

PassVault

Documentation

Overview:

The PassVault app is built using Zend Framework 3 and follows PHP and Zend best practices.

The code is split up into modules (Zend best practice). So you can modify and/or delete any modules and the application would still be functional.

This makes extending our application a breeze! You can add features without touching any existing code!

If you're using Apache, we've provided a .htaccess file which rewrites requests to the public/index.php script. If you're using NGINX however, you'll have to configure your nginx config to rewrite requests yourself. In the installation instructions below, we've provided sample configuration for both Apache and NGINX.

Please note, that any domains, urls or file and directory paths may be different to your installation or OS, so please refer to your software version and OS version documentation.

Requirements:

- PHP 7.0+
- MySQL OR MariaDB
- NGINX or APACHE

Installation:

Since version 1.5.3, we have included a self-installer.

- 1) Copy the contents of the App folder to your server's public document root.
- 2) Now head to your URL, an installer page will open.

- 3) Enter your database credentials.
- 4) Submit!
- 5) The installer should then redirect to the main application.
- 6) That's it.

Manual Installation:

- 1) Copy the contents of the App folder to your server.
- 2) In the project root (the main folder of the application), open up the .env file. If the file doesn't exist, simply copy the .env.dist file and rename it to .env.
- 3) Edit the .env file and add your database credentials and the app url. The app url is the url where you've installed the application, for example: <http://example.com/> or if you've installed it in a subfolder: <http://example.com/subfolder/public/>
- 4) Import the `app.sql` file into your database (you can do this using phpmyadmin or similar tools).
- 5) That's all that's needed from the application to run. If you don't already have Apache or Nginx configured, there's a tutorial below for that.

Updating:

We regularly release new updates to our software. Following this guide will help you to install them. Please note, that you may need to copy any uploads in your public/upload folder.

- 1) Unzip the download and copy the contents of the App folder to a new folder on your server.
- 2) Make a backup of your existing installation.
- 3) Since you're updating and not installing, open the config/modules.config.php file, look for the module called "Install", comment that out (add // or # before the line). This will prevent the installer from running.
- 4) Re-run the app.sql file. (Some queries will fail as the tables already exist, so just remove those queries).
- 5) Give it a quick test, if it's working, then move this installation to your existing app folder.
- 6) Done!

Setting up Apache:

If you've installed the application in the main Apache www directory, then you don't need to change any configuration. However, if you're using this application from a subdirectory and you want to remove the `/public/` from the URL, you'll need to make the below changes:

You can use something called VirtualHost on apache:

- 1) You'll need to either edit the `/apache2/sites-available/000-default.conf` file or copy it and change the name to your new url. (the location of your apache2 config might differ from the one above).
- 2) Open up the `000-default.conf` file or if you're creating a new file, open that up.
- 3) Add the following configuration to the file (change example.com and www.example.com to your url).

```
<VirtualHost *:80>
    ServerName example.com
    DocumentRoot /var/www/example.com/public
</VirtualHost>
```

- 4) You'll have to enable the VirtualHost files now, to do this you can use the a2ensite tool that Apache provides.

```
sudo a2ensite example.com.conf
```

 ← Replace example.com.conf with your filename.

- 5) Once that's done, you now have to restart Apache:

```
sudo service apache2 restart
```

Setting up NGINX:

Setting up NGINX configuration is completely different to Apache. Below is an example Server block which you can find in `/etc/nginx/sites-available/default`

```
server {
    listen 80 default_server;
```

```

listen [::]:80 default_server;
server_name example.com;
root /usr/share/nginx/html/public;

index index.html index.htm index.php;

location / {
    try_files $uri $uri/ /index.php?$query_string;
}

location ~ /\.php$ {
    try_files $uri =404;
    fastcgi_split_path_info ^(.+\.(php|php5|php7|php8|php9|html))(/.+)$;
    fastcgi_pass unix:/var/run/php7.0-fpm.sock;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    include fastcgi_params;
}
}

```

Setting up Recaptcha

By default, the recaptcha integration is disabled. Follow the below instructions to get it set up.

- 1) Navigate to the admin settings page and click on “Captcha Settings”.
- 2) Tick the “enable recaptcha” checkbox.
- 3) Head to the [ReCaptcha website](#).
- 4) Register a new site and choose “reCaptcha v2” and add your domains.
- 5) Copy the Site key and Secret key and paste into the admin settings page.

Now, whenever you visit the login or register page, you will see the ReCaptcha plugin loaded.

Setting up Google Login

To set up the Google login, follow the steps below:

- 1) Head to console.developers.google.com
- 2) Go to the Credentials page and click “create credentials”.
- 3) Choose “OAuth Client ID”.
- 4) Choose “Web Application” for application type.
- 5) For the “Authorized redirect URIs” enter the domain where you installed the application, with `/user/google/callback` appended. So it will look something like: `http://example.com/user/google/callback`

- 6) Copy both the Client ID and Client Secret and paste them into the Admin Settings "OAuth Settings" section.
- 7) Enable the Google Login feature by clicking the checkbox, and saving.

Setting up Two Factor Integration

We've provided a Two Factor integration with apps like Google Authenticator. Setting it up only requires you enter an email address and App Name in the admin settings.

Setting up Encryption

By default, the encryption key is generated when you import the app.sql file into your database. If you're migrating your application to another host, make sure that you keep the same key, otherwise you won't be able to decrypt existing encrypted data.

Setting up Google Chrome Extension

Inside the app directory, you'll see the chrome-extension folder. This contains the source file required to set up your own Chrome extension!

There's a few changes you need to make, so just follow along.

- 1) Open up the manifest.json file and edit these following lines: `name` - enter your own App name. `author` - enter your name, or company name. `permissions` - change the `http://password.development/` value to your installation url.
- 2) Open up the `source/` folder and edit these following files: `popup.html`, `popup.js`. In these files, whenever you see a reference to a url or the PassVault name, change it to your own values.

Once you've edited the chrome extension, you can now follow the next steps.

- 1) Open up Google Chrome, and go to <https://chrome.google.com/webstore/developer/dashboard>
- 2) Zip the chrome-extension directory ready for uploading.
- 3) Once uploaded, you can now edit the details!

That's it! Install the chrome extension and try it out with your PassVault account.

Updating Dependencies:

We've used Composer (a PHP dependency manager), which pulls in any 3rd party code such as the Zend Framework. By default, we've included the vendor folder which is where Composer installs the 3rd party code.

If you have Composer installed and already use composer, you can delete the vendor folder and use the following commands yourself:

COMMAND	DESCRIPTION
composer update	This will install any dependencies from the composer.json file and will also download any updates.
composer install	This command will install all of the dependencies from the composer.lock file. It wont download any updates.

Directory Notes:

Below are the directories you will see in the application:

DIRECTORY	DESCRIPTION
config/	All of the application configuration files live here.
data/	This is where the application will store the cache files, Doctrine proxies and application logs. <i>Make sure that this folder has the correct permissions.</i>
module/	This is the source code of the application.
public/	This is the directory that can be publicly accessed on your server. The css, js images and uploads are stored here.
vendor/	This is where composer will install any dependencies.

File Notes:

Below are some important files:

FILE	DESCRIPTION
.env	The .env file contains environment specific configuration. This is explained in further detail in the .env configuration section.
doctrine.php	This file is used to run the Doctrine CLI application, which you can use to manage doctrine and the database.
.htaccess	This is required for Apache based installations, if you're using NGINX, you can delete it.

.Env Configuration

The .env file is file specifically used for storing environment-specific configuration. The .env file is required and won't work unless it exists.

NAME	DESCRIPTION
DB_VM_HOST	This is the hostname of your database. If your database is on the same server as the application, you can use <code>localhost</code> or <code>127.0.0.1</code>
DB_PORT	This is the port of your database, usually 3306.
DB_USERNAME	The database user's username.
DB_NAME	The database name where you imported the <code>app.sql</code> file.
APP_URL	This is the URL where you installed the application. It's important to add the trailing slash at the end of the url. It's used for the location of the uploaded file, which is then stored in the database.

Misc Notes:

Rewriting the /public/ from the URL:

We've written this .htaccess file

<https://gist.github.com/nic-d/b8ca22bc268ba1e2eaf0fe5218a0b0ed>

Which you can use to remove the /public/ from the url, to use it, copy and paste this file into the root .htaccess file in the App folder.

Admin User:

When accessing the /user/* route, the application will automatically generate a default admin user if none exist in the database. The default admin details can be found below:

Username: admin@example.com

Password: admin123@

We highly recommend that you change the admin password as soon as you login.

Development Configuration:

By default, the application is in production mode, if you're running the application in development, you can enable development configuration files:

- 1) Navigate to the `config/` folder
- 2) Copy the `development.config.php.dist` file and rename it to `development.config.php`
- 3) Navigate to the `config/autoload` folder
- 4) Copy the `development.local.php.dist` file and rename it to `development.local.php`

This will run the application in development mode.

FAQ:

I changed the configuration, but it's not updating in the application?

If you aren't in development mode, you have to clear the cache files from `data/cache/`. There's also an option in the admin homepage to clear the cache.

I changed .env configuration, but it's not updated?

This is also due to the cache, if you're not running the application in development mode, simply clear the `data/cache` folder and then on the next application request you'll see that the `.env` is updated and working.

How can I recompile the sass?

You'll see two folders within the public folder called "src" and "build".

Build: contains the *compiled* sass files.

Src: contains the raw uncompiled sass files.

You can make your changes in src, and then compile it. Make sure that you compile them into the build directory.

I've installed the application in a subdirectory, and it doesn't work?

This is a common issue with installing PHP applications into a subdirectory of the www root. We've written an entire section above called "Setting up Apache".

If you can't or don't want to change the Apache configuration, you can access the application through the `/subdirectory/public/` folder.

For example, your application is installed at example.com, but it's inside the "test" subdirectory, you can access it by visiting: `example.com/test/public`

Thank you!

Thanks for purchasing this application, if you have any questions or have a bug report, please don't hesitate to contact us through our profile:

<https://codecanyon.net/user/medusasoftware/portfolio>. If you like the application, a review would help us massively!