

# Asynchronous communications between programs with ActiveMQ

---

## Introduction

**Message broker** is an intermediary program module which translates a message from the formal messaging protocol of the sender to the formal messaging protocol of the receiver. Message brokers are elements in telecommunication networks where programs (software applications) communicate by exchanging formally-defined messages. Message brokers are a building block of Message Oriented Middleware ([https://en.wikipedia.org/wiki/Message\\_broker](https://en.wikipedia.org/wiki/Message_broker)).

**Apache ActiveMQ** is an open source message broker written in Java together with a full Java Message Service (JMS) client. It provides "Enterprise Features" which in this case means fostering the communication from more than one client or server. Supported clients include Java via JMS 1.1 as well as several other "cross language" clients. The communication is managed with features such as computer clustering and ability to use any database as a JMS persistence provider besides virtual memory, cache and journal persistency ([https://en.wikipedia.org/wiki/Apache\\_ActiveMQ](https://en.wikipedia.org/wiki/Apache_ActiveMQ)).

## Installation procedure

Before downloading ActiveMQ, check the Java version installed on your machine (JDK 1.6 is recommended but higher version should work).

Set if necessary the environment variable JAVA\_HOME (it must point to a valid JDK root directory).

Download Apache ActiveMQ binary version (version 5.x.x is recommended): follow the link <http://activemq.apache.org/>.

Uncompress the archive file.

Start ActiveMQ with the following command:

```
bin\activemq start
```

(you must be in the parent directory of bin).

## Configuring Queue and Topic

Start the web console: <http://localhost:8161/admin>

Default user name: admin

Default password: admin

Create a Queue, and a Topic.

Before starting coding answer the questions: what is a queue? What is a Topic?

## Project creation

Download a template for your project at <https://github.com/charroux/ActiveMQTemplate> .

This is a Gradle project.

Use the commands `gradle build` and `gradle eclipse` to download the required libraries and to convert the Gradle project into an Eclipse project.

Import the project into Eclipse.

## Coding

The given code contains only:

- Creation of a queue and a topic (see `src/main/resources/applicationContextJMS.xml`)
- Skeletons for accessing a queue in Java (see `src/main/java`)

Work to be done: study the slides of the course JMS API and complete the code to send messages from one sender to one receiver (using a queue and a topic).

Is a queue able to receive message from many senders?

Is a queue able to send message to many receivers?

Write code to test what is possible.

Is a topic able to receive message from many publishers?

Is a queue able to send message to many subscribers?

Write code to test what is possible.

## Advanced message sending

Code each possibility of message sending:

- Persistent mode
- Time to live
- Priority

Try to transmit messages into a transaction.