
AgendaV Documentation

Release 2.0.0-beta1

Jorge López Pérez

August 26, 2015

1	Installation and configuration	3
1.1	Installation	3
1.2	Upgrading	6
1.3	Configuration	7
1.4	Troubleshooting AgenDAV	10
2	Translating AgenDAV	11
2.1	How to add a translation	11
3	Development	13
3.1	Virtual Machine	13
3.2	Working with scripts and stylesheets	14
4	Release notes	15
4.1	2.0.0-beta1 (2015-08-26)	15
4.2	1.2.6.1 and 1.2.6.2 (2012-10-15)	15
4.3	1.2.6 (2012-09-03)	15
4.4	1.2.5.1 (2012-06-11)	15
4.5	1.2.5 (2012-06-07)	16
4.6	1.2.4 (2012-01-16)	16
4.7	1.2.3 (2011-11-08)	16
4.8	1.2.2 (2011-10-25)	17
4.9	1.2.1 (2011-10-24)	17
4.10	1.2 (2011-10-17)	17
4.11	1.1.1 (2011-09-24)	17

AgendaV is a CalDAV web client which features an AJAX interface to allow users to manage their own calendars and shared ones. It's released under the GPLv3 license.

Contents:

Installation and configuration

1.1 Installation

In this section you will find instructions on how to install AGenDAV.

1.1.1 Prerequisites

AgenDAV 2.0.0-beta1 requires the following software to be installed:

- A CalDAV server
- A web server
- PHP $\geq 5.4.0$
- PHP ctype extension
- PHP mbstring extension
- PHP mcrypt extension
- PHP cURL extension
- A database backend

Most popular database backends are supported, such as MySQL, PostgreSQL or SQLite.

Look for supported databases on this [Doctrine DBAL driver list](#).

1.1.2 Downloading AGenDAV and uncompressing

AgenDAV 2.0.0-beta1 can be obtained at [AgenDAV webpage](#).

Uncompress it using `tar`:

```
$ tar agendav-...tar.gz
$ cd agendav-.../
```

1.1.3 Download dependencies (only for Git)

If you downloaded AGenDAV from GitHub using `git`, you will need to download [Composer](#) and use it to fetch AGenDAV dependencies.

Composer is a PHP package manager, and some libraries used by AgenDAV are downloaded using it.

Composer installation is really simple:

```
$ cd web/
$ curl -s https://getcomposer.org/installer | php
$ php composer.phar install --prefer-dist --no-dev
```

For production environments it is recommended to run the following composer command that improves loading performance:

```
$ php composer.phar dump-autoload --optimize
```

1.1.4 Database and tables

AgenDAV requires a database to store some extra information.

First of all you have to set up your database. If you plan using MySQL or PostgreSQL, here you will find some basic instructions about how to set up them.

Setting up a MySQL database

Create a user in MySQL and let it use a new *agendav* database:

```
$ mysql --default-character-set=utf8 -uroot -p
Enter password:
[...]
mysql> GRANT ALL PRIVILEGES ON agendav.* TO agendav@localhost IDENTIFIED BY 'yourpassword'
mysql> CREATE DATABASE agendav CHARACTER SET utf8 COLLATE utf8_general_ci;
mysql> FLUSH PRIVILEGES;
mysql> ^D
```

Setting up a PostgreSQL database

Use the special `postgres` system user to manage your installation. You can add a new user and a new database the following way:

```
# su postgres
$ psql
postgres=# CREATE USER agendav WITH PASSWORD 'somepassword';
postgres=# CREATE DATABASE agendav ENCODING 'UTF8';
postgres=# GRANT ALL PRIVILEGES ON DATABASE agendav TO agendav;
postgres=# \q
$ exit
```

Then you have to edit the file `pg_hba.conf`, which is usually located at `/var/lib/pgsql/`. Add the following line before other definitions:

```
# TYPE DATABASE USER CIDR-ADDRESS METHOD
local agendav agendav md5
```

Setting up a SQLite database

SQLite is not recommended for production environments, but will be more than enough for testing and single user environments.

You will need a dedicated directory for the database:

```
# mkdir database
# touch database/agendav.sqlite
# chown -R www-data:www-data database/
```

Create AgenDAV tables

AgenDAV tables are created by running the provided `agendavcli` script.

Before being able to run it, you will have to configure at least your database connection details. Have a look at the `db.options` parameter.

After configuring your database connection, just run the script like this:

```
$ cd bin/
$ php agendavcli migrations:migrate
```

Confirm the operation, and your database should be ready.

1.1.5 Configuring Apache to serve AgenDAV

Apache has to be configured to point to `web/public` directory.

Example using a dedicated virtualhost:

```
<VirtualHost 1.2.3.4:443>
  ServerAdmin admin@email.host
  DocumentRoot /path/to/agendav/web/public
  ServerName agendav.host
  ErrorLog logs/agendav_error_log
  CustomLog logs/agendav_access_log common

  <Location />
    RewriteEngine On
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteRule ^ index.php [QSA,L]
  </Location>
</VirtualHost>
```

Note: Make sure that you have the following PHP settings *disabled*:

- `magic_quotes_gpc`
- `magic_quotes_runtime`

You can enable development mode by following the instructions at [Development environment](#).

Other web servers

AgenDAV should run on any other web server software as well. Just read the [Silex Webserver configuration guide](#).

1.1.6 Change directory permissions

You should change the owner and group for all AgenDAV files to the ones your webserver uses. Make sure you allow your webserver user to write on the `var/` directory. The following example assumes your webserver runs as `www-data` user and `www-data` group:

```
# chown -R www-data:www-data web/
# chmod -R 750 web/var/
```

1.1.7 Configure AgenDAV

Now you can proceed to fully configure AgenDAV following the [Configuration](#) section.

1.2 Upgrading

Before starting this process, **make sure you have a backup** of your current AgenDAV directory, specially the `web/config/` directory, and a dump of your database schema and contents.

Please, do not continue unless you have both backups.

Read all the [Release notes](#) starting at the version you are currently using, because some important changes may have happened. Apply those changes after updating the files from AgenDAV.

1.2.1 Upgrading from 1.x.x

If you are upgrading AgenDAV from 1.x.x, make sure you have the latest 1.x release installed.

Warning: Current AgenDAV development version will remove all your current shares from the database. This will be fixed on next stable release

After that, just follow the steps below.

1.2.2 Upgrade AgenDAV code

a) Updating from a tar.gz file

After downloading the new tar.gz file and uncompressing it, copy your configuration files from the old directory:

```
$ cp -a /path/to/old_agendav/web/config/settings.php \  
/path/to/new/agendav/web/config/
```

This will only work if you are upgrading from AgenDAV 2.x, as older releases used different configuration files.

b) Updating from git

If you downloaded AgenDAV from the git repository at GitHub then you can checkout latest stable release from the `master` branch, or an specific version using its tag.

Just pull latest changes and checkout the release you want. For example, checking out AgenDAV 2.0.0 can be achieved with:

```
$ git pull  
[...]  
$ git checkout 2.0.0
```

Next step is downloading latest AgenDAV dependencies using Composer. If you already have Composer installed, just run:

```
$ cd web/  
$ composer install
```

If you are upgrading from AgenDAV 1.2.x, you will need to install Composer. Follow the instructions you'll find in the installation section.

1.2.3 Database upgrade

The database upgrade process included in AgenDAV lets you apply the latest schema changes without having to deal with `.sql` files and with no need to check which files you should apply to your current version.

Follow the guide at [Configuration](#) to create a new `settings.php` file inside `web/config` which contains at least the database connection details.

Once you have your database configuration prepared, run the provided `bin/agendavcli` script this way:

```
$ php bin/agendavcli migrations:migrate
```

Warning: This development version will remove all your current shares from the database. This will be fixed on next stable release

1.2.4 Clear sessions and caches

It is recommended to remove all active sessions. Do it by running the following command:

```
$ php bin/agendavcli sessions:clear
```

If you are running AgenDAV on a production environment, you should clear several caches:

- Remove the contents of the `_twig_` cache directory. The cache path is configured using the option `twig.options` on your `settings.php` file. If you did not override the default value, it should be found at `web/var/cache/twig/` subdirectory:

```
$ rm -rf web/var/cache/twig/*
```

1.3 Configuration

Configuring AgenDAV requires creating a `settings.php` file in the `web/config/` directory.

You will find all available settings inside the file `default.settings.php`. Please, do not modify `default.settings.php`, as any changes will be lost if you upgrade AgenDAV.

Save a copy of the `default.settings.php` file as `settings.php`, or just copy the settings you want to modify, and start configuring your instance.

site.title

Title of every page

site.logo

Image filename which will be used as a logo. Has to be a valid filename placed inside `web/public/img/` directory.

site.footer

Text to be placed in the footer.

proxies

Array of IPs of trusted proxies, on which the `HTTP_X_FORWARDED_FOR` header will be honored.

db.options

Database connection parameters. Uses Doctrine DBAL syntax, so follow the guide at <http://docs.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/configuration.html> for a better reference. Example:

```
$app['db.options'] = [
    'dbname' => 'agendav',
    'user' => 'user',
    'password' => 'password',
    'host' => 'localhost',
    'driver' => 'pdo_mysql',
];
```

Warning: When using an SQLite database, note that there is a bug when using URL based configurations. Use the alternative syntax (*path* and *driver*) instead.

encryption.key

Key that will be used to encrypt passwords when storing them on the database, so use a long one.

csrf.secret

Name that the CSRF parameter will have.

log.path

Full path where logs will be created. Add a trailing slash. Example:

```
$app['log.path'] = '/var/log/agendav/';
```

Make sure the user that runs your web server has write permission on that directory.

caldav.baseurl

Base CalDAV URL used to build all CalDAV URLs.

Examples:

```
// SabreDAV
$app['caldav_base_url'] = 'http://caldav.server/cal.php';

// DAViCal
$app['caldav_base_url'] = 'http://caldav.server/caldav.php';
```

caldav.authmethod

You have to specify which HTTP authentication method does your CalDAV server require. Both basic and digest authentication methods are supported.

Example:

```
// SabreDAV
$app['caldav.authmethod'] = 'digest';

// DAViCal
$app['caldav.authmethod'] = 'basic';
```

caldav.publicurls

Whether to show CalDAV URL links or not in the edit dialog

See also:

[*caldav.baseurl.public*](#)

caldav.baseurl.public

The base URL that will be shown to users if *caldav.publicurls* is enabled. It's not used for internal connections.

Do not add a trailing slash.

Example:

```
$app['caldav.baseurl.public'] = 'https://public.caldav.tld';
```

calendar.sharing

Enables calendar sharing between users

Note that calendar sharing requires full WebDAV ACL support on your CalDAV server. Some servers, such as SabreDAV (at least on current release, 1.5.6), don't have full ACL support, so you should set this option to `false` if your server can't handle ACLs.

defaults.timezone

Timezone to be used by default.

Make sure you use a valid timezone from <http://php.net/timezones>

defaults.language

Default language to be used on AgendaV interface.

Have a look at directory `web/lang` for a list of available languages.

defaults.time_format

Preferred time format: 12 hours (e.g. 3pm / 2:30am) or 24 hours (e.g. 15:00 / 2:30).

Set this option using a **string** ('12' or '24').

defaults.date_format

Default date format.

Available options are:

- ymd: e.g. 2011/10/22
- dmy: e.g. 22/10/2011
- mdy: e.g. 10/22/2011

defaults.weekstart

Which day should be considered the default first day of week.

0 means Sunday, 1 means Monday

Specify it as a number.

logout.redirection

When logging out from AgendaV, the URL the user will be redirected to.

Can be left empty to redirect user to login page again.

1.3.1 Sessions

AgendaV uses `php.ini session settings`. You can override most of them inside `settings.php` by using the `session.storage.options`. Just copy it from `prod.php` and set any parameters you wish.

The following example makes sessions expire after 20 minutes of closing your browser in a low traffic instance:

```
$app['session.storage.options'] = [
    'name' => 'agendav_sess',
    'cookie_lifetime' => 0,
    // Every request has 10% chance of triggering session GC
    'gc_probability' => 1,
    'gc_divisor' => 10,
```

```
'gc_maxlifetime' => 1200,  
'lifetime' => 1200,  
];
```

1.4 Troubleshooting AgendaV

If you are having problems with AgendaV, check you have met all the requisites and search AgendaV logs/web server logs for error lines.

You can write to [AgendaV general list](#) asking for help. Make sure you include the following information:

- Software details (OS, PHP version, web server you're using, CalDAV server)
- Clear description of your problem
- Important log lines

Try the following before writing:

1.4.1 Development environment

You can switch to development environment easily by setting the environment variable `AGENDA_V_ENVIRONMENT` to `dev`.

Environment variables have to be set on your webservice configuration file. Apache lets you do it using `SetEnv`, or even better, using `SetEnvIf` to enable the development environment just for some IPs. Example:

```
<Location />  
  SetEnvIf Remote_Addr ^1\.2\.3\.4$ AGENDA_V_ENVIRONMENT=dev  
</Location>
```

Then point your browser to `http://your.agendav.host/`. A debugging toolbar will appear, logs will be more verbose and a new HTTP debug log will be generated.

Note that your application will be more slow and logs will grow really fast.

1.4.2 Debug your browser status

Most browsers can show you network activity and JavaScript errors using its own interfaces. They can be very helpful if you happen to find a bug on AgendaV. Some examples of browser which include this support are:

- Mozilla Firefox with Firebug extension
- Google Chrome/Chromium with Developer Tools (no addon required)

Translating AgenDAV

AgenDAV translation system is based on labels which get translated into full sentences/paragraphs. They can contain placeholders which get replaced by the system depending on the context.

2.1 How to add a translation

1. Access [AgenDAV project in Transifex](#) and use the *Create language_* to add the missing language. You'll have to wait until I approve the request. Once you have it created, you'll be able to use Transifex interface to translate AgenDAV strings.
2. Search the corresponding [CodeIgniter translation](#) and download its zip file.
3. Uncompress the CodeIgniter translation into `web/application/language`
4. Edit `web/config/languages.php` and add a new entry like this:

```
$config['lang_rels']['fr_FR'] = array(  
    'codeigniter' => 'french',  
);
```

This supposes the language file you downloadad for CodeIgniter resulted in a directory called `french/`

5. You're done! Set `default_language` to your new language name (`fr_FR` in our example)

Development

Please, read this section if you want to contribute to AgenDAV with code. You can even use this information to confirm a bug.

3.1 Virtual Machine

Configuring a working environment for AgenDAV can be difficult. There are so many requirements (web server, PHP, PHP libraries, Composer, a CalDAV server...).

To make the task of setting up an environment easier, AgenDAV ships with a [Vagrant](#) file and an [Ansible](#) playbook. If you don't know what that means, don't worry! They are just two tools used to generate a virtual machine with everything prepared to work on AgenDAV.

Software you will need:

- [VirtualBox](#): virtualization software
- [Vagrant](#): VM manager
- [Ansible](#): automation platform

Once you have them installed, let Vagrant initialize the virtual machine:

```
$ vagrant up
```

A base image has to be downloaded, and lot of packages have to be installed inside the machine, so this will take a while. Go grab a drink until the machine is ready!

The virtual machine can be stopped and started again using Vagrant. The initialization process will only run the first time you do `vagrant up`, and next starts will just require a few seconds.

You can stop the machine with `vagrant halt`, and running `vagrant up` again will bring it back to life.

Your current working directory is shared with the virtual machine, so you can develop on your local machine and your changes will be visible inside the virtual machine.

3.1.1 Accessing the virtual machine

The environment created inside the virtual machine will be accessible using the following URLs and commands:

- AgenDAV: <http://localhost:8080/>
- Baïkal server: <http://localhost:8081/>

- SSH: `vagrant ssh`. Your local machine working directory will be mounted at `/vagrant` inside the virtual machine

The credentials for this environment are:

- Username: `demo`
- Password: `demo`

Note that *Development environment* will be enabled by default.

3.2 Working with scripts and stylesheets

AgenDAV uses some widely known tools to help on development, such as [Grunt](#), [Less](#) and [Bower](#).

3.2.1 Working with grunt

Perhaps you already have them installed on your local machine, but to make things simpler you already have them installed on the virtual machine.

If you are going to work on AgenDAV stylesheets or templates, you could benefit from running the following command inside the virtual machine (i.e. run first `vagrant ssh`):

```
$ cd /vagrant
$ grunt watch
```

If you keep that session open, a Grunt task will look for modified `.less` and `.dust` files and will compile them for you.

3.2.2 How to download or update AgenDAV frontend dependencies

You will have to use `grunt` to download and copy all dependencies to AgenDAV `public/` directory. Run the following command:

```
$ grunt
```

3.2.3 Changing AgenDAV stylesheets

Stylesheets are built using the Less pre-processor. They are written as `.less` files and can be found inside the `web/assets/stylesheets` directory.

If you have the `grunt watch` command running, it will automatically compile any modified `.less` files.

Release notes

4.1 2.0.0-beta1 (2015-08-26)

- AgenDAV needs [Composer](#). If you are using the tarball from the website you don't have to worry at all. Otherwise, you'll have to install it and fetch AgenDAV dependencies. You'll find instructions on [download_dependencies](#).
- AgenDAV now uses a single `settings.php` file. Details on this file are provided on [Configuration](#).
- Deep database changes are required. Instructions are provided on [Database upgrade](#)
- **This beta release cannot handle shared calendars**, and your existing shares will be removed from the database. Please, make sure you are not upgrading a production environment which uses calendar sharing.
- Read the [CHANGELOG](#) for a detailed list of changes.

4.2 1.2.6.1 and 1.2.6.2 (2012-10-15)

This release just fixes a problem with some timezones which have three components, such as America/Argentina/Cordoba. AgenDAV 1.2.6 refused to parse calendars that contained events with those kind of timezones.

4.3 1.2.6 (2012-09-03)

- This release requires neither configuration changes nor DB schema updates.
- Reminders feature has been added. Reminders will be attached to events, but this version of AgenDAV is not capable of showing popups for them yet. Other CalDAV clients, such as smartphones and desktop applications, will understand them.
- A new `log_create_permissions` option has been added to `advanced.php` configuration file. Default is 0640

4.4 1.2.5.1 (2012-06-11)

Note: This version has a bug that will make [Database upgrade](#) to fail if you have AgenDAV configured to use a language other than English. Please, set `default_language` to `en` before running `agendav dbupdate`

- This release fixes some invalid SQL queries included in the database update process

4.5 1.2.5 (2012-06-07)

- DB schema needs to be updated, but a new database upgrade process is now available. No more *.sql files, just read *Database upgrade* section.
- This release requires PHP cURL support. Make sure your PHP installation has the cURL module enabled
- Now you can share a calendar in read-only mode. You're advised to create a new `caldav.php` configuration file using the provided template
- Date and time format options have been moved inside `config.php`. Prior to this release they were set inside lang files, which didn't make much sense. Have a look at new options `format_full_date`, `format_column_month`, `format_column_week`, `format_column_day`, `format_column_table`, `format_title_month`, `format_title_week`, `format_title_day` and `format_title_table`
- Translations are now managed using [Transifex](#). Note that default language has been renamed from `en_US` to `en`, as I think it's a neutral English

4.6 1.2.4 (2012-01-16)

This release requires no database changes. Contains a fix for localization support for IE7, and lots of aesthetical changes, besides upgraded libraries (jQuery 1.7.1, qTip2 and iCalcreator). Session cookies are now smaller and contain only session identifiers.

- You have to create a new `caldav.php` config file using the provided template (`caldav.php.template`) because `caldav_url` has been split into two new configuration parameters. Some CalDAV servers use different URLs for principals and calendar collections/resources, and the previous model didn't work with them:
- `caldav_principal_url` will be used to generate principal URLs
- `caldav_calendar_url` will be used to generate calendar and resource URLs
- A new `enable_calendar_sharing` setting has been added to `caldav.php` to disable calendar sharing for those servers that don't support WebDAV ACLs
- `advanced.php` configuration file has been updated, make sure you don't overwrite your own changes.
- Note that required PHP version is 5.3.0 and not 5.2, there was an error on the requisites list.
- A setup test script has been included to check if your system meets some basic requisites. Follow the [Troubleshooting AgendaV](#) section for more details.

4.7 1.2.3 (2011-11-08)

This release fixes some bugs with am/pm indicators under some circumstances, editing recurring events, `include_path` problems and other bugs.

It also introduces dynamic text color calculation and new default calendar colors, so it's recommended to remove the following settings from `config.php`:

- `default_calendar_color`: now the first color from `calendar_colors` is used
- `additional_calendar_colors`: this option has been replaced by `calendar_colors`

Copy `calendar_colors` definition from `config.php.template` to your `config.php`

This release requires no database upgrades.

4.8 1.2.2 (2011-10-25)

This release fixes important issues with recurrent events. It also fixes the am/pm indicator on times.

It requires no database upgrades.

4.9 1.2.1 (2011-10-24)

This release mainly fixes issues with timezone differences and Daylight Save Time problems.

Also updates iCalcreator library to latest version (2.10.15) and qTip2.

It requires no database upgrades.

4.10 1.2 (2011-10-17)

- DB schema needs to be altered. UTF8 wasn't being used by default, and sessions table wasn't using InnoDB. Apply the changes on `sql/changes/1.1.1_to_1.2.mysql`, which are the following:

```
ALTER DATABASE agendav CHARACTER SET utf8 COLLATE utf8_general_ci;
ALTER TABLE sessions CONVERT TO CHARACTER SET utf8;
ALTER TABLE sessions ENGINE InnoDB;
ALTER TABLE shared CONVERT TO CHARACTER SET utf8;
```

- Main configuration file (`config.php`) has been completely **rewritten** to make it easier to write. Please, use the provided `config.php.template` as the base for a new `config.php`
- Interface translation and timezone configuration is now possible in AgendaV. Please, make sure you set correct values on `config.php`
- AgendaV has lots of corrections and fixes. See the CHANGELOG

4.11 1.1.1 (2011-09-24)

- Fix DB schema. Wasn't properly updated on `sql/schema.sql`, which caused a problem with sessions

To correct this issue without completely recreating your current database, run the following two queries:

```
CREATE INDEX last_activity_idx ON sessions(last_activity);
ALTER TABLE sessions MODIFY user_agent VARCHAR(120);
```

- Remove LDAP dependency. AgendaV now authenticates against CalDAV server.

Before this change, AgendaV authenticated users at first using LDAP, and then your CalDAV server had to authenticate them again. With this change, AgendaV completely relies on your CalDAV server.

C

caldav.authmethod
 configuration value, 8
caldav.baseurl
 configuration value, 8
caldav.baseurl.public
 configuration value, 8
caldav.publicurls
 configuration value, 8
calendar.sharing
 configuration value, 9
configuration value
 caldav.authmethod, 8
 caldav.baseurl, 8
 caldav.baseurl.public, 8
 caldav.publicurls, 8
 calendar.sharing, 9
 csrf.secret, 8
 db.options, 7
 defaults.date_format, 9
 defaults.language, 9
 defaults.time_format, 9
 defaults.timezone, 9
 defaults.weekstart, 9
 encryption.key, 8
 log.path, 8
 logout.redirection, 9
 proxies, 7
 site.footer, 7
 site.logo, 7
 site.title, 7
csrf.secret
 configuration value, 8

D

db.options
 configuration value, 7
defaults.date_format
 configuration value, 9
defaults.language

 configuration value, 9
defaults.time_format
 configuration value, 9
defaults.timezone
 configuration value, 9
defaults.weekstart
 configuration value, 9

E

encryption.key
 configuration value, 8

L

log.path
 configuration value, 8
logout.redirection
 configuration value, 9

P

proxies
 configuration value, 7

S

site.footer
 configuration value, 7
site.logo
 configuration value, 7
site.title
 configuration value, 7