



OptimoRoute

**The OptimoRoute Web Service
Application Programming Interface
Specification**

v1.15

Table of Contents

1. Introduction	4
1.1. Our Web Service API is HTTP based	4
1.2. We use JSON for structured data exchange	4
1.3. SSL is required	4
1.4. Limited number of concurrent requests	4
1.5. General error codes	4
1.6. Allow for additional fields in the response messages	4
2. Authentication	5
3. The REST API	6
3.1. Create Order	6
<i>Description</i>	6
<i>Input format</i>	6
<i>Request example</i>	10
<i>Return values</i>	10
<i>Error and warning codes</i>	11
3.2. Delete Order	13
<i>Description</i>	13
<i>Input format</i>	13
<i>Request example</i>	13
<i>Return values</i>	13
<i>Error codes</i>	13
3.3. Delete All Orders	14
<i>Description</i>	14
<i>Input format</i>	14
<i>Request example</i>	14
<i>Return values</i>	14
<i>Error codes</i>	14
3.4. Get Routes	15
<i>Description</i>	15
<i>Input format</i>	15
<i>Request example</i>	15
<i>Return values</i>	15
<i>Error codes</i>	17
3.5. Get Scheduling Information	18
<i>Description</i>	18
<i>Input format</i>	18
<i>Request example</i>	18
<i>Return values</i>	18
<i>Error codes</i>	19
3.6. Start Planning	20
<i>Description</i>	20

<i>Input format</i>	20
<i>Request example</i>	22
<i>Return values</i>	22
<i>Error codes</i>	22
3.7. Stop Planning.....	24
<i>Description</i>	24
<i>Input format</i>	24
<i>Request example</i>	24
<i>Return values</i>	24
<i>Error codes</i>	24
3.8. Get Planning Status	25
<i>Description</i>	25
<i>Input format</i>	25
<i>Request example</i>	25
<i>Return values</i>	25
<i>Error codes</i>	25
3.9. Update Driver Parameters	26
<i>Description</i>	26
<i>Input format</i>	26
<i>Request example</i>	27
<i>Return values</i>	27
<i>Error codes</i>	27
3.10. Get Mobile Events	28
<i>Description</i>	28
<i>Input format</i>	28
<i>Request example</i>	28
<i>Return values</i>	29

1. Introduction

This document is the official reference for the OptimoRoute Web Service Application Programming Interface (WS API).

1.1. Our Web Service API is HTTP based

The methods to retrieve data from our Web Service (WS) API require an HTTP GET request method.

The methods that submit, change, or destroy data require a POST request method.

The API methods will return an error if you do not make your request with the correct HTTP method.

1.2. We use JSON for structured data exchange

Our API uses the JSON (JavaScript Object Notation) format.

More information about JSON and how it works can be found here: <http://json.org/> and here: <http://en.wikipedia.org/wiki/JSON>. There are many readily available libraries to convert to and from the JSON format both for popular and for more esoteric programming languages.

1.3. SSL is required

Using SSL (https) is required to avoid passing both the authentication key and potentially confidential data in clear text over the web.

1.4. Limited number of concurrent requests

The maximum number of concurrent web service API requests for one account or for one IP address is limited to 5.

1.5. General error codes

The following error codes are applicable to all API operations:

- AUTH_KEY_MISSING – the authentication key is missing
- AUTH_KEY_UNKNOWN – a wrong authentication key was supplied
- MALFORMED_REQUEST – something is wrong with the input
- ERR_MISSING_MAND_FIELD – one of the mandatory fields is missing
- ERR_INVALID_PARAM_FORMAT – one of the specified fields is not in a valid format
- ERR_TOO_MANY_CONNECTIONS – there are too many concurrent requests
- ERR_INTERNAL – an internal server error occur

There are also error codes specific that are described in the *Return values* section of each method.

1.6. Allow for additional fields in the response messages

Additional fields might be added to the JSON response messages in the future, so your client code should ignore additional fields.

2. Authentication

The OptimoRoute Web Service API authentication key parameter is required with all API requests, in addition to all the standard parameters.

To enable the API and generate the API key please log into the OptimoRoute web application. In the **Administration** section select **Settings** -> **WS API** and enable the API. The API key will be generated for you.

The Web Service API is also available during the free trial period.

3. The REST API

3.1. Create Order

Description

Creates a new order in the system or updates an existing order.

URL: https://api.optimoroute.com/v1/create_order

HTTP Method: POST

Input format

*Note: Properties marked with * are mandatory.*

Property	Type	Default value	Description
operation	string enum	CREATE	<p>CREATE – creates a new order in the system.</p> <p>UPDATE – updates an existing order in the system. If the <i>orderNo</i> does not exist the system will return an error.</p> <p>SYNC – recommended when the end goal is to make sure orders are reflected in OptimoRoute. If this is a new order, we will create it. If there is an existing order in the system, it will be updated. SYNC makes a clean sync-up of order data. Whatever we have on this order will be replaced by the data you send us on SYNC.</p> <p>MERGE – used to only update specific fields and leave all the other fields unchanged (for example if some fields were manually edited in OptimoRoute and should not be updated). If this is a new order, we will create it. If there is an existing order in the system, it will be updated.</p>
orderNo	string		A user specified order identifier, also displayed in the web application. It can also be used for deleting orders via the WS API.
relatedOrderNo	string		<p><i>orderNo</i> of the related order. Used to link pickups and deliveries in situations where goods are transported directly from one customer location to the another customer location.</p> <p>The relationship should be specified only on the second order that is created (the first order</p>

			has to already exist in the system so that it can be referenced).
acceptDuplicateOrderNo	boolean	False	If set to <i>True</i> the system will accept orders with <i>orderNo</i> that already exists in the system.
type*	string enum	D	Order type: Delivery (D), Pickup (P) or Task (T).
date*	date (string)		Delivery date. YYYY-MM-DD format, for example 2013-12-20.
location*	location object		Delivery/Service location.
duration*	integer	5	The time in minutes required to unload the goods or perform a task at the given location.
twFrom	time (string)		Deprecated, please use the <i>timeWindows</i> field. The earliest time allowed to begin the service (if the driver arrives too early, he will be forced to wait). 24-hour (military) clock format, for example: 16:00 The valid values range from 00:00 to 23:59.
twTo	time (string)		Deprecated, please use the <i>timeWindows</i> field. The deadline to end the service. 24-hour (military) clock format, for example 17:00 The valid values range from 00:00 to 23:59.
timeWindows	list of time windows	[]	List of time windows when the order can be serviced. Each time window is an object with <i>twFrom</i> and <i>twTo</i> fields, for example: <code>{"twFrom": "10:00", "twTo": "14:00"}</code> If <i>timeWindows</i> field is set, then <i>twFrom</i> and <i>twTo</i> fields within the order are ignored.
allowedWeekdays	list of strings	[]	Restricts days of the week when the order can be scheduled: Sunday (Sun), Monday (Mon), Tuesday (Tue), Wednesday (Wed), Thursday (Thu), Friday (Fri), Saturday (Sat). For example if the order can be scheduled on Monday or Tuesday the value should be: <code>["mon", "tue"]</code>
allowedDates	daterange object	No restriction	Restricts the dates when the order can be scheduled. Daterange is specified as an object with <i>from</i> and <i>to</i> fields, for example: <code>{"from": "2018-06-01", "to": "2018-07-01"}</code>
assignedTo	string		The serial number of the Driver that this Order must be assigned to. Setting this field forces this Order to be served by the specified Driver.
priority	string	M	Order priority. Values can be:

			<p>L – Low M – Medium H – High C – Critical</p> <p>High priority orders are scheduled earlier and are less likely to be left unscheduled if not all orders can be serviced.</p>
load1	integer	0	<p>The load requirements of the order, i.e. how many load units (Number of boxes, Kilos, Pounds, Liters etc) should be delivered. The meaning of this entry depends on the provided configuration of load/capacity constraints.</p>
load2	integer	0	See load1.
load3	integer	0	See load1.
load4	integer	0	See load1.
vehicleFeatures	list of strings		The vehicle features used to differentiate some Vehicles from the others. The required vehicle features are defined as a list of vehicle feature codes.
skills	list of strings		The driver skills used to differentiate some Drivers from the others. The required skills are defined as a list of skill codes.
notes	string		The optional note that will accompany the driver's instructions. Notes do not affect the optimization process. A free form string.
customField1	string		Value that will be saved in custom field #1.
customField2	string		Value that will be saved in custom field #2.
customField3	string		Value that will be saved in custom field #3.
customField4	string		Value that will be saved in custom field #4.
customField5	string		Value that will be saved in custom field #5.
email	string		Customer email. If enabled, order notifications can be sent to that email.
phone	string		Customer phone number.
notificationPreference	string	email	<p>Defines the delivery method for order tracking notifications for this order. Notifications will only be sent if this account has order tracking configured.</p> <p>Allowed values:</p> <ul style="list-style-type: none"> • email – send notifications to email address specified in the email field • dont_notify – don't send notifications

Location object can be defined with the following fields:

Property	Type	Default value	Description
address	string		The full address including the country, for example <i>393 Hanover St, Boston, MA 02113, US</i>
acceptMultipleResults	boolean	False	Used only if a new location is created by geocoding the <i>address</i> field. If set to <i>False</i> , the system will not accept the geocoded addresses where several results matching the provided address have been found and an error will be raised.
acceptPartialMatch	boolean	False	Used only if a new location is created by geocoding the <i>address</i> field. If set to <i>False</i> , the system will not accept the geocoded addresses that were only a partial match (lower geocoding confidence) and an error will be raised.
locationNo	string		The unique identifier for the given location.
locationName	string		Location name.
latitude	decimal		Location GPS latitude.
longitude	decimal		Location GPS longitude.
checkInTime	integer	0	Minimum waiting time at the location (in minutes). Upon arrival at the location the driver will spend at least <i>checkInTime</i> minutes before starting the service of one or several consecutive orders at the same location. If several orders at the same location have different values for check in time, the system will use the maximum value. Example: if we have 3 orders at the same location: <ul style="list-style-type: none"> • A > duration: 10min, check in time: 15min • B > duration: 5min, <i>check in time</i>: 15min • C > duration: 20min, <i>check in time</i>: 15min If one driver serves all three orders consecutively the driver would arrive at the location, spend 15 minutes at the location (for example parking and checking in) and then start servicing orders A, B and C. Total time at the location would be 50min = 15min (shared duration) + 10min (order A) + 5min (order B) + 20min (order C).

A location is defined by one of the following:

- **locationNo**– the location number of a location already existing in the system. An existing location is used. (**NOTE:** the **address**, **latitude + longitude** and **locationName** should **NOT** be specified, otherwise the system will try to geocode the location **again** or create one with a defined GPS location)
- **latitude + longitude + locationName** (with optional **address** and **locationNo**) – the location is defined by the GPS latitude and longitude.

The location name will be set to the value of *locationName* field.

The location address will be set to *address* if *address* is set, otherwise it will be left blank.

The location number will be set to *locationNo* if *locationNo* is set, otherwise it will be left blank.

- **address** (with optional **locationName** and **locationNo**) – the system will try to find an existing locations with the specified *address* (and *locationName* if one is specified). If an existing location is not found the location address will be geocoded based on the supplied *address*.

The location name will be set to *locationName* field if *locationName* is supplied, otherwise it will be set to the value of *address* field.

The location number will be set to *locationNo* if *locationNo* is set, otherwise it will be left blank.

Request example

```
curl -d '@reqbody.json' 'https://api.optimoroute.com/v1/create_order?key=AUTH_KEY'
```

(where reqbody.json is a local file containing the JSON data to be posted. See the request body example here below)

Request body example:

```
{
  "operation": "CREATE",
  "orderNo": "ORD001",
  "type": "D",
  "date": "2014-10-14",
  "location": {
    "address": "393 Hanover St, Boston, MA 02113, USA",
    "locationNo": "LOC001",
    "locationName": "Green Cross Pharmacy North End",
    "acceptPartialMatch": true
  },
  "duration": 20,
  "twFrom": "10:00",
  "twTo": "10:59",
  "load1": 10,
  "load2": 25,
  "vehicleFeatures": ["FR"],
  "skills": ["SK001", "SK002"],
  "notes": "Deliver at back door"
}
```

Return values

Property	Type	Default value	Description
success*	boolean		True if order was saved, false if there was an error.
code	string		An error code or a warning code (not set if the operation was successful).

message	string		An error description or a warning description (not set if the operation was successful).
location	location data		The location data for the created order (only if the operation was successful).
geocodingResults	list		The list of geocoding results if the location is created from the supplied address. Each geocoding result is defined as: [geocodedAddress*, latitude*, longitude*, partialMatchFlag]

Location data:

Property	Type	Default value	Description
address	string		Location address.
locationNo	string		Location number/identifier.
locationName	string		Location name.
latitude	decimal		Location GPS latitude.
longitude	decimal		Location GPS longitude.

Error and warning codes

- ERR_ORD_EXISTS - an order with the specified *orderNo* already exists in the system (checked only if *orderNo* field is set)
- ERR_RELATED_ORD_MISSING - the order with the ID passed in *relatedOrderNo* does not exist
- ERR_RELATED_ORD_MULTIPLE - multiple orders with ID passed in *relatedOrderNo* exist
- ERR_RELATED_ORD_LINK - order can not be linked to order with ID passed in *relatedOrderNo* (only pickups and deliveries can be linked)
- ERR_DRV_NOT_EXISTS - the driver with the serial number passed in *assignedTo* does not exist
- ERR_DRV_MULTIPLE - there is more than one drivers with the *assignedTo* serial number
- ERR_LOC_NOT_VALID - the specified location is not valid
- ERR_LOC_GEOCODING - the specified address could not be geocoded
- ERR_LOC_GEOCODING_MULTIPLE - multiple results have been found during geocoding
- ERR_LOC_GEOCODING_PARTIAL - the geocoder did not return an exact match for the original request
- ERR_LOC_NON_EXISTING_LOC - the location specified by *locationNo* does not exist
- ERR_LOC_MULTIPLE_LOC - multiple locations with specified *locationNo* have been found
- ERR_VF_NOT_EXISTS - the vehicle feature does not exist (for one of the codes specified in *vehicleFeatures* field)
- ERR_VF_MULTIPLE - multiple vehicle features exist (for one of the codes specified in the *vehicleFeatures* field)
- ERR_SK_NOT_EXISTS - the vehicle skill does not exist (for one of the codes specified in the *skills* field)
- ERR_SK_MULTIPLE - multiple skills exist (for one of the codes specified in *skills* field)
- ERR_TIMEOUT - only applicable if CREATE_IF_FEASIBLE operation is set - this error is raised if the operation timed out while waiting for another optimization to finish or the optimization took too long.

- `ERR_NOT_FEASIBLE` – only applicable if `CREATE_IF_FEASIBLE` operation is set – this error is raised if it is not possible to fulfill the order (taking in consideration all the existing orders and constraints)
- `WAR_LOC_GEOCODING_MULTIPLE` – multiple results have been found during geocoding (but the order was created because *acceptMultipleResults* was set to True)
- `WAR_LOC_GEOCODING_PARTIAL` – the geocoder did not return an exact match for the original request (but the order was created because *acceptPartialMatch* was set to True)

3.2. Delete Order

Description

Removes an order from the system.

URL: https://api.optimoroute.com/v1/delete_order

HTTP Method: POST

Input format

*Note: Properties marked with * are mandatory.*

Property	Type	Default value	Description
orderNo*	string		The user-specified identifier of the order to be deleted.

Request example

```
curl -d '@reqbody.json' 'https://api.optimoroute.com/v1/delete_order?key=AUTH_KEY'
```

(where reqbody.json is a local file containing the JSON data to be posted. See the request body example here below)

Request body example:

```
{
  "orderNo": "ORD001"
}
```

Return values

Property	Type	Default value	Description
success*	boolean		True if order was deleted, false if there was an error.
code	string		An error code or a warning code (not set if the operation was successful).
message	string		An error description or a warning description (not set if the operation was successful).
planningId	integer		The id of the planning process in case optimization is running for this order.

Error codes

- ERR_ORD_NOT_FOUND – the order with the matching *orderNo* was not found
- ERR_MULTIPLE_ORD_FOUND – there are multiple orders matching the *orderNo*
- ERR_OPT_RUNNING – optimization is running for this order

3.3. Delete All Orders

Description

Removes all orders and planned routes for the specified date from the system. If no date is set, all orders and routes are removed from the system.

URL: https://api.optimoroute.com/v1/delete_all_orders

HTTP Method: POST

Input format

*Note: Properties marked with * are mandatory.*

Property	Type	Default value	Description
date	date (string)		Date for which orders and routes will be deleted. YYYY-MM-DD format, for example 2013-12-20.

Request example

```
curl -d '@reqbody.json' 'https://api.optimoroute.com/v1/delete_all_orders?key=AUTH_KEY'
```

(where reqbody.json is a local file containing the JSON data to be posted. See the request body example here below)

Request body example:

```
{
  "date": "2014-10-14"
}
```

Return values

Property	Type	Default value	Description
success*	boolean		True if orders were deleted, false if there was an error.
code	string		An error code or a warning code (not set if the operation was successful).
message	string		An error description or a warning description (not set if the operation was successful).
planningId	integer		The id of the planning process in case optimization is running for this order.

Error codes

- ERR_OPT_RUNNING – optimization is running for this date

3.4. Get Routes

Description

Gets the Routes for a specific date.

URL: https://api.optimoroute.com/v1/get_routes

HTTP Method: GET

Input format

*Note: Properties marked with * are mandatory.*

Property	Type	Default value	Description
date*	date (string)		Queried date. YYYY-MM-DD format, for example <i>2013-12-20</i> .
driverSerial	string		Optionally filter by Serial number of the Driver.
vehicleRegistration	string		Optionally filter by Vehicle registration.
includeRoutePolyline	boolean	False	Optional property to include route polyline in the output. This polyline can be used to display the route on the map. The polyline is a list of coordinates encoded with Encoded Polyline Algorithm .

Request example

```
curl 'https://api.optimoroute.com/v1/get_routes?key=AUTH_KEY&date=2013-12-20'
```

Return values

Property	Type	Default value	Description
success*	boolean		True if the query was successful, false if there was an error.
routes*	list of route objects	[]	List of routes matching the query.
code	string		An error code or a warning code (not set if the operation was successful).
message	string		An error description or a warning description (not set if the operation was successful).

Route object:

Property	Type	Default value	Description
driverSerial*	string		The serial number assigned to the driver.
driverName*	string		The driver's name.
vehicleRegistration*	string		Vehicle registration.
vehicleLabel*	string		Vehicle label.
duration*	integer		Route duration in minutes.
distance*	decimal		Route distance in kilometers.
load1*	integer	0	Route load #1.
load2*	integer	0	Route load #2.
load3*	integer	0	Route load #3.
load4*	integer	0	Route load #4.
load5*	integer	0	Route load #5.
stops*	list of stop objects	[]	An ordered list of stops/orders on the route.
routePolyline	string		Route polyline encoded with Encoded Polyline Algorithm . Only included if <i>includeRoutePolyline</i> is set to <i>True</i> .

Stop object:

Property	Type	Default value	Description
stopNumber*	integer		Stop number. Starts at 1.
orderNo*	string		The order number.
locationNo*	string		Location number.
locationName*	string		Location name.
address*	string		Location address.
latitude*	decimal		Location latitude.
longitude*	decimal		Location longitude.
scheduledAt*	time (string)		The scheduled time to begin the service 24-hour (military) clock format, for example: <i>16:00</i> The valid values range from <i>00:00</i> to <i>23:59</i> .
scheduledAtDt*	datetime (string)		The scheduled date and time to begin the service. Format: <i>YYYY-mm-dd HH:MM:SS</i> , for example <i>2018-01-01 13:10:37</i> .
arrivalTimeDt*	datetime (string)		The date and time when the driver arrived at the location. Because of time windows the driver might need to wait to begin the service. Format: <i>YYYY-mm-dd HH:MM:SS</i> , for example <i>2018-01-01 13:10:37</i> .
travelTime*	integer		Travel time from previous stop (in seconds).
distance*	integer		Distance from previous stop (in meters).

type			Order type – only added for lunch breaks (value will be “break”) and for depot trips (value will be “depot”).
------	--	--	---

Error codes

None.

3.5. Get Scheduling Information

Description

Gets the scheduling information for the specified Order.

URL: https://api.optimoroute.com/v1/get_scheduling_info

HTTP Method: GET

Input format

*Note: Properties marked with * are mandatory.*

Property	Type	Default value	Description
orderNo*	string		The user-specified identifier for the order.

Request example

```
curl 'https://api.optimoroute.com/v1/get_scheduling_info?key=AUTH_KEY&orderNo=ORD001'
```

Return values

Property	Type	Default value	Description
success*	boolean		True if the query was successful, false if there was an error.
orderScheduled	boolean		True if the order is scheduled, False otherwise.
scheduleInformation	schedule info object		The order scheduling information.
code	string		An error code or a warning code (not set if the operation was successful).
message	string		An error description or a warning description (not set if the operation was successful).

Order scheduling information object:

Property	Type	Default value	Description
driverSerial*	string		The serial number assigned to the driver.
driverName*	string		The driver's name.
vehicleRegistration*	string		Vehicle registration.
vehicleLabel*	string		Vehicle label.
stopNumber*	integer		Stop number on the route. Starts at 1.
scheduledAt*	time (string)		The scheduled time to begin the service 24-hour (military) clock format, for example: 16:00 The valid values range from 00:00 to 23:59.

scheduledAtDt*	datetime (string)		The scheduled date and time to begin the service. Format: <i>YYYY-mm-dd HH:MM:SS</i> , for example <i>2018-01-01 13:10:37</i> .
arrivalTimeDt*	datetime (string)		The date and time when the driver arrived at the location. Because of time windows the driver might need to wait to begin the service. Format: <i>YYYY-mm-dd HH:MM:SS</i> , for example <i>2018-01-01 13:10:37</i> .
travelTime*	integer		Travel time from previous stop (in seconds).
distance*	integer		Distance from previous stop (in meters).

Error codes

- ERR_ORD_NOT_FOUND – order with the matching *orderNo* was not found
- ERR_MULTIPLE_ORD_FOUND – multiple orders with matching *orderNo* were found

3.6. Start Planning

Description

Starts the planning process for the specified date.

URL: https://api.optimoroute.com/v1/start_planning

HTTP Method: POST

Input format

*Note: Properties marked with * are mandatory.*

Property	Type	Default value	Description
date*	date (string)		Date to be planned. YYYY-MM-DD format, for example 2013-12-20.
balancing	string	OFF	Route balancing settings. Allowed values: OFF – No balancing ON – Balance routes ON_FORCE – Balance routes and use all drivers No balancing: returns the best routes, some drivers might not be used. Balance routes: balance the work load between drivers, some drivers may not be used. Balance routes and use all drivers: balance work load between all available drivers.
balanceBy	string	WT	If route balancing is turned on, this defines the criteria for balancing the routes. Allowed values: WT – Working time NUM – Number of orders per driver
balancingFactor	decimal	0.3	Importance of balancing compared to route costs. Increasing the balancing factor will result in more balanced routes. Only applicable in combination with ON_FORCE (otherwise ignored). Minimum value: 0.0 Maximum value: 1.0
startWith	string	EMPTY	Start planning from existing routes or from scratch. Allowed values: EMPTY – Ignore existing routes and start from scratch

			CURRENT – Start planning with the existing routes
lockType	string	NONE	Applicable if <i>startWith</i> is set to CURRENT. Allowed values: NONE – Allow all changes to the existing routes ROUTES - Keep planned Routes unchanged and add new Orders to unused RESOURCES - Keep Drivers/Vehicles for planned Orders and fit in new Orders
depotTrips	boolean	False	If <i>true</i> drivers will be scheduled to return to the warehouse to reload (or unload in case of pickups) if there is still time within their working hours to do another route. More information is available on our blog: https://optimoroute.com/stop-restock-and-continue-on-your-way/
depotVisitDuration	integer	0	Time in minutes required to reload or unload the goods in the warehouse before the driver goes out for another run. Only applicable if <i>depotTrips</i> is set to <i>true</i> .
useDrivers	list of SelectedDriver objects	[]	Selected drivers that should be used in the optimization. By default (if <i>useDrivers</i> is not set or the list is empty) all available drivers for the specified date are used.
useOrders	list of strings	[]	Selected orders that should be included in the optimization. Orders are specified as a list of order identifiers (<i>orderNo</i> strings). By default (if <i>useOrders</i> is not set or the list is empty) all orders for the specified date are included. Orders that are not selected, but are already scheduled to drivers that will be used in the optimization will be unscheduled unless the <i>includeScheduledOrders</i> flag is set to <i>true</i> .
includeScheduledOrders	boolean	true	Determines behavior when a subset of orders is selected but there are other orders already scheduled to the applicable drivers. If <i>includeScheduledOrders</i> is true, these orders will be included in the optimization, otherwise they will be unscheduled.

SelectedDriver object (use either *driverExternalId* or *driverSerial* - one field is enough):

Property	Type	Default value	Description
driverExternalId	string		The external identifier assigned to the driver in driver administration.
driverSerial	string		The driver's serial number.

Request example

```
curl -d '@reqbody.json' 'https://api.optimoroute.com/v1/start_planning?key=AUTH_KEY'
```

(where reqbody.json is a local file containing the JSON data to be posted. See the request body example here below)

Request body example:

```
{
  "date": "2013-12-20"
}
```

Return values

Property	Type	Default value	Description
success*	boolean		True if order was deleted, false if there was an error.
code	string		Error or warning code (not set if operation was successful).
planningId	integer		Returns id of the started optimization if operation was successful, or the already running optimization in case of ERR_OPT_RUNNING_FOR_DATE error. Planning id can be used later for getting the planning status or stopping the planning process.
missingOrders	list of strings	[]	In case ERR_ORD_NOT_FOUND error is raised this field will contain the list of missing order identifiers (<i>orderNo</i> strings).
missingDrivers	list of SelectedDriver objects	[]	In case ERR_DRIVER_NOT_FOUND error is raised this field will contain a list of missing drivers. Each driver is a <i>SelectedDriver</i> object.

Error codes

- ERR_OPT_TRIAL_ENDED – free trial has ended for this account
- ERR_OPT_RUNNING_FOR_DATE– planning is already running for this date
- ERR_OPT_RUNNING_FOR_ORDER – optimization is already running for one or more orders listed in *useOrders*
- ERR_OPT_NO_REQUESTS – no orders exist for the specified date
- ERR_OPT_NO_RESOURCES – no drivers are available for this date

- ERR_OPT_RESOURCES_EXCEEDED – number of driver exceeded
- ERR_OPT_REQUESTS_EXCEEDED – number of orders exceeded
- ERR_OPT_COULD_NOT_START – internal error, please contact support
- ERR_ORD_NOT_FOUND – one or more orders specified in *useOrders* were not found
- ERR_DRIVER_NOT_FOUND – one or more drivers specified in *useDrivers* were not found
- ERR_MULTIPLE_ORD_FOUND - multiple orders with specified identifier have been found
- ERR_MULTIPLE_DRV – multiple drivers with specified identifier have been found

3.7. Stop Planning

Description

Stops the planning process.

URL: https://api.optimoroute.com/v1/stop_planning

HTTP Method: POST

Input format

*Note: Properties marked with * are mandatory.*

Property	Type	Default value	Description
planningId*	integer		The id of the planning process to be stopped

Request example

```
curl -d '@reqbody.json' 'https://api.optimoroute.com/v1/stop_planning?key=AUTH_KEY'
```

(where reqbody.json is a local file containing the JSON data to be posted. See the request body example here below)

Request body example:

```
{
  "planningId": 8828
}
```

Return values

Property	Type	Default value	Description
success*	boolean		True if order was deleted, false if there was an error.
code	string		An error code or a warning code (not set if the operation was successful).

Error codes

- ERR_JOB_NOT_FOUND – planning job with specified id was not found
- ERR_OPT_NOT_RUNNING – planning job is not running
- ERR_OPT_COULD_NOT_STOP – internal error, please contact support

3.8. Get Planning Status

Description

Returns the status of the planning process.

URL: https://api.optimoroute.com/v1/get_planning_status

HTTP Method: GET

Input format

*Note: Properties marked with * are mandatory.*

Property	Type	Default value	Description
planningId*	integer		The id of the planning process

Request example

```
curl 'https://api.optimoroute.com/v1/get_planning_status?key=AUTH_KEY&planningId=8828'
```

Return values

Property	Type	Default value	Description
success*	boolean		True if the request was a success, false if an error occurred.
code	string		An error code or a warning code (not set if the operation was successful).
status	string		N - New R - Running C - Cancelled by the user F - Finished E - Error occurred
percentageComplete	integer		Percentage complete – values range from 0 to 100.

Error codes

- **ERR_JOB_NOT_FOUND** – the planning job with the specified id was not found

3.9. Update Driver Parameters

Description

Updates driver parameters for a particular date and driver start and end location. Any existing routes for the specified date/driver will be unscheduled. The following parameters can be changed:

- Enable/disable driver
- Driver work time from
- Driver work time to
- Assigned vehicle
- Vehicle load capacity #1
- Vehicle load capacity #2
- Vehicle load capacity #3
- Vehicle load capacity #4
- Driver start location
- Driver end location

URL: https://api.optimoroute.com/v1/update_driver_parameters

HTTP Method: POST

Input format

*Note: Properties marked with * are mandatory.*

Property	Type	Default value	Description
externalId *	string		The external identifier assigned to the driver in driver administration.
date*	date		Date to be planned. YYYY-MM-DD format, for example <i>2013-12-20</i> .
enabled	boolean	unchanged	Enable or disable driver for the specified date.
workTimeFrom	time (string)	unchanged	Driver's work time from for the specified date. 24-hour (military) clock format, for example: <i>08:00</i> The valid values range from <i>00:00</i> to <i>23:59</i> .
workTimeTo	time (string)	unchanged	Driver's work time end for the specified date. 24-hour (military) clock format, for example: <i>08:00</i> The valid values range from <i>00:00</i> to <i>23:59</i> .
assignedVehicle	string	unchanged	The external identifier of the vehicle to be assigned to the driver for the specified date. This will also update the vehicle load capacity parameters (unless they are overridden with vehicleCapacityX settings).
vehicleCapacity1	integer	unchanged	The new vehicle load capacity #1 for the specified date.
vehicleCapacity2	integer	unchanged	The new vehicle load capacity #2 for the specified date.
vehicleCapacity3	integer	unchanged	The new vehicle load capacity #3 for the specified date.
vehicleCapacity4	integer	unchanged	The new vehicle load capacity #4 for the specified date.
startLatitude	decimal	unchanged	Driver's start location GPS latitude will be changed (this new location will be used for all future optimizations). If startLongitude is not defined this value will be ignored.

startLongitude	decimal	unchanged	Driver's start location GPS longitude (this new location will be used for all future optimizations). If startLatitude is not defined this value will be ignored.
startAddress	string	unchanged	Driver's start location address that will be displayed on reports (this new location will be used for all future optimizations). If startLatitude and startLongitude are not defined this value will be ignored.
endLatitude	decimal	unchanged	Driver's end location GPS latitude (this new location will be used for all future optimizations). If endLongitude is not defined this value will be ignored.
endLongitude	decimal	unchanged	Driver's end location GPS longitude (this new location will be used for all future optimizations). If endLatitude is not defined this value will be ignored.
endAddress	string	unchanged	Driver's end location address that will be displayed on reports (this new location will be used for all future optimizations). If endLatitude and endLongitude are not defined this value will be ignored.

Request example

```
curl -d '@reqbody.json' 'https://api.optimoroute.com/v1/update_driver_parameters?key=AUTH_KEY'
```

Where reqbody.json is a local file containing the JSON data to be posted.

Request body example:

```
{
  "externalId": "3945300304540",
  "date": "2015-09-25",
  "workTimeFrom": "09:30",
  "workTimeTo": "18:00"
}
```

Return values

Property	Type	Default value	Description
success*	boolean		True if the status was updated, false if an error occurred.
code	string		An error code or a warning code (not set if the operation was successful).

Error codes

- ERR_DRIVER_NOT_FOUND – the driver with the specified *externalId* was not found
- ERR_MULTIPLE_DRV – multiple drivers with specified *externalId* have been found
- ERR_VEH_NOT_FOUND – the vehicle with the specified *externalId* was not found
- ERR_MULTIPLE_VEH – multiple vehicles with specified *externalId* have been found

3.10. Get Mobile Events

Description

Retrieve mobile events such as **success**, **on_duty**, **failed**, ... for the currently active plan (i.e. the last plan that was sent to drivers).

Each **get_events** request returns up to 500 events which occurred after the specified tag. The result will contain a new tag which you can use in the next **get_events** request to skip already received events.

URL: https://api.optimoroute.com/v1/get_events

HTTP Method: GET

Input format

Property	Type	Default value	Description
after_tag	string		Specify the after_tag to retrieve only the events which occurred since a specific prior call to get_events . In this case, the after_tag needs to contain the tag returned in the prior call. See example below.

Request example

In the first request we won't specify the **after_tag** parameter (we'll leave it empty):

```
curl 'https://api.optimoroute.com/v1/get_events?key=AUTH_KEY&after_tag='
```

```
{
  "remainingEvents": 0,
  "tag": "abcd1234xyz",
  "events": [
    {
      "unixTimestamp": 1516294800,
      "utcTime": "2018-01-18T17:00:00",
      "localTime": "2018-01-18T09:00:00",
      "driverName": "Charlie",
      "driverSerial": "0011",
      "event": "on_duty",
    },
    {
      "unixTimestamp": 1516296030,
      "event": "success",
      "utcTime": "2018-01-18T17:20:30",
      "localTime": "2018-01-18T09:20:30",
      "driverName": "Charlie",
      "driverSerial": "0011",
      "orderNo": "ORD232"
    }
  ]
}
```

```
"success": true
}
```

We'll retry the query in 10 seconds, but this time we'll use the tag we received in the last result as our `after_tag` parameter:

```
curl 'https://api.optimoroute.com/v1/get_events?key=AUTH_KEY&after_tag=abcd1234xyz'
```

```
{
  "remainingEvents": 0,
  "tag": "abcd1234xyz",
  "events": [],
  "success": true
}
```

There were no new events in the meantime. We'll retry the query in 10 seconds:

```
curl 'https://api.optimoroute.com/v1/get_events?key=AUTH_KEY&after_tag=abcd1234xyz'
```

```
{
  "remainingEvents": 0,
  "tag": "qwe9876541213",
  "events": [
    {
      "unixTimestamp": 1516298030,
      "event": "success",
      "utcTime": "2018-01-18T17:53:50",
      "localTime": "2018-01-18T09:53:50",
      "driverName": "Charlie",
      "driverSerial": "0011",
      "orderNo": "ORD771"
    }
  ],
  "success": true
}
```

Return values

Property	Type	Default value	Description
success*	string		True unless an error occurred.
events	list of event objects	[]	A list of events objects in the order they arrived from drivers' mobile apps.
tag	string		Tag marks the point of time of the last event received in this response. Use it to specify the <code>after_tag</code> query parameter in the next <code>get_events</code> request.

remainingEvents	integer	unchanged	The number of events that occurred but were not fetched in this response (in each response at most 500 events are returned).
-----------------	---------	-----------	--

Event object

Property	Type	Description
event*	string	Event type. Currently one of the following: on_duty – the driver went on duty off_duty – the driver went off duty success – the specified order was completed successfully failed – the driver failed to complete the order rejected – the driver rejected the order
unixTimestamp*	integer	The time at which event occurred. Number of seconds elapsed since 00:00 UTC January 1st 1970.
utcTime*	ISO 8601 date (string)	The UTC date and time at which the specified event occurred.
localTime*	ISO 8601 date (string)	The local time at which the specified event occurred.
driverName	string	The driver's name. Will be sent for events that are related to a specific driver.
driverSerial	string	The driver's serial number. Will not be sent if empty or if the event is not related to a specific driver.
driverExternalId	string	The driver's external identifier. Will not be sent if empty or if the event is not related to a specific driver.
orderNo	string	The identifier of the affected order. Not sent for some events such as on_duty .

The returned event fields **unixTimestamp**, **utcTime** and **localTime** all refer to the same moment at which the event occurred, so you can use whichever is more convenient.

Please note that some mobile devices might be offline temporarily and they will sync their events only after they reconnect so the events you'll receive might arrive out of chronological order. For example, the first driver might successfully complete the order 001 at 9:00 AM, but he might be offline until 09:15 AM. The second driver could finish the order 002 at 9:10 AM, but he might be online the whole time. The server will receive the event for the order 002 first and only then the event for the order 001.