

Acessando DLQ

Transcrição

Vimos que por padrão, o MOM possui uma fila que recebe as mensagens que não puderam ser entregues. Esta fila é uma fila como qualquer outra, e podemos acessar ao sistema de administração do ActiveMQ para verificar o seu nome:

Name ↑	Number Of Pending Messages	Number Of Consumers	Messages Enqueued	Messages Dequeued	Views	Operations
ActiveMQ.DLQ	1	0	1	0	Browse Active Consumers Active Producers atom rss	Send To Purge Delete
fila.financeiro	0	0	0	0	Browse Active Consumers Active Producers atom rss	Send To Purge Delete

Para conseguirmos consumir as mensagens desta fila, temos que adiciona-la no nosso `jndi.properties`, assim como fizemos com o a nossa `fila.financeiro`. Vamos chama-la de `DLQ`, e adiciona-la abaixo da `fila.financeiro`:

```
// jndi.properties
// código omitido

queue.financeiro = fila.financeiro
queue.DLQ = ActiveMQ.DLQ
```

Também precisamos de um Consumer para consumir nossas mensagens, então vamos criar um `ConsumidorDLQ`, muito parecido com o nosso `ConsumidorFila`, porém vamos dar um simples `System.out.println(message)` na nossa mensagem.

```
public class ConsumidorDLQ {

    @SuppressWarnings("resource")
    public static void main(String[] args) throws Exception {

        InitialContext context = new InitialContext();
        ConnectionFactory factory = (ConnectionFactory) context.lookup("ConnectionFactory");

        Connection connection = factory.createConnection();
        connection.start();
        Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);

        Destination fila = (Destination) context.lookup("DLQ");
        MessageConsumer consumer = session.createConsumer(fila);

        consumer.setMessageListener(new MessageListener() {
```

```
@Override
public void onMessage(Message message) {

    System.out.println(message);
}

});

new Scanner(System.in).nextLine();

session.close();
connection.close();
context.close();
}
}
```

Testando o nosso ConsumidorDLQ , vemos que ele está funcionando corretamente:

