**Java Messaging Service (JMS):**

* JMS is an API that provides the facility to **create**, **send** and **read** messages.
* JMS is mainly used to send and receive messages fro one application to another.
* It provides loosely coupled, reliable and asynchronous communication.

**Messaging** is a ***technique to communicate application*** or software components.

**Requirement of JMS:**

Generally, user send messages to application. But if we want to send message from one application to another, we need to use JMS API.

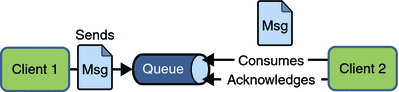
Eg. Consider a scenario, one application A is running in UK and another application B is running in INDIA. To send message from application A to B we need to use JMS.

**Types of messaging domains in JMS:**

**1.Point to Point messaging domain:**

In PTP model, one message is delivered to one receiver only, Queue is used as message -oriented middleware.

The Queue is responsible to hold the message to until receiver is ready. In PTP model, there is no timing dependency between sender and receiver.

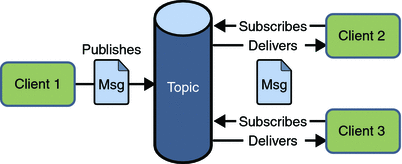


**2.Publisher/Subscriber messaging domain:**

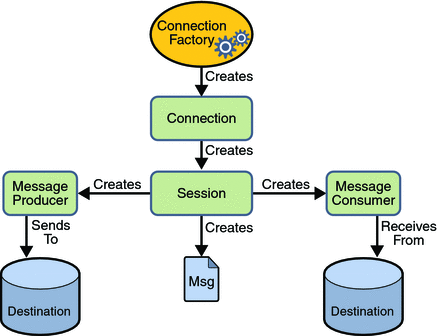
In Pub/Sub model, one message is delivered to all the subscribers. It is like broadcasting. Here Topic is used as a message-oriented middleware.

The Topic is responsible to hold and deliver messages.

In Pub/Sub there is timing dependency between publisher and subscriber.



**JMS Programming model:**



**JMS Queue Example:**

To develop JMS queue example, you need to install any application server. Here, we are using glassfish3 server where we are creating two JNDI.

Create connection factory named myQueueConnectionFactory

Create destination resource named myQueue

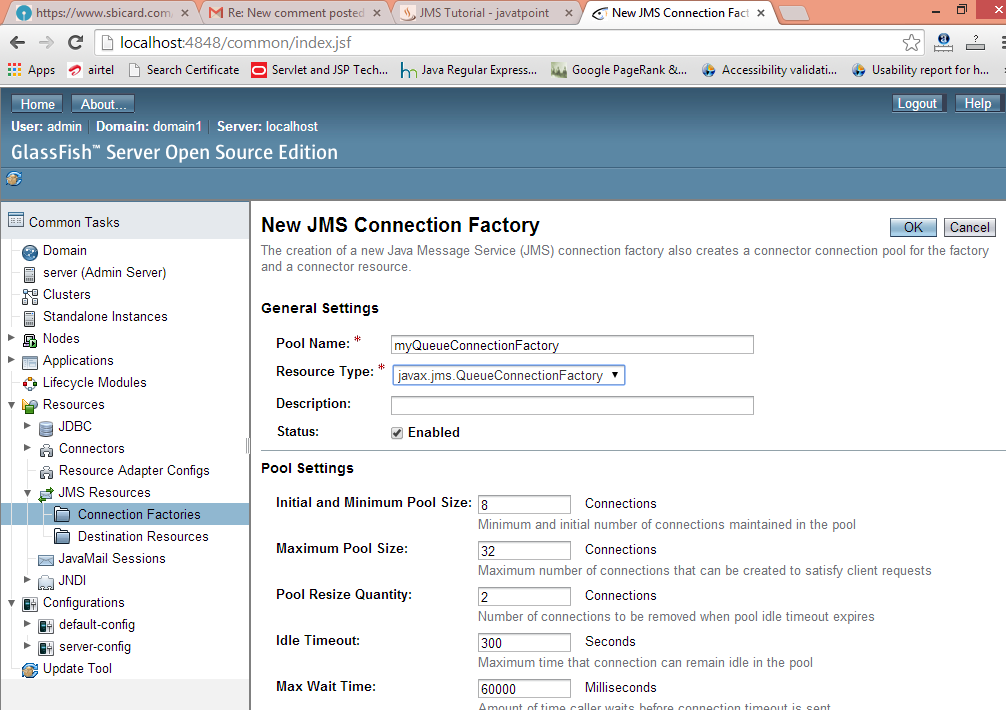
After creating JNDI, create server and receiver application. You need to run server and receiver in different console. Here, we are using eclipse IDE, it is opened in different console by default.

1) Create connection factory and destination resource

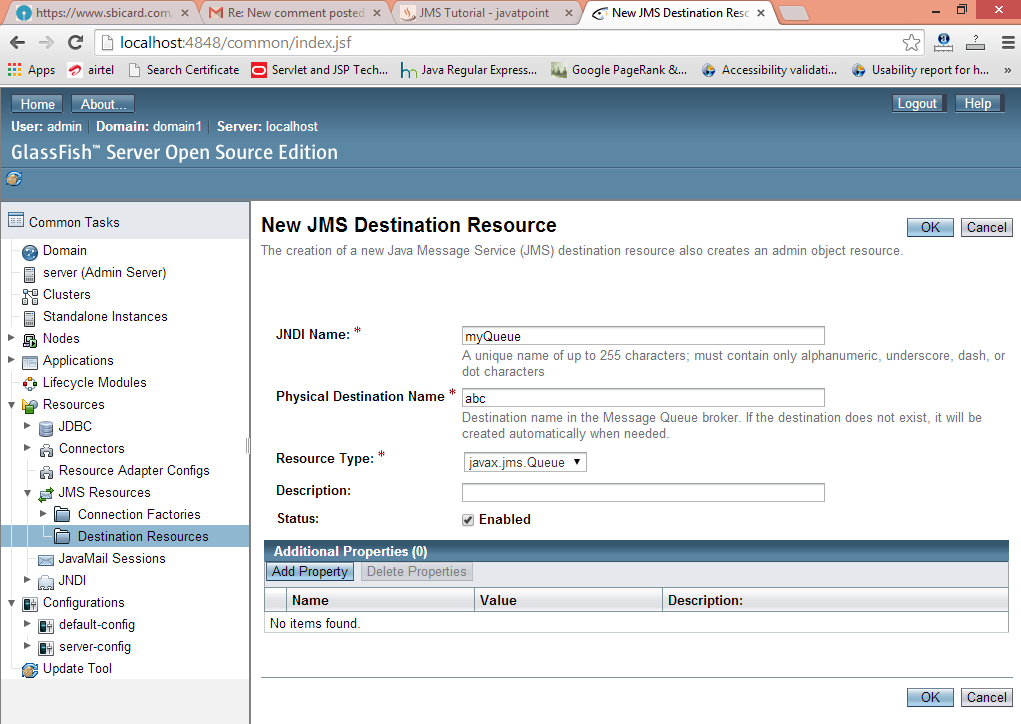
Open server admin console by the URL http://localhost:4848

Login with the username and password.

Click on the JMS Resource -> Connection Factories -> New, now write the pool name and select the Resource Type as QueueConnectionFactory then click on ok button.



Click on the JMS Resource -> Destination Resources -> New, now write the JNDI name and physical destination name then click on ok button.



**2) Create sender and receiver application**

Let's see the Sender and Receiver code. Note that Receiver is attached with listener which will be invoked when user sends message.

File: MySender.java

import java.io.BufferedReader;

import java.io.InputStreamReader;

import javax.naming.\*;

import javax.jms.\*;

public class MySender {

    public static void main(String[] args) {

        try

        {   //Create and start connection

            InitialContext ctx=new InitialContext();

            QueueConnectionFactory f=(QueueConnectionFactory)ctx.lookup("myQueueConnectionFactory");

            QueueConnection con=f.createQueueConnection();

            con.start();

            //2) create queue session

            QueueSession ses=con.createQueueSession(false, Session.AUTO\_ACKNOWLEDGE);

            //3) get the Queue object

            Queue t=(Queue)ctx.lookup("myQueue");

            //4)create QueueSender object

            QueueSender sender=ses.createSender(t);

            //5) create TextMessage object

            TextMessage msg=ses.createTextMessage();

            //6) write message

            BufferedReader b=new BufferedReader(new InputStreamReader(System.in));

            while(true)

            {

                System.out.println("Enter Msg, end to terminate:");

                String s=b.readLine();

                if (s.equals("end"))

                    break;

                msg.setText(s);

                //7) send message

                sender.send(msg);

                System.out.println("Message successfully sent.");

            }

            //8) connection close

            con.close();

        }catch(Exception e){System.out.println(e);}

    }

}

File: MyReceiver.java

import javax.jms.\*;

import javax.naming.InitialContext;

public class MyReceiver {

    public static void main(String[] args) {

        try{

            //1) Create and start connection

            InitialContext ctx=new InitialContext();

            QueueConnectionFactory f=(QueueConnectionFactory)ctx.lookup("myQueueConnectionFactory");

            QueueConnection con=f.createQueueConnection();

            con.start();

            //2) create Queue session

            QueueSession ses=con.createQueueSession(false, Session.AUTO\_ACKNOWLEDGE);

            //3) get the Queue object

            Queue t=(Queue)ctx.lookup("myQueue");

            //4)create QueueReceiver

            QueueReceiver receiver=ses.createReceiver(t);

            //5) create listener object

            MyListener listener=new MyListener();

            //6) register the listener object with receiver

            receiver.setMessageListener(listener);

            System.out.println("Receiver1 is ready, waiting for messages...");

            System.out.println("press Ctrl+c to shutdown...");

            while(true){

                Thread.sleep(1000);

            }

        }catch(Exception e){System.out.println(e);}

    }

}

File: MyListener.java

import javax.jms.\*;

public class MyListener implements MessageListener {

    public void onMessage(Message m) {

        try{

        TextMessage msg=(TextMessage)m;

        System.out.println("following message is received:"+msg.getText());

        }catch(JMSException e){System.out.println(e);}

    }

}

Run the Receiver class first then Sender class.

**JMS Topic Example**

It is same as JMS Queue, but you need to change Queue to Topic, Sender to Publisher and Receiver to Subscriber.

You need to create 2 JNDI named myTopicConnectionFactory and myTopic.

File: MySender.java

import java.io.BufferedReader;

import java.io.InputStreamReader;

import javax.naming.\*;

import javax.jms.\*;

public class MySender {

    public static void main(String[] args) {

        try

        {   //Create and start connection

            InitialContext ctx=new InitialContext();

            TopicConnectionFactory f=(TopicConnectionFactory)ctx.lookup("myTopicConnectionFactory");

            TopicConnection con=f.createTopicConnection();

            con.start();

            //2) create queue session

            TopicSession ses=con.createTopicSession(false, Session.AUTO\_ACKNOWLEDGE);

            //3) get the Topic object

            Topic t=(Topic)ctx.lookup("myTopic");

            //4)create TopicPublisher object

            TopicPublisher publisher=ses.createPublisher(t);

            //5) create TextMessage object

            TextMessage msg=ses.createTextMessage();

            //6) write message

            BufferedReader b=new BufferedReader(new InputStreamReader(System.in));

            while(true)

            {

                System.out.println("Enter Msg, end to terminate:");

                String s=b.readLine();

                if (s.equals("end"))

                    break;

                msg.setText(s);

                //7) send message

                publisher.publish(msg);

                System.out.println("Message successfully sent.");

            }

            //8) connection close

            con.close();

        }catch(Exception e){System.out.println(e);}

    }

}

File: MyReceiver.java

import javax.jms.\*;

import javax.naming.InitialContext;

public class MyReceiver {

    public static void main(String[] args) {

        try {

            //1) Create and start connection

            InitialContext ctx=new InitialContext();

            TopicConnectionFactory f=(TopicConnectionFactory)ctx.lookup("myTopicConnectionFactory");

            TopicConnection con=f.createTopicConnection();

            con.start();

            //2) create topic session

            TopicSession ses=con.createTopicSession(false, Session.AUTO\_ACKNOWLEDGE);

            //3) get the Topic object

            Topic t=(Topic)ctx.lookup("myTopic");

            //4)create TopicSubscriber

            TopicSubscriber receiver=ses.createSubscriber(t);

            //5) create listener object

            MyListener listener=new MyListener();

            //6) register the listener object with subscriber

            receiver.setMessageListener(listener);

            System.out.println("Subscriber1 is ready, waiting for messages...");

            System.out.println("press Ctrl+c to shutdown...");

            while(true){

                Thread.sleep(1000);

            }

        }catch(Exception e){System.out.println(e);}

    }

}

File: MyListener.java

import javax.jms.\*;

public class MyListener implements MessageListener {

    public void onMessage(Message m) {

        try{

        TextMessage msg=(TextMessage)m;

        System.out.println("following message is received:"+msg.getText());

        }catch(JMSException e){System.out.println(e);}

    }

}