MICT - Labor 2

@author Gioia Frolik
@version 2023-12-12

Description

eine Queue pro Warehouse

Rest API

Headquarter
Empfänger (lesender Prozess - Daten über log file zur Verügung stellen)
Apache Active MQ
WH1
WH2 (für EK min 2 Warehouse)

XML JSON
(Rest API) -> nicht in Verwendung
{Warehouse Linz}

(Rest API) -> nicht in Verwendung
{Warehouse Klagenfurt}

EKv
1-Seiter Konzept Schawachstelle -
kontrollieren ob Nachrichten wirklich in der Zentrale angekommen sind

```
docker run -d --name activemq -p 61616:61616 -p 8161:8161
webcenter/activemq

docker ps -a

-> fa17b357ca18
webcenter/activemq
"/app/run.sh"
13 days ago
Created
activemq
```

Versionen

Java: 17
Gradle: 8.5

MOM Basis

Git klonen: https://github.com/ThomasMicheler/DEZSYS_GK772_WINDPARK_MOM.git

Der Demo 2 Ordner wird nicht gebraucht -> kann gelöscht werden

build.gradle

`build.gradle`

add:

```
implementation 'org.springframework.boot:spring-boot-starter-web'
implementation 'org.apache.activemq:activemq-all:5.16.3'
```

```
plugins {
    id 'java'
    id 'org.springframework.boot' version '3.2.0'
    id 'io.spring.dependency-management' version '1.1.4'
}

group = 'com.example'
version = '0.0.1-SNAPSHOT'

java {
    sourceCompatibility = '17'
}

repositories {
    mavenCentral()
}

dependencies {
    implementation 'org.springframework.boot:spring-boot-starter-activemq'
    implementation 'org.springframework.boot:spring-boot-starter-web'
    testImplementation 'org.springframework.boot:spring-boot-starter-test'
    implementation 'org.apache.activemq:activemq-all:5.16.3'
}

// The following block is not necessary for recent versions of the Spring Boot plugin
// classpath 'org.springframework.boot:spring-boot-gradle-plugin:3.2.0'


tasks.named('test') {
    useJUnitPlatform()
}
```

Per MOM auf Warehouse zugreifen

Code

MOMApplication.java

```
package windpark;

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

import java.util.Collections;

@SpringBootApplication
public class MOMApplication {

    public static void main(String[] args) {
        // We need to run this app on another port because the API from the
warehouse is taking up port 8080
        SpringApplication app = new
SpringApplication(MOMApplication.class);
        app.setDefaultProperties(Collections.singletonMap("server.port",
"8081"));
        app.run(args);
    }
}
```

MOMController.java

```
package windpark;

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 class */@RestController
public class MOMController {
    private StringBuilder messageQueueResultsBuilder = new
StringBuilder();

    @CrossOrigin
    @RequestMapping(value = "/warehouse/all", produces =
MediaType.APPLICATION_JSON_VALUE)
```

```
    public String allWarehouseData()    {
        // send & read & return all messages from the queue
        new warehouse.MOMSender();
        formatJSONString(new MOMReceiver().getAllWarehouseData());
        return this.messageQueueResultsBuilder.toString();
    }


    /**
     * 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.le
ngth() - 1) == ']') {

this.messageQueueResultsBuilder.deleteCharAt(this.messageQueueResultsBuild
er.length() - 1);

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

MOMReceiver.java

```
package windpark;

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 Class * receives data from Topic */
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-LINZ";

    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();
    }

}
```

MOMSender.java

```
package warehouse;

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

import javax.jms.*;

/**
 * sender class *  sends the data of all articles in a warehouse as JSON
*/public class MOMSender {

    private static String warehouseUUID = "469d7240-b974-441d-9562-
2c56a7b28767";
    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-LINZ";

    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);
    }
}
```

Errors :(

```
> Task :bootRun FAILED
Exception in thread "main" java.lang.NoSuchMethodError:
'org.springframework.core.io.support.SpringFactoriesLoader
org.springframework.core.io.support.SpringFactoriesLoader.forDefaultResour
ceLocation(java.lang.ClassLoader)'
        at
org.springframework.boot.SpringApplication.getSpringFactoriesInstances(Spr
ingApplication.java:482)
        at
org.springframework.boot.SpringApplication.getSpringFactoriesInstances(Spr
ingApplication.java:478)
        at org.springframework.boot.SpringApplication.<init>
(SpringApplication.java:282)
        at org.springframework.boot.SpringApplication.<init>
(SpringApplication.java:262)
```

```
        at windpark.MOMApplication.main(MOMApplication.java:13)

Execution failed for task ':bootRun'.
> Process 'command '/usr/lib/jvm/java-17-openjdk/bin/java'' finished with
non-zero exit value 1

* Try:
> Run with --stacktrace option to get the stack trace.
> Run with --info or --debug option to get more log output.
> Run with --scan to get full insights.
> Get more help at https://help.gradle.org.
BUILD FAILED in 5s
3 actionable tasks: 3 executed
```

```
Execution failed for task ':bootRun'.
> Process 'command '/usr/lib/jvm/java-17-openjdk/bin/java'' finished with
non-zero exit value 1

* Try:
> Run with --stacktrace option to get the stack trace.
> Run with --info or --debug option to get more log output.
> Run with --scan to get full insights.
> Get more help at https://help.gradle.org.
BUILD FAILED in 5s
3 actionable tasks: 3 executed
```