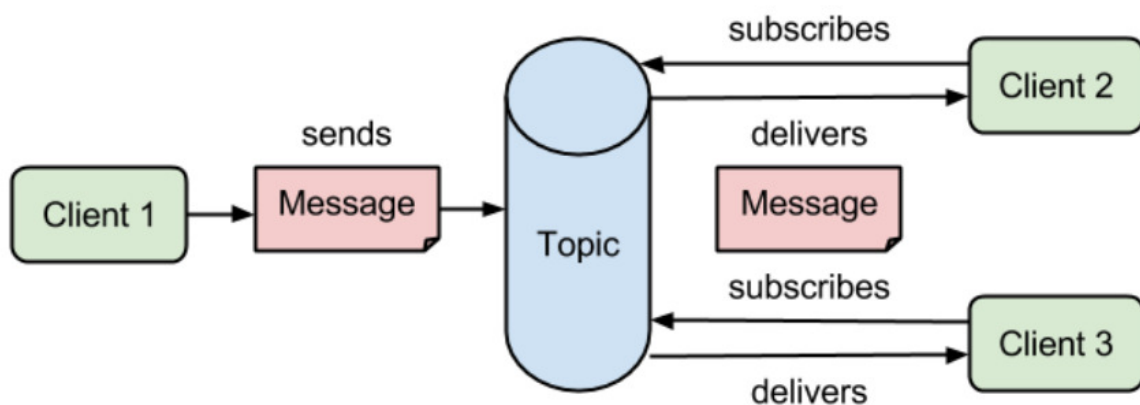


## Java Message Service (pub/sub)

In a publish/subscribe (pub/sub) product or application, clients address messages to a topic, which functions somewhat like a bulletin board. Subscribers can receive information, in the form of messages, from publishers. Topics retain messages only as long as it takes to distribute them to current subscribers.

### **Publish/Subscribe Messaging**



Pub/Sub messaging has the following characteristics:

- Each message can have **multiple consumers**.
- Publishers and subscribers have a **timing dependency**. A client that subscribes to a topic can consume only messages published after the client has created a subscription, and the subscriber must continue to be active in order for it to consume messages.

The JMS API **overcome** this timing dependency mentioned in the second point to some extent by allowing subscribers to create durable subscriptions, which receive messages sent while the subscribers are not active. **Durable subscriptions** provide the flexibility and reliability of queues but still allow clients to send messages to many recipients.

## ActiveMQ Example with Topic (Non Durable Subscription)

```
import javax.jms.Connection;
import javax.jms.ConnectionFactory;
import javax.jms.JMSEException;
import javax.jms.MessageProducer;
import javax.jms.Session;
import javax.jms.TextMessage;
import javax.jms.Topic;

import org.apache.activemq.ActiveMQConnectionFactory;

public class TopicProducer {

    Connection connection;
    Session session;
    MessageProducer messageProducer;
    String clientId;
    String message;
    String subject;
    public void getConnection(String clientId,String topicName)
                                throws JMSEException {

        this.clientId = clientId;
        this.subject = topicName;
        /*
         * Getting JMS connection from the server and starting it
         */
        ConnectionFactory connectionFactory =
            new ActiveMQConnectionFactory("tcp://localhost:61616");
        connection = connectionFactory.createConnection();
        // connection.setClientID(clientId);

        session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
        Topic topic = session.createTopic(topicName);

        messageProducer = session.createProducer(topic);
    }

    public void sendMessage(String message) throws JMSEException {

        this.message = message;
        TextMessage textMessage = session.createTextMessage(message);
        messageProducer.send(textMessage);
        System.out.println("Message Sent: "+message);
    }

    public void closeConnection() throws JMSEException {
        connection.close();
    }
}
```

## Subscriber:

```
import javax.jms.Connection;
import javax.jms.ConnectionFactory;
import javax.jms.JMSEException;
import javax.jms.Message;
import javax.jms.MessageConsumer;
import javax.jms.Session;
import javax.jms.TextMessage;
import javax.jms.Topic;

import org.apache.activemq.ActiveMQConnectionFactory;

public class TopicSubscriber_NonDurable {

    Connection connection;
    Session session;
    MessageConsumer messageConsumer;
    Topic topic;

    public void getConnection(String clientId,String subject)
                                throws JMSEException{

        /*
         * Getting JMS connection from the server and starting it
         */
        ConnectionFactory connectionFactory =
            new ActiveMQConnectionFactory("tcp://localhost:61616");
        connection = connectionFactory.createConnection();
//        connection.setClientID(clientId);
        session = connection
            .createSession(false, Session.AUTO_ACKNOWLEDGE);
        topic = session.createTopic(subject);
        messageConsumer = session.createConsumer(topic);
        connection.start();
    }

    public void getMessage() throws JMSEException {

        Message message = messageConsumer.receive(1000);

        TextMessage textMessage = (TextMessage) message;

        if(textMessage != null) {
            String msg = textMessage.getText();
            System.out.println("Message Recieved: "+msg);
        }else {
            System.out.println("Message Not Recieved...");
        }
    }

    public void closeConnection() throws JMSEException {
        connection.close();
    }

}
```

## Test:

```
import javax.jms.JMSEException;

public class TestNonDurable {
    public static void main(String[] args) throws JMSEException{

        String clientProducer = "1234";
        String msg = "Hello World";
        String topicName = "testTopic";

        TopicProducer producer = new TopicProducer();
        producer.getConnection(clientProducer, topicName);

        TopicSubscriber_NonDurable consumer1 =
            new TopicSubscriber_NonDurable();
        consumer1.getConnection("abcd1", topicName);

        TopicSubscriber_NonDurable consumer2 =
            new TopicSubscriber_NonDurable();
        consumer2.getConnection("abcd2", topicName);

        TopicSubscriber_NonDurable consumer3 =
            new TopicSubscriber_NonDurable();
        consumer3.getConnection("abcd3", topicName);

        producer.sendMessage(msg);
        consumer1.getMessage();
        consumer2.getMessage();

        consumer3.getMessage();
        consumer3.closeConnection();

        producer.sendMessage("Hello World 2");
        consumer1.getMessage();
        consumer2.getMessage();

        consumer3 = new TopicSubscriber_NonDurable();
        consumer3.getConnection("abcd3", topicName);
        consumer3.getMessage();

        producer.closeConnection();
        consumer1.closeConnection();
        consumer2.closeConnection();
        consumer3.closeConnection();
    }
}
```

# Durable Subscription

As mentioned in the beginning of this post it is also possible to create a durable subscription which allows to receive messages sent while the subscribers are not active. The JMS specification dictates that the identification of a specific durable subscription is done by a combination of the “*client ID*”, the “*durable subscription name*” and the “*topic name*”.

- The ***DurableSubscriber*** has three main differences with the previous ***Subscriber*** class:
  - 1- A *clientId* is mandatory on the connection in order to allow a JMS provider to uniquely identify a durable subscriber.
  - 2- A durable subscriber is created using *Session.createDurableSubscriber*.
  - 3- A *subscriptionName* is needed when creating the durable subscriber.

```
import org.apache.activemq.ActiveMQConnectionFactory;

public class DurableSubscriber {
    private String clientId;
    private Connection connection;
    private Session session;
    private MessageConsumer messageConsumer;

    private String subscriptionName;

    public void create(String clientId, String topicName,
                      String subscriptionName) throws JMSEException {
        this.clientId = clientId;
        this.subscriptionName = subscriptionName;

        ConnectionFactory connectionFactory =
            new ActiveMQConnectionFactory("tcp://localhost:61616");

        connection = connectionFactory.createConnection();

        connection.setClientID(this.clientId);

        session = connection
            .createSession(false, Session.AUTO_ACKNOWLEDGE);

        Topic topic = session.createTopic(topicName);

        messageConsumer = session
            .createDurableSubscriber(topic, subscriptionName);
    }
}
```

```

        connection.start();
    }

    public void removeDurableSubscriber() throws JMSEException {
        messageConsumer.close();
        session.unsubscribe(subscriptionName);
    }

    public void closeConnection() throws JMSEException {
        connection.close();
    }

    public void getMessage(int timeout) throws JMSEException {
        Message message = messageConsumer.receive(timeout);

        if(message != null) {
            TextMessage textMessage = (TextMessage) message;
            String text = textMessage.getText();
            System.out.println("Message Recieved: "+text);
        }else {
            System.out.println("Message not recieved");
        }
    }
}

```

## Test:

```

import javax.jms.JMSEException;

import com.test.apcahe.activemq.topic.nondurable.TopicProducer;

public class TestDurable {
    public static void main(String[] args) throws JMSEException{

        String clientProducer = "1234";
        String msg = "Hello World";
        String topicName = "durableTopic";

        TopicProducer producer = new TopicProducer();
        producer.getConnection(clientProducer, topicName);

        DurableSubscriber durableSubscriber = new DurableSubscriber();
        durableSubscriber.create("durableClient1", "durableTopic"
                                , "durableSubscription");
        durableSubscriber.closeConnection();

        producer.sendMessage(msg);

        durableSubscriber.create("durableClient1", "durableTopic"
                                , "durableSubscription");

        producer.sendMessage("Hello World 2");

        durableSubscriber.getMessage(1000);
        durableSubscriber.getMessage(1000);
    }
}

```

```
    durableSubscriber.removeDurableSubscriber();  
    durableSubscriber.closeConnection();  
    producer.closeConnection();
```

```
    }  
}
```