

RUMAP-PRO: Сервис анализа треков
Руководство по установке и настройке сервиса

Москва
2023

Аннотация

Данный документ содержит общее описание процесса установки и настройки «RUMAP-PRO: Сервиса анализа треков».

Во всех иных случаях предоставления доступа к сервису, в том числе демонстрационного или тестового, необходимо пользоваться той инструкцией, которая была передана менеджером проекта или иным ответственным лицом вместе с остальной документацией.

Оглавление

Перечень терминов, сокращений и обозначений	4
Введение	5
1. RUMAP-PRO: Сервис анализа треков	6
1.1. Общее описание.....	6
1.2. Принципы работы сервиса.....	7
1.3. Системные требования.....	7
1.4. Описание дистрибутива	8
1.5. Развертывание сервиса.....	8
1.6. Проверка работоспособности сервиса.....	9
1.7. Запуск и остановка сервиса	9
1.8. Деинсталляция сервиса	9
1.9. Настройка серверной части	10
1.10. Настройка функциональной части.....	12
1.11. Отладочная информация серверной части.....	16
1.12. Отладочная информация функциональной части	17

Перечень терминов, сокращений и обозначений

В настоящий документ введены специальные сокращения на русском и английских языках:

Сокращение /определение	Расшифровка
GeoJSON	открытый формат, предназначенный для хранения географических структур данных, основан на JSON.
БД	База данных
ПО	Программное обеспечение

Введение

В данном руководстве описан процесс установки и настройки «RUMAP-PRO: СЕРВИСА АНАЛИЗА ТРЕКОВ».

1. RUMAP-PRO: Сервис анализа треков

1.1. Общее описание

Сервис анализа треков («RUMAP-PRO: СЕРВИС АНАЛИЗА ТРЕКОВ») представляет собой российский геоинформационный онлайн-сервис от ЗАО «Геоцентр-Консалтинг», предназначенный для решения задач по нормализации треков движения автотранспорта и обогащения точек трека дополнительной информацией.

Сервис выполняет задачи нормализации треков движения автотранспорта относительно улично-дорожной сети, получения дополнительной информации по точкам трека, получения сводного отчета о превышениях скорости при движении по треку. Используется для работы с треками движения автотранспорта в веб-сайтах, мобильных, настольных приложениях, логистических системах, системах управления автотранспортом, страховой телематики.

Основные возможности сервиса:

- нормализация треков движения автотранспорта относительно улично-дорожной сети;
- обогащение трека дополнительной информацией (например, названия улиц и разрешенная скорость движения в каждой точке трека);
- реконструкция треков движения автотранспорта (выполняет задачу заполнения разрывов в треках движения автотранспорта для отображения данных без разрывов);
- получение сводного отчета по треку, включая:
 - ✓ наличие превышений скорости движения;
 - ✓ общий пройденный километраж;
 - ✓ километраж, пройденный по платным дорогам;
 - ✓ километраж, пройденный по дорогам системы «Платон»;
 - ✓ километраж, пройденный по дорогам без покрытия;
 - ✓ движение по улицам с односторонним движением в неправильном направлении.

В зависимости от режимов работы сервис позволяет получать краткое резюме по треку, обогащать исходный трек информацией о дорогах, дополнять исходный трек промежуточными точками, проецировать точки трека на граф дорожной сети, получать линию маршрута в соответствии с графом дорожной сети или список ребер графа дорожной сети, соответствующий исходному треку.

1.2. Принципы работы сервиса

Сервис работает по протоколу HTTP(S).

Параметры запроса передаются в формате QUERY STRING, значения параметров закодированы в PERCENT-ENCODING.

Запрос к сервису имеет следующий общий формат:

```
http://<адрес_сервера>/<вид_запроса>?guid=XXXXXXXX-XXXX-XXXX-XXXXXXXXXXXX&<остальные_параметры>
```

Значения указанных в строке запроса параметров описываются ниже в данном документе.

Результатом запроса является ответ в формате GeoJSON (JSON) в зависимости от вида запроса (регулируется параметром «формат выдачи результатов» (showMode)), в кодировке UTF-8.

Принципы обработки параметров:

- обработка параметров проводится без учета регистра (case insensitive);
- при отсутствии или ошибочном значении необязательного параметра в запросе используется значение по умолчанию;
- порядок следования параметров в запросе не имеет значения;
- при передаче нескольких одинаковых параметров в запросе используется значение первого параметра, все остальные значения игнорируются;
- при выполнении POST запроса в теле запроса должен содержаться документ в формате GeoJSON.

1.3. Системные требования

Для установки экземпляра «RUMAP-PRO:СЕРВИС АНАЛИЗА ТРЕКОВ» требуется наличие следующих ресурсов:

Процессор	не менее 2 ядер
Оперативная память	не менее 12Гб
Дисковое пространство	не менее 20 Гб. Размер дискового пространства зависит от размера обрабатываемых данных
Операционная система	Astra Linux Common Edition или Debian (ОС с открытой лицензией)

Требования к программно-аппаратному комплексу (ПАК) определяются по запросу в зависимости от территории, объема данных, необходимого или планируемого количества запросов к сервису и т.п.

1.4. Описание дистрибутива

Имя файла (набора данных)	Описание файла
/serverSettings.ini	ini файл с настройками серверной части (порт, число процессов)
/GCSPeedInspector.ini	ini файл с настройками сервиса получения отчета
/GCServiceInstall.sh	установщик сервиса
/GCServiceInstall_NoFirewall.sh	установщик сервиса без внесения правил исключения порта сервиса из брандмауэра
/GCServiceUninstall.sh	деинсталлятор сервиса
/GCServiceUninstall_NoFirewall.sh	деинсталлятор сервиса без удаления правил исключения порта сервиса из брандмауэра
/GCServiceStartAsApp.sh	запуск сервиса в режиме отладки (как приложение)
/SimpleRestServer	бинарный файл реализующий сервис http
/GCSPeedInspectorSRS.so	библиотека сервиса анализа треков
/GCSPeedInspectorSRS.sig	сгенерированный ключ для работы в режиме, защищенном лицензией
/data	директория данных сервиса
/lib	директория подключаемых библиотек для работы сервиса
/logs	директория с файлами логов
/plugins	директория подключаемых плагинов для работы БД

1.5. Развертывание сервиса

Для установки сервиса анализа треков необходимо выполнить следующие действия от имени пользователя root:

```
unzip GCSpeedInspector.zip -d /opt/ GCSpeedInspector
bash /opt/ GCSpeedInspector /GCServiceInstall.sh
```

1.6. Проверка работоспособности сервиса

Для проверки работоспособности сервиса анализа треков необходимо выполнить команды от имени пользователя root или пользователя с правами sudo:

```
Sudo service GCSpeedInspector status
```

В результате выполнения указанных команд должны быть получены сообщения «Active: active (running)», которые означают, что сервис работает в штатном режиме.

1.7. Запуск и остановка сервиса

Для запуска сервиса анализа треков необходимо выполнить команду от имени пользователя root:

```
service GCSpeedInspector start
```

Для остановки сервиса анализа треков необходимо выполнить команду от имени пользователя root:

```
service GCSpeedInspector stop
```

1.8. Деинсталляция сервиса

Для удаления сервиса анализа треков необходимо выполнить следующие действия от имени пользователя root:

```
/opt/ GCSpeedInspector /GCServiceUninstall.sh
rm -rf /opt/ GCSpeedInspector
```

1.9. Настройка серверной части

Настройка серверной части сервиса анализа треков осуществляется путем редактирования параметров, которые содержатся в конфигурационных файлах:

- /GCSpeedInspector/serverSettings.ini – настройки функционального модуля сервиса анализа треков;

Все конфигурационные файлы представлены в стандартном формате .ini.

Настройки функционального модуля сервиса анализа треков (/GCSpeedInspector/serverSettings.ini).

```
; Пропуск строки

; Настройки серверной части
[server]

; Номер порта, на котором работает сервис
port=4860

; Число параллельно запущенных процессов обработки информации,
; при nTreads=0 количество запущенных процессов будет равно количеству ядер
процессора
nTreads = 4

; Настройки функционального модуля
[lib]

; Путь и название файла - функционального модуля сервиса
path = ./GCSpeedInspectorSRS

; Рабочая директория, относительно которой определяются пути к информации,
; необходимой для работы сервиса. По умолчанию остается пустой
workingDirectory=./

; Настройки логирования
[logs]

; Признак необходимости записи логов
; write=1 - логи записываются
; write=0 - логи не записываются
write=1

; Детализация логов
; onlyErrors=1 - в логи сохраняются только ошибки запросов (например,
неправильные параметры запроса)
; onlyErrors=0 - в логи сохраняется вся информация (обычное логирование
запросов в формате W3C и ошибки запросов)
onlyErrors=0

; Название сервиса (для логов и для операционной системы)
softwareName=GCSpeedInspector

; Описание сервиса (для операционной системы)
softwareDescription="Геоцентр-консалтинг: Сервис анализа треков"
```

```
; Путь к каталогу с логами (создается автоматически)
dir=logs/

; Размер буфера для лога запросов (байт, 0 - нет буфера)
messageLogBuffer=0

; Размер буфера для лога ошибок (байт, 0 - нет буфера)
errorLogBuffer=0

; Режим записи отладочной информации для запросов в server.log
; debugRequests=0 - режим отключен
; debugRequests=1 - режим включен
debugRequests=0

; Настройки ограничения доступа (при наличии менеджера лицензий)
[license]

; Проверка лицензии
; check=1 - проверять лицензию
; check=0 - не проверять лицензию
check=0

; Прочие настройки
[connections]

; Замена IP адреса клиента в логах на содержимое из header'a запроса (X-
Forwarded-For)
; Актуально при перенаправлении запросов на сервис из других веб-серверов
(arache-httpd, IIS и проч)
; replaceIpWithXForwardedFor=0 - замена отключена
; replaceIpWithXForwardedFor=1 - замена активирована
replaceIpWithXForwardedFor=0

; Максимальная длина очереди запросов (количество запросов)
; При maxQueueLength=0 длина очереди запросов не ограничена
maxQueueLength=0

; Сжатие ответов при передаче информации
; allowCompression=1 - сжатие ответов gzip включено
; allowCompression=0 - сжатие ответов отключено
allowCompression=1

; Минимальный размер пакета для сжатия (байт)
; Параметр используется при включенном сжатии allowCompression=1
minLengthToCompress=512000

; Добавление длины данных в header ответа сервиса
addConnectionAndContentLengthHeaders=1

; Таймаут ожидания подключения (мс)
waitForConnectedTimeout=30000

; Таймаут ожидания получения данных (мс)
waitForReadyReadTimeout=30000

; Таймаут ожидания записи данных (мс)
waitForBytesWrittenTimeout=30000

; Таймаут ожидания закрытия соединения (мс)
```

```
waitForDisconnectedTimeout=30000
```

1.10. Настройка функциональной части

Настройка функциональной части сервиса анализа треков осуществляется путем редактирования параметров, которые содержатся в конфигурационных файлах:

- /GCSpeedInspector/GCSpeedInspectorSettings.ini – настройки функционального модуля сервиса анализа треков;

Все конфигурационные файлы с расширением .ini представлены в стандартном формате .ini.

Настройки функциональных модулей анализа треков (/GCSpeedInspector / GCSpeedInspectorSettings.ini).

```
; Пропуск строки
[attributes]
;название поля в исходных атрибутах, содержащего скорость
speed=velocity
;название поля в исходных атрибутах, содержащего время
time=unixtime
;название поля в исходных атрибутах, содержащего азимут
azimuth=direction
;название поля, откуда берется исходный азимут (копируется из названия
поля в исходных атрибутах, содержащего азимут)
azimuthForGeocode=azimuthindeg
[geocode_attributes]
;название поля, содержащего id прогеокодированного ребра
edgeId=objectIndex
;название поля, содержащего obj_cod прогеокодированного ребра
obj_cod=OBJ_COD

oneway=ONEWAY
rd_name=RD_NAME
speedlim=SPEEDLIM
azimuth=azimuthInDeg
distance=distance
geocode_object_id=geocode_object_id
geocode_object_id_num=geocode_object_id_num
projection=projection
[logs]
```

```
;использовать логирование работы сервиса
write=1
;название сервиса (отображается в названии лог файлов)
softwareName=SpeedInspectorSystem
;путь к логам сервиса
dir=logs/

[Debug]
;выводить отладочную информацию
debugOutput=0
;перечитывать ini при каждом обращении к сервису
debug_reloadIni=0
;признак, что установлен режим отладки
debugMode=0
;отображать текст и тело запроса
showQuery=0
;папка, куда складываем всю отладочную информацию
debug_outputDirectory=D:/temp_d/projects/QT/SpedInspector/data/DebugOut
;для отладки после фильтрации выставлять скорость 200
debugSetSpeed200AfterFilter = 0
;название gdb куда будут складываться все выходные слои
debug_gdb_name=out.gdb
;bat файл для конвертации из geojson в gdb
debug_geojsonToGdbBat=D:/temp_d/projects/QT/SpedInspector/data/gdal/GeoJsonToGdb.bat
;количество мсек, которое должен работать сервис над запросом, чтобы в
логе писать профилировку
debugElapsedMinimum=100
;печатать варианты маршрутов указанного столбца
debugRoutesForColumnPrint=107
;столбец, для которого можно посмотреть текущие маршруты
debugRouteColumn=1555
;отображать старые поля original,middle,externalRoute,internalRoute
debugShowOldFields=0
;нужно ли логировать запрос в пригодном для Jmeter формате
logQuery=0
;хост, который потом будет в логах для jmeter
debugHostForLog=192.168.1.85:4860
;искать треки с грунтом
debugSearchSurface=1
```

```

[Settings]
;расстояние, которым интерполируются пропуски вначале, м
interpolation_distance=130
;расстояние, которым интерполируются пропуски в итоге, м
interpolation_distance_final_m=150
;расстояние, м на котором считается, что точка слишком близко к interInter
nearInterInterDistance_m=30;
;расстояние, м на котором считается, что точка слишком близко к перекрестку
nearJunctionDistance_m=15;

;сколько метров добавить, чтоб искать связность после тоннеля
tonnel_addition=100
;максимально возможная скорость (для расчета реалистичности расстояния),
км/ч
maxSpeed=350;
;отображать все точки, а не только превысившие скорость
dont_filter_result_points=1
;максимально допустимое расстояние между точками при интерполяции (верхняя
граница при угле между точками =0 градусов)
filter_skip_point_distance_max=1500
;максимально допустимое расстояние между точками при интерполяции (нижняя
граница при угле между точками>=60 градусов)
filter_skip_point_distance_min = 500;
;длина, начиная с которой подозреваем движение по "хоботу"
trunk_length=500;
;максимальное время между точками, сек.
maxTimeSec=300
;неиспользуемый параметр
maxcount=1000
;азимут по умолчанию
azimuthinddeg=80
;допуск по азимуту
azimuthdeltaindeg=40
;штраф в метрах, дающий преимущество основной дороге перед дублером
frontagePenalty_m=0
route_active=1
;базовый адрес обращения к геосервису
;не использовать Localhost!!! писать конкретный адрес!!!
;geoservice_address=http://localhost:4851/batchreversegeo?&pattern=Nearest
Lines&spatialin=EPSG:4326&guid=30C66EF6-5FD4-4E50-8632-869C0FC08F8B

```

```

;geoservice_address=http://gis01.rumap.ru:4851/batchreversegeo?&pattern=NearestLines&spatialin=EPSG:4326&guid=30C66EF6-5FD4-4E50-8632-869C0FC08F8B
;название поля, содержащего идентификатор объекта во входных данных
route_address="http://routing.digimap.ru/directions?{%22speed%22:%22online%22,%22license%22:%2230C66EF6-5FD4-4E50-8632-869C0FC08F8B%22,%22edges%22:[{%22id%22:<id1>},{%22id%22:<id2>}],%22vehicles%22:{%22car%22:{}},%22changeDelay%22:10,%22simplify%22:false,%22return%22:[%22directions%22],%22routealternative%22:false}"
in_idfield_name=track_seg_point_id
;название поля, содержащего идентификатор геокодированного объекта
out_idfield_name=geocode_object_id
;файл, в котором прописана связность между ребрами графа
;sqliteFile=/data/rumap/RUSSIA_rugraph65_2018_06_15T12_26_06.sqlite
;новый rumap
sqliteFile=/data/rumap/russia_no_quarters.sqlite
;максимальное число точек на входе
point_limit=10000
;максимальное число прогеокодированных вариантов
maxAlternativePointsCount=7
;на сколько далеко вокруг дырки проверять связь точек маршрута
holeColumnsCheckLinks=5
;отображать выключенные точки (только добавлять им поле "switchedOff":1)
showSwitchedOffRoutePoints=1
;максимальное расстояние поиска связности,м
maxLinkDistance=400
;название поля, в котором хранится признак, что точка взята из источника, а не получена в результате фильтрации
originalPointFieldName=original
;межквартальные проезды -----
-----
;максимальная скорость для определения межквартальных проездов
maxQuarterlySpeed_kmh=20
;расстояние первоначальной идентификации межквартальных проездов
maxQuarterlyDistance_m=300
;минимальное количество точек для идентификации межквартальных проездов
minQuarterlyCount=4
;расстояние для окончательной идентификации межквартальных проездов
nearestQuarterlyDistance_m=200
;скачок азимута, который идет в общий учитывается при идентификации межквартальных проездов
deltaAzimuth_degree=100

```

```

;минимально непрогеокодированных точек у общему количеству должно быть
quarterlyNonGeocodedProportion=0.3
;Расстояние от ребра,м, на котором точка не может быть межкварталкой
nonQuarterEdgeDistance_m=20
;максимально допустимая скорость в группе медленной группе
slowGroupMaxGroupSpeed30=30
;скорость, с которой (или ниже) должны существовать точки, относящиеся к
медленной группе
slowGroupVerySlowSpeed=20
;скорость, начиная с которой медленная группа однозначно рвется
slowGroupVeryLargeSpeed=40
;максимальное допустимое число относительно быстрых точек, при котором
медленная группа еще мерджится с другой медленной группой
slowGroupMaxNonMergedFastPointsCount=2
rGeocodeDataDir = /data/rumap
;название слоя, содержащего информацию по всему графу
rGeocodeDataTline_infoLayerName=tline_info
;название слоя, содержащего информацию некоторым ребрам графа для поиска
движения во встречном направлении
rGeocodeDataTline_info_reversed_onewaysLayerName=tline_info_reversed_onewa
ys
;радиус поиска
maxdist=70
maxdist_opposite=10
;-----
-----

;максимальное количество просматриваемых маршрутов в оптимизаторе
maxRoutesCount=300

```

1.11. Отладочная информация серверной части

Каждый функциональный модуль сервиса анализа треков осуществляет запись и хранение отладочной информации (логов) в формате W3C Log File Format.

Ежедневно создается два файла с логами:

- <название сервиса>mes<YYYY><MM><DD>.log – подробная информация о всех запросах к сервису;
- <название сервиса>err<YYYY><MM><DD>.log – информация об ошибочных запросах к сервису .

Архивация логов происходит автоматически за предыдущий календарный месяц в архивы:

- <название сервиса>mes<YYYY><MM>.zip для логов с подробной информацией о всех запросах к сервису
- <название сервиса>err<YYYY><MM>.zip для логов с информацией об ошибочных запросах к сервису

Настройки параметров записи логов содержатся в конфигурационном файле, описанном в соответствующем разделе документации.

Кроме того, для каждого функционального модуля сервиса анализа треков создаются три дополнительных лог-файла, которые могут быть использованы для проверки работоспособности сервиса и анализа его функционирования:

- server.log – системный лог событий сервиса;
- <название функционального модуля>.log – перенаправление потока вывода stderr (логи сторонних библиотек);
- <название функционального модуля>_o.log – перенаправление потока вывода stdout (логи сторонних библиотек).

Дополнительные лог-файлы не архивируются и не защищены от переполнения, поскольку не содержат большого количества информации. В нормальном режиме работы должны быть пустыми и не содержать сообщений.

1.12. Отладочная информация функциональной части

Отладочная информация функциональной части каждого функционального модуля сервиса анализа треков представляет собой набор файлов вида <название функционального модуля>-<YYYY>-<MM>-<DD>.log

Каждый файл содержит телеметрию процессов, происходящих внутри функционального модуля, и может использоваться для мониторинга правильной работы фоновых процессов.

Формат записи информации:

- системная дата/время события;
- тип сообщения;
- текст сообщения.

Глубина хранения информации – 1 неделя в режиме ротации без архивирования и хранения истории.