

[博客园](#)[首页](#)[新随笔](#)[联系](#)[订阅](#)[管理](#)

## 公告



昵称： 风一样的码农

园龄： 4年

粉丝： 815

关注： 0

[+加关注](#)

## 搜索

[找找看](#)

## 常用链接

[我的随笔](#)[我的评论](#)[我的参与](#)[最新评论](#)[我的标签](#)

## 积分与排名

积分 - 461589

排名 - 542

## 随笔分类 (226)

[ActiveMQ\(1\)](#)[Apache Commons\(6\)](#)

## 基于Tomcat + JNDI + ActiveMQ实现JMS的点对点消息传送

### 阅读目录

- ❖ [前言](#)
- ❖ [什么是JNDI](#)
- ❖ [使用Tomcat配置JNDI](#)
- ❖ [启动ActiveMQ](#)
- ❖ [编写一个Web工程](#)
- ❖ [验证结果](#)
- ❖ [参考资料](#)

### 前言

写了一个简单的JMS例子，之所以使用JNDI 是出于通用性考虑，该例子使用JMS规范提供的通用接口，没有使用具体JMS提供者的接口，这样可以保证我们编写的程序适用于任何一种JMS实现(ActiveMQ、HornetQ...)。

### 什么是JNDI

JNDI(Java Naming and Directory Interface)是一个标准规范，类似于JDBC，JMS等规范，为开发人员提供了查找和访问各种命名和目录服务的通用、统一的接口。J2EE 规范要求所有 J2EE 容器都要提供 JNDI 规范的实现，因此Tomcat就实现了JNDI 规范。

### 使用Tomcat配置JNDI

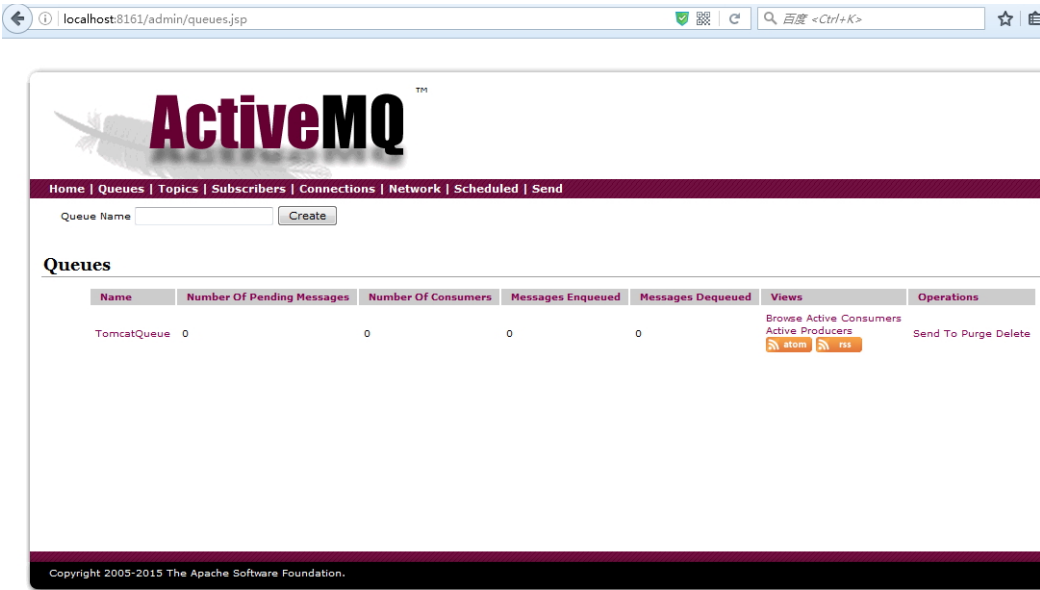
找到Tomcat安装路径下的conf文件夹，打开context.xml，添加如下配置：

```
<?xml version="1.0" encoding="UTF-8"?>
<Context path="/" xml:base="file:///C:/Program%20Files/Tomcat%206.0/conf/">
  <Resource name="queue/connectionFactory"
    auth="Container"
    type="org.apache.activemq.ActiveMQConnectionFactory"
    description="JMS Connection Factory"
    factory="org.apache.activemq.jndi.JNDIReferenceFactory"
    brokerURL="tcp://localhost:61616"
    brokerName="LocalActiveMQBroker" />
  <Resource name="queue/queue0"
    auth="Container"
    type="org.apache.activemq.command.ActiveMQQueue"
    description="My Queue"
    factory="org.apache.activemq.jndi.JNDIReferenceFactory"
    physicalName="TomcatQueue" />
</Context>
```

|                       |
|-----------------------|
| Apache Flume(4)       |
| interface21(3)        |
| JAVA 8(7)             |
| JAVA 9(1)             |
| JAVA IO&NIO(8)        |
| JAVA JVM(3)           |
| JAVA SOA(2)           |
| JAVA Web(11)          |
| JAVA 并发(12)           |
| Java 持续集成(7)          |
| JAVA 工具(1)            |
| JAVA 基础(12)           |
| JAVA 集合(8)            |
| JAVA 应用(4)            |
| Lucene&Solr(3)        |
| Mybatis-3             |
| Redis&Jms(7)          |
| Spring Boot(15)       |
| Spring cloud(1)       |
| Spring Data(2)        |
| Spring Framework(17)  |
| Spring IO Platform(3) |
| 不断更新系列(2)             |
| 常见问题(2)               |
| 环境搭建(3)               |
| 前端开发(5)               |

## 启动ActiveMQ

CMD到ActiveMQ安装路径下的bin目录，输入“activemq start”指令即可启动，可在浏览器中输入地址<http://localhost:8161/admin>，查看队列、话题等信息。



## 编写一个Web工程

eclipse上新建web工程，添加ActiveMQ依赖的jar包，然后开始编写两个Servlet，一个用于生产消息，另一个用于消费消息，如下代码：

消息生产者Servlet：

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.jms.DeliveryMode;
import javax.jms.Queue;
import javax.jms.QueueConnection;
import javax.jms.QueueConnectionFactory;
import javax.jms.QueueSender;
import javax.jms.QueueSession;
import javax.jms.Session;
import javax.jms.TextMessage;
import javax.naming.InitialContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class JMSTest
 */
@WebServlet("/Send")
public class Send extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
```

[设计模式\(25\)](#)[生活随笔\(2\)](#)[数据库\(9\)](#)[未分类\(8\)](#)[英文翻译\(32\)](#)[随笔档案 \(214\)](#)[2019年3月\(10\)](#)[2018年10月\(5\)](#)[2018年9月\(5\)](#)[2018年8月\(3\)](#)[2018年5月\(3\)](#)[2017年8月\(1\)](#)[2017年7月\(2\)](#)[2017年6月\(4\)](#)[2017年5月\(4\)](#)[2017年3月\(1\)](#)[2017年2月\(5\)](#)[2017年1月\(8\)](#)[2016年12月\(12\)](#)[2016年11月\(11\)](#)[2016年10月\(9\)](#)[2016年9月\(7\)](#)[2016年8月\(1\)](#)[2016年7月\(6\)](#)[2016年6月\(13\)](#)[2016年5月\(34\)](#)

```
* @see HttpServlet#HttpServlet()
*/
public Send() {
    super();
    // TODO Auto-generated constructor stub
}

/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
 * response)
 */
protected void doGet(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    PrintWriter out = response.getWriter();

    try {
        // get the initial context
        InitialContext context = new InitialContext();

        // lookup the queue object
        Queue queue = (Queue) context.lookup("java:comp/env/queue/queue0");

        // lookup the queue connection factory
        QueueConnectionFactory conFactory = (QueueConnectionFactory) context
            .lookup("java:comp/env/queue/connectionFactory");

        // create a queue connection
        QueueConnection queConn = conFactory.createQueueConnection();

        // create a queue session
        QueueSession queSession = queConn.createQueueSession(false,
            Session.DUPS_OK_ACKNOWLEDGE);

        // create a queue sender
        QueueSender queSender = queSession.createSender(queue);
        queSender.setDeliveryMode(DeliveryMode.NON_PERSISTENT);

        // create a simple message to say "Hello World"
        TextMessage message = queSession.createTextMessage("Hello World");

        // send the message
        queSender.send(message);

        // print what we did
        out.write("Message Sent: " + message.getText());

        // close the queue connection
        queConn.close();
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
 * response)
 */
protected void doPost(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
}
```

|              |
|--------------|
| 2016年4月 (13) |
| 2016年3月 (14) |
| 2016年2月 (21) |
| 2016年1月 (15) |
| 2015年8月 (1)  |
| 2015年6月 (1)  |
| 2015年3月 (1)  |
| 2014年12月 (4) |

|  |
|--|
| 阅读排行榜  |
| 1. Mysql中的存储过程(154880)                             |
| 2. Java中的关键字 transient(86256)                      |
| 3. 发现一个国内牛逼的maven仓库, 速度真的太快了(84565)                |
| 4. Mysql中的视图(73350)                                |
| 5. JMS(Java消息服务)入门教程(68821)                        |
| 6. Spring Data JPA例子[基于Spring Boot、Mysql](42983)   |
| 7. Mysql 中的事件//定时任务(35694)                         |
| 8. 一个简单的Java web服务器实现(35318)                       |
| 9. 使用iText库创建PDF文件(32623)                          |
| 10. Tomcat中的Session小结(31619)                       |
| 11. jstack简单使用, 定位死循环、线程阻塞、死锁等问题(25906)            |
| 12. 关于JAVA中子类 and 父类的构造方法(21868)                   |
| 13. Spring Session - 使用Redis存储HttpSession例子(19875) |
| 14. 【译】Spring 4 @PropertySource                    |

}



消息消费者Servlet:



```
import java.io.IOException;
import java.io.PrintWriter;

import javax.jms.Queue;
import javax.jms.QueueConnection;
import javax.jms.QueueConnectionFactory;
import javax.jms.QueueReceiver;
import javax.jms.QueueSession;
import javax.jms.Session;
import javax.jms.TextMessage;
import javax.naming.InitialContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class Receive
 */
@WebServlet("/Receive")
public class Receive extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Receive() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {
        PrintWriter out = response.getWriter();

        try {
            // get the initial context
            InitialContext context = new InitialContext();

            // lookup the queue object
            Queue queue = (Queue) context.lookup("java:comp/env/queue/queue0");

            // lookup the queue connection factory
            QueueConnectionFactory conFactory = (QueueConnectionFactory) context
                .lookup("java:comp/env/queue/connectionFactory");

            // create a queue connection
            QueueConnection queConn = conFactory.createQueueConnection();

            // create a queue session
```

e和@Value注解示例(18403)

15. Java数组在内存中是如何存放的(17838)

评论排行榜

1. spring + spring mvc + mybatis + react + reflux + webpack Web工程例子(27)

2. Tomcat中的Session小结(15)

3. 发现一个国内牛逼的maven仓库, 速度真的太快了(15)

4. HashMap源码分析(14)

5. Mysql中的视图(13)

推荐排行榜

1. Java中的关键字 transient(25)

2. Mysql中的存储过程(17)

3. Tomcat中的Session小结(14)

4. JMS(Java消息服务)入门教程(13)

5. 发现一个国内牛逼的maven仓库, 速度真的太快了(12)

6. JAVA并发编程J.U.C学习总结(12)

7. Mysql中的视图(11)

8. 使用iText库创建PDF文件(9)

9. ArrayList(8)

10. JAVA 8 函数式接口 - Functional Interface(8)

11. spring + spring mvc + mybatis + react + reflux + webpack Web工程例子(7)

12. HashMap源码分析(7)

13. Spring Data JPA例子[基于Spring Boot、Mysql](6)

```
QueueSession queSession = queConn.createQueueSession(false,
    Session.AUTO_ACKNOWLEDGE);

// create a queue receiver
QueueReceiver queReceiver = queSession.createReceiver(queue);

// start the connection
queConn.start();

// receive a message
TextMessage message = (TextMessage) queReceiver.receive();

// print the message
out.write("Message Received: " + message.getText());

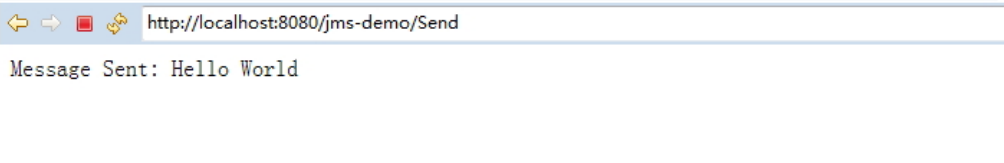
// close the queue connection
queConn.close();
} catch (Exception e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
}

}
```

验证结果

在Tomcat里运行该Web工程，执行消息生产者Servlet，返回消息发送成功标志，同时我们可以在<http://localhost:8161/admin/queues.jsp>查看到该消息，如下图所示



### 15. LinkedList(5)

## 参考资料

<http://howtodoinjava.com/jms/jms-point-to-point-message-example/>

@Author 风一样的码农

@HomePageUrl <http://www.cnblogs.com/chenpi/>

@Copyright 转载请注明出处, 谢谢~

分类: [Redis&Jms](#)

标签: [JAVA](#), [jms](#), [ActiveMQ](#), [JNDI](#)

好文要顶

关注我

收藏该文



风一样的码农

关注 - 0

粉丝 - 815

+加关注

3

0

« 上一篇: [JMS\(Java消息服务\)入门教程](#)

» 下一篇: [JMS发布/订阅消息传送例子](#)

posted @ 2016-06-07 00:06 风一样的码农 阅读(16443) 评论(3) 编辑 收藏

## 评论列表

#1楼 2017-05-16 14:22 hahah22

按照你这个例子运行为什么老是404

支持(0) 反对(0)

#2楼 2017-08-21 09:10 异想天开361

博主关于Tomcat的context.xml配置文件添加内容有一点没说清, 就是需要把需添加的内容放入<context>标签内部

支持(0) 反对(0)

#3楼 2018-04-14 12:13 陈跃进同志

排版很好,赞

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问](#) 网站首页。

Copyright © 2020 风一样的码农  
Powered by .NET Core 3.1.1 on Linux

