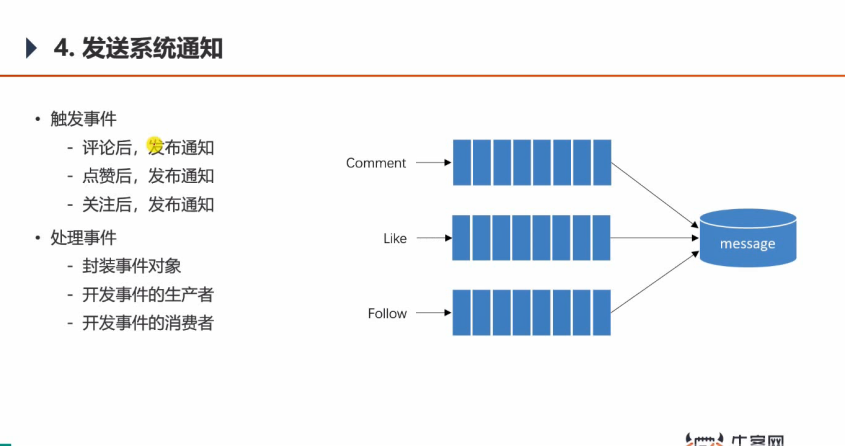
18配合起始

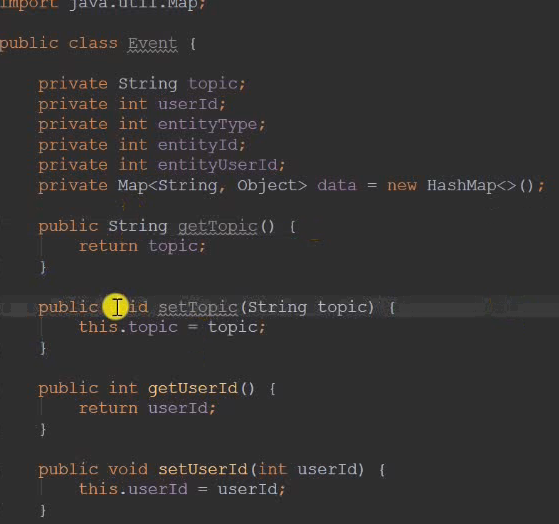


鉴于点赞关注后的系统通知实时性不需要特别高，因此选用了kafka实现异步的发送

系统通知。

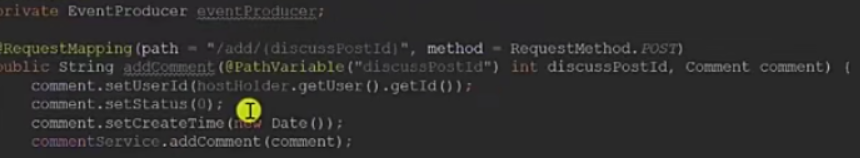
经常会发生的事，所以用消息队列。定义3个主题

创建事件类

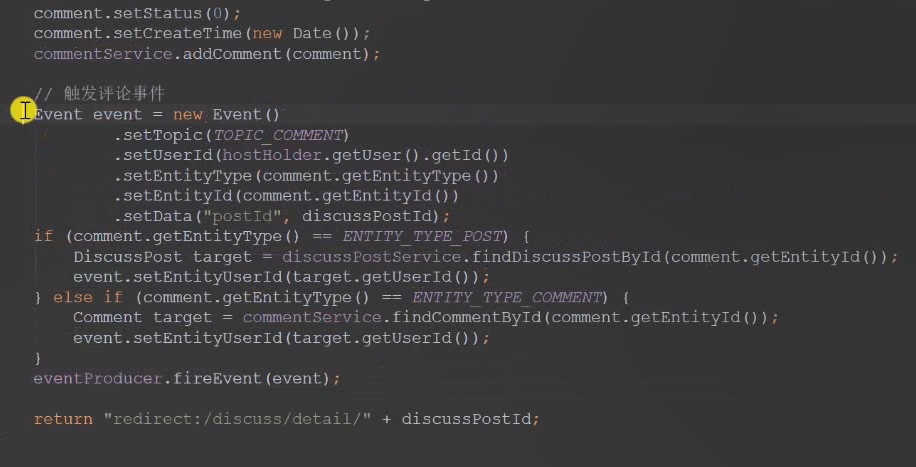


在评论后触发事件，要主动调生产者，消费者是被动触发。

大致就是发消息给消费者，消费者添加数据库

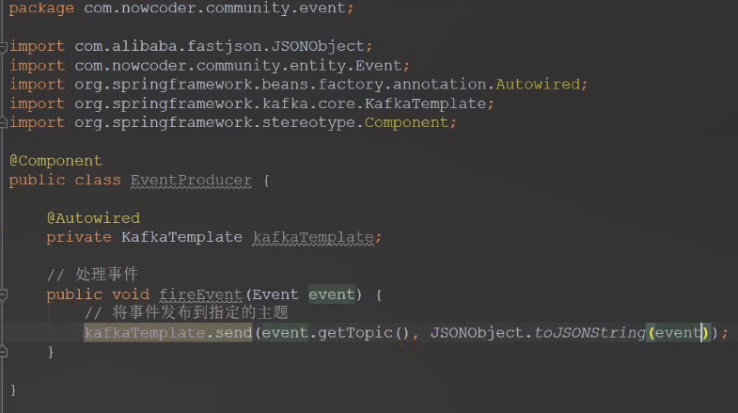


针对是评论帖子还是评论，要用不同service来获得target，再用里面userId来设置



最后一步是去调producer

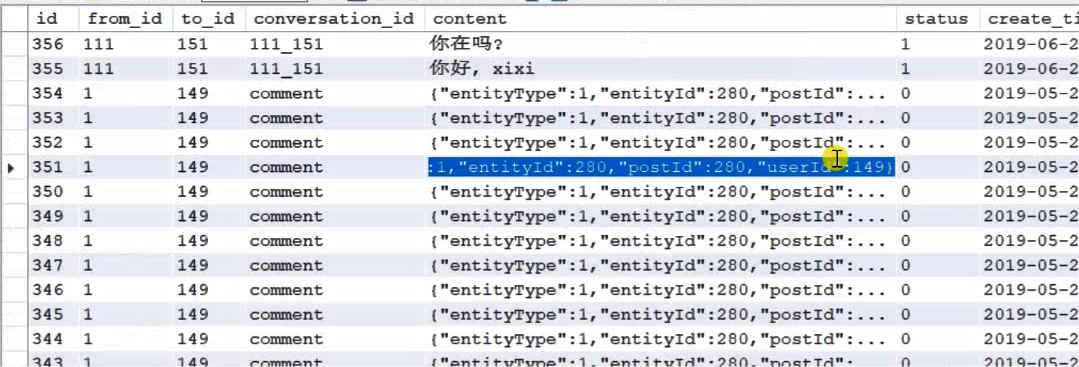
**事件生产者类：其中发送事件发送的是事件对象**（将其转为json格式的字符串发送）



**事件消费者类**：因为点赞、关注、评论回复的消息处理差不多

**发布通知将消息添加到message表**（两用户之间消息对话的那个表）里，***消费事件的处理方式是将事件放到数据库***

Message表：



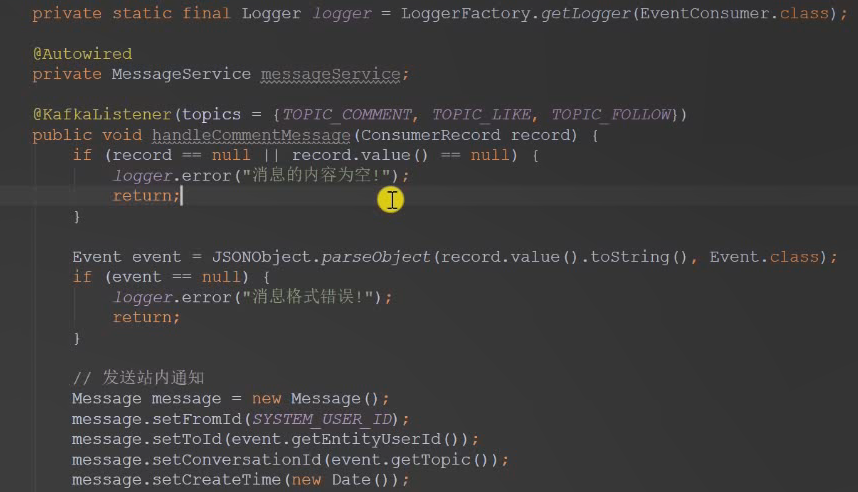
所以将评论、点赞、关注通过一个方法处理事件



**事件的处理就是将此事件传给数据库**，其中内容是json字符串因为后续显示还要处理

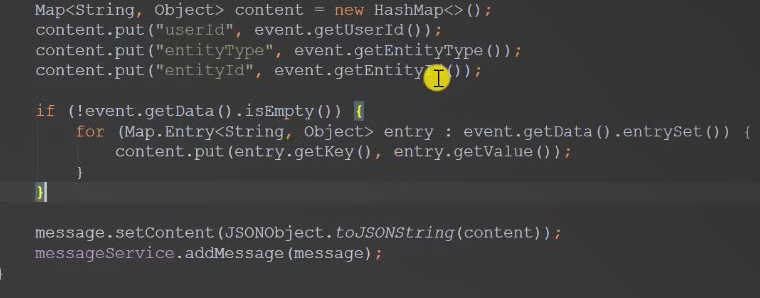
评论的controller层

event是事件触发时封装了相关数据

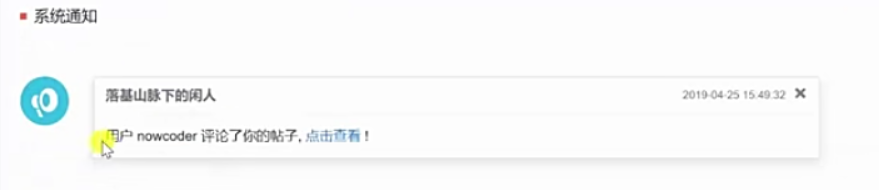


发送消息给发帖人是系统Id，也就是1

entityUserId就是发帖的人，接受到了比如(获得赞)的message, 所以to是他



map里放的目标是为了拼出下面图里的话



**为什么要用消息队列：**

1.在触发事件之后，此方法就可继续向下执行，消费者会新建一个线程进行操作，可以提高性能，处理效率高（异步）

2.当此贴是个热帖时：评论的人非常高，可以用消息队列攒着慢慢处理（削峰）

3.调用生产者的send不用注入消息表的service，消费者调用就可以，**比如点赞、评论、关注如果不用消息队列那么每个对应的controller都要注入消息表的service**，**但用了消息队列，只用消费者类里注入消息表的service就可以（解耦）**

**点赞/关注**

