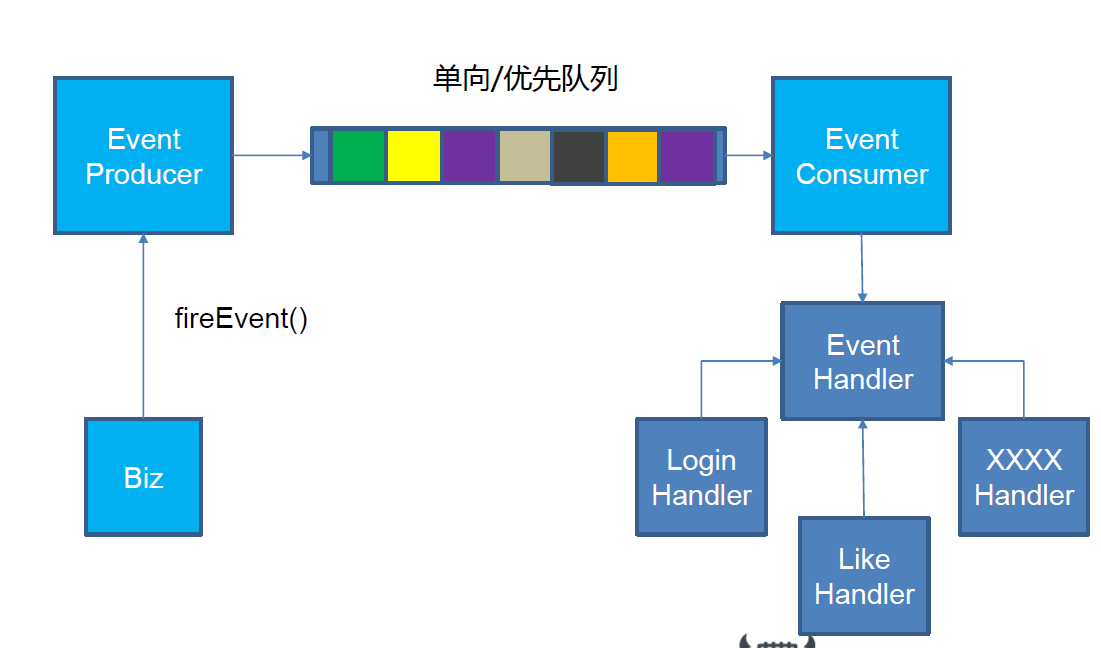
项目第八天

增加功能：增加异步队列和邮件发送功能

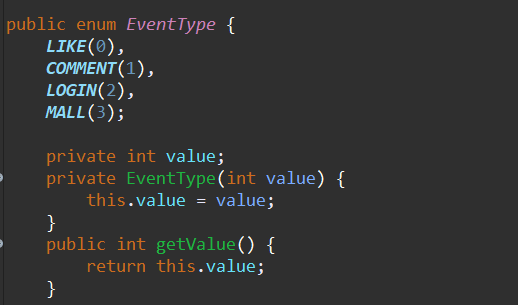
**一、异步队列：**

有时候前端用户进行一个操作时会触发后台许多变动，这时候就要将实时性要求高的和不高的分开处理，要求高的就立即响应，要求不高的相对耗时的操作，就转入异步队列，由另一线程处理，这样就不容易卡在某个地方 **。**

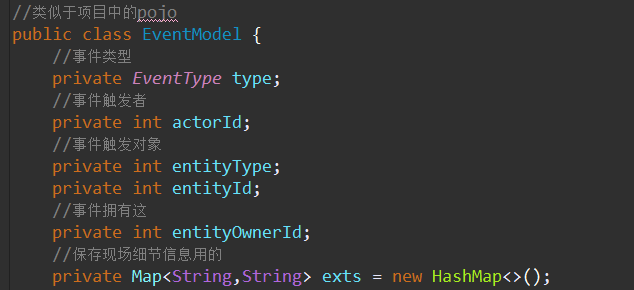
整体思路：在控制器端将实时性要求不高的动作不直接响应，而是将该请求信息打包成事件对象EventModel,传给EventProducer，这个函数对其进行序列化，然后加入队列中（用redis的lpush和brpop实现）。EventConsumer负责从队列循环取出需要处理的事件（在初始化的时候就先在map中映射好了事件类型EventType和处理该类型的Handler映射关系），首先反序列化成EventModel对象，然后根据其内部属性EventType（这个是枚举型，在Handler的实现类中都定好了需要去处理哪些EventType）来确定是什么类型的事件，然后分发给需要对其处理的Handler上，进行相应的处理。



1、EventType:实现定义好的需要处理的事件类型，枚举型



2、EventModel：就是实体类对象，是数据流转的载体



3、EventProducer:控制器端调用的入口，主要作用将待处理事件加入队列

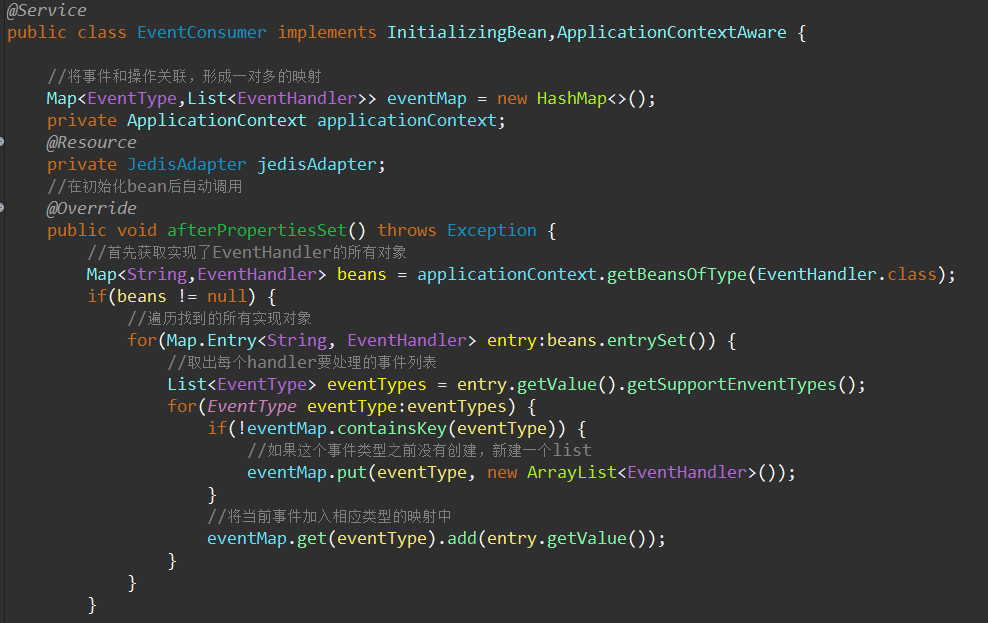


4、EventHandler对事件进行处理，这是个接口，具体由实现类完成（如LikeHandler）。doHandler就是需要执行的具体操作操作(如发站内信通知等)，getSuppoerEventTypes返回的是这个Handler需要对哪些类型的事件进行处理，这里主要用于告知调用者EventConsumer遇到了哪些事件需要调用自己去处理。

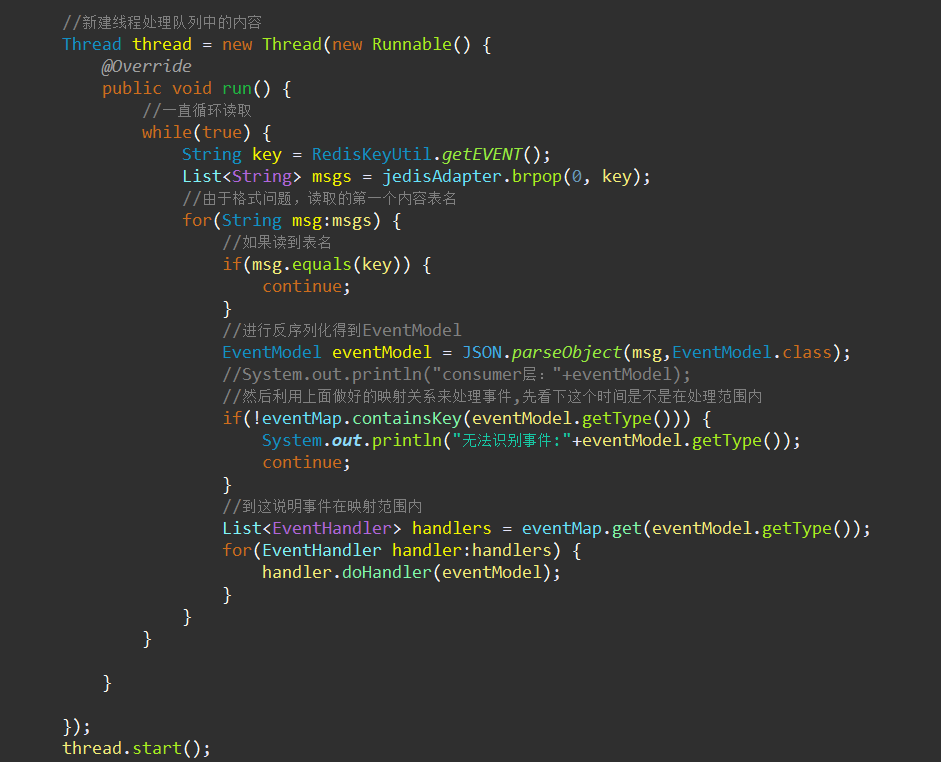


5、EventConsumer 主要有两个任务：

①是形成映射关系，将需要处理的事件类型和处理这个类型事件的Handler进行映射，这个在bean创建的时候就要初始化完成，因此实现了InitializitingBean接口，并在afterPropertiesSet方法中进行。先利用applicationContext获取所有实现了Eventhandler接口的类（spring容器中的beans），然后通过获得的这些Handler实现类中的getSuppoerEventTypes方法获取其需要处理的事件类型。因为整个映射是一对多的关系，采用Map<String,Lits<EventHandler>结构实现。最后将需要处理的事件类型和处理事件的handler形成了映射，那么每次从队列中取到一个事件，就可以先判断其事件类型，然后通过map的映射找到需要调用的Handler实现类去进行处理。

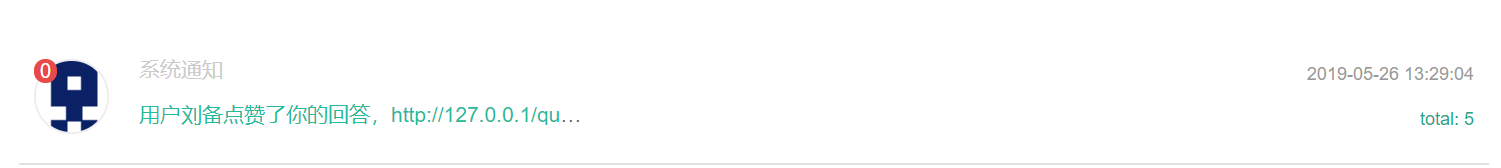


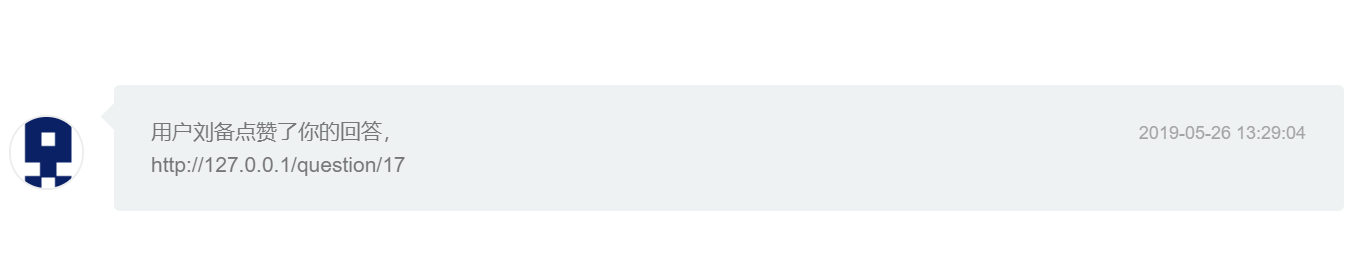
**②（后续拓展：这里可以考虑多线程处理）**新开辟一个线程处理队列中的事件。首先从队列中取出json串进行反序列化得到EventModel,由上面定义可知这个对象中包含了事件类型EventType,那么就可以根据之前建立好的映射关系，找到需要处理这个事件类型的Handler实现类的集合（map中的list）,然后遍历这个集合，调用其dohandler函数去处理事件。



上述是整个通用异步框架的搭建，接下来需要写具体的实现类来实现相关功能，以后要新增加异步功能都这样写。

需求分析：在用户点赞的情况下，通过异步的方式由系统发一个站内信给对方，提示有用户给他点了赞。





具体实现：

Controller层：

LikeController增加任务：

1、将事件打包成EventModel,因为是用户对回答点赞，因此actorId就是当前用户id,EntityOwnerId就是这个回答的userId（先根据commentId获取到Comment对象），Exts就是携带额外的信息，这里吧questionId放进去，让其写入到站内信息内容的跳转地址中，EntityType和EntityId就是这个回答实体的参数，EventType标识了这个时间类型，这里为LIKE(枚举类)。

2、利用eventProducer将这个对象进行序列化然后加入到redis的队列中，等待eventComsumer去取出来调用相应的Handler实现类处理（这里是LikeHandler,因为上面的类型定为了LIKE）。



Async.handler层：LikeHandler作为Handler的实现类，因此真正执行的业务都在这里写。

dohandler为具体处理函数，在上述的异步框架中已经把事件的封装对象反序列化，将model对象传入了，因此这里直接获得modle对象然后创建新的Message对象，将model中的信息对应填入就行了，调用之前写的messageService进行信息发送。

getSupportEventType中定义了Likehandler这个实现类（就是自己）要处理的事件类型，这里只添加了LIKE,说明只需要处理LIKE的事件类型。因此EventComcumer直接利用这个函数就能识别调用谁来处理LIKE事件类型。



**二、邮件发送：（还未实现）**