

**Общество с ограниченной ответственностью «Фирма «Ингит»**

## **GISWARE WEB SERVER**

**версия 3**

Программный комплекс веб-сервера с интерактивным программным интерфейсом

## **Описание программного интерфейса**

Версия 1.0.6

## Содержание

<b><u>Программный интерфейс GWS API</u></b>	<b>3</b>
<b><u>Перечень доступных функций</u></b>	<b>3</b>
<b><u>Открытие новой сессии</u></b>	<b>3</b>
<b><u>Получение тайла карты</u></b>	<b>3</b>
<b><u>Получение доступных ограничений дорожной сети</u></b>	<b>3</b>
<b><u>Решение задачи доставки</u></b>	<b>4</b>
<u>Объект GeoPoint</u>	4
<u>Строка Time string</u>	5
<u>Строка Work interval string</u>	5
<u>Объект DeliveryCar</u>	6
<u>Объект DeliveryOrder</u>	8
<u>Объект DeliveryParameters</u>	10
<u>Объект DeliveryZone</u>	13
<u>Объект DeliveryRequestData</u>	14
<u>Объект DeliveryRequest</u>	14
<u>Объект DeliveryResult</u>	14
<u>Объект DeliveryCarResult</u>	15
<u>Объект DeliveryOrderResult</u>	15
<u>Объект DeliveryResultPoint</u>	16
<b><u>Расчет маршрута</u></b>	<b>16</b>
<u>Объект RouteRequest</u>	16
<u>Объект RouteRequestData</u>	17
<u>Объект RouteParameters</u>	17
<u>Объект RouteResponse</u>	18
<u>Объект RouteResponsePoint</u>	18
<b><u>Поиск адреса</u></b>	<b>18</b>
<u>Объект AddressRequest</u>	18
<u>Объект AddressRequestData</u>	19
<u>Объект AddressResponse</u>	19
<u>Объект FoundAddress</u>	20
<b><u>Поиск города</u></b>	<b>20</b>
<u>Объект CityRequest</u>	20
<u>Объект CityRequestData</u>	20
<u>Объект CityResponse</u>	21
<u>Объект FoundCity</u>	21
<b><u>Поиск адреса по строке</u></b>	<b>21</b>
<u>Объект AddressRequest</u>	21
<u>Объект TryFindAddressRequestData</u>	22
<u>Объект AddressResponse</u>	22
<u>Объект FoundAddress</u>	22
<b><u>Поиск ближайшего адреса</u></b>	<b>23</b>
<u>Объект AddressRequest</u>	23
<u>Объект GetNearestAddressRequestData</u>	23
<u>Объект GetNearestAddressResponse</u>	23
<u>Объект FoundAddress</u>	24

## Программный интерфейс GWS API

Программный интерфейс позволяет по HTTP соединению производить прием и передачу данных на сервер GWS в формате JSON.

Все функции (кроме одной) требуют параметры текущей сессии пользователя. Их можно получить с помощью функции открытия новой сессии пользователя.

Часть функций требуют указания ключа доступа. В этом случае сессию пользователя необходимо создавать также с ключом доступа.

### Перечень доступных функций

Описание	Адрес	Формат запроса HTTP
<a href="#">Открытие новой сессии</a>	/api/new_session	GET
<a href="#">Получение тайла карты</a>	/api/tile/	GET
<a href="#">Расчет задачи доставки</a>	/api/delivery.gws	POST
<a href="#">Расчет маршрута</a>	/api/route.gws	POST
<a href="#">Поиск адреса</a>	/api/address.gws	POST
<a href="#">Поиск города</a>	/api/city.gws	POST
<a href="#">Поиск адреса по строке</a>	/api/try_find_address.gws	POST
<a href="#">Поиск ближайшего адреса</a>	/api/nearest_address.gws	POST

### Открытие новой сессии

Перед дальнейшим использованием GWS API необходимо открыть новую сессию пользователя. В ответе возвращаются параметры сессии, которые необходимо передавать при следующих запросах.

- Формат запроса: /api/new\_session[?access\_key='<ключ доступа>']
- Формат ответа: <идентификатор сессии>
- Ключ доступа: не обязателен.

### Получение тайла карты

Функция реализует запрос тайла в формате slippy map tiles (open street map, [http://wiki.openstreetmap.org/wiki/Slippy\\_Map](http://wiki.openstreetmap.org/wiki/Slippy_Map)).

- Формат запроса: /api/tile/{z}/{x}/{y}?sid=<идентификатор сессии>
- Формат ответа: изображение тайла карты.
- Ключ доступа: не нужен.

### Получение доступных ограничений дорожной сети

Функция возвращает возможные значения ограничений, которые есть в загруженной

дорожной сети.

- Формат запроса: /api/road\_variants?sid=<идентификатор сессии>
- Формат ответа: объект со следующими полями

	Тип	Описание
width	Array of uint	Массив возможных ограничений по ширине в дециметрах.
height	Array of uint	Массив возможных ограничений по высоте (тоннели и пр.) в дециметрах.
weight	Array of uint	Массив возможных ограничений по весу (мосты и пр.) в центнерах.
priority	Array of uint	Не рекомендуется к использованию.
cargo	Array of double	Массив возможных ограничений по весу в центнерах (легковой, грузовой до 12т. и пр.).
permit	Array of string	Массив доступных разрешений для транспорта.

- Ключ доступа: не нужен.

### **Решение задачи доставки**

Решение задачи развозки грузов позволяет найти оптимальные маршруты соответствующие заданным ограничениям. Исходными данными являются: список транспорта для развозки и список заказов (объект DeliveryRequest). Ответом является либо объект DeliveryResult, либо массив строк с описанием ошибок.

- Если в описании параметра написано, что он измеряется в произвольных единицах, то для него эти единицы должны быть одинаковыми для всех машин и заказов. Т.е. все параметры объемов и для транспорта и для заказов должны иметь одинаковую размерность, стоимость должна быть в одних и тех же условных единицах и т.д.
- Все идентификаторы являются целыми положительными числами: 1, 2, 3 ...
- Все поля объектов обязательны к заполнению. Если параметр не используется, то его необходимо заполнять «пустым» значением.

### **Объект GeoPoint**

Представляет географическую точку.

	Тип	Описание
lat	Double	Широта
lon	Double	Долгота

## Строка Time string

Является строкой, задающей локальное время в формате ISO 8601. Поддерживаются форматы:

- "YYYY-MM-DDThh:mm" для задания даты и времени. Например, "2015-06-20T22:58".
- "hh:mm" для указания только времени. Например, "22:58".

## Строка Work interval string

Является строкой, задающей набор временных интервалов, заданных через точку с запятой.

Каждый интервал имеет формат: время [дни недели] (срок действия).

Время задается в формате: h0:m1-h2:m2. Например, 9:00-18:00. Если время начала больше времени окончания, то считается, что интервал проходит через полночь (18:00-2:00).

Дни недели задаются перечислением своих номеров(1-понедельник .. 7-воскресенье), через "-" можно указывать диапазоны. Например: [12367] и [1-367] задают один и тот же набор дней недели (понедельник, вторник, среда, суббота, воскресенье). Задание дней недели можно опускать, это будет означать любой день недели.

Формат срока действия: m1.d1-m2.d2, где m1 и d1 - дата (месяц и день) начала, а m2 и d2 - дата окончания действия интервала. Начальная и конечная даты входят в срок действия интервала. Год не задается, а определяется таким образом, чтобы дата попадала в интервал от полугода до текущей даты до полугода после текущей даты. Если дата окончания меньше даты начала, то срок действия проходит через начало года. При одновременном задании дней недели и срока действия интервалы будут применяться к тем дням, которые попадают в заданный срок и соответствуют дням недели. Возможно также упрощенное задание интервалов времени для одного дня: h1:m1-h2:m2 mm:dd. Например, 8:00-11:00 16.11 будет соответствовать 8:00-11:00 (16.11-16.11), т.е. с 8 до 11 утра 16 ноября.

Примеры:

- 8:00-18:00 — каждый день с 8:00 до 18:00.
- 8:00-13:00;14:00-18:00 — каждый день с 8:00 до 18:00 с перерывом между 13:00 и 14:00.
- 8:00-13:00 [1-5];14:00-18:00 [1-4]; 9:00-14:00 [6]; - с понедельника по четверг с 8:00 до 18:00 с перерывом между 13:00 и 14:00: в пятницу только с 8:00 до 13:00, в субботу с 9:00 до 14:00.
- 10:00-18:00 [135] (21.10-21.11);11:00-19:00 [246] (21.10-21.11);12:00-18:00 [1-5](22.11-21.01) — с 21 октября по 21 ноября включительно - по понедельникам, средам и пятницам с 10:00 до 18:00, по вторникам, четвергам и субботам - с 11:00 до 19:00, с 22 ноября по 21 января в рабочие дни с 12:00 до 18:00. Даты могут переходить через начало года.
- 23:00-5:00[1-6] — время может переходить через полночь, при этом дата относится к началу интервала. Здесь - с 23:00 до 5 утра с понедельника по субботу, т.е. каждый ночь кроме ночи с воскресенья на понедельник.

## Объект DeliveryCar

Представляет собой транспорт для развозки грузов.

	Тип	Единицы измерения	Описание	«Пустое» значение
id	UInt		Идентификатор транспорта	
initialPosition	GeoPoint		Точка старта (гараж)	
maxWeight	UInt	кг.	Максимально перевозимый вес	0
maxVolume	UInt	пр. ед.	Максимальный объем 1	0
maxVolume2	UInt	пр. ед.	Максимальный объем 2	0
maxCost	UInt	пр. ед.	Максимально перевозимая стоимость.	0
speedCoefficient	Double		Коэффициент скорости	1
selfWeight	UInt	кг.	Вес транспорта	0
maxRouteLength	UInt	м.	Максимальная длина маршрута	0
maxRouteTime	UInt	с.	Максимальное время маршрута	0
fuelConsumptionPerKm	Double	пр. ед.	Потребление топлива	0
workStartTime	Time string		Время начала работы	""
workFinishTime	Time string		Время конца работы	""
lunchStartTime	Time string		Время начала обеда	""
lunchFinishTime	Time string		Время конца обеда	""
maxOrdersCount	UInt		Максимальное количество заказов	0
permit	String		Лицензия на дорожные ограничения	""
carType	UInt		Тип транспорта	0
cargoType	UInt	центнеры	Весовая категория в терминах дорожных знаков	0
width	UInt	дециметры	Ширина транспорта	0
height	UInt	дециметры	Высота транспорта	0
allowedLoadZones	Array of UInt		Зоны погрузки	[]

allowedUnloadZones	Array of UInt		Зоны разгрузки	[]
workTime	Work interval string		Время работы при многодневных маршрутах	""
useTollRoads	Int		Использовать платные дороги	-1

Подробнее:

- `id` — идентификатор транспорта.
- `initialPosition` — стартовые координаты транспорта, гараж.
- `maxWeight` — максимальный суммарный перевозимый вес в кг.
- `maxVolume` — максимально допустимый суммарный объем всех перевозимых заказов в произвольных единицах. Для двухобъемных машин: максимальный размер первого объема. Единицы измерения должны быть одними и теми же для всех машин и заказов.
- `maxVolume2` — для двухобъемных машин: максимальный размер второго объема. Единицы измерения должны быть одними и теми же для всех машин и заказов.
- `maxCost` — максимальная суммарная стоимость перевозимых товаров в произвольных единицах.
- `speedCoefficient` — коэффициент скорости (от 0.001 до 1000). На этот коэффициент умножается скорость машины на каждом участке маршрута. Т.о., машина с коэффициентом 1.5 пройдет тот же маршрут в полтора раза быстрее, чем машина с коэффициентом 1. Можно устанавливать этот коэффициент в зависимости от дня недели, от времени, от личных качеств и стиля вождения водителя для приближения к действительным значениям.
- `selfWeight` — собственный вес машины в килограммах. Этот вес используется для оптимизации маршрутов по параметру "длина \* вес", а также для определения участков дорог, которые нельзя использовать для прокладки маршрута для этой машины ввиду ограничений на вес транспортных средств.
- `maxRouteLength` — максимальная длина маршрута для машины в метрах. Если задано значение 0 (по умолчанию), то длина маршрута для машины не ограничивается.
- `maxRouteTime` — максимальная продолжительность маршрута для машины в секундах. Если задано значение 0, то продолжительность маршрута для машины не ограничивается.
- `fuelConsumptionPerKm` — потребление топлива. Используется при оптимизации маршрутов по расходу топлива. Наиболее длинные маршруты при прочих равных условиях будут назначаться машинам с наименьшим расходом топлива.
- `workStartTime` — время начала работы машины. Если задано «пустое» значение, то это означает, что машина может начать работу в любое время. Если присваивается ненулевое значение, то маршрут составляется таким образом, что машина начинает работу не раньше указанного времени. Все временные ограничения (для машин и для заказов) должны указываться одинаковым образом — или временем с датой, или временем без даты. В случае, если время начала работы больше, чем время окончания работы, и дата не указывается, считается, что работа машины заканчивается на следующий день.
- `workFinishTime` — время окончания работы машины. Если задано «пустое» значение, то это означает, что машина может закончить работу в любое время. Если присваивается ненулевое значение, то маршрут составляется таким образом, что

машина заканчивает работу не позже указанного времени. Все временные ограничения (для машин и для заказов) должны указываться одинаковым образом — или временем с датой, или временем без даты. В случае, если время начала работы больше, чем время окончания работы, и дата не указывается, считается, что работа машины заканчивается на следующий день.

- `lunchStartTime` — Задаёт время начала обеда. Если начало и конец обеда заданы, то при прокладке машины будет учитываться перерыв на обед. Когда при переезде из одной точки А маршрута до другой точки Б наступает время обеда, за точкой А вставляется специальная точка, которая имеет координаты точки А и время прибытия, равное времени выезда из точки А. При этом время пути из А в Б удлинится на время обеда. Это означает, что в промежутке между А и Б водитель может остановиться по пути, где ему удобнее, и сделать перерыв на обед. Время обеда может сместиться на более позднее время, если его начало придется на время погрузки/разгрузки в точке А. Обед не учитывается, если его конец наступает раньше, чем время начала работы, если он приходится на перерыв между рейсами или если он приходится на путь из последней точки до гаража. Хотя время обеда задается как полноценная дата-время, учитывается только время. Кроме того, в маршруте обед учитывается только 1 раз. Для многодневных маршрутов он неприменим.
- `lunchFinishTime` — Задаёт время окончания обеда.
- `maxOrdersCount` — максимальное количество перевозимых в одном рейсе заказов. Маршрут составляется таким образом, чтобы количество заказов в одном рейсе не превышало указанное. Значение 0 указывает, что количество заказов не ограничивается.
- `license` – имена лицензий, которые применяется к машине. Маршрут для машины прокладывается по тем участкам дорог, которые доступны при указанной лицензии.
- `carType` — тип машины. В заказе можно перечислить типы машин, в которых этот заказ может перевозиться, тогда заказ может распределиться только в те машины, которые имеют тип, включенный в этот список. Значение 0 задает универсальный тип машины.
- `cargoType` — вес машины в терминах дорожных знаков. Измеряется в центнерах. Число от 0 до 255. Значение 0 задает, что машина не попадает ни под одни ограничения.
- `width` — ширина машины. Измеряется в дециметрах.
- `height` — высота машины. Измеряется в дециметрах.
- `allowedLoadZones` — массив идентификаторов зон, где разрешено загружать заказ в машину.
- `allowedUnloadZones` — массив идентификаторов зон, где разрешено разгружать заказ из машины.
- `workTime` — набор временных интервалов, когда машине разрешено работать.
- `useTollRoads` – использовать платные дороги. Доступные значения: 0 — нет, 1 — да, -1 — использовать общее значение `useTollRoads` из параметров доставки.

## Объект `DeliveryOrder`

Представляет заказ, который необходимо доставить.

	Тип	Единицы измерения	Описание	«Пустое» значение
<code>id</code>	<code>UInt</code>		Идентификатор заказа	



loadPosition	GeoPoint		Точка загрузки	
handOverPosition	GeoPoint		Точки выгрузки (доставки) заказа	
loadStartTime	Time string		Загрузить заказ после	""
loadFinishTime	Time string		Загрузить заказ до	""
handOverStartTime	Time string		Выгрузить заказ после	""
handOverFinishTime	Time string		Выгрузить заказ до	""
priority	UInt		Приоритет	0
carTypes	Array of UInt		Типы транспорта для перевозки	[]
weight	UInt	кг.	Вес	0
volume	UInt	пр. ед.	Объем	0
cost	UInt	пр. ед.	Стоимость	0
carVolumeUsage	UInt		Используемый объем	0
loadDuration	UInt	с.	Продолжительность загрузки	0
handOverDuration	UInt	с.	Продолжительность выгрузки	0
maxTransportationTime	UInt	с.	Максимальное время транспортировки	0
loadTime	Work interval string		Время погрузки при многодневных маршрутах.	""
handOverTime	Work interval string		Время выгрузки при многодневных маршрутах.	""
loadPointIndexInRoute	Int		Индекс загрузки	-1
handOverPointIndexInRoute	Int		Индекс разгрузки	-1
attachedCarId	UInt		Прикрепленный транспорт	0

Подробнее:

- id — идентификатор заказа.
- loadPosition — координаты загрузки заказа.
- handOverPosition — координаты разгрузки заказа.
- loadStartTime — время, раньше которого заказ не готов к погрузке.
- loadFinishTime — время, до которого необходимо загрузить заказ.
- handOverStartTime — время, раньше которого заказ не разгружать.
- handOverFinishTime — время, до которого необходимо разгрузить заказ.

- `priority` — приоритет заказа. Изначально раскладываются заказы с наибольшим приоритетом. Не заданный приоритет равносильен 0. Высокий приоритет не означает, что доставка этого заказа осуществится раньше других, а только то, что вероятность того, что при прочих равных условиях, если машин не будет хватать на все заказы, распределится именно он, а не заказ с более низким приоритетом. Для того, чтобы доставка заказа осуществилась раньше других, используются свойства, определяющие время доставки заказа.
- `carTypes` — типы машин, которыми возможна доставка данного заказа.
- `weight` — вес заказа в кг.
- `volume` — объем заказа в произвольных единицах. Единицы измерения объёма должны быть одинаковым у всех заказов и машин. Влияет на то, сколько и каких заказов может перевозить машина одновременно, если у нее установлено свойство `maxVolume`.
- `carVolumeUsage` — в какой объем машины можно загружать заказ (для двухобъемных машин). Возможные значения:
  - 0 — подходит любой объем.
  - 1 — только в первый объем.
  - 2 — только во второй объем.
- `cost` — стоимость заказа. Влияет на то, сколько и каких заказов может перевозить машина за рейс, если у нее установлено свойство `cost`.
- `loadDuration` — продолжительность загрузки заказа в секундах.
- `handOverDuration` — продолжительность разгрузки заказа в секундах.
- `maxTransportationTime` — максимальное время заказа в пути в секундах.
- `loadTime` – время работы точки загрузки заказа при многодневном режиме расчета маршрутов.
- `handOverTime` – время работы точки разгрузки заказа при многодневном режиме расчета маршрутов.
- `loadPointIndexInRoute` — индекс точки загрузки в маршруте. Используется при восстановлении маршрута (`distributionMethod = "preDistributed"`).
- `handOverPointIndexInRoute` — индекс точки разгрузки в маршруте. Используется при восстановлении маршрута (`distributionMethod = "preDistributed"`).
- `attachedCarId` — идентификатор машины, к которой прикреплен заказ. Используется при восстановлении маршрута (`distributionMethod = "preDistributed"`).

## Объект `DeliveryParameters`

Представляют общие параметры при решении задачи доставки.

	Тип	Единицы измерения	Описание	«Пустое» значение
<code>optimizationMode</code>	String		Метод оптимизации маршрута	
<code>distributionMethod</code>	String		Метод распределения заказов	
<code>multiTripTimeInterval</code>	UInt	с.	Межрейсовый интервал	0
<code>countPathFromGarage</code>	Bool		Учитывать путь от гаража	false
<code>countPathToGarage</code>	Bool		Учитывать путь до гаража	false

allowGarageInMultiTrip	Bool		Учитывать заезд в гараж между рейсами	false
visitEachPointOnlyOnce	Bool		Только одноразовое посещение точки за рейс	false
allowMultiTripRoutes	Bool		Разрешить многорейсовые маршруты	false
useStackUnloading	Bool		Разгрузка в порядке обратном загрузке	false
isGroupDelivery	Bool		Доставка группами	false
additionalTimeInPoint	Int	м.	Дополнительное время	0
additionalDistanceInPoint	Int	с.	Дополнительный путь	0
includeAdditionsInRoute	Bool		Учитывать дополнительное время и путь в маршруте	false
multiDayRoutes	Bool		Многодневные маршруты.	false
useTollRoads	Bool		Использовать платные дороги	false

Подробнее:

- optimizationMode — критерий оптимизации маршрута. Возможные значения:
  - "optimizeTime" — маршруты рассчитываются таким образом, чтобы суммарное время всех маршрутов было минимальным.
  - "optimizeLength" — суммарная длина всех маршрутов минимальна.
  - "optimizeLengthWeight" — сумма произведений длины каждого участка маршрута на вес перевозимого по этому участку груза с учетом веса самой машины, минимальна, т.е., минимизация "тонно-километров".
  - "optimizeLengthConsumption" — по возможности самые длинные маршруты будут назначаться машинам с наименьшим расходом топлива.
- distributionMethod — метод распределения заказов. Возможные варианты:
  - "farthestFirst" — начинать с отдаленных.
  - "concurrentOrders" — выбирать попутные заказы.
  - "farthestDirections" — определять дальние направления.
  - "favorableCombinations" — искать самые выгодные совмещения.
  - "preDistributed" — восстановить маршрут по номерам точек. Для каждого заказа необходимо указать параметры loadPointIndexInRoute, handOverPointIndexInRoute, attachedCarId.
- multiTripTimeInterval — интервал между рейсами в секундах.
- countPathFromGarage — указывает, учитывать или нет в маршруте путь от гаража до точки погрузки первого заказа. Если да, то длина и время пути добавляется к общей длине и времени маршрута для машин. В многорейсовых маршрутах путь до гаража и от гаража между рейсами определяется свойством allowGarageInMultiTrip, а countPathFromGarage влияет только на первый рейс.
- countPathToGarage — указывает, учитывать или нет в маршруте путь от точки разгрузки последнего заказа до гаража. Если да, то длина и время пути добавляется к

общей длине и времени маршрута для машин. В многорейсовых маршрутах путь до гаража и от гаража между рейсами определяется свойством `allowGarageInMultiTrip`, а `countPathToGarage` влияет только на последний рейс.

- `allowGarageInMultiTrip` — указывает, учитывать или нет в маршруте путь до гаража после очередного рейса и путь из гаража до первой точки загрузки в следующем рейсе. Если да, то длина и время пути добавляется к общей длине и времени маршрута для машин. Время интервала между рейсами добавляется в любом случае.
- `visitEachPointOnlyOnce` — посещать каждую точку только 1 раз. Маршрут может составиться так, что машина несколько раз заезжает в одну точку (например, загрузить на складе сначала одну порцию заказов, потом их развезти, заехать снова на склад и загрузить вторую порцию или отвезти заказ в одну точку сначала одному клиенту, затем посетить другие точки и вернуться снова в ту же точку, чтобы разгрузить заказ второго клиента, что может быть обусловлено временными ограничениями на доставку). Если заезд в одну точку нежелателен, то свойство устанавливается в `TRUE`, и маршруты составляются таким образом, чтобы машины заезжали в каждую точку только единожды. При этом, например, машина будет загружать все заказы на складе только один раз и не будет больше на него возвращаться за следующими заказами, которые она не смогла загрузить сразу из-за ограниченной грузоподъемности или объема. Для того, чтобы после сделанного рейса машины вновь могла загрузиться на складе, используется свойство `allowMultiTripRoutes`.
- `allowMultiTripRoutes` — использовать многорейсовые маршруты. Из-за наложенных ограничений (заезд в каждую точку только 1 раз, максимальная стоимость заказов, максимальное количество заказов на 1 рейс) может получиться так, что машины могут загрузить и развезти только часть заказов, и при этом у машин еще остается много рабочего времени. Многорейсовые маршруты позволяют составить маршрут таким образом, что после сделанного рейса машина ждет какое-то время (`multiTripTimeInterval`), и далее едет развезти следующие заказы. Между рейсами машина также может заезжать в гараж, если установлено свойство `garagebetween`. Ограничение на количество и стоимость заказов для машины накладывается на каждый из рассчитанных рейсов. Ограничение на продолжительность и длину маршрута действует по всему маршруту.
- `useStackUnloading` — разгружать в порядке обратном загрузке. Допустим, если машина развозит мебель, то разгружать из неё удобнее заказ, который был погружен последним. В случае, когда все заказы развозятся с одного склада, это означает только то, что мебель загружается в порядке, обратном тому, в каком машина развозит её по заказчикам. В случае, когда все заказы развозятся с одного склада или привозятся на один склад, установка свойства `useStackUnloading` в `TRUE` значительно уменьшает число комбинаций для подсчета и тем самым значительно ускоряет вычисления. В случае же, когда заказы имеют разные места погрузок и разгрузок, если свойство установлено в `FALSE`, маршрут может быть составлен более эффективно. Конечно, при этом характер заказов должен позволять разгружать заказы из машины в произвольном порядке (например, курьер, который развозит письма от разных отправителей разным адресатам).
- `isGroupDelivery` — решать задачу как групповую развозку.
- `additionalTimeInPoint` — дополнительное время при заезде в точку. Дополнительное время в точке учитывается только при заезде в эту точку. Если до этой точки машина загружала или разгружала другие заказы в этой же точке, то дополнительное время не учитывается. Если несколько заказов загружаются один за другим в одной и той же точке, то дополнительное время будет учитываться только для первого из них. Если свойство `includeAdditionsInRoute` (включать дополнительные величины в маршрут) установлено в `true`, то время будет добавляться к результирующему времени

маршрута, иначе (false) это время будет просто влиять на эффективность маршрута — по возможности маршрут будет составляться так, чтобы заезжать в каждую точку минимально возможное количество раз, но в реальное время маршрута оно добавляться не будет.

- `additionalDistanceInPoint` — дополнительный путь при заезде в точку. Дополнительный путь в точке учитывается только при заезде в эту точку. Если до этой точки машина загружала или разгружала другие заказы в этой же точке, то дополнительный путь не учитывается. Если несколько заказов загружаются один за другим в одной и той же точке, дополнительный путь учитывается только для первого из них. Если свойство `includeAdditionsInRoute` (включать дополнительные величины в маршрут) установлено в true, то путь будет добавляться к результирующему пути маршрута, иначе (false) этот путь будет просто влиять на эффективность маршрута — по возможности маршрут будет составляться так, чтобы заезжать в каждую точку минимально возможное количество раз, но в результирующую длину маршрута он добавляться не будет.
- `includeAdditionsInRoute` — добавлять дополнительное время и дополнительный путь при заезде в точку в расчетное время и длину маршрута. Установка дополнительного времени и пути маршрута позволяет минимизировать количество заездов в точки, в которых находится несколько заказов, поскольку становится выгоднее заезжать в каждую точку минимальное количество раз. При этом, с одной стороны, это может служить более гибкой альтернативой свойству `visitEachPointOnlyOnce` (посещение точки только единожды), а с другой стороны, можно учитывать реальные условия в точках. Если дополнительные величины отражают реальные условия подъездов к точкам, следует установить свойство в true — в этом случае дополнительное время и путь включаются в маршрут. Если дополнительные величины используются только для уменьшения заездов в точки, то следует установить свойство в false.
- `multiDayRoutes` — использовать режим расчета многодневных маршрутов. При этом режиме свойства машин `workStartTime`, `workFinishTime`, `lunchStartTime`, `lunchFinishTime` игнорируются и используется свойство `workTime`. Аналогично, для заказов игнорируются свойства `loadStartTime`, `loadFinishTime`, `handOverStartTime`, `handOverFinishTime` и используются `loadTime`, `handOverTime`. Машина может передвигаться только в заданные интервалы. При наступлении конца интервала на маршруте создается точка остановки (тип точки 7), и движение продолжается с начала следующего интервала. Если машина приезжает к точке загрузки или разгрузки не в рабочий интервал, она ожидает начала следующего интервала. Если машина приезжает во время доступности, она загружается или разгружается, независимо от установленной продолжительности загрузки/разгрузки. Маршруты составляются таким образом, чтобы по возможности время ожидания было минимальным. Маршрут, как и при однодневном варианте, может ограничиваться максимальной продолжительностью (`maxRouteTime`), а для каждого заказа действует максимальное время транспортировки (`maxTransportationTime`).
- `useTollRoads` — использовать платные дороги. Это общее свойство для всего транспорта. Для каждого конкретного его можно переопределить.

## Объект `DeliveryZone`

Представляет зону ограничения на карте.

	Тип	Описание
<code>id</code>	<code>UInt</code>	Идентификатор зоны
<code>metrics</code>	<code>String</code>	Метрика зоны

Подробнее:

- `id` — идентификатор зоны.
- `metrics` — метрика зоны в формате "А сегмент1; ссегмент2; ...". Строка сегмента: "Е или I долгота1 широта1, долгота2 широта2, ...", где Е означает внешний сегмент, а I — внутренний. Пример, "А Е 30.212049 59.865185, 30.244493 59.890204, 30.292512 59.881072, 30.297512 59.852470, 30.273772 59.835774, 30.224710 59.851278, 30.212049 59.865185;"

## Объект `DeliveryRequestData`

Представляет исходные данные для решения задачи доставки.

	Тип	Описание
<code>parameters</code>	<code>DeliveryParameters</code>	Параметры расчета
<code>cars</code>	Array of <code>DeliveryCar</code>	Доступный транспорт
<code>orders</code>	Array of <code>DeliveryOrder</code>	Массив заказов
<code>zones</code>	Array of <code>DeliveryZone</code>	Массив зон

## Объект `DeliveryRequest`

Представляет собой запрос на решение задачи доставки.

	Тип	Описание	«Пустое» значение
<code>sessionId</code>	<code>UInt</code>	Идентификатор пользовательской сессии	
<code>accessKey</code>	<code>String</code>	Ключ доступа к серверу	""
<code>deliveryData</code>	<code>DeliveryRequestData</code>	Данные для расчета	

## Объект `DeliveryResult`

Представляет собой решение задачи.

	Тип	Описание
<code>cars</code>	Array of <code>DeliveryCarResult</code>	Результаты по транспорту
<code>orders</code>	Array of <code>DeliveryOrderResult</code>	Результаты по заказам

Подробнее:

- `cars` — посчитанные результаты для всего транспорта.
- `orders` — посчитанные результаты для всех заказов.

## Объект DeliveryCarResult

Представляет результат раскладки для машины.

	Тип	Единицы измерения	Описание
carId	UInt		Идентификатор транспорта
totalRouteLength	UInt	м.	Ключ доступа к серверу
totalRouteTime	UInt	с.	Данные для расчета
totalOrdersCount	UInt		Количество заказов для развозки
routeStartTime	Time string		Время начала маршрута
routeEndTime	Time string		Дата конца маршрута
errorPointsCount	UInt		Количество нарушений
route	Array of DeliveryResultPoint		Маршрут

Подробнее:

- carId — идентификатор транспорта.
- totalRouteLength — общая длина маршрута в метрах.
- totalRouteTime — общее время маршрута в секундах.
- totalOrdersCount — количество заказов, распределенных в данную машину.
- routeStartTime — дата и время начала маршрута.
- routeEndTime — дата и время конца маршрута.
- errorPointsCount — количество точек, в которых нарушаются заданные ограничения.
- route — массив точек маршрута.

## Объект DeliveryOrderResult

Представляет результат раскладки для заказа

	Тип	Описание
orderId	UInt	Идентификатор заказа
carId	UInt	Идентификатор транспорта
loadTime	Time string	Время погрузки
handOverTime	Time string	Время разгрузки

Подробнее:

- orderId — идентификатор заказа.
- carId — идентификатор машины, в которую заказ распределен. Значение 0 указывает, что заказ не распределен.
- loadTime — дата и время загрузки заказа в машину.
- handOverTime — дата и время разгрузки заказа.

## Объект DeliveryResultPoint

Представляет точку маршрута транспорта.

	Тип	Описание
coordinates	GeoPoint	Координаты точки
description	String	Описание точки
arrivalTime	Time string	Время проезда точки
type	UInt	Тип точки
Следующие поля есть, только когда тип точки не равен 10		
orderId	UInt	Идентификатор заказа.
volumeUsed	UInt	В каком объеме транспорта перевозится заказ
errorMessage	String	Сообщение о нарушении ограничений

Подробнее:

- coordinates — географические координаты.
- description — имя точки. Обычно имя улицы.
- arrivalTime — дата и время прибытия машины на точку.
- type — тип точки. Число. Возможные значения:
  - 0 — стартовая точка маршрута.
  - 1 — в этой точке происходит загрузка товара.
  - 2 — в этой точке происходит разгрузка товара.
  - 3 — конечная точка маршрута.
  - 4 — конечная точка рейса.
  - 5 — не распределенная точка.
  - 6 — точка обеда.
  - 7 — точка остановки в многодневном маршруте.
  - 10 — обычная точка маршрута.

Следующие поля есть только при условии, что точка не является обычной точкой маршрута. Т.е. когда тип точки не равен 10.

- orderId — идентификатор выгружаемого/загружаемого заказа, если тип точки равен 1 или 2; 0 — во всех остальных случаях.
- volumeUsed — объем машины в который распределился заказ.
- errorMessage — сообщение о нарушениях ограничений в данной точке.

## ***Расчет маршрута***

Функция реализует расчет маршрута по заданным параметрам.

## Объект RouteRequest

Представляет собой запрос для расчета маршрута на сервере.



	Тип	Описание	«Пустое» значение
sessionId	UInt	Идентификатор пользовательской сессии	
accessKey	String	Ключ доступа к серверу	""
routeData	RouteRequestData	Данные для расчета	

## Объект RouteRequestData

Представляет исходные данные для расчета маршрута.

	Тип	Описание
parameters	RouteParameters	Параметры расчета
restrictPoints	Array of <a href="#">GeoPoint</a>	Массив точек закрытых для проезда
points	Array of <a href="#">GeoPoint</a>	Массив точек для прокладки маршрута

## Объект RouteParameters

Параметры для расчета маршрута. Рекомендуется использовать значения из запроса [«Получение доступных ограничений»](#)

	Тип	Единицы измерения	Описание	«Пустое» значение
width	UInt	дециметры	Ограничение по ширине	0
height	UInt	дециметры	Ограничение по высоте	0
weight	UInt	центнеры	Ограничение по весу (мосты и пр.)	0
cargo	UInt	центнеры	Ограничение по весу в центнерах (грузовой до 12т. и пр.)	0
optimizeTimeDistance	Double	0.0 — 1.0	Оптимизировать время (0.0) или расстояние (1.0)	0
optimizePointOrder	Bool		Оптимизировать порядок прохождения точек	false
date	String		Дата старта маршрута в формате dd/mm/yyyy	""
time	String		Время старта маршрута в формате hh:mm:ss	""
useTollRoads	Bool		Разрешить проезд по платным дорогам	false

permit	String	Дорожное разрешение	""
--------	--------	---------------------	----

## Объект RouteResponse

Представляет собой ответ в случае успешного расчета маршрута на сервере.

	Тип	Описание
totalTime	UInt	Продолжительность маршрута в секундах.
totalLength	UInt	Длина маршрута в метрах.
points	Array of RouteResponsePoint	Точки маршрута.

## Объект RouteResponsePoint

Представляет собой рассчитанную точку маршрута. Если точка является исходной точкой для поиска маршрута, то у нее будут заполнены поля originalIndex, lengthToNextPoint, timeToNextPoint.

	Тип	Описание
name	String	Наименование дороги.
lat	Double	Географическая широта.
lon	Double	Географическая долгота.
speed	UInt	Ожидаемая скорость в точке (км/ч).
originalIndex	Int	Исходный индекс из массива точек для поиска маршрута. Если нет, то тогда -1.
lengthToNextPoint	UInt	Дистанция до следующей точки маршрута в метрах. Если точка не исходная или последняя, то тогда 0.
timeToNextPoint	UInt	Время до следующей точки маршрута в секундах. Если точка не исходная или последняя, то тогда 0.

## Поиск адреса

Функция реализует поиск адреса по заданным параметрам.

## Объект AddressRequest

Представляет собой запрос поиска адреса.

	Тип	Описание	«Пустое» значение
--	-----	----------	-------------------

sessionId	UInt	Идентификатор пользовательской сессии	
accessKey	String	Ключ доступа к серверу	""
addressData	AddressRequestData	Данные для поиска	

### Объект AddressRequestData

Представляет исходные данные для поиска адреса.

	Тип	Описание
centerLat	Double	Широта точки сортировки результатов
centerLon	Double	Долгота точки сортировки результатов
streetName	String	Шаблон поиска улицы (например, "лен*") или улица целиком
streetType	String	Тип улицы
toponim	String	Топоним
city	String	Населенный пункт
region	String	Район
ao	String	Автономный округ
province	String	Область
country	String	Страна
house	String	Номер дома

### Объект AddressResponse

Представляет собой ответ в случае успешного поиска адреса на сервере.

	Тип	Описание
firstLat	Double	Географическая широта первого найденного результата
firstLon	Double	Географическая долгота первого найденного результата
foundAddresses	Array of FoundAddress	Массив найденных адресов

## Объект FoundAddress

Представляет собой найденный адрес.

	Тип	Описание
streetName	String	Улица
streetType	String	Тип улицы
toponim	String	Топоним
city	String	Населенный пункт
region	String	Район
ao	String	Автономный округ
province	String	Область
country	String	Страна
house	String	Номер дома

## Поиск города

Функция реализует поиск города по заданным параметрам.

## Объект CityRequest

Представляет собой запрос поиска города.

	Тип	Описание	«Пустое» значение
sessionId	UInt	Идентификатор пользовательской сессии	
accessKey	String	Ключ доступа к серверу	""
cityData	CityRequestData	Данные для поиска	

## Объект CityRequestData

Представляет исходные данные для поиска города.

	Тип	Описание
centerLat	Double	Широта точки сортировки результатов
centerLon	Double	Долгота точки сортировки результатов

city	String	Шаблон поиска (например, "мос*") или населенный пункт целиком
region	String	Район
ao	String	Автономный округ
province	String	Область
country	String	Страна

### Объект CityResponse

Представляет собой ответ в случае успешного поиска города на сервере.

	Тип	Описание
firstLat	Double	Географическая широта первого найденного результата
firstLon	Double	Географическая долгота первого найденного результата
foundCities	Array of FoundCity	Массив найденных городов

### Объект FoundCity

Представляет собой найденный город.

	Тип	Описание
city	String	Населенный пункт
region	String	Район
ao	String	Автономный округ
province	String	Область
country	String	Страна

### ***Поиск адреса по строке***

Функция реализует попытку поиска адреса по заданной строке в свободном формате.

### Объект AddressRequest

Представляет собой запрос поиска адреса.

	Тип	Описание	«Пустое» значение
sessionId	UInt	Идентификатор пользовательской	

		сессии	
accessKey	String	Ключ доступа к серверу	""
addressData	TryFindAddressRequestData	Данные для поиска	

### Объект TryFindAddressRequestData

Представляет исходные данные для поиска адреса. Если под строку поиска подходит несколько вариантов, то выбирается ближайший к точке сортировки.

	Тип	Описание
centerLat	Double	Широта точки сортировки результатов
centerLon	Double	Долгота точки сортировки результатов
query	String	Строка поиска адреса целиком. Например, "ленинский 100 Санкт-Петербург"

### Объект AddressResponse

Представляет собой ответ в случае успешного поиска адреса на сервере.

	Тип	Описание
firstLat	Double	Географическая широта первого найденного результата
firstLon	Double	Географическая долгота первого найденного результата
foundAddresses	Array of FoundAddress	Массив найденных адресов. В данном случае массив содержит не более одного элемента

### Объект FoundAddress

Представляет собой найденный адрес.

	Тип	Описание
streetName	String	Улица
streetType	String	Тип улицы
toponim	String	Топоним
city	String	Населенный пункт

region	String	Район
ao	String	Автономный округ
province	String	Область
country	String	Страна
house	String	Номер дома

### **Поиск ближайшего адреса**

Функция реализует поиск ближайшего адреса по заданной точке.

#### **Объект AddressRequest**

Представляет собой запрос поиска адреса.

	Тип	Описание	«Пустое» значение
sessionId	UInt	Идентификатор пользовательской сессии	
accessKey	String	Ключ доступа к серверу	""
addressData	GetNearestAddressRequestData	Данные для поиска	

#### **Объект GetNearestAddressRequestData**

Представляет исходные данные для поиска адреса. Если под строку поиска подходит несколько вариантов, то выбирается ближайший к точке сортировки.

	Тип	Описание
lat	Double	Широта точки поиска ближайшего адреса
lon	Double	Долгота точки поиска ближайшего адреса

#### **Объект GetNearestAddressResponse**

Представляет собой ответ в случае успешного поиска адреса на сервере.

	Тип	Описание
firstLat	Double	Географическая широта точки, относительно которой велся поиск
firstLon	Double	Географическая долгота точки, относительно

		которой велся поиск
foundAddresses	Array of FoundAddress	Массив найденных адресов. В данном случае массив содержит не более одного элемента

### Объект FoundAddress

Представляет собой найденный адрес.

	Тип	Описание
streetName	String	Улица
streetType	String	Тип улицы
toponim	String	Топоним
city	String	Населенный пункт
region	String	Район
ao	String	Автономный округ
province	String	Область
country	String	Страна
house	String	Номер дома