

Manual Installation

Tim Morgan edited this page 23 days ago · 29 revisions

Operating System: Linux (Debian/Ubuntu recommended)

This document will walk you through all the steps to install OneBody on a Ubuntu Linux server with Apache, MySQL, Ruby, and some other stuff. If you'd rather serve the app with Nginx, you can still follow a lot of the steps here, but also see [Serving with Nginx and Passenger](#).

Beware

First, a word of warning: OneBody is not a PHP app that can be FTP'd to a cheap web host. You *will* need a dedicated Linux server (or Virtual Private Server) with 1 Gb of memory and *root shell access*.

If you don't feel comfortable logging into a web server from the console, typing lots of commands, reading the output, Googling error messages, and generally troubleshooting the whole process, then please don't try this. If you don't have an hour to spare, then don't try to rush it, as it will only end in frustration—it would be better for you find a consultant on freelancer.com or oDesk and pay them to do it.

On the other hand, if you're a I.T/sysadmin/hacker with guts, then please proceed!

Using Sudo

If you're logged in as the `root` user, then everywhere you see `sudo` below, you can pretend it's not there (don't type it).

Let's Go!

1. Get a Linux server with Ubuntu Linux 14.04 LTS on it. A newer version of Ubuntu will likely work. Debian should work, but extra googling may be required.

If you are using RedHat/CentOS, SuSE or something else "enterprise" worthy, you're kinda on your own.

2. Install all this software first:

```
sudo apt-get install -y vim build-essential curl \
libreadline-dev libcurl4-openssl-dev \
nodejs git mysql-server libmysqlclient-dev \
libaprutil1-dev libapr1-dev \
apache2 apache2-threaded-dev libapache2-mod-xsendfile \
imagemagick
```

That command is for Debian/Ubuntu; if you're using another Linux distro, you'll need to find and install the same packages (possibly different names) using your distro's package management tool.

On Debian, there isn't a "nodejs" package, but you can [install Node.js by following these steps from Joyent](#).

3. Install Ruby 2.1.2 or higher. Brightbox has a Ruby package for Ubuntu, which you can

Pages 34

Find a Page...

[Home](#)

[Amazon AMI](#)

[API](#)

[Authentication API](#)

[Backups](#)

[Batch People](#)

[Check In](#)

[Churches Using OneBody](#)

[Commercial Hosting Inquiries](#)

[Crontab Setup](#)

[Deployment with Capistrano](#)

[Email Setup](#)

[FAQs for Contributors](#)

[Frequently Asked Questions](#)

[Hashify Families](#)

Show 19 more pages...

Clone this wiki locally

<https://github.com/churchio/onebody/wiki/Manual-Installation> 

 Clone in Desktop

RootFile

install like this:

```
sudo apt-get install -y software-properties-common
sudo apt-add-repository -y ppa:brightbox/ruby-ng
sudo apt-get update
sudo apt-get install -y ruby2.1 ruby2.1-dev
```

RootFile

If you're on Debian, you can follow [the instructions here](#) for compiling Ruby from source.

Whatever method you choose, **don't move on from this step** until you can type `ruby --version` and see "ruby 2.1.2" in your console.

4. Now, let's get the OneBody source code!

```
cd /var/www
git clone git://github.com/churchio/onebody.git
cd onebody
```

RootFile

This will put the latest and greatest, bleeding edge OneBody code in the `/var/www/onebody` directory.

Now, it's recommended you switch to a tagged release of OneBody, which you can do with the command `git checkout 3.4.0`.

5. Make sure Apache will be able to write tmp files and logs and such:

```
mkdir -p tmp/pids log public/system
chmod -R 777 tmp log public/system
```

RootFile

6. Now, create your database:

```
mysql -u root -e "create database onebody default character set utf8 default co
```

If you get an error about access being denied, then you may need to use `mysql -u root -p -e "..."` and enter your root password.

You'll notice we set the username and password to "onebody" and "onebody". That is ok, as long as you: 1) trust all the users logging into this Linux server (or you're the only one), and 2) do not grant access to users outside of localhost (notice the `onebody@localhost` part). If you cannot answer yes to both of those questions, then please change the password to something else (you'll just need to change it also in the `config/database.yml` file in the next step) and make sure `/var/www/onebody` is not accessible to those devious users you let on your server. :-)

7. `cp config/database.yml{.example,}`

DeclareVariable
RootFile

Note: in the next release (4.4.0), this will change to `cp config/database.yml{.mysql-example,}`. I'm putting this note here while I'm thinking about it in case I forget to update this page whenever 4.4.0 is released. :-)

If you used something other than "onebody" for your MySQL password, then change it appropriately in this file using vim or nano or another text editor.

8. Install the bundler gem and then use bundler to install all OneBody gem dependencies:

```
sudo gem install bundler
bundle install
```

Do not continue if you get an error saying something could not be

downloaded/compiled/installed. There may be other development packages that need to be installed here, like "libxml", so read the errors carefully, install the needed stuff, then come back and run this command again.

9. `cp config/secrets.yml{.example,}` then edit the `config/secrets.yml` file and **add a random secret token** to the "production" section.

(You can use `bundle exec rake secret` to generate a new random secret that you can copy and paste into the `secrets.yml` file, or just make up something really long and random by smashing your head on the keyboard.)

10. **`RAILS_ENV=production bundle exec rake db:migrate db:seed`**

Watch the output! Don't move on if you see an error on the screen.

11. `RAILS_ENV=production bundle exec rake assets:precompile` to prepare all the CSS and JavaScript files.

If this step fails complaining about lack of JavaScript runtime, you should make sure you have Node.js installed. You should have installed Node.js in step 2.

12. Add the [Passenger APT repository](#) from Phusion (follow directions at that link) and then run:

```
sudo apt-get update
sudo apt-get install libapache2-mod-passenger
sudo a2enmod passenger
sudo service apache2 restart
```

Then, to make sure all that worked, run the following command:

```
apachectl -M
```

You should see `passenger_module` in the list. If you don't, go back, read the Passenger documentation, and try again.

13. Next we need to edit the default vhost (or create a new vhost) and point the `DocumentRoot` to the `onebody/public` folder.

For Ubuntu/Debian, that looks something like this: `vim /etc/apache2/sites-available/default` or `vim /etc/apache2/sites-available/000-default.conf` and **change the `DocumentRoot` to be `/var/www/onebody/public`.**

Next you need to add these two lines to the config: `XSendFile On` and `XSendFilePath /var/www/onebody/public/system`. So, to recap, you need to have the following three lines in your Apache vhost (not necessarily right next to each other):

```
DocumentRoot /var/www/onebody/public
XSendFile On
XSendFilePath /var/www/onebody/public/system
```

14. Now enable the `xsendfile` Apache module and restart Apache:

```
sudo a2enmod xsendfile
sudo service apache2 restart
```

Now, if you did all that right, you probably have OneBody running on your host. You will want to map a domain to your host with DNS, but to test it out, type `http://YOUR_IP_HERE` into a browser.

If nothing comes up, you'll need to troubleshoot a few things (not in the scope of this document), such as firewall (iptables), is that the right *public* IP address, etc. God be

RootFile
InsertIntoFile

RootFile

ChangeFileRegex

InsertIntoFile

RootFile

with you!

Oh, and a few more things:

15. Pick a path in the [Email Setup](#) page to set up incoming and outgoing email.

Incoming and outgoing email is a **must** for group interaction and person-to-person communication. Also, since signing up as a new user sends an email to admins, **sign-up will fail with a big ugly error message for new users** until this is complete.

16. Write the user crontab:

```
RAILS_ENV=production bundle exec whenever -w
```

This is necessary for a whole lot of things (you can see what it wrote by typing `crontab -l`) such as incoming email, group membership updates, news feed imports, etc.

17. Set up DNS on your domain so you can use `members.mychurch.org` instead of `111.222.333.444`.

You need an `A` record pointing your domain to your IP address. It's pretty simple.

You'll also need an `MX` record for incoming email. See [Email Setup](#) for help with that.

18. Get an SSL certificate and:

- i. Set up Apache to only serve OneBody on SSL port 443.
- ii. Redirect non-SSL traffic to the secure site.

Whew! We know that was a lot. If you made it this far, and OneBody is running, then congratulations!

What's next? Complete the form on the initial Setup screen, then head over to the Settings page in the Admin dashboard, and start customizing!

Troubleshooting

I only see a listing of files when I visit my site.

Passenger isn't installed properly. Go back to steps 12 and 13 and try again. You might need to consult the Passenger install docs for further help.

I cannot upload photos.

You need to create the `public/system`, `tmp` and `log` directories and make sure they are writable:

```
cd /var/www/onebody
mkdir tmp log public/system
sudo chmod -R 777 tmp log public/system
```

RootFile

Also ensure that ImageMagick is installed:

```
sudo apt-get install imagemagick
```

When new users try to sign up, they get an ugly error.

You need to set up your email server for outgoing email. Since signing up sends an request email to admins, and you don't have a mail server, Rails freaks out.

How to Upgrade

```
cd /var/www/onebody
git fetch origin
git checkout 3.4.0
bundle install
RAILS_ENV=production bundle exec rake db:migrate
RAILS_ENV=production bundle exec rake tmp:clear assets:precompile
touch tmp/restart.txt
```

RootFile

(Replace 3.4.0 with the version you're upgrading to. You can see the latest releases [here](#).)

Version-Specific Upgrade Information

Upgrading from 3.3.0 to 3.4.0

This is the first version that started using `bundle install` instead of `bundle install --deployment`.

To make this work, `rm -rf /var/www/onebody/vendor/bundle` and then run `bundle install` again so Rails can find the gems.

RootFile

If that doesn't work, run `bundle install --no-deployment && bundle update && bundle install` (see [here](#)).

We've had many reports of messed up permissions on the `tmp/cache` folder. This will cause an error when trying to view the admin dashboard. To fix, run: `chmod -R 777 /var/www/onebody/tmp`.

Upgrading from 3.2.0 to 3.3.0

No additional instructions

Upgrading from 3.1.0 to 3.2.0

1. Be sure you are upgrading from a OneBody version of 3.0.0 or later. If you are upgrading from a version in the 2.x series, you will need to *first* completely upgrade to 3.0.0, *then* upgrade to this version.
2. Follow upgrade instructions above, then:
 - i. Set your "Default Country" in the admin dashboard Settings screen.
 - ii. Run the following rake task to set your country on all existing family records:

RootFile

```
RAILS_ENV=production bundle exec rake onebody:set_country
```

Upgrading from 3.0.0 to 3.1.0

You need to add `secret_key_base` to your `secrets.yml` file. See [secrets.yml.example](#) for an example.

