**Created a project on JMS to publish messages**

- Before going to create a project on JMS, we need to know few terms:

**JMS Connector:**

The JMS connector enables your application to do messaging using the JMS implementation of your choice.

Its main features include:

- Pub/Sub pattern support on any given destination.

Listen/Reply pattern support on any given destination.

Publish-Consume pattern support on any given destination, with fixed or temporary reply Queue.

Fully compliant with JMS 2.0, 1.1, and 1.0.2 specifications.

**Uses:**

JMS Connector JMS (Java Messaging Service) is mostly used API enabling the application to communicate through the exchange of message. JMS connector is capable of sending and receiving message to and from Topics/Queues.

JMS supports two models for messaging:

Queue (point to point).

Topic (publish-subscribe).

**Queue**:

It enables one-to-one communication. It is also called point-to-point communication.

The sender will deliver a message to the queue and single receivers will pick the message from the queue.

The receiver doesn't need to listen to queue at the time when the message is sent to the queue.

**Flow be like:**

Http listener -> Logger -> Jms Publish -> Logger

**HTTP listener:**

The HTTP listener is an event source that enables you to set up an HTTP server and trigger flows when HTTP requests are received.

You can choose what methods the source accepts, such as GET, POST or a list of methods, and on which path to accept requests, thereby allowing the routing of requests through different flows.

Once a request is accepted by the listener, the corresponding flow is triggered with the HTTP body as payload and the HTTP data as attributes (headers, query parameters and so on).

When the flow finishes its execution, the HTTP listener enables you to customize the HTTP response based on whether the execution was successful or not, so that different status codes can be returned.

**Logger:**

This Core component helps you monitor and debug your Mule application by logging important information such as error messages, status notifications, payloads, and so on. You can add a Logger anywhere in a flow, and you can configure it to log a string that you specify, the output of a Data Weave expression you write, or any combination of strings and expressions.

**Jms Publish:**

This operation allows you to create a new JMS Message and send it to the specified destination, be it a Queue or a Topic. With it, you can configure not only the content of the message, but also all the headers that may be need.

\* Sending message to Queue and Topic is almost similar but when deal with multiple targets you go with Topic or else queue.

**Flow Explanation on JMS:**

1. Take Http listener to listener the requests

2. Take Logger to monitor the result on console.

3. Take Jms publish to create new JMS message and store on ActiveMQ.

4. Take Jms consumer to consume whatever the message you have created in

last.