DEZSYS_GK72_WAREHOUSE_MOM

@author: Sebastian Sailer

@version: 2023-11-28

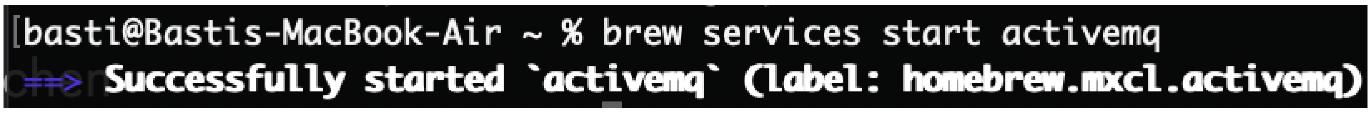
Arbeitsumgebung: MacOS Sonoma 14.0

ActiveMQ installieren:

```
brew install apache-activemq
```

Danach kann man den Service gleich starten:

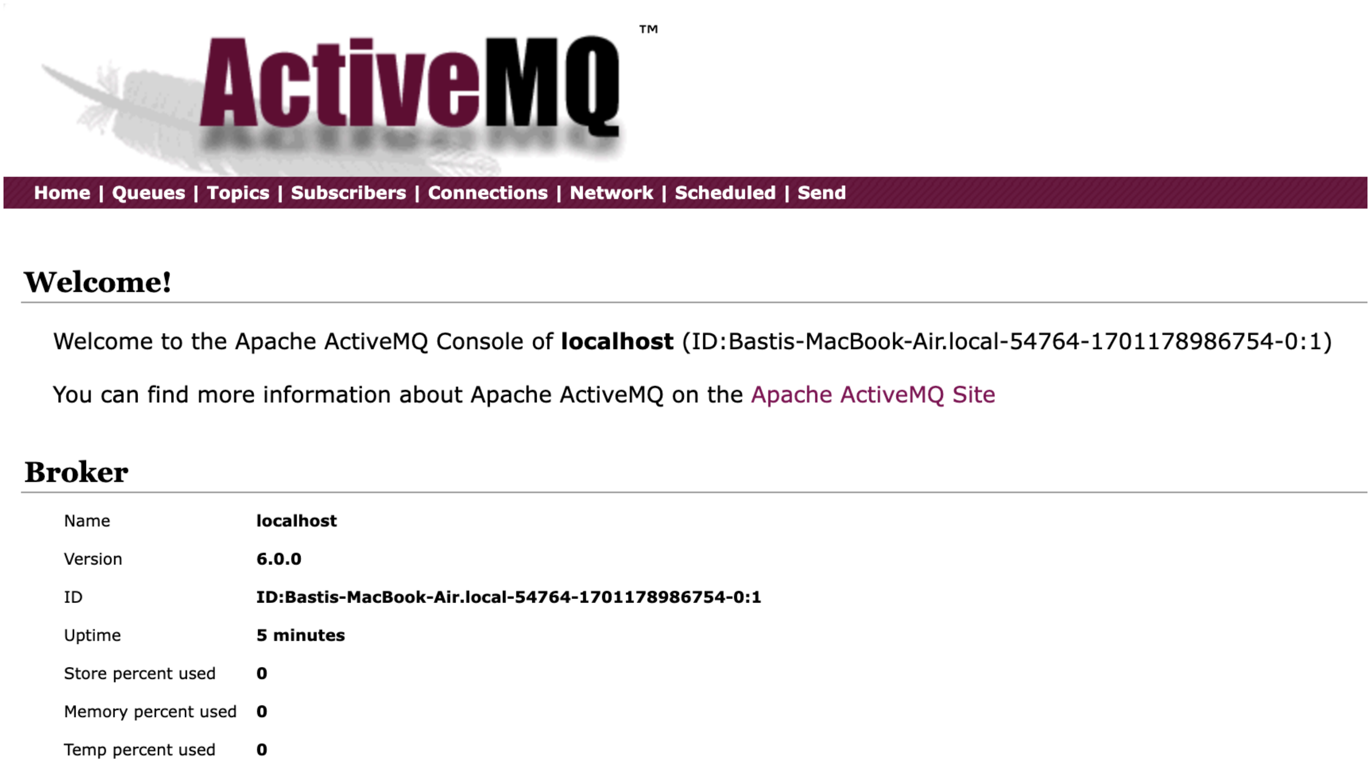
```
brew services start activemq
```



```
basti@Bastis-MacBook-Air ~ % brew services start activemq
==> Successfully started `activemq` (label: homebrew.mxcl.activemq)
```

Wird die Seite Admin aufgerufen wird man nach Benutzter und Passwort gefragt

(beides: admin) und dann ist man angemeldet


ActiveMQ™
Home | Queues | Topics | Subscribers | Connections | Network | Scheduled | Send

Welcome!

Welcome to the Apache ActiveMQ Console of **localhost** (ID:Bastis-MacBook-Air.local-54764-1701178986754-0:1)

You can find more information about Apache ActiveMQ on the Apache ActiveMQ Site

Broker

Name	**localhost**
Version	**6.0.0**
ID	**ID:Bastis-MacBook-Air.local-54764-1701178986754-0:1**
Uptime	**5 minutes**
Store percent used	**0**
Memory percent used	**0**
Temp percent used	**0**

Änderungen an MOMApplication.java

```java
package warehouse;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

import java.util.Collections;

/**
 * The Application that starts the MOM sender OR receiver
 * @author Sailer Sebastian
 * @version 12.12.2023
 */
@SpringBootApplication
public class MOMApplication {

    public static void main(String[] args) {
        SpringApplication app = new SpringApplication(MOMApplication.class);
        app.setDefaultProperties(Collections.singletonMap("server.port",
"8081"));
        app.run(args);
    }
}
```

Hier ändern wir den Port auf 8081, da 8080 bereits vom Warehouse der ersten Aufgabe belegt ist.

Änderungen an MOMController

```java
package warehouse;

import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.http.MediaType;


/**
 * Controller for consuming all the contents in the message queue
 * @author Sailer Sebastian
 * @version 12.12.2023
 */
@RestController
public class MOMController {
    private StringBuilder messageQueueResultsBuilder = new StringBuilder();

    @CrossOrigin
    @RequestMapping(value = "/warehouse/all", produces =
MediaType.APPLICATION_JSON_VALUE)
    public String allWarehouseData()    {

        new MOMSender();
        formatJSONString(new MOMReceiver().getAllWarehouseData());
        return this.messageQueueResultsBuilder.toString();
    }

    /**
     * Utility method that formats a java string into a valid JSON string
     * @param newMessage
     */
    public void formatJSONString(String newMessage)    {
        if (this.messageQueueResultsBuilder.isEmpty()) {
            this.messageQueueResultsBuilder.append("
[").append(newMessage).append("]");
        } else if
(this.messageQueueResultsBuilder.charAt(this.messageQueueResultsBuilder.length(
) - 1) == ']') {

this.messageQueueResultsBuilder.deleteCharAt(this.messageQueueResultsBuilder.le
ngth() - 1);

this.messageQueueResultsBuilder.append(",").append(newMessage).append("]");
        }
    }
}
```

Änderungen an MOMSender

Jetzt muss ein Sender eingerichtet werden, um Daten in die Message Queue schicken zu können.

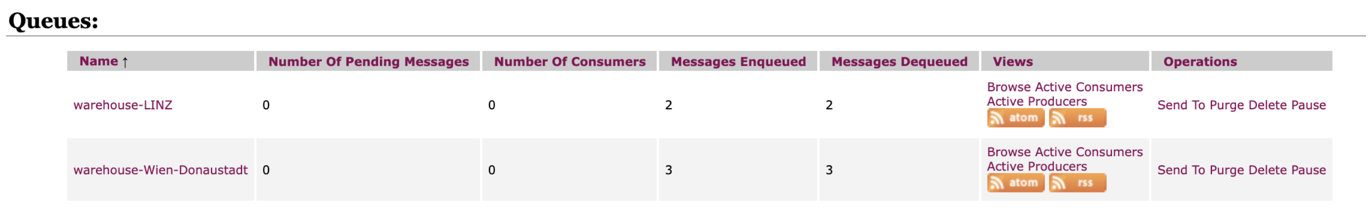
```java
package warehouse;

import org.apache.activemq.ActiveMQConnection;
import org.apache.activemq.ActiveMQConnectionFactory;
import org.springframework.web.client.RestTemplate;

import javax.jms.*;

/**
 * The sender class of the MOM Application - sends the data of all articles in
a warehouse as JSON
 * @author Sailer Sebastian
 * @version 12.12.2023
 */
public class MOMSender {

    private static String warehouseUUID = "001";
    private static String warehouseAPIUrl = "http://localhost:8080/warehouse/"
+ warehouseUUID + "/data";

    private static String user = ActiveMQConnection.DEFAULT_USER;
    private static String password = ActiveMQConnection.DEFAULT_PASSWORD;
    private static String url = ActiveMQConnection.DEFAULT_BROKER_URL;
    private static String queueName = "warehouse-Wien-Donaustadt";

    public MOMSender() {
        System.out.println("Sender started...");

        // create a connection to the apacheMQ broker
        Session session = null;
        Connection connection = null;
        MessageProducer producer = null;
        Destination destination = null;
        try {
            // init new connection
            ConnectionFactory connectionFactory = new
ActiveMQConnectionFactory(user, password, url);
            connection = connectionFactory.createConnection();
            connection.start();

            // Create the session
            session = connection.createSession(false,
Session.AUTO_ACKNOWLEDGE);
            destination = session.createQueue(queueName);

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

            // Create the message
            String currentWarehouseData = consumeWarehouseAPI();
            TextMessage message =
session.createTextMessage(currentWarehouseData);
            producer.send(message);
            System.out.println(message.getText());

            connection.stop();

        } catch (Exception e)   {
            System.out.println("[MessageProducer] Caught: " + e);
            e.printStackTrace();
        } finally {
            try { producer.close(); } catch ( Exception e ) {}
            try { session.close(); } catch ( Exception e ) {}
            try { connection.close(); } catch ( Exception e ) {}
        }
        System.out.println("Sender finished.");
    }

    public static String consumeWarehouseAPI() {
        System.out.println("Consuming the warehouse API with the url " +
warehouseAPIUrl + "...");
        RestTemplate restTemplate = new RestTemplate();
        return restTemplate.getForObject(warehouseAPIUrl, String.class);
    }
}
```

Änderungen MOMReceiver

```java
package warehouse;

import org.apache.activemq.ActiveMQConnection;
import org.apache.activemq.ActiveMQConnectionFactory;

import javax.jms.Connection;
import javax.jms.ConnectionFactory;
import javax.jms.Destination;
import javax.jms.MessageConsumer;
import javax.jms.Session;
import javax.jms.TextMessage;

/**
 * Receiver MOM Class - receives the data from a Topic
 * @author Sailer Sebastian
 * @version 12.12.2023
 */

public class MOMReceiver {
    private static String user = ActiveMQConnection.DEFAULT_USER;
    private static String password = ActiveMQConnection.DEFAULT_PASSWORD;
    private static String url = ActiveMQConnection.DEFAULT_BROKER_URL;
    private static String queueName = "warehouse-Wien-Donaustadt";

    public String getAllWarehouseData() {
        System.out.println( "Receiver started." );

        // Create the connection.
        Session session = null;
        Connection connection = null;
        MessageConsumer consumer = null;
        Destination destination = null;
        StringBuilder receivedMessages = null;

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

            // Create the session
            session = connection.createSession(false,
Session.AUTO_ACKNOWLEDGE);
            destination = session.createQueue(queueName);

            // Create the consumer
            consumer = session.createConsumer(destination);
            // Start receiving
            receivedMessages = new StringBuilder();

            TextMessage message = (TextMessage) consumer.receive(1000);
            while ( message != null ) {
                receivedMessages.append(message.getText());
                message.acknowledge();
                message = (TextMessage) consumer.receive(1000);
            }
            connection.stop();
        } catch (Exception e) {
            System.out.println("[MessageConsumer] Caught: " + e);
            e.printStackTrace();

        } finally {
            try { consumer.close(); } catch ( Exception e ) {}
            try { session.close(); } catch ( Exception e ) {}
            try { connection.close(); } catch ( Exception e ) {}

        }
        System.out.println( "Receiver finished." );
        System.out.println(receivedMessages.toString());
        return receivedMessages.toString();
    }

}
```

Anschließend über diesen Link den Sender und den Receiver triggern.

Es sollte folgendes Ergebnis dabei rauskommen:



Queues:

Name ↑	Number Of Pending Messages	Number Of Consumers	Messages Enqueued	Messages Dequeued	Views	Operations
warehouse-LINZ	0	0	2	2	Browse Active Consumers Active Producers atom rss	Send To Purge Delete Pause
warehouse-Wien-Donaustadt	0	0	3	3	Browse Active Consumers Active Producers atom rss	Send To Purge Delete Pause

```
[{"warehouseID":"001","warehouseName":"Linz Bahnhof","timestamp":"2023-12-12 16:49:13.348","warehouseProdukte":[{"timestamp":"2023-12-12
16:49:13.348","produktID":"1","produktName":"Apfel","produktBeschreibung":"lecker","produktPreis":1.0,"produktAnzahl":10,"produktEinheit":"Stück"},{"timestamp":"2023-12-12
16:49:13.348","produktID":"2","produktName":"Birne","produktBeschreibung":"lecker","produktPreis":1.0,"produktAnzahl":10,"produktEinheit":"Stück"},{"timestamp":"2023-12-12
16:49:13.348","produktID":"3","produktName":"Banane","produktBeschreibung":"gelb","produktPreis":2.5,"produktAnzahl":1,"produktEinheit":"Stück"},{"timestamp":"2023-12-12
16:49:13.348","produktID":"4","produktName":"Orange","produktBeschreibung":"rund","produktPreis":1.1,"produktAnzahl":12,"produktEinheit":"Stück"},{"timestamp":"2023-12-12
16:49:13.348","produktID":"5","produktName":"Kiwi","produktBeschreibung":"grün und haarig","produktPreis":1.4,"produktAnzahl":4,"produktEinheit":"Stück"},{"timestamp":"2023-12-12
16:49:13.348","produktID":"6","produktName":"Mango","produktBeschreibung":"gelb","produktPreis":1.0,"produktAnzahl":10,"produktEinheit":"Stück"},{"timestamp":"2023-12-12
16:49:13.348","produktID":"7","produktName":"Ananas","produktBeschreibung":"reif","produktPreis":4.5,"produktAnzahl":190,"produktEinheit":"Stück"},{"timestamp":"2023-12-12
16:49:13.348","produktID":"8","produktName":"Erdbeere","produktBeschreibung":"rot","produktPreis":6.25,"produktAnzahl":13,"produktEinheit":"Körbe"}]}
```