

Assignment 1 - Messaging In Action

```
public class Passenger implements Serializable {  
    private static final long serialVersionUID = 1L;  
    private int id;  
    private String firstName;  
    private String lastName;  
    private String email;  
    private Long phoneNo;  
    public Passenger() {  
    }  
    public Passenger(int id, String firstName, String lastName, String  
email, Long phoneNo) {  
        super();  
        this.id = id;  
        this.firstName = firstName;  
        this.lastName = lastName;  
        this.email = email;  
        this.phoneNo = phoneNo;  
    }  
    public int getId() {  
        return id;  
    }  
    public void setId(int id) {  
        this.id = id;  
    }  
    public String getFirstName() {
```

```

        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public Long getPhoneNo() {
        return phoneNo;
    }

    public void setPhoneNo(Long phoneNo) {
        this.phoneNo = phoneNo;
    }

    @Override
    public String toString() {
        return "Passenger [id=" + id + ", firstName=" + firstName + ", lastName="
            + lastName + ", email=" + email

```

```
+ ", phoneNo=" + phoneNo + "];"  
}  
}
```

```
public class FlightCheckInApp {  
  
    public static void main(String[] args) throws Exception {  
  
        InitialContext ctx = new InitialContext();  
  
        Queue passengerRequestQueue = (Queue)  
        ctx.lookup("queue/passengerRequestQueue");  
  
        //Queue passengerReplyQueue = (Queue)  
        ctx.lookup("queue/passengerReplyQueue");  
  
        try(ActiveMQConnectionFactory cf = new ActiveMQConnectionFactory();  
  
            JMSContext jmsContext = cf.createContext()) {  
  
            JMSProducer producer = jmsContext.createProducer();  
  
            TemporaryQueue temporaryQueue = jmsContext.createTemporaryQueue();  
  
            ObjectMessage objectMessage = jmsContext.createObjectMessage();  
  
            Passenger passenger = new Passenger();  
  
            passenger.setId(1);  
  
            passenger.setFirstName("Jane");  
  
            passenger.setLastName("Dark");  
  
            passenger.setEmail("janedark555@aol.com");  
  
            passenger.setPhoneNo(44444444444L);  
  
            objectMessage.setObject(passenger);  
  
            objectMessage.setJMSReplyTo(temporaryQueue);  
  
            producer.send(passengerRequestQueue, passenger);  
  
            System.out.println("Successfully published message to Reservation  
systemApp");  
        }  
    }  
}
```

```

System.out.println("FlightCheckInApp :: MessageID :: " +
objectMessage.getJMSMessageID());

// consume from reply queue

JMSConsumer consumer =
jmsContext.createConsumer(objectMessage.getJMSReplyTo());

MapMessage mapMessage = (MapMessage) consumer.receive(25000);

System.out.println("Is passenger ticker reserved: " +
mapMessage.getBoolean("isReserved"));

// load balancing

for(int i=0; i<=10; i++) {

producer.send(passengerRequestQueue, passenger);

}

}

}

}

public class ReservationSystemApp {

public static void main(String[] args) throws Exception {

InitialContext ctx = new InitialContext();

Queue passengerRequestQueue = (Queue)
ctx.lookup("queue/passengerRequestQueue");

try(ActiveMQConnectionFactory cf = new ActiveMQConnectionFactory();

JMSContext jmsContext = cf.createContext()) {

JMSConsumer consumer = jmsContext.createConsumer(passengerRequestQueue);

consumer.setMessageListener(new ReservationSystemListener());

Thread.sleep(15000);

// load balancing

JMSConsumer consumer1 = jmsContext.createConsumer(passengerRequestQueue);

```

```

JMSConsumer consumer2 = jmsContext.createConsumer(passengerRequestQueue);

for(int i=1; i<=10; i+=2) {

System.out.println("Consumer 1 : " + consumer1.receive());

System.out.println("Consumer 1 : " + consumer2.receive());

}

}

}

}

```

```

public class ReservationSystemListener implements MessageListener {

public void onMessage(Message message) {

if (message instanceof ObjectMessage) {

ObjectMessage objMsg = (ObjectMessage) message;

try(ActiveMQConnectionFactory cf = new ActiveMQConnectionFactory();

JMSContext jmsContext = cf.createContext()) {

InitialContext ctx = new InitialContext();

//Queue passengerReplyQueue = (Queue)
ctx.lookup("queue/passengerReplyQueue");

MapMessage mapMessage = jmsContext.createMapMessage();

Passenger passenger = (Passenger) objMsg.getObject();

System.out.println("ReservationSystemListener :: " + passenger);

if (passenger != null) {

mapMessage.setBoolean("isReserved", true);

} else {

mapMessage.setBoolean("isReserved", false);

}

mapMessage.setJMSCorrelationID(objMsg.getJMSMessageID());

```

```

JMSProducer producer = jmsContext.createProducer();
producer.send(objMsg.getJMSReplyTo(), mapMessage);
} catch (JMSException e) {
e.printStackTrace();
} catch (NamingException e) {
e.printStackTrace();
}
}
}
}
}

```

Assignment 2 -PUB SUB IN ACTION

CardApp:

```

package com.bharath.jms.card;

import javax.jms.JMSContext;
import javax.jms.Topic;
import javax.naming.InitialContext;
import javax.naming.NamingException;

import org.apache.activemq.artemis.jms.client.ActiveMQConnectionFactory;

```

```

import com.bharath.jms.hr.Employee;

public class CardApp {

    public static void main(String[] args) throws NamingException{
        InitialContext context = new InitialContext();
        Topic topic = (Topic)context.lookup("topic/empTopic");
        try(ActiveMQConnectionFactory cf = new ActiveMQConnectionFactory();
        JMSContext jmsContext = cf.createContext() ) {
            Card card = new Card();
            for(int i=1; i<=10; i++) {
                card.setId(100+i);
                card.setCardHolder("Murat Hildering" + i);
                card.setCardNumber("01236753" + i);
                card.setExpireMonth(i);
                card.setExpireYear(2020+i);
                card.setCvc("12"+i);
                jmsContext.createProducer().send(topic, card);
            }
            System.out.println("Message sent");
        }
    }
}

```

----- Card.java -----

```

package com.bharath.jms.card;

import java.io.Serializable;

```

```
public class Card implements Serializable {

    private static final long serialVersionUID = 2668802524807629044L;

    int id;

    String cardHolder;

    String cardNumber;

    int expireMonth;

    int expireYear;

    String cvc;

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    public String getCardHolder() {

        return cardHolder;

    }

    public void setCardHolder(String cardHolder) {

        this.cardHolder = cardHolder;

    }

    public String getCardNumber() {

        return cardNumber;

    }

}
```



```
public void setCardNumber(String cardNumber) {  
    this.cardNumber = cardNumber;  
}  
  
public int getExpireMonth() {  
    return expireMonth;  
}  
  
public void setExpireMonth(int expireMonth) {  
    this.expireMonth = expireMonth;  
}  
  
public int getExpireYear() {  
    return expireYear;  
}  
  
public void setExpireYear(int expireYear) {  
    this.expireYear = expireYear;  
}  
  
public String getCvc() {  
    return cvc;  
}  
  
public void setCvc(String cvc) {  
    this.cvc = cvc;  
}  
  
@Override  
public String toString() {  
    return "Card [cardHolder=" + cardHolder + ", cardNumber=" + cardNumber +  
    ", expireMonth=" + expireMonth  
    + ", expireYear=" + expireYear + ", cvc=" + cvc + "];"  
}  
}
```

SecurityApp

```
package com.bharath.jms.card;

import javax.jms.JMSConsumer;
import javax.jms.JMSContext;
import javax.jms.JMSException;
import javax.jms.Message;
import javax.jms.Topic;
import javax.naming.InitialContext;
import javax.naming.NamingException;

import org.apache.activemq.artemis.jms.client.ActiveMQConnectionFactory;

import com.bharath.jms.hr.Employee;

public class SecurityApp {

    public static void main(String[] args) throws NamingException,
        JMSException, InterruptedException {

        InitialContext context = new InitialContext();

        Topic topic = (Topic)context.lookup("topic/empTopic");

        try(ActiveMQConnectionFactory cf = new ActiveMQConnectionFactory();

        JMSContext jmsContext = cf.createContext() ) {
```

```

jmsContext.setClientID("securityApp");

JMSConsumer consumer = jmsContext.createDurableConsumer(topic,
"subscription1");

consumer.close();

Thread.sleep(10000);

consumer = jmsContext.createDurableConsumer(topic, "subscription1");

Message message = consumer.receive();

Card card = message.getBody(Card.class);

System.out.println("Security got: " + card);

consumer.close();

jmsContext.unsubscribe("subscription1");

}

}

}

```

AlertsApp

```

package com.bharath.jms.card;

import javax.jms.JMSConsumer;
import javax.jms.JMSContext;
import javax.jms.JMSException;
import javax.jms.Message;
import javax.jms.Topic;
import javax.naming.InitialContext;
import javax.naming.NamingException;

```

```
import org.apache.activemq.artemis.jms.client.ActiveMQConnectionFactory;

import com.bharath.jms.hr.Employee;

public class AlertsAp {

    public static void main(String[] args) throws NamingException,
    JMSException {

        InitialContext context = new InitialContext();

        Topic topic = (Topic)context.lookup("topic/empTopic");

        try(ActiveMQConnectionFactory cf = new ActiveMQConnectionFactory();

        JMSContext jmsContext = cf.createContext() ) {

            JMSConsumer consumer = jmsContext.createSharedConsumer(topic,
            "sharedConsumer");

            JMSConsumer consumer2 = jmsContext.createSharedConsumer(topic,
            "sharedConsumer");

            for(int i=1; i <= 10; i+=2) {

                Message message = consumer.receive();

                Card card1 = message.getBody(Card.class);

                System.out.println("Alerts consumer1 got: " + card1);

                Message message2 = consumer2.receive();

                Card card2 = message2.getBody(Card.class);

                System.out.println("Alerts consumer2 got: " + card2);

            }

        }

    }

}
```

Assignment 3 - Filters

```
package com.bharath.jms.claimmanagemment;

import javax.jms.JMSConsumer;
import javax.jms.JMSContext;
import javax.jms.JMSException;
import javax.jms.JMSProducer;
import javax.jms.ObjectMessage;
import javax.jms.Queue;
import javax.naming.InitialContext;
import javax.naming.NamingException;

import org.apache.activemq.artemis.jms.client.ActiveMQConnectionFactory;

public class ClaimManagement {

    public static void main(String[] args) throws NamingException,
        JMSException {

        InitialContext initialContext = new InitialContext();

        Queue claimQueue = (Queue) initialContext.lookup("queue/claimQueue");

        try(ActiveMQConnectionFactory cf = new ActiveMQConnectionFactory();

        JMSContext jmsContext = cf.createContext() ) {

            JMSProducer producer = jmsContext.createProducer();

            //JMSConsumer consumer = jmsContext.createConsumer(claimQueue, "doctorType
            in ('psych','neuro') OR JMSPriority BETWEEN 5 and 9");
```

```

JMSConsumer consumer = jmsContext.createConsumer(claimQueue,

"insuranceProvider in ('blue cross','american united') AND doctorName LIKE
'H%' AND claimAmount % 10 = 0");

ObjectMessage objectMessage = jmsContext.createObjectMessage();

//objectMessage.setIntProperty("hospitalId", 1);

objectMessage.setDoubleProperty("claimAmount", 1010);

objectMessage.setStringProperty("doctorName", "Holcer");

objectMessage.setStringProperty("insuranceProvider", "blue cross");

//bjectMessage.setStringProperty("doctorType", "gyna");

Claim claim = new Claim();

claim.setHospitalId(1);

claim.setClaimAmount(1000);

claim.setDoctorName("Hanna");

claim.setDoctorType("gyna");

claim.setInsuranceProvider("blue cross");

objectMessage.setObject(claim);

producer.send(claimQueue, objectMessage);

Claim receiveBody = consumer.receiveBody(Claim.class);

System.out.println(receiveBody.getClaimAmount());

}

}

}

```

```
package com.bharath.jms.guaranteedmessaging;

import java.io.Serializable;
import java.util.Date;

public class FlightCheckin implements Serializable {

    private static final long serialVersionUID = -5064692268410211449L;

    private int id;

    private String flightRoute;

    private String passengerName;

    private String passengerDocumentType;

    private String passengerDocumentNumber;

    private Date checkinTime;

    public FlightCheckin() {
        this.checkinTime = new Date();
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getFlightRoute() {
        return flightRoute;
    }

    public void setFlightRoute(String flightRoute) {
```

```

this.flightRoute = flightRoute;
}

public String getPassengerName() {
return passengerName;
}

public void setPassengerName(String passengerName) {
this.passengerName = passengerName;
}

public String getPassengerDocumentType() {
return passengerDocumentType;
}

public void setPassengerDocumentType(String passengerDocumentType) {
this.passengerDocumentType = passengerDocumentType;
}

public String getPassengerDocumentNumber() {
return passengerDocumentNumber;
}

public void setPassengerDocumentNumber(String passengerDocumentNumber) {
this.passengerDocumentNumber = passengerDocumentNumber;
}

public Date getCheckinTime() {
return checkinTime;
}

@Override

public String toString() {
return "FlightCheckin [id=" + id + ", flightRoute=" + flightRoute + ",
passengerName=" + passengerName

```



```

+ ", passengerDocumentType=" + passengerDocumentType + ",
passengerDocumentNumber="

+ passengerDocumentNumber + ", checkinTime=" + checkinTime + "]];

}

}

```

```

-----

```

```

package com.bharath.jms.guaranteedmessaging;

```

```

import javax.jms.JMSContext;

```

```

import javax.jms.JMSException;

```

```

import javax.jms.JMSProducer;

```

```

import javax.jms.ObjectMessage;

```

```

import javax.jms.Queue;

```

```

import javax.naming.InitialContext;

```

```

import javax.naming.NamingException;

```

```

import org.apache.activemq.artemis.jms.client.ActiveMQConnectionFactory;

```

```

public class CheckinApp {

```

```

    public static void main(String[] args) throws NamingException {

```

```

        InitialContext initialContext = new InitialContext();

```

```

        Queue requestQueue = (Queue) initialContext.lookup("queue/requestQueue");

```

```

        try(ActiveMQConnectionFactory cf = new ActiveMQConnectionFactory();

```

```

            JMSContext jmsContext = cf.createContext(JMSContext.SESSION_TRANSACTED)) {

```

```

                JMSProducer producer = jmsContext.createProducer();

```

```

                FlightCheckin checkin1 = new FlightCheckin();

```

```

                checkin1.setId(12);

```

```
checkin1.setFlightRoute("LHA-VCR");  
checkin1.setPassengerDocumentType("Canadian Passport");  
checkin1.setPassengerDocumentNumber("56756C456GH9");  
checkin1.setPassengerName("Jeffrey Chaucer");  
ObjectMessage msg1 = jmsContext.createObjectMessage();  
msg1.setObject(checkin1);
```

```
FlightCheckin checkin2 = new FlightCheckin();  
checkin2.setId(14);  
checkin2.setFlightRoute("LHA-VCR");  
checkin2.setPassengerDocumentType("Canadian Passport");  
checkin2.setPassengerDocumentNumber("59856C456GH9");  
checkin2.setPassengerName("Margaret Feller");  
ObjectMessage msg2 = jmsContext.createObjectMessage();  
msg2.setObject(checkin2);  
producer.send(requestQueue, msg1);  
producer.send(requestQueue, msg2);  
jmsContext.commit();  
} catch (JMSEException e) {  
    e.printStackTrace();  
};  
}
```

```
}
```

```
-----  
package com.bharath.jms.guaranteedmessaging;
```

```
import javax.jms.JMSConsumer;

import javax.jms.JMSContext;

import javax.jms.JMSEException;

import javax.jms.ObjectMessage;

import javax.jms.Queue;

import javax.jms.TextMessage;

import javax.naming.InitialContext;

import javax.naming.NamingException;


import org.apache.activemq.artemis.jms.client.ActiveMQConnectionFactory;


public class ReservationApp {

    public static void main(String[] args) throws NamingException {

        InitialContext initialContext = new InitialContext();

        Queue requestQueue = (Queue) initialContext.lookup("queue/requestQueue");

        try(ActiveMQConnectionFactory cf = new ActiveMQConnectionFactory();

        JMSContext jmsContext = cf.createContext(JMSContext.SESSION_TRANSACTED)) {

            JMSConsumer consumer = jmsContext.createConsumer(requestQueue);

            ObjectMessage message = (ObjectMessage) consumer.receive();

            FlightCheckin checkinData = message.getBody(FlightCheckin.class);

            System.out.println(checkinData);

            jmsContext.commit();

        } catch (JMSEException e) {

            e.printStackTrace();

        }
    }
}
```

}

}