



# Amusewiki for system administrators

Amusewiki for system administrators

[amusewiki.org](http://amusewiki.org)



# Contents

Location . . . . .	5
Services . . . . .	5
Logs . . . . .	6
Application . . . . .	6
Nginx . . . . .	6
Database . . . . .	7
Utilities . . . . .	7



```
amusewiki generate-nginx-conf
amusewiki import-redirections
amusewiki jobber
amusewiki letsencrypt
amusewiki populate-monthly
amusewiki populate-webfonts
amusewiki reset-password
amusewiki set-site-variable
amusewiki upgrade-db
amusewiki upgrade-lexicon
```

```
amusewiki shell
```

The shell subcommand will start a shell with the `amusewiki use`

See `man amusewiki-<action>` for usage of each command or run `amusewiki <action> --help`

Other useful commands:

```
amusewiki [ start | stop | restart ]
```

start and stop actions simply call `systemctl`, while the restart performs a restart without downtime (so being a bit more useful just `systemctl restart amusewiki-web amusewiki-jobber amusewiki`

Please note that lately the `amusewiki restart` command fails to kill a process it spawns, so it needs to be killed manually (you are provided the target PID).

This guide assumes you installed Amusewiki with the recommended Debian packages, which gives a standard and recommended setup, providing a more straightforward debugging if the need arises. If you have it installed from source, reading this should still be worth of it.

## Location

File produced by Amusewiki are located in `/var/lib/amusewiki`, including the sites repositories under `/var/lib/amusewiki/repo`

You should backup this directory or mirror the repositories using git.

If you want to setup a static site, serving `/var/lib/amusewiki/repo/site-code` as document root will do (just add an `index.html` pointing to `titles.html`).

## Services

The process table for the user running amusewiki should look like this:

```
# ps -u amusewiki -U amusewiki f
  PID TTY          STAT       TIME COMMAND
26702 ?           Ss          0:02 perl-fcgi-pm
26722 ?           S           18:16 \_ perl-fcgi
   3268 ?           S           16:49 \_ perl-fcgi
15818 ?           S           15:33 \_ perl-fcgi
15980 ?           S           15:19 \_ perl-fcgi
29350 ?           S            5:43 \_ perl-fcgi
26698 ?           Ss          1:18 /usr/bin/perl /usr/bin/amusewiki-jobber
```

The first process (in this example with PID 26702) is the FCGI process manager, while its children are the workers for the application itself. The web pages are served via Nginx as reverse proxy.

The last process (in this example with PID 26698) is the background daemon in charge for the publishing, indexing, PDF compilation. When active, you will see it spawning forks, while when not under load you should see just one of it.

These two services can operate independently and are controlled by two systemd unit: `amusewiki-web` and `amusewiki-jobber` so you can use standard systemd commands like `systemctl status` etc. here.

There is also a 3rd process, `amusewiki-cgit`, which is in charge to run `cgit` on localhost (which Amusewiki serves when you look at the history of a page). You are not supposed to have much to do with this service. When Amusewiki is installed from source, this service may or may be not be present, it could be Nginx directly serving that.

## Logs

### Application

By default, the application logs are left in `/var/lib/amusewiki/log/amw-201` with the timestamp on the filename (so each day is a new log). Errors are instead mailed directly to `root@localhost`.

The configuration file for the logging is `/var/lib/amusewiki/log4perl.conf` and explain what to do if you want to change that.

This is how a `log4perl.local.conf` could look like:

```
log4perl.logger=INFO,Log
log4perl.logger.main=INFO
# debug individual modules
log4perl.logger.AmuseWikiFarm.Controller.Edit=DEBUG
log4perl.logger.AmuseWikiFarm.Schema.Result.Revision=DEBUG
# define the 'Log' logger
log4perl.appender.Log=Log::Dispatch::File::Stamped
log4perl.appender.Log.filename=log/amw.log
log4perl.appender.Log.stamp_fmt=%Y-%m-%d
log4perl.appender.Log.mode=append
log4perl.appender.Log.layout=Log::Log4perl::Layout::PatternLayout
log4perl.appender.Log.layout.ConversionPattern=%d %p - %c - %F
```

Further reading on this topic at <https://metacpan.org/pod/Log::Log4perl> but this is more a developer tool, so the default settings should be good enough.

### Nginx

To get a grip on the load which the server is taking, you should look at the nginx logs located at `/var/log/nginx/*.log`. Each virtual host should have its own log.

For Amusewiki, Nginx is controlled by 2 files:

- `/etc/nginx/amusewikidebian_include`
- `/etc/nginx/sites-enabled/amusewikidebian`

They are usually autogenerated by the application, you are usually prompted to run the needed commands as root when e.g. installing new sites.

The same happens when calling `amusewiki generate-nginx-conf`

The master configuration file `/etc/nginx/nginx.conf` is your sole responsibility so Amusewiki never touches it. If you need to adjust logging format or global settings, this is usually the place.

## Database

Amusewiki can work with MySQL, PostgreSQL and SQLite (not recommended for production). DB installation is managed by the Debian package and should work out of the box. To find out what you are using the configuration file is `/var/lib/amusewiki/dbic.yaml` which contains the DSN and driver information.

## Utilities

If you need to run some Amusewiki command, you should look at the `amusewiki` executable.

Usage: `amusewiki action [ argument1, argument2]`

Run the `amusewiki` scripts as user `amusewiki` in the correct directory

Available commands:

```
amusewiki bootstrap-archive
amusewiki check-compiler
amusewiki clone-site
amusewiki create-doc-site
amusewiki export-titles
amusewiki generate-meta-config
```