

Проект PlantManager.

Краткое руководство по запуску проекта.

Backend часть:

Требуются:

1. БД **MySQL** – либо отдельно устанавливается сама БД (<https://www.mysql.com/>) либо используется какой-то из тулkitов, например Open Server Panel для Windows (<https://ospanel.io/>), последний содержит много других полезных модулей для веб разработки.
2. Собственно **PHP** (<https://www.php.net/>), я использую версию 7.4.20. Он, так же, входит в состав Open Server Panel. Не забываем добавить его в окружение PATH.
3. **Composer** (<https://getcomposer.org/>)
4. Редактор кода – любой на ваш выбор, я использую PhpStorm (<https://www.jetbrains.com/ru-ru/phpstorm/>). Как альтернатива Visual Studio Code (<https://code.visualstudio.com/>) или Atom (<https://atom.io/>). Есть и другие варианты.
5. Для Windows лучше еще установить Windows Terminal (<https://www.microsoft.com/ru-ru/p/windows-terminal/9n0dx20hk701?activetab=pivot:overviewtab>).

Порядок действий:

1. Загружаем проект PlantManager (через git или архивом)
2. Переходим в папку **./PlantManager/backend**
3. Запускаем в этой папке из терминала команду: `composer install`
4. Копируем файл `.env.example` в `.env`
5. Редактируем этот файл (приведены примеры настройки, которую использую я):

APP_URL=http://localhost:8000

DB_CONNECTION=mysql

DB_HOST=localhost

DB_PORT=3306

DB_DATABASE=plantmanager

DB_USERNAME=root

DB_PASSWORD=root

SANCTUM_STATEFUL_DOMAINS=localhost:8080,localhost:8000

SESSION_DOMAIN=localhost

SPA_URL=http://localhost:8080

Важно правильно указать название используемой БД (в моем случае plantmanager), причем, база данных уже должна быть создана и активна, логин и пароль пользователя, который будет иметь доступ к базе данных, а так же в последнем разделе для работы

frontend часть приложения указать домен и порты, откуда будем направлять запросы к backend части.

6. Создать базу данных можно через графический интерфейс, например PhpMyAdmin, HediSQL или MySQLQ Workbench или создать в командной строке:

```
>mysql -u root -p
# Вводите пароль пользователя root
>create database `plantmanager` character set utf8 collate utf8_general_ci;
>exit
```

7. Задаем ключ приложения в .env переменной APP_KEY запустив: **php artisan key:generate**
8. Выполняем миграции для создания структуры базы данных: **php artisan migrate**
9. Запускаем локальный веб сервер php с проектом: **php artisan serve**

Поскольку в проекте PlantManager frontend и backend части разделены – то дополнительные настройки связанные с node.js и npm здесь не требуются. В других проектах необходимо выполнить настройку аналогично frontend части ниже.

Frontend часть:

Требуется:

1. Node.js (<https://nodejs.org/ru/>) – я использую версию 14.16.0
2. Установить менеджер пакетов **npm** (<https://www.npmjs.com>) или/и **yarn** (<https://yarnpkg.com/>)
3. Переходим в терминале в каталог ./PlantManager/frontend
4. Выполняем следующую команду для установки требуемых пакетов: **npm install**
5. В папке ./PlantManager/frontend создаем файл .env.local и заносим в него переменную VUE_APP_API_URL с адресом backend API, в нашем случае:
VUE_APP_API_URL=http://localhost
6. В проекте используются функции подтверждения адреса электронной почты и восстановления пароля с использованием e-mail. Для проверки работы этих функций рекомендуется использовать пакет MailHog (<https://github.com/mailhog/MailHog/releases>). Для платформы Windows его можно запускать из следующего .bat файла:

```
MailHog_windows_amd64.exe -smtp-bind-addr "localhost:1025" -api-bind-addr  
"localhost:8025" -ui-bind-addr "localhost:8025" -ui-web-path "mailhog"
```

В этом случае пакет будет использовать порт 1025 для SMTP запросов (данный параметр настраивается в .env файле Laravel, переменная MAIL_PORT=1025, а по адресу <http://localhost:8025/mailhog> – будет доступен web интерфейс этого пакета, в котором можно контролировать все запросы по почте.

7. Далее запускаем frontend часть в режиме горячей перезагрузки, используя команду: **npm run serve**

Дополнительная информация:

Для **backend** части проекта **PlantManager** установлен пакет **artisan-gui**, который позволяем получить доступ к графическому интерфейсу команд **php artisan**, перейдя по адресу: <http://localhost:8000/~artisan>

Естественно, laravel проект должен быть запущен. Данный пакет можно установить в своем Laravel проекте, используя команду: **composer require infureal/artisan-gui:2.0.0@beta**

При настройке программы [HediSQL](#) или любой для работы с **MySQL**, которая использует jdbc подключение может возникать ошибка, связанная с не правильно установленной временной зоной. Для решения этой проблемы нужно установить (для Москвы):

serverTimezone=Europe/Moscow

Пример:

jdbc:mysql://localhost:3306/springTest?serverTimezone=Europe/Moscow

Иногда при создании нового проекта Laravel может возникать ошибка: "'mix' не распознается как внутренняя или внешняя команда". Для решения данной проблемы, необходимо выполнить команду: **npm install laravel-mix@latest --save-dev**