

---

# NodeBB Documentation

*0.4.1*

The NodeBB Team (<https://github.com/NodeBB/NodeBB/graphs/c>)

2014 07 22



<b>1</b>	<b>NodeBB</b>	<b>3</b>
1.1	NodeBB	3
1.2	NodeBB	9
<b>2</b>	<b>NodeBB</b>	<b>13</b>
2.1		13
2.2	Web /	16
<b>3</b>	<b>NodeBB</b>	<b>21</b>
3.1	NodeBB	21
<b>4</b>	<b>NodeBB</b>	<b>23</b>
4.1	NodeBB	23
<b>5</b>	<b>NodeBB</b>	<b>27</b>
5.1		27
5.2	Social Network SSOs	29
5.3	Image APIs	30
<b>6</b>	<b>NodeBB</b>	<b>31</b>
6.1	NodeBB	31
6.2		32
<b>7</b>		<b>33</b>
7.1	NodeBB	33
7.2		35
7.3		40
<b>8</b>		<b>47</b>
8.1	Writing Widgets for NodeBB	47
<b>9</b>		<b>49</b>
9.1	Creating a new NodeBB Theme	49
9.2	Rendering Engine	50
<b>10</b>		<b>55</b>
10.1		55
<b>11</b>	<b>NodeBB</b>	<b>57</b>

11.1	Helping out the NodeBB Project . . . . .	57
<b>12</b>	<b>/</b>	<b>59</b>
12.1	Need Help? . . . . .	59
<b>13</b>		<b>61</b>

**NodeBB** [web sockets](#) [NodeBB](#)

[NodeBB](#) [GitHub](#)



## 1.1 NodeBB

NodeBB

### 1.1.1 Ubuntu

:

```
$ sudo apt-get install git nodejs redis-server imagemagick npm
```

MongoDB, LevelDB, Redis, Configuring Databases .

**Node.js 0.8 (Ubuntu 12.10, 13.04), “node –version“ Node.js :**

```
$ sudo add-apt-repository ppa:chris-lea/node.js
$ sudo apt-get update && sudo apt-get dist-upgrade
```

:

```
$ git clone git://github.com/NodeBB/NodeBB.git nodebb
```

NodeBB :

```
$ cd nodebb
$ npm install
```

setup:

```
$ ./nodebb setup
```

redis

.

```
$ ./nodebb start
```

NodeBB , supervisor forever..

### 1.1.2 Debian

Ubuntu Debian NodeJS Redis

NodeBB :

- Node.js 0.8
- Redis 2.6
- cURL, sudo apt-get install curl

### Node.js

Debian 7, Debian 6 *nodejs* , Debian Node.js

### Wheezy Backport :

#### Debian 7, root :

```
$ echo "deb http://ftp.us.debian.org/debian wheezy-backports main" >> /etc/apt/sources.list
$ apt-get update
```

Node.js + NPM, :

```
$ apt-get install nodejs-legacy
$ curl --insecure https://www.npmjs.org/install.sh | bash
```

Node.js 0.8 ( 2014329 : 0.10.21)

:

Debian 6 (Squeeze) , root :

```
$ sudo apt-get install python g++ make checkinstall
$ src=$(mktemp -d) && cd $src
$ wget -N http://nodejs.org/dist/node-latest.tar.gz
$ tar xzvf node-latest.tar.gz && cd node-v*
$ ./configure
$ fakeroot checkinstall -y --install=no --pkgversion $(echo $(pwd) | sed -n -re 's/\.+node-v(.+)$/\1/p')
$ sudo dpkg -i node_*
```

### DotDeb

Dotdeb Debian LAMP

- Nginx,
- PHP 5.4 and 5.3 (useful PHP extensions : APC, imagick, Pinba, xcache, Xdebug, XHpro..)
- MySQL 5.5,
- Percona toolkit,
- Redis,
- Zabbix,
- Passenger...

Dotdeb :

- Debian 6.0 “Squeeze“ and 7 “Wheezy“
- both amd64 and i386 architectures

### Debian 7 (Wheezy) :

DotDeb :

```
$ sudo echo 'deb http://packages.dotdeb.org wheezy all' >> /etc/apt/sources.list
$ sudo echo 'deb-src http://packages.dotdeb.org wheezy all' >> /etc/apt/sources.list
```

GPC keys :

```
$ wget http://www.dotdeb.org/dotdeb.gpg
$ sudo apt-key add dotdeb.gpg
```

:

```
$ sudo apt-get update
```

### Debian 6 (Squeeze)

DotDeb :

```
$ sudo echo 'deb http://packages.dotdeb.org squeeze all' >> /etc/apt/sources.list
$ sudo echo 'deb-src http://packages.dotdeb.org squeeze all' >> /etc/apt/sources.list
```

GPC keys :

```
$ wget http://www.dotdeb.org/dotdeb.gpg
$ sudo apt-key add dotdeb.gpg
```

:

```
$ sudo apt-get update
```

## NodeBB

NodeJS Redis,

```
$ apt-get install redis-server imagemagick git
```

:

```
$ cd /path/to/nodebb/install/location
$ git clone git://github.com/NodeBB/NodeBB.git nodebb
```

NPM NodeBB :

```
$ cd /path/to/nodebb/install/location/nodebb (or if you are on your install location directory run :
$ npm install
```

*-setup* NodeBB :

```
$ ./nodebb setup
```

1. **URL ssh ip ip** : http://0.0.0.0 or http://example.org
2. **“NodeBB”** : : nginx 80
3. redis-server redis

```
.. NodeBB forum
```

```
$ ./nodebb start
```

```
: NodeBB , NodeBB (), supervisor and forever NodeBB :doc: <../running/index> !
```

NodeBB , ..

NodeBB 80 Nginx

### 1.1.3 SmartOS

NodeBB :

- Node.js 0.8
- Redis 2.6 ( Joyent’s package ).
- nginx, 1.3.13 ( nginx NodeBB ).

1. Joyent : [Joyent.com](http://Joyent.com)

2. : Create Instance

3. smartos nodejs .

```
      : smartos nodejs 13.1.0
```

4. instance *Running*

5. admin . Credentials ,

**Example:** ssh root@0.0.0.0 A#Ca{c1@3

6. admin root SSH : ssh admin@0.0.0.0

```
      : ssh Windows , : Cygwin.com
```

1. NodeBB :

```
$ sudo pkgin update
```

```
$ sudo pkgin install scmgit nodejs build-essential ImageMagick redis
```

```

:
$ pkgin search *failed-name*
$ sudo pkgin install *available-name*

```

## 2. **redis-server** (): : redis server

```

: redis-server .

$ svcadm enable redis
$ svcs

- svcs "/pkgsrc/redis:default":

$ svcadm clear redis

- redis-server :

$ svcadm disable redis

- redis-server :

$ svcadm enable redis

```

## 3. nodebb :

```
$ cd /parent/directory/of/nodebb/
```

## 4. NodeBB :

```
$ git clone git://github.com/NodeBB/NodeBB.git nodebb
```

## 5. NodeBB npm :

```
$ cd nodebb/
$ npm install

```

## 6. NodeBB :

```
$ node app --setup

(a) URL ssh ip ip
    : http://0.0.0.0 or http://example.org

(b) NodeBB :
    : nginx 80

(c) redis-server redis

```

## 7. NodeBB :

```

NodeBB :
: .
$ node app

```

```

8. ! : 4567: http://0.0.0.0:4567 or http://example.org:4567
    : :80

```

```
: <https://github.com/NodeBB/NodeBB/issues>'_
```

## NodeBB

: *NodeBB*.

### 1.1.4 Windows 8

:

- <https://windows.github.com/>
- <http://nodejs.org/>
- <http://sourceforge.net/projects/redis/files/redis-2.6.10/>

## NodeBB

Redis Server

Git Shell, . NodeBB repo:

```
git clone https://github.com/NodeBB/NodeBB.git
```

:

```
cd NodeBB
```

:

```
npm install
```

:

```
node app.js
```

!

```
node app.js
```

```
http://127.0.0.1:4567/
```

## Windows

NodeBB supervisor:

```
npm install -g supervisor
```

bash:

```
bash
```

“watch” NodeBB:

```
./nodebb watch
```

NodeBB

- *Ubuntu*

- *Debian*
- *SmartOS*
- *Windows*
- *CentOS ()*

## 1.2 NodeBB

PaaS NodeBB

### 1.2.1 Heroku

**Note:** Installations to Heroku require a local machine with some flavour of unix, as NodeBB does not run on Windows.

1. Download and install [Heroku Toolbelt](#) for your operating system
2. Log into your Heroku account: `heroku login`
3. Verify your Heroku account by adding a credit card (at <http://heroku.com/verify>)
4. Clone the repository: `git clone https://github.com/NodeBB/NodeBB.git /path/to/repo/clone`
5. `cd /path/to/repo/clone`
6. Install dependencies locally `npm install`
7. Create the heroku app: `heroku create`
8. Enable WebSocket support (beta): `heroku labs:enable websockets -a {APP_NAME}`, where `{APP_NAME}` is provided by Heroku, and looks something like `adjective-noun-wxyz.herokuapp.com` (NOTE: [See this doc](#)): drop the `.herokuapp.com` when entering `{APP_NAME}` above.
9. Enable [Redis To Go](#) for your heroku account: `heroku addons:add redistogo:nano`
10. Run the NodeBB setup script: `node app --setup` (information for your Heroku server and Redis to Go instance can be found in your account page)
  - Your server name is found in your Heroku app's "settings" page, and looks something like `adjective-noun-wxyz.herokuapp.com`
  - Use any port number. It will be ignored.
  - Your redis server can be found as part of the redis url. For example, for the url: `redis://redistogo:h28h3wgh37fns7@crestfish.redistogo.com:12345/`
  - The server is `fishyfish.redistogo.com`
  - The port is `12345`
  - The password is `h28h3wgh37fns7`
12. Add the following two packages to the dependencies section of your `package.json`:

```
"dependencies": {
  ...
  "nodebb-plugin-dbsearch": "0.0.10",
  "redis": "~0.10.1",
  "connect-redis": "~2.0.0"
```

```
},  
"devDependencies": {
```

13. Create a Procfile for Heroku: `echo "web: node app.js" > Procfile`

14. Commit the Procfile:

```
git add -f Procfile config.json package.json && git commit -am "adding Procfile and configs for Heroku"
```

15. Push to heroku: `git push heroku master` \* Ensure that a proper SSH key was added to your account, otherwise the push will not succeed!

16. Initialise a single dyno: `heroku ps:scale web=1`

17. Visit your app!

If these instructions are unclear or if you run into trouble, please let us know by [filing an issue](#).

### Keeping it up to date

If you wish to pull the latest changes from the git repository to your Heroku app:

1. Navigate to your repository at `/path/to/nodebb`
2. `git pull`
3. `npm install`
4. `node app --upgrade`
5. `git commit -am "upgrading to latest nodebb"`
6. `git push heroku master`

## 1.2.2 Cloud 9 IDE

### Cloud 9 IDE

**1:** Github NodeBB :

```
git clone git://github.com/NodeBB/NodeBB.git nodebb
```

Git nodebb nodebb

**2:** Cloud9 redis

```
nada-nix install redis
```

**3:** 16379 redis - Cloud 96379 “&”. \$IP Cloud 9 ip

```
redis-server --port 16379 --bind $IP &
```

**4:** ip NodeBB . Cloud 9 config.json \$IP ( node app -setup \$IP).

```
echo $IP
```

**5:** NodeBB :

```
npm install
```

**6:** nodebb :

```
node app --setup
```

```
URL 'http://workspace_name-c9-username.c9.io', workspace_name username NodeBB http jQuery http:// https://  
jQuery NodeBB
```

```
- Cloud9 804567
```

```
NodeBB? No
```

```
Redis IP: localhost ($IP )
```

```
IP : Enter what your $IP value holds here found in 4 $IP : 123.4.567.8
```

```
Redis : 16379
```

```
Redis : Redis Enter
```

```
Admin email
```

```
! IDE Run , :
```

```
node app
```

```
http://workspace_name-c9-username.c9.io
```

```
. Redis
```

```
redis-server --port 16379 --bind $IP
```

- *Heroku*
- *Cloud9*
- Openshift
- Digital Ocean ()



## 2.1

NodeBB (DBAL)

### 2.1.1 MongoDB

If you're afraid of running out of memory by using Redis, or want your forum to be more easily scalable, you can install NodeBB with MongoDB. This tutorial assumes you know how to SSH into your server and have root access.

**These instructions are for Ubuntu. Adjust them accordingly for your distro.**

**Note:** If you have to add `sudo` to any command, do so. No one is going to hold it against you ;)

#### 1 MongoDB

MongoDB ([MongoDB](#))

#### 2 node.js

MongoDB, the latest and greatest node.js is required (or at least greater than the package manager), so I'm leaving this to the official wiki. The instructions to install can be found on [Joyent](#).

**Note:** NPM is installed along with node.js, so there is no need to install it separately

#### 3 :

Enter the following into the terminal to install the base software required to run NodeBB:

```
# apt-get install git build-essential imagemagick
```

#### 4 :

Enter the following into the terminal, replacing `/path/to/nodebb/install/location` to where you would like NodeBB to be installed.

```
$ cd /path/to/nodebb/install/location
$ git clone git://github.com/NodeBB/NodeBB.git nodebb
```

### 5 : NodeBB

Go into the newly created *nodebb* directory and install the required dependencies by entering the following.

```
$ cd nodebb
$ npm install
```

### 6 :

MongoDB

```
$ mongo
```

*nodebb*

```
> use nodebb
```

*nodebb*

```
> db.createUser( { user: "nodebb",
...           pwd: "<>",
...           roles: [ "readWrite" ] } )
```

```
: readWrite (collection)
```

### 7 : MongoDB

MongoDB /etc/mongodb.conf

```
# nano /etc/mongodb.conf
```

```
setParameter=textSearchEnabled=true auth = true MongoDB
```

```
# service mongodb restart
```

### 8 : NodeBB

Make sure you are in your NodeBB root folder. If not, just type:

```
$ cd /path/to/nodebb
```

To setup the app, type:

```
$ node app --setup
```

- Change the hostname to your domain name.
- Accept the defaults by pressing enter until it asks you what database you want to use. Type *mongo* in that field.
- Accept the default port, unless you changed it in the previous steps.
- Change your username to *nodebb*, unless you set it to another username.
- Enter in the password you made in step 5.

- Change the database to `nodebb`, unless you named it something else.

Continue with the installation, following the instructions the installer provides you.

## Step 9: Starting the App

To start the app, run:

```
$ node app
```

Now visit `yourdomainorip.com:4567` and your NodeBB installation should be running.

NodeBB can also be started with helper programs, such as *supervisor or forever*. You can also use `nginx` as a *reverse proxy*).

### 2.1.2 LevelDB

Follow the *installation instructions* for your particular OS but feel free to omit the Redis installation.

After cloning NodeBB, ensure that you run:

```
npm install levelup leveledown
```

Finally, set up a directory to store your LevelDB database, for example:

```
mkdir /var/level/
```

Run the NodeBB install, select `level` when it prompts you for your database. If you created the folder as above, you can leave the rest of the questions as default.

- Redis (, )
- *Mongo*
- *Level*

---

```
: NodeBB
```

---

### 2.1.3

```
:
```

Both databases **must** be flushed before beginning - there isn't a mechanism yet that detects an existing installation on one database but not another. Until fail-safe's such as these are implemented this option is hidden under the `--advanced setup` flag.

```
node app --setup --advanced
```

Consult the other database guides for instructions on how to set up each specific database. Once you select a secondary database's modules, there's no turning back - until somebody writes an exporter/importer.

Currently this setup is being tested with Redis as the primary store (sets, lists, and sorted sets, because Redis is super fast with these), and Mongo as the hash store (post and user data, because ideally we wouldn't want this in RAM).

## 2.2 Web /

NodeBB

### 2.2.1 nginx

NodeBB 4567

`http://example.org:4567`

NodeBB NodeBB nginx NodeBB

- **NGINX v1.3.13**

- [Ubuntu](#), [NGINX](#) [NGINX PPA](#) [Debian](#) [DotDeb](#) [Nginx](#)
- `nginx nginx -V`

NGINX server nginx

- `/path/to/nginx/sites-available/* - ../sites-enabled`
- `/path/to/nginx/conf.d/*.conf - .conf`
- `/path/to/nginx/httpd.conf -`

nginx NodeBB 4567

```
server {
    listen 80;

    server_name forum.example.org;

    location / {
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
        proxy_set_header X-NginX-Proxy true;

        proxy_pass http://127.0.0.1:4567/;
        proxy_redirect off;

        # Socket.IO Support
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }
}
```

- `config.json use_port true false`
- `nginx 1.4.x websocketsDebian/Ubuntu 1.2 NodeBB`
- `NodeBB nginx proxy_pass IP 127.0.0.1 NodeBB`
- `nginx forum.example.org DNS forum.example.org nginx`

## 2.2.2 Configuring apache as a proxy

**Prerequisites to making this work:** Apache 2.4.x

### What if I'm on 2.2.x (Debian/Ubuntu)?

you need to manually compile and add the module “mod\_proxy\_wstunnel” to the Apache 2.2 branch. If you're running Ubuntu or Debian, you're likely on the 2.2 branch of code.

The following guide will assist with that if you're on Debian or Ubuntu. This is what I used to backport the mod\_proxy\_wstunnel module to the 2.2 code base of Apache;

[http://www.amoss.me.uk/2013/06/apache-2-2-websocket-proxying-ubuntu-mod\\_proxy\\_wstunnel/](http://www.amoss.me.uk/2013/06/apache-2-2-websocket-proxying-ubuntu-mod_proxy_wstunnel/)

### NOTE: On ubuntu, if you're missing the ./configure file

You need to first run ./buildconf. After this is complete, you will then be able to use ./configure.

**automake & libtool package was needed too.**

```
apt-get install automake libtool
```

### Enable the necessary modules

1. `sudo a2enmod proxy`
2. `sudo a2enmod proxy_http`
3. `sudo a2enmod proxy_wstunnel`

### Add the config to Apache

The next step is adding the configuration to your virtualhost.conf file, typically located in /etc/apache2/sites-available/. The below configuration assumes you've used 4567 (default) port for NobeBB installation. It also assumes you have the bind address set to 127.0.0.1.

```
ProxyRequests off

<Proxy *>
    Order deny,allow
    Allow from all
</Proxy>
ProxyPass /socket.io/1/websocket ws://127.0.0.1:4567/socket.io/1/websocket
ProxyPassReverse /socket.io/1/websocket ws://127.0.0.1:4567/socket.io/1/websocket
```

```
ProxyPass /socket.io/ http://127.0.0.1:4567/socket.io/  
ProxyPassReverse /socket.io/ http://127.0.0.1:4567/socket.io/
```

```
ProxyPass / http://127.0.0.1:4567/  
ProxyPassReverse / http://127.0.0.1:4567/
```

The last thing you need to be sure of is that the `config.json` in the NodeBB folder has `use_port: false`. Otherwise some functionality will not work properly.

### Example `nodebb/config.json`

```
{  
  "base_url": "http://www.yoursite.com",  
  "port": "4567",  
  "use_port": false,  
  "secret": "55sb254c-62e3-4e23-9407-8655147562763",  
  "bind_address": "127.0.0.1",  
  "database": "redis",  
  "redis": {  
    "host": "127.0.0.1",  
    "port": "6379",  
    "password": "",  
    "database": "0"  
  },  
  "bcrypt_rounds": 12,  
  "upload_path": "/public/uploads",  
  "relative_path": ""  
}
```

**Change the domain and dont use the secret in the example above.**

## 2.2.3 Configuring Varnish Cache

To be sure Varnish will work properly with NodeBB check that your configuration `/etc/varnish/default.vcl` is optimized for **websockets**.

```
backend nodebb {  
  .host = "127.0.0.1"; # your nodebb host  
  .port = "4567"; # your nodebb port  
}  
  
sub vcl_recv {  
  
  # Pipe websocket connections directly to Node.js  
  if (req.http.Upgrade ~ "(?i)websocket") {  
    set req.backend = nodebb;  
    return (pipe);  
  }  
  
  # NodeBB  
  if (req.http.host == "forum.yourwebsite.com") { # change this to match your host  
    if (req.url ~ "^/socket.io/") {  
      set req.backend = nodebb;  
      return (pipe); # return pass seems not working for websockets  
    }  
    return (pass); # don't cache  
  }  
}
```

```
    }  
}  
  
sub vcl_pipe {  
    # Need to copy the upgrade header  
    if (req.http.upgrade) {  
        set bereq.http.upgrade = req.http.upgrade;  
    }  
}
```

- *Nginx*
- *Apache*
- *Varnish Cache*



## 3.1 NodeBB

NodeBB NodeBB

- `./nodebb start NodeBB`
- `./nodebb stop NodeBB`
- `npm start npm stop`

NodeBB

### 3.1.1 Node.js

NodeBB node nodejs

```
$ cd /path/to/nodebb/install
$ node app
```

, NodeBB :

### 3.1.2 Supervisor

supervisor NodeBB NodeBB

```
$ npm install -g supervisor
$ supervisor app
```

supervisor stdout

### 3.1.3 Forever

NodeBB `forever` forever NodeBB NodeBB

```
$ npm install -g forever
$ forever start app.js
```



## 4.1 NodeBB

NodeBB

```
git NodeBB
```

```
git clone(master) 100%
```

```
** NodeBB
```

### 4.1.1 Upgrade Path

NodeBB's upgrade path is designed so that upgrading between versions is straightforward. NodeBB will provide upgrade compatibility (via the `--upgrade` flag) between the latest version of a lower branch and the latest version of the higher branch. For example, if `v0.2.2` is the latest version in the `v0.2.x` branch, you can switch to the `v0.3.x` branch and suffer no ill effects. Upgrading from `v0.2.0` to `v0.3.x` is not supported, and NodeBB will warn you when attempting to upgrade that you are not upgrading cleanly.

#### Upgrading between patch revisions

*e.g. v0.1.0 to v0.1.1*

Patch revisions contain bugfixes and other minor changes. Updating to the latest version of code for your specific version branch is all that is usually required.

**Execute steps 1 through 3.**

#### Upgrading between minor revisions

*e.g. v0.1.3 to v0.2.0*

Minor revisions contain new features or substantial changes that are still backwards compatible. They may also contain dependent packages that require upgrading, and other features may be deprecated (but would ideally still be supported).

Execute steps 1 through 4.

## 4.1.2 Upgrade Steps

**Note:** After upgrading between revisions (i.e. v0.0.4 to v0.0.5), it may be necessary to run the following upgrade steps to ensure that any data schema changes are properly upgraded as well:

### 1. Shut down your forum

While it is possible to upgrade NodeBB while it is running, it is definitely not recommended, particularly if it is an active forum:

```
$ cd /path/to/nodebb
$ ./nodebb stop
```

### 2. Back up your data

---

: This section is incomplete, please take care to back up your files properly!

---

#### Backing up Redis

As with all upgrades, the first step is to **back up your data!** Nobody likes database corruption/misplacement.

All of the textual data stored in NodeBB is found in a `.rdb` file. On typical installs of Redis, the main database is found at `/var/lib/redis/dump.rdb`.

**Store this file somewhere safe.**

#### Backing up MongoDB

To run a backup of your complete MongoDB you can simply run

```
mongodump
```

which will create a directory structure that can be restored with the `mongorestore` command.

It is recommended that you first shut down your database. On Debian / Ubuntu it's likely to be: `sudo service mongod stop`

#### Backing up LevelDB

As LevelDB is simply a collection of flat files, just copy the database over to a safe location, ex.

```
cp -r /path/to/db /path/to/backups
```

**Store this file somewhere safe.**

#### Avatars

Uploaded images (avatars) are stored in `/public/uploads`. Feel free to back up this folder too:

```
cd /path/to/nodebb/public
tar -czf ~/nodebb_assets.tar.gz ./uploads
```

### 3. Grab the latest and greatest code

Navigate to your NodeBB: `$ cd /path/to/nodebb.`

If you are upgrading from a lower branch to a higher branch, switch branches as necessary. **\*Make sure you are completely up-to-date on your current branch!\***

For example, if upgrading from `v0.3.2` to `v0.4.3`:

```
$ git fetch      # Grab the latest code from the NodeBB Repository
$ git checkout v0.4.x  # Type this as-is! Not v0.4.2 or v0.4.3, but "v0.4.x"!
$ git merge origin/v0.4.x
```

If not upgrading between branches, just run the following command:

```
$ git pull
```

This should retrieve the latest (and greatest) version of NodeBB from the repository.

Alternatively, download and extract the latest versioned copy of the code from [the Releases Page](#). Overwrite any files as necessary. This method is not supported.

### 4. Run the NodeBB upgrade script

This script will install any missing dependencies, upgrade any plugins or themes (if an upgrade is available), and migrate the database if necessary.

```
$ ./nodebb upgrade
```

**Note:** `./nodebb upgrade` is only available after `v0.3.0`. If you are running an earlier version, run these instead:

- `npm install`
- `ls -d node_modules/nodebb* | xargs -n1 basename | xargs npm update`
- `node app --upgrade`

### 6. Start up NodeBB & Test!

You should now be running the latest version of NodeBB.



## 5.1

---

: If you are interested in updating these (preferably with screenshots) let us know on the [community forum](#) because we are planning on a design overhaul of the ACP soon.

---

**To view the admin panel** (if you are an admin): [http://your\\_nodebb\\_domain.com/admin](http://your_nodebb_domain.com/admin)

### Top Menu (all can be easily reached by other means)

- NodeBB ACP (Administrator Control Panel: this view) \* [http://your\\_nodebb\\_domain.com/admin/index](http://your_nodebb_domain.com/admin/index) (see Home below)
- Forum \* [http://your\\_nodebb\\_domain.com](http://your_nodebb_domain.com) (your main forum)
- Home \* [http://your\\_nodebb\\_domain.com/admin/index](http://your_nodebb_domain.com/admin/index) (see Home below)
- Settings \* [http://your\\_nodebb\\_domain.com/admin/settings](http://your_nodebb_domain.com/admin/settings) (see Settings below)

### Side Menu: NodeBB

- Home \* [http://your\\_nodebb\\_domain.com/admin/index](http://your_nodebb_domain.com/admin/index) \* Links all go to nodebb.com homepage
    - NOTE: should all the links go to same place?
    - Message reminder of what version this is and to check for updates (See [Upgrading NodeBB](#)) \* NOTE: would a link to where the latest stable version is help?
    - Active Users \* lists number users per page-path (?) \* NOTE: not clear exactly what the paths mean or how to visit that path
  - Categories \* [http://your\\_nodebb\\_domain.com/admin/categories](http://your_nodebb_domain.com/admin/categories) \* Filters: Active, Disabled, Unit Tests \* List of Categories:
    - Icon, Name, Desc, Action: Disable
    - Actions: Save, Add New
  - Users \* [http://your\\_nodebb\\_domain.com/admin/users](http://your_nodebb_domain.com/admin/users) \* Filters: Latest Users, Top Posters, Most Reputation, Action: Search \* List of Users:
    - Icon, Link: Name, Reputation(star), Number Posts(pencil), Action: Ban
    - Action: Load More
  - Groups \* [http://your\\_nodebb\\_domain.com/admin/groups](http://your_nodebb_domain.com/admin/groups)
    - List of Groups
-

- Name, Desc, Icon
  - Action: Delete Group \* NOTE: What exactly can Groups be set up to do, besides Admin?
- Topics \* [http://your\\_nodebb\\_domain.com/admin/topics](http://your_nodebb_domain.com/admin/topics) \* List of Topics
  - Name [link to topic], Posted When and By, Number posts (Topic+Replies), Thread Actions: Pin(pushpin), Lock(lock), Delete(trashcan)
  - Action: Load More Topics
  - Topic [from List of Topics link]
  - Normal View of Topic+Reply Posts but with: \* Link, Edit, Delete Actions all enabled for each Post \* Thread Tools:
    - \* Pin, Lock, Move, Delete
- Themes (See [Theming NodeBB](#)) \* [http://your\\_nodebb\\_domain.com/admin/themes](http://your_nodebb_domain.com/admin/themes) \* List of (Custom | Bootswatch) Themes
  - Actions: Use, Preview
  - Action: Revert (to base)
- Plugins (See [Writing Plugins for NodeBB](#)) \* [http://your\\_nodebb\\_domain.com/admin/plugins](http://your_nodebb_domain.com/admin/plugins) \* List of Plugins
  - Action: De/activate
  - Info on making plugins
- Settings \* [http://your\\_nodebb\\_domain.com/admin/settings](http://your_nodebb_domain.com/admin/settings)
  - General Settings \* (textbox) Site Title \* (textbox) Site Description \* (textbox) Site Keywords \* (textbox) Imgur Client ID
    - \* NOTE: How does this function?
    - \* (textbox) Maximum User Image Size
  - Privilege Thresholds (Use privilege thresholds to manage how much reputation a user must gain to receive moderator access.) \* (textbox) Manage Thread \* (textbox) Moderate Users \* (textbox) Create Pinned Topics
  - Email Settings \* (textbox) Email Address (The following email address refers to the email that the recipient will see in the “From” and “Reply To” fields.) \* (textbox) SMTP Server Host (Default: 127.0.0.1) \* (textbox) SMTP Server Port
  - User Settings \* (textbox) Minimum Username Length \* (textbox) Maximum Username Length \* (textbox) Minimum Password Length
  - Post Settings \* (textbox) Post Delay \* (textbox) Minimum Title Length \* (textbox) Minimum Post Length \* (checkbox) Use Outgoing Links Warning Page
  - Action: Save
- Redis \* [http://your\\_nodebb\\_domain.com/admin/redis](http://your_nodebb_domain.com/admin/redis) \* Redis data storage stats
- Logger \* [http://your\\_nodebb\\_domain.com/admin/logger](http://your_nodebb_domain.com/admin/logger) \* (checkbox) Enable HTTP logging \* (checkbox) Enable socket.io event logging \* (textbox) Path to log file
- MOTD (Message of the Day) \* [http://your\\_nodebb\\_domain.com/admin/motd](http://your_nodebb_domain.com/admin/motd) \* (textarea) You can enter either full HTML or Markdown text. \* (checkbox) Show the Message of the Day

**Side Menu: Social Authentication** (See [Enabling Social Network Logins](#))

- Twitter
- `http://your_nodebb_domain.com/admin/twitter`
- Facebook
- `http://your_nodebb_domain.com/admin/facebook`
- Google+
- `http://your_nodebb_domain.com/admin/gplus`

**Side Menu: Plugins** (Shows installed plugins)

**Side Menu: Unit Tests** (Will run qunit tests)

## 5.2 Social Network SSOs

NodeBB supports integration for Facebook, Twitter, and Google through third party plugins:

- `npm install nodebb-plugin-sso-facebook`
- `npm install nodebb-plugin-sso-twitter`
- `npm install nodebb-plugin-sso-google`

Other SSO vendors are available, such as GitHub. Please check the [plugin directory](#) for a list of all SSO vendors.

After installing and activating them, they require an API key in order to function:

### 5.2.1 Facebook

Register an application via the [Facebook Developers](#) page. A credit card or mobile phone number may be required in order to create a Developer account.

Create a new application, and obtain an Application Key and Application Secret:

Ensure that “Website with Facebook Login” is checked, and that the URL to your NodeBB instance is specified in the “Site URL” box. Add that site’s domain to the “App Domains” field.

Paste this key and secret into the appropriate boxes in the NodeBB Administration Panel (accessible via `/admin` on your NodeBB install)

### 5.2.2 Twitter

Register an application at the [Twitter Developers](#) page. Create a new Application, and obtain the Access Token and Secret:

**Important:** While setting up your application, be sure to specify a Callback URL. It does not have to correspond to your installation, it just cannot be blank.

Paste this token and secret into the appropriate boxes in the NodeBB Administration Panel (accessible via `/admin` on your NodeBB install)

### 5.2.3 Google

Register an application at the [Google API Console](#), and obtain a Client ID and Secret.

Paste this ID and secret into the appropriate boxes in the NodeBB Administration Panel (accessible via /admin on your NodeBB install)

## 5.3 Image APIs

### 5.3.1 Imgur

imgur :

<https://api.imgur.com/oauth2/addclient>

: “Anonymous usage without user authorization” ()

“Client ID”.

nodebb-plugin-imgur:

```
npm install nodebb-plugin-imgur
```

NodeBB

Imgur Client ID “Imgur Client ID”

### 5.3.2 Amazon S3

---

: [the plugin thread](#)

---

## 6.1 NodeBB

NodeBB [Google Javascript](#)

### 6.1.1

---

: 20137 100%

---

### 6.1.2

NodeBB [One True Brace Style](#):

```
if (condition) {
    // code here ...
} else {
    // otherwise ...
}

if (!leTired) {
    haveANap();
}
```

### 6.1.3

```
someFunction(parameters, function(err, data) {
    if(err) {
        return callback(err); // or handle error
    }
    // proceed as usual
});
```

### 6.1.4

*var*

```
var foo = 'bar';
```

*var*

```
var foo = 'bar',  
    bar = 'baz';
```

### 6.1.5

### 6.1.6

```
functionNamesLikeThis, variableNamesLikeThis, ClassNamesLikeThis, EnumNamesLikeThis, methodNamesLikeThis
```

## 6.2

---

: This section is under construction. Have a look at the modules folder for more information, located at:

`public/src/modules`

---

### 6.2.1 Alerts

The alert module is a toaster notification that can be called via the following syntax:

```
app.alert({  
  title: 'Success!',  
  message: 'Here\'s an example of an alert!',  
  location: 'left-bottom',  
  timeout: 2500,  
  type: 'success',  
  image: 'https://i.imgur.com/dJBzcGT.jpg'  
});
```

The above code will result in this notification (default styling):

To style this, have a look at the vanilla theme's `modules/alert.less` and `templates/alert.tpl`.

#### Parameters:

1. `title` - string, which can be a language string as well. Some core language strings that you can use here include: `[[global:alert.success]]` and `[[global:alert.error]]`
2. `message` - string, which can be a language string as well.
3. `location` (optional) - `right-top` (default), `left-top`, `right-bottom`, `left-bottom`
4. `timeout` (optional) - integer in milliseconds, default is permanent until closed.
5. `type` - `error`, `success`, `info`, `warning/notify`
6. `image` (optional) - string, URL to image.
7. `closefn` (optional) - function. This is called when the user closes the alert via the (X) button.
8. `clickfn` (optional) - function. This is called when the user clicks on the alert.

---

---

## 7.1 NodeBB

NodeBB

WordPress NodeBB NodeBB NodeBB

### 7.1.1

NodeBB “”“”

NodeBB

NodeBB

### 7.1.2

plugin.json

```
{
  "id": "my-plugin",
  "name": "",
  "description": "",
  "url": " Github ",
  "library": "./my-plugin.js",
  "staticDirs": {
    "images": "public/images"
  },
  "less": [
    "assets/style.less"
  ],
  "hooks": [
    { "hook": "filter:post.save", "method": "filter" },
    { "hook": "action:post.save", "method": "emailme" }
  ]
}
```

id

library NodeBB

```
staticDirs staticDirs NodeBB /plugins/{YOUR-PLUGIN-ID}
```

- `staticDirs /path/to/your/plugin/public/images /plugins/my-plugin/images`

less NodeBB CSS

The hooks NodeBB ():

- hook, NodeBB
- method,
- priority, (: 10)

### 7.1.3

NodeBB

- 
- `:doc: <hooks>`

### 7.1.4

action:post.save plugin.json hooks

```
{ "hook": "action:post.save", "method": "myMethod" }
```

```
var MyPlugin = {  
  myMethod: function(postData) {  
    // postData  
  }  
};
```

### 7.1.5 NodeBB

NodeBB User exists module.parent.require NodeBB

```
var User = module.parent.require('./user');  
User.exists('foobar', function(err, exists) {  
  // ...  
});
```

### 7.1.6

npm "nodebb-plugin-" npm install

npm "nodebb-plugin-" NodeBB

v0.0.5 /plugins "" id NodeBB *NodeBB*

## 7.1.7

NodeBB

```
./nodebb dev
```

## 7.1.8

```
./nodebb reset plugins
```

```
./nodebb reset plugin="nodebb-plugin-im-broken"
```

## 7.2

NodeBB *NodeBB*.

```
()
```

```
: ()
```

### 7.2.1

**filter:admin.header\_build**

ACP

**filter:post.save**

: (markdown)

**filter:post.get**

: (javascript)

**filter:header.build**

**NodeBB**

**filter:register.build**

:

- *req* express (javascript)
- *res* express (javascript)
- *data* (javascript)

**‘data.captcha‘**

`filter:post.parse`

: ()

HTML () Markdown BBCode

`filter:posts.custom_profile_info`

`filter:register.check`

:

- *req* express (javascript)
- *res* express (javascript)
- *userData*

`filter:scripts.get`

JS

`filter:uploadImage`

`filter:uploadFile`

`filter:widgets.getAreas`

`filter:widgets.getWidgets`

`filter:search.query`

`filter:post.parse`

`filter:messaging.save`

`filter:messaging.parse`

`filter:sounds.get`

`filter:post.getPosts`

`filter:post.getFields`

`filter:auth.init`

`filter:composer.help`

`filter:topic.thread_tools`

`filter:user.create`

`filter:user.delete`

`filter:user.profileLinks`

`filter:user.verify.code`

`:confirm_code`

Ability to modify the generated verification code (ex. for using a shorter verification code instead for SMS verification)

`filter:user.custom_fields`

Parameters: userData

Allows you to append custom fields to the newly created user, ex. mobileNumber

`filter:register.complete`

Parameters: uid, destination

Set the post-registration destination, or do post-register tasks here.

`filter:widget.render`

## 7.2.2 Actions

`action:app.load`

**Argument(s):** None

Executed when NodeBB is loaded, used to kickstart scripts in plugins (i.e. cron jobs, etc)

`action:page.load`

**Argument(s):** An object containing the following properties:

- `template` - The template loaded
- `url` - Path to the page (relative to the site's base url)

`action:plugin.activate`

**Argument(s):** A String containing the plugin's `id` (e.g. `nodebb-plugin-markdown`)

Executed whenever a plugin is activated via the admin panel.

**Important:** Be sure to check the `id` that is sent in with this hook, otherwise your plugin will fire its registered hook method, even if your plugin was not the one that was activated.

`action:plugin.deactivate`

**Argument(s):** A String containing the plugin's `id` (e.g. `nodebb-plugin-markdown`)

Executed whenever a plugin is deactivated via the admin panel.

**Important:** Be sure to check the `id` that is sent in with this hook, otherwise your plugin will fire its registered hook method, even if your plugin was not the one that was deactivated.

`action:post.save`

**Argument(s):** A post object (javascript Object)

Executed whenever a post is created or edited, after it is saved into the database.

`action:email.send`

`action:post.setField`

`action:topic.edit`

`action:post.edit`

`action:post.delete`

`action:post.restore`

`action:notification.pushed`

**Argument(s):** A notification object (javascript Object)

Executed whenever a notification is pushed to a user.

`action:config.set`

`action:topic.save`

`action:user.create`

`action:topic.delete`

`action:user.verify`

Parameters: uid; a hash of confirmation data (ex. confirm\_link, confirm\_code) Useful for overriding the verification system. Currently if this hook is set, the email verification system is disabled outright.

`action:user.set`

Parameters: field (str), value, type ('set', 'increment', or 'decrement') Useful for things like awarding badges or achievements after a user has reached some value (ex. 100 posts)

`action:settings.set`

Parameters: hash (str), object (obj) Useful if your plugins want to cache settings instead of pulling from DB everytime a method is called. Listen to this and refresh accordingly.

## 7.2.3 Client Side Hooks

`filter:categories.new_topic`  
`action:popstate`  
`action:ajaxify.start`  
`action:ajaxify.loadingTemplates`  
`action:ajaxify.loadingData`  
`action:ajaxify.contentLoaded`  
`action:ajaxify.end`  
`action:reconnected`  
`action:connected`  
`action:disconnected`  
`action:categories.loading`  
`action:categories.loaded`  
`action:categories.new_topic.loaded`  
`action:topic.loading`  
`action:topic.loaded`  
`action:composer.loaded`  
`action:widgets.loaded`

## 7.3

NodeBB If you want to make your plugin customizable you may use the Settings Framework NodeBB offers.

### 7.3.1

First you need some default settings, just create a new object for this:

```
var defaultSettings = {
  booleans: {
    someBool: true,
    moreBools: [false, false, true]
  },
  strings: {
    someString: 'hello world',
    multiLineString: 'some\nlong\ntext',
  }
}
```

```

    arrayOfStrings: ['some\nlong\ntexts', 'and another one']
  },
  numbers: {
    multiArrayDimensions: [[42,42],[21,21]],
    multiArrayDimensions2: [[42,42],[[]],
    justSomeNumbers: [],
    oneNumber: 3,
    anotherNumber: 2
  },
  someKeys: ['C+S+#13'] // Ctrl+Shift+Enter
};

```

Now you can use the server-side settings-module to access the saved settings like this:

```

var Settings = module.parent.require('./settings');
var mySettings = new Settings('myPlugin', '0.1', defaultSettings, function() {
  // the settings are ready and can accessed.
  console.log(mySettings === this); // true
  console.log(this.get('strings.someString') === mySettings.get().strings.someString); // true
});

```

The second parameter should change at least every time the structure of default settings changes. Because of this it's recommended to use your plugins version.

To use the settings client-side you need to create a WebSocket that delivers the result of `mySettings.get()`.

The `mySettings`-object will cache the settings, so be sure to use methods like `mySettings.sync(callback)` when the settings got changed from somewhere else and `mySettings.persist(callback)` when you finished `mySettings.set(key, value)` calls.

You need to create a socket-listener like following to allow the admin to initiate a synchronization with the settings stored within database:

```

var SocketAdmin = module.parent.require('./socket.io/admin');
SocketAdmin.settings.syncMyPlugin = function() {
  mySettings.sync();
};

```

If you want to add a reset-functionality you need to create another socket-listener:

```

SocketAdmin.settings.getMyPluginDefaults = function (socket, data, callback) {
  callback(null, mySettings.createDefaultWrapper());
};

```

The methods of the `mySettings` object you probably want to use:

- `constructor()`
- **`sync([callback])`** Reloads the settings from database, overrides local changes.
- **`persist([callback])`** Saves the local changes within database.
- **`get([key])`** Returns the setting(s) identified by given key. If no key is provided the whole settings-object gets returned. If no such setting is saved the default value gets returned.
- **`set([key, ]value)`** Sets the setting of given key to given value. Remember that it's just a local change, you need to call `persist` in order to save the changes.
- **`reset([callback])`** Persists the default settings.
- **`getWrapper()`** Returns the local object as it would get saved within database.

- **createWrapper(version, settings)** Creates an object like it would get saved within database containing given information and settings.
- **createDefaultWrapper()** Creates an object like it would get saved within database containing the default settings.

### 7.3.2 Client-Side Access

The next step is making the settings available to the admin.

You need to use the *hooks* filter: `admin.header.build` (to display a link to your page within ACP) and `action:app.load` (to create the needed route).

Within your page you can access the client-side Settings API via

```
require(['settings'], function (settings) {
  var wrapper = $('#my_form_id');
  // [1]
  settings.sync('myPlugin', wrapper);
  // [2]
});
```

To make a button with the id `save` actually save the settings you can add the following at [2]:

```
$('#save').click(function(event) {
  event.preventDefault();
  settings.persist('myPlugin', wrapper, function(){
    socket.emit('admin.settings.syncMyPlugin');
  });
});
```

As said before the server-side settings-object caches the settings, so we emit a `WebSocket` to notify the server to synchronize the settings after they got persisted.

To use a reset-button you can add the following at [2]:

```
$('#reset').click(function(event) {
  event.preventDefault();
  socket.emit('admin.settings.getMyPluginDefaults', null, function (err, data) {
    settings.set('myPlugin', data, wrapper, function(){
      socket.emit('admin.settings.syncMyPlugin');
    });
  });
});
```

There you go, the basic structure is done. Now you need to add the form-fields.

Each field needs an attribute `data-key` to reference its position within the settings. The Framework does support any fields whose jQuery-object provides the value via the `val()` method.

The plugin to use for a field gets determined by its `data-type`, `type` or `tag-name` in this order.

**Additionally the following plugins are registered by default:**

- **array (types: `div`, `array`)** An Array of any other fields. Uses the object within `data-attributes` to define the array-elements. Uses `data-new` to define the value of new created elements.
- **key (types: `key`)** A field to input keyboard-combinations.
- **checkbox, number, select, textarea** Handle appropriate fields.

A full list of all attributes that may influence the behavior of the default Framework:

- `data-key`: the key to save/load the value within configuration-object
- `data-type`: highest priority type-definition to determine what kind of element it is or which plugin to associate
- `type`: normal priority type-definition
- `data-empty`: if `false` or `0` then values that are assumed as empty turn into null. `data-empty` of arrays affect their child-elements
- `data-trim`: if not `false` or `0` then values will get trimmed as defined by the elements type
- `data-split`: if set and the element doesn't belong to any plugin, it's value will get split and joined by its value into the field
- **array-elements:**
  - `data-split`: separator (HTML allowed) between the elements, defaults to `' , '`
  - `data-new`: value to insert into new created elements
  - `data-attributes`: an object to set the attributes of the child HTML-elements. `tagName` as special key will set the tag-name of the child HTML-elements
- **key-fields:**
  - `data-trim`: if `false` or `0` then the value will get saved as string else as object providing following properties: `ctrl`, `alt`, `shift`, `meta`, `code`, `char`
  - `data-split`: separator between different modifiers and the key-code of the value that gets saved (only takes effect if trimming)
  - `data-short`: if not `false` or `0` then modifier-keys get saved as first uppercase character (only takes effect if trimming)
- **select:**
  - `data-options`: an array of objects containing `text` and `value` attributes.

The methods of the `settings` module:

- **`registerPlugin(plugin[, types])`** Registers the given plugin and associates it to the given types if any, otherwise the plugins default types will get used.
- **`get ()`** Returns the saved object.
- **`set(hash, settings[, wrapper[, callback[, notify]])`** Refills the fields with given settings and persists them. `hash` Identifies your plugins settings. `settings` The object to save in database (settings-wrapper if you use server-side Settings Framework). `wrapper` (default: `'form'`) The DOM-Element that contains all fields to fill. `callback` (default: null) Gets called when done. `notify` (default: true) Whether to display saved- and fail-notifications.
- **`sync(hash[, wrapper[, callback]])`** Resets the settings to saved ones and refills the fields.
- **`persist(hash[, wrapper[, callback[, notify]])`** Reads the settings from given wrapper (default: `'form'`) and saves them within database.

For Settings 2.0 support the methods `load` and `save` are still available but not recommended.

### 7.3.3 Client-Side Example Template

An example template-file to use the same settings we already used server-side:

```

<h1>My Plugin</h1>
<hr />

<form id="my_form_id">
  <div class="row">
    <p>
      <h2>Settings</h2>
      A boolean: <input type="checkbox" data-key="booleans.someBool"></input><br>
      An array of checkboxes that are selected by default:
      <div data-key="booleans.moreBools" data-attributes='{ "data-type": "checkbox" }' data-new='t

      A simple input-field of any common type: <input type="password" data-key="strings.someStr

      A simple textarea: <textarea data-key="strings.multiLineString"></textarea><br>
      Array of textareas:
      <div data-key="strings.arrayOfStrings" data-attributes='{ "data-type": "textarea" }' data-ne

      2D-Array of numbers that persist even when empty (but not empty rows):
      <div data-key="numbers.multiArrayDimensions" data-split="<br>"
        data-attributes='{ "data-type": "array", "data-attributes": { "type": "number" } }' data-new=
      Same with persisting empty rows, but not empty numbers, if no row is given null will get
      <div data-key="numbers.multiArrayDimensions2" data-split="<br>" data-empty="false"
        data-attributes='{ "data-type": "array", "data-empty": true, "data-attributes": { "type": "nu

      Array of numbers (new: 42, step: 21):
      <div data-key="numbers.justSomeNumbers" data-attributes='{ "data-type": "number", "step": 21

      Select with dynamic options:
      <select data-key="numbers.oneNumber" data-options='[{"value": "2", "text": "2"}, {"value": "3

      Select that loads faster:
      <select data-key="numbers.anotherNumber"><br>
        <option value="2">2</option>
        <option value="3">3</option>
      </select>

      Array of Key-shortcuts (new: Ctrl+Shift+7):
      <div data-key="someKeys" data-attributes='{ "data-type": "key" }' data-new='Ctrl+Shift+#55'

    </p>
  </div>
  <button class="btn btn-lg btn-warning" id="reset">Reset</button>
  <button class="btn btn-lg btn-primary" id="save">Save</button>
</form>

<script>
  require(['settings'], function (settings) {
    var wrapper = $('#my_form_id');
    // [1]
    settings.sync('myPlugin', wrapper);
    $('#save').click(function (event) {
      event.preventDefault();
      settings.persist('myPlugin', wrapper, function() {
        socket.emit('admin.settings.syncMyPlugin');
      });
    });
    $('#reset').click(function (event) {
      event.preventDefault();
      socket.emit('admin.settings.getMyPluginDefaults', null, function (err, data) {
        settings.set('myPlugin', data, wrapper, function() {
          socket.emit('admin.settings.syncMyPlugin');
        });
      });
    });
  });

```

```

    });
  });
</script>

```

### 7.3.4 Custom Settings-Elements

If you want to define your own element-structure you can create a **plugin** for the Settings Framework.

This allows you to use a whole object like a single field which - besides comfort in using multiple similar objects - allows you to use them within arrays.

A plugin is basically an object that contains at least an attribute `types` that contains an array of strings that associate DOM-elements with your plugin.

You can add a plugin at [1] using the method `settings.registerPlugin`.

To customize the way the associated fields get interpreted you may add the following methods to your plugin-object:

All given elements are instances of JQuery.

All methods get called within Settings-scope.

- **use ()** Gets called when the plugin gets registered.
- **[HTML-Element|JQuery] create (type, tagName, data)** Gets called when a new element should get created (eg. by expansion of an array).
- **destruct (element)** Gets called when the given element got removed from DOM (eg. by array-splice).
- **init (element)** Gets called when an element should get initialized (eg. after creation).
- **[value] get (element, trim, empty)** Gets called whenever the value of the given element is requested. `trim` Whether the result should get trimmed. `empty` Whether considered as empty values should get saved too.
- **set (element, value, trim)** Gets called whenever the value of the given element should be set to given one. `trim` Whether the value is assumed as trimmed.

For further impression take a look at the [default plugins](#).

You should also take a look at the helper-functions within [Settings](#) in order to create your own plugins. There are a few methods that take response to call the methods of other plugins when fittingly.



---

---

## 8.1 Writing Widgets for NodeBB

See the original [blog post](#) for a high level overview and screenshots of the widget system.

### 8.1.1 Embedding HTML and JavaScript

You don't need to be a developer to figure this out. Head over to the Themes control panel and click on the Widgets tab. Create a new HTML widget by dragging and dropping the widget onto whatever template you want.

Copy and paste HTML or JavaScript into the widget and hit save - you're done!

You can optionally give your widget a container by dragging and dropping from the containers section onto your selected widget.

If you're looking for some sample scripts, head over to our [plugins section](#) and look for any topic labelled `nodebb-script-xyz`. Don't forget to submit your scripts and ideas as well!

### 8.1.2 Creating Widgets

You can define widgets in both plugins and themes. If you're building a plugin which simply delivers a widget (or collection of widgets), we strongly suggest you follow the `nodebb-widget-xyz` nomenclature instead when publishing.

#### Registering your widget

Listen to this hook to register your widget:

```
"hook": "filter:widgets.getWidgets", "method": "defineWidgets", "callbacked": true
```

Pass this back in the array:

`Content` defines the form that is displayed to customize your widget in the admin panel.

#### Listening to your widget

NodeBB core will call your widget on the appropriate page load by way of the hooks system. The hook will be named after your widget's namespace (see previous example) - like so: `filter:widget.render:widget_namespace`

This will pass in an object with the following useful properties:

- `obj.area` - will have `location`, `template`, `url`
- `obj.data` - will have your admin-defined data; in the example from the previous section you will be exposed an `obj.data.myKey`

### 8.1.3 Defining Widget Areas in Themes

A Widget Area is characterized by a template and a location. Themes can share widgets if they define the same Widget Areas. If an admin switches themes, widgets that were previously defined in a Widget Area incompatible with the new theme are saved.

Listen to this hook to register your Widget Area:

```
"hook": "filter:widgets.getAreas", "method": "defineWidgetAreas", "callbacked": true
```

Pass this back in the array:

```
{
  name: "Category Sidebar",
  template: "category.tpl",
  location: "sidebar"
}
```

And that's all. You can define as many Widget Areas in your theme as you wish. If you're still stuck, have a look at [this commit](#) which upgraded the Cerulean theme to use the widget system.

---

---

## 9.1 Creating a new NodeBB Theme

NodeBB is built on Twitter Bootstrap, which makes theming incredibly simple.

### 9.1.1 Packaging for NodeBB

NodeBB expects any installed themes to be installed via `npm`. Each individual theme is an `npm` package, and users can install themes through the command line, ex.:

```
npm install nodebb-theme-modern-ui
```

The theme's folder must contain at least two files for it to be a valid theme:

1. `theme.json`
2. `theme.less`

`theme.less` is where your theme's styles will reside. NodeBB expects LESS to be present in this file, and will precompile it down to CSS on-demand. For more information regarding LESS, take a look at [the project homepage](#).

**Note:** A *suggested* organization for `theme.less` is to `@import` multiple smaller files instead of placing all of the styles in the main `theme.less` file.

### 9.1.2 Configuration

The theme configuration file is a simple JSON string containing all appropriate meta data regarding the theme. Please take note of the following properties:

- `id`: A unique id for a theme (e.g. "my-theme")
- `name`: A user-friendly name for the theme (e.g. "My Theme")
- `description`: A one/two line description about the theme (e.g. "This is the theme I made for my personal NodeBB")
- `screenshot`: A filename (in the same folder) that is a preview image (ideally, 370x250, or an aspect ratio of 1.48:1)
- `url`: A fully qualified URL linking back to the theme's homepage/project

### 9.1.3 Child Themes

If your theme is based off of another theme, simply modify your LESS files to point to the other theme as a base:

#### topic.less

As `topic.less` from the theme `nodebb-theme-vanilla` was imported, those styles are automatically incorporated into your theme.

**Important:** If you depend on another theme, make sure that your theme specifically states this in its `package.json`. For example, for the above theme, as we depend on `nodebb-theme-vanilla`, we would explicitly state this by adding a new section into the `package.json` file:

```
"peerDependencies": {
  "nodebb-theme-vanilla": "~0.0.1"
}
```

## 9.2 Rendering Engine

### 9.2.1 How it works

Every page has an associated API call, Template file, and Language File.

For example, if you navigate to `/topic/351/nodebb-wiki`, the application will load three resources. The API return `/api/topic/351/nodebb-wiki` and the template, in this example, `topic.tpl`, and the appropriate language file `topic.json`\*.

Just prepend `api/` to the URL's path name to discover the JSON return. Any value in that return can be utilized in your template.

\*A page's name corresponds to the template and language's filename (ex. `http://domain.com/topic/xyz` correlates to `topic.tpl`). Sometimes this is not the case - ex. `/user/xyz` loads `account.tpl`. Have a look at the `custom_mapping` section in `public/templates/config.json` for more details.

### 9.2.2 Templating Basics

Using the API return as your guide, you can utilize any of those values in your template/logic. Using the above API call as an example, for anything in the root level of the return you can do something like:

```
{topic_name}
```

To access values in objects:

```
{privileges.read}
```

And finally you can loop through arrays and create blocks like so:

```
<!-- BEGIN posts -->
{posts.content}
<!-- END posts -->
```

The above will create X copies of the above block, for each item in the `posts` array.

## 9.2.3 Templating Logic

NodeBB's templating system implements some basic logic. Using the same API call as above for our example. You can write IF conditionals like so:

```
<!-- IF unreplied -->
This thread is unreplied!
<!-- ENDIF unreplied -->
```

Another example:

```
<!-- IF !disableSocialButtons -->
<button>Share on Facebook</button>
<!-- ELSE -->
Sharing has been disabled.
<!-- ENDIF !disableSocialButtons -->
```

We can check for the length of an array like so:

```
<!-- IF posts.length -->
There be some posts
<!-- ENDIF posts.length -->
```

While looping through an array, we can check if our current index is the @first or @last like so:

```
<!-- BEGIN posts -->
  <!-- IF @first -->
    <h1>Main Author: {posts.username}</h1>
  <!-- ENDIF @first -->
  {posts.content}
  <!-- IF @last -->
    End of posts. Click here to scroll to the top.
  <!-- ENDIF @last -->
<!-- END posts -->
```

For more advanced documentation, have a look at the [templates.js](#) repository

## 9.2.4 Exposing template variables to client-side JavaScript

There are two ways of letting our JS know about data from the server-side, apart from WebSockets (TODO: will be covered in a different article).

### Via jQuery.get

If we require data from a different page we can make a \$.get call to any other API call. For example, if we wanted to know more about a specific user we could make a call like so:

```
$.get(RELATIVE_PATH + '/api/user/psychobunny', {}, function(user) {
  console.log(user)
});
```

See this API call in action: <http://community.nodebb.org/api/user/psychobunny>

### Via Template Variables

In topic.tpl for example, we can add a hidden input like so:

```
<input type="hidden" template-variable="pageCount" value="{pageCount}" />
```

The template system will immediately parse all of these and expose them via the following method:

```
ajaxify.variables.get('pageCount');
```

This is the ideal method of letting JS know about important variables within the template.

### 9.2.5 Internationalization

The template engine interfaces with the internationalization system as well. We can embed variables into language strings. Let's use [this API call](#) as well as [this language file](#) as an example. We can now do something like the following:

```
[[register:help.username_restrictions, {minimumUsernameLength}, {maximumUsernameLength}]]
```

Which will translate this string:

```
A unique username between %1 and %2 characters
```

to

```
A unique username between 2 and 16 characters
```

### 9.2.6 Advanced Topics

#### Dynamically requiring and rendering a template file from client-side JavaScript

The template engine lazy loads templates on an as-needed basis and caches them. If your code requires a template or partial on-demand then you can :

```
ajaxify.loadTemplate('myTemplate', function(myTemplate) {
    var html = templates.parse(myTemplate, myData);
});
```

You can also access the individual blocks inside each template, which is handy for doing things like (for example) rendering a new post's `<li>` and dynamically sticking it in an already loaded `<ul>`

```
Some stuff here...
<!-- BEGIN posts -->
We just want to pull this block only.
<!-- END posts -->
... some stuff here

ajaxify.loadTemplate('myTemplate', function(myTemplate) {
    var block = templates.getBlock(myTemplate, 'posts');
    var html = templates.parse(block, myData);
});
```

#### Rendering templates on server-side Node.js

The templating system hooks into Express just like most other templating frameworks. Just use either `app.render` or `res.render` to parse the appropriate template.

```
res.render('myTemplate', myData);
```

```
app.render('myTemplate', myData, function(err, parsedTemplate) {  
    console.log(parsedTemplate);  
});
```



## 10.1

---

: This section is under construction.

---

### 10.1.1 Core

- [Building a new Admin Page \(Out of date\)](#)

### 10.1.2 Plugins

- [Developing plugins with Grunt and Coffeescript](#)
- [Writing your first NodeBB plugin](#)

### 10.1.3 Themes

### 10.1.4 Widgets



## 11.1 Helping out the NodeBB Project

NodeBB is an open source project, and will forever remain free. Here's a number of ways you can help us, even if you aren't a programmer.

- [Like and share our content on Facebook](#)
- [Follow us on Twitter](#) and perhaps tweet **#NodeBB is most awesome forum software @NodeBB**
- Update our wiki! ;) We need everything from development/design tutorials to user friendly how-to guides.
- Tell everybody about NodeBB, including your grandma and her cats.
- [Submit a pull request, or two, or three..](#)
- Build a new theme
- Write a plugin
- Keep the link back to us on the footer of your own NodeBB :)
- Blog about us! Give the gift of SEO juice this Christmas
- [Help Translate NodeBB](#) - It's a really simple translation tool and you don't need to know how to code.
- Join our [community](#) and give us a hard time about bugs and missing features

### 11.1.1 Translating NodeBB to another language

NodeBB uses Transifex, which is a user friendly visual tool which allows any individual to translate text into a language of their choice. You don't need to be a programmer to do this, so what are you waiting for? [Join the translation team now](#) :)

### 11.1.2 Writing Documentation

These docs were written using [Sphinx](#) and published using [rtfd.org](#).

You can edit these docs [directly on GitHub](#), or by clicking on "View page source" on the top right of any page.

If you wish, you can clone the repository and compile the documentation yourself. Check out the [Getting Started](#) section for more info on how to accomplish the latter.

Documentation are auto-compiled and pushed to [rtfd.org](#) after every commit.



---

## 12.1 Need Help?

### 12.1.1 Frequently Asked Questions

If you experience difficulties setting up a NodeBB instance, perhaps one of the following may help.

#### How do I start/stop/restart NodeBB?

You can call the `./nodebb` executable to start and stop NodeBB:

```
$ ./nodebb
Welcome to NodeBB
Usage: ./nodebb {start|stop|reload|restart|log|setup|reset|upgrade|dev|watch}

start    Start the NodeBB server
stop     Stops the NodeBB server
reload   Restarts NodeBB
restart  Restarts NodeBB
log      Opens the logging interface (useful for debugging)
setup    Runs the NodeBB setup script
reset    Disables all plugins, restores the default theme.
upgrade  Run NodeBB upgrade scripts, ensure packages are up-to-date
dev      Start NodeBB in interactive development mode
watch    Start NodeBB in development mode and watch for changes
```

#### How do I upgrade my NodeBB?

Please consult *Upgrading NodeBB*

#### I upgraded NodeBB and now X isn't working properly!

Please consult *Upgrading NodeBB*

#### I installed an incompatible plugin, and now my forum won't start!

If you know which plugin caused problems, disable it by running: `./nodebb reset plugin="nodebb-plugin-pluginName"`

Otherwise, disable all plugins by running: `./nodebb reset plugins`

### Is it possible to install NodeBB via FTP?

It is possible to transfer the files to your remote server using FTP, but you do require shell access to the server in order to actually “start” NodeBB. Here is a [handy guide for installing NodeBB on DigitalOcean](#)

### I’m getting an “npm ERR!” error

For the most part, errors involving npm are due to Node.js being outdated. If you see an error similar to this one while running `npm install`:

```
npm ERR! Unsupported
npm ERR! Not compatible with your version of node/npm: connect@2.7.11
```

You’ll need to update your Node.js version to 0.8 or higher.

To do this on Ubuntu:

```
# add-apt-repository ppa:chris-lea/node.js
# apt-get update && apt-get dist-upgrade -y
```

If successful, running the following command should show a version higher than 0.8

```
# apt-cache policy nodejs
```

### URLs on my NodeBB (or emails) still have the port number in them!

If you are using *nginx* or *Apache* as a reverse proxy, you don’t need the port to be shown. Simply run `.nodebb setup` and specify the base URL without a port number.

Alternatively, edit the `config.json` file using your favourite text editor and change `use_port` to `false`.

## 12.1.2 Submit Bugs on our Issue Tracker

Before reporting bugs, please ensure that the issue has not already been filed on our [tracker](#), or has already been resolved on our [support forum](#). If it has not been filed, feel free to create an account on GitHub and [create a new issue](#).

## 12.1.3 Ask the NodeBB Community

Having trouble installing NodeBB? Or did something break? Don’t hesitate to join our forum and ask for help. Hopefully one day you’ll be able to help others too :)

- *genindex*
- *modindex*
- *search*