# Инструкция по установке

**1) Что нужно установить заранее**

* **Docker Desktop** (Win/Mac) или Docker Engine + Docker Compose v2 (Linux).
* **Git**.
* **Node.js 18+** и **npm 9+** (для фронта).
* (Windows) включённый **WSL2** и виртуализация.

**2) Клонирование репозиториев**

# в любой папке

git clone <URL\_БЭКЕНДА> backend

git clone <URL\_ФРОНТЕНДА> frontend

**3) Backend (Laravel + Sail)**

**3.1 Установка зависимостей**

Если Composer установлен локально:

cd backend

composer install

Если Composer **не установлен**, используем “докеризированный” Composer от Sail (Linux/macOS):

cd backend

docker run --rm -u "$(id -u):$(id -g)" \

-v $(pwd):/var/www/html -w /var/www/html \

laravelsail/php83-composer:latest \

composer install --ignore-platform-reqs

*(В Windows PowerShell – убери $(id -u):$(id -g) и смонтируй текущую папку через ${PWD}.)*

**3.2 .env**

cp .env.example .env

Открой .env и проверь ключевые параметры под Sail (стандартные):

APP\_NAME="Transport Monitor"

APP\_URL=http://localhost

# БД по умолчанию от Sail

DB\_CONNECTION=mysql

DB\_HOST=mysql

DB\_PORT=3306

DB\_DATABASE=sail

DB\_USERNAME=sail

DB\_PASSWORD=password

# Очереди (если используешь database)

QUEUE\_CONNECTION=database

# или redis, если он включён в docker-compose

# Для SPA/Vite и куки (если используется Sanctum/CORS)

FRONTEND\_URL=http://localhost:8080

SANCTUM\_STATEFUL\_DOMAINS=127.0.0.1:8080,localhost:8080

SESSION\_DOMAIN=localhost

**3.3 Старт контейнеров**

./vendor/bin/sail up -d # фоновый режим

Первый запуск может занять 5–10 минут (скачивание образов).

**3.4 Инициализация приложения**

./vendor/bin/sail artisan key:generate

./vendor/bin/sail artisan migrate

./vendor/bin/sail artisan storage:link

Если сидеров нет, используй просто migrate.  
*(Если MySQL ещё не готов и миграция упала — повтори через пару секунд.)*

**3.5 Запуск очередей**

В отдельном терминале:

cd backend

./vendor/bin/sail artisan queue:work --timeout=300

**4) Frontend (Vite)**

**4.1 Установка зависимостей**

cd ../frontend

npm ci # или npm install

**4.2 Настройки env фронта**

Создай ./frontend/.env (или .env.local) и пропиши базовый URL API:

VITE\_APP\_BASE='http://localhost:8080'

VITE\_APP\_BASE\_API='http://localhost/api'

VITE\_API\_URL='http://localhost'

*(Если API отдается Nginx-ом Sail на 80 порту, то http://localhost корректно.)*

**4.3 Запуск dev-сервера**

npm run dev -- --host 127.0.0.1

Открой в браузере: [**http://localhost:8080**](http://localhost:8080)  
(Если порт 8080 занят — добавь --port 8081.)

**5) Как это проверять**

1. **Backend:** <http://localhost>
2. **Frontend:** <http://localhost:8080>  
     
   Если есть готовые креды — укажи их эксперту (например: admin@example.com / password).

**6) Частые проблемы и решения**

* **Порт 80 занят** (Apache/IIS/Nginx уже работают): останови локальный веб-сервер или измени порт в docker-compose.yml (с 80:80 на, например, 8090:80) и тогда APP\_URL=http://localhost:8090.
* **CORS/куки не работают:** проверь FRONTEND\_URL, SANCTUM\_STATEFUL\_DOMAINS, SESSION\_DOMAIN и что фронт реально открывается по 127.0.0.1:8080.
* **sail: command not found в Windows:** запускай bash vendor/bin/sail из Git Bash или WSL.
* **Миграции падают из-за БД:** дождись, пока MySQL поднимется (docker ps), и повтори migrate.
* **Vite порт занят:** npm run dev -- --host 127.0.0.1 --port 8080.
* **Queue нужен в фоне:** оставь queue:work в отдельном терминале

**7) Краткая «шпаргалка»**

cd backend

cp .env.example .env

./vendor/bin/sail up -d

./vendor/bin/sail artisan queue:work --timeout=300

# frontend

cd ../frontend

npm ci

cp .env.example .env

npm run dev -- --host 127.0.0.1