

As a standalone app

If you don't know Laravel, but have basic PHP knowledge and know how to deploy to a server, you can follow these instructions.

You can create a new Laravel application with Mixpost preinstalled using Composer.

Getting a license

In order to install Mixpost, you'll need to [get a license](#) first.

Creating the application

You can create the application with Mixpost pre-installed using this command

```
composer create-project inovector/mixpost-pro-team-app
```

During the execution of this command, Composer will ask for a user and a password. The user is the email address of your [mixpost.app](#) account. The password is the key to your [Mixpost license](#).

Configure the app URL

You will need to modify the value of the APP_URL in the `.env` file to your project URL.

For example: `APP_URL=https://your-domain.com`

Configure the database

You will need to modify the values of the DB_* entries in the `.env` file to make sure they are aligned with your database.

Then, run the migration to create all tables.

Configure the SMTP

By configuring SMTP, Mixpost will be able to send emails such as (password reset link). You will need to modify your .env file.

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.mailgun.org
MAIL_PORT=587
MAIL_USERNAME=
MAIL_PASSWORD=
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS="hello@example.com"
MAIL_FROM_NAME="${APP_NAME}"
```

File Permissions

Make sure you have the right file permissions by following this [tutorial](#).

Server configuration (Manual)

Server configuration Guide

Visit the UI

After performing all these steps, you should be able to visit the Mixpost UI at /mixpost.

Deploying to production

Change the value of APP_ENV from the .env file to **production**.

Debug Mode

In your production environment, this value should always be false. If the APP_DEBUG variable is set to true in production, you risk exposing sensitive configuration values to your application's end users.

Caching Configuration

When deploying to production, you should make sure that you run the `config:cache` command during your deployment process:

```
php artisan config:cache
```

Once the configuration has been cached, everything you change in the .env file will have no effect. To have an effect, you must repeat the execution of the cache command.

Caching Routes

To improve Mixpost application performance, run:

```
php artisan route:cache
```

Once the route has been cached, you have to repeat it every time you deploy/update/upgrade the Mixpost application.

Deploying Horizon

During your Mixpost deployment process (update/upgrade of Mixpost or changes some code), you should instruct the Horizon process to terminate so that it will be restarted by your process monitor and receive your code changes:

```
php artisan horizon:terminate
```

