

# Quick Start

## Kaazing WebSocket Gateway – JMS Edition

A STEP-BY-STEP GUIDE

This document takes you through the steps of using Kaazing WebSocket Gateway - JMS Edition (hereafter referred to as "the JMS Gateway" or more simply, "the Gateway") for first-time users.

The instructions are written for Microsoft Windows. Linux/Mac readers will need to adjust paths to Linux format. Other differences for Linux/Mac will be pointed out in the text.

## Getting Started

This section walks you through downloading, installing, and starting Kaazing JMS Gateway.

### Getting the Product

1. Download the JMS Gateway for your platform from <http://kaazing.com/download> and save it locally.

Make sure you download the Full version, not the Base. The Base version contains just the binaries needed for a production environment. The Full version contains everything, including documentation and demos.

2. Unzip the downloaded file to a new directory titled `c:\kaazing`, or on Linux/Mac untar the downloaded file to `/home/user/kaazing`. (This document assumes you will use `c:\kaazing` on Windows or `/home/user/kaazing` on Linux/Mac.) Inside `c:\kaazing` you should see two folders (although the version numbers you see may be different):

```
apache-activemq-5.3.0
```

```
kaazing-websocket-gateway-jms-3.2.5
```

If you don't see these folders because they ended up in another sub-directory, simply move them to `c:\kaazing`.

4. For convenience and ease of reading, rename `c:\kaazing\apache-activemq-n.n.n` to `c:\kaazing\activemq`.

This folder contains Apache ActiveMQ, a free open-source message broker with a JMS interface. The JMS Gateway can connect to any JMS message broker, but in case you don't have one you can use ActiveMQ. If you have your own JMS message broker already, you can connect to that.

5. For convenience and ease of reading, rename `c:\kaazing\kaazing-websocket-gateway-jms-n.n.n` to `c:\kaazing\gateway`.

This folder contains the files for Kaazing Gateway.

Your directory structure should now look like this:

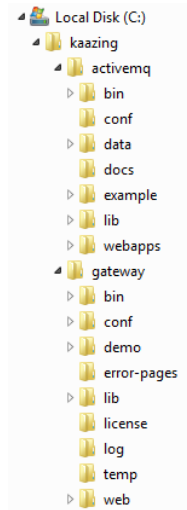


Figure 1: Directory Structure

## Starting the Gateway

- An overview of the architecture is shown in Figure 2. In the steps below you will start the JMS message broker first, and then you will start the Gateway, which connects to the JMS message broker. Finally, you'll use a web browser, which connects to the Gateway and runs a demo.

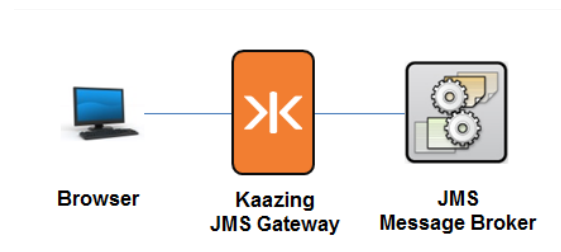


Figure 2: Architecture Overview

- Before starting the Gateway, you should start the JMS message broker that the Gateway is going to connect to. In these steps we'll use ActiveMQ, which is included in the Gateway download.  
Open a command line and navigate to `c:\kaazing\activemq\bin`.
- On Windows, run the following command from Command Prompt:  
`activemq.bat`  
On Linux/Mac, run the following command from the terminal:  
`./activemq`

Wait a few seconds for the screen output to stop, which means ActiveMQ is up and ready. As a test, you can open the ActiveMQ console in a browser to verify ActiveMQ is running: `http://0.0.0.0:8161/admin`.

Later when you want to stop ActiveMQ you can enter Ctrl-C from the command line.

9. Open a new command window and navigate to `c:\kaazing\gateway\bin`.

10. On Windows, run the following command from Command Prompt:

```
gateway.start.bat
```

On Linux/Mac, run the following command from the terminal:

```
./gateway.start
```

When you see the following line in the output you know the Gateway is running:

```
INFO Started server successfully in 0.775 secs at 2012-01-01
15:24:46
```

Here is an example output from the Gateway, followed by some explanations.

```
1: INFO Kaazing WebSocket Gateway - JMS Edition (3.2.5.3)
2: INFO Configuration file: /home/user/kaazing/gateway/conf/gateway-config.xml
3: INFO Checking license information
4: INFO Found license: Developer License, Maximum 50 connections, See LICENSE.txt
5: INFO Starting server
6: INFO Starting services
7: INFO http://localhost:8000/
8: INFO http://localhost:8001/
9: INFO http://localhost:8001/session
10: INFO ws://localhost:8001/echo
11: INFO ws://localhost:8001/interceptor
12: INFO ws://localhost:8001/jms
13: INFO ws://localhost:8001/stomp
14: INFO Started services
15: INFO Starting management
16: INFO jmx://localhost:2020/
17: INFO Started management
18: INFO Started server successfully in 0.775 secs at 2012-01-01 15:24:46
```

Here is an explanation of some of the output:

- Line 1: The product, edition, and version number of the Gateway. The version number consists of three numbers plus a build number. The first three numbers the major release, minor release, and patch number.
- Line 2: The configuration file used by the Gateway.
- Line 4: Shows the license and maximum number of connections this Gateway instance will accept.
- Line 7-13: The accept URLs for the various services configured for this Gateway instance.
- Line 18: Indicates the Gateway is started and ready to receive requests.
- Later when you want to stop the Gateway you can enter Ctrl-C from the command line.

## Connecting to the Gateway

11. Now that the JMS message broker and Gateway are running, you can test it. Using a browser, open <http://localhost:8000>. You will see a test page that indicates the Gateway is running.  
The default configuration uses port 8000 as the "base". The base directory is empty except for this one test page and is where you place your application files. For example, in a development environment you could build and test your application there, or in a production environment you would deploy your application there.
12. Since you downloaded the Full version of the Gateway, you also have documentation and demos. By default, these are available from port 8001. Using a browser, open <http://localhost:8001>. You will see the welcome page and links at the top for documentation and demos.
13. Click the **Demos** link at the top of the page.  
On the left-hand side you see a list of client technologies like JavaScript, Java, Silverlight, etc. Kaazing enables WebSocket not just using JavaScript in a browser, but in a variety of other environments too. So if you prefer to use WebSocket from a Flash application, for example, you can do that.

## Running a Basic JMS Demo

This section demonstrates subscribing to a JMS topic in order to send and receive messages.

1. Go to the demo page and click the **JavaScript** link to use the JavaScript demos. (The URL is <http://localhost:8001/demo/jms/javascript>.)
2. Click the **JMS Messaging Demo** link. This demo will let you perform some basic JMS operations.

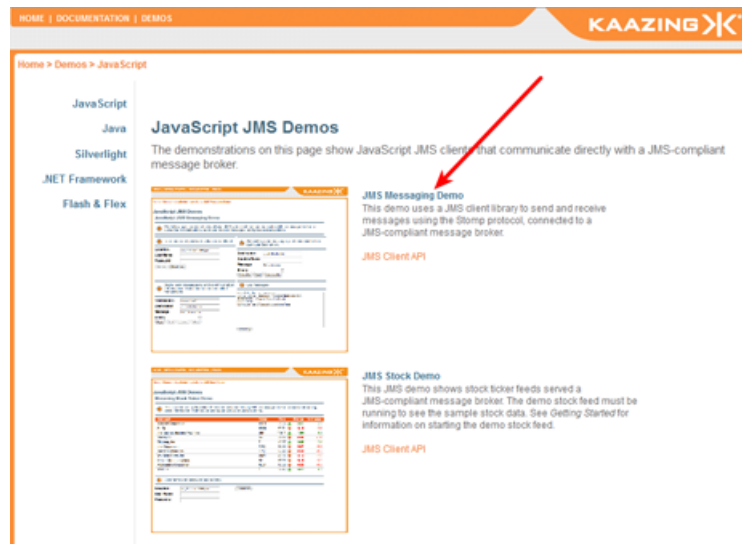


Figure 3: Starting the JMS Messaging Demo

3. Click the **Connect** button. This will establish a WebSocket connection to the given location: `ws://localhost:8001/jms`.

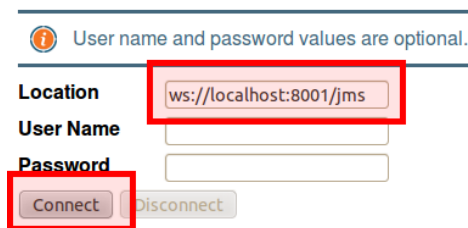


Figure 4: Architecture Overview

In the **Log Messages** box, you'll see that you're connected.

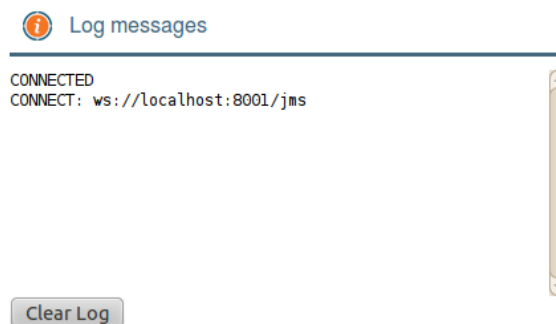


Figure 5: Log Message

- Click the **Subscribe** button. This will subscribe you to the destination in the **Destination** field, which is by default /topic/destination. In the **Log Messages** box you'll see that you're subscribed.

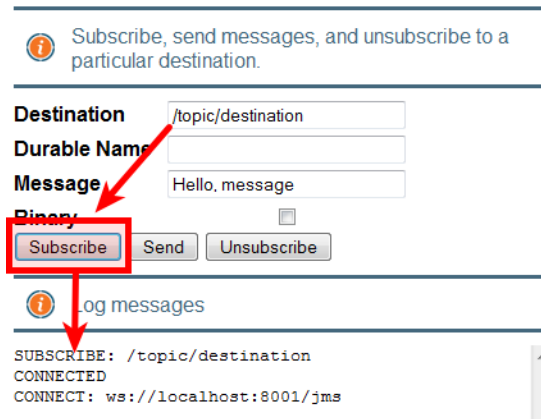


Figure 6: Subscribing to a Destination

- Open a second browser and arrange the two browsers side-by-side so you can see them both.
- In the second browser window, open the JavaScript JMS Demo (<http://localhost:8001/demo/jms/javascript/?d=jms-javascript>) and repeat the steps you just performed in step 3 and 4:
  - Click the **Connect** button to connect.
  - Click the **Subscribe** button to subscribe to the same topic.
- Back in the first browser window, type a message in the **Message** field and click **Send**:

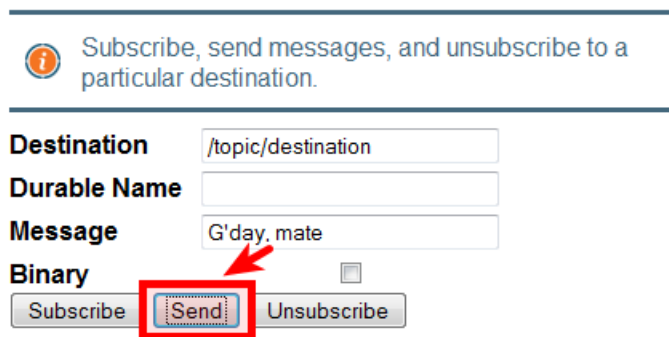


Figure 7: Sending a Message from Browser 1

- Look at the second browser window and you will see that the **Log messages** box is updated to show that subscriber received the message:

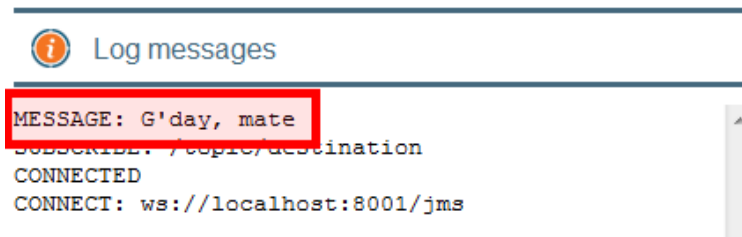


Figure 8: Receiving a Message in Browser 2

- Look in the **Log messages** box of the first (sending) browser. You will see two updates, one that a message was sent, but also that one was received. This is because this client is also subscribed to the topic. In other words, there are currently two clients subscribed to the topic, and they each received the message sent to that topic.

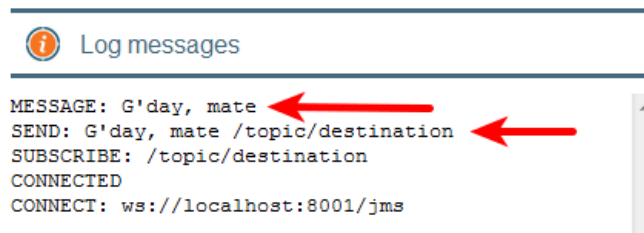


Figure 9: Sending and Receiving Messages in Browser 1

## Using a JMS Transaction

- In the first browser, in the transaction section on the lower-left side, click the **Begin** button:

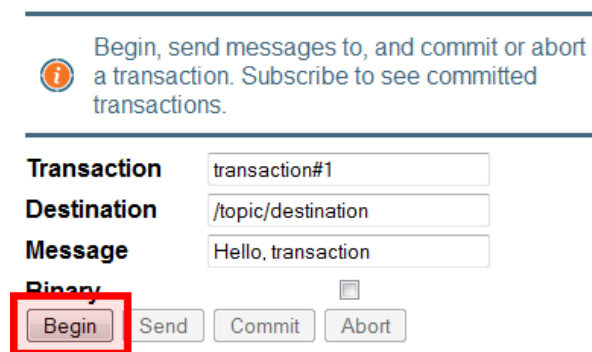



Figure 10: Starting a JMS Transaction in Browser 1

- In the **Message** field, type "Transaction 1", then click the **Send** button:


 Begin, send messages to, and commit or abort a transaction. Subscribe to see committed transactions.

---

**Transaction**   
**Destination**   
**Message**   
**Binary**

Figure 11: Entering a Message in Browser 1

- Look in the **Log messages** box in both the first browser. It indicates a message was sent, but no messages have been received:

 Log messages

---


```

SEND: Transaction 1 /topic/destination
BEGIN
SUBSCRIBE: /topic/destination
    
```

Figure 12: Sending a Transaction Message in Browser 1

Look in the second browser, and notice that no message has been received. No clients have received any messages because the transaction is still pending.

- Back in the first browser, enter "Transaction 2" in the **Message** field and press the Send:

 Begin, send messages to, and commit or abort a transaction. Subscribe to see committed transactions.

---

**Transaction**   
**Destination**   
**Message**   
**Binary**

Figure 13: Entering a Second Message in Browser 1

- Finally, enter "Transaction 3" in the **Message** field and press the **Send** button:

**i** Begin, send messages to, and commit or abort a transaction. Subscribe to see committed transactions.

---

**Transaction**

**Destination**

**Message**

**Binary**

Figure 14: Entering a Third Message in Browser 1

- At this point, no clients should have received any messages because the transaction is still open.

In the first browser, click the **Commit** button to commit the transaction:

**i** Begin, send messages to, and commit or abort a transaction. Subscribe to see committed transactions.

---

**Transaction**

**Destination**

**Message**

**Binary**

Figure 15: Committing the Transaction in Browser 1

- Look in the **Log messages** box in both browsers and see that all three messages were received:

**i** Log messages

---

```
MESSAGE: Transaction 3
MESSAGE: Transaction 2
MESSAGE: Transaction 1
MESSAGE: G'day, mate
SUBSCRIBE: /topic/destination
CONNECTED
CONNECT: ws://localhost:8001/jms
```

Figure 16: Received Transaction Messages in Browser 2

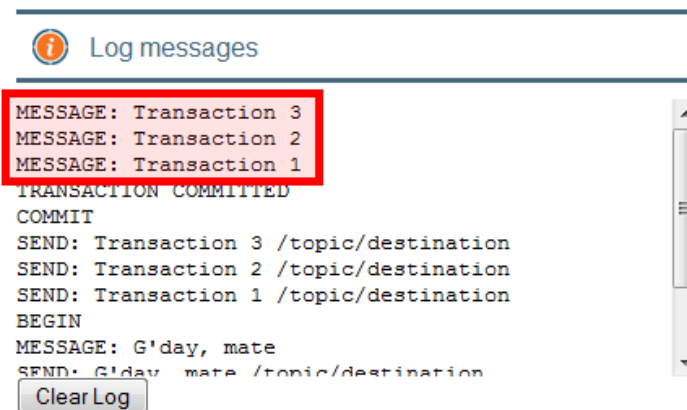


Figure 17: Received Transaction Messages in Browser 1

In this section you have seen how to send and receive JMS messages, including messages as part of a JMS transaction.

## Running the Stock Demo

This section walks you through how to run the stock demo.

The stock demo will list different stocks and their prices as they change in real time. Therefore, another process to pump data into the message broker needs to be started first. It is just a Java program that generates random stock prices and publishes them to a JMS topic. Clients can then subscribe to that topic and receive the stock prices.

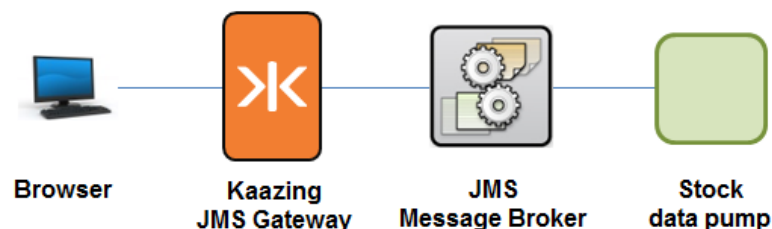


Figure 18: Architecture Including Stock Data Pump

1. Using a browser, navigate to the demo page:  
<http://localhost:8001/demo/jms/javascript>
2. Click **JMS Stock Demo**:

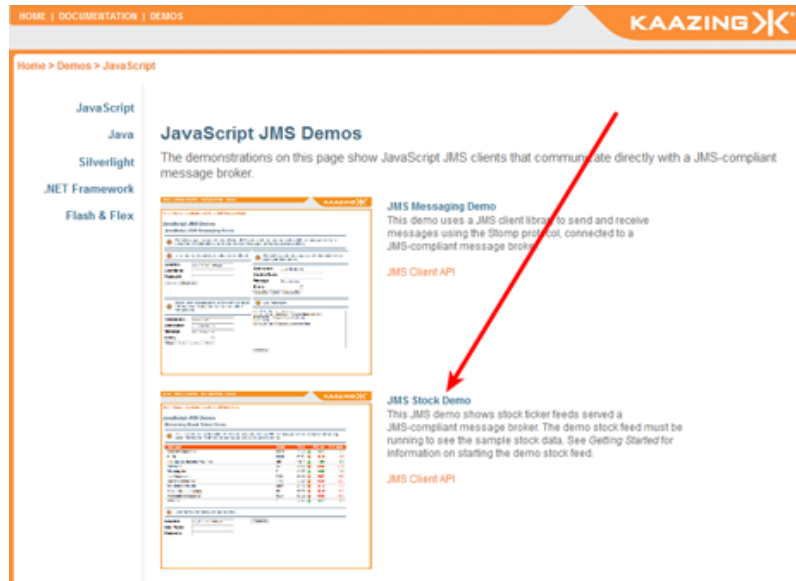


Figure 19: Starting the Stock Demo

- In the stock demo page, notice that the WebSocket URL is the same as it was in the JMS Messaging Demo (see Step 3 in the basic JMS section). This is because these demos connect to the same JMS message broker, even though it will use a different JMS subscription. Click the **Connect** button.

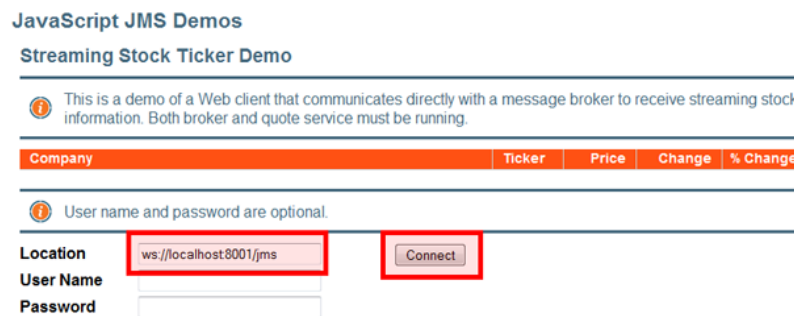


Figure 20: Connecting to the Stock Demo

- Notice that nothing happens. That's because the client has subscribed to the topic, but no producers are publishing messages to the topic.
- Open a new Command Prompt and navigate to `c:\kaazing\activemq\bin`. You might want to arrange the command line window and the browser so you can see them both at the same time.
- Run the following command:  
On Windows: `demo-services.start.bat`  
On Linux/Mac: `./demo-services.start`

This starts the Java process that generates stock prices and publishes them to a JMS topic.

Later when you want to stop this process you can press Ctrl-C from the command line.

7. Return to the browser and notice that prices are now being displayed because messages are being published to the topic.
8. You have now seen how the stock demo delivers messages rapidly to a browser or mobile client over the Web.

## Next Steps

- In addition to documentation being available with your Gateway installation, it is also available online: <http://tech.kaazing.com/documentation/jms>.
- The Setup Guide contains information about common setup tasks: <http://tech.kaazing.com/documentation/jms/setup-guide.html>. For example, if you would like to access your Gateway from a remote browser, you can follow the steps in the section [Setting Up Kaazing WebSocket Gateway on a Server](#) in the Setup Guide.
- Check out online demos at <http://demo.kaazing.com>.