

**ARTL for Java**

Java Technical Good

**<C0001-Camel+ActiveMQ >**

*Technical Design*

*Feb 2015 version 1*

# Contents

[1. Contents 2](#_Toc411973997)

[2. < C0001-Camel+ActiveMQ > TECHNICAL DESIGN 3](#_Toc411973998)

[ Deployment Descriptor(web.xml) 3](#_Toc411974000)

[ Spring configuration file (camel-config.xml) 3](#_Toc411974001)

[ Property File (camel.properties) 4](#_Toc411974002)

[ Loggers Configuration (log4j.xml) 4](#_Toc411974003)

[ Templates folder (Sample request and response) 5](#_Toc411974004)

[ EnrichProcessor.java 5](#_Toc411974005)

[ ReadXMLProcessor.java 6](#_Toc411974006)

[ TEST\_JMS\_REQUEST.xml 6](#_Toc411974000)

[ REQUEST\_HANDLER.xml 7](#_Toc411974001)

[ PROCESS\_HANDLER.xml 7](#_Toc411974002)

[ POM file 8](#_Toc411974003)

[ How to send the request on queue 8](#_Toc411974004)

[3. DOCUMENT CONTROL 11](#_Toc411974007)

[ Change history 11](#_Toc411974008)

[ Open issues 11](#_Toc411974009)

# 2. < C0001-CAMEL+ACTIVEMQ > Technical design

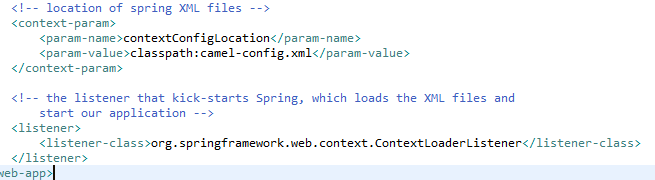
## Deployment Descriptor (web.xml)

In the web.xml two things are defined :

1. Spring configuration file (camel-config.xml) and ContextLoaderListener



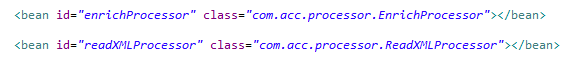
1. Camel servlet and its mapping (<url-pattern>/services/\*</url-pattern>)



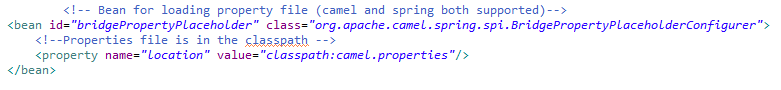
## Spring Configuration file (camel-config.xml)

This is the configuration file where following this have been defined:

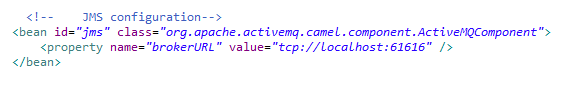
1. Java beans (*enrichProcessor and readXMLProcessor*).



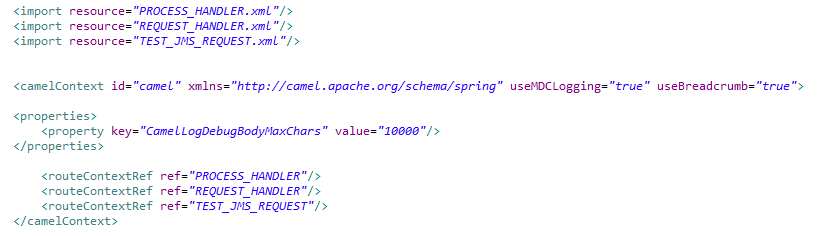
1. Property file configuration (camel.properties).



1. JMS configuration (configure ActiveMQ as Broker).

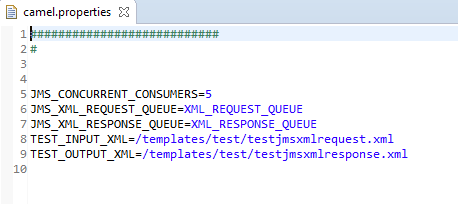


1. Import other files for defining camel routes for processing.



## Property File (camel.properties)

This file is imported in the camel-config.xml for defining the configurable things in the property file.

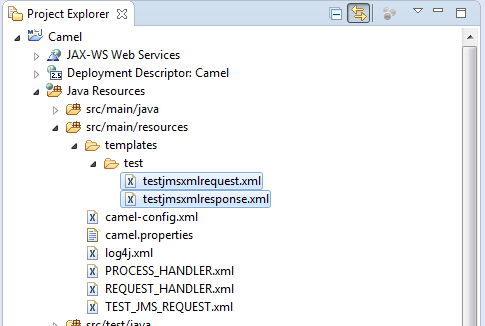


## Loggers Configuration (log4j.xml)

This file is imported in the camel-config.xml for defining the logging configuration for the project. It has been placed in the resource folder of the project.

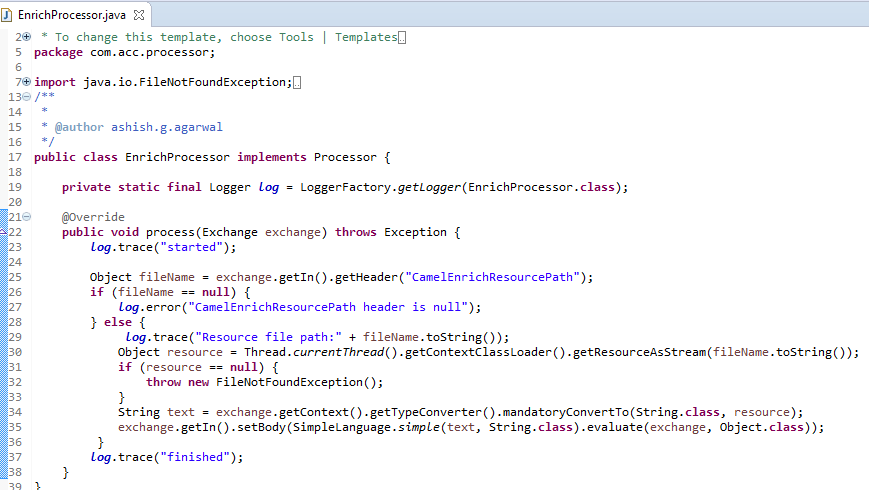
## Templates folder (Sample request and response)

This folder contains the sample request and response xml .



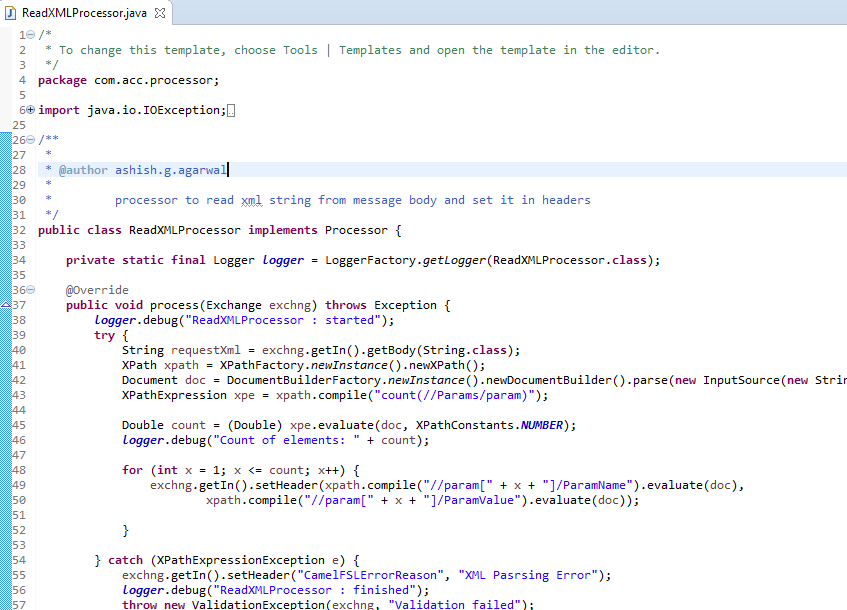
## EnrichProcessor.java

This processor class is used to read the document from the path which Is mentioned in the header named “CamelEnrichResourcePath” . Then this file is processed replace headers or properties mentioned in the file and then set this into the body of the exchange.



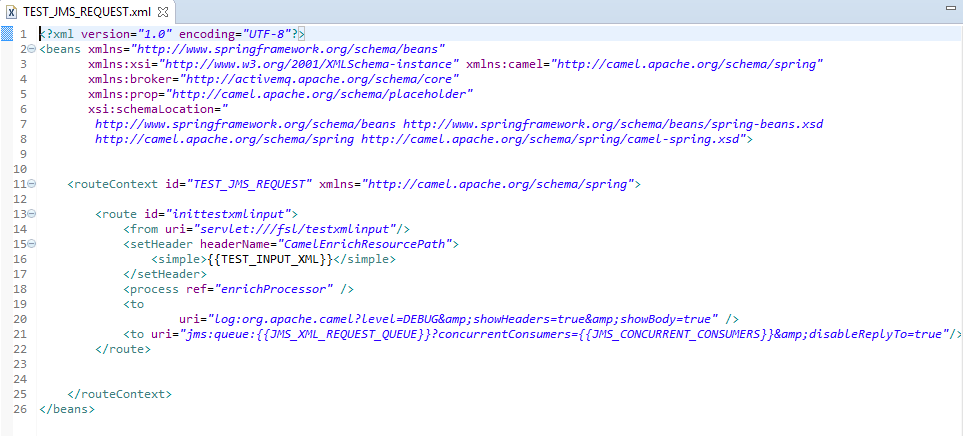
## ReadXMLProcessor.java

This processor class is used to read the XML file coming in the body of the exchange and sets the headers in the exchange by parsing the XML file which should be in the predefined format.



## TEST\_JMS\_REQUEST.xml

This is the example of how a request can be sent to the queue for processing . By invoking the servlet defined in the route of this file, request xml file is set to the body of the exchange and then it send the request to the queue by reading the values from the property file of the queueName and its concurrentConsumers.



## REQUEST\_HANDLER.xml

In this file request queue is defined which is triggered whenever any request came on this queue. After receiving the request further processing can be done .Reads the xml file coming in the body by invoking the readXMLProcessor class then redirect it to other routes for further processing.



## PROCESS\_HANDLER.xml

In this file request message is processed and response message is sent to the response queue.



## POM file

It contains all the specifications regarding the maven project. All the dependencies are defined in the POM file. Along with it plugins are also defined in this file for support in building and running the project.

## 

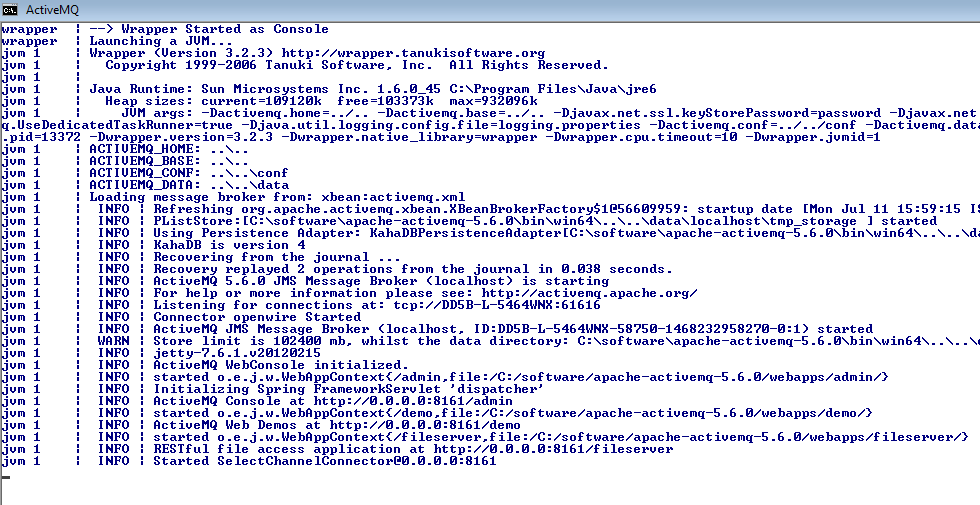
## How to send the request on queue

In the POM file of the project I have defined one plug-in “jetty-maven-plugin”. Jetty is the web server introduced by maven. So we can run the project on this server.

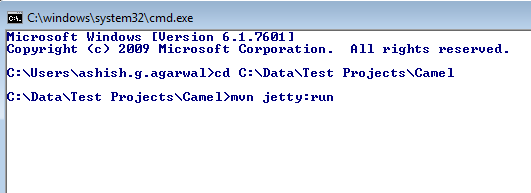


For running the project on jetty server follow below steps:

1. First of all start the Active MQ as mentioned in the installation and Configuration guide document.



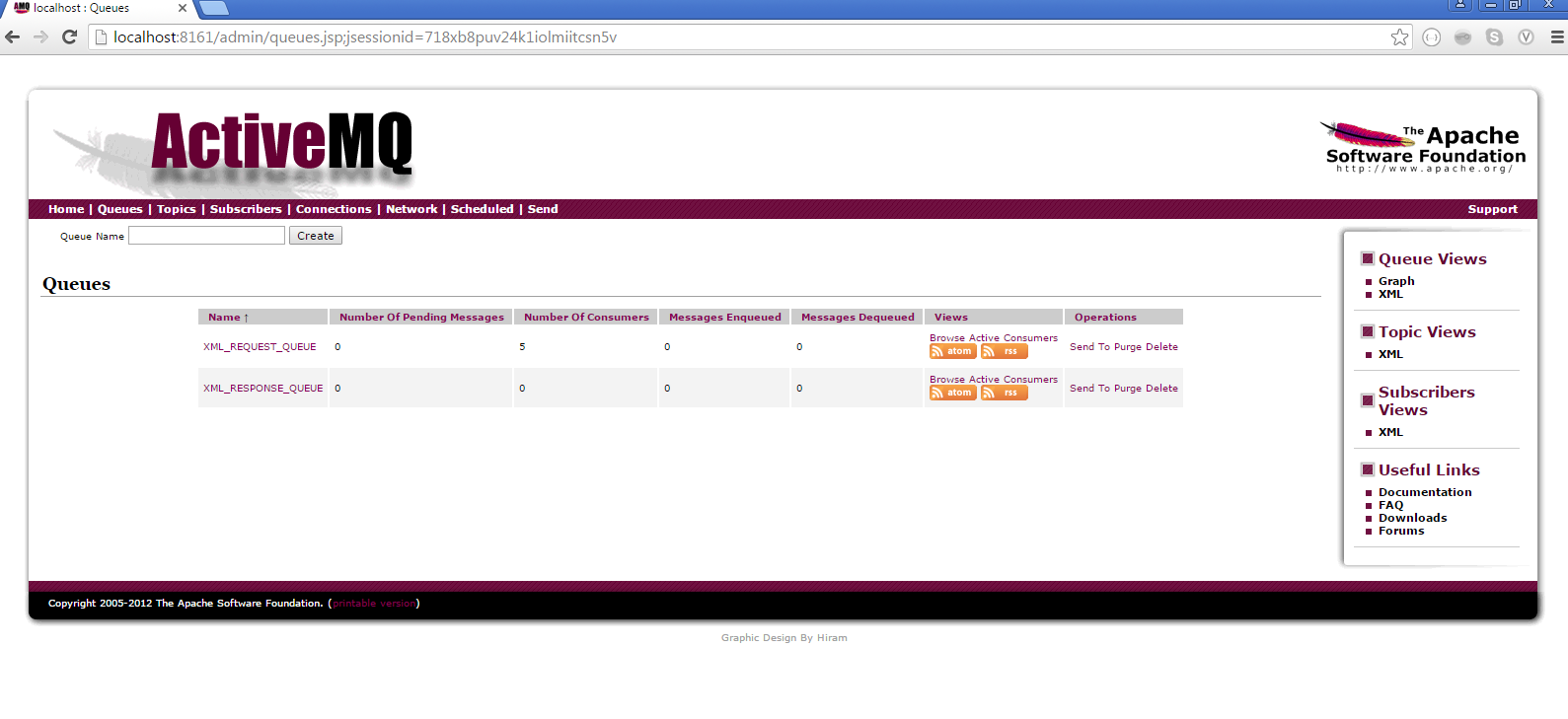
1. Go to the root folder of the project in the CMD where POM file is located.



1. Now run the command **mvn jetty:run**. (Make sure that maven is up and running in your system and system is connected to internet to download the dependencies which is the one time process)



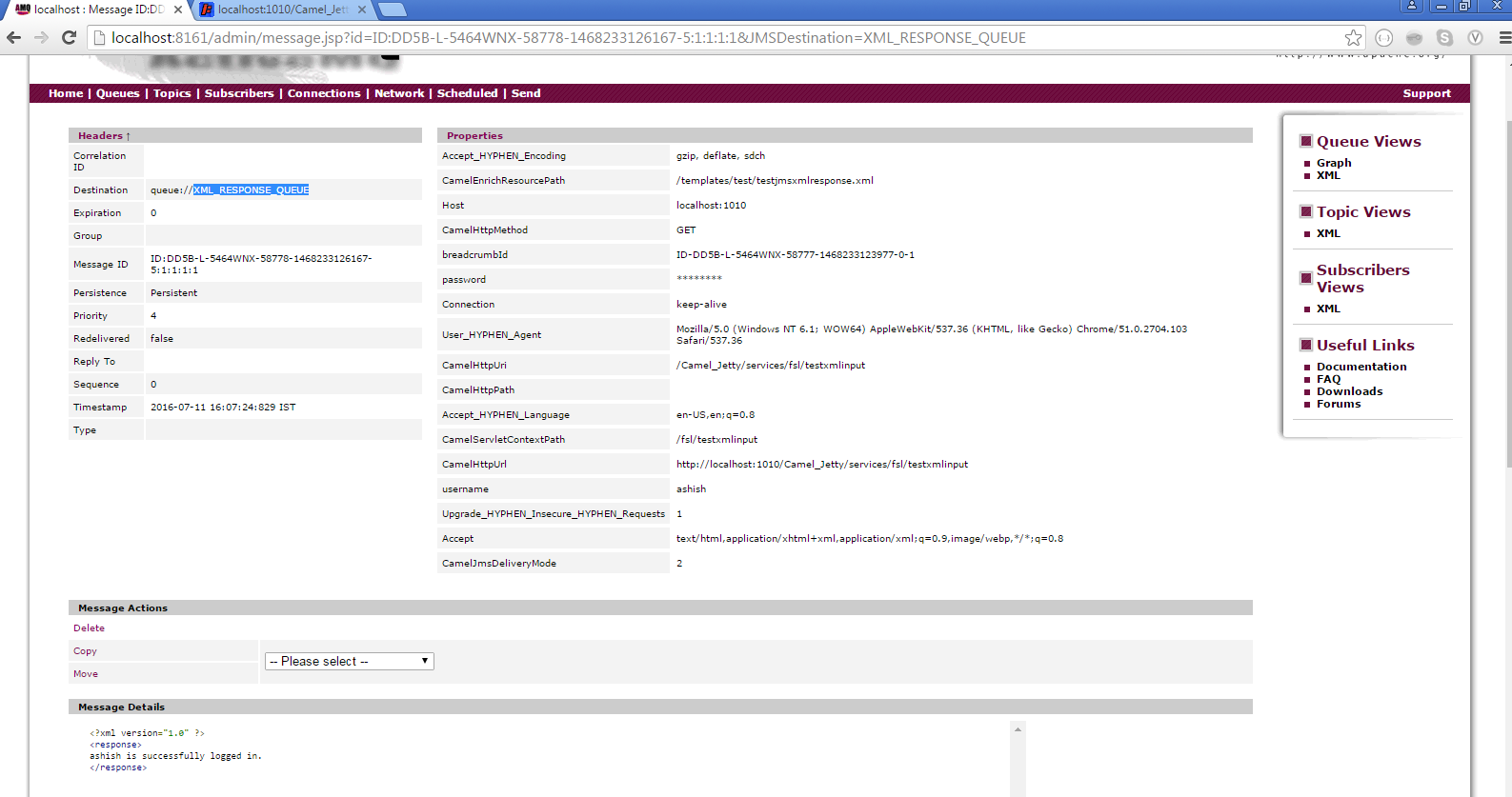
1. If everything goes right server will be started and you are ready to send the request to the queue.



1. Now open the browser and hit <http://localhost:1010/Camel_Jetty/services/fsl/testxmlinput> url.



1. This will invoke the request and response queues. Now browse the queue for verification.



# 3. DOCUMENT CONTROL

## Change history

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Doc Version** | **Date** | **Asset Version** | **Author** | **Approver** | **Comment** |
| 1 | 21/06/2016 | 1 | Ashish Agarwal |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Open issues

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Doc Version** | **Date** | **Asset Version** | **Author** | **Action** | **Owner** | **Status** |
|  | mm/dd/yyyy |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |