

## 79\_高阶实战：登录模块基于 RocketMQ 发送消息替代 Spring Event 事件

---

**儒猿架构官网上线**，内有石杉老师架构课最新大纲，儒猿云平台详细介绍，敬请浏览

官网：[www.ruyuan2020.com](http://www.ruyuan2020.com)（建议 PC 端访问）

---

### 1、开篇

上节课介绍了主动取消订单和延迟取消订单的测试过程，在小程序上进行下单、取消、等待操作模拟了两个场景，通过 log 日志的查看正式了我们的想法。本节课将针对登录模块进行改造，使用 RocketMQ 替代 Spring Event 事件完成发送消息的功能。今天包括如下内容：

- 定义登陆消息相关组件
- 发送以及消费登陆消息

### 2、定义登陆消息相关组件

之前对登陆的操作时通过 Spring Event 来实现的现在使用 RocketMQ 替换之，因此需要定义消息生产者、消息消费者和消息监听器。

如图 1 所示，在 spring/jms 目录下面建立 LoginProducerConfiguration 类，用来定义登陆消息的生产者。在类中定义了 producerGroupName（生产者组名字），nameServerAddr（名字服务器地址）。在初始化方法 init 中设置上述两个值，并且启动消息生产者。

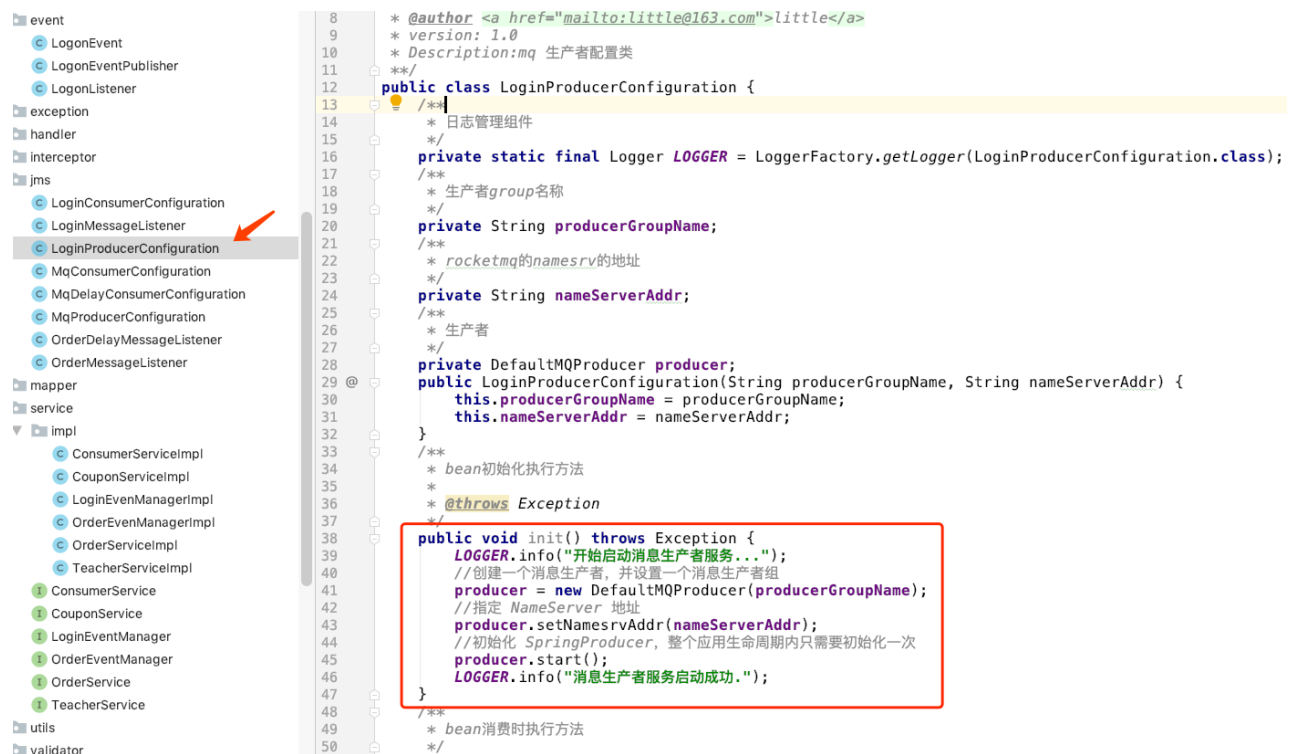


图 1

有了消息生产者，再定义消息的消费者。在同一目录下，创建 LoginConsumerConfiguration 类，如图 2 所示，在类中也需要定义 consumerGroupName（消费者群名字），nameServerAddr（名字服务器地址），通过 setConsumeFromWhere 方法定义队列消费的方式，通过 subscribe 方法定义消费 topic 下所有的消息以及 registerMessageListener 方法定义对应的消息监听器，最后启动消息消费者。



图 2

再来看看消费者的监听器，还是在 `spring/jms` 目录下创建 `LoginMessageListener` 类，该类实现了 `MessageListenerOrderly` 接口，特别是 `consumeMessage` 方法。如图 3 所示，该方法中首先将监听到的 `consumer` 消息从 `json` 对象解析成 `Consumer` 对象，依次判断首次登陆、生日优惠券等逻辑，最后更新登陆次数。



图 3

最后将上面定义的消息生产者、消费者和监听器都注入到 Spring IoC 容器中进行管理。

如图 4 所示，在 `resources` 目录下面的 `spring-jms.xml` 文件中创建 3 个 bean 节点。分别定义了 `loginMessageListener`、`loginConsumerConfiguration` 和 `loginProducerConfiguration` 其中关于初始化的配置信息都从 `application.properties` 文件中读取。



图 4

### 3、发送以及消费登陆消息

介绍完了登陆消息的生产者、消费者和监听器的定义以后，再来看看它是如何应用到登陆操作中的。

如图 5 所示，在 `spring/controller` 的目录下面，`ConsumerController` 作为后台 API 的入口类，其中有一个 `login` 方法，在方法体中调用了 `consumerService` 接口的 `login` 实现方法，并且传入 `consumer` 对象。



图 5

如图 6 所示，顺势看 `spring/service/impl` 下面的 `ConsumerServiceImpl` 类，其中实现了 `login` 方法，方法体中先通过传入的消费者 ID 获取消费者对象，然后通过 `loginEventManager` 的 `loginEvent` 方法发布登陆事件。

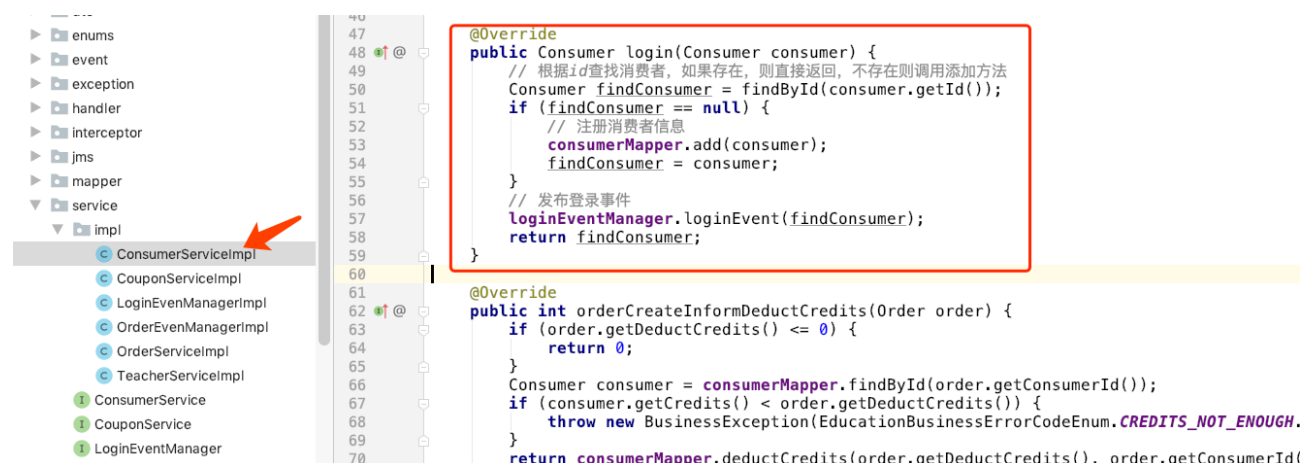


图 6

顺着 login 方法中调用 loginEvent 方法往下看, 如图 7 所示, 在 spring/service/impl 目录下面的 loginEventManagerImpl 类文件中实现了 loginEvent 方法, 在方法中调用了 sendLoginMessage 方法, 这个方法对传入的消息进行 setTopic 和 setBody 的处理然后通过 loginProducerConfiguration 中的 send 方法将消息发送出去。



图 7

#### 4、总结

本节课使用 RocketMQ 替代 Spring Event 发送登陆消息, 针对 RocketMQ 的消息生产者、消费者和监听器进行了定义和使用等操作。下节课会测试一下 RocketMQ 改造之后的代码, 看看执行效果。这里将[代码](#)给大家, 下期见, 拜拜。