CommanderAPI - REST API v1 specification

Index

Summary

This document is a summarization of usage of the Commander REST API. Here you can find all available REST services with examples of requests and responses for every service.

> ⚠ There is no need to call services "vehicles", "vehicles/{vehicleId}" or "deletedVehicles" everytime before calling last-positions or any other service. Service is intended to be used only to read or update vehicle list and it should be called e.g. once a day. You can store current list of vehicles on your server and update it only when necessary. Accounts calling these service all the time can have limited access for the services.

Rate limiting

All requests are rate limited per IP for unauthenticated requests and per company for authenticated requests. After hitting rate limits, server responds with status code 429 - "Too Many Requests" and header "Retry-After" with time until which are limits valid.

Status of current rate limitations and remaining number of requests are returned in response headers **X-RateLimit-Limit**, **X-RateLimit-Remaining** and **X-RateLimit-Reset**.

Header key	Header value
X-RateLimit-Limit	300
X-RateLimit-Remaining	293
X-RateLimit-Reset	1685364715

Services

GET /vehicles - Retrieve a list of all available vehicles

Retrieve a list of all vehicles available for logged API user.

Request

Method	Headers	URL
GET	• Content-Type : application/json • Authorization : Basic Auth	https://online.commander-systems.com/api/v1/vehicles[?vin={vin}]

Optional URL parameters accepted by the service (functionality enabled according to contract):

Parameter name	Data type	Description
vin	String	VIN (Vehicle Identification Number)

Attributes after "deleted" are shown only with special setting (on demand).
Examples of reponse data are displayed with and without these attributes.
Empty values of attributes can have values of: empty string (""), null, or 0. When specified, 0 must be taken as not empty value.
Numeric values must be checked as they can be shown as string or with comma as decimal point

Service do not accept any data in the request body.

Response

In the case of success the response is returned with status 200 and data in JSON format.

Example of a successful response:

Status 200

Response data


```
{
  "vehicles":[
    {
      "vehicleId":"3653622561558",
      "vehicleName":"name of a vehicle",
      "vehicleRegistrationPlate":"BA010AB",
      "vehicleDefaultDriver":12345,
      "lastCommunication":1496744431,
      "vin":"WF0WXXGCD00000001",
      "deleted":"0",
      "serial_number":"123",
      "region":"city",
      "costCenter": "main center",
      "ref": "S",
      "model": "model of a vehicle",
      "manufactureYear": "2022",
      "commissioningDate": "2022-01-01",
      "mainFuelType": "Gasoline",
      "fuelTypeNote": "Only 98 octane",
      "fuelTankCapacity": 40,
      "licencePlateHistory": "BA010AA",
      "objectType": "Personal car",
      "odometerType": 0,
```

```
      "theoreticalConsumption": 5.0,
      "urbanConsumption": 7.5,
      "extraUrbanConsumption": 5.0,
      "combinedEngineHourConsumption": 3.0,
      "combinedKmConsumption": 6.0,
      "costPerKm": 0.123,
      "refuelId1": "abcdefghijk",
      "refuelId2": "abcdefghijk2",
      "depreciation": 0.120,
      "isElectricCar": 0
    },
    {
      "vehicleId":"46535422562345",
      "vehicleName":"name of another vehicle",
      "vehicleRegistrationPlate":"BA020AB",
      "vehicleDefaultDriver":0,
      "lastCommunication":0,
      "deleted":"0"
    }
  ]
}
```


The response contains an array of registered vehicles.

Every vehicle listed in the "vehicles" array consists of:

Parameter name	Data type	Description
vehicleId	String	Unique ID of a vehicle
vehicleName	String	User-friendly name of a vehicle
vehicleRegistrationPlate	String	Registration plate number of a vehicle
vehicleDefaultDriver	Integer	ID of assigned driver in CCC v2 (settings - vehicle - edit vehicle). If ID=0 no driver is assigned.
lastCommunication	Integer	Unix timestamp of last received packet
vin	String	VIN (Vehicle Identification Number) - optional parameter (API account settings dependent)
deleted	Integer	Flag indicating if the vehicle is considered "deleted" (or archived) in the system. String "0" indicates that the vehicle is active, string "1" indicates that the vehicle is deleted.
serial_number	String	Serial number of a vehicle
region	String	Region of a vehicle
costCenter	String	Cost center of vehicle
ref	String	S or R
model	String	model of a vehicle
manufactureYear	String	Manufacture year of a vehicle
commissioningDate	String	Commissioning date of a vehicle
mainFuelType	String	It can be empty or these values: (Gasoline,Diesel,LPG,LNG,CNG,Elektricity,Hydrogen)

fuelTypeNote	String	Note for vehicle fuel type. Can be anything.
fuelTankCapacity	String	Fuel tank capacity in litres
licencePlateHistory	String	History of licence plates
objectType	String	It can be 26 types of vehicles, listed in table below
odometerType	String	0 = kilometers, 1 = engine hours
theoreticalConsumption	String	Theoretical consumption of vehicle
urbanConsumption	String	Consumption in urban areas (litres)
extraUrbanConsumption	String	Consumption outside of urban areas (litres)
combinedEngineHourConsumption	String	Combined consumption for one engine hour (used if vehicle is set to odometerType = 1) (litres)
combinedKmConsumption	String	Combined consumption for 100 km ((used if vehicle is set to odometerType = 0) (litres)
costPerKm	String	Cost per one km in customer's currency (usually € or CZK, not shown here)
refuelId1	String	Identification for auto importing of refuels (usually number of refueling card)
refuelId2	String	Identification for auto importing of refuels (usually number of refueling card)
depreciation	String	Deprecation cost per one km in customer's currency (usually € or CZK, not shown here)
isElectricCar	Integer	0 = vehicle is not electric car, 1 = vehicle is electric car

Example of a response with an error in request processing:

Status	Response data
4xx or 5xx	`{ "status":"error", "message":"error message" }`

The response contains status indicating an error during request processing and a message describing the error.

GET /deletedVehicles- Retrieve a list of all deleted (archived) vehicles

Retrieve a list of all archived vehicles available for logged API user.

Needs to be set up for API user, if not, API returns http 403

Request

Method	Headers	URL
GET	• Content-Type : application/json • Authorization : Basic Auth	https://online.commander-systems.com/api/v1/deletedVehicles

Attributes after "deleted" are shown only with special setting (on demand).

Examples of reponse data are displayed with and without these attributes.

Empty values of attributes can have values of: empty string (""), null, or 0. When specified, 0 must be taken as not empty value.

Numeric values must be checked as they can be shown as string or with comma as decimal point

Service do not accept any data in the request body.

Response

In the case of success the response is returned with status 200 and data in JSON format.

Example of a successful response:

Status	Response data

200	{
	"vehicles":[
	{
	"vehicleId":"3653622561558",
	"vehicleName":"name of a vehicle",
	"vehicleRegistrationPlate":"BA010AB",
	"vehicleDefaultDriver":12345,
	"lastCommunication":1496744431,
	"vin":"WF0WXXGCD00000001",
	"deleted":"1",
	"serial_number":"123",
	"region":"city",
	"costCenter": "main center",
	"ref": "S",
	"model": "model of a vehicle",
	"manufactureYear": "2022",
	"commissioningDate": "2022-01-01",
	"mainFuelType": "Gasoline",
	"fuelTypeNote": "Only 98 octane",
	"fuelTankCapacity": 40,
	"licencePlateHistory": "BA010AA",
	"objectType": "Personal car",
	"odometerType": 0,
	"theoreticalConsumption": 5.0,
	"urbanConsumption": 7.5,
	"extraUrbanConsumption": 5.0,
	"combinedEngineHourConsumption": 3.0,
	"combinedKmConsumption": 6.0,
	"costPerKm": 0.123,
	"refuelId1": "abcdefghijk",
	"refuelId2": "abcdefghijk2",
	"depreciation": 0.120,
	"isElectricCar": 1
	},
	{
	"vehicleId":"46535422562345",
	"vehicleName":"name of another vehicle",
	"vehicleRegistrationPlate":"BA020AB",
	"vehicleDefaultDriver":0,
	"lastCommunication":0,
	"deleted":"1"
	}
	]
	}

The response contains an array of archived vehicles.

Every vehicle listed in the "vehicles" array consists of:

Parameter name	Data type	Description
vehicleId	String	Unique ID of a vehicle
vehicleName	String	User-friendly name of a vehicle
vehicleRegistrationPlate	String	Registration plate number of a vehicle
vehicleDefaultDriver	Integer	ID of assigned driver in CCC v2 (settings - vehicle - edit vehicle). If ID=0 no driver is assigned.
lastCommunication	Integer	Unix timestamp of last received packet
vin	String	VIN (Vehicle Identification Number) - optional parameter (API account settings dependent)
deleted	Integer	Flag indicating if the vehicle is considered "deleted" (or archived) in the system. String "0" indicates that the vehicle is active, string "1" indicates that the vehicle is deleted.
serial_number	String	Serial number of a vehicle
region	String	Region of a vehicle
costCenter	String	Cost center of vehicle
ref	String	S or R
model	String	model of a vehicle
manufactureYear	String	Manufacture year of a vehicle
commissioningDate	String	Commissioning date of a vehicle
mainFuelType	String	It can be empty or these values: (Gasoline,Diesel,LPG,LNG,CNG,Elektricity,Hydrogen)
fuelTypeNote	String	Note for vehicle fuel type. Can be anything.
fuelTankCapacity	String	Fuel tank capacity in litres
licencePlateHistory	String	History of licence plates
objectType	String	It can be 26 types of vehicles, listed in table below
odometerType	String	0 = kilometers, 1 = engine hours
theoreticalConsumption	String	Theoretical consumption of vehicle
urbanConsumption	String	Consumption in urban areas (litres)
extraUrbanConsumption	String	Consumption outside of urban areas (litres)
combinedEngineHourConsumption	String	Combined consumption for one engine hour (used if vehicle is set to odometerType = 1) (litres)
combinedKmConsumption	String	Combined consumption for 100 km ((used if vehicle is set to odometerType = 0) (litres)
costPerKm	String	Cost per one km in customer's currency (usually € or CZK, not shown here)
refuelId1	String	Identification for auto importing of refuels (usually number of refueling card)

refuelId2	String	Identification for auto importing of refuels (usually number of refueling card)
depreciation	String	Deprecation cost per one km in customer's currency (usually € or CZK, not shown here)
isElectricCar	Integer	0 = vehicle is not electric car, 1 = vehicle is electric car

Example of a response with an error in request processing:

Status	Response data

4xx or 5xx	{
	"status":"error",
	"message":"error message"
	}

The response contains status indicating an error during request processing and a message describing the error.

GET /vehicles/{vehicleId} - Retrieve data of a specific vehicle

Retrieve data of a specific vehicle. Vehicle is identified by the <vehicleId> specified in URL

Request

Method	Headers	URL
GET	• Content-Type : application/json • Authorization : Basic Auth	https://online.commander-systems.com/api/v1/vehicles/{vehicleId}

Attributes in response after "deleted" are shown only with special setting (on demand).
Examples of reponse data are displayed with and without these attributes.
Empty values of attributes can have values of: empty string (""), null, or 0. When specified, 0 must be taken as not empty value.
Numeric values must be checked as they can be shown as string or with comma as decimal point

URL parameters accepted by the service (mandatory):

Parameter name	Data type	Description
vehicleId	String (20)	Unique ID of a vehicle

Service do not accept any data in the request body.

Response

In the case of success the response is returned with status 200 and data in JSON format.

Examples of a successful responses:

Response with no additional data

Status	Response data

200	{
	"vehicle":{
	"vehicleId":"3653622561558",

```
    "vehicleName":"name of a vehicle",
    "vehicleRegistrationPlate":"BA010AB",
    "vehicleDefaultDriver":12345
    "lastCommunication":1496744431,
    "vin":"WF0WXXGCD00000001",
    "deleted":"0"
  }
}
```


Response with additional data

Status	Response data

200	{
	"vehicle":{
	"vehicleId":"3653622561558",
	"vehicleName":"name of a vehicle",
	"vehicleRegistrationPlate":"BA010AB",
	"vehicleDefaultDriver":12345
	"lastCommunication":1496744431,
	"vin":"WF0WXXGCD00000001",
	"deleted":"0",
	"serial_number":"123",
	"region":"city",
	"costCenter": "main center",
	"ref": "S",
	"model": "model of a vehicle",
	"manufactureYear": "2022",
	"commissioningDate": "2022-01-01",
	"mainFuelType": "Gasoline",
	"fuelTypeNote": "Only 98 octane",
	"fuelTankCapacity": 40,
	"licencePlateHistory": "BA010AA",
	"objectType": "Personal car",
	"odometerType": 0,
	"theoreticalConsumption": 5.0,
	"urbanConsumption": 7.5,
	"extraUrbanConsumption": 5.0,
	"combinedEngineHourConsumption": 3.0,
	"combinedKmConsumption": 6.0,
	"costPerKm": 0.123,
	"refuelId1": "abcdefghijk",
	"refuelId2": "abcdefghijk2",
	"depreciation": 0.120,
	"isElectricCar": 0
	}
	}

The response contains an object with data of requested vehicle. The object consists of:

Parameter name	Data type	Description
vehicleId	String	Unique ID of a vehicle

vehicleName	String	User-friendly name of a vehicle
vehicleRegistrationPlate	String	Registration plate number of a vehicle
vehicleDefaultDriver	Integer	ID of assigned driver in CCC v2 (settings - vehicle - edit vehicle). If ID=0 no driver is assigned.
lastCommunication	Integer	Unix timestamp of last received packet
vin	String	VIN (Vehicle Identification Number) - optional parameter (API account settings dependent)
deleted	Integer	Flag indicating if the vehicle is considered "deleted" (or archived) in the system. String "0" indicates that the vehicle is active, string "1" indicates that the vehicle is deleted.
serial_number	String	Serial number of a vehicle
region	String	Region of a vehicle
costCenter	String	Cost center of vehicle
ref	String	S or R
model	String	model of a vehicle
manufactureYear	String	Manufacture year of a vehicle
commissioningDate	String	Commissioning date of a vehicle
mainFuelType	String	It can be empty or these values: (Gasoline,Diesel,LPG,LNG,CNG,Elektricity,Hydrogen)
fuelTypeNote	String	Note for vehicle fuel type. Can be anything.
fuelTankCapacity	String	Fuel tank capacity in litres (or in kWh when vehicle is electric car)
licencePlateHistory	String	History of licence plates
objectType	String	It can be 26 types of vehicles, listed in table below
odometerType	String	0 = kilometers, 1 = engine hours
theoreticalConsumption	String	Theoretical consumption of vehicle
urbanConsumption	String	Consumption in urban areas (litres)
extraUrbanConsumption	String	Consumption outside of urban areas (litres)
combinedEngineHourConsumption	String	Combined consumption for one engine hour (used if vehicle is set to odometerType = 1) (litres)
combinedKmConsumption	String	Combined consumption for 100 km ((used if vehicle is set to odometerType = 0) (litres)
costPerKm	String	Cost per one km in customer's currency (usually € or CZK, not shown here)
refuelId1	String	Identification for auto importing of refuels (usually number of refueling card)
refuelId2	String	Identification for auto importing of refuels (usually number of refueling card)
depreciation	String	Deprecation cost per one km in customer's currency (usually € or CZK, not shown here)
isElectricCar	Integer	0 = vehicle is not electric car, 1 = vehicle is electric car

Example of a response with an error in request processing:

Status	Response data

4xx or 5xx	<pre>{
	  "status":"error",
	  "message":"error message"
	}</pre>

The response contains status indicating an error during request processing and a message describing the error.

GET /all-rides - Retrieve a list of all completed rides for logged customer

Retrieve a list of all completed rides for given customer that ended during given time period.

There are two types of rides: "BUSINESS_RIDE" and "PRIVAT_RIDE". The GPS positions and adresses are returned only for "BUSINESS_RIDE".

Returned data are paged (100 records per page by default) and the page number and limit are input parameters (see the URL section in the table below).

Request

Method	Headers	URL
GET	• Content-Type : application/json • Authorization : Basic Auth	https://online.commander-systems.com/api/v1/all-rides?datetimeStart={startTime}&datetimeEnd={endTime}&page={pageNumber}&limit={limit}

URL parameters accepted by the service (optional):

Parameter name	Data type	Description
datetimeStart	Integer	Unix epoch timestamp
datetimeEnd	Integer	Unix epoch timestamp
page	Integer	Number of page that should by returned.
limit	Integer	Record count per page (indicated in response in the "count" atribute). Max. value = 1000
changed	String	Returns only rides created or edited after or at given datetime. Datetime string must be in ISO 8601 standard. Must be enabled by API provider.

Service do not accept any data in the request body.

Response

In the case of success the response is returned with status 200 and data in JSON format.

Attribute "odometerType" is avaible only with special permission.

Example of a successful response:

Status	Response data
200	


```
{
 "rides":[
  {
   "rideId":"856757",
   "rideType":"BUSINESS_RIDE",
   "vehicleId":"987458",
   "odometerType":0,
   "vehicleRegistrationPlate":"BA 010 AB",
   "driverId":"12354",
   "driverName":"Janko Vodič",
   "note":"Jazda do Košíc",
   "startTime":1503391595,
   "stopTime":1503407852,
   "latStart":48.1928,
   "lonStart":17.1409,
   "latStop":48.7047,
   "lonStop":21.2602,
   "startAddress":"Račianska 2430/168, 831 54 Bratislava, Slovensko",
   "stopAddress":"Rastislavova 784/68, 040 01, Košice, Slovensko",
   "avgSpeed":35.9,
   "maxSpeed":137.9,
   "duration":16257,
   "distance":350.23,
   "odometerStart":123876,
   "odometerStop":124226,
   "engineHoursStart":10.56
   "fuelLevelStart":43,
   "fuelLevelStop":12,
   "fuelConsumed":21.81,
   "averageConsumption":6.23,
   "refueling":[
    {
     "datetime":1503392895,
     "odometer":123998.39,
     "volume":41.52,
     "priceVatExc":52.49,
     "priceVatInc":62.99,
     "currency":"EUR"
    },
    {
     "datetime":1503393629,
     "odometer":124382.47,
     "volume":45.39,
     "priceVatExc":59.25,
     "priceVatInc":71.1,
     "currency":"EUR"
    }
   ],
   "waypoints":[
    {
     "waypointId":3520,
     "waypointName":"name_of_waypoint_with_id_3520",
     "waypointTime":1503391595
    },
    {
     "waypointId":183,
     "waypointName":"name_of_waypoint_with_id_183",
```

```
        "waypointTime":1503391600
      }
    ],
   "notes":[
     {
       "noteId":9073,
       "rideId":856757,
       "text":"Dvere nákladového priestoru otvorené",
       "timestamp":1419328781,
       "duration":{
          "value":44,
          "unit":"seconds"
       }
     }
  ],
  "contracts": null
  },
  {
   "rideId":"856758",
   "rideType":"PRIVAT_RIDE",
   "vehicleId":"987458",
   "odometerType":0,
   "vehicleRegistrationPlate":"BA 010 AB",
   "driverId":"12354",
   "driverName":"Janko Vodič",
   "note":"Jazda do Košíc",
   "startTime":1503391595,
   "stopTime":1503407852,
   "latStart":null,
   "lonStart":null,
   "latStop":null,
   "lonStop":null,
   "startAddress":null,
   "stopAddress":null,
   "avgSpeed":35.9,
   "maxSpeed":137.9,
   "duration":16257,
   "distance":350.23,
   "odometerStart":123876,
   "odometerStop":124226,
   "engineHoursStart":11.01
   "fuelLevelStart":43,
   "fuelLevelStop":12,
   "fuelConsumed":0,
   "averageConsumption":0,
   "refueling":[
     {
      "datetime":1503392895,
      "odometer":123998.39,
      "volume":41.52,
      "priceVatExc":52.49,
      "priceVatInc":62.99,
      "currency":"EUR"
     },
     {
      "datetime":1503393629,
      "odometer":null,
      "volume":45.39,
      "priceVatExc":59.25,
      "priceVatInc":71.1,
```

```
        "currency":"EUR"
       }
      ],
      "waypoints":[
       {
        "waypointId":3521,
        "waypointName":"name_of_waypoint_with_id_3521",
        "waypointTime":1503391597
       },
       {
        "waypointId":173,
        "waypointName":"name_of_waypoint_with_id_173",
        "waypointTime":1503391602
       }
      ],
     "notes":null,
     "contracts":[1,2]
    }
  ],
  "page":7,
  "count":100,
  "totalPages":11,
  "totalCount":1050,
  "links":[
   {
    "rel":"prev",
    "href":"https://online.commander-systems.com/api/v1/rides/987458?datetimeStart=1483225200&datetimeEnd=1504705692&page=6"
   },
   {
    "rel":"next",
    "href":"https://online.commander-systems.com/api/v1/rides/987458?datetimeStart=1483225200&datetimeEnd=1504705692&page=8"
   }
  ]
}
```


The response contains an array of rides for specified vehicle and specified time period (only rides that starts in given time period are returned). Only first 100 records are returned (one page). If you need next page of records please use the "page" attribute in the URL or use the "links" section in returned data to access next or previous page.

Paging attributes in the response are listed in the table below

Parameter name	Data type	Description
page	Integer	Returned page number
count	Integer	Number of records per page
totalPages	Integer	Overall number of pages
totalCount	Integer	Overall number of records
links	Array	Array of links that can be used for implementation of paging in GUI or for automatic crawling
links[x].rel	"prev", "next"	Identification of link usage: • prev - previous page with records (not present when page=0) • next - next page with records (not present when page=totalPages-1)

links[x].href	String	URL of specific page with records

Every ride listed in the "rides" array consists of:

Parameter name	Data type	Description
rideId	String	Database ID of this ride
rideType	String	Type of the ride: "BUSINESS_RIDE" or "PRIVATE_RIDE"
vehicleId	String	Database ID of the vehicle used in the ride
odometerType*	Integer	0 = vehicle's odometer is set to "km", 1 = vehicle's odometer is set to engine hours
vehicleRegistrationPlate	String	Registration plate of the vehicle
driverId	String	Database ID of the driver
driverName	String	The name of the driver
note	String	Note about the ride (typically written by the driver)
startTime	Integer	Time and date of the start of the ride in Unix timestamp format
stopTime	Integer	Time and date of the end of the ride
latStart	Float	Latitude of the starting point of the ride in decimal format
lonStart	Float	Longitude of the starting point of the ride in decimal format
latStop	Float	Latitude of the ending point of the ride in decimal format
lonStop	Float	Longitude of the ending point of the ride in decimal format
startAddress	String	Postal address of the starting point of the ride
stopAddress	String	Postal address of the ending point of the ride
avgSpeed	Float	Average speed during the ride in km/h
maxSpeed	Integer	Maximum speed during the ride in km/h (rounded to 0 decimal numbers)
duration	Integer	Duration of the ride in seconds (rounded to 0 decimal numbers)
distance	Float	Distance traveled from the starting point to the ending point in km after correction
odometerStart	Float	The value of the odometer in the starting moment of the ride in km after correction
odometerStop	Float	The value of the odometer in the ending moment of the ride in km after correction
engineHoursStart**	Float	Engine hours of vehicle at the beginning of ride. Engine hours are not always matching stopTime-startTime.
fuelLevelStart	Integer	The level of fuel in the starting moment of the ride in liters (or in kWh when vehicle is electric car), (rounded to 0 decimal numbers)
fuelLevelStop	Integer	The level of fuel in the ending moment of the ride in liters (or in kWh when vehicle is electric car), (rounded to 0 decimal numbers)
fuelConsumed	Float	Consumed liters (or in kWh when vehicle is electric car) from LVCAN during ride (rounded to 2 decimal numbers)
averageConsumption	Float	Average consumption from "fuelConsumed" and "distance" (rounded to 2 decimal numbers)

refueling	Array	Array of all refuelings assigned to the ride (zero or more)
refueling[x].datetime	Integer	Time and date of the refueling in Unix timestamp format
refueling[x].odometer	Float	The value of the odometer in the moment of refueling in km
refueling[x].volume	Float	The fuel volume of the refueling
refueling[x].priceVatExc	Float	The price of the refueling excluding VAT
refueling[x].priceVatInc	Float	The price of the refueling including VAT
refueling[x].currency	String	The currency of the refueling transaction
waypoints	Array	Array of waypoints that were entered or leaved during the ride
waypoints[x].waypointId	String	Internal ID of a waypoint (can be used for waypoint matching from other API services)
waypoints[x].waypointName	String	User friendly identification of a waypoint (exclusivity is not guaranteed and value can be changed by the user)
waypoints[x].waypointTime	Integer	Time and date of waypoint entering in Unix timestamp format
notes	Array/null	Array of notes recorded during the ride (optional and API account settings dependent attribute), or null when notes are allowed but there is no note in ride
notes[x].noteId	Integer	Database ID of ride note
notes[x].rideId	Integer	Database ID of ride
notes[x].text	String	Ride note description
notes[x].timestamp	Integer	Ride note creation unix timestamp
notes[x].duration.value	Integer	Ride note event duration in seconds
notes[x].duration.unit	String	"seconds"
contracts***	Array/null	Array of ID of contracts in ride. When empty, parameter is null

* Available only with special permission

** Engine hours for ride are counted as: engineHoursStart at next ride - engineHoursStart at current ride.
If next ride does not exist, it is counted as: (stopTime-startTime)/3600

*** Available only with special permission

Example of a response with an error in request processing:

Status	Response data

4xx or 5xx	{
	"status":"error",
	"message":"error message"
	}

The response contains status indicating an error during request processing and a message describing the error.

GET /rides/{vehicleId} - Retrieve a list of all completed rides of a vehicle

Retrieve a list of all completed rides of a vehicle that started during given time period.

There are two types of rides: "BUSINESS_RIDE" and "PRIVAT_RIDE". The GPS positions and adresses are returned only for "BUSINESS_RIDE".

Returned data are paged (100 records per page) and the page number is a input parameter (see the URL section in the table below).

Request

Method	Header	URL
GET	• Content-Type : application/json • Authorization : Basic Auth	https://online.commander-systems.com/api/v1/rides/{vehicleId}?datetimeStart={startTime}&datetimeEnd={endTime}&page={pageNumber}

URL parameters accepted by the service (mandatory):

Parameter name	Data type	Description
vehicleId	String	Unique ID of a vehicle
datetimeStart	Integer	Unix epoch timestamp
datetimeEnd	Integer	Unix epoch timestamp
page	Integer	Number of page that should by returned. One page contains 100 records (indicated in response in the "count" atribute)

Service do not accept any data in the request body.

Response

In the case of success the response is returned with status 200 and data in JSON format.

Attribute "odometerType" is avaible only with special permission.

Example of a successful response:

Status	Response data
200	

```
{
  "rides":[
    {
      "rideId":"856757",
      "rideType":"BUSINESS_RIDE",
      "vehicleId":"987458",
      "odometerType":0,
      "vehicleRegistrationPlate":"BA 010 AB",
      "driverId":"12354",
      "driverName":"Janko Vodič",
      "note":"Jazda do Košíc",
      "startTime":1503391595,
      "stopTime":1503407852,
      "latStart":48.1928,
      "lonStart":17.1409,
      "latStop":48.7047,
```

```
        "lonStop":21.2602,
        "startAddress":"Račianska 2430/168, 831 54 Bratislava, Slovensko",
        "stopAddress":"Rastislavova 784/68, 040 01, Košice, Slovensko",
        "avgSpeed":35.9,
        "maxSpeed":137.9,
        "duration":16257,
        "distance":350.23,
        "odometerStart":123876,
        "odometerStop":124226,
        "engineHoursStart":10.56
        "fuelLevelStart":43,
        "fuelLevelStop":12,
        "fuelConsumed":21.81,
        "averageConsumption":6.23,
        "refueling":[
          {
            "datetime":1503392895,
            "odometer":123998.39,
            "volume":41.52,
            "priceVatExc":52.49,
            "priceVatInc":62.99,
            "currency":"EUR"
          },
          {
            "datetime":1503393629,
            "odometer":124382.47,
            "volume":45.39,
            "priceVatExc":59.25,
            "priceVatInc":71.1,
            "currency":"EUR"
          }
        ],
        "waypoints":[
          {
            "waypointId":3520,
            "waypointName":"name_of_waypoint_with_id_3520",
            "waypointTime":1503391595
          },
          {
            "waypointId":183,
            "waypointName":"name_of_waypoint_with_id_183",
            "waypointTime":1503391599
          }
        ],
        "notes":[
          {
            "noteId":9073,
            "rideId":5345,
            "text":"Dvere nákladového priestoru otvorené",
            "timestamp":1419328781,
            "duration":{
              "value":44,
              "unit":"seconds"
            }
          }
        ],
        "contracts":null
      },
      {
        "rideId":"856758",
```

```
      "rideType":"PRIVAT_RIDE",
      "vehicleId":"987458",
      "odometerType":0,
      "vehicleRegistrationPlate":"BA 010 AB",
      "driverId":"12354",
      "driverName":"Janko Vodič",
      "note":"Jazda do Košíc",
      "startTime":1503391595,
      "stopTime":1503407852,
      "latStart":null,
      "lonStart":null,
      "latStop":null,
      "lonStop":null,
      "startAddress":null,
      "stopAddress":null,
      "avgSpeed":35.9,
      "maxSpeed":137.9,
      "duration":16257,
      "distance":350.23,
      "odometerStart":123876,
      "odometerStop":124226,
      "engineHoursStart":11.01
      "fuelLevelStart":43,
      "fuelLevelStop":12,
      "fuelConsumed":0,
      "averageConsumption":0,
      "refueling":[
        {
          "datetime":1503392895,
          "odometer":123998.39,
          "volume":41.52,
          "priceVatExc":52.49,
          "priceVatInc":62.99,
          "currency":"EUR"
        },
        {
          "datetime":1503393629,
          "odometer":null,
          "volume":45.39,
          "priceVatExc":59.25,
          "priceVatInc":71.1,
          "currency":"EUR"
        }
      ],
      "waypoints":[
        {
          "waypointId":3521,
          "waypointName":"name_of_waypoint_with_id_3521",
          "waypointTime":1503391596
        },
        {
          "waypointId":173,
          "waypointName":"name_of_waypoint_with_id_173",
          "waypointTime":1503391601
        }
      ],
      "notes":null,
      "contracts":[1,2]
    }
  ],
```

```
  "page":7,
  "count":100,
  "totalPages":11,
  "totalCount":1050,
  "links":[
    {
      "rel":"prev",
      "href":"https://online.commander-systems.com/api/v1/rides/987458?datetimeStart=1483225200&datetimeEnd=1504705692&page=6"
    },
    {
      "rel":"next",
      "href":"https://online.commander-systems.com/api/v1/rides/987458?datetimeStart=1483225200&datetimeEnd=1504705692&page=8"
    }
  ]
}
```


The response contains an array of rides for specified vehicle and specified time period (only rides that starts in given time period are returned). Only first 100 records are returned (one page). If you need next page of records please use the "page" attribute in the URL or use the "links" section in returned data to access next or previous page.

Paging attributes in the response are listed in the table below

Parameter name	Data type	Description
page	Integer	Returned page number
count	Integer	Number of records per page
totalPages	Integer	Overall number of pages
totalCount	Integer	Overall number of records
links	Array	Array of links that can be used for implementation of paging in GUI or for automatic crawling
links[x].rel	"prev", "next"	Identification of link usage: • prev - previous page with records (not present when page=0) • next - next page with records (not present when page=totalPages-1)
links[x].href	String	URL of specific page with records

Every ride listed in the "rides" array consists of:

Parameter name	Data type	Description
rideId	String	Database ID of this ride
rideType	String	Type of the ride: "BUSINESS_RIDE" or "PRIVATE_RIDE"
vehicleId	String	Database ID of the vehicle used in the ride
odometerType*	Int	0 = vehicle's odometer is set to "km", 1 = vehicle's odometer is set to engine hours
vehicleRegistrationPlate	String	Registration plate of the vehicle
driverId	String	Database ID of the driver
driverName	String	The name of the driver

note	String	Note about the ride (typically written by the driver)
startTime	Integer	Time and date of the start of the ride in Unix timestamp format
stopTime	Integer	Time and date of the end of the ride
latStart	Float	Latitude of the starting point of the ride in decimal format
lonStart	Float	Longitude of the starting point of the ride in decimal format
latStop	Float	Latitude of the ending point of the ride in decimal format
lonStop	Float	Longitude of the ending point of the ride in decimal format
startAddress	String	Postal address of the starting point of the ride
stopAddress	String	Postal address of the ending point of the ride
avgSpeed	Float	Average speed during the ride in km/h
maxSpeed	Integer	Maximum speed during the ride in km/h (rounded to 0 decimal numbers)
duration	Integer	Duration of the ride in seconds (rounded to 0 decimal numbers)
distance	Float	Distance traveled from the starting point to the ending point in km after correction
odometerStart	Float	The value of the odometer in the starting moment of the ride in km after correction
odometerStop	Float	The value of the odometer in the ending moment of the ride in km after correction
engineHoursStart**	Float	Engine hours of vehicle at the beginning of ride. Engine hours are not always matching stopTime-startTime.
fuelLevelStart	Integer	The level of fuel in the starting moment of the ride in liters (or in kWh when vehicle is electric car), (rounded to 0 decimal numbers)
fuelLevelStop	Integer	The level of fuel in the ending moment of the ride in liters (or in kWh when vehicle is electric car), (rounded to 0 decimal numbers)
fuelConsumed	Float	Consumed liters (or in kWh when vehicle is electric car) from LVCAN during ride (rounded to 2 decimal numbers)
averageConsumption	Float	Average consumption from "fuelConsumed" and "distance" (rounded to 2 decimal numbers)
refueling	Array	Array of all refuelings assigned to the ride (zero or more)
refueling[x].datetime	Integer	Time and date of the refueling in Unix timestamp format
refueling[x].odometer	Float	The value of the odometer in the moment of refueling in km
refueling[x].volume	Float	The fuel volume of the refueling
refueling[x].priceVatExc	Float	The price of the refueling excluding VAT
refueling[x].priceVatInc	Float	The price of the refueling including VAT
refueling[x].currency	String	The currency of the refueling transaction
waypoints	Array	Array of waypoints that were entered or leaved during the ride
waypoints[x].waypointId	String	Internal ID of a waypoint (can be used for waypoint matching from other API services)

waypoints[x].waypointName	String	User friendly identification of a waypoint (exclusivity is not guaranteed and value can be changed by the user)
waypoints[x].waypointTime	Integer	Time and date of waypoint entering in Unix timestamp format
notes	Array	Array of notes recorded during the ride (optional and API account settings dependent attribute)
notes[x].noteId	Integer	Database ID of ride note
notes[x].rideId	Integer	Database ID of ride
notes[x].text	String	Ride note description
notes[x].timestamp	Integer	Ride note creation unix timestamp
notes[x].duration.value	Integer	Ride note event duration in seconds
notes[x].duration.unit	String	"seconds"
contracts***	Array/null	Array of ID of contracts in ride. When empty, parameter is null

*Avaible only with special permission

**Engine hours for ride are counted as: engineHoursStart at next ride - engineHoursStart at current ride.
If next ride does not exist, it is counted as: (stopTime-startTime)/3600

*** Available only with special permission

Example of a response with an error in request processing:

Status	Response data

4xx or 5xx	{
	"status":"error",
	"message":"error message"
	}

The response contains status indicating an error during request processing and a message describing the error.

GET /waypoints - Retrieve a list of all available waypoints

Retrieve a list of all waypoints available for logged API user.

Request

Method	Header	URL
GET	• Content-Type : application/json • Authorization : Basic Auth	https://online.commander-systems.com/api/v1/waypoints

Service do not accept any data in the request body.

Response

In the case of success the response is returned with status 200 and data in JSON format.

Example of a successful response:

Status	Response data
200
```
{
 "waypoints":[
  {
   "waypointId":"123",
   "waypointName":"name of this waypoint",
   "waypointPolygon":[
    {
     "lon":14.390626,
     "lat":50.076788
    },
    {
     "lon":14.390755,
     "lat":50.076213
    },
    {
     "lon":14.390875,
     "lat":50.073421
    }
   ]
  },
  {
   "waypointId":"223",
   "waypointName":"name of another waypoint",
   "waypointPolygon":[
    {
     "lon":14.396532,
     "lat":50.077453
    },
    {
     "lon":14.398755,
     "lat":50.076234
    },
    {
     "lon":14.398421,
     "lat":50.077144
    }
   ]
  }
 ]
}
```
 |

The response contains an array of all available waypoints.

Every waypoint listed in the "waypoints" array consists of:

Parameter name	Data type	Description
waypointId	String	Internal ID of a waypoint (can be used for waypoint matching from other API services)
waypointName	String	User friendly identification of a waypoint (exclusivity is not guaranteed and value can be changed by the user)
waypointPolygon	Array	List of geographical points or "nodes" of the waypoint polygon. These points are listed in order in which they are connected geographically - they form a cyclic path.
waypointPolygon[x].lat	Double	Latitude of this polygon point

waypointPolygon[x].lon	Double	Longitude of this polygon point

Example of a response with an error in request processing:

Status	Response data

4xx or 5xx	{
	"status":"error",
	"message":"error message"
	}

The response contains status indicating an error during request processing and a message describing the error.

GET /waypoint-groups - Retrieve a list of all available waypoint groups

Retrieve a list of all waypoint groups available for logged API user.

Request

Method	Header	URL
GET	• Content-Type : application/json • Authorization : Basic Auth	https://online.commander-systems.com/api/v1/waypoint-groups

Service do not accept any data in the request body.

Response

In the case of success the response is returned with status 200 and data in JSON format.

Example of a successful response:

Status	Response data

200	[
	{
	"waypointGroupID": 123123,
	"waypointGroupName": "Test group"
	},
	{
	"waypointGroupID": 123124,
	"waypointGroupName": "Test group2"
	}
	]

The response contains an array of all available waypoints.

Every waypoint group listed in the array consists of:

Parameter name	Data type	Description
waypointGroupID	String	Internal ID of a waypoint group

waypointGroupName	String	User friendly identification of a waypoint (exclusivity is not guaranteed and value can be changed by the user)

Example of a response with an error in request processing:

Status	Response data

4xx or 5xx	<pre>{
	"status":"error",
	"message":"error message"
	}</pre>

The response contains status indicating an error during request processing and a message describing the error.

GET /last-positions - Retrieve a list of all vehicles with their last position

Retrieve a list of all vehicles with assigned gps unit with their last gps position

Request

Method	Header	URL
GET	• Content-Type : application/json • Authorization : Basic Auth	https://online.commander-systems.com/api/v1/last-positions

URL parameters accepted by the service:

Parameter name	Data type	Description
page	Integer	Number of page that should by returned.
limit	Integer	Record count per page (indicated in response in the "count" atribute). Max. value = 1000 When addresses are allowed, max limit is set to 100

Service do not accept any data in the request body.

Response

In the case of success the response is returned with status 200 and data in JSON format.

Example of a successful response:

Status	Response data

200	<pre>{
	"positions": [
	{
	"vehicleId": 123456,
	"gpsTime": 1563957600,</pre>

```
    "gpsLat": 49.208797,
    "gpsLon": 16.154687,
    "gpsLAlt": 200,
    "gpsAzimut": 10,
    "gpsSpeed": 56,
    "carIgnition": 1,
    "voltage": 14.4,
    "canSpeed": 60,
    "canThrottle": 20,
    "canConsumed": 1256,
    "canTankValue": 50,
    "canRpm": 2500,
    "canEngineHours": 100,
    "canOdometer": 5000,
    "temperatures":[
    ]
   },
   {
    "vehicleId": 123457,
    "gpsTime": 1563952312,
    "gpsLat": 48.598746,
    "gpsLon": 18.12569,
    "gpsLAlt": 10,
    "gpsAzimut": 0,
    "gpsSpeed": 0,
    "carIgnition": 0,
    "voltage": 12.0,
    "canSpeed": 0,
    "canThrottle": 0,
    "canConsumed": 0,
    "canTankValue": 0,
    "canRpm": 0,
    "canEngineHours": 0,
    "canOdometer": 0,
    "temperatures":{"temperature":6.8,"fridge":7,"freezer":""}
   }
  ],
  "page": 1,
  "count": 2,
  "totalPages": 1,
  "totalCount": 2
}
```


The response contains an array of last gps positions.

Every position listed in the "positions" array consists of:

Parameter name	Data type	Description
vehicleId	Integer	Database ID of the vehicle
gpsTime	Integer	Time and date of the gps position
gpsLat	Float	Latitude of the gps position
gpsLon	Float	Longitude of the gps position
gpsLAlt	Integer	Altitude of vehicle
gpsAzimut	Integer	Azimut of vehicle

gpsSpeed	Integer	Vehicle speed from gps at gpsTime
carIgnition	Integer	1 = ignition on, 0 = ignition off
voltage	Float	Voltage of battery connected to gps unit
canSpeed	Float	Vehicle speed from CANBUS
canThrottle	Float	Vehicle throttle level (0-100%) from CANBUS
canConsumed	Float	Sum of consumed fuel from CANBUS (or in kWh when vehicle is electric car)
canTankValue	Float	Value of fuel tank in litres from CANBUS (or in kWh when vehicle is electric car)
canRpm	Float	Value of engine RPM from CANBUS
canEngineHours	Float	Value of engine hours from CANUBS
canOdometer	Float	Value of odometer from CANBUS
temperatures	Array	List of available temperatures for vehicle. If temperature is not sent, value is empty string ""
address	String	Must be allowed for API user by special user right. Address of position. If carIgnition = 0, returns address from last ride. If carIgnition = 1, returns current address. Max limit for returned positions is set to 100.

CANBUS values are >0 only if LVCAN is installed in vehicle and value is supported.

Any value from CANBUS can be 0 when ignition is off!

Example of a response with an error in request processing:

Status	Response data

4xx or 5xx	{
	"status":"error",
	"message":"error message"
	}

The response contains status indicating an error during request processing and a message describing the error.

GET /drivers- Retrieve a list of all active drivers

Retrieve a list of all active drivers under Customer. List does not include deleted drivers.

Request

Method	Header	URL
GET	• Content-Type : application/json • Authorization : Basic Auth	https://online.commander-systems.com/api/v1/drivers

URL parameters accepted by the service:

Parameter name	Data type	Description

page	Integer	Number of page that should by returned.
limit	Integer	Record count per page (indicated in response in the "count" atribute). Max. value = 200

Service do not accept any data in the request body.

Response

In the case of success the response is returned with status 200 and data in JSON format.

Example of a successful response:

Status	Response data

200	{
	"drivers": [
	{
	"id": 123456,
	"name": "Name",
	"surname": "Surname",
	"personal_number": "0002",
	"identification_keys": [
	{
	"code": "123456789",
	"type": 0,
	"name": "ABCDEF"
	},
	{
	"code": "987654321",
	"type": 1,
	"name": "ABC12"
	}
	],
	"cost_center_text":"",
	"cost_center_id":0
	},
	{
	"id": 123457,
	"name": "Name2",
	"surname": "Surname2",
	"personal_number": "",
	"identification_keys": [],
	"cost_center_text":"Cost center 123",
	"cost_center_id":5
	}
	],
	"page": 1,
	"count": 2,
	"totalPages": 1,
	"totalCount": 2
	}

The response contains an array of last gps positions.

Every position listed in the "positions" array consists of:

Parameter name	Data type	Description
id	Integer	Database ID of the driver
name	String	Name of driver (can include titles)
surname	String	Surname of driver
personal_number	String	Personal number of driver
identification_keys	Array	Array of identification keys
identification_keys[x].code	String	Usually hexa code of identification key, uppercase
identification_keys[x].type	Integer	0 = dallas, 1 = rfid, 2 = tachograph
identification_keys[x].name	String	Name of identification key assigned to driver
cost_center_text	String	Name of cost center, can be different from cost center in “cost_center_id”
cost_center_id	Integer	ID of cost center from list of cost centers. 0 = empty cost center

Example of a response with an error in request processing:

Status	Response data

4xx or 5xx	{
	"status":"error",
	"message":"error message"
	}

The response contains status indicating an error during request processing and a message describing the error.

GET /refueling-import- Retrieve a list of fuel import

Retrieve a list of filtered reufeling imports.

Service must be allowed with special user right.

Request

Method	Header	URL
GET	• Content-Type : application/json • Authorization : Basic Auth	https://online.commander-systems.com/api/v1/refueling-import

URL parameters accepted by the service:

Parameter name	Data type	Description	Mandatory
dateStart	String/date	Date format YYYY-MM-DD	yes
dateEnd	String/date	Date format YYYY-MM-DD	yes

filter	Integer	0 = all, 1 = processed, 2 = not processed. If not sent, default filter value is 0	no
page	Integer	Number of page that should by returned.	no
limit	Integer	Record count per page (indicated in response in the "count" atribute). Max. value = 100	no

Service do not accept any data in the request body.

Response

In the case of success the response is returned with status 200 and data in JSON format.

Example of a successful response:

Status	Response data

200	{
	"imports": [
	{
	"importTime": 1644648245,
	"importValue": 40.02,
	"importCost": 64.32,
	"importCostVat": 76.84,
	"importCostPerLiter": 1.60,
	"importCostPerLiterVat": 1.92,
	"importVat": 20,
	"importObjectIdentifier": "AA000AA",
	"importStatus": 1,
	"importObjectId": 0,
	"importRideId": 0
	}
	],
	"page": 1,
	"count": 1,
	"totalPages": 1,
	"totalCount": 1
	}

The response contains an array of refueling imports

Every refueling import listed in the "imports" array consists of:

Parameter name	Data type	Description
importTime	Integer	Unix timestamp of refueling import
importValue	Float	Value in litres
importCost	Float	Total cost without VAT
importCostVat	Float	Total cost with VAT
importCostPerLiter	Float	Cost per liter without VAT
importCostPerLiterVat	Float	Cost per liter with VAT
importVat	Integer	VAT in %

importObjectIdentifier	String	Identifier of vehicle, can be licence plate or card number
importStatus	Integer	1 = processed, 2 = not processed
importObjectId	Integer	ID of object for processed or unprocessed item
importRideId	Integer	ID of ride for processed item

Example of a response with an error in request processing:

Status	Response data
4xx or 5xx	{ "status":"error", "message":"error message" }

The response contains status indicating an error during request processing and a message describing the error.

All vehicle object types (objectType in GET /vehicles | /vehicles/{vehicleId} | /deletedVehicles)

List of all possible vehicle types. More vehicle types can be added later.

- Personal car
- Van
- Truck
- Person
- Working machine
- Drive simulator
- Motorcycle
- Trailer
- Train
- Helicopter
- Ship
- Bicycle
- Plane
- Bus
- Snow plow
- Excavator
- Bulldozer
- Roller
- Loader
- Truck 3 axles
- Truck 4 axles
- Semitruck
- Electric car
- Wood export
- Harvestor

- Container

GET /contracts - Retrieve a list of contracts

Retrieve a list of all active (not deleted) contracts.

Service must be allowed with special user right.

Request

Method	Header	URL
GET	• Content-Type : application/json • Authorization : Basic Auth	https://online.commander-systems.com/api/v1/contracts

URL parameters accepted by the service:

Parameter name	Data type	Description	Mandatory
page	Integer	Number of page that should by returned.	no
limit	Integer	Record count per page (indicated in response in the "count" atribute). Max. value = 100	no

Service do not accept any data in the request body.

Response

In the case of success the response is returned with status 200 and data in JSON format.

Example of a successful response:

Status	Response data

200	{
	"contracts": [
	{
	"contractId": 1,
	"contractName": "Contract 1"
	},
	{
	"contractId": 1,
	"contractName": "Contract 2"
	}
	],
	"page": 1,
	"count": 2,
	"totalPages": 1,
	"totalCount": 2
	}

The response contains an array of all contracts

Every contract listed in the "contracts" array consists of:

Parameter name	Data type	Description
contractId	Integer	ID of contract
contractName	String	Name of contract

Example of a response with an error in request processing:

Status	Response data

4xx or 5xx	<pre>{
	"status":"error",
	"message":"error message"
	}</pre>

The response contains status indicating an error during request processing and a message describing the error.

GET /cost-centers - Retrieve a list of cost centers for driver

Retrieve a list of all cost centers.

Request

Method	Header	URL
GET	• Content-Type : application/json • Authorization : Basic Auth	https://online.commander-systems.com/api/v1/cost-centers

URL parameters accepted by the service:

Parameter name	Data type	Description	Mandatory
page	Integer	Number of page that should by returned.	no
limit	Integer	Record count per page (indicated in response in the "count" atribute). Max. value = 100	no

Service do not accept any data in the request body.

Response

In the case of success the response is returned with status 200 and data in JSON format.

Example of a successful response:

Status	Response data

200	<pre>{
	"costCenters": [
	{
	"costCenterId": 5,</pre>

```
      "costCenterNumber": "001",
      "costCenterName": "Office",
      "costCenterDeleted": 0

    },
    {
      "costCenterId": 6,
      "costCenterNumber": "002",
      "costCenterName": "Office 2",
      "costCenterDeleted": 1
    }
  ],
  "page": 1,
  "count": 2,
  "totalPages": 1,
  "totalCount": 2
}
```


The response contains an array of all cost centers

Every cost center listed in the "costCenters" array consists of:

Parameter name	Data type	Description
costCenterId	Integer	ID of cost center
costCenterNumber	String	Number of cost center
costCenterName	String	Name of cost center
costCenterDeleted	Integer	1 = deleted cost center, 0 = active cost center

Example of a response with an error in request processing:

Status	Response data
4xx or 5xx	`{ "status":"error", "message":"error message" }`

The response contains status indicating an error during request processing and a message describing the error.

GET /current-tacho/{vehicle_id} - Retrieve current state of tachometer and engine hours

Request

Method	Header	URL
GET	• Content-Type : application/json • Authorization : Basic Auth	https://online.commander-systems.com/api/v1/current-tacho/{vehicle_id}

Service do not accept any data in the request body.

Response

In the case of success the response is returned with status 200 and data in JSON format.

Example of a successful response:

Status	Response data
200	`{"currentTacho":{"km":12569,"engine_hours":20.15}}`

Parameter name	Data type	Description
km	Float/ Integer	Returns current state of tachometer in kilometers, rounded to two decimal numbers. Can return integer when there is no decimal part.
engineHours	Float/ Integer	Returns current state of tachometer in engine hours, rounded to two decimal numbers. Can return integer when there is no decimal part.

Example of a response with an error in request processing:

Status	Response data
4xx or 5xx	`{ "status":"error", "message":"error message" }`

The response contains status indicating an error during request processing and a message describing the error.