

jabsocket, XMPP over WebSocket connection manager

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME
0.1.0	2013-07-24		AJ

Contents

1	Introduction	1
2	Building and Installing	2
3	Configuration	3
4	Running	4

Abstract

This document describes the purpose, architecture and usage of jabsocket, XMPP-over-WebSocket connection manager.

jabsocket is a server program for connecting with XMPP servers via WebSocket protocol.

1 Introduction

XMPP is Extensible Messaging and Presence Protocol, originally known as Jabber. Jabber was designed for instant messaging, exchanging presence information, and managing contact lists. It was developed as an open and extensible protocol. These features together with the fact that the XMPP architecture is decentralized, led to the standard being expanded for use in many applications beyond the instant messaging.

In XMPP, the main components are clients and servers. Clients communicate with servers using client-to-server protocol and servers communicate with other servers using server-to-server protocol. Both of these protocols are based on XML format and use TCP protocol as the transport protocol.

An XMPP client can be implemented in many ways, but in this project we are concerned with only one particular type of implementation: as a web browser-based application written in JavaScript. This approach has upsides and downsides. The upsides are the hardware and Operating System independence due to the ubiquity of web browsers and the fact that it does not require the user to install any software other than a web browser.

However, the browser-based approach has a problem that HTTP, the protocol for communication between the browser and the web server, was designed for synchronous communication: the browser sends a request and the server sends back a response. XMPP is by its nature an asynchronous protocol, where both the client and the server can send data at any time.

Before WebSocket protocol came, the only way this problem could be solved was by using a protocol called Bidirectional streams Over Synchronous HTTP (BOSH), which is similar to the techniques known as Comet. This approach has a number of issues, all stemming from the fact that HTTP was not designed for asynchronous, bidirectional communication. The main problem is a large overhead in HTTP request and response. See RFC 6202 for a discussion of all issues with BOSH.

WebSocket was designed to solve the problems that HTTP has with network communication. After going through a number of incompatible revisions and fixes for discovered security problems, WebSocket was standardised in [RFC 6455](#) in 2011. Please refer the RFC and [Wikipedia page on WebSocket](#) for more details. Here we will only mention that the standard has two parts: the protocol (defining the handshake and message format) and the API for use from JavaScript.

In the parlance of RFC 6455, XMPP over WebSocket is a sub-protocol. As of July 2013, it is not yet standardised, but there is a draft, currently in its fourth iteration: [An XMPP Sub-protocol for WebSocket](#).

There are two ways to implement XMPP over WebSocket:

- The XMPP server can support WebSocket directly
- There can be a connection manager that server as an intermediary between a browser and an XMPP server.

jabsocket implements the second way, a connection manager. It aims to follow the XMPP-over-WebSocket draft.

jabsocket is not the only implementation of a connection manager (or "gateway"). I am aware of these:

- [node-xmpp-bosh](#), An XMPP BOSH & WebSocket server (connection manager) written on node.js using Javascript
- [WXG](#), a gateway for XMPP over WebSocket implemented in Java
- [mod_xmpp](#), Apache module which proxies XMPP client connections over WebSockets



Warning

At the moment (July 24 2013), jabsocket is still in the alpha stage, which means not yet production-ready. See TODO for the things I am planning to work on.

2 Building and Installing

jabsocket is implemented in C. Before building it, you have to have the following libraries installed:

- libevent 2.0.21 or later
- Expat 2.1.0 or later
- libyaml 0.1.3-1 or later
- libssl 0.9.8 or later

Note

The code may work with earlier versions of these libraries; the version numbers listed are simply what I have tested with on Ubuntu 10.04.3 LTS.

The build system uses CMake. The steps to build jabsocket are the following.

Get the latest source by running

```
git clone git@github.com:vogonsoft/jabsocket.git
```

Run cmake to generate the Makefile:

You first have to use CMake to generate the Makefile. To use the default configuration (which is release):

```
cmake .
```

To generate Makefile for debug:

```
cmake -D CMAKE_BUILD_TYPE=DEBUG .
```

To generate Makefile for release

```
cmake -D CMAKE_BUILD_TYPE=RELEASE .
```

Build:

```
make
```

To run make verbosely, run:

```
make VERBOSE=1
```

Run unit tests:

```
make check
```

Install:

```
sudo make install
```

Useful make targets:

- make help - list all available make targets
 - make package - build binary package
 - make package_source - build source package
-

3 Configuration

jabsocket reads configuration from a YAML-formatted file. By default, this file is

```
/etc/jabsocket.conf
```

but it can be changed with `-c` command line parameter, like this:

```
jabsocket -c ./jabsocket.conf
```

Here is an example configuration:

```
# jabsocket configuration

# Listening port for WebSocket connection
port: 5000

# CIDR to listen on (0.0.0.0 for all)
listen: 0.0.0.0

# Host - the value of the Host key in the request must match
# this parameter
host: example.com:5000

# Resource that is expected in the first line
# e.g. in the line GET /mychat HTTP/1.1 the resource is /mychat
resource: /mychat

# Acceptable origin
origin:
  - http://firstdomain.com
  - http://seconddomain.com

# Minimum log level
log_level: LOG_DEBUG
```

jabsocket currently recognizes the following configuration parameters:

- `port` - the port on which it listens for WebSocket connections
- `listen` - the CIDR on which to listen, for example 0.0.0.0 would mean listen on all available interfaces, 127.0.0.1 would mean localhost
- `host` - the required value of the Host key in the opening handshake; if this parameter does not exist in the configuration file, then we don't check
- `resource` - the required value of resource in the request line, e.g. if the request line is GET /mychat HTTP/1.1, the resource is /mychat; if this parameter does not exist in the configuration file, then we don't check
- `origin` - the list of web origins from which the service will accept the connection
- `log_level` - minimum log level - one of the standard syslog levels: LOG_EMERG is the highest and LOG_DEBUG is the lowest

Web origins listed under "origin" key are patterns that define which origins will be accepted. In the opening handshake of a WebSocket connection, the browser will send a header that looks like this:

Origin: `http://vogonsoft.com`

In that string, "http" is scheme, "vogonsoft.com" is the host, and port is not specified. If you want your server to accept only this origin, put this in the configuration:

```
origin:
- http://vogonsoft.com
```

If you want to accept vogonsoft.com and all its subdomains, use the pattern:

```
origin:
- http://vogonsoft.com
- http://*.vogonsoft.com
```

The patterns in the list of origins have the same syntax as file patterns: "*" matches any sequence of 0 or more characters, and "?" matches any single character.

Recognized log levels, from highest to lowest:

- LOG_EMERG
- LOG_ALERT
- LOG_CRIT
- LOG_ERR
- LOG_WARNING
- LOG_NOTICE
- LOG_INFO
- LOG_DEBUG

When diagnosing a problem, set the level to LOG_DEBUG.

Note

When you change the configuration, you have to restart the service for it to read the new configuration parameters.

4 Running

jabsocket accepts the following command-line parameters or options:

<code>--config, -c <config file></code>	- Configuration file (default is <code>/etc/jabsocket.conf</code>)
<code>--nodaemon, -n</code>	- Don't run as daemon (default is running as daemon)
<code>--version, -v</code>	- Display the version
<code>--help, -h</code>	- Help message