

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра «Информатика»

Практический пример №8
JMS в Spring.

Составил: Старший преподаватель кафедры «Информатика»
Черниговский Алексей Сергеевич

Красноярск 2020

Содержание

1 Создание Maven проекта.....	3
2 Зависимости.....	3
3 Структура проекта.....	3
4 jmsConfig.xml конфигурация.....	4
5 Сущность Alert.....	4
6 Интерфейс AlertService.....	5
7 Реализация AlertServiceImpl.....	5
8 Класс приложения App.....	6
9 ActiveMQ.....	6
10 Описание.....	7

1 Создание Maven проекта

В данном примере будет использоваться пример созданный из архетипа `maven-archetype-quickstart`. Использование данного примера в своем webapp приложении не должно вызывать затруднений.

Зададим ему Artifact Id: `jms`.

2 Зависимости

```
<properties>
  <spring.version>5.2.8.RELEASE</spring.version>
</properties>

<dependency>
  <groupId>org.apache.activemq</groupId>
  <artifactId>activemq-spring</artifactId>
  <version>5.16.0</version>
</dependency>

<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-jms</artifactId>
  <version>${spring.version}</version>
</dependency>

<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context</artifactId>
  <version>${spring.version}</version>
</dependency>

<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-core</artifactId>
  <version>${spring.version}</version>
</dependency>
```

3 Структура проекта

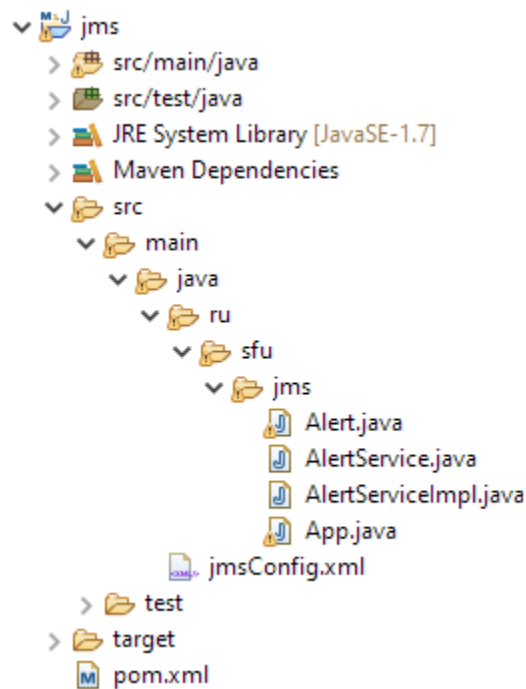


Рисунок 1 — структура проекта

4 jmsConfig.xml конфигурация

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:jms="http://www.springframework.org/schema/jms"
xmlns:amq="http://activemq.apache.org/schema/core"
xsi:schemaLocation="http://activemq.apache.org/schema/core
http://activemq.apache.org/schema/core/activemq-core-5.5.0.xsd
http://www.springframework.org/schema/jms
http://www.springframework.org/schema/jms/spring-jms-3.0.xsd
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

<amq:connectionFactory id="connectionFactory"
brokerURL="tcp://localhost:61616"
trustAllPackages="true"/>

<bean id="jmsTemplate"
class="org.springframework.jms.core.JmsTemplate">
<property name="connectionFactory" ref="connectionFactory" />
<property name="defaultDestinationName" value="student.alert.queue"/>
</bean>

</beans>
```

5 Сущность Alert

```
package ru.sfu.jms;

import java.io.Serializable;

public class Alert implements Serializable {
public Alert() {
```

```

}
public Alert(String to, String body) {
    super();
    this.to = to;
    this.body = body;
}
@Override
public String toString() {
    return "Alert [to=" + to + ", 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;
}
private String to;
private String body;
}

```

6 Интерфейс AlertService

```

package ru.sfu.jms;

import org.springframework.stereotype.Component;

@Component
public interface AlertService {
    void sendAlert(Alert alert);
    Alert getAlert();
}

```

7 Реализация AlertServiceImpl

```

package ru.sfu.jms;

import javax.jms.JMSException;
import javax.jms.Message;
import javax.jms.ObjectMessage;
import javax.jms.Session;

import org.springframework.jms.core.JmsTemplate;
import org.springframework.jms.core.MessageCreator;
import org.springframework.jms.support.JmsUtils;

public class AlertServiceImpl implements AlertService {
    public void sendAlert(final Alert alert) {
        jmsTemplate.send( new MessageCreator() {
            public Message createMessage(Session session) throws JMSException {
                return session.createObjectMessage(alert); // Создание
                                                         сообщения
            }
        }
    }
}

```

```

    );
}

public Alert getAlert() {
    try {
        ObjectMessage receivedMessage =
            (ObjectMessage)jmsTemplate.receive(); // Прием сообщения
        return (Alert) receivedMessage.getObject(); // Извлечение объекта
    } catch (JMSException jmsException) {
        // Возбудить преобразованное исключение
        throw JmsUtils.convertJmsAccessException(jmsException);
    }
}

public AlertServiceImpl(JmsTemplate jmsTemplate){
    this.jmsTemplate = jmsTemplate;
}

JmsTemplate jmsTemplate;
}

```

8 Класс приложения App

```

package ru.sfu.jms;

import org.springframework.context.support.ClassPathXmlApplicationContext;
import org.springframework.jms.core.JmsTemplate;

public class App {

    public static void main(String[] args ) {

        ClassPathXmlApplicationContext context = new
            ClassPathXmlApplicationContext("jmsConfig.xml");
        JmsTemplate jmsTemplate = context.getBean("jmsTemplate",
                                                    JmsTemplate.class);

        AlertService alertService = new AlertServiceImpl(jmsTemplate);
        alertService.sendAlert(new Alert("Neo!", "Wake up!"));

        Alert alert = alertService.getAlert();
        System.out.println(alert.getBody());
    }
}

```

9 ActiveMQ

Для работы приложения необходим брокер сообщений, будем использовать ActiveMQ.

Скачать его можно по адресу <http://activemq.apache.org/components/classic/download/>

Выберите брокер для своей операционной системы, в моем случае это Windows.

Распакуйте архив, в папке bin пройдите по пути, соответствующему битности вашей ОС, в моем случае это /bin/win64 и через командную строку запустите activemq.bat. Брокер сообщений готов к работе.

Для завершения работы брокера нажмите сочетание клавиш `ctrl+c`.

10 Описание

Приложение представляет собой простейшую отправку объекта в очередь сообщений и его последующую приемку. За подробностями обратитесь к конспекту лекции.