

Tracker

This document contains tracker object structure and API calls to interact with it. Tracker is one of the key entities in our API. It represents tracking device registered in our GPS monitoring system. Lots of API calls created for manipulation of tracker and/or its properties.

{% hint style="warning" %}

The number of devices per account is restricted to 25,000, not including the deleted or hidden devices.

{% endhint %}

Tracker object structure

```
{
  "id": 123456,
  "label": "tracker label",
  "clone": false,
  "group_id": 167,
  "avatar_file_name": "file name",
  "source": {
    "id": 234567,
    "device_id": 99999999888888,
    "model": "telfmb920",
    "blocked": false,
    "tariff_id": 345678,
    "status_listing_id": null,
    "creation_date": "2011-09-21",
    "tariff_end_date": "2016-03-24",
    "phone": "71234567890"
  },
  "tag_bindings": [
    {
      "tag_id": 456789,
      "ordinal": 4
    }
  ]
}
```

```
]
}
```

- `id` - int. Tracker ID aka `object_id`.
- `label` - string. Tracker label.
- `clone` - boolean. `true` if this tracker is clone.
- `group_id` - int. Tracker group ID, 0 when no group.
- `avatar_file_name` - string. Optional. Passed only if present.
- `source` - object.
 - `id` - int. Source ID.
 - `device_id` - string. Device ID aka `source_imei`.
 - `model` - string. Tracker model name from "models" table.
 - `blocked` - boolean. `true` if tracker blocked due to tariff end.
 - `tariff_id` - int. An ID of tracker tariff from "main_tariffs" table.
 - `status_listing_id` - int. An ID of the status listing associated with this tracker, or null.
 - `creation_date` - [date/time](#). Date when the tracker registered.
 - `tariff_end_date` - [date/time](#). Date of next tariff prolongation, or null.
 - `phone` - string. Phone of the device. Can be null or empty if device has no GSM module or uses bundled SIM which number hidden from the user.
- `tag_binding` - object. List of attached tags. Appears only for [tracker/list](#) call.
 - `tag_id` - int. An ID of tag. Must be unique for a tracker.
 - `ordinal` - int. Number that can be used as ordinal or kind of tag. Must be unique for a tracker. Max value is 5.

Tracker output info

```
{
  "number": 1,
  "title": "OUT1"
}
```

- `number` - int. Output number.
- `title` - string. User-defined output name.

Tracker input types

- **ignition** - Car's ignition. There can be only one sensor of this type.
- **engine** - Engine's working status.
- **mass** - Car's "ground".

- **car_alarm** - Expected to be “on” when car alarm triggered.
- **sos_button** - An emergency “red” button.
- **hood** - “on” if engine’s hood is open.
- **door** - “on” if car’s door is open.
- **car_lock** - “on” if car’s central lock is open.
- **custom** - user-defined type. In general, should have non-empty “name” field.

API actions

API base path: `/tracker` .

read

Gets tracker info by ID.

Parameters

name	description	type	format
tracker_id	ID of the tracker (aka “object_id”)	int	999199

Examples

```
{% tabs %}
```

```
{% tab title="cURL" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/read' \
  -H 'Content-Type: application/json' \
  -d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "tracker_id": 123456}'
```

```
{% endtab %}
```

```
{% tab title="HTTP GET" %}
```

```
{% code overflow="wrap" %}
```

```
https://pro.lepton-seguridad.com/api-v2/tracker/read?hash=a6aa75587e5c59c32d347d
```

```
{% endcode %}
```

```
{% endtab %}
```

```
{% endtabs %}
```

Response

```
{
  "success": true,
  "value": {
    "id": 123456,
    "label": "Object 1",
    "group_id": 0,
    "source": {
      "id": 10021901,
      "device_id": "123456789009876",
      "model": "atrack_ak11",
      "blocked": false,
      "tariff_id": 1294,
      "phone": "79161234533",
      "status_listing_id": null,
      "creation_date": "2021-09-20",
      "tariff_end_date": "2021-09-24"
    },
    "tag_bindings": [],
    "clone": false
  }
}
```

See tracker object structure description [here](#).

Errors

- 201 - Not found in the database – if tracker not found.

list

Gets user's trackers with optional filtering by labels. We described this API call in the [guide](#).

Parameters

name	description	type	format
------	-------------	------	--------

labels	Optional. List of tracker label filters. If specified, only trackers that labels contains any of the given filter will be returned.	string array	["aa", "b"]
--------	---	--------------	-------------

Constraints for labels:

- Labels array size: minimum 1, maximum 1024.
- No null items.
- No duplicate items.
- Item length: minimum 1, maximum 60.

For example, we have trackers with labels "aa1", "bb2", "cc3", if we pass `labels=["aa","b"]` only trackers containing "aa1" and "bb2" will be returned.

Examples

```
{% tabs %}
{% tab title="cURL" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/list' \
-H 'Content-Type: application/json' \
-d '{"hash": "a6aa75587e5c59c32d347da438505fc3"}'
```

```
{% endtab %}
{% tab title="HTTP GET" %}
```

```
https://pro.lepton-seguridad.com/api-v2/tracker/list?hash=a6aa75587e5c59c32d347da4
```

```
{% endtab %}
{% endtabs %}
```

Response

```
{
  "success": true,
  "list": [
    {
      "id": 123456,
      "label": "tracker label",
```

```

"clone": false,
"group_id": 167,
"avatar_file_name": "file name",
"source": {
  "id": 234567,
  "device_id": 99999999888888,
  "model": "telfmb920",
  "blocked": false,
  "tariff_id": 345678,
  "status_listing_id": null,
  "creation_date": "2011-09-21",
  "tariff_end_date": "2016-03-24",
  "phone": "+71234567890"
},
"tag_bindings": [
  {
    "tag_id": 456789,
    "ordinal": 4
  }
]
}
]
}

```

See tracker object structure description [here](#).

Errors

[General](#) types only.

corrupt

Marks tracker as deleted and corrupt its source, device_id and phone.

required sub-user rights: `tracker_register` .

Parameters

name	description	type	format
tracker_id	ID of the tracker (aka "object_id"). Tracker must belong to authorized user and not be blocked.	int	999119

Examples

```
{% tabs %}
```

```
{% tab title="cURL" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/corrupt' \  
-H 'Content-Type: application/json' \  
-d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "tracker_id": 265489}'
```

```
{% endtab %}
```

```
{% tab title="HTTP GET" %}
```

```
{% code overflow="wrap" %}
```

```
https://pro.lepton-seguridad.com/api-v2/tracker/corrupt?hash=a6aa75587e5c59c32d34
```

```
{% endcode %}
```

```
{% endtab %}
```

```
{% endtabs %}
```

Response

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

Errors

- 13 – Operation not permitted – if tracker already connected to server, or if user has insufficient rights.
- 243 – Device already connected.
- 201 – Not found in the database - if tracker not found.
- 219 – Not allowed for clones of the device - if source tracker is clone itself.
- 252 – Device already corrupted.
- 208 – Device blocked.

delete

Deletes a tracker if it is “clone”. Will not work if specified ID of the original tracker.

required sub-user rights: `admin` (available only to master users).

Parameters

name	description	type	format
tracker_id	ID of the tracker (aka "object_id"). Tracker must belong to authorized user and not be blocked.	int	999119

Examples

{% tabs %}

{% tab title="cURL" %}

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/delete' \  
-H 'Content-Type: application/json' \  
-d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "tracker_id": 265489}'
```

{% endtab %}

{% tab title="HTTP GET" %}

{% code overflow="wrap" %}

```
https://pro.lepton-seguridad.com/api-v2/tracker/delete?hash=a6aa75587e5c59c32d347
```

{% endcode %}

{% endtab %}

{% endtabs %}

Response

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

Errors

- 201 - Not found in the database – if tracker not found.
- 249 - Operation available for clones only – if tracker is not clone.
- 203 - Delete entity associated with – if there are some rules or vehicles associated with tracker.

```
{
  "success": false,
  "status": {
    "code": 203,
    "description": "Delete entity associated with"
  },
  "rules": [10]
}
```

or

```
{
  "success": false,
  "status": {
    "code": 203,
    "description": "Delete entity associated with"
  },
  "vehicles": [11]
}
```

- rules - list of associated rule IDs.
- vehicles - list of associated vehicle IDs.

change_phone

Changes tracker's phone and setup new apn.

required sub-user rights: tracker_configure .

Parameters

name	description	type	format
tracker_id	ID of the tracker (aka "object_id"). Tracker must belong to authorized user and not be blocked.	int	999199
phone	The phone number of the sim card inserted into device in international format without "+" sign.	string	"6156680000"

apn_name	The name of GPRS APN of the sim card inserted into device. Max length 40.	string	"fast.tmobile.com"
apn_user	The user of GPRS APN of the sim card inserted into device. Max length 40, can be empty.	string	"tmobile"
apn_password	The password of GPRS APN of the sim card inserted into device. Max length 40, can be empty.	string	"tmobile"

Examples

cURL

{% code overflow="wrap" %}

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/change_phone' \
-H 'Content-Type: application/json' \
-d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "tracker_id": 265489, "phone": "6
```

{% endcode %}

Response

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

Errors

- 201 - Not found in the database – if tracker not found.
- 208 – Device blocked - if tracker exists but was blocked due to tariff restrictions or some other reason.
- 219 – Not allowed for clones of the device - if specified tracker is a clone.
- 214 – Requested operation or parameters are not supported by the device - if device does not have GSM module.
- 223 – Phone number already in use - if specified phone number already used in another device.
- 241 – Cannot change phone to bundled sim. Contact tech support. If specified phone number belongs to sim card bundled

with the device.

get_diagnostics

Gets last CAN and OBD sensors and states values received from the device.

Parameters

name	description	type	format
tracker_id	ID of the tracker (aka "object_id"). Tracker must belong to authorized user and not be blocked.	int	999119

Examples

```
{% tabs %}
```

```
{% tab title="cURL" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/get_diagnostics' \  
-H 'Content-Type: application/json' \  
-d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "tracker_id": 265489}'
```

```
{% endtab %}
```

```
{% tab title="HTTP GET" %}
```

```
{% code overflow="wrap" %}
```

```
https://pro.lepton-seguridad.com/api-v2/tracker/get_diagnostics?hash=a6aa75587e5c59
```

```
{% endcode %}
```

```
{% endtab %}
```

```
{% endtabs %}
```

Response

```
{  
  "success": true,  
  "user_time": "2021-05-20 13:49:09",  
  "inputs": [  
    {  
      "label": "OBD: RPM",
```

```

"units": "",
"name": " obd_rpm",
"type": "rpm",
"value": 889.0,
"units_type": "custom",
"converted_units_type": null,
"converted_value": null
}
],
"states": {
"obd_vin": "123",
"obd_mil_status": "0"
},
"update_time": "2021-05-20 13:48:02"
}

```

- `user_time` - [date/time](#). Current time in user's timezone.
- `inputs` - list of `sensor value` objects.
 - `label` - string. Sensor's label. E.g. "Sensor #1".
 - `name` - [enum](#). Name of sensor's raw input.
 - `type` - [enum](#). Type of quantity, measured by a sensor.
 - `value` - float. Reading's value, measured in units from an eponymous field. E.g. 100.0.
 - `units_type` - [enum](#). Unit of measurement of input to the sensor.
 - `units` - string. User label for sensor's units.
 - `converted_units_type` - [enum](#). Unit of measurement system preferred by current user (according to user/settings), suitable for this sensor. Can be null, if there is no need in conversion (unit of sensor's input (field `units_type`) belongs to user's measurement system).
 - `converted_value` - float. Reading's value in units from field `converted_units_type` . Can be null if there is no need in conversion.
- `states` - map of last state values or null (see below).
- `update_time` - [date/time](#). Date and time when the data updated.

List of available sensor's input names for the object `sensor value` :

- **obd_consumption.**
- **obd_rpm.**

- **obd_fuel.**
- **obd_coolant_t.**
- **obd_intake_air_t.**
- **obd_throttle.**
- **obd_speed.**
- **obd_engine_load.**
- **obd_absolute_load_value** (normalised value of air mass per intake stroke in percents).
- **obd_control_module_voltage** (in volts).
- **obd_time_since_engine_start** (run time since engine start in seconds).
- **obd_mil_run_time** (in minutes).
- **can_engine_temp.**
- **can_engine_hours.**
- **can_mileage.**
- **can_throttle.**
- **can_consumption.**
- **can_rpm.**
- **can_speed.**
- **can_r_prefix.**
- **can_coolant_t.**
- **can_intake_air_t.**
- **can_engine_load.**
- **can_adblue_level.**
- **can_fuel_rate** (instant fuel consumption liter/hour).
- **raw_can_x** (range for x: [1 – 16]).
- **can_axle_load_x** (range for x: [1 – 15]).

List of state names for the field `states` :

- **obd_vin** (value type: string).
- **obd_dtc_number** (DTC codes number; value type: integer).
- **obd_dtc_codes** (DTC codes; value type: string).
- **obd_dtc_cleared_distance** (distance traveled since codes cleared in km; value type: double).
- **obd_mil_activated_distance** (distance traveled with MIL on in km; value type: double).
- **hardware_key** (driver identification key; value type: string).
- **external_power_state** (connected/disconnected; value type: string).
- **driver_ident_state** (identified/not identified; value type: string).
- **tacho_vin** (value type: string).
- **tacho_card1_sn** (value type: string).

- **tacho_card2_sn** (value type: string).
- **tacho_vin_last_download** (value type: string).
- **tacho_card1_last_download** (value type: string).
- **tacho_card2_last_download** (value type: string).
- **can_hood_state** (value type: string, 0 or 1 means "close" or "open").
- **can_airbag_state** (value type: string, 0 or 1 means "normal" or "malfunction").
- **can_trunk_state** (value type: string, 0 or 1 means "close" or "open").
- **can_seat_belt_driver_state** (value type: string, 0 or 1 means "untied" or "tied").
- **can_seat_belt_passenger_state** (value type: string, 0 or 1 means "untied" or "tied").
- **can_door_state** (value type: string, 0 or 1 means "close" or "open").
- **can_door_driver_state** (value type: string, 0 or 1 means "close" or "open").
- **can_door_passenger_state** (value type: string, 0 or 1 means "close" or "open").

You can locate all inputs, states, and definitions by utilizing the [tracker/sensor/input_name/list](#) API call.

Errors

- 201 – Not found in the database - if there is no tracker with such ID belonging to authorized user.
- 208 – Device blocked - if tracker exists but was blocked due to tariff restrictions or some other reason.

get_fuel

Gets current fuel level (in liters) of tracker's fuel tanks.

Parameters

name	description	type	format
tracker_id	ID of the tracker (aka "object_id"). Tracker must belong to authorized user and not be blocked.	int	999119

Examples

```
{% tabs %}
```

```
{% tab title="cURL" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/get_fuel' \
-H 'Content-Type: application/json' \
```

```
-d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "tracker_id": 265489}'
```

```
{% endtab %}
```

```
{% tab title="HTTP GET" %}
```

```
{% code overflow="wrap" %}
```

```
https://pro.lepton-seguridad.com/api-v2/tracker/get_fuel?hash=a6aa75587e5c59c32d34
```

```
{% endcode %}
```

```
{% endtab %}
```

```
{% endtabs %}
```

Response

```
{
  "success": true,
  "user_time": "2021-05-20 13:49:09",
  "inputs": [
    {
      "label": "Sensor #1",
      "name": "can_fuel",
      "type": "fuel",
      "value": 100.0,
      "units_type": "litre",
      "units": "litres",
      "converted_units_type": null,
      "converted_value": null
    }
  ],
  "update_time": "2021-05-20 13:48:02"
}
```

- `user_time` - [date/time](#). Current time in user's timezone.
- `inputs` - array of last readings of fuel-related sensors. Items are object listed below.

List of available sensor's input names for the object `sensor value` :

- **fuel_level.**
- **fuel_frequency.**

- **lls_level_x** (range for x: [1 – 16]).
- **fuel_consumption**.
- **rs232_x** (range for x: [1 – 6]).
- **can_fuel** (fuel level in percents or in unknown units).
- **can_fuel_2** (fuel level in percents or in unknown units).
- **can_fuel_litres** (fuel level in litres).
- **can_fuel_economy** (fuel economy in km/litres).
- **update_time** - **date/time**. Date and time when the data updated.

Errors

- 201 – Not found in the database - if there is no tracker with such ID belonging to authorized user.
- 208 – Device blocked - if tracker exists but was blocked due to tariff restrictions or some other reason.

get_inputs

Gets current state of tracker’s digital inputs and “semantic” inputs (ignition, buttons, car alarms, etc.)

bound to them (if any).

Parameters

name	description	type	format
tracker_id	ID of the tracker (aka “object_id”). Tracker must belong to authorized user and not be blocked.	int	999119

Examples

```
{% tabs %}
```

```
{% tab title="cURL" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/get_inputs' \
  -H 'Content-Type: application/json' \
  -d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "tracker_id": 265489}'
```

```
{% endtab %}
```

```
{% tab title="HTTP GET" %}
```

```
{% code overflow="wrap" %}
```

https://pro.lepton-seguridad.com/api-v2/tracker/get_inputs?hash=a6aa75587e5c59c32c

{% endcode %}

{% endtab %}

{% endtabs %}

Response

```
{
  "success": true,
  "user_time": "2021-05-20 13:49:09",
  "inputs": [true, true, false],
  "states": [
    {
      "type": "ignition",
      "name": "DIN1",
      "status": true,
      "input_number": 1
    }
  ],
  "update_time": "2021-05-20 13:48:02"
}
```

- `user_time` - [date/time](#). Current time in user's timezone.
- `inputs` - array (boolean) of states of all digital inputs. `[true, true, false]` means input 1 is on, input 2 is on, input 3 is off.
- `states` - array of state objects.
 - `type` - [enum](#). One of predefined semantic [input types](#)
 - `name` - string. User-defined name for semantic input, or null if not specified.
 - `status` - boolean. True if input is active, false otherwise.
 - `input_number` - int. Number of the associated discrete input.
- `update_time` - [date/time](#). Date and time when the data updated.

Errors

- 201 – Not found in the database - if there is no tracker with such ID belonging to authorized user.

- 208 – Device blocked - if tracker exists but was blocked due to tariff restrictions or some other reason.

batch_get_inputs

Gets current state of trackers' digital inputs and "semantic" inputs (ignition, buttons, car alarms, etc.)

bound to them (if any).

Parameters

name	description	type	format
trackers	IDs of the trackers (aka "object_id"). Each tracker must belong to authorized user and not be blocked.	int array	[999199, 999919]

Examples

```
{% tabs %}
```

```
{% tab title="cURL" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/batch_get_inputs' \
-H 'Content-Type: application/json' \
-d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "trackers": [265489]}'
```

```
{% endtab %}
```

```
{% tab title="HTTP GET" %}
```

```
{% code overflow="wrap" %}
```

```
https://pro.lepton-seguridad.com/api-v2/tracker/batch_get_inputs?hash=a6aa75587e5c
```

```
{% endcode %}
```

```
{% endtab %}
```

```
{% endtabs %}
```

Response

```
{
  "success": true,
```

```
"user_time": "2021-05-20 13:49:09",
"data": {
  "265489": {<input info>}
}
}
```

- `user_time` - [date/time](#). Current time in user's timezone.
- `data` - object. Input info mapped to tracker ids.

Input info:

```
{
  "inputs": [true, true, false],
  "states": [
    {
      "type": "ignition",
      "name": "DIN1",
      "status": true,
      "input_number": 1
    }
  ],
  "update_time": "2021-05-20 13:48:02"
}
```

- `inputs` - array (boolean) of states of all digital inputs. `[true, true, false]` means input 1 is on, input 2 is on, input 3 is off.
- `states` - array of state objects.
 - `type` - [enum](#). One of predefined semantic [input types](#)
 - `name` - string. User-defined name for semantic input, or null if not specified.
 - `status` - boolean. True if input is active, false otherwise.
 - `input_number` - int. Number of the associated discrete input.
- `update_time` - [date/time](#). Date and time when the data updated.

Errors

- 217 - List contains nonexistent entities - if one of `trackers` either does not exist or is blocked.
- 221 - Device limit exceeded - if too many IDs were passed in `trackers` parameter.

get_outputs

Gets tracker's outputs info

Parameters

name	description	type	format
tracker_id	ID of the tracker (aka "object_id"). Tracker must belong to authorized user and not be blocked.	int	999119

Examples

```
{% tabs %}
```

```
{% tab title="cURL" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/get_outputs' \  
-H 'Content-Type: application/json' \  
-d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "tracker_id": 265489}'
```

```
{% endtab %}
```

```
{% tab title="HTTP GET" %}
```

```
{% code overflow="wrap" %}
```

```
https://pro.lepton-seguridad.com/api-v2/tracker/get_outputs?hash=a6aa75587e5c59c3;
```

```
{% endcode %}
```

```
{% endtab %}
```

```
{% endtabs %}
```

```
=== "HTTP GET"
```

Response

```
{  
  "success": true,  
  "result": [<output info>]  
}
```

- **result** - array of objects. Array of output info objects described [here](#).

Errors

- 201 – Not found in the database - if there is no tracker with such ID belonging to authorized user.
- 208 – Device blocked - if tracker exists but was blocked due to tariff restrictions or some other reason.

batch_get_outputs

Gets trackers' outputs info

Parameters

name	description	type	format
trackers	IDs of the trackers (aka "object_id"). Each tracker must belong to authorized user and not be blocked.	int array	[999199, 999919]

Examples

```
{% tabs %}
```

```
{% tab title="cURL" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/batch_get_outputs' \  
-H 'Content-Type: application/json' \  
-d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "trackers": [265489]}'
```

```
{% endtab %}
```

```
{% tab title="HTTP GET" %}
```

```
{% code overflow="wrap" %}
```

```
https://pro.lepton-seguridad.com/api-v2/tracker/batch_get_outputs?hash=a6aa75587e5
```

```
{% endcode %}
```

```
{% endtab %}
```

```
{% endtabs %}
```

Response

```
{
  "success": true,
  "result": {
    "265489": [<output info>]
  }
}
```

- `result` - object. Array of output info objects described [here](#) mapped to tracker ids.

Errors

- 217 - List contains nonexistent entities - if one of `trackers` either does not exist or is blocked.
- 221 - Device limit exceeded - if too many IDs were passed in `trackers` parameter.

output/update

Updates tracker's outputs info

Parameters

name	description	type
tracker_id	ID of the tracker (aka "object_id"). Tracker must belong to authorized user and not be blocked.	int
tracker_output	Output info object containing number and title.	output info

Examples

cURL

```
{% code overflow="wrap" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/output/update' \
  -H 'Content-Type: application/json' \
  -d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "tracker_id": 265489, "tracker_ou
```

```
{% endcode %}
```

Response

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

Errors

- 201 – Not found in the database - if there is no tracker with such ID belonging to authorized user.
- 208 – Device blocked - if tracker exists but was blocked due to tariff restrictions or some other reason.

get_last_gps_point

Gets last point of the tracker located by GPS. Points located by GSM LBS are excluded from consideration.

Parameters

name	description	type	format
tracker_id	ID of the tracker (aka "object_id"). Tracker must belong to authorized user and not be blocked.	int	999119

Examples

```
{% tabs %}
{% tab title="cURL" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/get_last_gps_point' \
  -H 'Content-Type: application/json' \
  -d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "tracker_id": 265489}'
```

```
{% endtab %}
```

```
{% tab title="HTTP GET" %}
{% code overflow="wrap" %}
```

```
https://pro.lepton-seguridad.com/api-v2/tracker/get_last_gps_point?hash=a6aa75587e5
```

```
{% endcode %}
{% endtab %}
```

```
{% endtabs %}
```

Response

```
{
  "success": true,
  "value": {
    "get_time": "2012-03-05 12:00:00",
    "heading": 11,
    "lat": 22.0,
    "lng": 33.0,
    "satellites": 5,
    "speed": 20,
    "precision": 100
  }
}
```

- **value** - track point object.
 - **get_time** - **date/time**. GPS timestamp of the point, in user's timezone.
 - **heading** - int. Direction bearing in degrees (0-360).
 - **lat** - float. Latitude.
 - **lng** - float. Longitude.
 - **satellites** - int. Number of satellites used in fix for this point.
 - **speed** - int. Speed in km/h.
 - **precision** - int. Optional. Exists if not equal to 0. Precision in meters.

Errors

- 201 - Not found in the database – if there is no tracker with such ID belonging to authorized user.
- 208 - Device blocked – if tracker exists but was blocked due to tariff restrictions or some other reason.

get_readings

Gets last sensor values for sensors that are:

- **metering.**
- **not can- or obd-based.**
- **not "fuel" sensors.**

Parameters

name	description	type	format
tracker_id	ID of the tracker (aka "object_id"). Tracker must belong to authorized user and not be blocked.	int	999119

Examples

{% tabs %}

{% tab title="cURL" %}

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/get_readings' \  
-H 'Content-Type: application/json' \  
-d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "tracker_id": 265489}'
```

{% endtab %}

{% tab title="HTTP GET" %}

{% code overflow="wrap" %}

```
https://pro.lepton-seguridad.com/api-v2/tracker/get_readings?hash=a6aa75587e5c59c3
```

{% endcode %}

{% endtab %}

{% endtabs %}

Response

```
{  
  "success": true,  
  "user_time": "2021-05-20 13:49:09",  
  "inputs": [  
    {  
      "label": "Board voltage",  
      "units": "V",  
      "name": "board_voltage",  
      "type": "power",  
      "value": 13.562,  
      "units_type": "custom",  
      "converted_units_type": null,  
    }  
  ]  
}
```

```

    "converted_value": null
  }
],
"update_time": "2021-05-20 13:48:02"
}

```

- `user_time` - [date/time](#). Current time in user's timezone.
- `inputs` - list of `sensor value` objects. See below.
 - `label` - string. Sensor's label. E.g. "Sensor #1".
 - `name` - [enum](#). Name of sensor's raw input.
 - `type` - [enum](#). Type of quantity, measured by a sensor.
 - `value` - float. Reading's value, measured in units from an eponymous field. E.g. 100.0.
 - `units_type` - [enum](#). Unit of measurement of input to the sensor.
 - `units` - string. User label for sensor's units.
 - `converted_units_type` - [enum](#). Unit of measurement system preferred by current user (according to user/settings), suitable for this sensor. Can be null, if there is no need in conversion (unit of sensor's input (field `units_type`) belongs to user's measurement system).
 - `converted_value` - float. Reading's value in units from field `converted_units_type`. Can be null if there is no need in conversion.
- `update_time` - [date/time](#). Date and time when the data updated.

List of available sensor's input names for the object `sensor value` :

- **composite.**
- **input_status.**
- **analog_x** (range for x: [1 – 8]).
- **freq_x** (range for x: [1 – 8]).
- **impulse_counter_x** (range for x: [1 – 8]).
- **fuel_temperature.**
- **lts_temperature_x** (range for x: [1 – 16]).
- **rs232_x** (range for x: [1 – 6]).
- **board_voltage.**
- **temp_sensor.**
- **ext_temp_sensor_x** (range for x: [1 – 10]).

Errors

- 201 – Not found in the database - if there is no tracker with such ID belonging to authorized user.
- 208 – Device blocked - if tracker exists but was blocked due to tariff restrictions or some other reason.

get_state

Gets current tracker state (gps, gsm, outputs, etc.).

name	description	type	format
tracker_id	ID of the tracker (aka "object_id"). Tracker must belong to authorized user and not be blocked.	int	999119

Examples

```
{% tabs %}
```

```
{% tab title="cURL" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/get_state' \
  -H 'Content-Type: application/json' \
  -d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "tracker_id": 265489}'
```

```
{% endtab %}
```

```
{% tab title="HTTP GET" %}
```

```
{% code overflow="wrap" %}
```

```
https://pro.lepton-seguridad.com/api-v2/tracker/get_state?hash=a6aa75587e5c59c32d347da438505fc3
```

```
{% endcode %}
```

```
{% endtab %}
```

```
{% endtabs %}
```

Response

```
{
  "user_time": "2022-08-31 13:47:13",
  "state": {
    "source_id": 545139,
    "gps": {
```

```

"updated": "2022-08-31 13:47:09",
"signal_level": 100,
"location": {
  "lat": 42.82769,
  "lng": -78.26290833333333
},
"heading": 45,
"speed": 0,
"alt": 0
},
"connection_status": "active",
"movement_status": "parked",
"movement_status_update": "2022-08-31 13:40:44",
"ignition": false,
"ignition_update": "2022-08-31 13:40:44",
"gsm": {
  "updated": "2022-08-31 13:47:09",
  "signal_level": 100,
  "network_name": "Mobile",
  "roaming": false
},
"last_update": "2022-08-31 13:47:09",
"battery_level": 97,
"battery_update": "2022-08-31 13:47:09",
"inputs": [false, false, false],
"inputs_update": "2022-08-31 13:47:09",
"outputs": [true, false],
"outputs_update": "2022-08-31 13:47:09",
"additional": {
  "hardware_key": {
    "value": "20910998202956382057",
    "updated": "2022-08-31 10:47:09"
  }
},
"actual_track_update": "2022-08-31 13:40:44"
},
"success": true
}

```

- `user_time` - [date/time](#). Current time in user's timezone.

- `source_id` - int. Tracker data source ID (from "sources" table).
- `gps` - gps object.
 - `updated` - `date/time`. Date of last gps coordinates update in a timezone of the user or null if there are no updates.
 - `signal_level` - int. GPS signal level in percent, e.g. 25, or null if device cannot provide such info.
 - `lat` - float. Latitude.
 - `lng` - float. Longitude.
 - `heading` - int. Direction bearing in degrees (0-360).
 - `speed` - int. Speed in km/h, e.g. 20.
 - `alt` - int. Altitude in meters, e.g. 10.
 - `precision` - int. Optional. Precision in meters.
 - `gsm_lbs` - boolean. Optional. True if location detected by GSM LBS.
- `connection_status` - `enum`. Device connection status, possible values: "signal_lost", "just_registered", "just_replaced", "offline", "idle", "active".
- `movement_status` - `enum`. Movement status, possible values: "moving", "stopped", "parked".
- `movement_status_update` - `date/time`. The date and time when the movement status was last changed or null if there are no changes.
- `ignition` - boolean. Optional. State of vehicle's or virtual ignition sensor.
- `ignition_update` - `date/time`. Optional. The date and time when the ignition state was last changed.
- `gsm` - object. Can be null if device does not support transmission of gsm info.
 - `updated` - `date/time`. Date of last gsm status update in a timezone of the user or null if there are no updates.
 - `signal_level` - int. GSM signal level in percent, e.g. 25, or null if device cannot provide such info.
 - `network_name` - string. GSM network name, e.g. "T-MOBILE", or null if device cannot provide such info.
 - `roaming` - boolean. Roaming state, or null if device cannot provide such info.
- `last_update` - `date/time`. Date of last device state update in a timezone of the user or null if there are no updates.
- `battery_level` - int. Battery level in percent, e.g. 25, or null if device cannot provide such info.
- `battery_update` - `date/time`. Date of last battery update in a timezone of the user or null if there are no updates.

- `inputs` - array of boolean. States of all digital inputs. `[true, true, false]` means input 1 is on, input 2 is on, input 3 is off.
- `inputs_update` - `date/time`. Date of last inputs update in a timezone of the user or null if there are no updates.
- `outputs` - array of boolean. States of all digital outputs. `[true, true, false]` means output 1 is on, output 2 is on, output 3 is off.
- `outputs_update` - `date/time`. Date of last outputs update in a timezone of the user or null if there are no updates.
- `additional` - object. map of additional states, keys depends on tracker model.
 - `hardware_key` - last scanned hardware key object.
 - `value` - int. Hardware key.
 - `updated` - `date/time`. Date of last hardware key update in a timezone of the user or null if there are no updates.
- `actual_track_update` - `date/time`. When the last track was updated last time, when device last time moved.

Errors

- 201 – Not found in the database (if there is no tracker with such ID belonging to authorized user).
- 208 – Device blocked (if tracker exists but was blocked due to tariff restrictions or some other reason).

get_states

Gets current states (gps, gsm, outputs, etc.) for several trackers.

Parameters

name	description	type	format
trackers	The ID of trackers (also called "object_id" or "tracker_id"). Trackers must belong to an authorized user and must not be blocked. There is a limit of 2000 trackers per request. If you have more than 2000 devices, please split them into separate requests.	int array	<code>[999119, 999199]</code>

list_blocked	Optional. If <code>true</code> call returns list of blocked tracker IDs instead of error 208. Default is <code>false</code> .	boolean	true/false
allow_not_exist	Optional. If <code>true</code> call returns list of nonexistent tracker IDs instead of error 217 or 201. Default is <code>false</code> .	boolean	true/false

Examples

cURL

```
{% code overflow="wrap" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/get_states' \
-H 'Content-Type: application/json' \
-d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "trackers": [1234567, 888999, 11
```

```
{% endcode %}
```

Response

```
{
  "user_time": "2024-08-01 13:45:23",
  "states": {
    "1234567": {
      "source_id": 227676,
      "gps": {
        "updated": "2024-08-01 13:45:18",
        "signal_level": 100,
        "location": {
          "lat": 45.0888968,
          "lng": 10.0343262
        },
        "heading": 304,
        "speed": 64,
        "alt": 270
      },
      "connection_status": "active",
      "movement_status": "moving",
      "movement_status_update": "2024-08-01 13:38:55",
```

```
"ignition": true,
"ignition_update": "2024-08-01 13:37:34",
"last_update": "2024-08-01 13:45:23",
"gsm": {
  "updated": "2024-08-01 13:45:18",
  "signal_level": 35,
  "network_name": "T-Mobile",
  "roaming": null
},
"battery_level": 100,
"battery_update": "2024-08-01 13:45:18",
"inputs": [
  false,
  false,
  true
],
"inputs_update": "2024-08-01 13:45:18",
"outputs": [
  true,
  false
],
"outputs_update": "2024-08-01 13:45:18",
"additional": {
  "moving": {
    "value": "1",
    "updated": "2024-08-01 13:45:18"
  },
  "status": {
    "value": "82961756",
    "updated": "2024-08-01 13:45:18"
  }
},
"actual_track_update": "2024-08-01 13:38:55"
},
"blocked": [
  888999
],
"not_exist": [
  111333
```

```

],
"success": true
}

```

- `user_time` - [date/time](#). Current time in user's timezone.
- `states` - object. A map containing state objects for requested trackers, where the key is the tracker ID and the value is the state (see state object description in [tracker/get_state](#) response).
- `blocked` - array of tracker IDs. Returned only if `list_blocked= true` .
- `not_exist` - array of tracker IDs. Returned only if `allow_not_exist= true` .

Errors

- 201 – Not found in the database (if tracker corrupted and `allow_not_exist = false`).
- 208 – Device blocked (if `list_blocked = false` and tracker exists but was blocked due to tariff restrictions or some other reason).
- 217 – List contains nonexistent entities (if `allow_not_exist = false` and there are nonexistent trackers belonging to an authorized user).

list_models

Gets all integrated tracker models (from "models" table).

Parameters

name	description	type	format
<code>compact_view</code>	Optional. <code>true</code> to compact view. Default is <code>false</code> .	boolean	true/false
<code>compact_index</code>	Optional. <code>true</code> to compact view the indexed inputs: returns only input with max index. Default is <code>false</code> , but this value is deprecated.	boolean	true/false
<code>codes</code>	Optional. Array of model codes. If passed only given models will be returned.	string array	[<code>model_1</code> , <code>model_2</code> , ...]

Examples

```
{% tabs %}
```

```
{% tab title="cURL" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/list_models' \  
-H 'Content-Type: application/json' \  
-d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "compact_index": true}'
```

```
{% endtab %}
```

```
{% tab title="HTTP GET" %}
```

```
{% code overflow="wrap" %}
```

```
https://pro.lepton-seguridad.com/api-v2/tracker/list_models?hash=a6aa75587e5c59c32
```

```
{% endcode %}
```

```
{% endtab %}
```

```
{% endtabs %}
```

Response

```
{  
  "id": 2450,  
  "vendor": "lepton",  
  "code": "lepton_ngp",  
  "parent_code": null,  
  "type": "vehicle",  
  "name": "lepton Generic Protocol",  
  "id_type": "ascii,6,64",  
  "has_phone": false,  
  "has_apn_settings": false,  
  "register": true,  
  "has_auto_registration": false,  
  "port": null,  
  "battery": {  
    "min_charge": 0.0,  
    "low_charge": 0.1,  
    "max_charge": 1.0  
  },  
  "altitude": true,  
  "satellites": true,
```

```
"gsm_level": true,  
"gsm_network": true,  
"gsm_roaming": true,  
"has_detach_button": false,  
"has_fuel_input": true,  
"analog_inputs": 32,  
"digital_inputs": 8,  
"digital_outputs": 8,  
"rs232_inputs": 0,  
"track_control": "none",  
"output_control": "default",  
"special_control": "none",  
"rules": [  
  "speedup",  
  "inoutzone",  
  "route",  
  "offline",  
  "track_change",  
  "sensor_range",  
  "driver_change",  
  "task_status_change",  
  "fuel_level_leap",  
  "distance_control",  
  "excessive_driving",  
  "excessive_parking",  
  "state_field_control",  
  "sos",  
  "input_change",  
  "output_change",  
  "idling_soft"  
],  
"inputs": [  
  "analog_32",  
  "board_voltage",  
  "ext_temp_sensor_32",  
  "humidity_32",  
  "hw_mileage",  
  "lvs_level_32",  
  "lvs_temperature_32",  
  "temp_sensor"
```

```

],
"state_fields": [
  "event_id",
  "hardware_key",
  "moving",
  "obd_vin"
],
"special_settings": [
  "none"
],
"sms_control": [],
"connection": [
  {
    "protocol": "NGP",
    "transport": "HTTPS",
    "url": "https://ngp-tracker.example.com"
  }, {
    "protocol": "NGP",
    "transport": "MQTTS",
    "url": "mqtt://ngp_device:secretword@example.com:8883/ngp",
    "description": "Credentials for connection: login: ngp_device, password: secretword,
  }
],
"has_led_control": false,
"has_location_request": false,
"has_gprs_location_request": false,
"has_gsm_lbs_location_request": false,
"has_chat": false,
"has_custom_fields": true,
"has_odometer": true,
"has_lbs": true,
"has_motion_sensor": true,
"has_hardware_key": true,
"register_fields": []
}

```

- `id` - int. Model ID.
- `vendor` - string. Vendor name.
- `code` - string. Model text code
- `parent_code` - string. Can be null.

- `type` - `enum`. Can be "logger", "portable", "vehicle", or "personal".
- `name` - string. Model name.
- `id_type` - string. Identifier type see description below.
- `has_phone` - boolean. `true` if the tracker has phone.
- `has_apn_settings` - boolean. `true` if the tracker has APN settings.
- `register` - boolean. `true` if the tracker is available for registration.
- `has_auto_registration` - boolean. If `true` device may register by automatic commands from the platform.
- `port` - int, optional. The port number to connect to the tracking server. Can be null if the model does not support anything, or if it supports multiple connection types, the details will be in the `connection` field.
- `battery` - object. An internal device's battery.
 - `min_charge` - float. Minimum battery level. Used to calculate the current battery level.
 - `low_charge` - float. Charge level for the "low battery" rule triggers.
 - `max_charge` - float. Maximum battery level. Used to calculate the current battery level.
- `altitude` - boolean. `true` if the tracker supports altitude.
- `satellites` - boolean. `true` if the tracker supports the number of satellites.
- `gsm_level` - boolean. `true` if the tracker supports GSM signal level strength.
- `gsm_network` - boolean. `true` if the tracker supports GSM network name.
- `gsm_roaming` - boolean. `true` if the tracker supports GSM roaming state.
- `has_detach_button` - boolean. `true` if the tracker has detaching sensor.
- `has_fuel_input` - boolean. `true` if the tracker has fuel sensor.
- `analog_inputs` - int. Number of analog inputs.
- `digital_inputs` - int. Number of digital inputs.
- `digital_outputs` - int. Number of digital outputs.
- `rs232_inputs` - int. Number of RS232 inputs.
- `output_control` - `enum`. Can be "none", "default", "batch", "stateless", "async", "async_offline" or "batch_async".
- `special_control` - string. Additional specific types of tracker control (see [settings/special](#)). If multiple are separated by commas.
- `inputs` - array of `enum`. All available input types.
- `state_fields` - array of `enum`. All available state fields.
- `rules` - array of `enum`. Supported rules.
- `special_settings` - array of `enum`. Additional specific types of tracker control (see [settings/special](#)).
- `sms_control` - array of `enum`. Supported SMS control commands.
- `has_led_control` - boolean. `true` if a switching LED supported by this tracker.

- `has_location_request` - boolean. `true` if the tracker has an opportunity to request a location with a command by SMS.
- `has_gprs_location_request` - boolean. `true` if the tracker has an opportunity to request a location with a command over a GPRS connection.
- `has_gsm_lbs_location_request` - boolean. `true` if the tracker has an opportunity to request a location by LBS with a command over a GPRS connection.
- `has_chat` - boolean. `true` if chat is available for a device with this model.
- `has_custom_fields` - boolean, optional, default `false`. `true` if the protocol of this model supports transmission of fields (attributes) names. It allows to set a custom `input_name` for [sensors](#).
- `has_odometer` - boolean. `true` if the tracker has an integrated odometer.
- `has_lbs` - boolean. `true` if the tracker sends information about cell info.
- `has_motion_sensor` - boolean. `true` if the tracker has an integrated motion sensor.
- `has_hardware_key` - boolean. `true` if the tracker has an opportunity for identification of a driver by a hardware key.
- `additional_fields` - optional. List of descriptions of special fields using for control trackers that users fill on time of registration.
- `connection` - array of objects, optional. A list of options for connecting the model to the platform.
 - `protocol` - string. The name of the application layer protocol.
 - `transport` - string. Transport layer protocol.
 - `url` - string. Uniform Resource Locator - full details for the connection. Such as protocol, host, port, credentials, etc.
 - `description` - string. Connection details.

ID type

An ID type used to determine the information needed to register device in our system (see [tracker/register](#)).

Possible values are:

- **imei** – means device uses IMEI as its identifier, e.g. "356938035643809". See [Wikipedia article](#). When needed, you should pass only digits of IMEI, no spaces, minus signs, etc.
- **meid** means device uses MEID consisting of 14 HEX digits as its identifier, e.g. "A10000009296F2". See [Wikipedia article](#).

- **id,n** – means device uses n-digit identifier (factory ID with length N), for example, “id,7” means that you must pass 7-digit number, for example “1234567”.
- **n,m** – n-digit generated ID starting with M. This means that device has configurable ID and our platform generates and configures it automatically. You don’t need to pass any identifier during device registration in this case.

Errors

General types only.

tags/set

Set tags for a tracker. Tags must be created.

Parameters

name	description	type	format
tracker_id	ID of the tracker (aka “object_id”). Tracker must belong to authorized user and not be blocked.	int	999119
tag_bindings	List of tag_binding objects.	array of json objects	[{"tag_id": 1, "ordinal": 1}, {"tag_id": 2, "ordinal": 2}]

Examples

cURL

```
{% code overflow="wrap" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/tags/set' \
-H 'Content-Type: application/json' \
-d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "tracker_id": 123456, "tag_bindin
```

```
{% endcode %}
```

Response

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

Errors

General types only.

location_request

Execute this command to get current position of the device. The device must support requesting function.

Parameters

name	description	type	format
tracker_id	ID of the tracker (aka "object_id"). Tracker must belong to authorized user and not be blocked.	int	999119
type	Optional. Default type <code>sms</code> .	enum	"sms"

Request types:

- **sms** – GNSS data via SMS. Will send an SMS to request location. SMS gateway must be installed for the panel.
- **gsm** – GSM LBS data via GPRS. Device must have `online` or `GPS not updated` status.
- **gprs** – GNSS data via GPRS. Device must have `online` or `GPS not updated` status.

Examples

```
{% tabs %}
```

```
{% tab title="cURL" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/location_request' \
  -H 'Content-Type: application/json' \
  -d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "tracker_id": 123456}'
```

```
{% endtab %}
```

```
{% tab title="HTTP GET" %}
{% code overflow="wrap" %}
```

```
https://pro.lepton-seguridad.com/api-v2/tracker/location_request?hash=a6aa75587e5c!
```

```
{% endcode %}
{% endtab %}
{% endtabs %}
```

Response

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

Errors

- 201 – Not found in the database - if there is no tracker with such ID belonging to authorized user.
- 208 – Device blocked - if tracker exists but was blocked due to tariff restrictions or some other reason.
- 213 – Cannot perform action: the device is offline.
- 214 – Requested operation or parameters are not supported by the device.
- 256 – Location already actual.

register

Registers a new tracker device. During registration, device linked with current API user's account

and automatically configured to send data to our servers (if device model supports it).

The panel must have installed SMS gateway.

Find detailed instructions on tracker registration [there](#).

required sub-user rights: tracker_register .

Parameters

Important!

Because of the variety of tracker models and business applications, there are

different ways to register tracker in our system. They are called [Registration plugins](#). Each of registration plugins has its own set of additional parameters.

In addition to parameters specified in this section, pass all parameters which are required by the plugin you have chosen. See example below.

Common parameters are:

name	description	type	format
label	User-defined label for this tracker. Must consist of printable characters and have length between 1 and 60.	string	"Courier"
group_id	Tracker group ID, 0 if tracker does not belong to any group. The specified group must exist. See group/list .	int	0
model	A code of one of the supported models. See tracker/list_models .	string	"pt10"
plugin_id	An ID of a registration plugin which will be used to register the device. See Registration plugins .	int	37
device_id	Must be specified if device model uses fixed device ID. See tracker/list_models .	string	"4568005588562"
send_register_commands	Indicates send or not to send activation commands to device (via SMS or GPRS channel). If parameter is not specified or equals <code>null</code> will be used the platform settings. Default: <code>null</code> .	boolean	true or false

Examples

In this example we use plugin ID = 37 (see [Plugin description](#)) to register Quealink GV55Lite. We chose to include the device to default group, so group ID is 0.

As this device identified by IMEI, we include it as device ID (123451234512346).

Also, we include **phone**, **apn_name**, **apn_user**, **apn_password** of the sim card installed in device and **activation_code** since these parameters required by the plugin.

```
{% tabs %}
```

```
{% tab title="cURL" %}
```

```
{% code overflow="wrap" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/register' \
  -H 'Content-Type: application/json' \
  -d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "label": "Courier", "group_id": 0, "
```

```
{% endcode %}
```

```
{% endtab %}
```

```
{% tab title="HTTP GET" %}
```

```
{% code overflow="wrap" %}
```

```
https://pro.lepton-seguridad.com/api-v2/tracker/register?hash=a6aa75587e5c59c32d34
```

```
{% endcode %}
```

```
{% endtab %}
```

```
{% endtabs %}
```

Response

```
{
  "success": true,
  "value": {
    "id": 833389,
    "label": "Courier",
    "group_id": 0,
    "source": {
      "id": 526383,
```

```
"device_id": "123451234512346",
"model": "qlgv55lite",
"blocked": false,
"tariff_id": 12163,
"phone": "79123122312",
"status_listing_id": null,
"creation_date": "2021-06-03",
"tariff_end_date": "2021-06-17"
},
"clone": false
}
}
```

For `tracker` object structure, see [tracker/](#).

Errors

- 13 – Operation not permitted – if user has insufficient rights.
- 204 – Entity not found - if specified group does not exist. See [group/list](#).
- 220 – Unknown device model - if specified device model does not exist.
- 221 – Device limit exceeded - if device limit set for the user's dealer has been exceeded.
- 222 – Plugin not found - if specified plugin not found or is not supported by device model.
- 223 – Phone number already in use - if specified phone number already used in another device.
- 224 – Device ID already in use - if specified device ID already registered in the system.
- 225 – Not allowed for this legal type - if tariff of the new device is not compatible with user's legal type.
- 226 – Wrong ICCID. Plugin specific: if specified ICCID was not found.
- 227 – Wrong activation code. Plugin specific: if specified activation code not found or is already activated.
- 258 – Bundle not found. Plugin specific: if bundle not found for specified device ID.

register_retry

Resends registration commands to the device. The panel must have installed SMS gateway.

required sub-user rights: `tracker_register` .

Parameters

name	description	type	format
tracker_id	ID of the tracker (aka "object_id"). Tracker must belong to authorized user and not be blocked.	int	999119
device_id	Optional. Device ID that was used to register, e.g. IMEI. It can be used instead of <code>tracker_id</code> for models with a fixed ID.	string	"4568005588562"
apn_name	The name of GPRS APN of this sim card inserted into device. Max length 40.	string	"fast.tmobile.com"
apn_user	The user of GPRS APN of this sim card inserted into device. Max length 40, can be empty.	string	"tmobile"
apn_password	The password of GPRS APN of the sim card inserted into device. Max length 40, can be empty.	string	"tmobile"
send_register_commands	Indicates send or not to send activation commands to device (via SMS or GPRS channel). If parameter is not specified or equals <code>null</code> will be used the platform settings. Default: <code>null</code> .	boolean	true or false

Examples

```
{% tabs %}
```

```
{% tab title="cURL" %}
```

```
{% code overflow="wrap" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/register_retry' \  
-H 'Content-Type: application/json' \  
-d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "tracker_id": 999119, "apn_name
```

{% endcode %}

{% endtab %}

{% tab title="HTTP GET" %}

{% code overflow="wrap" %}

```
https://pro.lepton-seguridad.com/api-v2/tracker/register_retry?hash=a6aa75587e5c59c
```

{% endcode %}

{% endtab %}

{% endtabs %}

Response

```
{  
  "success": true,  
  "value": {  
    "id": 123456,  
    "label": "tracker label",  
    "clone": false,  
    "group_id": 167,  
    "avatar_file_name": "file name",  
    "source": {  
      "id": 234567,  
      "device_id": 99999999888888,  
      "model": "telfmb920",  
      "blocked": false,  
      "tariff_id": 345678,  
      "status_listing_id": null,  
      "creation_date": "2011-09-21",  
      "tariff_end_date": "2016-03-24",  
      "phone": "+71234567890"  
    },  
    "tag_bindings": [  
      {  
        "tag_id": 456789,  

```

```
"ordinal": 4
}
]
}
}
```

For `tracker` object structure, see [tracker/](#).

Errors

- 13 – Operation not permitted – if user has insufficient rights.
- 201 – Not found in the database - if there is no tracker with such ID belonging to authorized user.
- 208 – Device blocked - if tracker exists but was blocked due to tariff restrictions or some other reason.
- 219 – Not allowed for clones of the device - if specified tracker is a clone.
- 214 – Requested operation or parameters are not supported by the device - if device does not have GSM module.
- 242 – Device already connected - if tracker connected to the server.

register_quick

Registers a new tracker using only IMEI. Automatic SMS commands will not be sent for a register.

The device must be preconfigured. This API call can be used only for bundles.

required sub-user rights: `tracker_register` .

Parameters

name	description	type	format
label	User-defined label for this tracker. Must consist of printable characters and have length between 1 and 60.	string	"Courier"
group_id	Tracker group ID, 0 if tracker does not belong to any group. The specified group must exist. See group/list .	int	0
imei	Tracker's IMEI.	string	"35645587458999"

Examples

```
{% tabs %}
{% tab title="cURL" %}
{% code overflow="wrap" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/register_quick' \
-H 'Content-Type: application/json' \
-d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "label": "Courier", "group_id": 0, "
```

```
{% endcode %}
{% endtab %}
```

```
{% tab title="HTTP GET" %}
{% code overflow="wrap" %}
```

```
https://pro.lepton-seguridad.com/api-v2/tracker/register_quick?hash=a6aa75587e5c59c
```

```
{% endcode %}
{% endtab %}
{% endtabs %}
```

Response

```
{
  "success": true,
  "value": {
    "id": 123456,
    "label": "tracker label",
    "clone": false,
    "group_id": 167,
    "avatar_file_name": "file name",
    "source": {
      "id": 234567,
      "device_id": 99999999888888,
      "model": "telfmb920",
      "blocked": false,
      "tariff_id": 345678,
      "status_listing_id": null,
      "creation_date": "2011-09-21",
      "tariff_end_date": "2016-03-24",
      "phone": "71234567890"
    }
  }
}
```

```
},
"tag_bindings": [
  {
    "tag_id": 456789,
    "ordinal": 4
  }
]
}
}
```

For `tracker` object structure, see [tracker/](#).

Errors

- 13 – Operation not permitted – if user has insufficient rights.
- 201 – Not found in the database - if there is no bundle with such IMEI.
- 204 – Entity not found - if specified group does not exist.
- 220 – Unknown device model - if specified device model does not exist.
- 221 – Device limit exceeded - if device limit set for the user's dealer has been exceeded.
- 222 – Plugin not found - if specified plugin not found or is not supported by device model.
- 223 – Phone number already in use - if specified phone number already used in another device.
- 224 – Device ID already in use - if specified device ID already registered in the system.
- 225 – Not allowed for this legal type - if tariff of the new device is not compatible with user's legal type.
- 226 – Wrong ICCID - if specified ICCID was not found.
- 227 – Wrong activation code - if specified activation code not found or is already activated.

replace

Lets to replace the device without losing its history and some of its settings.

Replacement allows you to register a new device with history, sensors (optional), and rules (optional) of the current tracker saved.

required sub-user rights: `tracker_configure` .

Parameters

Important!

Because of the variety of tracker models and business applications, there are different ways to register a new tracker in our system. They are called [Registration plugins](#).

Each of registration plugins has its own set of additional parameters.

In addition to parameters specified in this section, pass all parameters which are required by the plugin you have chosen. See example below.

Common parameters are:

name	description	type	format
tracker_id	ID of the tracker (aka "object_id"). Tracker must belong to authorized user and not be blocked.	int	
model	A code of one of the supported models. See tracker/list_models .	string	"pt10"
device_id	Must be specified if device model uses fixed device ID. See tracker/list_models .	string	"4568005588562"
plugin_id	An ID of a registration plugin which will be used to register the device. See Registration plugins .	int	37
send_register_commands	Indicates send or not to send activation commands to a new device (via SMS or GPRS channel). If parameter is not specified or equals <code>null</code> will be used the platform settings. Default: <code>null</code> .	boolean	true/false

Examples

In this example we use plugin ID = 37 (see [Plugin description](#)) to replace device with Queclink GV55Lite.

As this device identified by IMEI, we include it as device ID (123451234512346).

Also, we include **phone**, **apn_name**, **apn_user**, **apn_password** of the sim card installed in device. Activation code is not used when replacing a device.

```
{% tabs %}
```

```
{% tab title="cURL" %}
```

```
{% code overflow="wrap" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/replace' \  
-H 'Content-Type: application/json' \  
-d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "tracker_id": 123456, "plugin_id":
```

```
{% endcode %}
```

```
{% endtab %}
```

```
{% tab title="HTTP GET" %}
```

```
{% code overflow="wrap" %}
```

```
https://pro.lepton-seguridad.com/api-v2/tracker/replace?hash=a6aa75587e5c59c32d34
```

```
{% endcode %}
```

```
{% endtab %}
```

```
{% endtabs %}
```

Response

```
{  
  "success": true,  
  "value": {  
    "id": 833389,  
    "label": "Courier",  
    "group_id": 0,  
    "source": {  
      "id": 526383,  
      "device_id": "123451234512346",  
      "model": "qlgv55lite",  
      "blocked": false,  
      "tariff_id": 12163,  
      "phone": "79123122312",
```

```
"status_listing_id": null,
"creation_date": "2021-06-03",
"tariff_end_date": "2021-06-17"
},
"clone": false
}
}
```

For `tracker` object structure, see [tracker/](#).

Errors

- 7 – Invalid parameters - if fields violate restrictions described above or one of the models is a mobile app.
- 13 – Operation not permitted - if user has insufficient rights.
- 204 – Entity not found - if specified group does not exist. See [group/list](#).
- 220 – Unknown device model - if specified device model does not exist.
- 221 – Device limit exceeded - if device limit set for the user's dealer has been exceeded.
- 222 – Plugin not found - if specified plugin not found or is not supported by device model.
- 223 – Phone number already in use - if specified phone number already used in another device.
- 224 – Device ID already in use - if specified device ID already registered in the system.
- 225 – Not allowed for this legal type - if tariff of the new device is not compatible with user's legal type.
- 226 – Wrong ICCID. Plugin specific: if specified ICCID was not found.
- 258 – Bundle not found. Plugin specific: if bundle not found for specified device ID.
- 266 – Cannot perform action for the device in current status: if the device is not activated yet

replace_quick

Replaces a device using only IMEI. Automatic SMS commands will not be sent for an activation.

The replacement device must be preconfigured. This API call can be used only for bundles.

required sub-user rights: `tracker_configure` .

Parameters

name	description	type	format
tracker_id	ID of the tracker (aka "object_id"). Tracker must belong to authorized user and not be blocked.	int	
imei	IMEI of the new device	string	"35645587458999"

Examples

```
{% tabs %}  
{% tab title="cURL" %}  
{% code overflow="wrap" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/replace_quick' \  
-H 'Content-Type: application/json' \  
-d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "tracker_id": 123456, "imei": "356
```

```
{% endcode %}  
{% endtab %}  
  
{% tab title="HTTP GET" %}  
{% code overflow="wrap" %}
```

```
https://pro.lepton-seguridad.com/api-v2/tracker/replace_quick?hash=a6aa75587e5c59c
```

```
{% endcode %}  
{% endtab %}  
{% endtabs %}
```

Response

```
{  
  "success": true,  
  "value": {  
    "id": 123456,  
    "label": "tracker label",  
    "clone": false,  
    "group_id": 167,
```

```
"avatar_file_name": "file name",
"source": {
  "id": 234567,
  "device_id": 99999999888888,
  "model": "telfmb920",
  "blocked": false,
  "tariff_id": 345678,
  "status_listing_id": null,
  "creation_date": "2011-09-21",
  "tariff_end_date": "2016-03-24",
  "phone": "71234567890"
},
"tag_bindings": [
  {
    "tag_id": 456789,
    "ordinal": 4
  }
]
}
}
```

For `tracker` object structure, see [tracker/](#).

Errors

- 7 – Invalid parameters - if fields violate restrictions described above or one of the models is a mobile app.
- 13 – Operation not permitted - if user has insufficient rights.
- 201 – Not found in the database - if there is no bundle with such IMEI.
- 204 – Entity not found - if specified group does not exist.
- 220 – Unknown device model - if specified device model does not exist.
- 221 – Device limit exceeded - if device limit set for the user's dealer has been exceeded.
- 222 – Plugin not found - if specified plugin not found or is not supported by device model.
- 223 – Phone number already in use - if specified phone number already used in another device.
- 224 – Device ID already in use - if specified device ID already registered in the system.
- 225 – Not allowed for this legal type - if tariff of the new device is not compatible with user's legal type.

- 226 – Wrong ICCID - if specified ICCID was not found.
- 266 – Cannot perform action for the device in current status: if the device is not activated yet

replace_retry

Resends registration commands to the new device. The panel must have installed SMS gateway.

required sub-user rights: tracker_configure .

Parameters

name	description	type	format
tracker_id	ID of the tracker (aka "object_id"). Tracker must belong to authorized user and not be blocked.	int	999119
apn_name	The name of GPRS APN of this sim card inserted into device.	string	"fast.tmobile.com"
apn_user	The user of GPRS APN of this sim card inserted into device.	string	"tmobile"
apn_password	The password of GPRS APN of the sim card inserted into device.	string	"tmobile"

Examples

```
{% tabs %}
```

```
{% tab title="cURL" %}
```

```
{% code overflow="wrap" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/register_retry' \
  -H 'Content-Type: application/json' \
  -d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "tracker_id": 999119, "apn_name
```

```
{% endcode %}
```

```
{% endtab %}
```

```
{% tab title="HTTP GET" %}
{% code overflow="wrap" %}
```

```
https://pro.lepton-seguridad.com/api-v2/tracker/register_retry?hash=a6aa75587e5c59c
```

```
{% endcode %}
{% endtab %}
{% endtabs %}
```

Response

```
{
  "success": true,
  "value": {
    "id": 123456,
    "label": "tracker label",
    "clone": false,
    "group_id": 167,
    "avatar_file_name": "file name",
    "source": {
      "id": 234567,
      "device_id": 99999999888888,
      "model": "telfmb920",
      "blocked": false,
      "tariff_id": 345678,
      "status_listing_id": null,
      "creation_date": "2011-09-21",
      "tariff_end_date": "2016-03-24",
      "phone": "+71234567890"
    },
    "tag_bindings": [
      {
        "tag_id": 456789,
        "ordinal": 4
      }
    ]
  }
}
```

For `tracker` object structure, see [tracker/](#).

Errors

- 13 – Operation not permitted – if user has insufficient rights.
 - 204 – Entity not found - if there is no tracker with such ID belonging to authorized user.
 - 208 – Device blocked - if tracker exists but was blocked due to tariff restrictions or some other reason.
 - 219 – Not allowed for clones of the device - if specified tracker is a clone.
 - 214 – Requested operation or parameters are not supported by the device - if device does not have GSM module.
 - 242 – Device already connected - if tracker connected to the server.
 - 266 – Cannot perform action for the device in current status: if the old device is not activated yet
-

send_command

Sends command to tracker for performing special control, determined with `special_control` field of tracker model.

required sub-user rights: `tracker_configure` , `tracker_set_output` .

common command format is:

```
{
  "command": {
    "name": "command name",
    "some_parameter1": 12,
    "some_parameter2": "parameter",
    "special_settings": {
      "type": "settings type",
      "some_field1": 10,
      "some_field2": 32
    }
  }
}
```

- `name` - Command name.
- `some_parameter` - Parameters depend on certain command.
- `special_settings` - optional field. Its structure defined with `special_control` field of tracker model.

Certain commands which can be used is defined with `special_control` field of **tracker model** and corresponds the table below:

special control	available commands
jointech_lock_password	electronic_lock_command, set_special_settings_command
hhd_lock_password	electronic_lock_command, set_special_settings_command
vg_lock_password	electronic_lock_command, set_special_settings_command
any other special control	set_special_settings_command

command types

electronic_lock_command

This command used to seal/unseal electronic lock.

```
{
  "name": "electronic_lock_command",
  "command_code": "unseal",
  "special_settings": {<special settings JSON object>}
}
```

- `command_code` - **enum**. Can be "seal" or "unseal".
- `special_settings` - This command is equivalent to API call [tracker/settings/special/update](#).

```
{
  "name": "set_special_settings_command",
  "special_settings": {<special settings JSON object>}
}
```

See [special settings JSON object](#)

Parameters

name	description	type	format
tracker_id	ID of the tracker (aka "object_id"). Tracker must belong to authorized user and not be blocked.	int	999119

command	Command that will be sent to device. Not Null.	JSON object	See format above
---------	--	-------------	------------------

Examples

cURL

{% code overflow="wrap" %}

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/send_command' \  
-H 'Content-Type: application/json' \  
-d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "tracker_id": 999119, "command"
```

{% endcode %}

Response

```
{  
  "success": true,  
  "list": [  
    {  
      "id": 123456,  
      "label": "tracker label",  
      "clone": false,  
      "group_id": 167,  
      "avatar_file_name": "file name",  
      "source": {  
        "id": 234567,  
        "device_id": 1234567890,  
        "model": "telfmb920",  
        "blocked": false,  
        "tariff_id": 345678,  
        "status_listing_id": null,  
        "creation_date": "2011-09-21",  
        "tariff_end_date": "2016-03-24",  
        "phone": "+71234567890"  
      },  
      "tag_bindings": [  
        {  
          "tag_id": 456789,  
            
```

```
    "ordinal": 4
  }
]
}
]
}
```

For `tracker` object structure, see [tracker/](#).

Errors

[General](#) types only.

raw_command/send

Sends the GPRS command to the device, processing it in a protocol-dependent manner beforehand.

Find more information about this API call usage in our [instructions](#).

required sub-user rights: `tracker_configure` , `tracker_set_output` .

Parameters

name	description	type
<code>tracker_id</code>	ID of the tracker (aka "object_id"). Tracker must belong to authorized user and not be blocked.	int
<code>command</code>	Text or hexadecimal representation of the command.	string
<code>type</code>	Optional. <code>text</code> or <code>hex</code> format. Default is <code>text</code> .	string
<code>reliable</code>	Optional. <code>false</code> if the command does not need to be resent when the device is disconnected or if no acknowledgment is received. Default is <code>true</code> .	boolean

Example

cURL

```
{% code overflow="wrap" %}
```

```
curl -X POST 'https://pro.lepton-seguridad.com/api-v2/tracker/raw_command/send' \  
-H 'Content-Type: application/json' \  
-d '{"hash": "a6aa75587e5c59c32d347da438505fc3", "tracker_id": 265489, "command'
```

{% encode %}

Response

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

Errors

- 7 - Invalid parameters.
- 201 - Not found in the database – if there is no tracker with such device ID belonging to authorized user.

Example response with an error:

```
{  
  "success": false,  
  "status": {  
    "code": 7,  
    "description": "Invalid parameters"  
  },  
  "errors": [  
    {  
      "parameter": "command",  
      "error": "Non-hex string"  
    }  
  ]  
}
```
