



链滴

Active MQ 入门 2, Spring Boot 整合 Active MQ

作者: [DongXiaokai0819](#)

原文链接: <https://ld246.com/article/1582728276197>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

Active MQ 入门 2, Spring Boot 整合 Active MQ

1、搭建 Spring Boot 项目

首先创建一个 Spring Boot 的项目, 在 pom 文件中加入:

```
<parent>
</parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>.1.0.RELEASE</version>
<relativePath><!-- lookup parent from repository -->
</parent>
</pre>
```

加入 Spring Boot 对 Active MQ 支持的 starter 包: starter-activemq

还有 Active MQ 的连接池: activemq-pool

```
<dependency>
</dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-activemq</artifactId>
</dependency>
<dependency>
</dependency>
<groupId>org.apache.activemq</groupId>
<artifactId>activemq-pool</artifactId>
</dependency>
</pre>
```

2、配置相关信息

在 application.properties 加入 Active MQ 的相关配置信息

```
spring.activemq.broker-url=tcp://localhost:61616
spring.activemq.username=admin
spring.activemq.password=admin
spring.activemq.pool.enabled=true
</pre>
```

3、编写配置类

Spring Boot 规定必须配置 Active MQ 的连接工厂, 和 Spring 的配置方法一样, 只不过这是解与 Java 代码的方式进行配置

```
@Configuration
@EnableJms
public class ActivMqConfiguration {
```

```

</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> @Value("${spring.
g.activemq.user}")
</span></span><span class="highlight-line"><span class="highlight-cl"> private String u
rName;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> @Value("${spring
g.activemq.password}")
</span></span><span class="highlight-line"><span class="highlight-cl"> private String p
assword;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> @Value("${spring
g.activemq.broker-url}")
</span></span><span class="highlight-line"><span class="highlight-cl"> private String b
rokerUrl;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> @Bean
</span></span><span class="highlight-line"><span class="highlight-cl"> public Connecti
nFactory connectionFactory(){
</span></span><span class="highlight-line"><span class="highlight-cl"> ActiveMQCo
nnectionFactory connectionFactory
</span></span><span class="highlight-line"><span class="highlight-cl"> = new A
ctiveMQConnectionFactory();
</span></span><span class="highlight-line"><span class="highlight-cl"> connectionFa
ctory.setBrokerURL(brokerUrl);
</span></span><span class="highlight-line"><span class="highlight-cl"> connectionFa
ctory.setUsername(usrName);
</span></span><span class="highlight-line"><span class="highlight-cl"> connectionFa
ctory.setPassword(password);
</span></span><span class="highlight-line"><span class="highlight-cl"> return conne
tionFactory;
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> @Bean("jmsTop
icListenerContainerFactory")
</span></span><span class="highlight-line"><span class="highlight-cl"> public JmsListe
nerContainerFactory jmsTopicListenerContainerFactory(
</span></span><span class="highlight-line"><span class="highlight-cl"> Connectio
nFactory connectionFactory
</span></span><span class="highlight-line"><span class="highlight-cl"> ){
</span></span><span class="highlight-line"><span class="highlight-cl"> DefaultJmsLis
tnerContainerFactory factory
</span></span><span class="highlight-line"><span class="highlight-cl"> = new
DefaultJmsListenerContainerFactory();
</span></span><span class="highlight-line"><span class="highlight-cl"> factory.setCo
nnectionFactory(connectionFactory);
</span></span><span class="highlight-line"><span class="highlight-cl"> factory.setPu
bSubDomain(true);
</span></span><span class="highlight-line"><span class="highlight-cl"> return factory
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code></pre>

```

4-发送端代码-生产者- </h3>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
```

```

cl">@Service
</span></span><span class="highlight-line"><span class="highlight-cl">public class Produ
erQueue {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> @Autowired
</span></span><span class="highlight-line"><span class="highlight-cl"> private JmsMes
agingTemplate jmsTemplate;//简单操作
</span></span><span class="highlight-line"><span class="highlight-cl"> @Autowired
</span></span><span class="highlight-line"><span class="highlight-cl"> private JmsTe
plate template;//可以做更细微的控制
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> // 发送消息, de
tination是发送到的队列, message是待发送的消息
</span></span><span class="highlight-line"><span class="highlight-cl"> public void sen
MessageByJmsMessagingTemplate(Destination destination, final String message){
</span></span><span class="highlight-line"><span class="highlight-cl">     jmsTemplate.
onvertAndSend(destination, message);
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> public void sen
MessageByJmsTemplate(Destination destination, final String message){
</span></span><span class="highlight-line"><span class="highlight-cl">     template.sen
(destination, session -&gt; {
</span></span><span class="highlight-line"><span class="highlight-cl">         TextMessa
e msg = session.createTextMessage();
</span></span><span class="highlight-line"><span class="highlight-cl">         msg.setTex
(message + "othre information");
</span></span><span class="highlight-line"><span class="highlight-cl">         return msg
</span></span><span class="highlight-line"><span class="highlight-cl">     });
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl">>}
</span></span></code></pre>

```

5-接收端代码-消费者- >5、接收端代码 (消费者) </h3>

```

<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">@Component
</span></span><span class="highlight-line"><span class="highlight-cl">public class Cons
merAQueue {
</span></span><span class="highlight-line"><span class="highlight-cl"> // 使用JmsListe
er配置消费者监听的队列, 其中text是接收到的消息
</span></span><span class="highlight-line"><span class="highlight-cl"> @JmsListener(
estination = "springboot.queue")
</span></span><span class="highlight-line"><span class="highlight-cl"> public void rece
veQueue(String text){
</span></span><span class="highlight-line"><span class="highlight-cl">     System.out.pr
ntln(this.getClass().getName()+" receive msg:"+text);
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl">>}
</span></span></code></pre>

```

6-编写测试类 >6、编写测试类 </h3>

首先我们启动 Spring Boot 程序。

然后我们在 test 目录下编写测试类 </p>

```

<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">@RunWith(SpringRunner.class)

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">@SpringBootTest
</span></span><span class="highlight-line"><span class="highlight-cl">public class AmSpr
ngbootApplicationTests {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    @Autowired
</span></span><span class="highlight-line"><span class="highlight-cl">    private Produce
Queue producerQueue;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">    @Test
</span></span><span class="highlight-line"><span class="highlight-cl">    public void tes
t() {
</span></span><span class="highlight-line"><span class="highlight-cl">        Destination
</span></span><span class="highlight-line"><span class="highlight-cl">        = new A
ctiveMQQueue("springboot.queue");
</span></span><span class="highlight-line"><span class="highlight-cl">        for(int i=0; i<
=3; i++){
</span></span><span class="highlight-line"><span class="highlight-cl">            producer
</span></span><span class="highlight-line"><span class="highlight-cl">            .sendMessageByJmsMessagingTemplate(destination,
</span></span><span class="highlight-line"><span class="highlight-cl">            "NO:
</span></span><span class="highlight-line"><span class="highlight-cl">            +i+";my name is Mark!!!");
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span></code></pre>

```