根据项目需求，选用RabbitMQ作为消息队列，作为应用之间的传参介质。RabbitMQ集群已经搭建完成，可以直接使用，下面主要介绍一下RabbitMQ在项目中的使用方法。

1. 配置相关jar包，在POM文件加入以下依赖，RabbitMQ客户端依赖于Spring，需加入spring的相关jar包，如以包含，请忽略。

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEn coding>

<!-- RabbitMq客户端版本号 -->

<rabbitClient.version>1.3.5.RELEASE</rabbitClient.version>

<!-- spring版本号 -->

<spring.version>4.0.2.RELEASE</spring.version>

</properties>

<dependencies>

<!--rabbitmq依赖 -->

<dependency>

<groupId>org.springframework.amqp</groupId>

<artifactId>spring-rabbit</artifactId>

<version>${rabbitClient.version}</version>

</dependency>

<!-- 添加Spring依赖 -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context-support</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aspects</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-tx</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${spring.version}</version>

</dependency>

1. 在src/main/resources加入配置文件application-rabbitMQ.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:rabbit=*"http://www.springframework.org/schema/rabbit"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-3.1.xsd*

*http://www.springframework.org/schema/rabbit*

*http://www.springframework.org/schema/rabbit/spring-rabbit-1.0.xsd"*>

<!-- 激活annotation功能 -->

<context:annotation-config />

<!--配置connection-factory，指定连接rabbit server参数 -->

<rabbit:connection-factory id=*"connectionFactory"*

username=*"admin"* password=*"admin"* host=*"localhost"* port=*"5672"* />

<!--定义rabbit template用于数据的接收和发送 -->

<rabbit:template id=*"amqpTemplate"* connection-factory=*"connectionFactory"*

exchange=*"exchangeTest"* />

<!--通过指定下面的admin信息，当前producer中的exchange和queue会在rabbitmq服务器上自动生成 -->

<rabbit:admin connection-factory=*"connectionFactory"* />

<!--定义queue -->

<rabbit:queue name=*"queueTest"* durable=*"true"* auto-delete=*"false"*

exclusive=*"false"* />

<!-- 定义direct exchange，绑定queueTest -->

<rabbit:direct-exchange name=*"exchangeTest"* durable=*"true"* auto-delete=*"false"*>

<rabbit:bindings>

<rabbit:binding queue=*"queueTest"* key=*"queueTestKey"*></rabbit:binding>

</rabbit:bindings>

</rabbit:direct-exchange>

<!-- 消息接收者 -->

<bean id=*"MessageProducer"* class=*"com.amqp.producer.MessageProducer"*></bean>

<!-- 消息接收者 -->

<bean id=*"messageReceiver"* class=*"com.amqp.consumer.MessageConsumer"*></bean>

<!-- queue litener 观察 监听模式 当有消息到达时会通知监听在对应的队列上的监听对象 -->

<rabbit:listener-container connection-factory=*"connectionFactory"*>

<rabbit:listener queues=*"queueTest"* ref=*"messageReceiver"* />

</rabbit:listener-container>

</beans>

1. 配置web.xml

确认web.xml有如下配置，如没有，需加入该配置

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:application\*.xml</param-value>

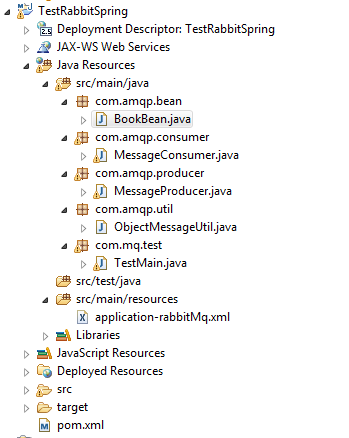
</context-param>

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

1. Demo实例请参考TestRabbitSpring工程



MessageConsumer/MessageConsumer是mq的生产者类和消费者类

BookBean是测试VO发送接收的对象

ObjectMessageUtil是发送和接收过程中，VO和字符串转换的工具类，里面提供了两个静态方法

TestMain里演示了字符串和VO的发送接收方法（本项目的消费者采用的是监听模式）。