ActiveMQ Installation Windows

## Prerequesites

* Java 8 minimum installed

## Introduction

**Apache ActiveMQ** is an [open source](https://en.wikipedia.org/wiki/Open_source) [message broker](https://en.wikipedia.org/wiki/Message_broker) written in Java together with a full [Java Message Service](https://en.wikipedia.org/wiki/Java_Message_Service) (JMS) client. It provides "Enterprise Features" which in this case means fostering the communication from more than one client or server.

Since its inception, ActiveMQ has turned into a strong competitor of the commercial alternatives, such as WebSphereMQ, EMS/TIBCO and SonicMQ and is deployed in production in some of the top companies in industries ranging from financial services to retail.

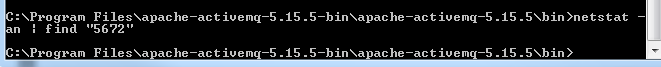
Using messaging as an integration or communication style leads to many benefits such as:

1. Allowing applications built with different languages and on different operating systems to integrate with each other
2. Location transparency – client applications don’t need to know where the service applications are located
3. Reliable communication – the producers/consumers of messages don’t have to be available at the same time, or certain segments along the route of the message can go down and come back up without impacting the message getting to the service/consumer
4. Scaling – can scale horizontally by adding more services that can handle the messages if too many messages are arriving
5. Asynchronous communication – a client can fire a message and continue other processing instead of blocking until the service has sent a response; it can handle the response message only when the message is ready
6. Reduced coupling – the assumptions made by the clients and services are greatly reduced as a result of the previous 5 benefits. A service can change details about itself, including its location, protocol, and availability, without affecting or disrupting the client.

## Install and Run the ActiveMQ broker on Windows

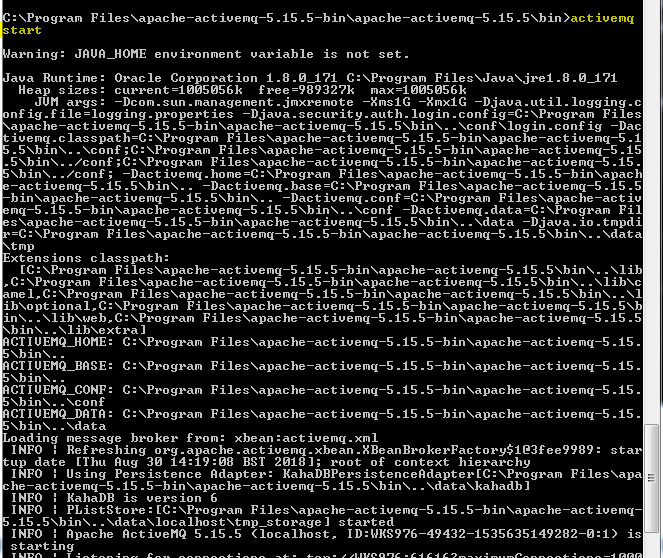
1. Download the zip archive from the ApacheMQ website and extract it locally.
2. Launch a terminal command prompt.
3. Navigate to the folder where the Apachemq server archive was extracted.
4. Check that port **5672** is free (no other message broker is running on it) and nothing is listening on it:

netstat –an | find “5672”



1. Go to the **bin** directory and start the **activemq** server by running:

activemq start



* Check the **ApacheMQ** server is running (listening on both default ports 5672 and 61616):

**netstat –an | find “5672”**



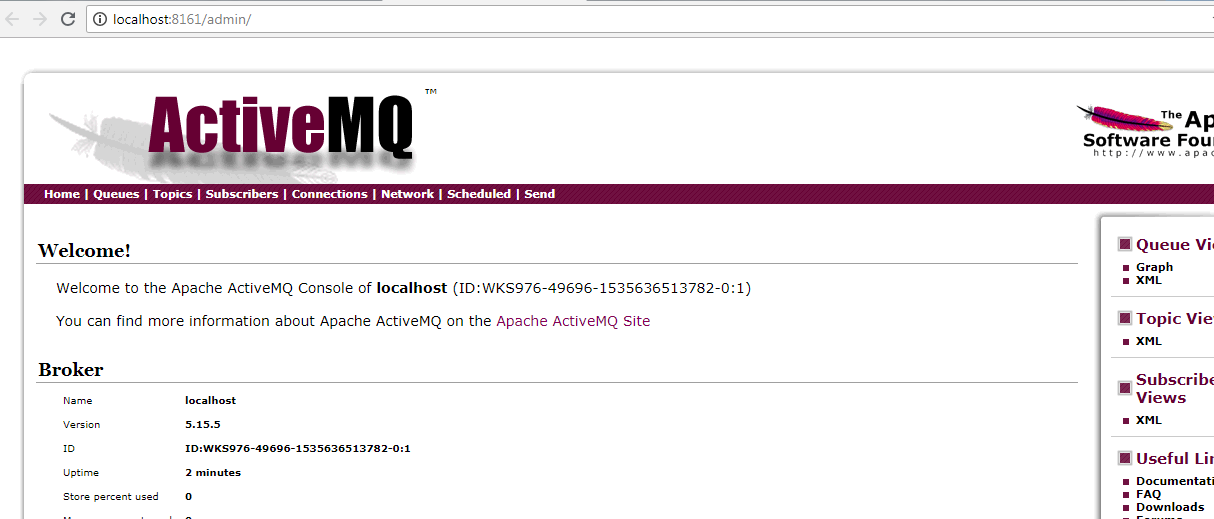
**netstat –an | find “61616”**



* Check your access to the administration page by visiting the following URL:

http://localhost:8161/admin/

Logon with the default user **admin/admin**



## Java application using the ActiveMQ Client

**Maven dependencies**

<dependency>

<groupId>org.apache.activemq</groupId>

<artifactId>activemq-broker</artifactId>

<version>5.15.4</version>

</dependency>

<dependency>

<groupId>org.apache.activemq</groupId>

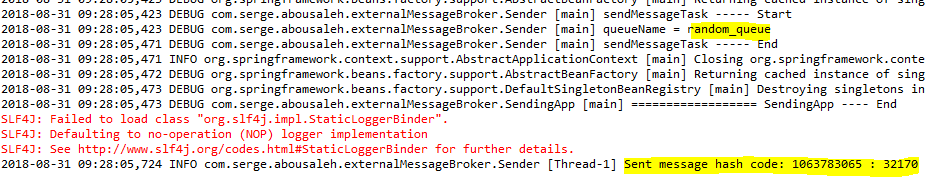
<artifactId>activemq-kahadb-store</artifactId>

<version>5.15.4</version>

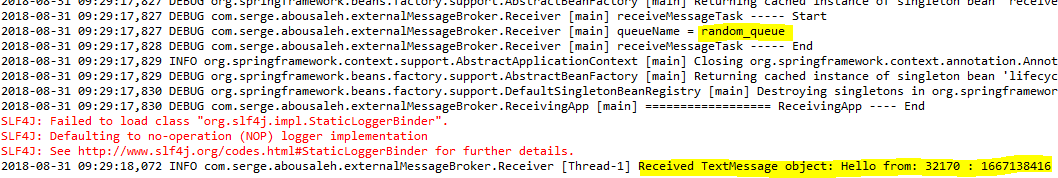
</dependency>

(Full code on <https://github.com/asergio13/activemq-simple-poc>)

**SendingApp**



**ReceivingApp**



Check the created queue by the Java Client on the **ActiveMQ Message Broker**

