

Apache Camel with SQL Example

[Home \(index.php\)](#) / [Integration Frameworks](#) / [Apache Camel \(apache-camel.php\)](#)
/ [Apache Camel with SQL Example](#)

This example shows how to exchange data using a shared database table.

The example has two Camel routes. The first route insert new data into the table, triggered by a timer to run every 5th second.

The second route pickup the newly inserted rows from the table, process the row(s), and mark the row(s) as processed when done; to avoid picking up the same rows again..

NewTopicBean.java

```

package com.javavillage.camel.sql;

import java.util.Date;
import java.util.HashMap;
import java.util.Map;
import java.util.Random;

/**
 * Bean that generates and process NewTopic.
 */
public class NewTopicBean {

    private Random ran = new Random();

    /**
     * Generates a new topic structured as a {@link Map}
     */
    public Map<String, Object> generateNewTopic() {
        Map<String, Object> answer = new HashMap<String, Object>();
        answer.put("TopicId", ran.nextInt());
        answer.put("TopicName", "Camel in Action");
        answer.put("url", "Camel in Action");
        answer.put("ModuleId", ran.nextInt());
        answer.put("CreateDate", new Date());
        return answer;
    }

    /**
     * Processes the NewTopic
     * @param data the NewTopic as a {@link Map}
     * @return the transformed NewTopic
     */
    public String processNewTopic(Map<String, Object> data) {
        return "Processed NewTopic id " + data.get("TopicId") + " TopicName "
            + data.get("TopicName")
            + " of " + data.get("ModuleId") + " copies of " + data.get("url");
    }
}

```

sql.properties

```

## notice we use named parameters in the queries, eg :#name.
## A named query parameter must start with :#
## sql that insert new orders
sql.insertNewTopic=INSERT INTO newtopic(TopicId, TopicName, url, ModuleId, CreateDate) VALUES
(:#TopicId, :#TopicName, :#url, :#ModuleId, :#CreateDate)

## sql that select all unprocessed NewTopics
sql.selectNewTopic=select * from newtopic

## sql that update the NewTopic as being processed
sql.markNewTopic=update newtopic set TopicName = 'Apache Camel' where TopicId = :#TopicId

```

log4j.properties

```
#  
# The logging properties  
#  
log4j.rootLogger=INFO, out  
  
#log4j.logger.org.apache.camel.component.sql=DEBUG  
  
# CONSOLE appender not used by default  
log4j.appender.out=org.apache.log4j.ConsoleAppender  
log4j.appender.out.layout=org.apache.log4j.PatternLayout  
log4j.appender.out.layout.ConversionPattern=[%30.30t] %-30.30c{1} %-5p %m%n  
#log4j.appender.out.layout.ConversionPattern=%d [%-15.15t] %-5p %-30.30c{1} - %m%n
```

Router in Xml Configuration:(applicationContext-camel.xml)

```

<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:camel="http://camel.apache.org/schema/spring"
       xsi:schemaLocation="http://www.springframework.org/schema/beans

       http://www.springframework.org/schema/beans/spring-beans.xsd
       http://camel.apache.org/schema/spring
       http://camel.apache.org/schema/spring/camel-spring.xsd">

    <bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource"
          destroy-method="close">
        <property name="driverClassName" value="com.mysql.jdbc.Driver" />
        <property name="url" value="jdbc:mysql://localhost/javavill_forum" />
        <property name="username" value="root" />
        <property name="password" value="" />
    </bean>

    <!-- configure the Camel SQL component to use the JDBC data source -->
    <bean id="sqlComponent" class="org.apache.camel.component.sql.SqlComponent">
        <property name="dataSource" ref="dataSource" />
    </bean>

    <bean id="topicBean" class="com.javavillage.camel.sql.NewTopicBean" />

    <!-- here is Camel configured with a number of routes -->
    <camelContext xmlns="http://camel.apache.org/schema/spring">

        <!-- use Camel property placeholder loaded from the given file -->
        <propertyPlaceholder id="placeholder" location="classpath:sql.properties" />

        <!-- route that generate new orders and insert them in the database -->
        <route id="generateOrder-route">
            <from uri="timer:foo?period=5s" />
            <transform>
                <method ref="topicBean" method="generateNewTopic" />
            </transform>
            <to uri="sqlComponent:{{sql.insertNewTopic}}" />
            <log message="Inserted new NewTopic ${body[TopicId]}" />
        </route>

        <!--
            route that process the NewTopics by picking up new rows from the
            database and when done processing then update the row to mark it as
            processed
        -->
        <route id="processNewTopic-route">
            <from uri="sqlComponent:{{sql.selectNewTopic}}?
                                consumer.onConsume={{sql.markNewTopic}}" />
            <to uri="bean:topicBean?method=processNewTopic" />
            <log message="${body}" />
            <log message="Updated new NewTopic " />
        </route>

    </camelContext>

```

```
</beans>
```

Maven entries for pom.xml

```
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-core</artifactId>
  <version>2.13.0</version>
</dependency>
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-spring</artifactId>
  <version>2.13.0</version>
</dependency>

<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-jdbc</artifactId>
  <version>2.13.0</version>
</dependency>
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-sql</artifactId>
  <version>2.13.0</version>
</dependency>
<dependency>
  <groupId>commons-dbcp</groupId>
  <artifactId>commons-dbcp</artifactId>
  <version>1.4</version>
</dependency>
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.6</version>
</dependency>
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-api</artifactId>
  <version>1.7.5</version>
</dependency>

<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-log4j12</artifactId>
  <version>1.7.5</version>
</dependency>
```

Below is my application execution

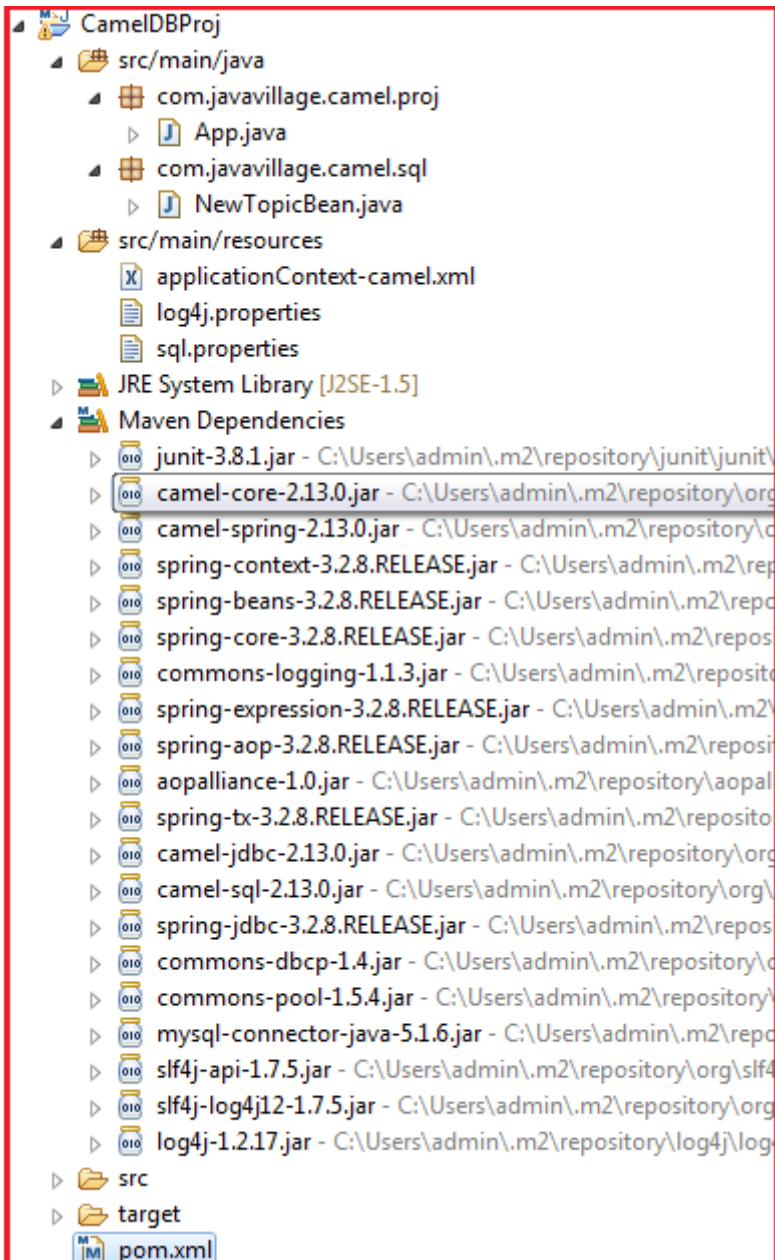
```
package com.javavillage.camel.proj;

import org.springframework.context.support.AbstractApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

/**
 * Hello world!
 *
 */
public class App
{
    public static void main( String[] args )
    {
        AbstractApplicationContext ctx = new ClassPathXmlApplicationContext
            ("applicationContext-camel.xml");

        ctx.start();
        try {
            Thread.sleep(5000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        System.out.println("Entered>>>>");
        ctx.stop();
    }
}
```

Apache Camel with SQL example Application Structure



Execute Apache Camel with SQL Application:

```

terminated> App (2) [Java Application] C:\JAVA\JDK\bin\javaw.exe (Oct 1, 2014 2:32:14 PM)
main] ClassPathXmlApplicationContext INFO Refreshing org.springframework.context.s
main] XmlBeanDefinitionReader INFO Loading XML bean definitions from class
main] DefaultListableBeanFactory INFO Pre-instantiating singletons in org.spri
main] SpringCamelContext INFO Apache Camel 2.13.0 (CamelContext: camel
main] ManagedManagementStrategy INFO JMX is enabled
main] DefaultTypeConverter INFO Loaded 176 type converters
main] SpringCamelContext INFO AllowUseOriginalMessage is enabled. If a
main] SpringCamelContext INFO StreamCaching is not in use. If using st
main] SpringCamelContext INFO Route: generateOrder-route started and c
main] SpringCamelContext INFO Route: processNewTopic-route started and
main] SpringCamelContext INFO Total 2 routes, of which 2 is started.
main] SpringCamelContext INFO Apache Camel 2.13.0 (CamelContext: camel
m%20newtopic] processNewTopic-route INFO Processed NewTopic id 90790042 TopicName
m%20newtopic] processNewTopic-route INFO Updated new NewTopic
timer://foo] generateOrder-route INFO Inserted new NewTopic -960933482
m%20newtopic] processNewTopic-route INFO Processed NewTopic id 1925721975 TopicNa
m%20newtopic] processNewTopic-route INFO Updated new NewTopic
m%20newtopic] processNewTopic-route INFO Processed NewTopic id 1106029111 TopicNa
m%20newtopic] processNewTopic-route INFO Updated new NewTopic
m%20newtopic] processNewTopic-route INFO Processed NewTopic id -960933482 TopicNa
m%20newtopic] processNewTopic-route INFO Updated new NewTopic
m%20newtopic] processNewTopic-route INFO Processed NewTopic id 90790042 TopicName
m%20newtopic] processNewTopic-route INFO Updated new NewTopic

```

Insert query will be called based on the timer and update query will be no time limit, same thing can be observed from screenshot.





Copyright © JavaVillage.in | JavaVillage.in blog (<http://javavillagein.blogspot.in/>)