

RUMAP-PRO: Сервис прокладки маршрутов
Руководство по установке и настройке сервиса

Москва
2022

Аннотация

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

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 прокладки маршрутов	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	9
1.10. Настройка функциональной части	11
1.11. Отладочная информация серверной части	16
1.12. Отладочная информация функциональной части	17

Перечень терминов, сокращений и обозначений

Используемые в настоящем документе термины и основные понятия области автоматизированных систем определены в ГОСТ 34.003-90. Также в тексте настоящего документа приведены следующие термины.

Термин/сокращение	Расшифровка
JSON	(JavaScript Object Notation) - текстовый формат для передачи данных. Он был основан на JavaScript и обычно с ним и применяется
query string	«строка запроса». Значение строки запроса присоединяется к адресу файла через знак «?» и отображается в адресной строке
REST	Стиль построения архитектуры распределенного приложения
UTF-8	(от англ. Unicode Transformation Format, 8-bit — «формат преобразования Юникода, 8-бит») — распространённый стандарт кодирования текста, позволяющий более компактно хранить и передавать символы Юникода, используя переменное количество байт (от 1 до 4)
ОС	Операционная система
РФ	Российская Федерация
ТС	Транспортное средство

Введение

В данном руководстве описан процесс установки и настройки «RUMAP-PRO:Сервис прокладки маршрутов».

1. RUMAP-PRO: Сервис прокладки маршрутов

1.1. Общее описание

Сервис предназначен для решения широкого спектра задач, связанных с построением маршрутов по графу транспортной сети. Весь функционал сервиса, описанный в данном документе, доступен при наличии полной лицензии на сервис. В случае неполной лицензии часть функций сервиса может быть недоступна.

Первый блок задач, решаемых сервисом - прокладка маршрутов. Маршруты прокладываются между двумя заданными точками "старт"/"финиш" и произвольным количеством промежуточных точек с учетом фактической организации дорожного движения. Точки задаются координатами, либо адресами (в случае подключения "RuMap: Сервис прямого геокодирования"). В результате работы сервис выдает маршрут в виде линии, текстовое описание маршрута, общие характеристики маршрута (длина, время). Данный набор функций используется для прокладки маршрутов в веб-сайтах, мобильных, настольных приложениях, логистических и аналитических системах. Построение маршрутов возможно:

- с учетом критерия оптимизации маршрута "по минимальному времени прохождения маршрута" или "по кратчайшей протяженности маршрута";
- с учетом способа перемещения (легковой/грузовой автомобильный, общественным транспортом, пешеходный);
- с учетом и без учета "пробок" (с учетом текущей скорости автомобильного потока на каждом отрезке улицы/дороги), а также с учетом прогноза загруженности дорожной сети на 1 неделю вперед.

Второй блок задач, решаемых сервисом - это построение зон доступности. Зоны доступности представляют собой полигоны - участки территории для заданной точки, из которой можно доехать к любой другой точке улично-дорожной сети этого участка не более, чем в заданный промежуток времени. Результатом построения зоны доступности является полигон в формате GeoJSON. Данный режим используется для построения зон доступности для дальнейшего анализа или получения дополнительных статистических показателей по ним в веб-сайтах, мобильных, настольных приложениях, системах геомаркетинга, пространственного анализа и проч. Построение зон доступности возможно по тем же критериям, что и построение маршрутов.

В своей работе сервис использует специально подготовленные данные единого транспортного графа, информацию об актуальных скоростях движения транспорта, прогнозные скорости движения транспорта на неделю вперед.

При прокладке маршрута с учетом общественного транспорта используется интервальная модель, основанная на расписаниях движения общественного транспорта.

Маршруты прокладываются при помощи алгоритма Дейкстры с использованием набором оптимизаций и альтернативных подходов.

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>?<параметры_в_формате_JSON>`

Значения указанных в строке запроса параметров описываются ниже в данном документе. Запрос POST выполняется таким же образом, при этом параметры в формате JSON передаются в теле запроса.

Результатом запроса является ответ в формате GeoJSON в кодировке UTF-8.

В сервисе реализована возможность выполнения следующих видов запросов:

1. Запрос на построение маршрутов (directions).
2. Запрос на построение зоны доступности от точки (reachabilityLim).
3. Запрос на построение зоны доступности до точки (reachabilityToPointLim).
4. Запрос на определение расстояний до множества точек (reachabilityMatrixLim).

1.3. Системные требования

Для установки экземпляра сервиса прокладки маршрутов требуется наличие следующих ресурсов:

Процессор	не менее 2 ядер
Оперативная память	не менее 24 Гб
Дисковое пространство	Не менее 80 Гб. Размер дискового пространства зависит от размера обрабатываемых данных
Операционная система	Astra Linux Common Edition или Debian (ОС с открытой лицензией)

1.4. Описание дистрибутива

Имя файла (набора данных)	Описание файла
/GCRoutePlanner.zip	Комплект программного обеспечения и данных для сервиса прокладки маршрутов
/GCRoutePlanner.zip/RoutePlanner.so	Функциональный модуль прокладки маршрутов
/GCRoutePlanner.zip/RoutePlanner.ini	Конфигурационный файл функционального модуля прокладки маршрутов
/GCRoutePlanner.zip/GCServiceInstall.sh	Скрипт установки функционального модуля прокладки маршрутов
/GCRoutePlanner.zip/GCServiceInstall_NoFirewall.sh	Скрипт установки функционального модуля прокладки маршрутов без открытия портов
/GCRoutePlanner.zip/GCServiceStartAsApp.sh	Запуск функционального модуля прокладки маршрутов в режиме приложения
/GCRoutePlanner.zip/GCServiceUninstall.sh	Скрипт удаления функционального модуля прокладки маршрутов
/GCRoutePlanner.zip/GCServiceUninstall_NoFirewall.sh	Скрипт удаления функционального модуля прокладки маршрутов без закрытия портов
/GCRoutePlanner.zip/serverSettings.ini	Конфигурационный файл веб-сервера функционального модуля прокладки маршрутов
/GCRoutePlanner.zip/SimpleRestServer	Веб-сервер функционального модуля прокладки маршрутов
/GCRoutePlanner.zip/lib	Вспомогательные библиотеки функционального модуля прокладки маршрутов
/GCRoutePlanner.zip/plugins	Вспомогательные библиотеки функционального модуля прокладки маршрутов
/GCRoutePlanner.zip/spatialite-4.3.0a	Вспомогательные библиотеки функционального модуля прокладки маршрутов
/GCRoutePlanner.zip/data	Набор данных для функционального модуля прокладки маршрутов

1.5. Развертывание сервиса

Для установки сервиса прокладки маршрутов необходимо выполнить следующие действия от имени пользователя root:

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

1.6. Проверка работоспособности сервиса

Для проверки работоспособности сервиса прокладки маршрутов необходимо выполнить следующие действия от имени пользователя root:

```
service GCRoutePlanner status
```

В результате выполнения указанной команды должно быть получено сообщение «Active: active (running)», которое означает, что сервис прокладки маршрутов работает в штатном режиме.

1.7. Запуск и остановка сервиса

Для запуска сервиса прокладки маршрутов необходимо выполнить команду от имени пользователя root:

```
service GCRoutePlanner start
```

Для остановки сервиса прокладки маршрутов необходимо выполнить команду от имени пользователя root:

```
service GCRoutePlanner stop
```

1.8. Деинсталляция сервиса

Для удаления сервиса прокладки маршрутов необходимо выполнить следующие действия от имени пользователя root:

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

1.9. Настройка серверной части

Настройка серверной части сервиса прокладки маршрутов осуществляется путем редактирования параметров, которые содержатся в конфигурационном файле GCRoutePlanner\serverSettings.ini.

Конфигурационный файл представлен в стандартном формате .ini.

Ниже приведен состав конфигурационного файла с краткими комментариями к параметрам и их настройке.

```
; Пропуск строки

; Настройки серверной части
[server]

; Номер порта, на котором работает сервис
port= 9065

; 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 = ./RoutePlanner.so

; 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= GCRoutePlanner

; Описание сервиса (для операционной системы)
softwareDescription="Геоцентр-консалтинг: GCRoutePlanner"

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

; Размер буфера для лога запросов (байт, 0 - нет буфера)
messageLogBuffer=4048

; 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 прокладки маршрутов осуществляется путем редактирования параметров, которые содержатся в конфигурационном файле GCRoutePlanner\RoutePlanner.ini.

Конфигурационный файл представлен формате ini.

Ниже приведен состав конфигурационного файла с краткими комментариями к параметрам и их настройке.

```
; Базовые настройки
[RoutePlanner/Base]
; Внимание! Журналирование влияет на скорость многопоточного приложения.
; Для повышения быстродействия следует задать ограничения:
;
; logMode = "QtCriticalMsg,QtFatalMsg"
; log2Console = false
; log2File = true
; на рабочем сервере.
;
; log2File = false полностью отключит запись журнала в файл

; Режим журналирования, перечень типов выводимых сообщений через запятую, по
умолчанию включены все типы
; enum QtMsgType { QtDebugMsg, QtWarningMsg, QtCriticalMsg, QtFatalMsg,
QtInfoMsg, QtSystemMsg = QtCriticalMsg };
; logMode="QtDebugMsg,QtWarningMsg,QtCriticalMsg,QtFatalMsg,QtInfoMsg,QtSystem
Msg"
logMode="QtDebugMsg,QtCriticalMsg,QtFatalMsg"

; Формат вывода сообщений в журнал, QT_MESSAGE_PATTERN, см. документацию Qt
; по умолчанию "[%{time yyyyMMdd h:mm:ss.zzz t} %{if-debug}D%{endif}%{if-
info}I%{endif}%{if-warning}W%{endif}%{if-critical}C%{endif}%{if-
fatal}F%{endif}] %{file}:%{line} - %{message}"
logFormat="[%{time yyyyMMdd h:mm:ss.zzz} %{if-debug}D%{endif}%{if-
info}I%{endif}%{if-warning}W%{endif}%{if-critical}C%{endif}%{if-
fatal}F%{endif}] %{file}:%{line} - %{message}"

; Включение вывода журнала в консоль.
log2Console = false

; Включение вывода журнала в файл.
log2File = false

; Путь к журналам.
logPath = ./logs

; Необязательный параметр, префикс имени лог-файла.
logFilePrefix = "RoutePlanner."

; режим транслитерации вывода. По-умолчанию равно false.
logTransliterate = false

; Настройки сервиса прокладчика
[RoutePlanner/Service]
; Включает режим периодической перезагрузки базы данных из файлов по шаблону
; Экспериментальный режим, названия и состав параметров могут поменяться
databaseReloadingEnable=true
; Интервал проверки обновления файла базы данных в мс (при появлении нового
файла в директории произойдет перезагрузка)
databaseUpdateInterval=60000

; Интервал в который обновления делать не следует
; databaseUpdateGap="2:10-2:40"

; Включает или отключает использование данных о пробках при прокладке
маршрута. [true, false]
```

```

trafficReloadingEnable = true

; Интервал перезагрузки файла базы данных скоростей.
; При появлении нового файла в директории произойдет перезагрузка,
; по таймауту файл будет перезагружаться, поскольку загружается не весь файл,
; а временной срез.
trafficDatabaseUpdateInterval = 60000
trafficDatabaseUpdateInterval = 300000

; Интервал перезагрузки файла базы данных прогноза скоростей.
; При появлении нового файла в директории произойдет перезагрузка
trafficPredictionDatabaseUpdateInterval=300000

; Путь данным о пробках в максимально свободное время (максимальные
скорости).
;trafficJamsMaximumPath=
"/data/TrafficJams/History/tline_source_max_speeds/tline_source_max_speeds_{
MaxISODate}.sqlite"
trafficJamsMaximumPath =
"/opt/TrafficJamsRuMap/History/tline_source_max_speeds/tline_source_max_speed
s_{MaxISODate}.sqlite"
;trafficJamsMaximumPath=
"/mnt/traffic_jams/History/tline_source_max_speeds/tline_source_max_speeds_{M
axISODate}.sqlite"

; Использовать фиктивную дату для загрузки исторических пробок, например
"2016-10-24". По умолчанию "".
; trafficJamsMaximumDebugDate = "2016-10-24"

; Путь к обновляемым данным о пробках.
;trafficJamsOnlinePath=
"/data/TrafficJams/Online/rumap/tline_source_{MaxISODate}.sqlite"
trafficJamsOnlinePath =
"/opt/TrafficJamsRuMap/Online/rumap/tline_source_{MaxISODate}.sqlite"
;trafficJamsOnlinePath=
"/mnt/traffic_jams/Online/rumap/tline_source_{MaxISODate}.sqlite"

; Использовать фиктивную дату для загрузки онлайн пробок, например "2016-10-
24". По умолчанию ""
;trafficJamsOnlineDebugDate = "2016-10-24"

; Путь к историческим данным о пробках.
; trafficJamsHistoryPath =
./data/TrafficJams/History/rumap/tline_source_{MaxISODate}.sqlite
; trafficJamsHistoryPath =
Z:/History/rumap/{HistoryMonth}/tline_source_{MaxISODate}.sqlite
; trafficJamsHistoryPath =
"/opt/TrafficJamsRuMap/History/tline_source_max_speeds/tline_source_max_speed
s_{MaxISODate}.sqlite"

; Путь к данным прогноза пробок.
; trafficJamsPredictionPath =
./data/TrafficJams/Prediction/rumap/tline_source_{MaxISODate}.sqlite
; trafficJamsPredictionPath =
Z:/Prediction/rumap/tline_source_{MaxISODate}.sqlite
trafficJamsPredictionPath =
"/opt/TrafficJamsRuMap/Prediction/rumap/tline_source_{MaxISODate}.sqlite"

; Включает или отключает использование оптимизации arc-flags
arcFlagsEnable = false

```

```

; Настройки алгоритма прокладчика
[RoutePlanner/Calculation]
; максимальное количество итераций увеличения радиуса поиска ребер,
radius*pow(2,radiusIterations)
maxRadiusIterations=10
; максимальный радиус поиска ребер
maxRadius=1000
; Скорость ТС по умолчанию, в км/ч
defaultSpeed = 30
; Скорость пешехода по умолчанию, в км/ч.
defaultPedestrianSpeed = 4
; Упрощать выводимый маршрут объединяя участки с одинаковым типом маневра и
названием улицы
simplifyDirections = true
; Максимальное количество попыток проложить маршрут
attemptsCount = 1
; Включает или отключает использование коэффициента ожидания на остановке ОТ
(учет интервалов движения ОТ)
pubTransportIntervalsEnabled=true
; Коэффициент ожидания на остановке ОТ (процент от максимального времени
между рейсами ОТ)
waitingRatio=0.9
; максимальное расстояние, на котором учитываются пешеходные ребра при
построении маршрута ОТ
maxDistanceFromPublicTransport = 500
; максимальное расстояние от дорог 1-2 уровня
maxDistanceFromHighways = 4000
; радиус в метрах для экстенда финишной зоны, используется для отключения
учета ппт дорог маршрутов автотранспорта
maxFinishDistance = 2000
; радиус в метрах для экстенда финишной зоны, используется для отключения
учета maxDistanceFromPublicTransport на конечном этапе построения маршрута ОТ
maxFinishDistanceFromPublicTransport = 5000
; максимальное количество альтернативных маршрутов
alternativeAttemptsCount = 3
; включение построения альтернативных маршрутов
routeAlternative = false
; максимальное количество стартовых ребер
maxStartingEdgesCount = 3
; максимальное количество финишных ребер
maxEndingEdgesCount = 3
; максимальное расстояние от старта и до финиша маршрута в метрах, на котором
ребрам не увеличивается стоимость при повторной прокладке
ignoredAlternativeDistance = 1000
; Коэффициент ограничения на длину/время альтернативного маршрута, если
полученный маршрут длинее/дольше в заданное кол-во раз, то не выводим его
alternativeUpperLimitFactor = 3.5
; максимальное количество циклов обратной трассировки
maxBacktraceIterations=50000
; коэффициент увеличения fullSearchExtent для ограничения прокладки по ребрам
1-3 уровня
searchExtentFactor = 10.0
; плотность ниже которой прокладка разрешена всегда
densityIgnoreValue = 1.5
; плотность ниже которой прокладка разрешена для ребер длиной от 5 до 10 км
densityLessThanAvgLength5_10 = 3.402767
; плотность ниже которой прокладка разрешена для ребер длиннее 10 км
densityLessThanAvgLength10 = 2.696413

; коэффициенты умножения стоимости ранее использованных ребер альтернативных
маршрутов по видам транспорта
[RoutePlanner/Alternative]
car=1.5,1.6,1.7,1.8,1.9,2.0
pedestrian=1.5,1.6,1.7,1.8,1.9,2.0

```

`all=1.5,1.6,1.7,1.8,1.9,2.0`

`; Настройки алгоритма прокладчика`

`[RoutePlanner/Reachability]`

`; Необязательный параметр! Путь к данным кеша зон доступности автоматически формируется как имя_бд_графа_reachability.bin и сохраняется рядом с БД`

`; если его задать то файл будет записан по указанному пути а не рядом с БД`

`; cacheBasePath = ./data`

`; Максимальное значение порогового значения зоны доступности в секундах`

`; maxCost = 2400`

`; Максимальное значение максимального расстояния движения вне УДС в секундах`

`; maxOffroadMaxDistKm = 600`

`; Максимальное значение стоимости километра при движении вне УДС в секундах`

`; maxOffroadCostPerKm = 600`

`[RoutePlanner/ReachabilityLim]`

`; Максимальное значение порогового значения зоны доступности в секундах`

`maxMaxCost = 1800`

`; Максимальное значение максимального расстояния движения вне УДС в секундах`

`; maxOffroadMaxDistKm = 600`

`; Максимальное значение стоимости километра при движении вне УДС в секундах`

`; maxOffroadCostPerKm = 600`

`[ReachabilityToPointLim]`

`; Максимальное значение порогового значения зоны доступности в секундах`

`maxMaxCost = 1800`

`; Максимальное значение максимального расстояния движения вне УДС в секундах`

`; maxOffroadMaxDistKm = 600`

`; Максимальное значение стоимости километра при движении вне УДС в секундах`

`; maxOffroadCostPerKm = 600`

`; Базовые настройки базы данных`

`[Rugraph/db]`

`; Путь к БД, например filePath=./data/rugraph_{MaxISODate}.sqlite`

`; по шаблону rugraph_{MaxISODate}.sqlite`

`; будет загружен последний файл вида rugraph_2016_08_08T13_39_25.sqlite`

`; {MaxISODate} - обрабатывается аналогично '*'`

`filePath=./data/rugraph/rugraph_{MaxISODate}.sqlite`

`;filePath=/media/psf/SSD/data/rugraph_{MaxISODate}.sqlite`

`; Команды выполняемые при подключении к БД`

`;onConnect="PRAGMA temp_store=MEMORY;PRAGMA synchronous=OFF;PRAGMA`

`journal_mode=OFF;PRAGMA locking_mode=NORMAL; "`

`; onConnect="PRAGMA synchronous=OFF;PRAGMA journal_mode=OFF;PRAGMA`

`locking_mode=EXCLUSIVE; "`

`; onConnect="PRAGMA temp_store=0;PRAGMA PAGE_SIZE=4096;PRAGMA cache_size=-`

`100000;PRAGMA synchronous=OFF;PRAGMA journal_mode=OFF;PRAGMA`

`locking_mode=NORMAL; "`

`onConnect="PRAGMA main.mmap_size=6000000000;PRAGMA temp_store=0;PRAGMA`

`PAGE_SIZE=4096;PRAGMA cache_size=-100000;PRAGMA synchronous=OFF;PRAGMA`

`journal_mode=OFF;PRAGMA locking_mode=NORMAL; "`

`;onConnect="PRAGMA main.mmap_size=6000000000"`

```

; список расширений разделенный запятыми. Относительный путь начинающийся с
точки будет приведен к абсолютному.
; linux mod_spatialite, в релизе для linux не следует использовать ./
поскольку путь прописывается с помощью ldconfig. Для ОС Microsoft Windows
необходимо указывать расширение .dll
;loadExtensions="./mod_spatialite.dll"
loadExtensions="./spatialite-4.3.0a/mod_spatialite"

; Дополнительные настройки базы данных
[RoutePlanner/db]
; Необязательный список имен таблиц с дополнительными данными для ребра,
должны иметь внешний ключ tline_id -> tline.id, разделитель - запятая.
; extraEdgeInfoTables=tline_edge_info,tline_extra_data
extraEdgeInfoTables=
; настройка кеширования ребер для которых есть дополнительная информация
cacheExtraEdgeInfo=false

```

1.11. Отладочная информация серверной части

Сервис прокладки маршрутов 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, сервис прокладки маршрутов создает 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 сервиса прокладки маршрутов 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.