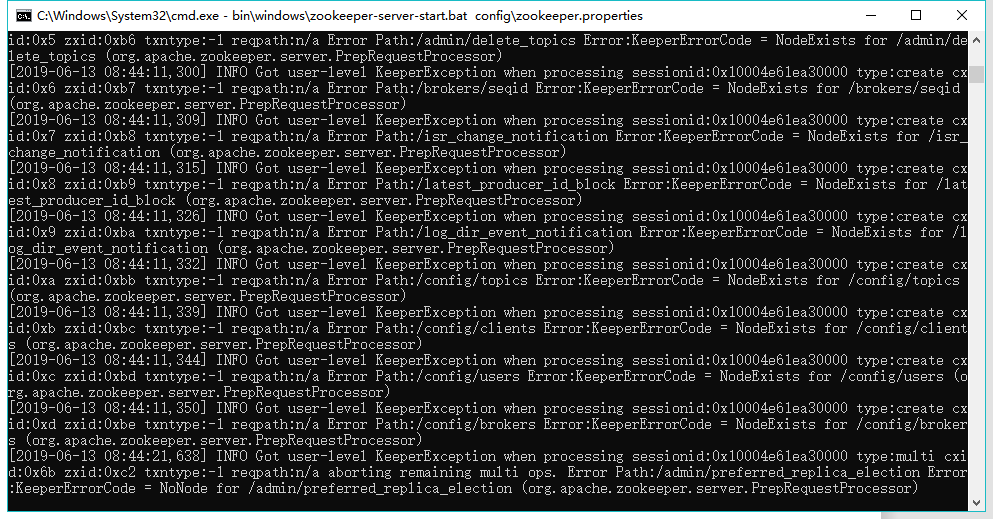


1. 接受消息，并进行数据处理与存储，模拟

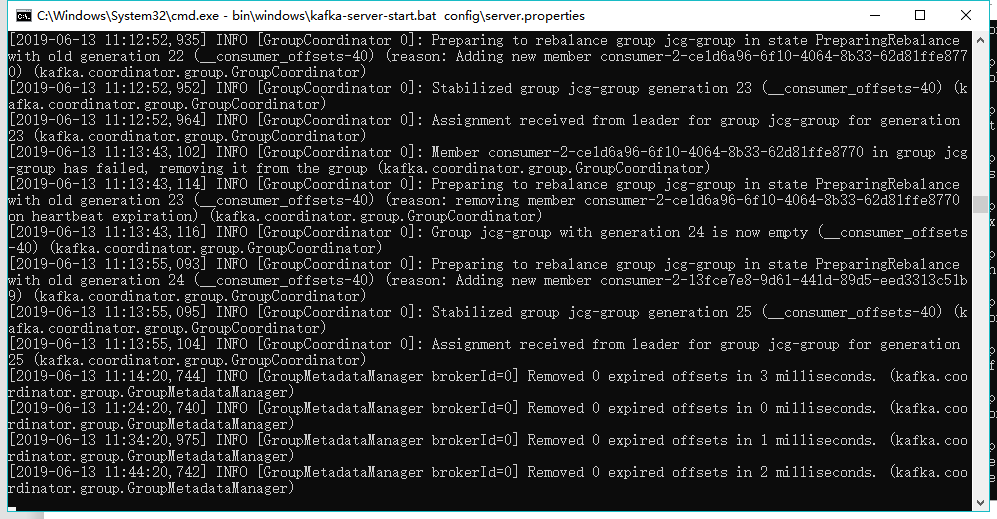


## 运行情况

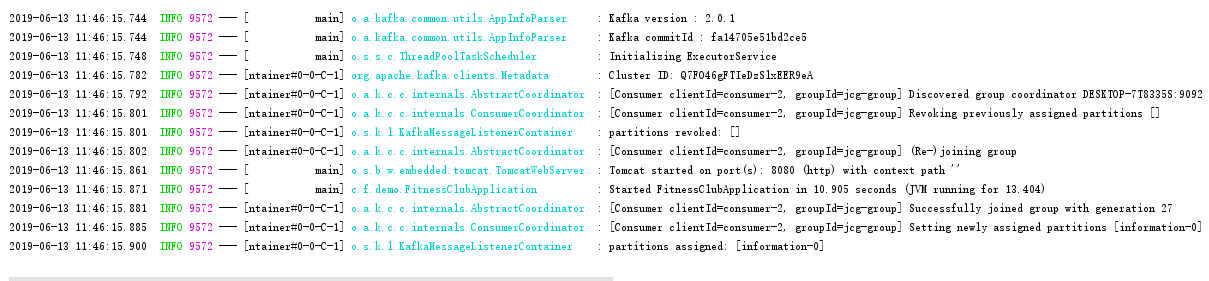
1. 运行zookeeper



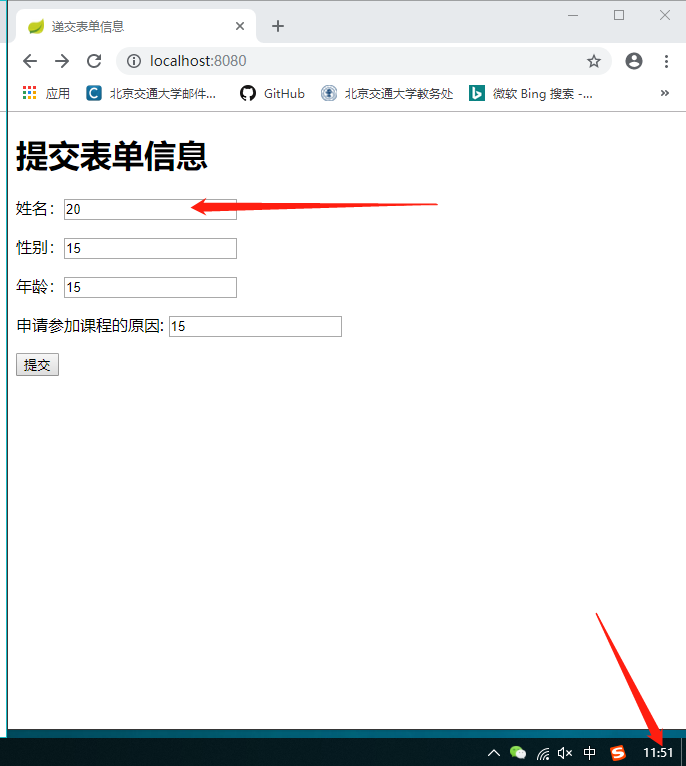
1. 运行kafka



1. 运行spring-boot程序



1. 提交信息



（5）查看结果

