JournalDev Q Simple JMS 1.1 Producer and Consumer Example With Eclipse and Embedded JBoss HornetQ Server 2 Comments Filed Under: JMS

Before going to through this post, please read my previous post at "JMS API 1.1 Producer and Consumer" to understand some baby steps to develop JMS 1.1 Producer

**Post Brief TOC** Introduction Steps to Develop a JMS V.1.1 Example

In this post, we are going to develop a Simple JMS 1.1 Producer and Consumer Example With Eclipse IDE and Embedded JBoss HornetQ Server.

Home » JMS » Simple JMS 1.1 Producer and Consumer Example With Eclipse and Embedded JBoss HornetQ Server

and Consumer programs.

application now.

Steps to Develop a JMS V.1.1 Example:

Configure one user in "hornetq-users.xml" to run this program.

xsi:schemaLocation="urn:hornetq /schema/hornetq-users.xsd">

Here we have defined one user "jduser" to use them in HornetQ Configuration XML file to configure Security.

xsi:schemaLocation="urn:hornetq /schema/hornetq-configuration.xsd">

<permission type="consume" roles="jduser"/>

<defaultuser name="jduser" password="jduser">

<role name="jduser"/>

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

<acceptor name="in-vm">

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

<configuration xmlns="urn:hornetq"</pre>

</entries> </connection-factory>

</queue> </configuration>

import javax.jms.\*;

try

understanding purpose.

■ EmbeddedJMS1.1Example ▲ B src/main/java

■ src/main/resources

x hornetq-jms.xml x hornetq-users.xml

jboss-logging-3.1.0.GA.jar b 👼 jboss-logmanager-1.2.2.GA.jar b 👼 hornetq-commons-2.3.0.BETA1.jar hornetq-journal-2.3.0.BETA1.jar

hornetq-jms-client-2.3.0.BETA1.jar hornetq-jms-server-2.3.0.BETA2.jar

netty-3.4.5.Final.jar

🌣 📴 javax.jms-api-1.1.jar

Run it as "Java Application"

Consumer Example in my coming posts.

**F**acebook

Consumer

JMS API 2.0 Producer and

Yogesh Dhavan says:

- The POM's imports are now:

Romain says:

org.hornetq hornetq-core 2.3.0.BETA1

org.hornetq

2.3.0.BETA1

org.hornetq

2.3.0.BETA2

javax.jms

jms 1.1

Name \*

**Post Comment** 

He loves Open source technologic

JournalD has become his passion.

hornetq-jms-client

hornetq-jms-server

Hello,

D 🗁 data

b 🗀 target m pom.xml

**Final Project Structure:-**

de com.jd.embedded.jms.hornetq

x hornetq-configuration.xml

▶ ■ EmbeddedHornetQJMSExample.java

<queue name="JDQueue">

<security-setting match="#">

Here We have configured JBoss HornetQ In-Memory Server and Security details.

<entry name="JDConnectionFactory"/>

Create a Simple Java class "EmbeddedHornetQJMSExample" and copy below sample code.

<entry name="/queue/JDQueue"/>

things to previous configured JBoss HornetQ In-Memory Server.

package com.jd.embedded.jms.hornetq;

public class EmbeddedHornetQJMSExample

jmsServer.start();

import org.hornetq.jms.server.embedded.EmbeddedJMS;

public static void main(final String[] args)

EmbeddedJMS jmsServer = new EmbeddedJMS();

ConnectionFactory connectionFactory =

Connection connection = null;

//1. Producer Code Start

producer.send(message);

If we observe the project in Eclipse IDE, it's final structure looks like as below:

Here we can see both Producer Sent message and Consumer Received message.

Please drop me a comment if you like my post or have any issues/suggestions.

WhatsApp

**Twitter** 

Want to do the same with Topic. Tried with same code.

thank you for this tutorial. If I may add my 2 cents:

The closing tag of hornetq-configuration.xml is incomplete

COnmfigured Topic in hornetq-jms.xml file.

System.out.println("JD: Embedded JMS Server started.");

Queue queue = (Queue)jmsServer.lookup("/queue/JDQueue");

MessageProducer producer = session.createProducer(queue);

System.out.println("Sending message: " + message.getText());

</connector>

</acceptor>

<security-settings>

</connectors> <acceptors>

</acceptors>

hornetq-jms.xml

hornetq-configuration.xml

Create a Mavenized Java project in Eclipse IDE

Project Name: EmbeddedJMS1.1Example

Final Project Structure Execute and Observe the Output

Introduction

As a Novice Developer to JMS API, it is bit tough to understand about JMS Servers, ConnectionFactory Creation, Queue Creation, Topic Creation etc. We will discuss these concepts in detail in coming posts. First of all, we should understand how to write a simple JMS Producer and Consumer programs without much effort.

To develop this example, we are going to use Eclipse IDE, JMS 1.1 API, Maven and JBoss Embedded HornetQ JMS Server. It is very easy to configure ConnectionFactory, Queue, Topic etc. with this embedded server. We just need to configure some XML files.

Use the following pom.xml file.

To use Embedded JMS Server, we don't need to download and install any software. We just need to configure some Jars in our Maven Project. Let's start developing

oject xmlns="https://maven.apache.org/POM/4.0.0" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="https://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion> <groupId>com.journaldev.jms

<artifactId>EmbeddedJMS1.1Example</artifactId> <version>1.0.0

<dependencies> <dependency> <groupId>org.hornetq <artifactId>hornetq-core</artifactId> <version>2.3.0.BETA1 </dependency>

<dependency> <groupId>org.hornetq <artifactId>hornetq-jms-client</artifactId>

<version>2.3.0.BETA1 </dependency>

<dependency> <groupId>org.hornetq

<artifactId>hornetq-jms-server</artifactId> <version>2.3.0.BETA1 </dependency>

<dependency> <groupId>javax.jms</groupId> <artifactId>javax.jms-api</artifactId>

<version>1.1</version>

**NOTE:-**This pom.xml contains 3 dependencies

hornetq-core-x.x.jar:- It contains to provide base API for both JMS server and client programs.

hornetq-jms-server-x.x.jar:- It contains API to provide Embedded JBoss HornetQ Server.

hornetq-core-x.x.jar:- It contains JMS API implementation for JMS Session, Queue, Topic etc.

hornetq-users.xml

<?xml version="1.0" encoding="UTF-8"?> <configuration xmlns="urn:hornetq"</pre> xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"

</defaultuser> </configuration>

<persistence-enabled>false</persistence-enabled> <connectors> <connector name="in-vm"> <factory-class>org.hornetq.core.remoting.impl.invm.InVMConnectorFactory</factory-class>

<factory-class>org.hornetq.core.remoting.impl.invm.InVMAcceptorFactory</factory-class>

<configuration xmlns="urn:hornetq" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"</pre>

Configure in-memory server configuration "hornetq-configuration.xml" to run this program. It is used to configure In-Memory JMS Server.

<permission type="send" roles="jduser"/> </security-setting> </security-settings> </config

Configure in-memory server ConnectionFactory and Queue "hornetq-jms.xml" to run this program. It is used to configure JMS Settings.

xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:hornetq /schema/hornetq-jms.xsd"> <connection-factory name="JDConnectionFactory"> <connectors> <connector-ref connector-name="in-vm"/> </connectors> <entries>

Here we have configured ConnectionFactory with "JDConnectionFactory" name and Queue destination with "/queue/JDQueue" JNDI name. We have linked these two

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

TextMessage message = session.createTextMessage("Hello, I'm from JournalDev(JD).");

Just for simplicity, I've created both Message Producer and Consumer in a single Java file. Please go through those two sub-sections(separated by comments) for

// This EmbeddedJMS class acts as a JMSServer for example: JBoss HornetQ Server

(ConnectionFactory)jmsServer.lookup("JDConnectionFactory");

src/test/java src/test/resources ⇒ Mark System Library [JavaSE-1.7] Maven Dependencies hornetq-core-2.3.0.BETA1.jar b 📴 jgroups-3.0.10.Final.jar

**Execute and Observe the Output:-**In this section, we will run our JMS Program and observe the results in Eclipse IDE Console. Right click on "EmbeddedHornetQJMSExample.java" Program

Observe the output in Eclipse IDE Console Markers 🗀 Properties 🙉 Servers 🙀 Data Source Explorer 🔝 Snippets 🔘 Console 🗯 cteminated EmbeddedHometQMSExample (2) [Java Application E Useal To G3hbrigavaw.ess (Nov 28, 2015, 912:30 PM)

Nov 28, 2015 9:12:34 PM org. hornetq.core.server.impl.HometQServerEmplScharedNothingLiveActivation run

INFO: HQ11899: Server is now live

Nov 28, 2015 9:12:34 PM org. hornetq.core.server.impl.HometQServerEmplScharedNothingLiveActivation run

INFO: HQ11899: Server is now live

Nov 28, 2015 9:12:34 PM org. hornetq.core.server.impl.HometQServerEmpl start

INFO: HQ11893: HornetQ Server version 2.3.0.85TA1 (MornetQ sting, 122) [7713b2d6-95ef-1le5-b46d-99168f195eb1]

JD: Embedded JMS Server started.]

Sending message: Hello, I'm from JournalDev(JD).

Received message: Hello, I'm from JournalDev(JD).

Received message: Hello, I'm from JournalDev(JD).

INFO: HQ11894: NormetQ Server version 2.3.0.85TA1 (MornetQServerEmpl stop

INFO: HQ11894: NormetQ Server version 2.3.0.85TA1 (MornetQ sting, 122) [7713b2d6-95ef-1le5-b46d-99168f185eb1] stopped

JD: Embedded JMS Server stopped.

That's it all about developing Simple JMS API 1.1 Producer and Consumer application with Embedded JBoss HornetQ Server. We will discuss JMS API 2.0 Producer and

🕳 Reddit

in LinkedIn

Email

JMS 1.1 Producer and

IBoss 6.0 AS

Consumer Example With Eclipse IDE, EJB Project and

December 11, 2019 at 12:44 am

August 14, 2018 at 10:02 am

Reply

NEXT ... PREV Comments

Thank you again for your help in my understanding of HornetQ Reply Leave a Reply Your email address will not be published. Required fields are marked \* Comment

Email \*

JournalDev Most Popular \_\_\_\_ Favorite Sites \_\_\_\_ JournalDev is one of the most popular websites for Java, Java / Java EE Tutorials AskPython Python, Android, and related technical articles. Our CodeForGeek tutorials are regularly updated, error-free, and complete. Core Java Tutorial Every month millions of developers like you visit GoLangDocs Python Tutorials JournalDev to read our tutorials. LinuxForDevices Java Interview Questions JournalDev Was founded by Pankaj Kumar in Core Java Interview Questions ience and learnings with the Mkyong

Java Design Patterns

ditions · Contact Us · About · Part of JournalDev IT Services Private Limited

Spring Tutorial

VM-Help

WP-Design