

Server Setup for IM with NodeJS

This tutorial requires advanced knowledge. We recommend you get professionals to do this if you are not at an advanced level.

Requires: *phpFox version >= 4.5.2*

The IM app requires NodeJS and Redis. In this article, we will go over how to setup NodeJS and Redis on the Linux Flavor, Ubuntu. We recommend that you separate your NodeJS, Redis and HTTP server from each other. Your HTTP server is the main web server that has your phpFox source code.

Install NodeJS and Redis

You can choose one of 2 following solutions:

Use Hosting Services

There are cloud-based services where you can host your NodeJS code and Redis database:

- [NodeJS Host](#)
- [Redis Host](#)

By using these services, you don't need to install NodeJS and Redis as they are installed on servers by default. You can move to "Configure NodeJS and Redis" step

Install NodeJS and Redis on Your Own Servers

If you decide to setup NodeJS and Redis on your own web environments, let's first set up NodeJS.

Connect to your NodeJS server via SSH and run the command

```
sudo apt-get install nodejs
sudo ln -s /usr/bin/nodejs /usr/bin/node
sudo apt-get install npm
```

Next, we need to install Redis.

```
sudo apt-get install redis-server
```

Configure NodeJS and Redis

At this point your NodeJS and Redis are installed on your server. We now continue setting up them.

Connect FTP/SSH to the server and upload the folder /PF.Site/Apps/core-im/server/ to a secure directory on your server that does not have HTTP access to it. You can find this folder in the phpFox ZIP package you get when you downloaded our product.

Via SSH, navigate to the folder you just uploaded. You can use the CD command to do this. Example:

```
cd /some/folder
```

Once you reached your NodeJS script you need to install all the libraries required to run the script.

```
npm update
npm install
```

Next, rename the sample config file.

```
mv config.js.new config.js
```

Update the config file with your information:

- **chat_server_key**: Can use any string value as server key for this param. Then you will need to configure the same key to **Node JS server key** setting of IM app in AdminCP

- **redis**: information of your Redis server: host, port, and password

```
module.exports = {
  port: 3000,
  chat_server_key: "your-chat-server-key",
  secure: false, // use ssl ?
  privateKey: "./private_key.key", // path to private key ?
  cert: "./cert.crt", // path to cert key ?
  redis: {
    host: '127.0.0.1',
    port: 6379,
    password: ''
  }
};
```

SSL Configurations

Please skip this step if your phpFox site doesn't support SSL.

It requires your Chat supports SSL also if you want to use IM app on an SSL site. So, make sure that you have enabled SSL before doing the below step: Open the file **config.js**, set **true** for a **secure** property, you also need to provide the path of **private key** file and **cert key** file, an example of the change:

```
module.exports = {
  port: 3000,
  chat_server_key: "your-chat-server-key",
  secure: true, // use ssl ?
  privateKey: "./private_key.key", // path to private key ?
  cert: "./cert.crt", // path to cert key ?
  redis: {
    host: '127.0.0.1',
    port: 6379
  }
};
```

Note: In case you are using **Nginx** server, you can add below configurations to your server instead of updating file **config.js** as above. Don't forget to update the path for your **certificate** and **private key** files. (For many distributions, the file will be located at `/etc/nginx/nginx.conf`. If it does not exist there, it may also be at `/usr/local/nginx/conf/nginx.conf` or `/usr/local/etc/nginx/nginx.conf`)

```
# your SSL configuration
#ssl on;
#ssl_certificate /path/to/ssl_cert.crt;
#ssl_certificate_key /path/to/ssl_private_key.key;

location /socket.io/ {
  # switch off logging
  access_log off;

  # redirect all HTTP traffic to 127.0.0.1:3000
  proxy_pass 127.0.0.1;
  proxy_set_header X-Real-IP $remote_addr;
  proxy_set_header Host $host;
  proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

  # WebSocket support (nginx 1.4)
  proxy_http_version 1.1;
  proxy_set_header Upgrade $http_upgrade;
  proxy_set_header Connection "upgrade";
}
```

Complete Setup of NodeJS and Redis

Congratulation! Chat server has been set up completely, run below command to start it:

```
node index.js
```

Enabling from AdminCP

With both NodeJS and Redis installed, you can now visit your AdminCP and head over to the IM App: **AdminCP > Apps > Instant Messaging > Settings page**

For the **Provide your Node JS server** setting enter: <http://your-nodejs-server.com:3000> or <https://your-nodejs-server.com:3000>

- If your site is using SSL, you need to configure this setting with domain name of NodeJS server instead of IP address.

In **Provide your Node JS server key**, enter the value of *chat_server_key* that you have configured in the *config.js* file.

Make sure to change [your-nodejs-server.com](#) with the IP or Domain that leads to your NodeJS server. For SSL case, you will need to configure with the domain only.

Congratulations you have successfully set up the IM App.

To use the IM, [see this tutorial](#).