

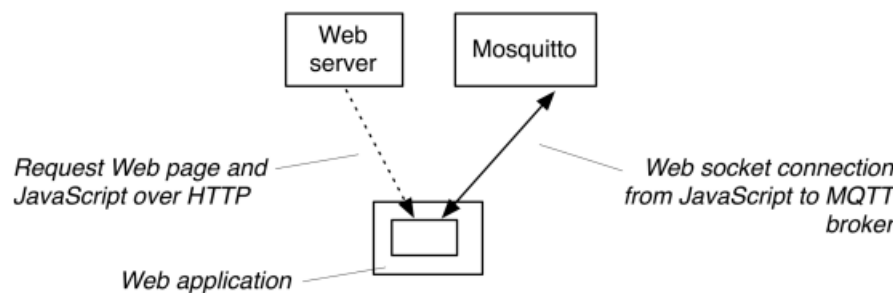
Jan-Piet Mens

+ About @jpmens Archives Pages Search  DNS

The Mosquitto MQTT broker gets Websockets support

When I wrote about the [HiveMQ](#) MQTT broker I said it had a killer-feature: support for Websockets. I'll admit to have been pestering [Roger Light](#), creator of [Mosquitto](#), so much that to shut me up (I think) he's been adding support for Websockets into the [Mosquitto broker](#)!

This makes [using MQTT](#) even more fun.



Websocket support must be explicitly enabled when building Mosquitto, and building requires [libwebsockets](#). Once enabled, we can add *listeners* with the websocket protocol. For example, this `mosquitto.conf` provides a default MQTT listener on port 1883, and two Websocket listeners, one with [TLS](#), one without:

```
autosave_interval 1800
persistence true
persistence_file m2.db
persistence_location /var/mosquitto/
connection_messages true
log_timestamp true

acl_file /etc/mosquitto/jp.acl
password_file /etc/mosquitto/jp.pw

listener 1883

listener 9001 127.0.0.1
protocol websockets

listener 9002 127.0.0.1
protocol websockets
cafile /etc/mosquitto/tls/ca.crt
certfile /etc/mosquitto/tls/tiggr.ww.mens.de.crt
keyfile /etc/mosquitto/tls/tiggr.ww.mens.de.key
```

(In case you're interested, I create the X.509 certificates for [TLS](#) using [generate-CA.sh](#) from the [OwnTracks](#) project.)

[Mosquitto](#) ACLs work for Websockets just as they do for MQTT publishes & subscribes; when accessing the broker via Websockets a different TCP transport channel is used -- everything else remains the same.

Using the [Paho MQTT JavaScript client](#), we can create something like this which shows messages published to the Mosquitto broker as soon as they're received.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Mosquitto Websockets</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<script src="mqttws31.js" type="text/javascript"></script>
<script src="jquery.min.js" type="text/javascript"></script>
<script src="config.js" type="text/javascript"></script>

<script type="text/javascript">
var mqtt;
var reconnectTimeout = 2000;

function MQTTconnect() {
  mqtt = new Messaging.Client(
    host,
    port,
    "web_" + parseInt(Math.random() * 100,
    10));

  var options = {
    timeout: 3,
    useSSL: useTLS,
    cleanSession: cleansession,
    onSuccess: onConnect,
    onFailure: function (message) {
      $('#status').val("Connection failed: " + message.errorMessage + "Retrying");
      setTimeout(MQTTconnect, reconnectTimeout);
    }
  };

  mqtt.onConnectionLost = onConnectionLost;
  mqtt.onMessageArrived = onMessageArrived;

  if (username != null) {
    options.userName = username;
    options.password = password;
  }
  console.log("Host=" + host + ", port=" + port + " TLS = " + useTLS + " username=" + username + " password=" + password);
  mqtt.connect(options);
}

function onConnect() {
  $('#status').val('Connected to ' + host + ':' + port);
  // Connection succeeded; subscribe to our topic
  mqtt.subscribe(topic, {qos: 0});
  $('#topic').val(topic);
}

function onConnectionLost(response) {
  setTimeout(MQTTconnect, reconnectTimeout);
  $('#status').val("connection lost: " + responseObject.errorMessage + ". Reconnecting");
}

function onMessageArrived(message) {

  var topic = message.destinationName;
  var payload = message.payloadString;

  $('#ws').prepend('<li>' + topic + ' = ' + payload + '</li>');
};

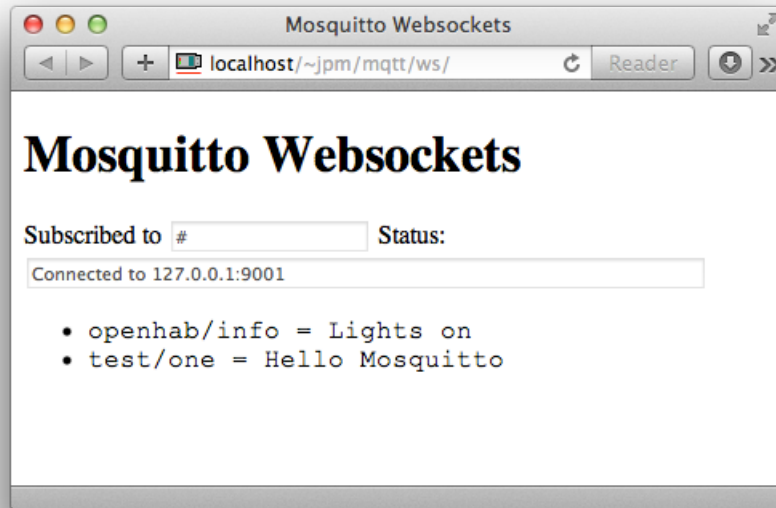
$(document).ready(function() {
  MQTTconnect();
});

</script>
</head>
<body>
<h1>Mosquitto Websockets</h1>
<div>
```

```
<div>Subscribed to <input type='text' id='topic' disabled />
Status: <input type='text' id='status' size="80" disabled /></div>

<ul id='ws' style="font-family: 'Courier New', Courier, monospace;"></ul>
</div>
</body>
</html>
```

I've put the source to this simple example [on Github](#), and I've verified it works with Safari, Chrome, and FireFox.



Instead of writing your own Web application, you can easily use the [mqtt.io](#) or the [HiveMQ Websocket](#) clients; configure them to use the host name and TCP port number (from the listener directive) of your Mosquitto broker.

Mosquitto with Websocket support should be with you very soon, and I hope you'll agree with me, that this is very good news. :-)



[<Earlier](#) [Later>](#)

Comments

MQTT :: 03 Jul 2014 :: [e-mail](#)

52 Comments

Jan-Piet Mens

 Login Recommend Share

Sort by Oldest



Join the discussion...

**joe_c** • 2 years ago

Ha, I was just looking for this and thought I could maybe visit your blog. Seems I've made it just in time :-)

  • Reply • Share**James Mills** • 2 years ago

Ahh this is fantastic! :)

  • Reply • Share**CZECHer** • 2 years ago

Fantastic news, thank you very much! Is it actually possible to response to MQTT messages via this websocket connection?

I want to send question dialogs to the browser via this connection and publish the answers to another topic.

  • Reply • Share**Jan-Piet Mens** Mod  CZECHer • 2 years ago

Of course, yes. Publish and subscribe both work.

  • Reply • Share**CZECHer**  Jan-Piet Mens • 2 years ago

Thank you for your fast response. One more question: Where will the rendering of the webside take place when I am sending XML data to the MQTT Broker and access the messages in the browser via the mentioned websocket connection? It seems to be on client side. Is there a way to do it server/broker side?

  • Reply • Share**Jan-Piet Mens** Mod  CZECHer • 2 years ago

No idea

  • Reply • Share**sebastian** • 2 years ago

Sounds great. But which version do I need to install on my raspi to make this work? Tried installation as you described in one of your previous posts. But version 1.3.1 seems to not support websockets protocol (Error: Unknown configuration variable "protocol").

  • Reply • Share**Jan-Piet Mens** Mod  sebastian • 2 years ago

I said it wasn't published yet. This is 1.4 from bitbucket with a modified libsockets version.

  • Reply • Share**Sebatsian**  Jan-Piet Mens • 2 years ago

Ah, I see. Thanks for the fast response! Looking forward to release...

  • Reply • Share**Jeremy Gresh** • 2 years ago

