|  |  |
| --- | --- |
| **СОГЛАСОВАНО**  Генеральный директор  **ООО «Софтел»**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Р. А. Мединцев/  «\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. |  |



**Руководство администратора**

**Оглавление**

[**1.** **MySQL** 3](#_Toc198655416)

[**2.** **Nginx** 4](#_Toc198655417)

[**3.** **Бэкенд** 5](#_Toc198655418)

[**4.** **Фронтенд** 10](#_Toc198655419)

# **MySQL**

Общие рекомендации:

* При выполнении команд на Linux нужно использовать `sudo`.

Если MySQL уже установлен и настроен - этот пункт можно пропустить.

1. Обновить пакетный менеджер:

sudo apt-get update

1. Установить MariaDB

sudo apt install mariadb-server

1. Запустить и включить MariaDB:

sudo systemctl start mariadb

sudo systemctl enable mariadb

1. Проверить статус MariaDB:

sudo systemctl status mariadb

1. Создать пользователя MySQL:

sudo mysql

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

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

FLUSH PRIVILEGES;

1. 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`.

# **Nginx**

1. Установить nginx:

sudo apt-get install -y nginx

1. Проверить наличие `client\_max\_body\_size 3M;` для секции `location /api`.

Это позволяет увеличить допустимый размер body у запросов (необходимо для сохранения картинок).

1. Скопировать файл `sofiot.conf` из архива в директорию `/etc/nginx/sites-available/`.
2. Создать ссылку для конфигурационного файла, используя команду:

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

1. Удалить из числа активных хостов хост по умолчанию:

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

1. 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

1. Перезапустить nginx:

sudo systemctl reload nginx

1. Проверить статус nginx:

sudo systemctl status nginx

# **Бэкенд**

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

sudo apt-get update

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

* 1. **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.corpitech.by",

"Port": 25,

"Username": "sofiot@corpitech.by",

"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": "7676074759:AAERE36R9Gc\_UPQXkNgeP5nCSYWYGLrMW-U"

}

* Указать ключ сервера для капчи:

"Captcha": {

"Secret": "ysc2\_xN4aVSEfwwKxajisnkWBiCoTPoityRyzHQvHh2ZH03666130"

},

* Указать ключи ЯндексID для авторизации через Яндекс

"Authentication": {

"Yandex": {

"ClientId": "48b2a949462d4675ab2a13d1ba33646f",

"ClientSecret": "90462b0e5e414ae4a9eb94c186b4f9b6"

}

}

* 1. **Подключение Telegram-Бота**

Если Telegram-Бот ещё не создан смотреть <https://botcreators.ru/blog/kak-sozdat-svoego-bota-v-botfather/>

Если бот создан:

1. Настройки существующего бота можно получить перейдя по ссылке <https://t.me/BotFather>
2. Необходимо заходить в *@BotFather* с того telegram аккаунта, с которого он создавался.
3. Чтобы посмотреть своих ботов нужно отправить боту команду

/mybots

1. После этого выбрать бота из предложенных.

В сообщении появятся кнопки управления ботом и строка вида:

Here it is: Софиот @sofiot\_bot.

What do you want to do with the bot?

1. Скопировать имя бота, например `sofiot\_bot` (указан после символа `@`), и вставить вместо BotName в appsettings.json.
2. В чате с *@BotFather* будут отображены кнопки управления ботом.
3. Нажать кнопку `API Token`.
4. Заменить BotToken в appsettings.json на токен полученный из сообщения.
5. Готово. Выбранный бот будет использоваться в качестве бота рассылки уведомлений.
   1. **Настройка Яндекс капчи**

Для создания яндекс капчи перейти на сайт <https://yandex.cloud/ru/services/smartcaptcha>

1. Зайти в яндекс консоль
2. Создать капчу (см. скриншот)
3. На главной странице отобразится капча. Для неё будет указан ключ клиента и ключ сервера, которые необходимо скопировать (см. скриншот).
4. Ключ сервера необходимо разместить в файле appsettings.json

"Captcha": {

"Secret": "ysc2\_xN4aVSEfwwKxajisnkWBiCoTPoityRyzHQvHh2ZH03666130"

},

1. Ключ клиента пригодится при настройке фронтенд приложения.
   1. **Настройка Яндекс ID**
2. Подключить Яндекс ID на сайте <https://yandex.ru/id/about>.

Смотреть скриншоты `Настройки\_ЯндексID.png`

1. Отображать виджет, если у пользователя нет авторизации в Яндексе: \*\*Нет\*\*.
2. Указать \*\*Доступ к данным\*\*: везде поставить галочки.
3. Платформы приложения: установить галочку \*\*Веб-сервисы\*\*.
4. Заполнить \*\*Redirect URI\*\* для веб-сервисов (см. картинку в качестве примера):

http://{ваш адрес}/signin-yandex

https://{ваш адрес}/signin-yandex

1. Не указывать \*\*Suggest Hostname\*\*
2. В файле `appsettings.json` установить `ClientID`, `Client secret` в соответствующую секцию:

"Authentication": {

"Yandex": {

"ClientId": "48b2a949462d4675ab2a13d1ba33646f",

"ClientSecret": "90462b0e5e414ae4a9eb94c186b4f9b6"

}

}

* 1. **Запуск API**

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

1. Скопировать файл `sofiotbackend.service` в директорию /etc/systemd/system/
2. Перезагрузить systemd:

sudo systemctl daemon-reload

1. Запустить сервис:

sudo systemctl start sofiotbackend.service

1. Включить автозапуск сервиса:

sudo systemctl enable sofiotbackend.service

# **Фронтенд**

* 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>

* 1. **Сборка фронтенда**

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

sudo apt install nodejs npm

1. Папка `sofiot-front` содержит нужные файлы для сборки проекта.

Необходимо разместить папку `sofiot-front` в директории /tmp/

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

cd /tmp/sofiot-front

npm install

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

cd /tmp/sofiot-front

npm run build

1. Создать папки frontend на linux:

sudo mkdir -p /var/www/frontend

1. Установить права для папки:

sudo chmod -R 755 /var/www/frontend

1. 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/

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

sudo rm -rf /tmp/sofiot-front

1. Перейти в новую директорию и установить зависимости:

cd /var/www/frontend

npm install

1. Фронтенд будет доступен на порту 8081. Он указан в `package.json`:

"start": "next start -p 8081"

* 1. **Запуск приложения с помощью PM2**

1. Установить процесс-менеджер для Node.js:

npm install -g pm2

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

cd /var/www/frontend

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

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

pm2 save

pm2 startup

1. Управление приложением:

* Просмотр логов: `pm2 logs sofiot-front`
* Перезапуск: `pm2 restart sofiot-front`
* Остановка: `pm2 stop sofiot-front`
* Удаление: `pm2 delete sofiot-front`