



Инструкция по скачиванию и установке АСУДД SmartTraffic

Москва

Содержание

1. Требования к аппаратному и системному программному обеспечению...	3
2. Скачивание и распаковка дистрибутива.....	4
3. Установка API (backend)	4
3.1. Подключение репозитория с sumo.....	4
3.2. Обновите информацию и новых пакетах	4
3.3. Установка зависимостей	4
3.4. Установка NoneJS	4
3.5. Создание базы данных	5
3.6. Копирование и настройка datasource.json и config.yaml.....	5
3.7. Изменения конфигурационных файлов datasource.json	5
3.8. Сборка фреймворка loopback	5
4. Установка frontend	7

1. Требования к аппаратному и системному программному обеспечению

1. Требования к аппаратному обеспечению:

- Объем оперативной памяти не менее 4 Гб.

2. Требования к системному программному обеспечению:

- Ubuntu Linux не ниже 16.04.6 LTS.

2. Распаковка дистрибутива

Архив дистрибутива `bg-development.zip` распакуйте в директорию `/opt/`<наименование директории>.

Предоставьте права на чтение, запись и изменение файлов в директории `bg/`, пользователю, от имени которого будут запускаться скрипты `node package manager`.

3. Установка API (backend)

3.1. Подключение репозитория с sumo

Команда на подключение:

```
sudo add-apt-repository ppa:sumo/stable.
```

В случае, если при выполнении команды на подключение репозитория проявится ошибка – «`command not found`», то в этом случае выполните команду на обновление репозитория:

```
apt-get update.
```

Далее установите пакет `software-properties-common`:

```
sudo apt-get install software-properties-common
```

3.2. Обновите информацию и новых пакетах

```
sudo apt-get update
```

3.3. Установка зависимостей

Список пакетов для установки:

- Redis
- PostgreSQL (`postgresql-client-10`, `postgresql-client-common`, `postgresql-common`, `postgresql-server-dev-10`, `postgis`)
- Wget
- Osmosis
- `libpq-dev`
- `python-dev`, `python-psycopg2`, `python3-dev`
- `python3-venv`
- `sumo`, `sumo-tools`, `sumo-doc`
- `gcc`

Выполните следующую команду:

```
sudo apt-get install redis postgresql-10 postgresql-client-10
postgresql-client-common postgresql-common postgresql-server-dev-10
wget osmosis libpq-dev python-dev python-psycopg2 postgis python3-venv
sumo sumo-tools sumo-doc gcc python3-dev
```

3.4. Установка NoneJS

Выполните команду:

```
sudo snap install node --classic -channel12
```

Дождитесь окончания установки

3.5. Создание базы данных

Перейдите в оболочку psql

```
sudo -u postgres psql
```

Выполните следующие команды для создания БД, пользователя,

```
CREATE DATABASE yourdbname;
CREATE USER youruser WITH ENCRYPTED PASSWORD 'yourpass';
GRANT ALL PRIVILEGES ON DATABASE yourdbname TO youruser;
```

Далее перейти в созданную базу данных и выполнить команду:

```
CREATE EXTENSION IF NOT EXISTS postgis;
```

3.6. Копирование и настройка datasource.json и config.yaml

Для копирования выполните команду:

```
cp api/server/example_datasources.json api/server/datasources.json
```

3.7. Изменения конфигурационных файлов datasource.json

В конфигурационном файле datasource.json, в блоке «postgres» укажите необходимые хост, логин, пароль, имя, название базы

- host
- Datasource
- Password
- Name
- User

Пример для блока postgres

```
"postgres": {
  "host": "localhost",
  "port": 5432,
  "url": "",
  "database": "bg",
  "password": "bguser",
  "name": "postgre",
  "user": "bguser",
  "connector": "postgresql"
```

3.8. Сборка фреймворка loopback

Выполните поочередно следующие команды (команды выполняются в директории opt/<наименование директории>/api):

```
sudo git config --global http.sslVerify false
sudo npm config set strict-ssl false
```

В процессе выполнения установки возможны ошибки, которыми можно пренебречь:

```
npm install --unsafe-perm
```

Выполните запуск скрипта подключения к БД:

```
npm run init-db
```

Далее выполните скрипт подключения пользователей:

```
npm run init-users
```

Затем выполните скрипт запуска sdk loopback:

```
npm run lb-sdk
```

После выполнения всех скриптов, запустите сервер:

```
npm start
```

Для проверки, перейдите по адресу вебсервера `http://<адрес сервера>:4000` или по адресу `http://<адрес сервера>:4000/explorer` для проверки работоспособности LoopBack API Explorer.

Пример страницы, показывающей, что веб-сервер работает, показан на рисунке 1.

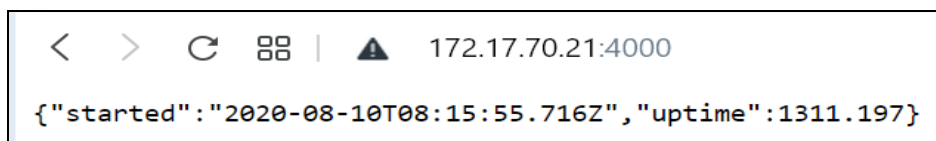


Рисунок 1 – Пример страницы при работающем сервере

Пример страницы REST API:

После перехода на страницу с REST API, необходимо дождаться загрузки моделей

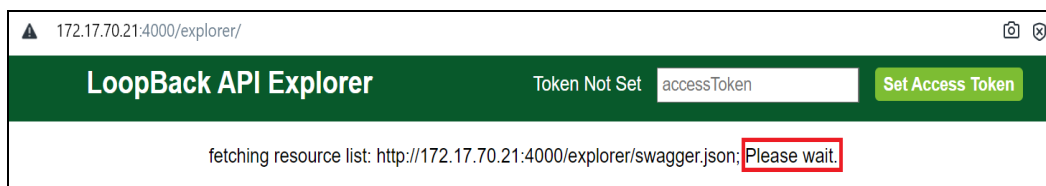


Рисунок 2 – Страница с REST API при ожидании загрузки

Страница с полностью загруженными моделями показана на рисунке 3.

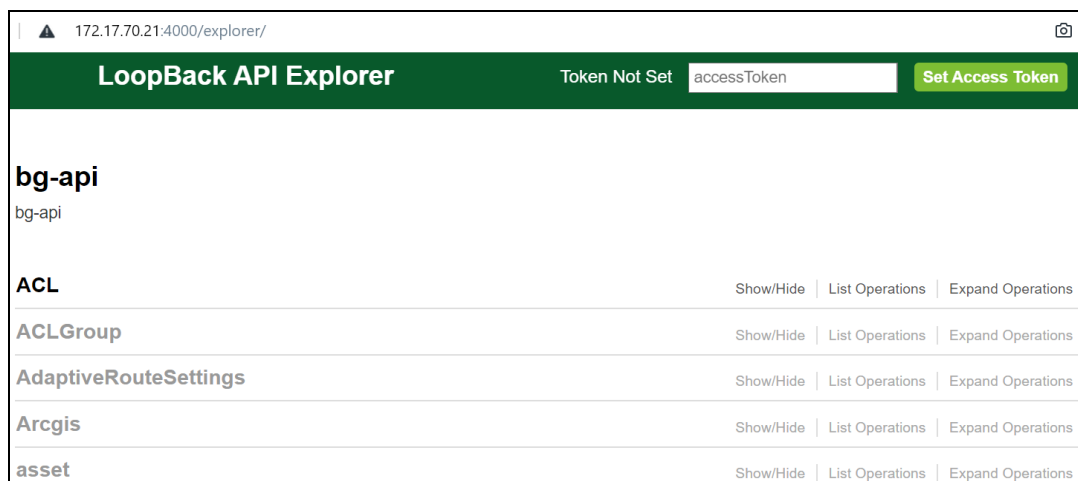


Рисунок 3 - Страница с полностью загруженными моделями

4. Установка frontend

Перейдите в директорию `opt/<наименование директории>`.

Скопируйте конфигурационный файл `default.config.json` в `config.json` при помощи следующей команды:

```
cp default.config.json config.json
```

Если проверка работоспособности frontend будет осуществляться на том же сервере, на котором развернут frontend, то корректировать `config.json` нет необходимости. Если обращение к frontend будет выполняться с другого ресурса, то в `config.json`, в блок `loopback` требуется внести изменения.

Пример конфигурационного файла `config.json` для API:

```
"loopback": {
  "baseUrl": "http://172.17.70.21:4000",
  "apiVersion": "api",
  "ioURL": "http://172.17.70.21:4000",
  "ioPath": "/socket.io"
```

Измените значения для параметров `baseUrl` и `ioURL`, на необходимые

Для запуска frontend выполните команду

```
node --max_old_space_size=4096 node_modules/@angular/cli/bin/ng serve --
host 0.0.0.0
```

Дождитесь окончания сборки и запуска сервера.

Далее перейдите по соответствующему адресу. Порт для работы с frontend – 4200.

Пример адреса `http://172.17.70.21:4200`

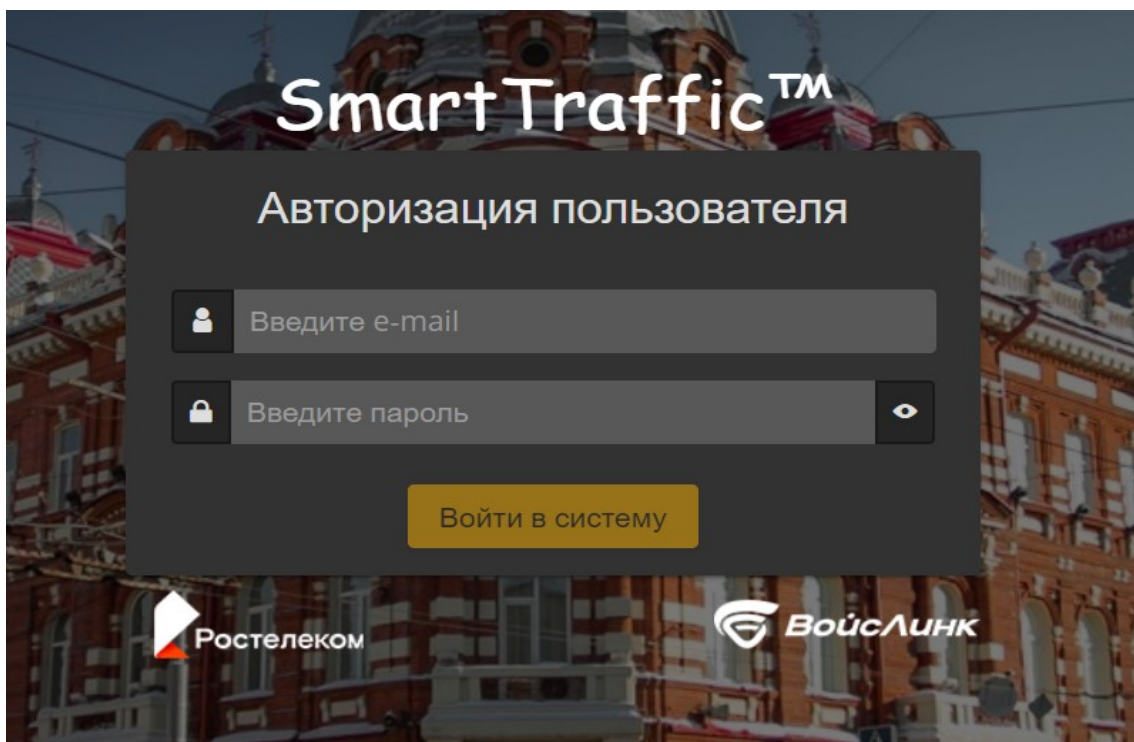


Рисунок 4 - Пример страницы входа в систему

В поле «ведите e-mail» укажите root@root.ru, в поле «Введите пароль» укажите пароль root. После ввода учетных данных откроется страница с картой Москвы.

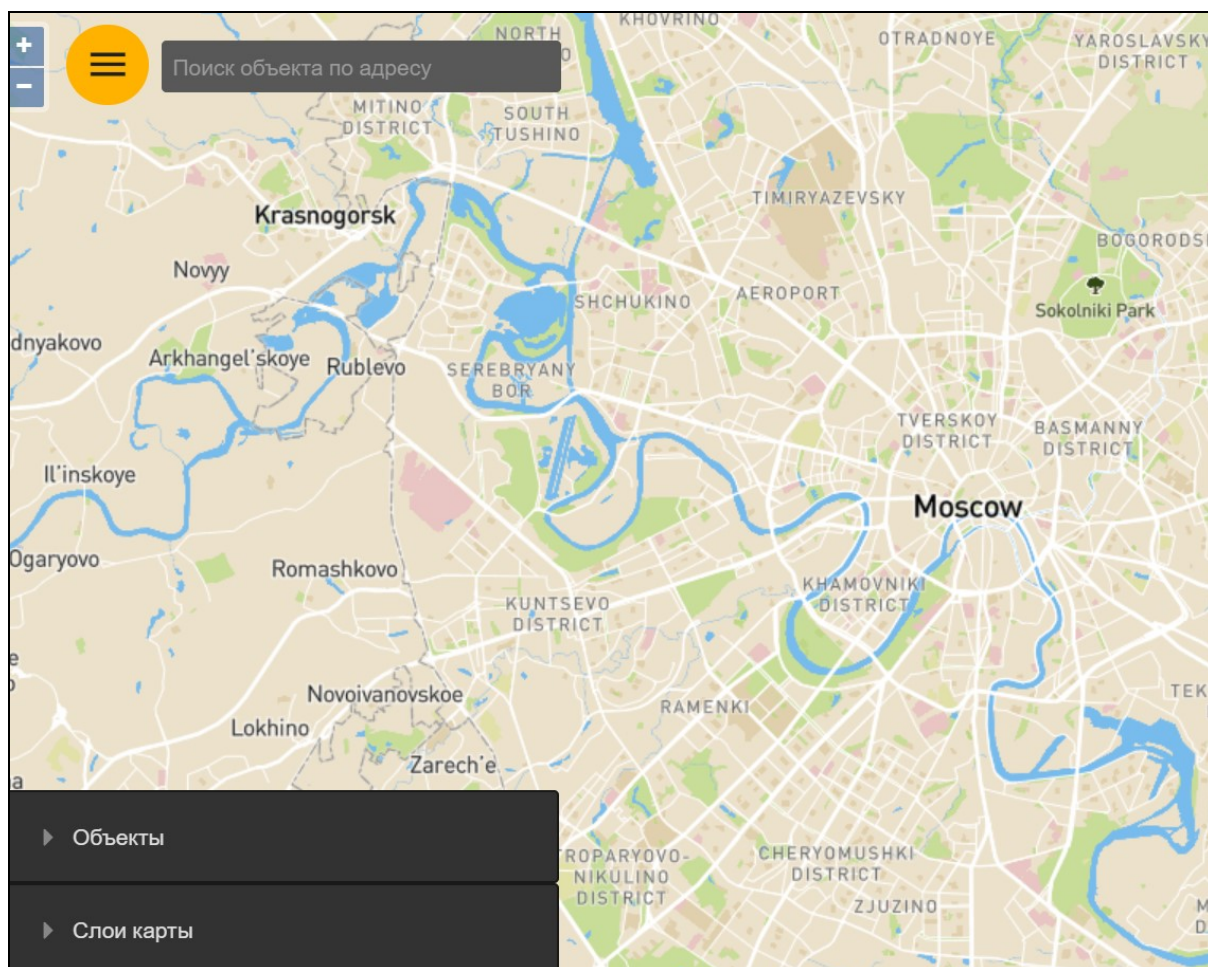


Рисунок 5 – Главное окно системы