TP N°1_JMS :

Implémentation d'une communication asynchrone avec JMS, Spring et ActiveMQ (Artemis)

1. Prérequis

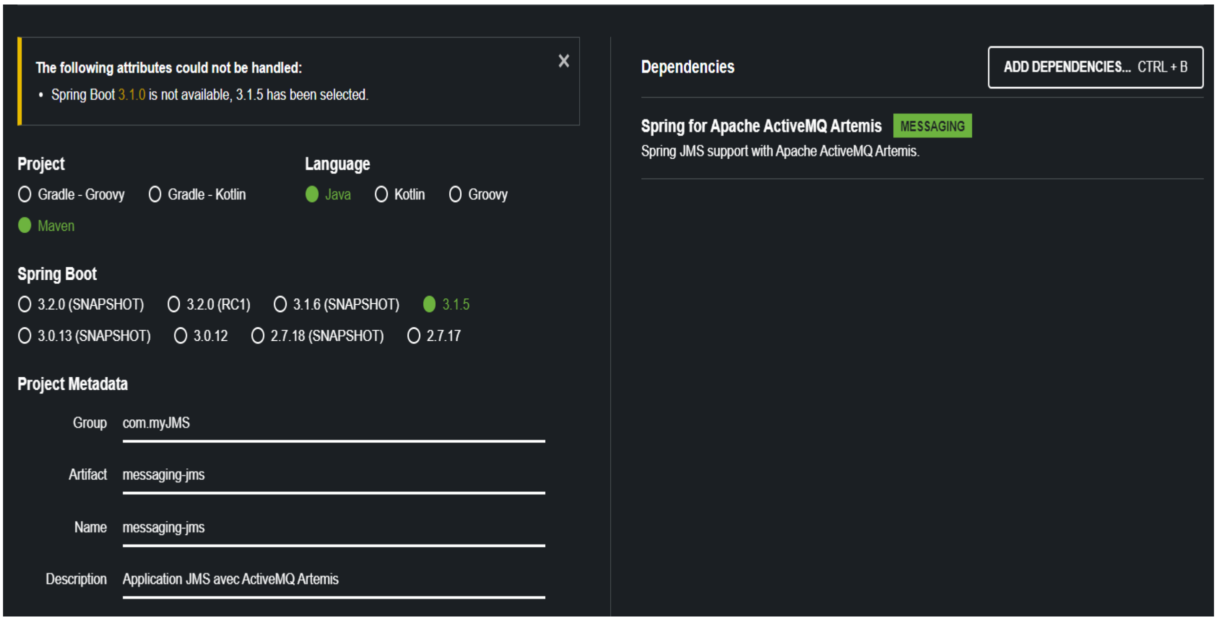
- ➢ JDK 17
- ➢ Connexion internet

2. Objectifs

1. Développement d’un producer JMS : @EnableJms, JmsTemplate
2. Développement d’un consumer JMS en mode asynchrone : @JmsListener
3. Envoyer/Réceptionner un objet Java « Email » à travers le Broker de type Embded ActiveMQ(Artemis)

3. Développement

- Utiliser l’initializer pour créer le projet « messaging-jms »
- Ou bien créer un projet MAVEN « messaging-jms »



The following attributes could not be handled:

- Spring Boot 3.1.0 is not available, 3.1.5 has been selected.

Project

○ Gradle - Groovy ○ Gradle - Kotlin ● Maven

Language

● Java ○ Kotlin ○ Groovy

Spring Boot

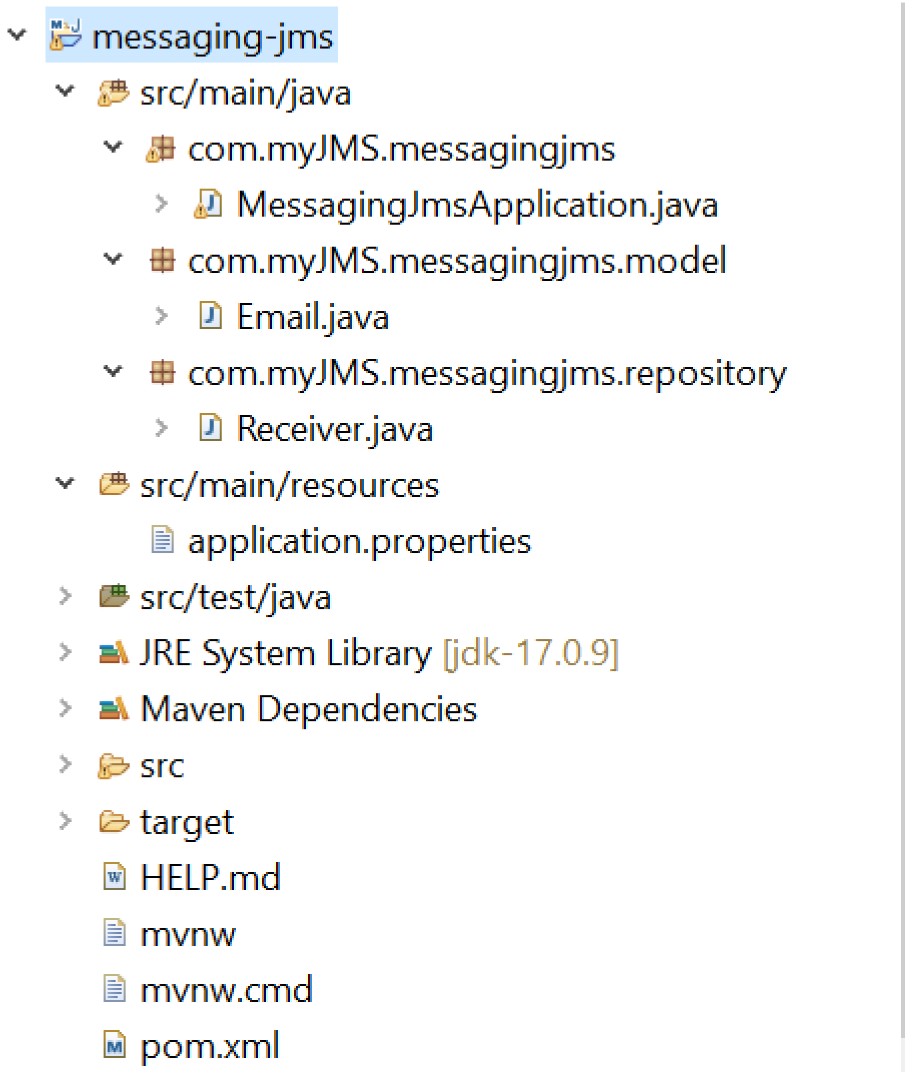
○ 3.2.0 (SNAPSHOT) ○ 3.2.0 (RC1) ○ 3.1.6 (SNAPSHOT) ● 3.1.5
○ 3.0.13 (SNAPSHOT) ○ 3.0.12 ○ 2.7.18 (SNAPSHOT) ○ 2.7.17

Project Metadata

Group: com.myJMS
Artifact: messaging-jms
Name: messaging-jms
Description: Application JMS avec ActiveMQ Artemis

Dependencies

Spring for Apache ActiveMQ Artemis MESSAGING
Spring JMS support with Apache ActiveMQ Artemis.



```
messaging-jms
    src/main/java
        com.myJMS.messagingjms
            MessagingJmsApplication.java
        com.myJMS.messagingjms.model
            Email.java
        com.myJMS.messagingjms.repository
            Receiver.java
    src/main/resources
        application.properties
    src/test/java
    JRE System Library [jdk-17.0.9]
    Maven Dependencies
    src
    target
    HELP.md
    mvnw
    mvnw.cmd
    pom.xml
```

- Modifier le fichier « pom.xml » généré comme suit :

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
	<modelVersion>4.0.0</modelVersion>
	<parent>
		<groupId>org.springframework.boot</groupId>
		<artifactId>spring-boot-starter-parent</artifactId>
		<version>3.1.5</version>
		<relativePath/> <!-- lookup parent from repository -->
	</parent>
	<groupId>com.myJMS</groupId>
	<artifactId>messaging-jms</artifactId>
	<version>0.0.1-SNAPSHOT</version>
	<name>messaging-jms</name>
	<description>Application JMS avec ActiveMQ Artemis</description>
	<properties>
		<java.version>17</java.version>
	</properties>
	<dependencies>
```

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-artemis</artifactId>
</dependency>
<dependency>
  <groupId>org.apache.activemq</groupId>
  <artifactId>artemis-jakarta-server</artifactId>
</dependency>

		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-json</artifactId>
		</dependency>

		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-test</artifactId>
			<scope>test</scope>
		</dependency>
	</dependencies>

	<build>
		<plugins>
			<plugin>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-maven-plugin</artifactId>
			</plugin>
		</plugins>
	</build>

</project>
```

a. Le fichier « application.properties »

```
spring.artemis.mode=embedded
```

a. La classe model« Email »

```
package com.myJMS.messagingjms.model;

public class Email {
	private String to;
	private String body;

	public Email() {
	}
	public Email(String to, String body) {
	  this.to = to;
```

```java
     this.body = body;
    }
    public String getTo() {
     return to;
    }
    public void setTo(String to) {
     this.to = to;
    }
    public String getBody() {
     return body;
    }
    public void setBody(String body) {
     this.body = body;
    }
    @Override
    public String toString() {
     return String.format("Email{to=%s, body=%s}",
getTo(), getBody());
    }
   }
```

b. Développement du producer « `MessagingJmsApplication` »:

```java
package com.myJMS.messagingjms;

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

import jakarta.jms.ConnectionFactory;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import
org.springframework.boot.autoconfigure.jms.DefaultJmsListenerContainerFactoryCon
figurer;
import org.springframework.context.ConfigurableApplicationContext;
import org.springframework.context.annotation.Bean;
import org.springframework.jms.annotation.EnableJms;
import org.springframework.jms.config.DefaultJmsListenerContainerFactory;
import org.springframework.jms.config.JmsListenerContainerFactory;
import org.springframework.jms.core.JmsTemplate;
import
org.springframework.jms.support.converter.MappingJackson2MessageConverter;
import org.springframework.jms.support.converter.MessageConverter;
import org.springframework.jms.support.converter.MessageType;

import com.myJMS.messagingjms.model.Email;

@SpringBootApplication

//@EnableJms enables detection of @JmsListener annotations on any Spring-managed
bean in the container.
@EnableJms
public class MessagingJmsApplication {
  @Bean
  public JmsListenerContainerFactory<?> myFactory(ConnectionFactory
connectionFactory,DefaultJmsListenerContainerFactoryConfigurer configurer) {

    DefaultJmsListenerContainerFactory factory = new
DefaultJmsListenerContainerFactory();

    // This provides all auto-configured defaults to this factory, including the
message converter
    configurer.configure(factory, connectionFactory);

    // You could still override some settings if necessary.
    return factory;
  }

  @Bean // Serialize message content to json using TextMessage
  public MessageConverter jacksonJmsMessageConverter() {
    MappingJackson2MessageConverter converter = new
MappingJackson2MessageConverter();
    converter.setTargetType(MessageType.TEXT);
    converter.setTypeIdPropertyName("_type");
    return converter;
  }

  public static void main(String[] args) {
    // Launch the application
        System.out.println(" *** Demarrage : MessagingJmsApplication 1");
        ConfigurableApplicationContext context =
SpringApplication.run(MessagingJmsApplication.class, args);

        System.out.println(" *** MessagingJmsApplication 2 : apres appel
ConfigurableApplicationContext ");

        JmsTemplate jmsTemplate = context.getBean(JmsTemplate.class);
        System.out.println(" *** MessagingJmsApplication 3 : apres appel
context.getBean(JmsTemplate.class) ");

    // Send a message with a POJO - the template reuse the message converter
    System.out.println(" *** MessagingJmsApplication 3 : Sending an email
message.");
    jmsTemplate.convertAndSend("mailbox", new Email("info@example.com", "Hello
from Spring JMS"));

    System.out.println(" *** MessagingJmsApplication 4 :
jmsTemplate.convertAndSend.");
  }
}
```

c. Développement du consommateur

```
package com.myJMS.messagingjms.repository;

import org.springframework.jms.annotation.JmsListener;
import org.springframework.stereotype.Component;

import com.myJMS.messagingjms.model.Email;
```

```java
@Component
public class Receiver {

  @JmsListener(destination = "mailbox", containerFactory =
"myFactory")
    public void receiveMessage(Email email) {

    System.out.println("---- repository.Received from sender: <" +
email + ">");
  }

}
```

d. Exécuter la classe main : `MessagingJmsApplication`

```
*** Demarrage : MessagingJmsApplication 1
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v3.1.5)

Starting MessagingJmsApplication using Java 17.0.9 with PID 11320
(JMS\TP_JMS\messaging-jms)

o.apache.activemq.artemis.core.server    : AMQ221000: live Message Broker is starting
with configuration Broker Configuration
o.apache.activemq.artemis.core.server    : AMQ221057: Global Max Size is being adjusted
to 1/2 of the JVM max size (-Xmx). being defined as 2106589184
o.apache.activemq.artemis.core.server    : AMQ221043: Protocol module found: [artemis-
server]. Adding protocol support for: CORE
org.apache.activemq.audit.base           : AMQ601138: User anonymous@unknown is
getting notification info on target resource: ActiveMQServerImpl::name=localhost
o.apache.activemq.artemis.core.server    : AMQ224092: Despite disabled persistence,
page files will be persisted.
o.apache.activemq.artemis.core.server    : AMQ221080: Deploying address DLQ supporting
[ANYCAST]
org.apache.activemq.artemis.core.management.impl.QueueControlImpl@4207609e
o.apache.activemq.artemis.core.server    : AMQ221080: Deploying address ExpiryQueue
supporting [ANYCAST]
o.apache.activemq.artemis.core.server    : AMQ221007: Server is now live
o.apache.activemq.artemis.core.server    : AMQ221001: Apache ActiveMQ Artemis
Message Broker version 2.28.0 [localhost, nodeID=239d450c-9c50-11ee-b307-
a8934acf8ae7]
```

```
_type=com.myJMS.messagingjms.model.Email,_AMQ_ROUTING_TYPE=1]]@1493357564,
context: RoutingContextImpl(Address=mailbox, routingType=ANYCAST,
PreviousAddress=mailbox previousRoute:ANYCAST, reusable=true, version=-2147483642)
.................................................
***** durable queues mailbox:
- queueID=14 address:mailbox name:mailbox filter:null
***** non durable for mailbox:
.................................................

 *** MessagingJmsApplication 4 : jmsTemplate.convertAndSend.

---- repository.Received from sender: <Email{to=info@example.com, body=Hello from
Spring JMS}>
org.apache.activemq.audit.message       : AMQ601759: User anonymous@invm:0 added
acknowledgement of a message from mailbox:
CoreMessage[messageID=19,durable=true,userID=24118ab9-9c50-11ee-b307-
a8934acf8ae7,priority=4, timestamp=Sat Dec
```