

Руководство по установке программного обеспечения

1. Общие положения

Настоящее руководство определяет порядок установки и настройки программного обеспечения (ПО), основанного на платформе Laravel, на сервер или в контейнер Docker.

2. Подготовительные мероприятия

Перед установкой требуется клонировать репозиторий проекта, за исключением случаев использования готовых образов Docker из регистра.

2.1. Системные требования

Сервер	CPU	RAM	Диск	Сеть
Веб-сервер (Nginx + PHP + PostgreSQL)	8 vCPU	16 GB	200 GB NVMe	1 Gbit/s

3. Установка напрямую на сервер

3.1. Конфигурация подключения к базе данных

Перед выполнением команд установки убедитесь в корректности данных подключения к БД, указанных в файле окружения `.env`. Пример заполнения файла `.env`:

```
APP_KEY=
DB_CONNECTION=pgsql
DB_HOST=127.0.0.1
DB_PORT=5432
DB_DATABASE=laravel
DB_USERNAME=laravel
DB_PASSWORD=password
```

3.2. Выполните последовательность команд

```
composer install
php artisan migrate
php artisan make:filament-user
php artisan passport:keys
php artisan shield:install
php artisan storage:link
php artisan filament:assets
```

3.3. Настройки времени и локализации

В файле `.env` измените строки:

```
APP_TIMEZONE=Europe/Moscow
APP_LOCALE=ru
```

3.4. Генерация ключа шифрования

В случае ошибки шифрования приложения выполните:

```
php artisan key:generate
```

4. Установка дополнительного ПО

4.1. PHP

Для установки PHP выполните следующие команды:

- Ubuntu/Debian:

```
sudo apt install php php-cli php-common php-fpm php-pgsql php-redis php-zip php-gd  
php-mbstring php-curl php-xml php-bcmath
```

4.2. PostgreSQL

- Ubuntu/Debian:

```
sudo apt install postgresql postgresql-contrib  
sudo -u postgres createuser --interactive  
sudo -u postgres createdb laravel
```

4.3. Redis

- Ubuntu/Debian:

```
sudo apt install redis-server
```

4.4. Nginx

- Ubuntu/Debian:

```
sudo apt install nginx
```

Настройте серверный блок Nginx в директории `/etc/nginx/sites-available/` и создайте ссылку на него в `/etc/nginx/sites-enabled/`.

4.5. FFMPEG

Для генерации видео-превью:

- Windows: Скачать и распаковать сборку, указать пути в `.env`.
- Linux:

```
sudo apt install ffmpeg
```

Указать пути в `.env`:

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

Запуск очереди генерации превью:

```
php artisan queue:work
```

Фоновый запуск на Linux:

```
nohup php artisan queue:work --daemon &
```

4.6. APK

- Windows: Установить Android SDK, указать путь в `.env`.
- Linux:

```
sudo apt install aapt
```

Указать путь в `.env`:

```
AAPT_PATH=/usr/bin/aapt
```

4.7. Планировщик задач (Cron)

Добавьте в crontab:

```
* * * * * cd /путь-к-проекту && php artisan schedule:run >> /dev/null 2>&1
```

Или запустите вручную:

```
php artisan schedule:work
```

5. Установка с помощью Docker

5.1. Подготовка файла окружения

Создайте или скопируйте файл `.env` с переменными:

```
HOST_IP=127.0.0.1
HOST_IP_EXTERNAL=127.0.0.1
NGINX_SERVER_NAME=localhost
WWWGROUP=<id -g>
WWWUSER=<id -u>
XDEBUG_MODE=off
XDEBUG_CONFIG="client_port=9003"
```

5.2. Запуск Docker-контейнеров

```
docker compose up --build
docker compose up -d
docker compose exec --user laravel php-laravel bash
composer install --no-interaction --optimize-autoloader
php artisan migrate
php artisan shield:install --fresh
php artisan passport:keys
php artisan storage:link
php artisan key:generate
php artisan filament:assets
```

Выйдите из контейнера:

```
Ctrl+D
```

6. Первичное наполнение БД (Seeders)

Заполните БД тестовыми данными:

```
php artisan db:seed --class=ContentSeeder  
php artisan db:seed --class=PanelSeeder
```

7. Работа с хранилищем S3

Добавьте в `.env` переменные доступа для подключения S3.

8. Подключение сервиса DaData

Добавьте ключи доступа в `.env` и `.env.testing`:

```
DADATA_KEY=...  
DADATA_SECRET_KEY=...
```

9. Запуск тестов

Запустите тесты:

```
php artisan test
```

Настоящее руководство обязательно к исполнению при установке и развертывании ПО.