

WICKRIO NODE.JS ADDON INTERFACE

Prepared for: Wickr Inc, Engineering Group

Prepared by: Paul Cushman, Project Lead

Date: July 10, 2018

Version: 4.41.15

# USING WICKRIO WITH NODE.JS

## Objective

This document contains information associated with the installation, configuration, and use of the Node.js interface to the WickrIO client architecture. This interface allows Node.js programs access to the WickrIO API via a Node.js addon. The Node.js addon provides the same set of capabilities as the RESTful API interface.

## Assumptions

This document is intended for Systems Administrators and developers that have a working knowledge of Linux server administration, package installation, API usage, and scripting knowledge.

Wickr staff is available to assist in the deployment and configuration, but for security reasons, at no time should Wickr have access to the actual machine where the deploy is taking place. Screen sharing sessions can be used for troubleshooting.

### Document Overview

The following is a list of sections in this document:

* [Solution Overview](#_SOLUTION_OVERVIEW)
* [Software Installation](#_SOFTWARE_INSTALLATION)
* [Configuration](#_CONFIGURATION)
* [Node.js Addon API Description](#_REST_API_DESCRIPTION)
* [Broadcast Feature Examples](#_BROADCAST_FEATURE_EXAMPLES)
* [Troubleshooting](#_TROUBLESHOOTING)

# SOLUTION OVERVIEW

## WickrIO Components

The components of the WickrIO Node.js addon solution include a WickrIO Docker container, and Node.js sample software. You will have to supply an appropriate host machine to run the WickrIO Docker container and any Node.js software you develop. Included are several Node.js sample solutions that use the WickrIO Node.js addon.

### Host Machine

The WickrIO Docker container should run on any capable host machine. At the time this docuent was written testing was only performed using Ubuntu 16.04 host machines. The WickrIO client software will require access to the file system running on the host machine so that persistent data can be stored.

### WickrIO Docker Container

The WickrIO Docker container contains the WickrIO client(s), WickrIO client service and configuration software. The WickrIO clients are Wickr clients that provide software interfaces to the Wickr client capabilities, instead of via a graphical user interface (GUI). The interface to the Wickr features is through the Node.js addon. This interface provides the ability to send and receive messages, as well as create secure rooms and group conversations. Details of the Messaging API is described in this document.

WARNING: The WickrIO clients implemented in the Docker container are Wickr Bot clients. Visualization of the Wickr Bot clients in the Wickr Clients is under design and development.

### Node.js Scripts

The Node.js scripts are scripts or any software that are implemented by users of the WickrIO Node.js addon interface. Sample scripts using the WickrIO Node.js addon are provided at the end of this document.

# SOFTWARE INSTALLATION

## WickrIO Node.js Installation Components

There are basically three components to installing the WickrIO software:

### Host Machine

The host machine must be capable of running Docker containers. You will have to install appropriate Docker software before you can install the WickrIO Docker Container.

TBD: Add details of setting up Docker and an appropriate Docker account

### WickrIO Docker Container

The WickrIO Docker container includes all of the WickrIO software necessary to configure, run and maintain the WickrIO client(s).

## WickrIO Installation Steps

The following is an example of the components that will be included in the WickrIO distribution, version numbers may change:

* Wickr/bot-cloud-alpha:4.41.15.03
* WickrIO\_node.js\_samples.zip

### Host Machine Setup

The host machine used to run the WickrIO Docker container needs to be setup to be running the appropriate Docker software.   
  
The WickrIO Docker container will also require access to the host file system. Persistent data will be stored in a location on the host’s file system.

The host machine should be secure so that it is impossible for people to tamper with the system.

### WickrIO Docker Container

The WickrIO Docker container can be pulled down to the host machine using the following command (note: the version number may change):

docker pull wickr/bot-cloud-alpha:4.41.15.03

The WickrIO Docker contain includes all of the necessary software to run the WickrIO client(s).

### Node.js addon Software

The Node.js addon software contains the Wickr Node.js addon as well as several samples that can be used to access the WickrIO Node.js interface. These samples can be used as a base to develop your own Wickr applications.

Extract the software to a directory on your host machine.

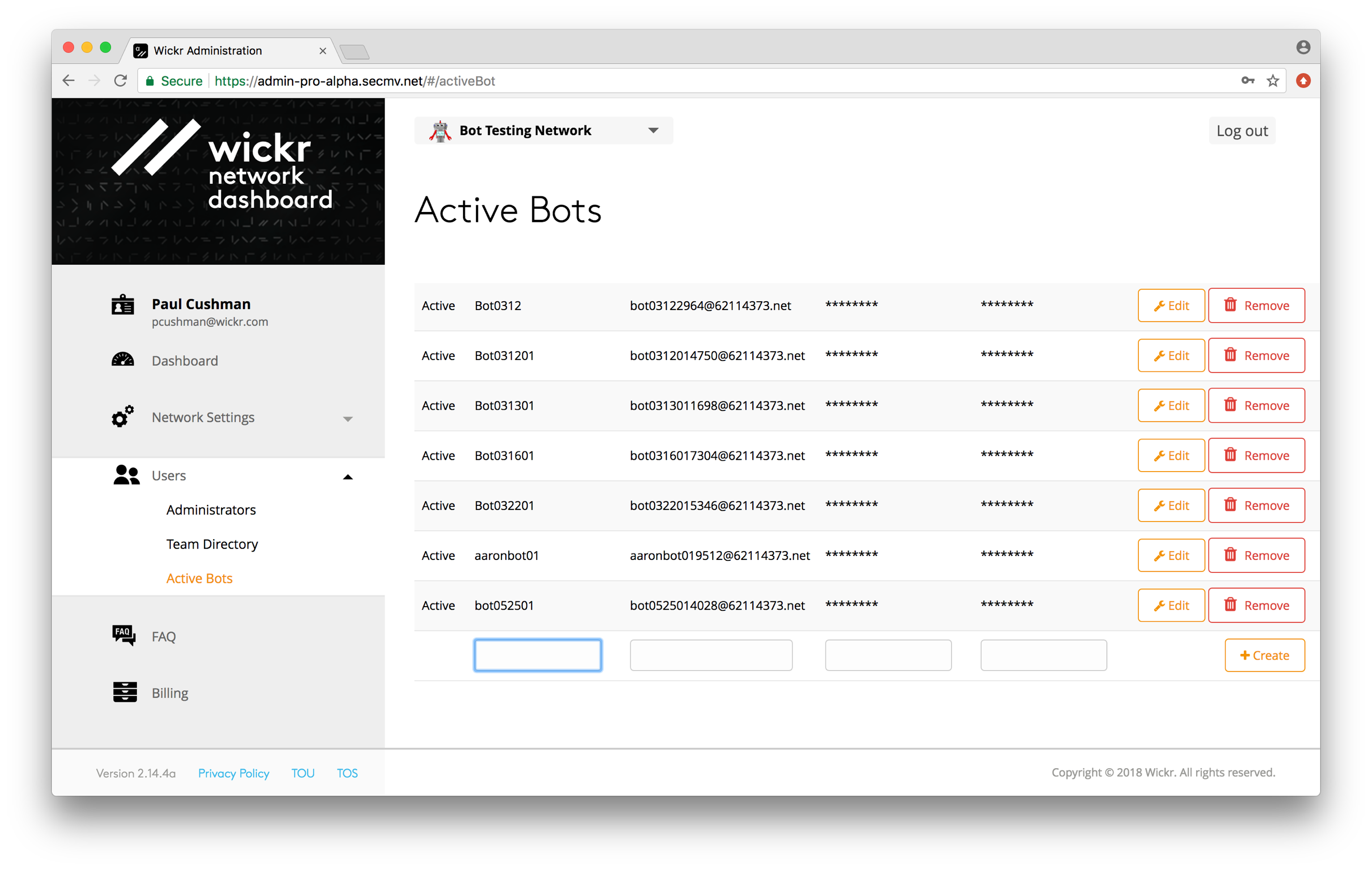
# CONFIGURATION

## WickrIO Configuration Steps

Once the software has been installed there are several things that need to be configured before the WickrIO clients can be used.

### WickrIO Client Creation

The WickrIO Client(s) will need to be created and configured. WickrIO clients are actually Wickr Bot accounts. You will need to create these accounts using the Active Bots screen of the Wickr Admin Console, see below:



When adding a Wickr Bot to the Active Bots screen you will input the Bot display name and the password fields. The console will generate the Bot Username. You will need the Bot username and password fields to configure the Wickr Bot.

## Host Machine Setup

The host machine must have sufficient disk space to support the WickrIO clients running within the WickrIO Docker container.

TBD: Add disk sizing information

The WickrIO clients will need to save persistent data to a location on the host machine. Normally this is located in the /opt/WickrIODebug directory. When running the WickrIO Docker container you will identify where this is located on the host machine, for example:

docker run -v /opt/WickrIODebug:/opt/WickrIODebug -ti wickr/bot-cloud-alpha:4.41.15.03

In this example the host has a directory named /opt/WickrIODebug that is where the WickrIO client(s) running in the Docker container will store persistent data.

## WickrIO Configuration Steps

All the configuration steps for WickrIO are performed from the command interface that is presented when you run the WickrIO Docker container. To start the WickrIO Docker container the following command should be entered, replace the version number with the appropriate version number:

docker run -v /opt/WickrIODebug:/opt/WickrIODebug -ti wickr/bot-cloud-alpha:4.41.15.03

When you run the WickrIO Docker container for the first time there will be no WickrIO clients configured, you will see the following output:

There are no clients currently configured!

Enter command:

When you enter the WickrIO Docker container, after having configured one or more WickrIO clients, the list of currently configured WickrIO clients will be displayed. You will then be prompted to enter a command. You will have the choice of entering the following commands:

* add: add a new client
* delete: deleting a client
* list: see a list of currently created clients
* modify: modify the settings of a client
* pause: pause a running client
* start: start a client
* quit: exits the docker container
* help or ‘?’: display the list of commands

To configure a new WickrIO client run the “add” command. You will be prompted for two fields, the Wickr username and the password. The username is the bot user name generated on the Wickr Admin console, and the password is the password entered on the Wickr Admin console. The “add” command will then provision this WickrIO client and create the appropriate data structures.

Once the client is created you can start the client by running the “start” command with the appropriate index number that you can get from running the “list” command. The “list” command will also display the current status of each WickrIO client.

# NODE.JS ADDON DESCRIPTION

This section will describe the WickrIO Node.js addon and how to use it. There are several samples provided that show how the addon is used.

The WickrIO Node.js addon interface supports a finite set of functions that you can access via your javascript code.

* cmdGetStatistics
* cmdClearStatistics
* cmdGetRooms
* cmdAddRoom
* cmdModifyRoom
* cmdGetRoom
* cmdLeaveRoom
* cmdDeleteRoom
* cmdAddGroupConvo
* cmdDeleteGroupConvo
* cmdGetGroupConvo
* cmdGetGroupConvos
* cmdGetReceivedMessage
* cmdSend1to1Message
* cmdSendRoomMessage

There are several functions that are used to setup and shutdown the addon interface to the WickrIO Client:

* clientInit
* closeClient

Before accessing any of the addon interface function you will need to run the clientInit function with the WickriO client name.

# TROUBLESHOOTING

## Troubleshooting WickrIO Components

This section will describe some possible issues you may run into while using the WickrIO client and the associated services.

### WickrIO Client does not start

First you need to determine if the WickrIO client is running or not. To do so you can run the ps command to see that the client process is running. The following should return an entry for each WickrIO client that is running:

ps -aef | grep wickrio\_bot

If nothing is returned from that command, then verify that the background service is running. Using ps, the following command should return an entry for the background service:

ps -aef | grep WickrIOSvr

If nothing is returned then use the WickrIOConsoleCmd to start the background service, this is described in a section above. If you cannot get the background service to start, then contact Wickr support.

If the background service process is running but there is no process running for the WickrIO client, then use the WickrIOConsoleCmd program to help diagnose the problem.

1. At the top level of the WickrIOConsoleCmd program enter the “client” command.
2. Enter the “list” command to see the list of WickrIO clients.
   1. If the WickrIO client’s state is “Paused”, then use the “start” command to start the client.
   2. If the WickrIO client’s state is “Running”, and there was no associated process running, then check the output file for the background services (described later) to see if the background service is having a problem starting the client. If it looks like the service is not trying to start the client, then restarting the background service should fix the problem. To do so, go to the “server” section of the WickrIOConsoleCmd program and stop and start the server. If this does not fix the issue the contact Wickr support for further assistance.
   3. If the WickrIO client’s state is “Down”, this is typically related to the background service not running. As per the previous step, go to the “server” section of the WickrIOConsoleCmd program and stop and start the server. If this does not fix the issue the contact Wickr support for further assistance.

Further diagnosis of problems with the WickrIO client or background service should be done with the help of the Wickr support team.

### Log and Output Files

The WickrIO Client and the background service will generate log and output files that can be used to determine possible issues. These files should be sent to Wickr Support to allow them to diagnose any issues that cannot be easily fixed. Output and log files will only be allowed to reach a certain file size. Once that size is reached a new file will be created. The maximum number of files saved to disk should only be 5.

The WickrIO Client log and output files are located in the following location:

/opt/WickrIO/clients/<client name>/logs

There are several files found in that directory. The file with the “.output” extension contains the most information and is useful in diagnosing issues with the WickrIO client.

The background service will write log and output files to the following location:

/opt/WickrIO/logs

The name of the background service files will start with WickrIOSvr. These files will also be limited in size and number of saved files. The background service output file can be used to diagnose any possible issues with starting a WickrIO client.