



**ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК И
ИНФОРМАЦИЯ ПО УСТАНОВКЕ ЭКЗЕМПЛЯРА ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ**

Система поиска маршрутов Antereal Routes



1 СОДЕРЖАНИЕ

1	СОДЕРЖАНИЕ	2
2	ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК	3
2.1	ПОСТРОЕНИЕ ДОРОЖНОГО ГРАФА.....	3
2.1.1	Проверка данных.....	3
2.1.2	Построение графа	3
2.2	ПОСТРОЕНИЕ МАРШРУТОВ.....	3
2.2.1	Библиотека построения маршрутов	4
3	ИНФОРМАЦИЯ ПО УСТАНОВКЕ ПО.....	5
3.1	ГИС-СПЕЦИАЛИСТ. СЕТЬ ДОРОГ.....	5
3.1.1	Требования к программному обеспечению	5
3.1.2	Установка плагина	5
3.1.3	Удаление плагина	6
3.2	СЕРВИС ПОСТРОЕНИЯ МАРШРУТОВ	6
3.2.1	Минимальные системные требования	6
3.2.2	Предварительная настройка сервера	7
3.2.3	Предварительная настройка веб-сервера NGINX.....	7
3.2.4	Создание пользователя, управляющего приложением	9
3.2.5	Установка	9
3.2.6	Проверка работы сервиса.....	12
3.2.7	Мониторинг функционирования приложения	12
3.2.8	Конфигурация сервиса	12
3.2.9	Дорожный граф.....	16
3.2.10	Описание систем координат	16

2 ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК

2.1 ПОСТРОЕНИЕ ДОРОЖНОГО ГРАФА

Построение дорожного графа выполняется в настольной ГИС QGIS с помощью плагина «ГИС-Специалист. Сеть дорог».

Перед построением графа выполняется проверка данных.

2.1.1 ПРОВЕРКА ДАННЫХ

Инструмент предназначен для анализа данных на наличие проблем, препятствующих построению корректного дорожного графа:

- Разрывы дорог;
- Перехлесты дорог;
- Самопересечение объектов;
- Самоналожение объектов;
- Наложения участков дорог в одном уровне;
- Разрывы между дорогами и объектами.

Найденные ошибки отображаются в табличном виде, что существенно упрощает их локализацию и исправление.

2.1.2 ПОСТРОЕНИЕ ГРАФА

Инструмент предназначен для построения взвешенного и ориентированного графа для дальнейшего использования при построении маршрутов.

Дорожный граф состоит из:

- Вершин – начала и концы участков дорог, соединения и пересечения дорог.
- Ребер – участков дорог с геометрией и атрибутами, включая вес.
- Прочих служебных сущностей.

Результат построения дорожного графа – файл в формате GeoPackage, который будет использоваться **Сервисом построения маршрутов**.

2.2 ПОСТРОЕНИЕ МАРШРУТОВ

Построение маршрутов выполняется Сервисом, который предоставляет REST API. При этом используется ранее созданный в **Плагине** файл дорожного графа.

Основные возможности:

1. Поиск **маршрутов, состоящих из множества пунктов**, с условиями:
 - a. По типу передвижения:
 - i. **Автомобильный** – строится по дорогам, где может проехать транспорт.
 1. **Учитывать сезонные дороги**. Для построения маршрута используются дороги, действующие только в определенное время года (например, зимние дороги).
 2. **Учитывать проектируемые дороги**.
 - ii. **Вертолетный** – строится напрямую между пунктами (по воздуху).
 - b. По способу построения:

- i. С указанным порядком следования пунктов (неоптимальный).
 - ii. **Оптимальный.** Для определения оптимальности автомобильного маршрута учитывается длина участка, качество дорожного покрытия, разрешенная скорость и др. параметры. Для вертолетного маршрута учитывается только длина перелета между пунктами.
- 2. Учет запретов поворотов, одностороннего движения.
 - 3. Высокая скорость построения маршрутов.
 - 4. Одновременное построение тысяч маршрутов.
 - 5. Определение типов дорожного полотна, пересечений с искусственными преградами (мосты и пр.).
 - 6. Расчет длины маршрута на эллипсоиде.

2.2.1 БИБЛИОТЕКА ПОСТРОЕНИЯ МАРШРУТОВ

Перечисленные выше возможности **Сервиса** фактически реализуются в **программной библиотеке построения маршрутов («Библиотека»)**.

Библиотека (dll) принимает на вход **файл дорожного графа** и предоставляет **.NET API** для построения маршрутов.

Может подключаться в проекты на .NET, Python и на других фреймворках, позволяющих использование .NET-библиотек.

3 ИНФОРМАЦИЯ ПО УСТАНОВКЕ ПО

3.1 ГИС-СПЕЦИАЛИСТ. СЕТЬ ДОРОГ

3.1.1 ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Для корректной работы плагина необходим персональный компьютер под управлением операционной системы Windows 7/8/8.1/10, QGIS версии 3.34.6-3.40.

3.1.2 УСТАНОВКА ПЛАГИНА

1. Перейдите на вкладку «Установить из ZIP файла» и нажмите на значок выбора директории

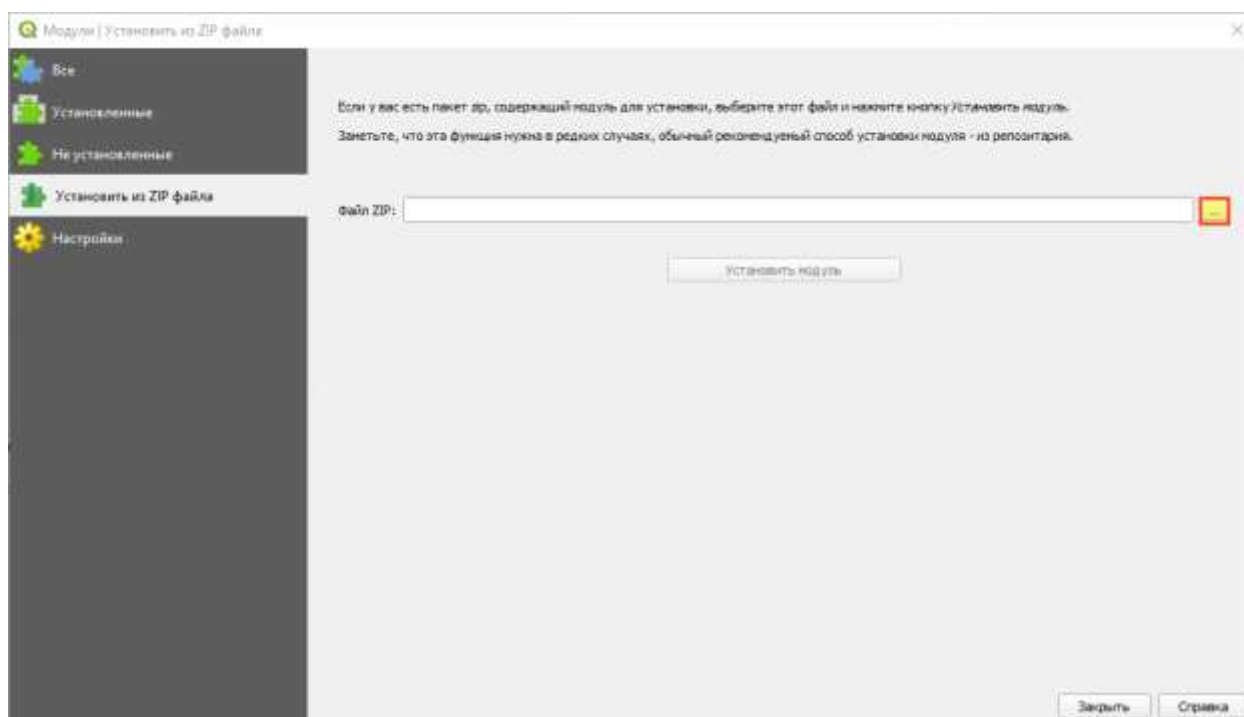


Рисунок 1. Значок выбора директории

2. Перейдите в директорию расположения архива и выберите .zip архив
3. Нажмите «Открыть»
4. Нажмите кнопку «Установить модуль»

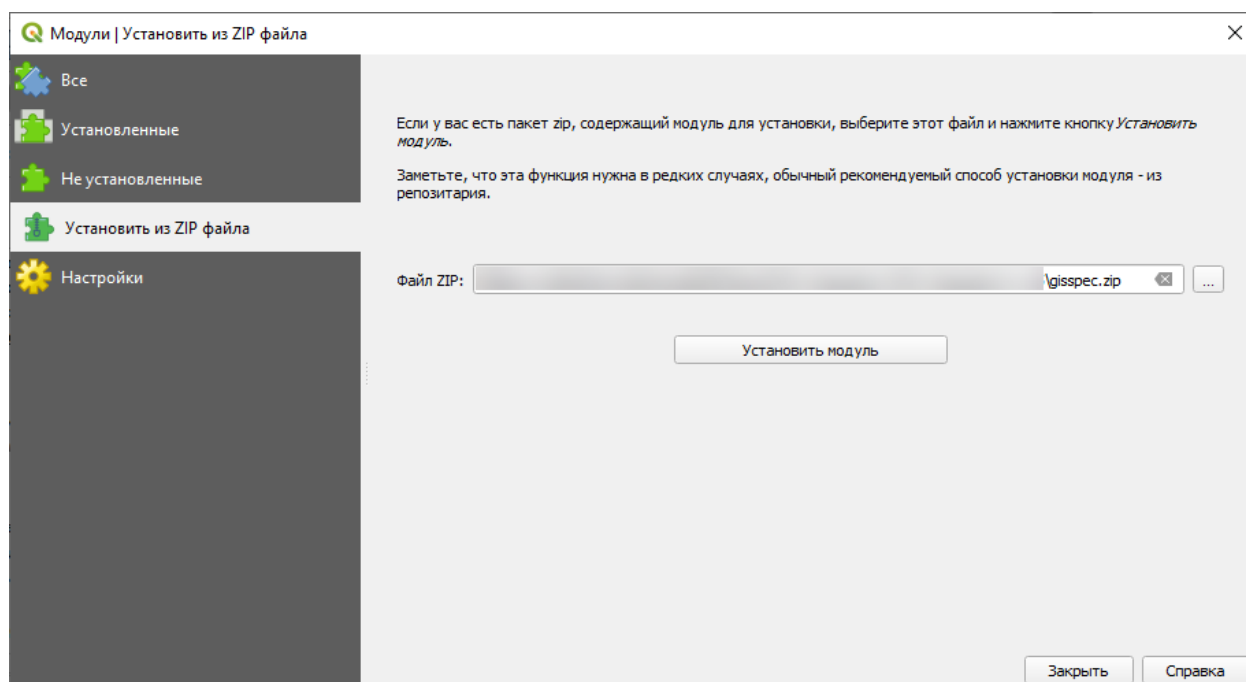


Рисунок 2. Установка

3.1.3 УДАЛЕНИЕ ПЛАГИНА

Если установлен плагин предыдущей версии, перед установкой новой версии его необходимо удалить, для этого:

1. Выберите плагин в списке.

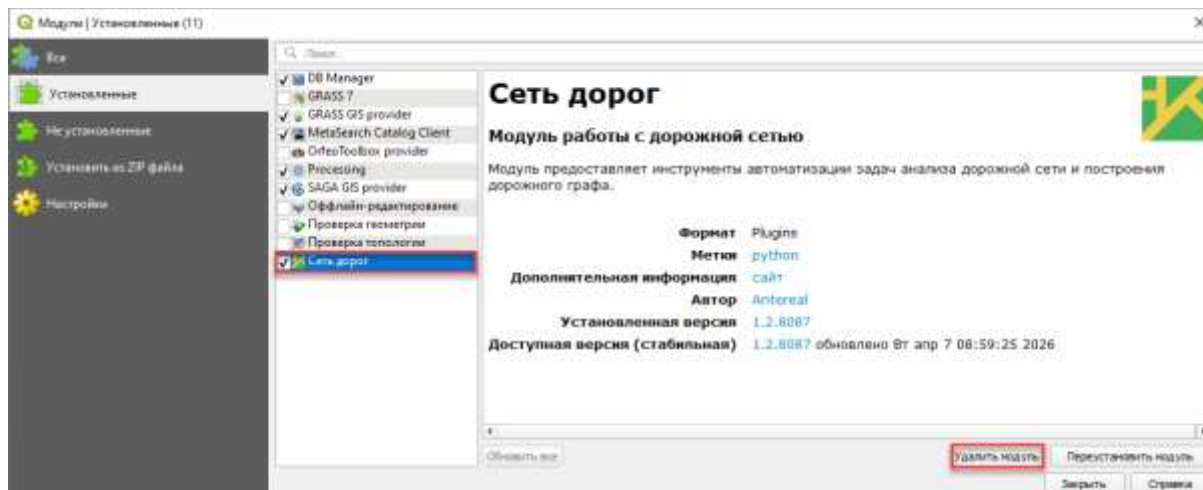


Рисунок 3. Выбор плагина

2. Нажмите кнопку «Удалить модуль»

3.2 СЕРВИС ПОСТРОЕНИЯ МАРШРУТОВ

3.2.1 МИНИМАЛЬНЫЕ СИСТЕМНЫЕ ТРЕБОВАНИЯ

1. ОС Linux x64, которые поддерживают ASP.Net Core 8.0
2. 110 Мб HDD для установки службы

Внимание!

Дополнительное дисковое пространство потребуется для размещения файлов дорожного графа и определений систем координат

3. 8+ Гб RAM
4. Установленные компоненты:
 - веб-сервер NGINX

3.2.2 ПРЕДВАРИТЕЛЬНАЯ НАСТРОЙКА СЕРВЕРА

Далее по инструкции потребуется выполнить разграничение прав доступа к исполняемым, конфигурационным и прочим файлам, необходимым для работы сервиса. Для этого предварительно на сервере, где планируется размещение сервиса, требуется создать группу и добавить в нее пользователя, который будет выполнять установку и обслуживание сервиса. Для примера, в данном документе будет использоваться группа **routeservice_admins**.

```
sudo groupadd routeservice_admins
sudo usermod -a -G routeservice_admins username
```

Внимание!

username - пользователь, который будет выполнять установку и обслуживание сервиса.

3.2.3 ПРЕДВАРИТЕЛЬНАЯ НАСТРОЙКА ВЕБ-СЕРВЕРА NGINX

Если требуется обеспечить возможность работы службы по защищенному протоколу HTTPS, необходима предварительная настройка веб-сервера NGINX. Далее приведена инструкция по выпуску самоподписанного сертификата и настройке NGINX.

3.2.3.1 ВЫПУСК САМОПОДПИСАННОГО СЕРТИФИКАТА

Все действия по выпуску сертификата производятся на любом сервере, работающем под управлением ОС из семейства Linux. Для выпуска сертификата требуется установка пакета **openssl**.

3.2.3.1.1 СОЗДАНИЕ КОРНЕВОГО СЕРТИФИКАТА

1. Создайте закрытый ключ и самоподписанный сертификат

```
openssl req -x509 -nodes -new -sha512 \
-days 5475 -newkey rsa:4096 -keyout ca.key \
-out ca.pem -subj "/C=RU/CN=COMPANYNAME"
```

Внимание!

COMPANYNAME - название вашей организации.

2. Создайте файл сертификата с расширением .crt:

```
openssl x509 -outform pem -in ca.pem -out ca.crt
```

3.2.3.1.2 СОЗДАНИЕ СЕРТИФИКАТА ДЛЯ ЦЕЛЕВОГО СЕРВЕРА

1. Создайте файл расширения x509 с названием **v3.ext**. Содержимое файла:

```
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment,
dataEncipherment
```

```
subjectAltName = @alt_names
[alt_names]
# Локальные хостинги
DNS.1 = localhost
DNS.2 = 127.0.0.1
DNS.3 = ::1
# Перечислите доменные имена
DNS.4 = hostname
DNS.5 = hostname.domain
DNS.6 = hostname.domain.ru
```

Внимание!

Обновите атрибуты DNS.4, DNS.5 и DNS.6 в соответствии с вашими параметрами домена. Обычно вместо hostname указывается имя сервера, для которого вы генерируете сертификаты.

2. Создайте закрытый ключ и запрос на подпись сертификата

```
openssl req -new -nodes -newkey rsa:4096 \
-keyout HOSTNAME.key -days 5475 -out HOSTNAME.csr \
-subj "/C=RU/ST=Russia/L=CITY/O=COMPANYNAME/CN=HOSTNAME"
```

Внимание!

CITY - название вашего города;
COMPANYNAME - название вашей организации;
HOSTNAME - имя сервера.

3. Создайте самоподписанный сертификат

```
openssl x509 -req -sha512 -days 5475 \
-extfile v3.ext \
-CA ca.crt -CAkey ca.key -CAcreateserial \
-in HOSTNAME.csr \
-out HOSTNAME.crt
```

Внимание!

HOSTNAME – имя сервера.

Далее, когда сертификат выпущен, необходимо перенести файлы **HOSTNAME.crt** и **HOSTNAME.key** на целевой сервер, если выпуск производился на другом сервере.

4.2. НАСТРОЙКА HTTPS В NGINX

Далее все действия выполняются на сервере, где предполагается размещение сервиса построения маршрутов.

1. Создать директорию хранения сертификатов:

```
sudo mkdir -p /etc/ssl/cert/
```

2. Разместить в директории файлы, созданные в предыдущем разделе, *.crt и *.key, под именами current_server.crt и current_server.key, соответственно;

3. Задать права доступа с разрешением на чтение для группы **routeservice_admins**:

```
sudo chown root:routeservice_admins -R /etc/ssl/cert/
sudo chmod 750 -R /etc/ssl/cert/
```

4. Изменить файл конфигурации `/etc/nginx/nginx.conf`, добавив в блок `server` следующие настройки:

```
server {  
    listen 443;  
    ssl on;  
    ssl_certificate /etc/ssl/cert/current_server.crt;  
    ssl_certificate_key /etc/ssl/cert/current_server.key;  
}
```

5. Перезагрузить nginx командой

```
sudo nginx -s reload
```

3.2.4 СОЗДАНИЕ ПОЛЬЗОВАТЕЛЯ, УПРАВЛЯЮЩЕГО ПРИЛОЖЕНИЕМ

Перед разворачиванием приложения должна быть создана учетная запись **routeservice**. Данная учетная запись будет использоваться для запуска службы и не предполагает вход пользователя в систему.

Т.к. далее по инструкции владельцем исполняемых файлов, файлов конфигураций и логов будет назначаться именно эта учетная запись, она должна быть включена в группу **routeservice_admins** (см. [раздел Предварительная настройка сервера](#)). Это необходимо для того, чтобы пользователь имел права на чтение и редактирование конфигурационных файлов и файлов лога.

```
sudo adduser --disabled-login routeservice  
sudo usermod -a -G routeservice_admins routeservice
```

3.2.5 УСТАНОВКА

1. Создать директории для сервиса и хранения его логов:

```
sudo mkdir /var/www/routeservice  
sudo mkdir /var/log/routeservice
```

2. Скопировать в `/var/www/routeservice` содержимое директории

Дистрибутив/routeservice.

3. Предоставить права доступа к директориям пользователю routeservice:

- 3.1. Изменить владельца директорий

```
sudo chown -R routeservice:routeservice_admins  
/var/www/routeservice  
sudo chown -R routeservice:routeservice_admins  
/var/log/routeservice
```

- 3.2. Задать права доступа к директориям

```
sudo chmod -R 770 /var/www/routeservice  
sudo chmod -R 770 /var/log/routeservice
```

- 3.3. Разрешить выполнение исполняемого файла

```
sudo chmod +x /var/www/routeservice/RouteService
```

4. Скопировать файл определения службы Дистрибутив/routeservice.service в директорию `/etc/systemd/system`.

Внимание!

Если не были выполнены шаги по предварительной настройке HTTPS в NGINX (см. раздел Предварительная настройка веб-сервера NGINX), необходимо убрать адрес `https://localhost:43124` в строке:
`Environment=ASPNETCORE_URLS=https://localhost:43124;http://localhost:43123`

5. Активировать и запустить службу:

```
sudo systemctl enable routeservice  
sudo systemctl start routeservice
```

6. Выполнить активацию лицензии (см. раздел Активация лицензии)

7. Выполнить команду для проверки, что служба работает.
У поля Active должно стоять значение active (running):

```
sudo systemctl status routeservice
```

3.2.5.1 АКТИВАЦИЯ ЛИЦЕНЗИИ

Когда лицензия еще не активирована сервис построения маршрутов при запуске выводит в лог сообщение о необходимости активации и завершает работу. Поэтому первого запуска сервиса необходимо:

1. Открыть файл лога в каталоге `/var/log/routeservice`;
Файл должен содержать следующую запись:

```
FATAL [2026-01-01 00:00:00,000] [DOMAIN\USERNAME]  
[RouteService.Program] [] Лицензия не активирована. Код активации:  
'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'. Ожидаемое расположение:  
'license/data.lic'.
```

2. Скопировать код активации и передать в службу технической поддержки Antereal (support@antereal.com). Служба технической поддержки предоставит файл лицензии;
3. Поместить файл лицензии по указанному в сообщении выше пути. По умолчанию это `license/data.lic` в каталоге установки сервиса - `/var/www/routeservice`.

Примечание:

При необходимости путь к файлу лицензии может быть переопределен в конфигурации приложения. Подробнее см. раздел [Конфигурация сервиса](#).

4. Выполнить перезапуск сервиса:

```
sudo systemctl restart routeservice
```
5. При успешной активации в лог не будет новых записей о необходимости активации, а сервис сразу будет готов к работе.

3.2.5.2 НАСТРОЙКИ NGINX

1. Создать файл `/etc/nginx/sites-available/routeservice.location.conf`:

```
location ^~ /routeservice {  
    if ($request_method = OPTIONS) {  
        add_header Content-Length 0;
```

```
add_header Content-Type text/plain;
add_header Access-Control-Allow-Methods "GET, POST, PUT, DELETE,
OPTIONS";
add_header Access-Control-Allow-Origin $http_origin;
add_header Access-Control-Allow-Headers "Authorization,Content-Type,X-
ServiceUserName";
add_header Access-Control-Allow-Credentials true;
return 200;
}

rewrite ^/routeservice(/.*)$ $1 break;

proxy_pass $scheme://routeservice-$scheme;
proxy_set_header Host $host;
proxy_http_version 1.1;
proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection $connection_upgrade;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
proxy_set_header X-Forwarded-Prefix /routeservice;
proxy_hide_header Server;
proxy_pass_header Set-Cookie;
proxy_set_header Authorization $http_authorization;
proxy_pass_header Authorization;
}
```

Внимание !

Строка конфигурации `add_header Access-Control-Allow-Origin $http_origin;` указывает, что обращение к сервису разрешено с любого домена. Замените `$http_origin` на нужный домен, либо удалите строку, если необходимо ограничить кросс-доменные запросы (CORS).

2. Создать файл `/etc/nginx/sites-available/routeservice.upstream.conf`. Порты в блоках `upstream` должны соответствовать портам в файле ***routeservice.service***.

```
upstream routeservice-http {
    server 127.0.0.1:43123;
    keepalive 32;
}
upstream routeservice-https {
    server 127.0.0.1:43124;
}
```

Внимание !

Блок `upstream routeservice-https` необходим только в том случае, если были выполнены шаги по предварительной настройке HTTPS в NGINX (см. раздел [Предварительная настройка веб-сервера NGINX](#)).

3. Установить права доступа:

```
sudo chown root /etc/nginx/sites-available/routeservice.location.conf
sudo chown root /etc/nginx/sites-available/routeservice.upstream.conf
sudo chmod 644 /etc/nginx/sites-available/routeservice.location.conf
sudo chmod 644 /etc/nginx/sites-available/routeservice.upstream.conf
```

4. В файле `/etc/nginx/sites-available/default` перед блоком `server` добавить строки:

```
map $http_upgrade $connection_upgrade {
    default upgrade;
    "" keep-alive;
}
```

```
include /etc/nginx/sites-available/*.upstream.conf;
```

5. Внутри блока `server` в его конце добавить строку:

```
include /etc/nginx/sites-available/*.location.conf;
```

6. Перезагрузить `nginx` командой:

```
sudo nginx -s reload
```

3.2.6 ПРОВЕРКА РАБОТЫ СЕРВИСА

Для проверки работы сервиса перейдите по ссылке на страницу документации для API `https://server/routeservice/swagger/`, заменив **server** на актуальное имя сервера.

Внимание!

В том случае, если шаги по предварительной настройке HTTPS в NGINX (см. раздел [Предварительная настройка веб-сервера NGINX](#)) не были выполнены, необходимо изменить схему с `https` на `http`.

3.2.7 МОНИТОРИНГ ФУНКЦИОНИРОВАНИЯ ПРИЛОЖЕНИЯ

Мониторинг функционирования приложения осуществляется через:

- Работющую страницу описания API (см. [Проверка работы сервиса](#))
- Текстовые логи:
 - `/var/log/routeservice/common.log` (см. параметры в `/var/www/routeservice/config/log4net.config`)

3.2.8 КОНФИГУРАЦИЯ СЕРВИСА

В каталоге установки сервиса `/var/www/routeservice` располагаются файлы конфигурации службы:

- `config/log4net.config` – файл конфигурации логирования. Представляет собой стандартный xml-файл настроек для `log4net`;
- `config/appsettings.json` – файл конфигурации сервиса.

Основные настройки, доступные в **`appsettings.json`**:

- секция `RouteService`
 - `roadsGraphPath` – путь к файлу графа (см. [Дорожный граф](#));

- `spatialReferencesPath` – путь к файлу с настройками систем координат (см. [Описание систем координат](#));
- `maximumRoadDistance` – радиус поиска ближайшей дороги в метрах по умолчанию;

Пояснение:

Значение данного параметра может быть переопределено клиентом в запросе на построение маршрута.

Значение параметра следует задавать исходя из максимальной величины погрешности при вводе координат пунктов маршрута. Не рекомендуется использовать слишком малые значения (менее 100), т.к. это может усложнить построение маршрута при ручном выборе местоположения пунктов, например, при помощи курсора на карте. Также не рекомендуется использовать слишком большие значения (более 10 000), т.к. это может влиять на производительность службы из-за избыточности доступных вариантов дорог в местах высокой плотности дорожной сети.

- `limitRoadSearchDistance` - верхняя граница радиуса поиска ближайшей дороги в метрах для клиентского значения параметра `maximumRoadDistance`;

Пояснение:

Задаёт ограничение для входного параметра запроса на построение маршрута. Необходимо для того, чтобы клиент не мог использовать слишком большие значения, и тем самым создавать чрезмерную нагрузку на вычислительные мощности и оперативную память сервера.

Рекомендации по выбору значения аналогичны параметру `maximumRoadDistance`.

- `buildParallelism` – количество параллельно рассчитываемых маршрутов. Задаёт количество потоков, доступных для расчетов маршрутов.

Пояснение:

Значение параметра следует задавать исходя из количества процессорных ядер, доступных системе, наличия нескольких экземпляров службы, а также нагрузки на сервер.

Если предполагается работа одного экземпляра службы на выделенном сервере, то рекомендуется в качестве значения использовать количество процессорных ядер. Если же предполагается использование нескольких экземпляров службы, то следует разделить количество доступных процессорных ядер между всеми экземплярами в равных долях.

Для ограничения нагрузки на сервер, создаваемой службой, можно уменьшить значение данного параметра, чтобы высвободить вычислительные мощности для других приложений, работающих на сервере.

- `maxRoutesInQueue` - максимальное количество маршрутов в очереди на построение.

Пояснение:

При превышении заданного количества сервис будет возвращать **429 Too Many Requests** в ответ на новые запросы построения до тех пор, пока не освободится место в очереди.

Значение данного параметра подбирается эмпирически исходя из ожидаемого количества клиентов, производительности службы и доступных вычислительных ресурсов.

Не рекомендуется задавать слишком малые значения, т.к. это может привести к слишком быстрому заполнению очереди, что будет усложнять работу клиентов со службой. Также не рекомендуется использовать слишком большие значения, т.к. это может приводить к чрезмерному потреблению оперативной памяти или слишком долгому ожиданию выполнения запросов.

- `useExitsByDefault` – использование выездов с объектов по умолчанию.

Пояснение:

Значение данного параметра может быть переопределено клиентом в запросе на построение маршрута.

- `storageTimeout` - время хранения результатов построения маршрута перед тем, как они будут безвозвратно удалены из оперативной памяти, в секундах.

Пояснение:

Если клиент не запросил результаты построения маршрута за заданное время, они будут удалены во время ближайшего обращения любого другого клиента после истечения таймута.

Рекомендуется задавать значение исходя из ожидаемого временного интервала между запросами клиента статуса построения маршрута. Не следует задавать значение меньше интервала между запросами, т.к. в этом случае клиент не будет успевать запрашивать результаты до их удаления. Также не следует использовать слишком большие значения, т.к. это может приводить к излишнему потреблению оперативной памяти в тех случаях, когда клиенты по какой-либо причине не запрашивают результаты (обрыв соединения, программный сбой на стороне клиента и пр.).

- `useRouteParamsLogging` - включить логирование запросов клиента и ответов службы (см. секцию `HttpLogin`).
- `maxRouteBuildTimeInSeconds` - максимальное время построения маршрута в секундах.

Пояснение:

Задаёт время в секундах, отведенное службе на построение одного маршрута. По истечении заданного времени процесс построения маршрута будет прерван.

Значение данного параметр подбирается эмпирически исходя из ожидаемой сложности маршрутов, плотности дорожной сети, производительности службы и доступных вычислительных ресурсов.

Не рекомендуется задавать слишком малые значения (менее 5), т.к. это может приводить к невозможности построения сложного и протяженного маршрута в местах высокой плотности дорожной сети. Также не рекомендуется задавать слишком большие значения (более 180), т.к. это может приводить чрезмерному потреблению оперативной памяти и нагрузке на вычислительные мощности при построении сложных маршрутов.

- секция `Logging`
 - секция `LogLevel` – настройка уровней логирования;
 - секция `Debug` - настройки вывода debug сообщений;
- секция `HttpLogging`
 - `LogRequestBody` - логировать тело запроса;
 - `LogResponseBody` - логировать тело ответа;
 - `RequestBodyLimit` - максимальная длина тела запроса в байтах для вывода в лог;
 - `ResponseBodyLimit` - максимальная длина тела ответа в байтах для вывода в лог;
- секция `Kestrel` - настройки встроенного в сервис веб-сервера;
 - секция `Certificates` - настройки сертификатов работы по протоколу HTTPS (см. раздел [Предварительная настройка веб-сервера NGINX](#));
 - Секция `Default`
 - `Path` - путь к .crt файлу;
 - `KeyPath` - путь к .key файлу;
 - `AllowInvalid` - указывает, следует ли учитывать недопустимые сертификаты, например самозаверяющие сертификаты.

Внимание!

В данной секции необходимо использовать те же сертификаты, что бы созданы для настройки NGINX (см. раздел [Предварительная настройка веб-сервера NGINX](#)). Если настройка HTTPS для NGINX не была выполнена, данный раздел конфигурации следует исключить.

Пример настроек в секции `Kestrel`:

```
"Kestrel": {
  "Certificates": {
    "Default": {
      "Path": "/etc/ssl/cert/current_server.crt",
      "KeyPath": "/etc/ssl/cert/current_server.key",
      "AllowInvalid": true
    }
  }
}
```

- Секция License - настройки лицензии;
 - Path - путь к файлу лицензии.

Примечание:

Подробнее про активацию лицензии см. в разделе [Активация лицензии](#).

3.2.9 ДОРОЖНЫЙ ГРАФ

Дорожный граф представляет собой файл в формате GeoPackage, который содержит все необходимые для работы сервиса пространственные данные.

Дорожный граф должен быть предварительно создан и размещен в любом доступном для сервиса хранилище. У пользователя и группы, используемых при развертывании службы (см. раздел [“Создание пользователя, управляющего приложением”](#)), должны быть права на чтение и редактирование файла:

```
sudo chown routeservice:routeservice_admins /path-to-geopackage
sudo chmod 770 /path-to-geopackage
```

Внимание!

/path-to-geopackage - путь к файлу дорожного графа.

3.2.10 ОПИСАНИЕ СИСТЕМ КООРДИНАТ

Файл **spatialReferences.json** содержит описание поддерживаемых сервисом систем координат в формате **wkt**.

Пример содержимого файла:

```
{
  "spatialReferences": [
    {
      "wkid": 4284,
      "wkt": "GEOGCS[\"Pulkovo
1942\", DATUM[\"Pulkovo_1942\", SPHEROID[\"Krassowsky
1940\", 6378245, 298.3, AUTHORITY[\"EPSG\", \"7024\"]], TOWGS84[23.57, -140.95, -
79.8, 0, 0.35, 0.79, -
0.22], AUTHORITY[\"EPSG\", \"6284\"]], PRIMEM[\"Greenwich\", 0, AUTHORITY[\"EPS
G\", \"8901\"]], UNIT[\"degree\", 0.017453292519943295, AXIS[\"Geodetic
latitude\", NORTH], AXIS[\"Geodetic longitude\",
EAST], AUTHORITY[\"EPSG\", \"4284\"]]"
    },
    {
      "wkid": 3857,
      "wkt": "PROJCS[\"WGS_1984_Web_Mercator_Auxiliary_Sphere\", GEOGCS[\"GCS_WGS_1984\""
```

```
,DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137.0,298.257223563]],PRIME["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Mercator_Auxiliary_Sphere"],PARAMETER["False_Easting",0.0],PARAMETER["False_Northing",0.0],PARAMETER["Central_Meridian",0.0],PARAMETER["Standard_Parallel_1",0.0],PARAMETER["Auxiliary_Sphere_Type",0.0],UNIT["Meter",1.0],AUTHORITY["EPSG",3857]]",
  // Параметры трансформации datum-ов: прямая трансформация
  "inputDatumTransform": [-23.57, 140.95, 79.8, 0, -0.35, -0.79, -0.22],
  // обратная трансформация
  "outputDatumTransform": [23.57, -140.95, -79.8, 0, 0.35, 0.79, 0.22]
}
]
```

Трансформации datum-ов задаются относительно СК дорожного графа, т.е. прямая трансформация - это перевод из входной СК в СК дорожного графа.

Файл описания систем координат должен быть предварительно создан и размещен в любом доступном для сервиса хранилище. У пользователя и группы, используемых при развертывании службы (см. раздел [“Создание пользователя, управляющего приложением”](#)), должны быть права на чтение и редактирование файла:

```
sudo chown routeservice:routeservice_admins /path-to-spatial-references
sudo chmod 770 /path-to-spatial-references
```

Внимание!

/path-to-spatial-references - путь к файлу опи