



链滴

创建和使用 Springboot+ActiveMQ 项目

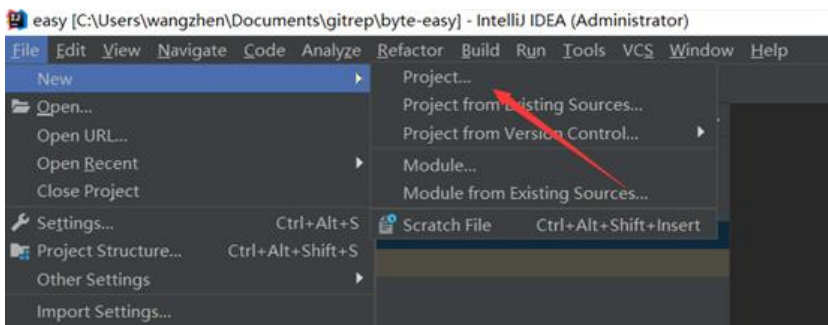
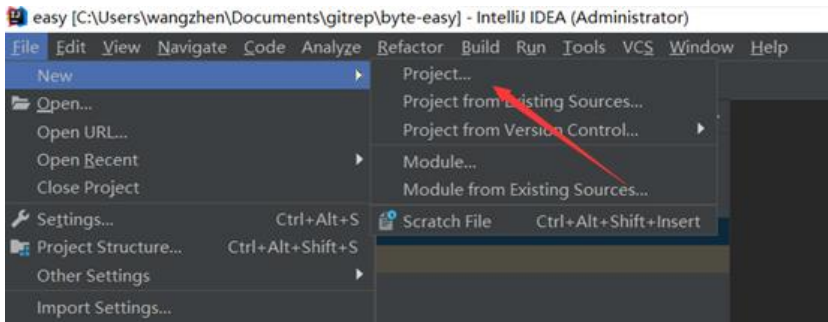
作者: [byte2018](#)

原文链接: <https://ld246.com/article/1557054493391>

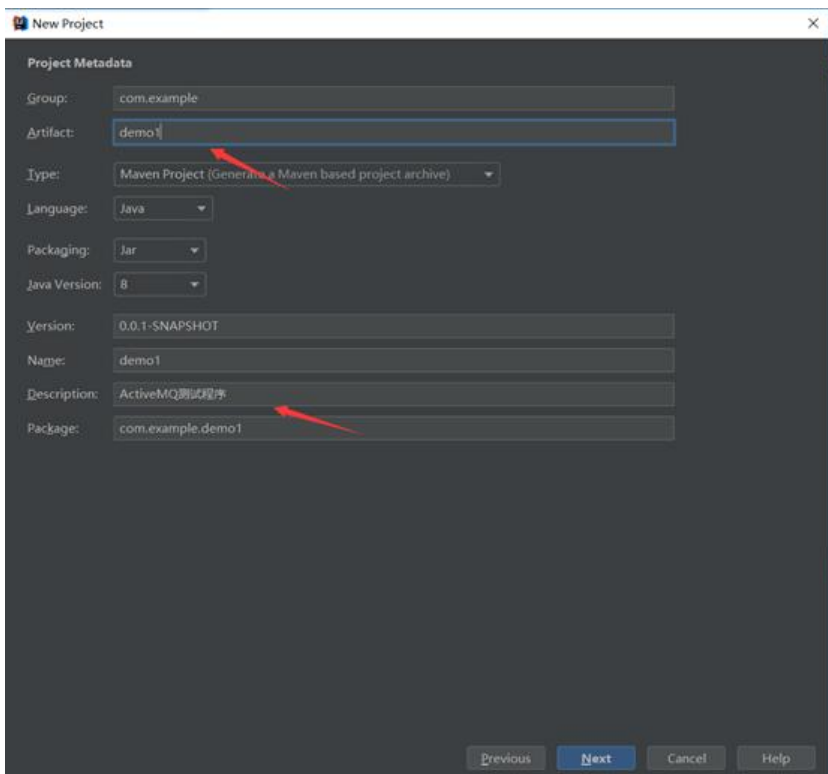
来源网站: [链滴](#)

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

- 1、 本项目使用IDEA工具搭建，项目可以直接导入到eclipse中，如果使用eclipse创建，请自行百度。
- 2、 首先新建项目。

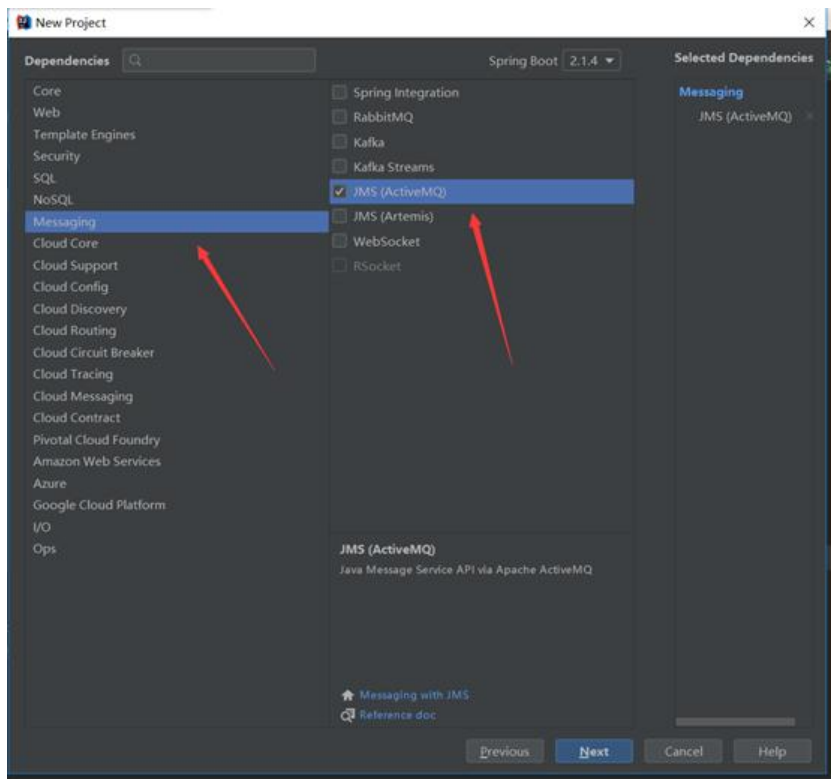


- 3、 点击next



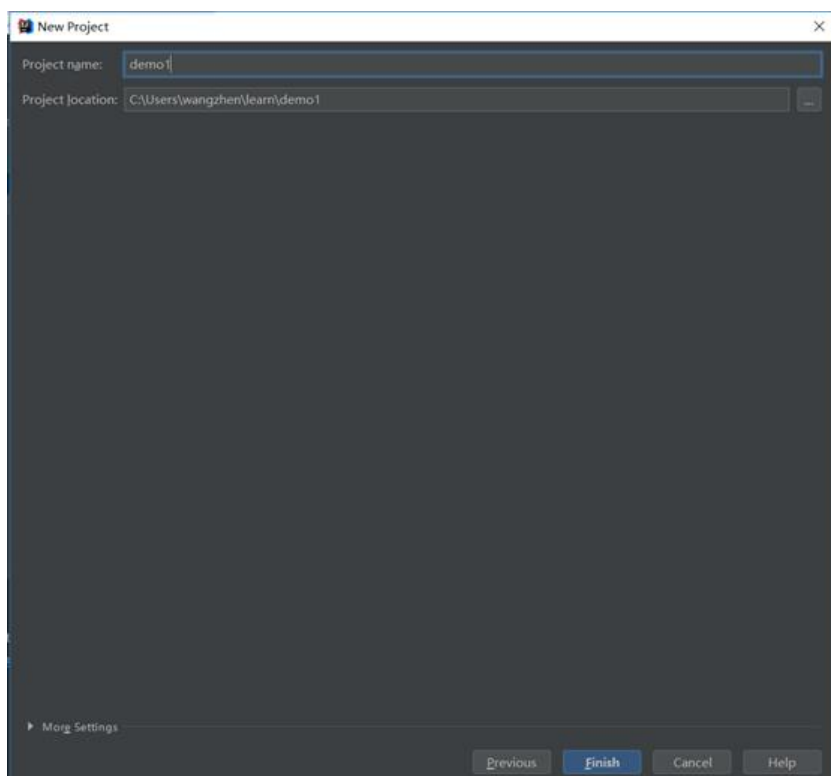
填写自己想要的信息

- 4、 点击next

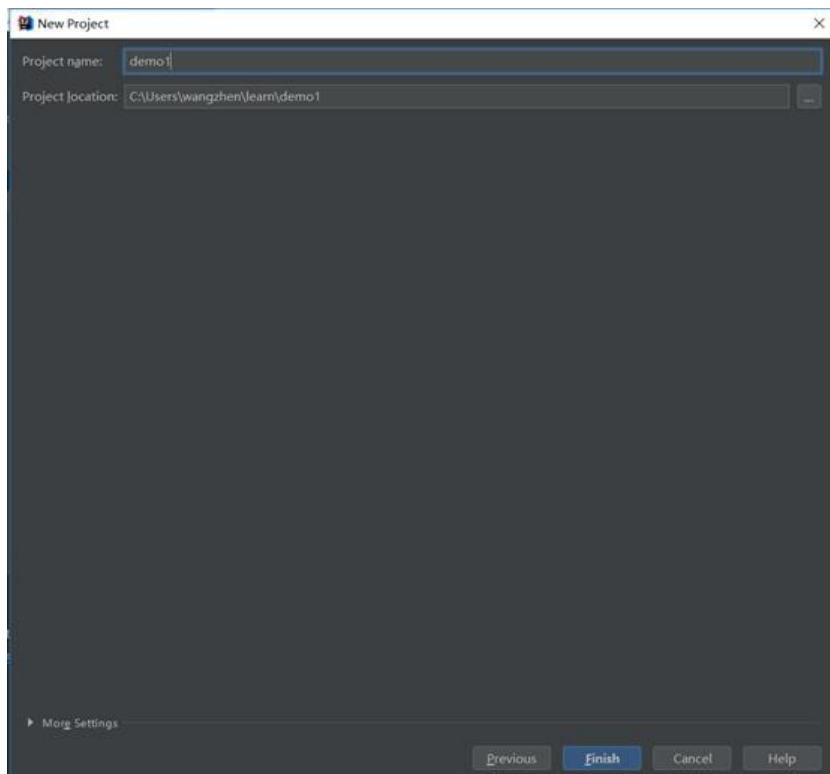


选择箭头选择的

5、 点击next

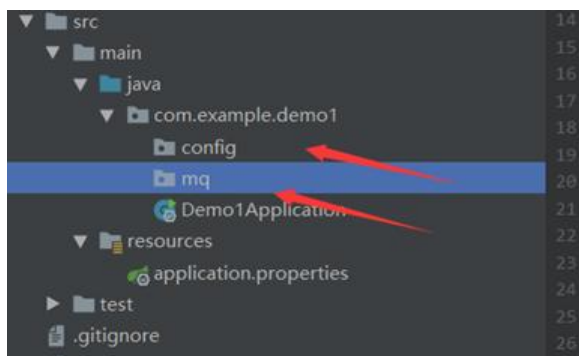


6、 点击finish，等待程序初始化和加载包完成。



pom文件中添加了对activemq的支持

7、 先创建包，出现如下程序结构



8、 创建ActiveMQ配置类

```
package com.mbyte.easy.config;
```

```
import javax.jms.Queue;  
import javax.jms.Topic;
```

```
import org.apache.activemq.command.ActiveMQQueue;  
import org.apache.activemq.command.ActiveMQTopic;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.jms.annotation.EnableJms;
```

```
/**  
 * @program: easy
```

```

* @description: ActiveMQ配置类
* @author: 王震
* @create: 2019-05-05 11:26
**/
@Configuration
@EnableJms
public class ActiveMQConfiguration {

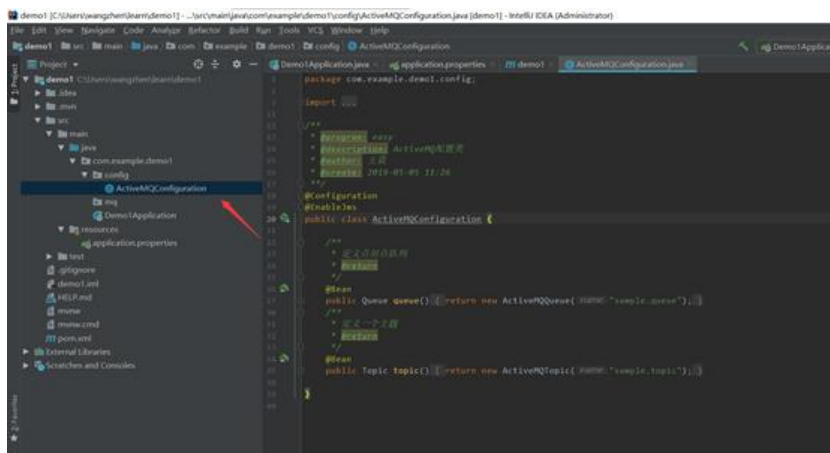
    /**
     * 定义点对点队列
     * @return
     */
    @Bean
    public Queue queue() {
        return new ActiveMQQueue("sample.queue");
    }

    /**
     * 定义一个主题
     * @return
     */
    @Bean
    public Topic topic() {
        return new ActiveMQTopic("sample.topic");
    }

}

```

程序结构如下图所示



9、 创建生产者和消费者

定义生产者：

```
package com.mbyte.easy.mq;
```

```

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Configuration;
import org.springframework.jms.core.JmsMessagingTemplate;
import org.springframework.scheduling.annotation.Scheduled;

```

```
import javax.jms.Queue;
```

```

import javax.jms.Topic;

/**
 * @program: easy
 * @description: 定义生产者
 * @author: 王震
 * @create: 2019-05-05 11:29
 */
@Configuration
public class Producer {

    @Autowired
    private JmsMessagingTemplate jmsMessagingTemplate;

    @Autowired
    private Queue queue;

    @Autowired
    private Topic topic;

    /**
     * 每5S执行一次
     */
    @Scheduled(fixedRate = 5000, initialDelay = 3000)
    public void send() {
        //发送队列消息
        this.jmsMessagingTemplate.convertAndSend(this.queue, "生产者: http://www.byte.ac.cn"
);
        System.out.println("生产者: 生产者: http://www.byte.ac.cn");
        //发送订阅消息
        this.jmsMessagingTemplate.convertAndSend(this.topic, "生产者生产的订阅/发布消息: 生
者: http://www.byte.ac.cn");
        System.out.println("生产者: 生产的订阅/发布消息: http://www.byte.ac.cn");
    }

}

```

定义消费者

```

package com.mbyte.easy.mq;

import org.springframework.jms.annotation.JmsListener;
import org.springframework.stereotype.Component;

/**
 * @program: easy
 * @description: 定义消费者
 * @author: 王震
 * @create: 2019-05-05 11:32
 */
@Component
public class Consumer {

```

```

    @JmsListener(destination = "sample.queue")
    public void receiveQueue(String text) {
        System.out.println("消费者：来源于生产者的消息：" + text);
    }

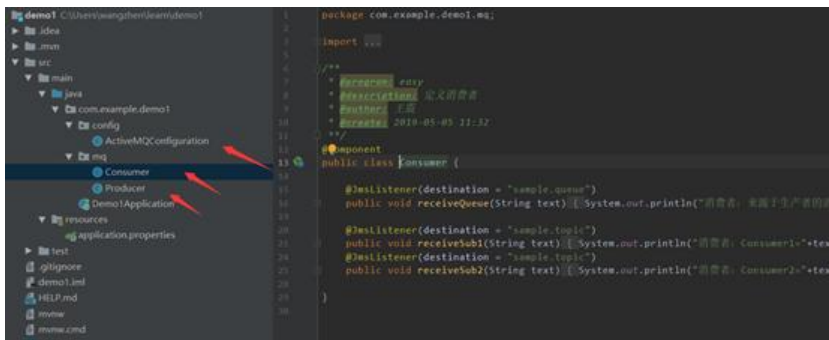
    @JmsListener(destination = "sample.topic")
    public void receiveSub1(String text) {
        System.out.println("消费者： Consumer1=" + text);
    }

    @JmsListener(destination = "sample.topic")
    public void receiveSub2(String text) {
        System.out.println("消费者： Consumer2=" + text);
    }

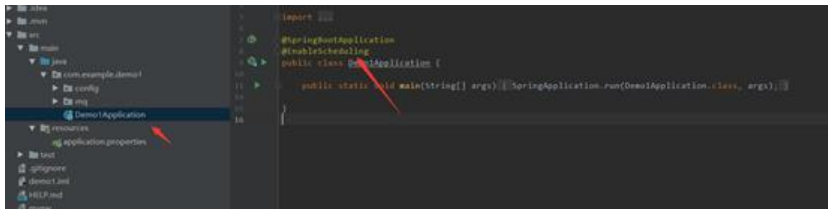
}

```

最终代码结构如下图所示：



10、 Springboot需要开启调度，生产数据，在Demo1Application中添加@EnableScheduling注解



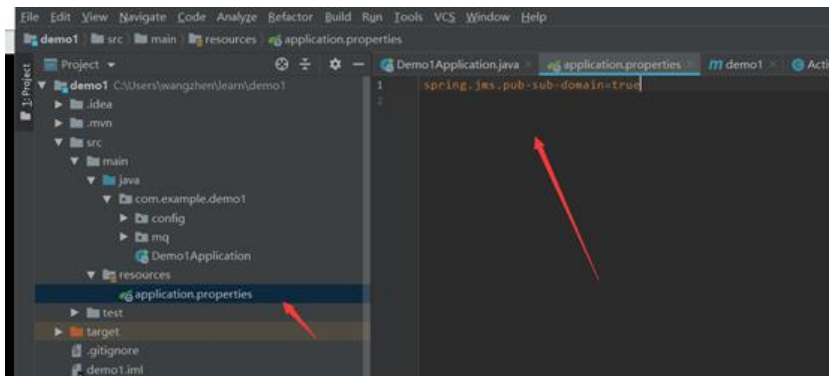
11、 现在开始运行程序



12、 程序正常运行

```
2019-05-05 18:47:52.945 INFO 7764 --- [main] o.a.activemq.broker...
2019-05-05 18:47:52.975 INFO 7764 --- [main] com.example.demo1
生产者: 生产者: http://www.byte.ac.cn
消费者: 来源于生产者的消息: 生产者: http://www.byte.ac.cn
生产者: 生产的订阅/发布消息: http://www.byte.ac.cn
生产者: 生产者: http://www.byte.ac.cn
消费者: 来源于生产者的消息: 生产者: http://www.byte.ac.cn
生产者: 生产的订阅/发布消息: http://www.byte.ac.cn
生产者: 生产者: http://www.byte.ac.cn
生产者: 生产的订阅/发布消息: http://www.byte.ac.cn
消费者: 来源于生产者的消息: 生产者: http://www.byte.ac.cn
生产者: 生产者: http://www.byte.ac.cn
生产者: 生产的订阅/发布消息: http://www.byte.ac.cn
```

13、 Spring boot默认P2P的模式, 启用发布订阅模式, 可以在application.properties中添加spring.jms.pub-sub-domain=true



14、 再次运行程序

```
2019-05-05 18:50:54.771 INFO 14072 --- [main] o.apache.activemq...
2019-05-05 18:50:54.771 INFO 14072 --- [main] o.apache.activemq...
2019-05-05 18:50:54.790 INFO 14072 --- [main] o.a.activemq.broke...
2019-05-05 18:50:54.816 INFO 14072 --- [main] com.example.demo1.
生产者: 生产者: http://www.byte.ac.cn
生产者: 生产的订阅/发布消息: http://www.byte.ac.cn
消费者: Consumer1=生产者生产的订阅/发布消息: 生产者: http://www.byte.ac.cn
消费者: Consumer2=生产者生产的订阅/发布消息: 生产者: http://www.byte.ac.cn
```

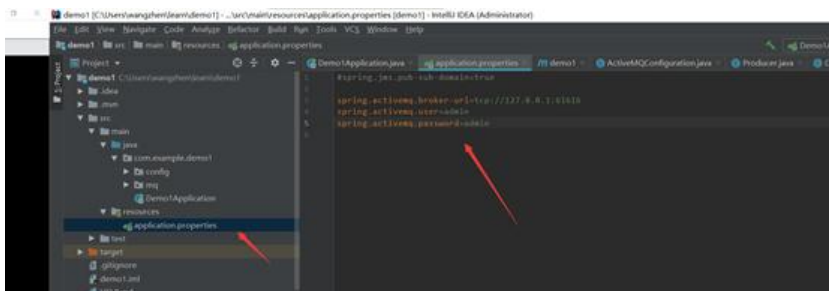
发布订阅模式已经启用

15、 将ActiveMQ配置成单独的服务, 连接ActiveMQ服务, 可以在可以在application.properties添加

spring.activemq.broker-url=tcp://127.0.0.1:61616

spring.activemq.user=admin

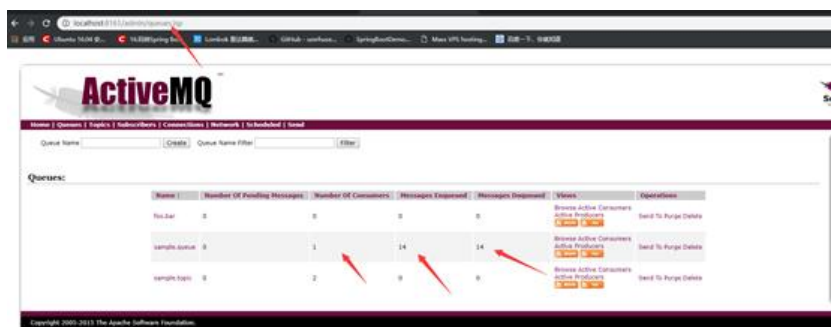
spring.activemq.password=admin



16、 重新启动程序


```
生产者: 生产者: http://www.byte.ac.cn
生产者: 生产的订阅/发布消息: http://www.byte.ac.cn
消费者: 来源于生产者的消息: 生产者: http://www.byte.ac.cn
生产者: 生产者: http://www.byte.ac.cn
生产者: 生产的订阅/发布消息: http://www.byte.ac.cn
消费者: 来源于生产者的消息: 生产者: http://www.byte.ac.cn
生产者: 生产者: http://www.byte.ac.cn
生产者: 生产的订阅/发布消息: http://www.byte.ac.cn
消费者: 来源于生产者的消息: 生产者: http://www.byte.ac.cn
生产者: 生产者: http://www.byte.ac.cn
生产者: 生产的订阅/发布消息: http://www.byte.ac.cn
```

17、 程序正常运行，查看ActiveMQ管理后台



已经连接到ActiveMQ服务。

18、 程序下载