Atitit mq读取队列信息 范例

目录

[1.1. 读取原理与主要流程 1](#_Toc16159)

[1.2. 范例项目 C:\0wkspc\MqDemoPrj 1](#_Toc16612)

[1.3. 范例代码 1](#_Toc26933)

## 读取原理与主要流程

和数据库一样，mq可以看作一个高性能数据库

建立conn，打开session

连接队列按照队列名称，，队列可以看作一个表格

session.createQueue("text-msg");

//根据session，创建一个接收者对象

consumer = session.createConsumer(destination);

//实现一个消息的监听器

## 范例项目 C:\0wkspc\MqDemoPrj

## 范例代码

import javax.jms.Connection;

import javax.jms.ConnectionFactory;

import javax.jms.Destination;

import javax.jms.JMSException;

import javax.jms.Message;

import javax.jms.MessageConsumer;

import javax.jms.MessageListener;

import javax.jms.Session;

import javax.jms.TextMessage;

import org.apache.activemq.ActiveMQConnectionFactory;

public class PTPReceive {

//连接账号

private String userName = "";

//连接密码

private String password = "";

//连接地址

private String brokerURL = "tcp://localhost:61616";

//connection的工厂

private ConnectionFactory factory;

//连接对象

private Connection connection;

//一个操作会话

private Session session;

//目的地，其实就是连接到哪个队列，如果是点对点，那么它的实现是Queue，如果是订阅模式，那它的实现是Topic

private Destination destination;

//消费者，就是接收数据的对象

private MessageConsumer consumer;

public static void main(String[] args) {

PTPReceive receive = new PTPReceive();

receive.start();

}

public void start(){

try {

//根据用户名，密码，url创建一个连接工厂

factory = new ActiveMQConnectionFactory(userName, password, brokerURL);

//从工厂中获取一个连接

connection = factory.createConnection();

//测试过这个步骤不写也是可以的，但是网上的各个文档都写了

connection.start();

//创建一个session

//第一个参数:是否支持事务，如果为true，则会忽略第二个参数，被jms服务器设置为SESSION\_TRANSACTED

//第二个参数为false时，paramB的值可为Session.AUTO\_ACKNOWLEDGE，Session.CLIENT\_ACKNOWLEDGE，DUPS\_OK\_ACKNOWLEDGE其中一个。

//Session.AUTO\_ACKNOWLEDGE为自动确认，客户端发送和接收消息不需要做额外的工作。哪怕是接收端发生异常，也会被当作正常发送成功。

//Session.CLIENT\_ACKNOWLEDGE为客户端确认。客户端接收到消息后，必须调用javax.jms.Message的acknowledge方法。jms服务器才会当作发送成功，并删除消息。

//DUPS\_OK\_ACKNOWLEDGE允许副本的确认模式。一旦接收方应用程序的方法调用从处理消息处返回，会话对象就会确认消息的接收；而且允许重复确认。

session = connection.createSession(false, Session.AUTO\_ACKNOWLEDGE);

//创建一个到达的目的地，其实想一下就知道了，activemq不可能同时只能跑一个队列吧，这里就是连接了一个名为"text-msg"的队列，这个会话将会到这个队列，当然，如果这个队列不存在，将会被创建

destination = session.createQueue("text-msg");

//根据session，创建一个接收者对象

consumer = session.createConsumer(destination);

//实现一个消息的监听器

//实现这个监听器后，以后只要有消息，就会通过这个监听器接收到

consumer.setMessageListener(new MessageListener() {

@Override

public void onMessage(Message message) {

try {

//获取到接收的数据

String text = ((TextMessage)message).getText();

System.out.println("receed"+text);

} catch (JMSException e) {

e.printStackTrace();

}

}

});

//关闭接收端，也不会终止程序哦

// consumer.close();

} catch (JMSException e) {

e.printStackTrace();

}

}

}

Atitit 消息系统 mq 之道 attilax总结

Atitit activemq 消息系统服务使用说总结