

Installation

Server Requirements

For running app make sure you have:

- PHP >= 8.0.2 version (8.1+ recommended)
- MySQL 5.6+
- Nginx or Apache (Nginx recommended)

These PHP Extensions are require:

- Intl
- GD
- BCMath
- PDO
- SQLite3
- Ctype
- Fileinfo
- JSON
- Mbstring
- OpenSSL
- Tokenizer
- XML
- Exif

Installation

- [How to install VPS with Debian 10](#)
- [How to Set Up AWS S3](#)
- [How to Set Up Digital Ocean Spaces](#)

1. Upload files on your server

Upload project files to the web root folder of your domain. It's mostly located in `html`, `www` or `public_html` folder name.

2. Configure your Document Root

Configure your domain document root to the point of the files you previously uploaded directly into `/public` folder. So, if you uploaded files into `/public_html` folder, your document root must be set as `/public_html/public`. It should look like this

Don't forget to enable Force HTTPS Redirect.

3. Set write permissions

Set `755` permission (CHMOD) to these files and folders directory within all children subdirectories:

- `/bootstrap`
- `/storage`
- `/.env`

4. Open your application in your web browser

Then open your application in web browser. If everything works fine, you will be redirected to the setup wizard installation process.

5. Server Check

On the first page you will see server check. Make sure all items are green. If not, then correct your server setup by recommended values and refresh your setup wizard page.

6. Follow setup wizard steps

That was the hardest part of installation process. Please follow instructions in every step of Setup Wizard to successfully install VueFileManager.

7. Set up Cron

If you are running VueFileManager on shared web hosting (CPanel, Plesk etc.)

1. Create new cron job

2. Set execution cycle every minute
3. Search the absolute directory path where you uploaded VueFileManager files (like `/www/project_files`). The path must start with `/`.
4. Copy the command below, paste it to the command text area and replace in command string `replace_by_your_path` exactly with your path you found in step 3.
5. It should look like this with your pasted project path.

```
php replace_by_your_path/artisan schedule:run >> /dev/null 2>&1
```

1. If you have multiple php versions installed on your server, you should specify php path to the latest php version (8+). So, you should edit `php` in command above and replace it by path. For Example:

```
/usr/bin/php8.1/php replace_by_your_path/artisan schedule:run >> /dev/null 2>&1
```

If you are running VueFileManager on linux server

1. Search the absolute directory path where you uploaded VueFileManager files (like `/www/project_files`). The path must start with `/`.
2. Copy the command below, paste it to your cron list and replace in command string `/www/project_files` exactly with your path you found in step 1.

```
* * * * * cd /www/project_files && php artisan schedule:run >> /dev/null 2>&1
```

8. CORS Configuration (When you Set External S3 Storage)

In your s3 bucket settings you should have option to set up your CORS (Cross-Origin Resource Sharing). It's basically adding your app url to the list of allowed CORS. This step is required for reading pdf documents from s3 in your VueFileManager app without loading issues.

Updating Application

1. Replace all files where the app is located except `/storage` folder and `.env` file.
2. Clear the application cache (Admin / Settings / Application).

3. In 5 minutes the app update stuff automatically on the background if needed.

Nginx Configuration

If you running VueFileManager under Nginx, don't forget set this value in your `nginx.conf` file:

```
http {
    client_max_body_size 1024M;
}
```

And example Nginx config for your domain:

```
server {
    listen 80;
    listen [::]:80;

    # Log files for Debugging
    access_log /var/log/nginx/laravel-access.log;
    error_log /var/log/nginx/laravel-error.log;

    # Webroot Directory for Laravel project
    root /var/www/vuefilemanager/public;
    index index.php index.html index.htm;

    # Your Domain Name
    server_name example.com;
    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }

    # PHP-FPM Configuration Nginx
    location ~ \.php$ {
        try_files $uri =404;
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_pass unix:/run/php/php7.3-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

Apache Configuration

Make sure you have enabled `mod_rewrite`. There is an example config for running VueFileManager under apache:

```

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

    <Directory "/var/www/vuefilemanager/public">
        AllowOverride All
        allow from all
        Options +Indexes
        Require all granted
    </Directory>

    RewriteEngine on
    RewriteCond %{SERVER_NAME} =example.com
    RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,NE,R=permanent]
</VirtualHost>

```

Migrating to Another Domain

If you move your VueFileManager application into another domain or subdomain, you have to manually change values in your `.env` file.

1. Open your `/.env` file.
2. Find `APP_URL` variable and write your new domain location. Don't forget start with defining `https://` protocol.
3. Find `SANCTUM_STATEFUL_DOMAINS` variable and write your new domain location without http protocol.
4. Remove your cached config file which is located `/bootstrap/cache/config.php`.

Broadcasting

About Broadcasting

Broadcasting is responsible for real time app experience. If broadcasting is set, you will be able to get just in time updates in your app.

For example, remote upload works on the background, while the files are downloading and showing in your view immediately after they are ready without refreshing the app. Or, you will get immediately notifications about new events like team invitations, file requests and many more.

More real time features will be implemented in incoming updates.

VueFileManager support 2 types of connections:

1. Pusher - Suitable for all users and all hosting platforms
2. VueFileManager Broadcast Server - Free of charge, suitable for experienced users, limited for VPS servers

Install Broadcast Server

We strongly recommend only for experienced users to set up and running VueFileManager broadcast server. For others, we recommend to use Pusher service to easy set up and host broadcasting service.

Server Requirements

For running app make sure you have:

- VPS server with SSH access
- PHP >= 8.0.2 version (8.1+ recommended)
- Nginx
- Supervisor
- Certbot

Installation

We assume you have installed and running properly your VPS server.

Websocket Server Set Up

Upload VueFileManager into your server, for example path directory `/var/www/sockets` would be great. Next connect to your server with ssh and change your terminal directory into VueFileManager:

```
cd /var/www/sockets
```

Run installation command for websocket server. You will be prompted to type host of which you want to allow incoming requests.

```
php artisan websockets:install
```

Domain & Nginx Set Up

Create subdomain for your socket host, for example `socket.vuefilemanager.com` and direct this subdomain to your vps where socket server will be running.

Then, create nginx config for your subdomain:

```
nano /etc/nginx/sites-available/socket.conf
```

And paste the template below, just replace subdomain with yours:

```
server {
    listen      80;
    server_name socket.vuefilemanager.com;

    location / {
        proxy_pass      http://127.0.0.1:6001;
        proxy_read_timeout    60;
        proxy_connect_timeout 60;
        proxy_redirect      off;

        # Allow the use of websockets
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Enable your created config by this command:

```
ln -s /etc/nginx/sites-available/socket.conf /etc/nginx/sites-enabled/
```

Restart your nginx:

```
systemctl restart nginx
```

In the last step, install ssl certificate for your previously created subdomain with certbot:

```
certbot --nginx
```

Supervisor Configuration

We need supervisor to manage running your websocket server on the background. In `/etc/supervisor/conf.d` (Debian/Ubuntu) or `/etc/supervisord.d/` (Red Hat/CentOS) create a file `websockets.conf`.

Just edit php path, project path and user when they are different for your vps:

```
[program:websockets]
command=/usr/bin/php /var/www/socket/artisan websockets:serve
numprocs=1
autostart=true
autorestart=true
user=www-data
```

Run command below to start your websocket server:

```
supervisorctl start websockets
```

Run command below to stop your websocket server:

```
supervisorctl stop websockets
```

When you update code or server for some reason, you must also update running supervisor:

```
supervisorctl reload
```

VueFileManager Set Up

Log in to your VueFileManager admin account and go to `Admin / Settings / Environment`.

Find Broadcasting form and select `VueFileManager` as broadcasting driver. Set your hostname and save the form.

That's all, you are running your own Broadcast server.

Subscription Configuration

Configuring Production/Testing Environment

To set up your subscription, please follow these steps below.

1. If you didn't set up your subscription type in Setup Wizard, go to the `Admin / Settings / Application` and find subscription widget. Next set value as `Fixed`.
2. Go to the `Admin / Billings` and fill the inputs with your billing information.
3. Go to the `Admin / Payments` and turn on the switch `Allow Subscription Payments`.
4. Set up credentials for all payment gateway you want. If you set production mode, make sure you fill your credentials with production keys, and vice versa. If needed, don't forget to turn on `live mode` for PayPal.
5. Set up your webhooks, you can find your webhook url in payment gateway widget.
6. Go to the `Admin / Plans` and create your first plan. Make sure all payment gateways support the currency you want, especially for Paystack, it supports only `GHS, NGN, USD and ZAR`.

Upgrading From Testing Environment to the Production Mode

1. Go to the `Admin / Payments` and set up credentials for all payment gateway you want with production keys type. Don't forget to turn on `live mode` for PayPal.
2. Go to the `Admin / Plans` and delete all your previously created plans. They will be archived.
3. Create new production plans you want offer.

Developers

Running Environment On Your Localhost

For running development environment make sure you have:

- Node >= 14
- NPM >= 6

Express Installation

If you would like to have express installation without Setup Wizard process, please update your database credentials in `.env` file

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=
```

Next, run this command below. Admin account will be created with credentials `howdy@hi5ve.digital` and password `vuefilemanager`.

```
php artisan setup:prod
```

Express Installation with Demo Data

If you would like to generate demo content, run this command below. Admin account will be created with credentials `howdy@hi5ve.digital` and password `vuefilemanager`.

```
php artisan setup:dev
```

Generate Demo Data for Metered Subscription

If you would like to generate demo content for the subscription with the metered billing, run this command below.

```
php artisan subscription:demo metered
```

Generate Demo Data for Fixed Subscription

If you would like to generate demo content for the subscription with the fixed billing, please fill the credentials in your `.env` file for the payment systems you'd like to use:

```
STRIPE_SECRET_KEY=
STRIPE_PUBLIC_KEY=
STRIPE_WEBHOOK_SECRET=

PAYSTACK_SECRET=
PAYSTACK_PUBLIC_KEY=

PAYPAL_CLIENT_ID=
PAYPAL_CLIENT_SECRET=
```

```
PAYPAL_WEBHOOK_ID=  
PAYPAL_IS_LIVE=false
```

and then run this command to generate demo content for fixed subscription:

```
php artisan subscription:demo fixed
```

Sanctum Stateful Domains

After installation, please make sure your current host/domain where you are running app is included in your `.env` file in `SANCTUM_STATEFUL_DOMAINS` variable.

Running your Local Server

To start server on your localhost, run command below.

```
php artisan serve
```

For developing Vue front-end, you have to install npm modules by this command:

```
npm install
```

To compiles and hot-reloads for front-end development. Then run this command:

```
npm run hot
```

Building Your App for Production

To compiles for production build, run this command

```
npm run prod
```

Support

The following support channels are available at your fingertips:

- [CodeCanyon support message](#)

Supporting VueFileManager

We are trying to make the best for VueFileManager. There are a lot of things to do, and a lot of features we can make. But, it can't be done without you, development is more and more complicated and we have to hire new colleagues to help us. There is couple way you can support us, and then, we support you with all great new features we can make. Thank you for participating on this awesome application!

- [Buy me a Coffee](#)
- [Become a Patreon](#)
- [One-time donation via PayPal](#)

Security Vulnerabilities

If you discover a security vulnerability within this project, please send an e-mail to peterpapp@makingcg.com. All security vulnerabilities will be promptly addressed.