

**Инструкция по установке экземпляра программного обеспечения «Платформа Софиот», предоставленного для проведения экспертной проверки**

Оглавление

[**1. Общие рекомендации** 3](#_Toc203735542)

[**2. Установка и настройка СУБД MariaDB** 3](#_Toc203735543)

[**3. Установка и настройка веб-сервера Nginx** 4](#_Toc203735544)

[**4. Установка бэкэнд-приложения** 7](#_Toc203735545)

[**5. Запуск бэкэнд-приложения** 9](#_Toc203735546)

[**6. Установка и запуск веб-приложения** 9](#_Toc203735547)

# **1. Общие рекомендации**

При выполнении команд на Linux нужно использовать `sudo`.

# **2. Установка и настройка СУБД MariaDB**

1. Обновить пакетный менеджер:

sudo apt-get update

2. Установить MariaDB

sudo apt install mariadb-server

3. Запустить и включить MariaDB:

sudo systemctl start mariadb

sudo systemctl enable mariadb

4. Проверить статус MariaDB:

sudo systemctl status mariadb

5. Создать пользователя MySQL:

sudo mysql

CREATE USER 'root'@'localhost' IDENTIFIED BY '123';

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'localhost' WITH GRANT OPTION;

FLUSH PRIVILEGES;

6. Миграции для базы данных представлены в виде sql скриптов в папке `/DbUpdates`.

Если база данных ещё не создана выполнить скрипт `Init.sql`.

В противном случае - выполнить транзакции из файлов в формате `YYYY-MM-DD.sql` для обновления базы данных.

Далее нужно заполнить таблицы данными: `DeviceModels`, `Devices`, `MeasureTypes`, `Parameters`, `PacketElements`, `PacketElementTypes`, `Manufactures`, `EquipmentModels`.

В таблице `Parameters` для записей необходимо указать `EquipmentTypeId`.

# **3. Установка и настройка веб-сервера Nginx**

1. Установить nginx:

sudo apt-get install -y nginx

2. Проверить наличие `client\_max\_body\_size 3M;` для секции `location /api`. Это позволяет увеличить допустимый размер body у запросов (необходимо для сохранения картинок).

3. Скопировать файл `sofiot.conf` из архива в директорию `/etc/nginx/sites-available/`.

4. Создать ссылку для конфигурационного файла, используя команду:

sudo ln -sf /etc/nginx/sites-available/sofiot.conf /etc/nginx/sites-enabled/

5. Удалить из числа активных хостов хост по умолчанию:

sudo rm /etc/nginx/sites-enabled/default

6. Запустить диагностику nginx:

sudo nginx -t

В ответ должно отобразиться примерно следующее:

nginx: the configuration file /etc/nginx/nginx.conf syntax is ok

nginx: configuration file /etc/nginx/nginx.conf test is successful

7. Перезапустить nginx:

sudo systemctl reload nginx

8. Проверить статус nginx:

sudo systemctl status nginx

# 

# **4. Установка бэкэнд-приложения**

1. Установка зависимостей (зависимости для работы .NET приложения)

sudo apt-get update

sudo apt-get install -y dotnet-sdk-7.0

2. Настройка приложения

В файле `appsettings.json` можно изменить следующие настройки:

- Поменять порт:

"Kestrel": {

"Endpoints": {

"Http": {

"Url": "http://\*:8083"

}

}

}

- Указать настройки SMTP почты:

"AuthMessageSenderSettings": {

"Host": "mail.sofiot.ru",

"Port": 25,

"Username": "cabinet@sofiot.ru",

"Password": "Sofiot.Sofiot",

"EnableSsl": true

}

- Установить строку подключения MySQL:

"ConnectionStrings": {

"DefaultConnection": "server=localhost;port=3306;database=sofiot;user=root;password=123;SslMode=None"

}

- Указать Telegram-Бота для рассылки уведомлений:

"TelegramBotSettings": {

"BotName": "sofiot\_bot",

"BotToken": "TOKEN"

}

- Указать ключ сервера для капчи:

"Captcha": {

"Secret": "Secret"

},

- Указать ключи ЯндексID для авторизации через Яндекс

"Authentication": {

"Yandex": {

"ClientId": "ID",

"ClientSecret": "SECRET"

}

# **5. Запуск бэкэнд-приложения**

После установки Nginx на linux появится директория /var/www/

1. Необходимо расположить папку публикации backend в директории /var/www/

2. Установить права доступа для папки:

sudo chown -R www-data:www-data /var/www/backend

sudo chmod -R 755 /var/www/backend

3. Скопировать файл `sofiotbackend.service` в директорию /etc/systemd/system/

4. Перезагрузить systemd:

sudo systemctl daemon-reload

5. Запустить сервис:

sudo systemctl start sofiotbackend.service

6. Включить автозапуск сервиса:

sudo systemctl enable sofiotbackend.service

# **6. Установка и запуск веб-приложения**

1. Настройка Яндекс-капчи

В файле `.env` (`sofiot-front/.env`) фронтенд приложения необходимо поменять `NEXT\_PUBLIC\_YANDEX\_CAPTCHA\_KEY` на ключ клиента, полученный на сайте <https://yandex.cloud/ru/services/smartcaptcha>

2. Сборка веб-приложения

1. Установить nodejs на linux, если он ещё не установлен:

sudo apt install nodejs npm

2. Папка `sofiot-front` содержит нужные файлы для сборки проекта. Необходимо разместить папку `sofiot-front` в директории /tmp/

3. Установить зависимости (находясь в директории `/tmp/sofiot-front`):

cd /tmp/sofiot-front

npm install

4. Собрать проект (находясь в директории `/sofiot-front`):

cd /tmp/sofiot-front

npm run build

5. Создать папку frontend:

sudo mkdir -p /var/www/frontend

6. Установить права для папки:

sudo chmod -R 755 /var/www/frontend

7. Скопировать нужные файлы билда в /var/www/frontend:

sudo cp -r /tmp/sofiot-front/.next /var/www/frontend/

sudo cp -r /tmp/sofiot-front/public /var/www/frontend/

sudo cp /tmp/sofiot-front/package.json /var/www/frontend/

sudo cp /tmp/sofiot-front/package-lock.json /var/www/frontend/

8. Папка /var/www/sofiot-front/ больше не нужна, её можно удалить:

sudo rm -rf /tmp/sofiot-front

9. Перейти в новую директорию и установить зависимости:

cd /var/www/frontend

npm install

10. веб-приложение будет доступен на порту 8081. Он указан в `package.json`:

"start": "next start -p 8081"

3. Запуск приложения с помощью PM2

1. Установить процесс-менеджер для Node.js:

npm install -g pm2

2. Запустить приложение (выполнять команду в директории `/var/www/frontend`):

cd /var/www/frontend

pm2 start "npm run start" --name sofiot-front

3. Сохранить настройки PM2 для автоматического запуска после перезагрузки сервера:

pm2 save

pm2 startup

4. Управление приложением:

- Просмотр логов:

pm2 logs sofiot-front

- Перезапуск:

pm2 restart sofiot-front

- Остановка:

pm2 stop sofiot-front

- Удаление:

pm2 delete sofiot-front