Server Configuration - Mixpost

If you installed Mixpost using the "As a standalone app" or "As a package in an existing Laravel app" method, it is necessary to install and configure other software on the server. These configurations are very important, please do not skip this step.

Requirements

Softwares

- PHP 8.1 or higher
- Database (eg: MySQL, PostgreSQL, SQLite)
- Redis 6.2 or higher
- Web Server (eg: Apache, Nginx, IIS)
- URL Rewrite (eg: mod_rewrite for Apache)
- Supervisor
- FFmpeg
- Curl
- Zip
- Unzip
- Cron

PHP extensions

- php-curl
- php-mysql
- php-bcmath
- php-gd
- php-mbstring
- php-redis
- php-xml
- php-zip
- php-intl

These extensions are version-specific for PHP, so if you have PHP 8.2.x installed you would run:

```
sudo apt install php8.2-curl php8.2-mysql php8.2-bcmath php8.2-gd php8.2-mbstring php8.2-redis
php8.2-xml php8.2-zip php8.2-intl
```

Default public web root

If you installed Mixpost using the Standalone method you may instruct your web server(Nginx/Apache/Anything else) the default public web root.

You should set the default public web root to the "/public" folder of the Mixpost Standalone project.

Nginx

In Ubuntu/Debian, you can find the:

- The main configuration file in "**/etc/nginx/nginx.conf**"
- Default server block: "**/etc/nginx/sites-available/default**"
- Other server blocks (virtual hosts): Additional files in "**/etc/nginx/sites-available/**" and they are symlinked to "**/etc/nginx/sites-enabled/**" when activated.

```
server {
    listen 80;
    root /var/www/your-mixpost-project/public;
    index index.php index.html;

    // your nginx configs
}
```

Remember, after making changes to any Nginx configuration file, you should restart the Nginx service to apply the changes.

```
sudo systemctl restart nginx
```

Apache

In Ubuntu, you can find the default virtual host: "**/etc/apache2/sites-available/000-default.conf**".

```
<VirtualHost *:80>
    ServerAdmin webmaster@yourdomain.com
```

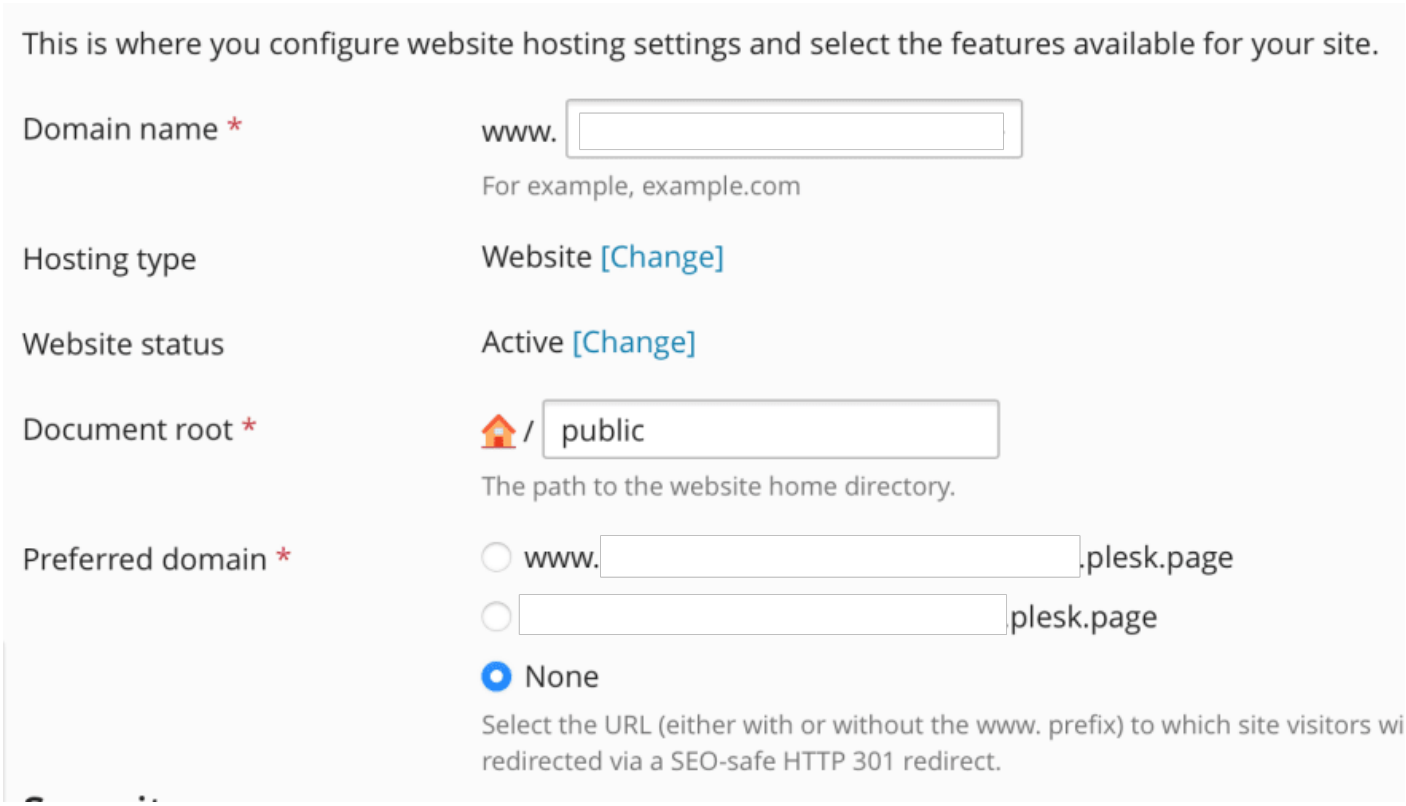
```
    DocumentRoot "/var/www/your-mixpost-project/public"
    ServerName yourdomain.com
    ServerAlias www.yourdomain.com

    <Directory "/var/www/your-mixpost-project/public">
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    // your apache configs
</VirtualHost>
```

Remember, after making changes to any Apache configuration file, you should restart (or reload) the Apache service to apply the changes.

```
sudo systemctl restart apache2
sudo systemctl reload apache2
```

Plesk

Open your Plesk panel, then go to “**Hosting settings**”. Change the “**Document root**” value to match the “public” folder.


This is where you configure website hosting settings and select the features available for your site.
Domain name *
www.
For example, example.com
Hosting type
Website [Change]
Website status
Active [Change]
Document root *
/ public
The path to the website home directory.
Preferred domain *
www. .plesk.page
plesk.page
None
Select the URL (either with or without the www. prefix) to which site visitors wi
redirected via a SEO-safe HTTP 301 redirect.

Installing FFmpeg

Mixpost has the ability to generate images from video while uploading a video file. This would not be possible without FFmpeg installed on your server.

You need to follow FFmpeg installation instructions on their official website. Usually, you can install it with:

```
sudo apt-get install ffmpeg
```

> After installation, depending on the operating system, set your paths to the "ffmpeg" and "ffprobe" binary files (not the folder they're in!). Default folder path: `/usr/bin/`. If FFmpeg is there, there is no need to change it.

If it is somewhere else, navigate to your Mixpost application and put this in your `.env` file

```
FFMPEG_PATH=/usr/bin/ffmpeg
FFPROBE_PATH=/usr/bin/ffprobe
```

Common Plesk problem:

If you get **open_basedir** errors, you can move "ffmpeg" and "ffprobe" to a folder inside your "httpdocs".

```
sudo cp /usr/bin/ffmpeg /var/www/vhosts/domain/httpdocs/ffmpeg
sudo cp /usr/bin/ffprobe /var/www/vhosts/domain/httpdocs/ffprobe
```

And then change your paths in ".env" to:

```
FFMPEG_PATH=/var/www/vhosts/domain/httpdocs/ffmpeg
FFPROBE_PATH=/var/www/vhosts/domain/httpdocs/ffprobe
```

Installing Redis

So that the posts can be scheduled, Mixpost puts them in the queue.

To be able to do this, you need to install Redis. Then, you will need to modify the values of the REDIS_* entries in the `.env` file to make sure they are aligned with your redis instance.

Installing & Configuring Supervisor

Installing Supervisor

You need to configure a process monitor. To install Supervisor on Ubuntu, you may use the following command:

```
sudo apt-get install supervisor
```

Configuring Supervisor

Supervisor configuration files are typically stored in the `/etc/supervisor/conf.d`.

Create the file `mixpost-horizon.conf` inside of `conf.d` folder and put this configuration content:

```
[program:mixpost_horizon]
process_name=%(program_name)s
command=php /path-to-your-project/artisan horizon
autostart=true
autorestart=true
user=your_user_name
stopwaitsecs=3600
```

Once the configuration file has been created, you may update the Supervisor configuration and start the processes using the following commands:

```
sudo supervisorctl reread

sudo supervisorctl update

sudo supervisorctl start mixpost_horizon:*
```

Cron

Add a cron that runs the scheduler every minute:

```
* * * * * cd /path-to-your-project && php artisan schedule:run >> /dev/null 2>&1
```

Plesk

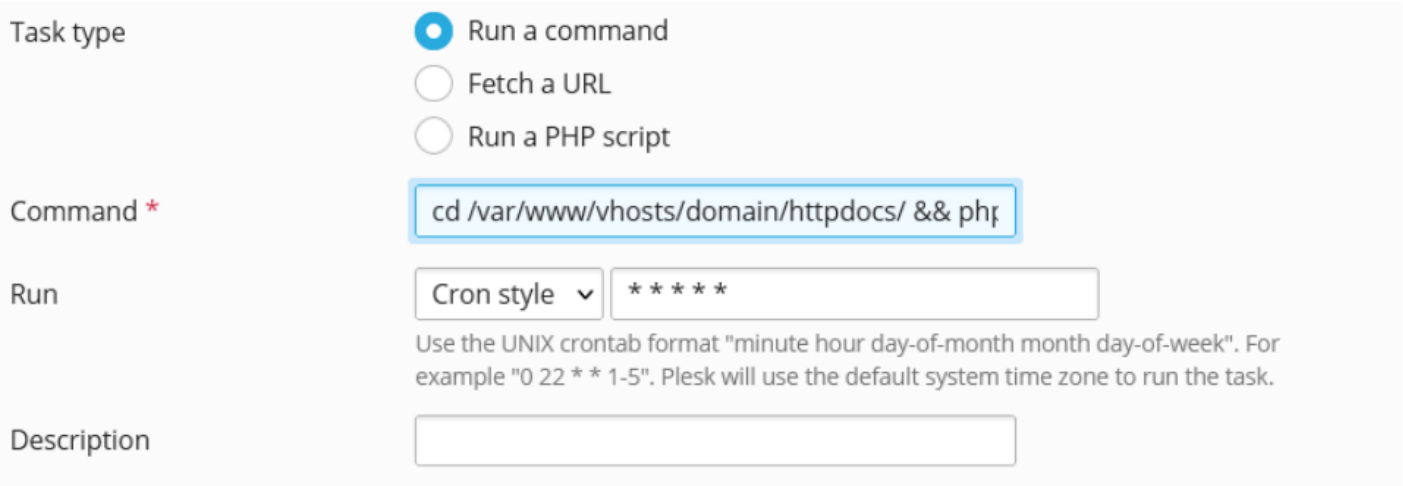
In Plesk, the command will look something like:

```
cd /var/www/vhosts/domain/httpdocs/ && php artisan schedule:run >> /dev/null 2>&1
```

*Replace "**domain**" with your real domain.*

Make sure to set the "Run" to "Cron Style" and insert:

```
* * * * *
```



Other things to consider

Some files that are uploaded, video for example, can be up to 200 mb, by default most web servers have configured a much smaller limit. You will need to check this.

In `php.ini`:

```
post_max_size = 220M
upload_max_filesize = 200M
```

On Ubuntu, you can edit "php.ini":

- Apache: `/etc/php/VERSION/apache2/php.ini`
- FPM (FastCGI Process Manager): `/etc/php/VERSION/fpm/php.ini`

In the paths above, replace `VERSION` with the specific version of PHP installed on your system (e.g., `8.1`, `8.2`).

Then, restart your PHP process. For php fpm, you can restart with:

```
sudo systemctl restart php8.1-fpm.service
sudo systemctl reload php8.1-fpm.service
```

For Apache, you can restart with:

```
sudo systemctl restart apache2
```

In `nginx.conf`:

```
http {
    client_max_body_size 200M;
}
```

Then, `sudo systemctl restart nginx`

For Apache, `/etc/httpd/conf/httpd.conf.`

```
LimitRequestBody 209715200
```

Then: `sudo systemctl restart httpd`