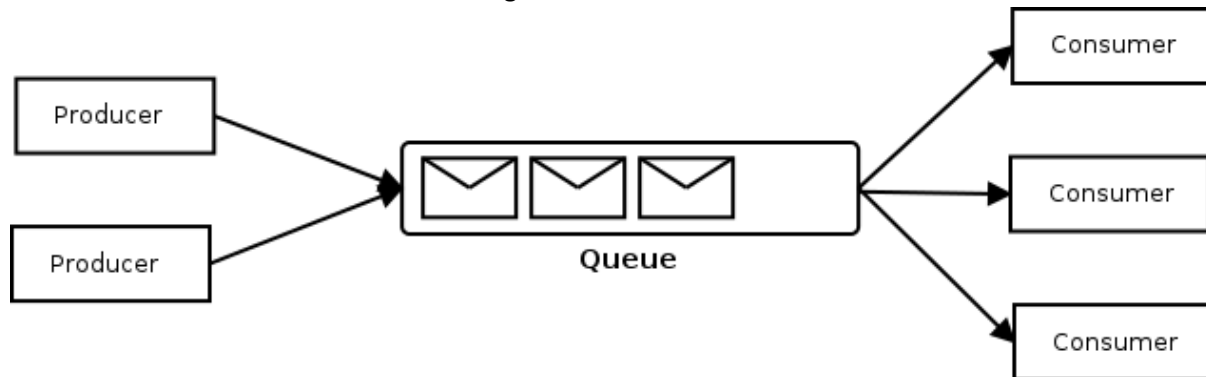


Zunächst einmal ist wichtig festzuhalten was unter der Activemq versteht. Mithilfe der Activemq lassen sich im Grunde genommen Nachrichten in einer Art Warteschlange festhalten, bis sie aufgerufen werden. Somit muss der Nutzer, der bestimmte Nachrichten erhalten will, nicht zur selben Zeit online sein, zu der die Nachrichten geschickt werden.

Dieses Prinzip wird als Queue verstanden. Die Nachrichten in der Warteschlange warten sozusagen darauf, bis sie aufgerufen werden.

Hier ein kleines bild zur Veranschaulichung:



Wie man hier sieht, gibt es zwei Rollen, einmal den Producer, der die Nachrichten an die Queue schickt, wo sie dann auf die Entgegennahme des Consumers warten. Es gibt verschiedene Arten der Authentifikation, die von maßgeblicher Bedeutung sind, allerdings sieht diese Aufgabe es nicht vor diese explizit näher zu behandeln, für weitere Informationen, ist die Dokumentation von Activemq sehr hilfreich.

Kommen wir nun zum technischen Teil. Activemq habe ich in einem Docker-Container mit dem Befehl „docker run -d --name activemq-webcenter -p 8161:8161 -p 61616:61616“ webcenter/activemq laufen lassen. Hier habe ich erstmal eine Fehlermeldung bekommen:

```
6e4fa9324041bff1c39ad075112ddd1c2123a407efdea8893686fc921d2e9e47
```

```
docker: Error response from daemon: Ports are not available: exposing port TCP 0.0.0.0:61616 -> 0.0.0.0:0: listen tcp 0.0.0.0:61616: bind: An attempt was made to access a socket in a way forbidden by its access permissions.
```

Es hat sich nach etwas Recherche herausgestellt, dass ein Feature von „Winat“ dafür sorgt, dass die Ports für Applikationen reserviert werden, auch wenn sie diese nicht benutzen, weshalb ich keinen Zugriff hatte. Dafür habe ich einen leichten Workaround gefunden. Und zwar habe ich die Konsole als Administrator gestartet und diese drei Befehle eingegeben:

```
„net stop winat“
```

```
„docker start activemq-webcenter“
```

```
„net start winat“
```

Hier habe ich einfach kurzzeitig den winat Dienst gestoppt, sodass ich keine Portschwierigkeiten habe.

Der Dienst der Activemq Nachrichten entgegennimmt läuft standartmäßig auf tcp://localhost:61616

Wie man hier unschwer erkennen kann, wird hier Gebrauch von TCP gemacht, um die Nachrichten zu versenden. In java kann man dann so Nachrichten an die Queue schicken:

```
ConnectionFactory connectionFactory = new ActiveMQConnectionFactory(user, password, url);
connection = connectionFactory.createConnection();
connection.start();

// Create the session
session = connection.createSession(transacted: false, Session.AUTO_ACKNOWLEDGE);
destination = session.createQueue(subject); // Use createQueue for point-to-point commun

// Create the producer.
producer = session.createProducer(destination);
producer.setDeliveryMode(DeliveryMode.NON_PERSISTENT);

// Convert the WarehouseData object to a JSON string
Gson gson = new Gson();
String jsonString = gson.toJson(warehouseData);

// Create the message
TextMessage message = session.createTextMessage(jsonString);
producer.send(message);
System.out.println(message.getText());
```

Mithilfe der Library Gson kann man dann ganz einfach das Warehouse zu einem Json-String formatieren.

Nachdem ich das gemacht habe, sieht man hier, dass die Nachricht erfolgreich an die Queue ermittelt wurde:

Queues

Name	Number Of Pending Messages
------	----------------------------

Warehouse	1
-----------	---

Der Consumer, der diese dann entgegennimmt verbindet sich dann mit derselben URL und nimmt dann die Nachrichten entgegen und stellt sie ganz einfach dar. In meinem Fall geschieht das mit den Default Passwörten von Activemq admin admin.

```
[{"warehouseID":"772","warehouseName":"Pandabuy Warehouse","timestamp":"2023-12-05
14:27:15.831","warehouseCountry":"China","warehouseCity":"Shanghai","address":"Wing ding long xong","productData":
[{"productId":"PD822","productName":"Coffee","productCategory":"Coffee","productAmount":"457","productUnit":"pcs"}, {"productId":"PD393","productName":"Orange
Juice","productCategory":"Juice","productAmount":"940","productUnit":"pcs"}, {"productId":"PD137","productName":"Orange
Juice","productCategory":"Juice","productAmount":"168","productUnit":"pcs"}, {"productId":"PD035","productName":"Coffee","productCategory":"Coffee","productAmount":"119","productUnit":"pcs"}]]
```

Wie man sieht war dies ein voller Erfolg.

```
try {  
    ConnectionFactory connectionFactory = new ActiveMQConnectionFactory(user, password, brokerURL);  
    Connection connection = connectionFactory.createConnection();  
    connection.start();  
  
    Session session = connection.createSession(transacted: false, Session.AUTO_ACKNOWLEDGE);  
    Destination destination = session.createQueue(subject);  
    MessageConsumer consumer = session.createConsumer(destination);  
  
    while (true) {  
        Message message = consumer.receive(timeout: 1000); // Timeout set to 1 second  
  
        if (message instanceof TextMessage) {  
            TextMessage textMessage = (TextMessage) message;  
            messages.add(textMessage.getText());  
        } else if (message == null) {  
            break; // No more messages  
        }  
    }  
}
```

Literaturverzeichnis:

<https://activemq.apache.org/>

<https://blog.deanosim.net/windows-10-winnat-and-why-your-programs-cant-listen-on-certain-ports/>

<https://stackoverflow.com/questions/65272764/ports-are-not-available-listen-tcp-0-0-0-50070-bind-an-attempt-was-made-to>