

<https://github.com/killme2008/Metamorphosis/wiki/Spring-supports#%E8%AE%BE%E7%BD%AE%E9%BB%98%E8%AE%A4topic>

#### 相关链接

- [会话工厂MessageSessionFactory](#)
- [发送消息MessageProducer](#)
- [订阅消息MessageConsumer](#)
- [遍历消息TopicBrowser](#)

## 一. 配置消息会话工厂

在Spring容器内配置一个MessageSessionFactory:

```
<!-- message session factory -->
<bean id="sessionFactory"
      class="com.taobao.metamorphosis.client.extension.spring
        .MetaqMessageSessionFactoryBean">
  <property name="zkConnect" value="127.0.0.1:2181"/>
  <property name="zkSessionTimeoutMs" value="30000"/>
  <property name="zkConnectionTimeoutMs" value="30000"/>
  <property name="zkSyncTimeMs" value="5000"/>
</bean>
```

主要是zookeeper参数配置，需要跟服务端的zk配置保持一致。更多参数参见AbstractMetaqMessageSessionFactory的javadoc。

## 配置XA事务消息会话工厂

简单地将MetaqMessageSessionFactoryBean替换为XAMetaqMessageSessionFactoryBean即可，创建的对象将是XAMessageSessionFactory。

## 二. 消息体转换器

**MetaQ的消息载荷是一个byte数组，你需要将具体的消息内容序列化成为byte数组才能发送。** MessageBodyConverter接口用于做这个转换：

```
/** Message body object converter. */
public interface MessageBodyConverter<T> {

    /** Convert a message object to byte array. */
    public byte[] toByteArray(T body) throws MetaClientException;

    /** Convert a byte array to message object. */
    public T fromByteArray(byte[] bs) throws MetaClientException;
}
```

MetaQ客户端提供了Java序列化实现的

JavaSerializationMessageBodyConverter给你使用，你也可以自定义自己的消息body转换器，比如采用其他序列化协议，如protobufs,hessian等。

配置一个消息body转换器，比如我们就使用Java序列化：

```
<!-- message body converter using java serialization. -->
<bean id="messageBodyConverter"
      class="com.taobao.metamorphosis.client.extension.spring
           .JavaSerializationMessageBodyConverter"/>
```

## 三. 使用MetaqTemplate发送消息

### 1. 配置MetaqTemplate:

MetaqTemplate用于发送MetaQ消息，基本配置：

```
<!-- template to send messages. -->
<bean id="metaqTemplate"
      class="com.taobao.metamorphosis.client.extension.spring.MetaqTemplate"
      <property name="messageSessionFactory" ref="sessionFactory"/>
      <property name="messageBodyConverter"
ref="messageBodyConverter"/>
</bean>
```

metaqTemplate用到了上面配置的messageSessionFactory和messageBodyConverter，用来创建producer和转换消息体。

配置了metaqTemplate之后，你将可以在你要发送消息的JavaBean里引用这个

对象，并使用它发送消息。

## 2. MessageBuilder创建消息并发送

MetaqTemplate的send方法不直接接收Message对象，而是MessageBuilder构建器，用来构建消息：

```
final String topic = "date";  
final SendResult sendResult  
    = template.send(MessageBuilder.withTopic(topic).withBody(new Date()));
```

上面的例子发送topic为date的消息，消息体为java.util.Date对象，MetaqTemplate将调用messageBodyConverter将消息体的date对象转换为byte数组，构建message对象再发送。

因为我们使用了JavaSerializationMessageBodyConverter做消息体转换，因此任何可序列化的JavaBean其实都可以作为消息体发送，使用MessageBuilder.withBody方法设置即可。如果还是想发送自定义的byte数组，可以使用MessageBuilder.withPayload(byte [])设置。但是两者不能同时设置。

send(MessageBuilder)方法同样有重载方法用于异步发送和设置请求超时，具体见javadoc。

## 3. 共享MessageProducer

MetaqTemplate会为每个发送的topic创建一个MessageProducer并发布topic，你也可以共享一个MessageProducer来发送多个topic的消息，只要在配置metqTempalte的设置属性shareProducer为true即可：

```
<!-- template to send messages. -->  
<bean id="metaqTemplate"  
    class="com.taobao.metamorphosis.client.extension.spring.MetaqTemplate">  
    <property name="messageSessionFactory" ref="sessionFactory"/>  
    <property name="messageBodyConverter"  
ref="messageBodyConverter"/>  
    <property name="shareProducer" value="true"/>  
</bean>
```

## 4. 设置默认topic

同样，MetaqTemplate可设置默认topic，当使用共享MessageProducer的时候，可设置这个MessageProducer的默认topic，关于默认topic的作用参见[发](#)

送消息MessageProducer一节。

```
<!-- template to send messages. -->
<bean id = "metaqTemplate"
      class="com.taobao.metamorphosis.client.extension.spring.MetaqTemplate"
      >
    <property name="messageSessionFactory" ref="sessionFactory"/>
    <property name="messageBodyConverter"
ref="messageBodyConverter"/>
    <property name="shareProducer" value="true"/>
    <property name="defaultTopic" value="*/>
</bean>
```

## 四. 订阅消息

MetaQ同样提供了一套API来支持使用Spring框架订阅消息，你除了需要编写消息处理器之外，几乎不用再写代码。

### 1. 配置订阅Topic

例如你想订阅topic为date的消息，首先需要配置该topic:

```
<!-- topics to be subscribed. -->
<bean id = "dateTopic"
      class="com.taobao.metamorphosis.client.extension.spring.MetaqTopic"
      >
    <!-- consumer group -->
    <property name="group" value="testGroup"/>
    <!-- topic -->
    <property name="topic" value="date"/>
    <!-- max buffer size to fetch messages -->
    <property name="maxBufferSize" value="16384"/>
</bean>
```

配置了订阅者的分组，想要订阅的topic以及maxBufferSize大小，更多参数见MetaqTopic的javadoc。

### 2. 继承DefaultMessageListener

接下来创建消息处理器，继承DefaultMessageListener类并实现onReceiveMessages(MetaqMessage msg)方法即可：

```
package com.taobao.metamorphosis.example.spring;
import
com.taobao.metamorphosis.client.extension.spring.DefaultMessageListener;
```



```
import com.taobao.metamorphosis.client.extension.spring.MetaqMessage;
import java.util.Date;

/** Process date messages listener. */
public class DateMessageListener extends DefaultMessageListener<Date> {

    @Override
    public void onReceiveMessages(MetaqMessage<Date> msg) {
        Date date = msg.getBody();
        System.out.println("receive date message:" + date);
    }
}
```

onReceiveMessages收到的是封装Message后的MetaqMessage对象，它会使用你配置的消息体转换器将byte数组转换成java.util.Date对象，你可以直接通过getBody获取上文metaqTemplate例子中发送的日期对象。当然，如果你还需要原始的Message对象，可以通过getRawMessage()方法获取：

```
Message rawMsg = msg.getRawMessage();
```

编写好消息处理器之后，需要配置消息处理器：

```
<!-- message listener -->
<bean id= "messageListener"
class="com.taobao.metamorphosis.example.spring.DateMessageListener">
    <!-- threads to process these messages. -->
    <property name="processThreads" value="10"/>
</bean>
```

processThreads设置用于处理消息的线程数，如果为0或者负数表示在抓取线程中处理消息。

### 3. 配置MessageListenerContainer

有了MetaqTopic和消息处理器之后，我们就可以使用MessageListenerContainer来真正地订阅消息了：

```
<!-- listener container to subscribe topics -->
<bean id = "listenerContainer"
class="com.taobao.metamorphosis.client.extension.spring
.MessageListenerContainer">
    <property name="messageSessionFactory" ref="sessionFactory"/>
    <property name="messageBodyConverter"
ref="messageBodyConverter"/>
    <property name="subscribers">
        <map>
```

```

        <entry key-ref="dateTopic" value-ref="messageListener"/>
    </map>
</property>
</bean>

```

在某些版本的Spring里，可能要求subscribers的配置是这样：

```
<entry key="bean:name=dateTopic" value-ref="messageListener"/>
```

注意到listenerContainer引用到messageSessionFactory和messageBodyConverter，用于创建consumer和转换收到的消息体。**subscribes**是一个map，指定了订阅的topic以及对应的messageListener。你可以订阅多个topic。MessageListenerContainer同时可以配置全局的RejectConsumptionHandler。

## 4. 共享订阅者

同样，MessageListenerContainer会为每个topic创建一个MessageConsumer并自动调用subscribe和completeSubscribe方法。但是有时候你可能想共享MessageConsumer，这可以通过设置shareConsumer为true来实现，同样你也可以设置共享的MessageConsumer默认订阅的topic，比如前面的配置也可以修改为：

```

<!-- listener container to subscribe topics -->
<bean id="listenerContainer"
    class="com.taobao.metamorphosis.client.extension.spring
        .MessageListenerContainer">
    <property name="messageSessionFactory" ref="sessionFactory"/>
    <property name="messageBodyConverter"
ref="messageBodyConverter"/>
    <property name="shareConsumer" value="true"/>
    <property name="defaultTopic" ref="dateTopic"/>
    <property name="defaultMessageListener" ref="messageListener"/>
</bean>

```

## 五. 资源释放

MetaqMessageSessionFactoryBean, MetaqTemplate, DefaultMessageListener和MessageListenerContainer都能正确地处理资源释放，无需额外代码和配置。