

**ОБЩЕСТВО С
ОГРАНИЧЕННОЙ
ОТВЕТСТВЕННОСТЬЮ
«ВОЛЬТ»**

УТВЕРЖДАЮ

Генеральный директор

ООО «Вольт»



А.С. Румянцев

М.П.

« 19 » ноября 2025 г.

Программное обеспечение

Единое цифровое пространство электрозарядной инфраструктуры

Договор

от «05» сентября 2025 г. № 03/25/3735/25

о предоставлении права использования программного обеспечения

РУКОВОДСТВО АДМИНИСТРАТОРА

Листов 15

2025

Содержание

Определения, обозначения, сокращения	3
Введение	4
Назначение ПО	5
Условия применения программного обеспечения.....	6
1.1 Сфера деятельности администратора	6
1.2 Требования к подготовке администратора	6
1.3 Требования к используемым техническим средствам конфигурации	6
1.4 Описание установки ПО	7
1.5 Установка ПО через Docker.....	9
1.6 Описание переменных файла values.yaml	9
1.7 Проблема с БД.....	13
1.8 Проблема с Redis	13
1.9 Проблема с визуальной частью	13
1.10 Проблема с серверной частью	13
Приложение А (справочное) Общие сведения о документе	14

Определения, обозначения, сокращения

В настоящем документе применяют следующие сокращения с соответствующими обозначениями (см. Таблицу Таблица).

Таблица – Обозначения и сокращения

Сокращение	Расшифровка
ПО	Программное обеспечение
API	Application Programming Interface

В настоящем документе применяют следующие термины с соответствующими определениями (см. Таблицу Таблица).

Таблица – Термины и определения

Термин	Определение
ПО	Модуль формирования аналитической группировки образов
API	Интерфейс программирования приложений.
HTTP API	Программный интерфейс, основанный на стандарте HTTP
Docker	Программная платформа контейнеризации, обеспечивающая упаковку, передачу и изолированный запуск приложений в стандартных контейнерах
БД	База данных

Введение

Настоящий документ представляет собой Руководство администратора программного обеспечения «Единое цифровое пространство электрозарядной инфраструктуры» (далее – ПО), производства ООО «Вольт».

Настоящий документ содержит сведения о развертывании ПО.

Подробные сведения о документе приведены в Приложении А.

Назначение ПО

ПО представляет собой комплексную платформу с визуальным и программным интерфейсом для управления и мониторинга инфраструктуры зарядных станций электромобилей. ПО предназначено для:

- мониторинга состояния зарядной инфраструктуры;
- управления данными о зарядных станциях;
- обработки заявок граждан на установку станций;
- управления субсидиями и партнерами;
- генерации аналитических отчетов;
- предоставления информации инвесторам.

ПО предоставляет программный интерфейс API для управления функциями ПО. Предоставляется возможность протоколирования и мониторинга выполняемых действий с возможностью сбора показателей мониторинга внешним программным обеспечением.

Подробное описание функций ПО, а также логическая и техническая структура ПО приведены в документе «Описание программы».

Условия применения программного обеспечения

1.1 Сфера деятельности администратора

В сферу деятельности администратора входят следующие операции:

- Подбор оборудования развертывания ПО, удовлетворяющего требованиям к программно-аппаратной платформе.
- Развертывание ПО.
- Настройка в части интеграции ПО со смежными системами посредством программного интерфейса REST API.

1.2 Требования к подготовке администратора

Администратор ПО должен обладать знаниями и квалификацией на уровне senior linux administrator, владеть знаниями на уровне администратора в части программного обеспечения Kubernetes, docker, helm, а также знаниями в квалификации DevOps-инженер и программист REST API.

1.3 Требования к используемым техническим средствам конфигурации

К техническим средствам, используемым ПО, предъявляются следующие требования:

- ПО функционирует на ОС Linux (Ubuntu 20.04+, Debian 11+, CentOS 8);
- ПО стабильно функционирует с выполнением всех характеристик на следующих вычислительных ресурсах (не ниже):
 - CPU: 4+ ядер - RAM: 8+ GB - Диск: 50+ GB SSD, Сеть: 1 Gbps

Установка ПО

1.4 Описание установки ПО

- **Клонирование репозитория:**
 - `git clone <repository-url>`
 - `cd electropass`
 - **Настройка базы данных:**
 - **Создание базы данных PostgreSQL**
 - `createdb electropass`
 - **Включение PostGIS расширения**
 - `psql electropass -c "CREATE EXTENSION IF NOT EXISTS postgis;"`
 - **Настройка серверной части:**
 - `cd backend-main`
 - **Установка зависимостей**
 - `uv sync`
 - **Создание .env файла**
 - `cp .env.example .env`
 - **Редактирование .env (настройте DATABASE_URL и другие переменные)**
 - `nano .env`
- Пример .env файла:
 - `DATABASE_URL=postgresql://username:password@localhost:5432/electropass`
 - `REDIS_URL=redis://localhost:6379`
 - `JWT_SECRET=your-super-secret-jwt-key`

```
DEBUG=true  
CORS_ORIGINS=["http://localhost:3000"]
```

- **Применение миграций**

- `make migrate`

- **Запуск backend**

- `uv run app`

Backend будет доступен по адресу: `http://localhost:8000`

- **Настройка визуальной части**

- `cd frontend-main`

- **Установка зависимостей**

- `npm install`

- **Создание `.env.local` файла**

- `cp .env.example .env.local`

- **Редактирование `.env.local`**

- `nano .env.local`

- **Пример `.env.local` файла:**

- `NEXT_PUBLIC_API_URL=http://localhost:8000`
`NEXT_PUBLIC_MAP_API_KEY=your_yandex_maps_key`

- **Запуск визуальной части**

- `npm run dev`

Визуальная часть будет доступна по адресу: `http://localhost:3000`

- **Проверка работоспособности**

- Откройте `http://localhost:3000` в браузере

- Войдите в систему (используйте тестовые данные)

- Проверьте основные функции:
 - Карта зарядных станций
 - Аналитика
 - Управление заявками

1.5 Установка ПО через Docker

1.6 Описание переменных файла values.yaml

При установке ПО требуется создать и задать значения переменные в файле docker-compose.yaml. В случае если в приведенном ниже описании отсутствует какой-либо параметр из docker-compose.yaml оставьте его без изменений.

Общие значения

Создание docker-compose.yaml:

```
version: '3.8'
```

```
services:
```

```
frontend:
```

```
    build: ./frontend-main
```

```
    ports:
```

```
    - "3000:3000"
```

```
environment:
```

```
    - NEXT_PUBLIC_API_URL=http://backend:8000
```

```
depends_on:
```

```
    - backend
```

```
backend:
```

```
    build: ./backend-main
```

```
    ports:
```

```
    - "8000:8000"
```

```
environment:
```

```
    - DATABASE_URL=postgresql://user:pass@db:5432/electropass
```

```
    - REDIS_URL=redis://redis:6379
```

```
    - JWT_SECRET=your-jwt-secret
```

- DEBUG=true
- CORS_ORIGINS=["http://localhost:3000"]

depends_on:

- db
- redis

db:

image: postgres:14

environment:

- POSTGRES_DB=electropass
- POSTGRES_USER=user
- POSTGRES_PASSWORD=pass

volumes:

- postgres_data:/var/lib/postgresql/data

ports:

- "5432:5432"

redis:

image: redis:6-alpine

ports:

- "6379:6379"

volumes:

postgres_data:

Запуск ПО

Сборка и запуск всех сервисов

docker-compose up --build

Запуск в фоновом режиме

docker-compose up -d --build

После запуска необходимо убедиться в работоспособности:

Backend API доступен по адресу <http://localhost:8000>

Swagger документация доступна по адресу <http://localhost:8000/docs>

Frontend приложение доступно по адресу <http://localhost:3000>

База данных подключена и миграции применены

Redis подключен и работает

Аутентификация работает

Карта зарядных станций отображается

API endpoints отвечают корректно

Описание операций в части настройки ПО

Настройка ПО выполняется посредством его интеграции со смежными системами через программный интерфейс REST API

Обслуживание и устранение неисправностей

1.7 Проблема с БД

- Проверка подключения к PostgreSQL

```
psql -h localhost -U username -d electropass -c "SELECT version();"
```

- Проверка миграций

```
cd backend-main
```

```
uv run python -c "from app.db.piccolo_app import piccolo_app;
print(piccolo_app.tables)"
```

1.8 Проблема с Redis

- Проверка подключения к Redis

```
redis-cli ping
```

- Проверка через Python

```
uv run python -c "import redis; r = redis.Redis(); print(r.ping())"
```

1.9 Проблема с визуальной частью

- Очистка кеша Next.js

```
cd frontend-main
```

```
rm -rf .next
```

```
npm run dev
```

- Проверка переменных окружения

```
npm run build
```

1.10 Проблема с серверной частью

- Проверка зависимостей

```
cd backend-main
```

```
uv pip list
```

- Проверка конфигурации

```
uv run python -c "from app.config import get_settings;
print(get_settings())"
```

- Запуск в debug режиме

```
DEBUG=true uv run app
```

Приложение А**(справочное)****Общие сведения о документе****А.1 Происхождение документа**

Информация о происхождении документа приведена ниже (см. Таблицу Таблица).

Таблица – Происхождение документа

Документ создан:	18.11.2025
Ответственный:	Румянцев А.С.
Автор:	Румянцев А.С.
Принимали участие:	

А.2 История модификации документа

История модификации документа приведена ниже (см. Таблицу Таблица).

Таблица – История модификации документа

Версия	Дата	Комментарий	Автор
1.0	18.11.2025	Создание документа	Румянцев А.С.

Лист регистрации изменений

[illegible]