

Nathaniel “Eeems” van Diepen

OmnomIRC3

SMF integrated chat client and server application. IRC Integration.

Contents

[Contents](#)

[Abstract](#)

[Server](#)

[Technologies](#)

[Settings](#)

[API](#)

[Client](#)

[Technologies](#)

[API](#)

[OmnomIRC](#)

[Tab](#)

[Hooks](#)

[Plugins](#)

[API](#)

[Theming](#)

[API](#)

[CSS](#)

[Package manager](#)

[Abstract](#)

[Usage](#)

[Install](#)

[Remove](#)

[Update](#)

[Stats](#)

[Search](#)

Abstract

SMF integrated chat client and server application. IRC Integration.

Server

Technologies

- Node.js
 - http://nodejs.org/api/cluster.html#cluster_cluster
 - <https://hacks.mozilla.org/2013/01/building-a-node-js-server-that-wont-melt-a-node-js-holiday-season-part-5/>
 - <http://nodejsdb.org/>
- Socket.io
- Redis server

Settings

Server settings are stored in a json file called options.json. By default the file doesn't exist and the server uses the built in defaults. If you create the file and add only the options you want to change, they will override the defaults.

- port - port to run the server on.
- loglevel - amount of detail to log. 0 - error, 1 - warn, 2 - info, 3 - debug.
- threads - number of threads to spawn. Defaults to the number of cores the server has. Can't go lower than 1.
- redis - redis sub-settings.
 - port - port the redis server runs on.
 - host - hostname of the redis server.
- debug - if set to true the debug console is enabled on the server.
- paths - folder and file path sub-settings.
 - www - path to the folder containing the web files.
 - api - path to the folder containing the api script files.
 - plugins - path to the folder containing the server plugins.
- irc - IRC settings.
 - host - hostname of the irc server.
 - port - port of the irc server.
 - nick - nick for the bot to use.
 - name - realname for the bot.
 - channels - array of channels to join.
 - message - event messages to use.
 - quit - Message to use when quitting.

API

Client

Technologies

- Socket.io
- jQuery
- HTML5Shiv
- Console-shim

API

OmnomIRC

- version();
 - returns the version string
- register
 - command(name,function,help)
 - name - command string to use (eg: 'me' for the /me command)
 - function(args) - function to run when command is called
 - help - help string to be displayed when /help <cmd> is called
 - theme(name)
 - name - name of theme folder
 - plugin(name);
 - name - name of plugin
 - setting(name,type,default,validate,values,callback);
 - name - name of the setting
 - type - data type that the setting is.
 - default - default value to use for this setting
 - validate - function to run when the setting has changed. Return a boolean for validation state.
 - old - old value of setting
 - new - value setting is going to be changed to
 - values - array of values that this setting is set to default to
 - setting - name of the setting
 - callback - function to run after value has changed
 - value - value of setting
 - setting - name of setting
 - render - if rendering should happen during this setting change
 - hook(event,function)
 - event - name of the event to hook into
 - function - function to run when the event happens
- plugin
 - register
 - see OmnomIRC.register.plugin.

- start
 - name - name of plugin to start.
- stop
 - name - name of plugin to stop.
- remove
 - name - name of plugin to stop and remove.
- dir
 - name - name of the plugin to get the relative path to.
- hook(event,function)
 - see OmnomIRC.register.hook.
- ui
 - render
 - settings();
 - render the settings pane
 - users();
 - render the userlist
 - tab();
 - render the currently selected tab's body
 - tablist();
 - render the tab list
 - tabs
 - add(name);
 - name - name of tab to add
 - remove(name);
 - name - name of tab to remove
 - selected();
 - returns tab object of currently selected tab
 - idForName(name);
 - name - name of tab to get id for
 - tab(id_or_name);
 - id_or_name - id or name of tab to get tab object for
 - dom(id_or_name);
 - id_or_name - id or name of tab to get the dom for.
 - obj(id_or_name);
 - id_or_name - id or name of tab to get tablist object for.
 - select(id_or_name);
 - id_or_name - id or name of tab to select.
 - current();
 - returns the current tab object.
 - chat
 - connect(server);
 - server - server to connect to. Defaults to configured server
 - disconnect();

- disconnect from the current connection
 - connected();
 - returns state of connection
 - send(message,room);
 - message - message to send
 - room - room to send message to. Defaults to selected tab
 - auth();
 - send authorization request.
- get(setting);
 - setting - setting to get value of.
- set(setting,value);
 - setting - setting to change.
 - value - new value for setting.
- prop(name);
 - name - property to get the value of.
- send(message);
 - message - message to send to the current channel.
- event(event_name,message);
 - event_name - name of event to log
 - message - message to log
- fire(hook_name);
 - hook_name - name of hook to run

Tab

- name
 - name of the tab's channel
- body
 - Tabs document fragment. Used to render the tab.
- date
 - last time the body was updated.
- send(message);
 - message - message to send to the tabs room.
- close();
 - close the tab and disconnect from the room.
- users
 - array containing a list of users in the room.
- names();
 - refreshes the user list.
- select();
 - selects this tab.
- clear();
 - clears the body on the tab.

Hooks

- message - runs when a message is received.
 - message - string containing the message.
 - from - the name of the user who sent it.
 - room - the room that it was sent to.
 - origin - origin ID for the message.
- theme - runs when a new theme is registered.
 - name - name of the theme.
- untheme - runs when a theme is changed.
 - old - theme changing from.
 - new - theme changing to.
- themechange - runs right after the theme was changed.
 - name - name of the theme.
- join - runs when a user joins a room.
 - name - nick of the user who joined the room.
 - room - name of the room.
- part - runs when a user leaves a room.
 - name - nick of the user who left the room.
 - room - name of the room.
- connect - runs when a connection is made.
- disconnect - runs when connection is lost.
- reconnect - runs when a connection is re-established automatically.
- setting - runs when a setting changes. Return true or false based on if the setting change is valid.
 - setting - name of the setting that is changing.
 - old - old value of the setting.
 - new - new value of the setting.
 - values - array of values that this setting is set to default to.
- send - runs when a message is sent by the user. Return true or false if you want to cancel the send.
 - msg - the message sent by the user.
 - room - the room it is being sent to.
- start - runs when a plugin starts.
 - name - name of the plugin that started.
- stop - runs when a plugin stops.
 - name - name of the plugin that stopped.
- tabswitch - runs right before a tab is switched.
 - newTab - The new tab object.
 - oldTab - The old tab object.
- load - hook to run after the page has completed loading.
- authorized - hook to run after the user has authorized with the server.
- addtab - runs after a tab is added.

- tab - the tab object.
- removetab - runs right before a tab is removed.
 - tab - the tab object.

Plugins

API

Plugins treat \$o as the global variable, so instead of having to run \$o.register.plugin(), you will be able to run register.plugin(). Window will not be directly available to you, along with document, but you will be able to access \$ and jQuery.

hooks

Hooks are a little different for plugins, because when the plugin is stopped they are removed. The start and the stop hooks also act differently. They will only run when your plugin is stopped or started.

Theming

API

- load(function);
 - function - function to run when loading the theme
- unload(function);
 - function - function to run when unloading the theme

CSS

- **#head**
 - **#info**
 - **#title**
 - **#topic**
 - **#tabs-list**
 - **div.tab, div.tab.clicked**
 - **span.close-button**
 - **#tabs-scroll-left, #tabs-scroll-right.disabled**
 - **#tabs-scroll-right, #tabs-scroll-right.disabled**
- **#content**
 - **#content-list**
 - **li.row**
 - **span.cell**
 - **span.date_cell.cell**
- **#entry**
 - **#input**
 - **#send**
 - **#send.clicked**
 - **#users, #users.open**

- li
 - span.unselectable
 - span.close-button
 - #users-list
- #settings, #settings.open
 - span.unselectable
 - span.close-button
 - #settings-list
 - li.row
 - span.cell

Package manager

Abstract

OmnomIRC3 Package Manager (opm) is a package manager that allows you to install, remove and update plugins, themes and api extensions for OmnomIRC3 from the command line.

Usage

`./opm <command>[ <arguments>]`

Install

`./opm install <type> <name>`

install a package managed by opm.

Remove

`./opm remove <type> <name>`

Remove a package managed by opm.

Update

`./opm update[ <type>[ <name>]]`

Update packages managed by opm.

Stats

`./opm stats[ <type>[ <name>]]`

Returns statistics for packages managed by opm.

Search

`./opm[ -t <type>] <name>`

Returns all results that match