CommanderAPI - REST API v1 specification

Index

- Summary
- Rate limiting
- Services
 - GET /vehicles Retrieve a list of all available vehicles
 - GET /deleted Vehicles- Retrieve a list of all deleted (archived) vehicles
 - GET /vehicles/{vehicleId} Retrieve data of a specific vehicle
 - GET /all-rides Retrieve a list of all completed rides for logged customer
 - GET /rides/{vehicleId} Retrieve a list of all completed rides of a vehicle
 - GET /waypoints Retrieve a list of all available waypoints
 - GET /waypoint-groups Retrieve a list of all available waypoint groups
 - GET /last-positions Retrieve a list of all vehicles with their last position
 - GET /drivers- Retrieve a list of all active drivers
 - GET /refueling-import- Retrieve a list of fuel import
 - All vehicle object types (objectType in GET /vehicles | /vehicles/{vehicleId} | /deletedVehicles)
 - GET /contracts Retrieve a list of contracts
 - GET /cost-centers Retrieve a list of cost centers for driver
 - GET /current-tacho/{vehicle_id} Retrieve current state of tachometer and engine hours

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.

A 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/jsonAuthorization : 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.

Status	Res	ponse data
200	1	{
	2	"vehicles":[
	3	{
	4	"vehicleId":"3653622561558",
	5	"vehicleName":"name of a vehicle",
	6	"vehicleRegistrationPlate":"BA010AB",
	7	"vehicleDefaultDriver":12345,
	8	"lastCommunication":1496744431,
	9	"vin":"WF0WXXGCD0000001",
	10	"deleted":"0",
	11	"serial_number":"123",
	12	"region":"city",
	13	"costCenter": "main center",
	14	"ref": "S",
	15	"model": "model of a vehicle",
	16	"manufactureYear": "2022",
	17	"commissioningDate": "2022-01-01",
	18	"mainFuelType": "Gasoline",
	19	"fuelTypeNote": "Only 98 octane",
	20	"fuelTankCapacity": 40,
	21	"licencePlateHistory": "BA010AA",
	22	"objectType": "Personal car",
	23	"odometerType": 0,

24	"theoreticalConsumption": 5.0,
25	"urbanConsumption": 7.5,
26	"extraUrbanConsumption": 5.0,
27	"combinedEngineHourConsumption": 3.0,
28	"combinedKmConsumption": 6.0,
29	"costPerKm": 0.123,
30	"refuelId1": "abcdefghijk",
31	"refuelId2": "abcdefghijk2",
32	"depreciation": 0.120,
33	"isElectricCar": 0
34	},
35	{
36	"vehicleId":"46535422562345",
37	"vehicleName":"name of another vehicle",
38	"vehicleRegistrationPlate":"BA020AB",
39	"vehicleDefaultDriver":0,
40	"lastCommunication":0,
41	"deleted":"0"
42	}
43]
44	}

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
	1	

Status	Response data	
4xx or 5xx	1 2 3 4	{ "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/jsonAuthorization : 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.

Status	Resp	ponse data
200	1	{
	2	"vehicles":[
	3	{
	4	"vehicleId":"3653622561558",
	5	"vehicleName":"name of a vehicle",
	6	"vehicleRegistrationPlate":"BA010AB",
	7	"vehicleDefaultDriver":12345,
	8	"lastCommunication":1496744431,
	9	"vin":"WF0WXXGCD00000001",
	10	"deleted":"1",
	11	"serial_number":"123",
	12	"region":"city",
	13	"costCenter": "main center",
	14	"ref": "S",
	15	"model": "model of a vehicle",
	16	"manufactureYear": "2022",
	17	"commissioningDate": "2022-01-01",
	18	"mainFuelType": "Gasoline",
	19	"fuelTypeNote": "Only 98 octane",
	20	"fuelTankCapacity": 40,
	21	"licencePlateHistory": "BA010AA",
	22	"objectType": "Personal car",
	23	"odometerType": 0,
	24	"theoreticalConsumption": 5.0,
	25	"urbanConsumption": 7.5,
	26	"extraUrbanConsumption": 5.0,
	27	"combinedEngineHourConsumption": 3.0,
	28	"combinedKmConsumption": 6.0,
	29	"costPerKm": 0.123,
	30 31	"refuelId1": "abcdefghijk", "refuelId2": "abcdefghijk2",
	32	"depreciation": 0.120,
	33	"isElectricCar": 1
	34),
	35	{
	36	"vehicleId":"46535422562345",
	37	"vehicleName":"name of another vehicle",
	38	"vehicleRegistrationPlate":"BA020AB",
	39	"vehicleDefaultDriver":0,
	40	"lastCommunication":0,
	41	"deleted":"1"
	42	}
	43]
	44	}

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

Status	Response data	
4xx or 5xx	<pre>1 { 2 "status":"error", 3 "message":"error message" 4 }</pre>	

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/jsonAuthorization : 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	1 {
	2 "vehicle":{
	3 "vehicleId":"3653622561558",

4	"vehicleName":"name of a vehicle",
5	"vehicleRegistrationPlate":"BA010AB",
6	"vehicleDefaultDriver":12345
7	"lastCommunication":1496744431,
8	"vin":"WF0WXXGCD0000001",
9	"deleted":"0"
10	}
11	}

Response with additional data

Status	Resp	oonse data
200	1	{
	2	"vehicle":{
	3	"vehicleId":"3653622561558",
	4	"vehicleName":"name of a vehicle",
	5	"vehicleRegistrationPlate":"BA010AB",
	6	"vehicleDefaultDriver":12345
	7	"lastCommunication":1496744431,
	8	"vin":"WF0WXXGCD00000001",
	9	"deleted":"0",
	10	"serial_number":"123",
	11	"region":"city",
	12	"costCenter": "main center",
	13	"ref": "S",
	14	"model": "model of a vehicle",
	15	"manufacture Year": "2022",
	16	"commissioningDate": "2022-01-01",
	17	"mainFuelType": "Gasoline",
	18	"fuelTypeNote": "Only 98 octane",
	19	"fuelTankCapacity": 40,
	20	"licencePlateHistory": "BA010AA",
	21	"objectType": "Personal car",
	22	"odometerType": 0,
	23	"theoreticalConsumption": 5.0,
	24	"urbanConsumption": 7.5,
	25	"extraUrbanConsumption": 5.0,
	26	"combinedEngineHourConsumption": 3.0,
	27	"combinedKmConsumption": 6.0,
	28	"costPerKm": 0.123,
	29	"refuelId1": "abcdefghijk",
	30	"refuelId2": "abcdefghijk2",
	31	"depreciation": 0.120,
	32	"isElectricCar": 0
	33	}
	34	}

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

	1	
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)
combinedEngineHourConsumpti on	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

Status	Response data	
4xx or 5xx	1 2 3 4	{ "status":"error", "message":"error message" }

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.

i Attribute "odometerType" is avaible only with special permission.

Status	Resp	onse data
200	1	{
	2	"rides":[
	3	{
	4	"rideId":"856757",
	5	"rideType":"BUSINESS_RIDE",
	6	"vehicleId":"987458",
	7	"odometerType":0,
	8	"vehicleRegistrationPlate":"BA 010 AB",
	9	"driverId":"12354",
	10	"driverName":"Janko Vodič",
	11	"note":"Jazda do Košíc",
	12	"startTime":1503391595,
	13	"stopTime":1503407852,
	14	"latStart":48.1928,
	15	"lonStart":17.1409,
	16	"latStop":48.7047,
	17	"lonStop":21.2602,
	18	"startAddress":"Račianska 2430/168, 831 54 Bratislava, Slovensko",
	19	"stopAddress":"Rastislavova 784/68, 040 01, Košice, Slovensko",
	20	"avgSpeed":35.9,
	21	"maxSpeed":137.9,
	22	"duration":16257,
	23	"distance":350.23,
	24	"odometerStart":123876,
	25	"odometerStop":124226,
	26	"engineHoursStart":10.56
	27	"fuelLevelStart":43,
	28	"fuelLevelStop":12,
	29	"fuelConsumed":21.81,
	30	"averageConsumption":6.23,
	31	"refueling":[
	32	{
	33	"datetime":1503392895,
	34	"odometer":123998.39,
	35	"volume":41.52,
	36	"priceVatExc":52.49,
	37	"priceVatInc":62.99,
	38	"currency":"EUR"
	39	},
	40	{
	41	"datetime":1503393629,
	42	"odometer":124382.47,
	43	"volume":45.39,
	44	"priceVatExc":59.25,
	45	"priceVatInc":71.1,
	46	"currency":"EUR"
	47	}
	48],
	49	"waypoints":[
	50	{
	51	"waypointId":3520,
	52	"waypointName":"name_of_waypoint_with_id_3520",
	53	"waypointTime":1503391595
	54	},
	55	{
	56	"waypointId":183,
	57	"waypointName":"name_of_waypoint_with_id_183",
	0,	

58	"waypointTime":1503391600
59	}
60],
61	"notes":[
62	{
63	"noteId":9073,
64	"rideId":856757,
65	"text":"Dvere nákladového priestoru otvorené",
66	"timestamp":1419328781,
67	"duration":{
68	"value":44,
69	"unit":"seconds"
70	}
71	}
72],
73	"contracts": null
74	},
75	{
76	"rideId":"856758",
77	"rideType":"PRIVAT_RIDE",
78	"vehicleId":"987458",
79	"odometerType":0,
80 81	"vehicleRegistrationPlate":"BA 010 AB", "driverId":"12354",
82	"driverName":"Janko Vodič",
83	"note":"Jazda do Košíc",
84	"startTime":1503391595,
85	"stopTime":1503407852,
86	"latStart": null ,
87	"lonStart":null,
88	"latStop":null,
89	"lonStop":null,
90	"startAddress": null ,
91	"stopAddress": null ,
92	"avgSpeed":35.9,
93	"maxSpeed":137.9,
94	"duration":16257,
95	"distance":350.23,
96	"odometerStart":123876,
97	"odometerStop":124226,
98	"engineHoursStart":11.01
99	"fuelLevelStart":43,
100 101	"fuelLevelStop":12, "fuelConsumed":0,
101	"averageConsumption":0,
102	"refueling":[
103	{
105	"datetime":1503392895,
106	"odometer":123998.39,
107	"volume":41.52,
108	"priceVatExc":52.49,
109	"priceVatInc":62.99,
110	"currency":"EUR"
111	},
112	{
113	"datetime":1503393629,
114	"odometer": null,
115	"volume":45.39,
116	"priceVatExc":59.25,
117	"priceVatInc":71.1,

118	"currency":"EUR"
119	}
120],
121	"waypoints":[
122	{
123	"waypointId":3521,
124	"waypointName":"name_of_waypoint_with_id_3521",
125	"waypointTime":1503391597
126	},
127	{
128	"waypointId":173,
129	"waypointName":"name_of_waypoint_with_id_173",
130	"waypointTime":1503391602
131	}
132],
133	"notes": null ,
134	"contracts":[1,2]
135	}
136],
137	"page":7,
138	"count":100,
139	"totalPages":11,
140	"totalCount":1050,
141	"links":[
142	{
143	"rel":"prev",
144	"href":"https://online.commander-systems.com/api/v1/rides/987458?
	datetimeStart=1483225200&datetimeEnd=1504705692&page=6"
145	},
146	{
147	"rel":"next",
148	"href":"https://online.commander-systems.com/api/v1/rides/987458?
	datetimeStart=1483225200&datetimeEnd=1504705692&page=8"
149	}
150]
151	}

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	Res	sponse data
4xx or 5xx	1	{
	2	"status":"error",
	3	"message":"error message"
	4	}

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}

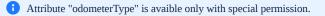
URL parameters accepted by the service (mandatory):

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.



Status	Response data					
200	1	{				
	2	"rides":[
	3	{				
	4	"rideId":"856757",				
	5	"rideType":"BUSINESS_RIDE",				
	6	"vehicleId":"987458",				
	7	"odometerType":0,				
	8	"vehicleRegistrationPlate":"BA 010 AB",				
	9	"driverId":"12354",				
	10	"driverName":"Janko Vodič",				
	11	"note":"Jazda do Košíc",				
	12	"startTime":1503391595,				
	13	"stopTime":1503407852,				
	14	"latStart":48.1928,				
	15	"lonStart":17.1409,				
	16	"latStop":48.7047,				

17	"lonStop":21.2602,
18	"startAddress":"Račianska 2430/168, 831 54 Bratislava, Slovensko",
19	"stopAddress":"Rastislavova 784/68, 040 01, Košice, Slovensko",
20	"avgSpeed":35.9,
21	"maxSpeed":137.9,
22	"duration":16257,
23	"distance":350.23,
24	"odometerStart":123876,
25	"odometerStop":124226,
26	"engineHoursStart":10.56
27	"fuelLevelStart":43,
28	"fuelLevelStop":12,
29	"fuelConsumed":21.81,
30	"averageConsumption":6.23,
31	"refueling":[
32	{
33	"datetime":1503392895,
34	"odometer":123998.39,
35	"volume":41.52,
36	"priceVatExc":52.49,
37	"priceVatInc":62.99,
38	"currency":"EUR"
39	},
40	{
41	"datetime":1503393629,
42	"odometer":124382.47,
43	"volume":45.39,
44	"priceVatExc":59.25,
45	"priceVatInc":71.1,
46	"currency":"EUR"
47	}
48],
49	"waypoints":[
50	{
51	"waypointId":3520,
52	"waypointName":"name_of_waypoint_with_id_3520",
53	"waypointTime":1503391595
54	},
55	{
56	"waypointId":183,
57	"waypointName":"name_of_waypoint_with_id_183",
58	"waypointTime":1503391599
59	}
60],
61	"notes":[
62	{ !!potsIJ!!0072
63	"noteId":9073,
64	"rideId":5345,
65	"text":"Dvere nákladového priestoru otvorené",
66	"timestamp":1419328781,
67	"duration":{
68	"value":44,
69 70	"unit":"seconds"
70	}
71	}
72], "contracte" well
73	"contracts": null
74	},
75	{ "ridaId"-"856758"
76	"rideId":"856758",

77	"rideType":"PRIVAT_RIDE",
78	"vehicleId":"987458",
79	"odometerType":0,
80	"vehicleRegistrationPlate":"BA 010 AB",
81	"driverId":"12354",
82	"driverName":"Janko Vodič",
83	"note":"Jazda do Košíc",
84	"startTime":1503391595,
85	"stopTime":1503407852,
86	"latStart":null,
87	"lonStart": null,
88	"latStop":null,
89	"lonStop": null,
90	"startAddress":null,
91	"stopAddress":null,
92	"avgSpeed":35.9,
93	"maxSpeed":137.9,
94	"duration":16257,
95	"distance":350.23,
96	"odometerStart":123876,
97	"odometerStop":124226,
98	"engineHoursStart":11.01
99	"fuelLevelStart":43,
100	"fuelLevelStop":12,
101	"fuelConsumed":0,
102	"averageConsumption":0,
103	"refueling":[
104	{
105	"datetime":1503392895,
106	"odometer":123998.39,
107	"volume":41.52,
108	"priceVatExc":52.49,
109	"priceVatInc":62.99,
110	"currency":"EUR"
111	},
112	{
113	"datetime":1503393629,
114	"odometer": null,
115	"volume":45.39,
116	"priceVatExc":59.25,
117	"priceVatInc":71.1,
118	"currency":"EUR"
119	}
120],
121	"waypoints":[
122	{
123	"waypointId":3521,
124	"waypointName":"name_of_waypoint_with_id_3521",
125	"waypointTime":1503391596
126	},
127	{
128	"waypointId":173,
129	"waypointName":"name_of_waypoint_with_id_173",
130	"waypointTime":1503391601
131	}
132],
133	"notes":null,
134	"contracts":[1,2]
135	}
136],

137	"page":7,
138	"count":100,
139	"totalPages":11,
140	"totalCount":1050,
141	"links":[
142	{
143	"rel":"prev",
144	"href":"https://online.commander-systems.com/api/v1/rides/987458?
	datetimeStart=1483225200&datetimeEnd=1504705692&page=6"
145	},
146	{
147	"rel":"next",
148	"href":"https://online.commander-systems.com/api/v1/rides/987458?
	datetimeStart=1483225200&datetimeEnd=1504705692&page=8"
149	}
150]
151	}

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

noeNinaNer abare the (nyinely syntem by the drive)surd TimeInce all date of the state of the side in Unix timescamp format.surd TimeInce all date of the stating point of the ride in decimal format.InstantFordInce all date of the stating point of the ride in decimal format.InstantFordInce all date of the stating point of the ride in decimal format.InstantFordInce all dates of the ending point of the ride in decimal format.Instant AddressNoneNoneSurg AddressSurg Address of the ending point of the ride in decimal format.Surg AddressSurg Address of the ending point of the ride in decimal format.Surg AddressSurg Address of the ending point of the ride in decimal format.Surg AddressSurg Address of the ending point of the ride in decimal format.Surg AddressSurg Address of the ending point of the ride in decimal format.Surg AddressSurg Address of the ending point of the ride in date.Surg AddressSurg Address of the ending point of the ride in date.Surg AddressSurg Address of the ending point of the ride in date.Surg Address of the ending surg Address of the ending point of the ride in date.Surg Address of the ending surg Address of the ending point of the ride in date.Surg Address of the ending surg Address of the ending address of the ending surg Address of the ending surg Address of the ending address of the ending Address of the ending address of the ending address				
InterpretationIntegralTime and date of the raidladSartFloatLatitude of the starting point of the ride in decimal formatladSartFloatCongitude of the starting point of the ride in decimal formatladSopFloatCongitude of the ending point of the ride in decimal formatladSopFloatCongitude of the ending point of the ride in decimal formatsonFAGStringPostal address of the starting point of the ridesonFAGStringPostal address of the ending point of the ridesonFAGStringPostal address of the starting point of the ridesonFAGFloatAverage speed during the ride in km/h (rounded to 0 decimal numbers)maxDevelHaterDatation of the ride in seconds (counded to 0 decimal numbers)diataceFloatDistance traveled from the starting point of the ride in km after correctiondiotereStortFloatFloatPostal address of the ending moment of the ride in km after correctionofoneterStortFloatSingle Floaus of velocineter in the starting moment of the ride in lines (or in kWh shen vehicle is electric car)fuelLevelStorRingSingle Counded to 0 decimal numbers)fuelLevelStorRingSingle Counded from "fuelCounsered" and "distance" (rounded to 2 decimal numbers)fuelLevelStorRingSingle Counded from "fuelCounsered" and "distance" (rounded to 2 decimal numbers)fuelLevelStorRingSingle Counded from "fuelCounsered" and "distance" (rounded to 2 decimal numbers)fuelLevelStorSingle Counded from "fuelCounsered" and "distance" (rounded to 2	note	String	Note about the ride (typically written by the driver)	
Instant Float Latitude of the starting point of the ride in decimal format lasStart Float Longitude of the starting point of the ride in decimal format lasStop Float Longitude of the ending point of the ride in decimal format lasStop Float Longitude of the ending point of the ride in decimal format startAdfress String Postal address of the starting point of the ride startAdfress String Postal address of the ending point of the ride avgSpeed Float Average speed during the ride in km/h (reunded to 0 decimal numbers) duration Integer Maximum speed during the ride in km/h (reunded to 0 decimal numbers) duration Integer Distance traveled from the starting moment of the ride in km after correction odometerStart Float The value of the odometer in the starting moment of the ride in km after correction odometerStart*** Float Startee of twe lot he starting moment of the ride in lines (or in kWn when vehicle is electric car), (counded to 0 decimal numbers) fuellevelStart Integer The level of fuel In the ending moment of the ride in lines (or in kWn when vehicle is electric car), (counded to 0 decimal numbers) fuelevelStare Float Nearege consa	startTime	Integer	Time and date of the start of the ride in Unix timestamp format	
IonStart Float Longitude of the starting point of the ride in decimal format larStop Float Latidude of the earding point of the ride in decimal format lonStop Float Langitude of the earding point of the ride in decimal format startAddress Sring Postal address of the earding point of the ride in decimal format startAddress Sring Postal address of the earding point of the ride argSpeed Float Average speed during the ride in km/h maxSpeed Ineger Datation of the ride in km/h (rounded to 0 decimal numbers) distance Float Datation of the ride in sconds (rounded to 0 decimal numbers) distance Float Datation of the ride in skonds (rounded to 0 decimal numbers) odometerStart Float To value of the odometer in the starting moment of the ride in km after correction figueHoursStart* Float To value of the odometer in the earding moment of the ride in km/h when vehicle is electric car), (rounded to 0 decimal numbes) fuelLevelStart Rose To scale of fuel in the earding moment of the ride in km when vehicle is electric car), (rounded to 0 decimal numbes) fuelLevelStart Rose Consumed tins the theding incolure of rounderof rounder to decimal numbes) </td <td>stopTime</td> <td>Integer</td> <td colspan="2">Time and date of the end of the ride</td>	stopTime	Integer	Time and date of the end of the ride	
Instrume Instrume IaRSop For Laitude of the ending point of the ide in decimal format IonStop Ford Longitude of the ending point of the ride in decimal format starAddress Sring Postal address of the ending point of the ride in decimal format starAddress Sring Postal address of the ending point of the ride starAddress Sring Postal address of the ending point of the ride starAddress Sring Postal address of the ending point of the ride starSpeed Ford Narsonus speed during the ride in kon/h (counded to 0 decimal numbers) duration Integer Datation of the ride in seconds (rounded to 0 decimal numbers) duration Ford The value of the odometer in the satiring moment of the ride in kan after correction odometerStar Ford The value of the odometer in the ending moment of the ride in kan after correction fuelLevelStart Integer Engine hours of value of the odometer in the ending moment of the ride in liters (or in kWh when vehicle is electric car), from duration electric car), from duration of decimal numbers) fuelLevelStart Integer Nareogr consumption form "fuelCarsonume" and "distarce" (rounded to 2 decimal numbers) fuelLeve	latStart	Float	Latitude of the starting point of the ride in decimal format	
Instage Final Longitude of the ending point of the ride in decimal format IonStop Final Rongitude of the ending point of the ride in decimal format stenAhdress Sring Postal address of the ending point of the ride arqSpeed Final 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 Distance traveled from the starting point of the ride in km after correction dometerStart Float The value of the dometer in the starting moment of the ride in km after correction odometerStop Float Engine hours of value of the dometer in the ending moment of the ride in km after correction ofuneterStop Float Engine hours of value of the dometer in the ending moment of the ride in km after correction fuelLevelStart Ringer Engine hours of value of the dometer in the ending moment of the ride in km where value is electric car), (rounded to 0 decimal numbers) fuelLevelStop Ringer The level of fuel in the ending moment of the ride in km where value is electric car), from LVCAN during ride (rounded to 2 decimal numbers) reluting kl_dutetime Ringer Noreage consumption form 'fuelConsumed' and 'distance' (rounded to 2 decimal numbers) <td>lonStart</td> <td>Float</td> <td>Longitude of the starting point of the ride in decimal format</td>	lonStart	Float	Longitude of the starting point of the ride in decimal format	
AutorianFor an articlestartAddressStringPostal address of the starting point of the ridestopAddressStringPostal address of the ending point of the rideavgSpeedFloatAverage speed during the ride in km/hmaxSpeedIntegerMaximum speed during the ride in km/hdurationIntegerDuration of the ride in seconds (counded to 0 decimal numbers)durationFloatSistance traveled from the starting point to the ending point in km after correctionodometerStartFloatThe value of the odometer in the starting moment of the ride in km after correctionodometerStopFloatThe value of the odometer in the ending moment of the ride in km after correctionofmeterStopFloatThe value of the odometer in the ending moment of the ride in km after correctionfueLevelStartIntegerThe 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)fueLevelStartIntegerThe 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)fueLevelStartRoteStrange consumption from "fueLConsumed" and "listance" (rounded to 2 decimal numbers)fueLevelStartFloatArray of all refuelings assigned to the ride ride ride in kmareageConsumptionFloatThe value of the erfueling in Unix dimestamp formatrefuelingk1, duterimFloatThe refueling web of the refueling in Unix dimestamp formatrefuelingk1, valuemeFloatThe refueling excluding VAT <td< td=""><td>latStop</td><td>Float</td><td>Latitude of the ending point of the ride in decimal format</td></td<>	latStop	Float	Latitude of the ending point of the ride in decimal format	
Intervention Form Description of the refueling 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 (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 odometerStart** Float The value of the odometer in the ending moment of the ride in km after correction fuelLevelStart Float 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) fuelLevelStart Integer The level of fuel in the ending moment of the ride in liters (or in kWh when vehicle is electric car), (rounded to 2 decimal numbers) fuelLevelStart Integer The level of fuel in the ending moment of the ride in liters (or in kWh when vehicle is electric car), (rounded to 2 decimal numbers) fuelLevelStart Integer The level of fuel in the ending moment of the ride in liters (or in kWh when vehicle is electric car), (rounded to 2 decimal numbers) fuelLevelStart	lonStop	Float	Longitude of the ending point of the ride in decimal format	
avgSpeed Float Average speed during the ride in km/h avgSpeed 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 Engine hours of vehicle at the beginning of ride. Engine hours are not always matching stopTime- startTime. fueLevelStart Integer The value of the undometer in the arting moment of the ride in litters (or in kWh when vehicle is electric car), (rounded to 0 decimal numbers) fueLevelStart Integer The level of fuel in the ending moment of the ride in litters (or in kWh when vehicle is electric car), (rounded to 0 decimal numbers) fueLevelStop Integer Streage consumption from "fueLConsumed" and "distance" (rounded to 2 decimal numbers) refueling(s)_dotamet Float Arcarge consumption form "fueLConsumed" and "distance" (rounded to 2 decimal numbers) refueling(s)_dotameter Float Arcarge consumption from "fueLConsumed" and "distance" (rounded to 2 decimal numbers) refueling(s)_dotameter Float	startAddress	String	Postal address of the starting point of the ride	
maxSpeedIntegerMaximum speed during the ride in kmh (rounded to 0 decimal numbers)durationIntegerDuration of the ride in seconds (rounded to 0 decimal numbers)distanceFloatDistance traveled from the starting point to the ending point in km after correctionodometerStartFloatThe value of the odometer in the starting moment of the ride in km after correctionodometerStartFloatThe value of the odometer in the ending moment of the ride in km after correctionengineHoursStart**RoltSingine hours of vehicle at the beginning of ride. Engine hours are not always matching stopTimesfuElevelStartIntegerThe level of fuel in the starting moment of the ride in litters (or in kWh when vehicle is electric car), (rounded to 0 decimal numbers)fuElevelStopIntegerSoltane Under the odometer in the roleing moment of the ride in litters (or in kWh when vehicle is electric car) (rounded to 0 decimal numbers)refuelingsLyncemetFloatNearge consumption from "fuelConsumed" and "distance" (rounded to 2 decimal numbers)refuelingsLyncemetFloatThe value of the odometer in the moment of refueling in kmrefuelingsLyncevatureFloatThe value of the refueling in Unix timestamp formatrefuelingsLyncevatureFloatThe value of the refueling refueling VATrefuelingsLyncevatureFloatThe value of the	stopAddress	String	Postal address of the ending point of the ride	
AdvancionIntegerDuration of the ride in seconds (rounded to 0 decimal numbers)dixancionFloatDistance traveled from the starting point to the ending point in km after correctionodometerStartFloatThe value of the odometer in the starting moment of the ride in km after correctionodometerStopFloatThe value of the odometer in the ending moment of the ride in km after correctionengineHoursStart**FloatEngine hours of vehicle at the beginning of ride. Engine hours are not always matching stopTime- startTime.fueLevelStartIntegerThe 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)fueLevelStopIntegerThe 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)fuelConsumedFloatAverage consumption from "fuelConsumed" and "distance" (rounded to 2 decimal numbers)refueling[x].datetimeIntegerTime and date of the refueling in Unix timestamp formatrefueling[x].odometerFloatAverage consumption from "fuelConsumed" and "distance" (rounded to 2 decimal numbers)refueling[x].priceVattscFloatThe value of the odometer in the moment of refueling in kmrefueling[x].priceVattscFloatThe value of the refueling in Unix timestamp formatrefueling[x].priceVattscFloatThe price of the refueling excluding VATrefueling[x].priceVattscFloatThe price of the refueling including VATrefueling[x].priceVattscFloatThe urrecy of the refueling transaction<	avgSpeed	Float	Average speed during the ride in km/h	
IdeaFloatDistance traveled from the starting point to the ending point in km after correctionodometerStartFloatThe value of the odometer in the starting moment of the ride in km after correctionodometerStopFloatThe value of the odometer in the ending moment of the ride in km after correctionengineHoursStart**FloatEngine hours of vehicle at the beginning of ride. Engine hours are not always matching stopTime- startTime.fueLevelStartIntegerThe 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)fueLevelStopIntegerConsumed liters (or in kWh when vehicle is electric car), (rounded to 0 decimal numbers)fueLonsumedFloatConsumed liters (or in kWh when vehicle is electric car), (rounded to 0 decimal numbers)refuelingFloatNareage consumption from "fueLConsumed" and "distance" (rounded to 2 decimal numbers)refueling(s).dotterieFloatTime and date of the refueling in Unix timestamp formatrefueling(s).dotterieFloatThe value of the odometer in the moment of refueling in kmrefueling(s).priceVatExcFloatThe price of the refueling excluding VATrefueling(s).priceVatExcFloatThe price of the refueling including VATrefueling(s).priceVatImStringThe price of the refueling including VATrefueling(s).priceVatImFloatThe price of the refueling including VATrefueling(s).priceVatImFloatThe price of the refueling including VATrefueling(s).priceVatImFloatThe price of the refueling includ	maxSpeed	Integer	Maximum speed during the ride in km/h (rounded to 0 decimal numbers)	
odometerStartFloatThe value of the odometer in the starting moment of the ride in km after correctionodometerStopFloatThe value of the odometer in the ending moment of the ride in km after correctionengineHoursStart**FloatEngine hours of vehicle at the beginning of ride. Engine hours are not always matching stopTime- startTime.fueLevelStartIntegerThe 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)fueLevelStopIntegerThe 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)fuelConsumedFloatConsumed liters (or in kWh when vehicle is electric car) from LVCAN during ride (rounded to 2 decimal numbers)refuelingFloatAverage consumption from "fuelConsumed" and "distance" (rounded to 2 decimal numbers)refueling(x)_datetimeIntegerTime and date of the refueling in Unix timestamp formatrefueling(x)_odometerFloatThe value of the odometer in the moment of refueling in kmrefueling(x)_priceVatExcFloatThe price of the refueling excluding VATrefueling(x)_priceVatIncFloatThe price of the refueling including VATrefueling(x)_rorencyStringThe currency of the refueling transactionwaypointsArray of all refueling including VAT	duration	Integer	Duration of the ride in seconds (rounded to 0 decimal numbers)	
odometerStopFloatThe value of the odometer in the ending moment of the ride in km after correctionengineHoursStart**FloatEngine hours of vehicle at the beginning of ride. Engine hours are not always matching stopTimes startTime.fueLevelStartIntegerThe 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)fueLevelStopIntegerThe 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)fueConsumedFloatConsumed liters (or in kWh when vehicle is electric car) from LVCAN during ride (rounded to 2 decimal numbers)refueIng(x)_datetimeFloatAverage consumption from "fueConsumed" and "distance" (rounded to 2 decimal numbers)refueling(x)_datetimeFloatTime and date of the refueling in Unix timestamp formatrefueling(x)_otometerFloatThe value of the odometer in the moment of refueling in kmrefueling(x)_priceVatExcFloatThe price of the refueling excluding VATrefueling(x)_priceVatExcFloatThe price of the refueling in cluding VATrefueling(x)_priceVatExcFloatThe price of the refueling including VATrefueling(x)_priceVatExcFloatThe price of the refueling including VATrefueling(x)_priceVatExcFloatThe currency of the refueling including VATrefueling(x)_priceVatExcFloatThe currency of the refueling including VATrefueling(x)_priceVatExcFloatThe currency of the refueling including VATrefueling(x)_priceVatExcFloat	distance	Float	Distance traveled from the starting point to the ending point in km after correction	
and the second	odometerStart	Float	The value of the odometer in the starting moment of the ride in km after correction	
IndexIntegerstartTime.fuelLevelStartIntegerThe 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)fuelLevelStopIntegerThe 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)fuelConsumedFloatConsumed liters (or in kWh when vehicle is electric car) from LVCAN during ride (rounded to 2 decimal numbers)averageConsumptionFloatAreage consumption from "fuelConsumed" and "distance" (rounded to 2 decimal numbers)refueling[x].datetimeIntegerTime and date of the refueling in Unix timestamp formatrefueling[x].odometerFloatThe value of the odometer in the moment of refueling in kmrefueling[x].priceVatExcFloatThe grice of the refueling excluding VATrefueling[x].priceVatExcFloatThe price of the refueling including VATrefueling[x].priceVatExcKingAray of any of	odometerStop	Float	The value of the odometer in the ending moment of the ride in km after correction	
index(rounded to 0 decimal numbers)fuelLevelStopIntegerThe 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)fuelConsumedFloatConsumed liters (or in kWh when vehicle is electric car) from LVCAN during ride (rounded to 2 decimal numbers)averageConsumptionFloatAverage consumption from "fuelConsumed" and "distance" (rounded to 2 decimal numbers)refueling[x].datetimeIntegerTime and date of the refueling in Unix timestamp formatrefueling[x].odometerFloatThe value of the odometer in the moment of refueling in kmrefueling[x].volumeFloatThe fuel volume of the refueling the fuel volume of the refueling XITrefueling[x].priceVatIccFloatThe price of the refueling saction YATrefueling[x].priceVatIccFloatThe price of the refueling including VATrefueling[x].priceVatIccFloatThe currency of the refueling including VATwappointsArrayArray of any of	engineHoursStart**	Float		
i.e.I.e.(rounded to 0 decimal numbers)fuelConsumedFloatConsumed liters (or in kWh when vehicle is electric car) from LVCAN during ride (rounded to 2 decimal numbers)averageConsumptionFloatAverage consumption "fuelConsumed" and "distance" (rounded to 2 decimal numbers)refuelingArrayArray of all refuelings assigned to the ride (zero or more)refueling[x].datetimeIntegerTime and date of the refueling in Unix timestamp formatrefueling[x].odometerFloatThe value of the odometer in the moment of refueling in kmrefueling[x].priceVatExcFloatThe price of the refueling excluding VATrefueling[x].priceVatExcFloatThe price of the refueling in Curding VATrefueling[x].currencyStringThe currency of the refueling transactionwapointsArrayArray of all refueling transaction	fuelLevelStart	Integer	-	
initialinitialaverageConsumptionFloatAverage consumption from "fuelConsumed" and "distance" (rounded to 2 decimal numbers)refuelingArrayArray of all refuelings assigned to the ride (zero or more)refueling[x].datetimeIntegerTime and date of the refueling in Unix timestamp formatrefueling[x].odometerFloatThe value of the odometer in the moment of refueling in kmrefueling[x].youmeFloatThe fuel volume of the refueling excluding VATrefueling[x].priceVatExcFloatThe price of the refueling in CMATrefueling[x].priceVatImeFloatThe currency of the refueling transactionwaypointsArrayArray of auto price of the refueling transaction	fuelLevelStop	Integer		
refuelingArrayArray of all refuelings assigned to the ride (zero or more)refueling[x].datetimeIntegerTime and date of the refueling in Unix timestamp formatrefueling[x].odometerFloatThe value of the odometer in the moment of refueling in kmrefueling[x].volumeFloatThe fuel volume of the refuelingrefueling[x].priceVatExcFloatThe price of the refueling excluding VATrefueling[x].priceVatIncFloatThe price of the refueling including VATrefueling[x].priceVatIncStringThe currency of the refueling transactionwaypointsArrayArray of waypoints that were entered or leaved during the ride	fuelConsumed	Float		
refueling[x].datetimeIntegerTime and date of the refueling in Unix timestamp formatrefueling[x].odometerFloatThe value of the odometer in the moment of refueling in kmrefueling[x].volumeFloatThe fuel volume of the refuelingrefueling[x].priceVatExcFloatThe price of the refueling excluding VATrefueling[x].priceVatIncFloatThe price of the refueling including VATrefueling[x].currencyStringThe currency of the refueling transactionwaypointsArrayArray of waypoints that were entered or leaved during the ride	averageConsumption	Float	Average consumption from "fuelConsumed" and "distance" (rounded to 2 decimal numbers)	
refueling[x].odometerFloatThe value of the odometer in the moment of refueling in kmrefueling[x].volumeFloatThe fuel volume of the refuelingrefueling[x].priceVatExcFloatThe price of the refueling excluding VATrefueling[x].priceVatIncFloatThe price of the refueling including VATrefueling[x].currencyStringThe currency of the refueling transactionwaypointsArrayArray of waypoints that were entered or leaved during the ride	refueling	Array	Array of all refuelings assigned to the ride (zero or more)	
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	refueling[x].datetime	Integer	Time and date of the refueling in Unix timestamp format	
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	refueling[x].odometer Float The value		The value of the odometer in the moment of refueling in km	
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	refueling[x].volume Float		The fuel volume of the refueling	
refueling[x].currency String The currency of the refueling transaction waypoints Array Array of waypoints that were entered or leaved during the ride	refueling[x].priceVatExc Float The price of the refueling excluding VAT		The price of the refueling excluding VAT	
waypoints Array Array of waypoints that were entered or leaved during the ride	refueling[x].priceVatInc	Float	The price of the refueling including VAT	
	refueling[x].currency	String	The currency of the refueling transaction	
waypoints[x].waypointId String Internal ID of a waypoint (can be used for waypoint matching from other API services)	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*** vaible only with special permission	Array/null	Array of ID of contracts in ride. When empty, parameter is null	

**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	1	{ "status":"error",			
	3	"message":"error message"			
	4	}			

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/jsonAuthorization : 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.

Status	Resp	onse data
200	1	{
	2	"waypoints":[
	3	{
	4	"waypointId":"123",
	5	"waypointName":"name of this waypoint",
	6	"waypointPolygon":[
	7	{
	8	"lon":14.390626,
	9	"lat":50.076788
	10	},
	11	{
	12	"lon":14.390755,
	13	"lat":50.076213
	14	},
	15	{
	16	"lon":14.390875,
	17	"lat":50.073421
	18	}
	19]
	20	},
	21	{
	22	"waypointId":"223",
	23	"waypointName":"name of another waypoint",
	24	"waypointPolygon":[
	25	{
	26	"lon":14.396532,
	27	"lat":50.077453
	28	},
	29	{
	30	"lon":14.398755,
	31	"lat":50.076234
	32	},
	33	{
	34	"lon":14.398421,
	35	"lat":50.077144
	36	}
	37]
	38	}
	39]
	40	}

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

|--|

Status	Res	Response data				
4xx or 5xx	1 2 3 4	{ "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/jsonAuthorization : 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	1	[
	2	{	
	3	"waypointGroupID": 123123,	
	4	"waypointGroupName": "Test group"	
	5	},	
	6	{	
	7	"waypointGroupID": 123124,	
	8	"waypointGroupName": "Test group2"	
	9	}	
	10	1	

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)

Status	Res	Response data		
4xx or 5xx	1 2 3 4	{ "status":"error", "message":"error message" }		

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/jsonAuthorization : 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.

1	Status	Response data				
2	200	1 {				
		2 "positions": [
		3 {				
		4 "vehicleId": 123456,				
		5 "gpsTime": 1563957600,				

6	"gpsLat": 49.208797,
7	"gpsLon": 16.154687,
8	"gpsLAlt": 200,
9	"gpsAzimut": 10,
10	"gpsSpeed": 56,
11	"carIgnition": 1,
12	"voltage": 14.4,
13	"canSpeed": 60,
14	"canThrottle": 20,
15	"canConsumed": 1256,
16	"canTankValue": 50 ,
17	"canRpm": 2500,
18	"canEngineHours": 100,
19	"canOdometer": 5000,
20	"temperatures":[
21]
22	},
23	{
24	"vehicleId": 123457,
25	"gpsTime": 1563952312,
26	"gpsLat": 48.598746,
27	"gpsLon": 18.12569,
28	"gpsLAlt": 10,
29	"gpsAzimut": 0,
30	"gpsSpeed": 0,
31	"carIgnition": 0,
32	"voltage": 12.0,
33	"canSpeed": 0,
34	"canThrottle": 0,
35	"canConsumed": 0,
36	"canTankValue": 0,
37	"canRpm": 0 ,
38	"canEngineHours": 0,
39	"canOdometer": 0,
40	"temperatures":{"temperature":6.8,"fridge":7,"freezer":""}
41	}
42],
43	"page": 1,
44	"count": 2,
45	"totalPages": 1,
46	"totalCount": 2
47	}

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	<pre>{ { :status":"error", "message":"error message" } }</pre>		

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/jsonAuthorization : Basic Auth	https://online.commander-systems.com/api/v1/drivers

URL parameters accepted by the service:

e 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	s Response data					
200	1	{				
	2	"drivers": [
	3	{				
	4	"id": 123456,				
	5	"name": "Name",				
	6	"surname": "Surname",				
	7	"personal_number": "0002",				
	8	"identification_keys": [
	9	{				
	10	"code": "123456789",				
	11	"type": 0,				
	12	"name": "ABCDEF"				
	13	},				
	14	{				
	15	"code": "987654321",				
	16	"type": 1,				
	17	"name": "ABC12"				
	18	}				
	19],				
	20	"cost_center_text":"",				
	21	"cost_center_id":0				
	22	},				
	23	{				
	24	"id": 123457,				
	25	"name": "Name2",				
	26	"surname": "Surname2",				
	27	"personal_number": "",				
	28	"identification_keys": [],				
	29	"cost_center_text":"Cost center 123",				
	30	"cost_center_id":5				
	31	}				
	32],				
	33	"page": 1,				
	34	"count": 2,				
	35	"totalPages": 1,				
	36	"totalCount": 2				
	37	}				

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

Status	Res	Response data				
4xx or 5xx	1	{				
	2	"status":"error",				
	3	"message":"error message"				
	4	}				

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/jsonAuthorization : Basic Auth	https://online.commander-systems.com/api/v1/refueling-import

URL parameters accepted by the service:

Parameter name Data type		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.	
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:

tatus	Resp	Response data					
00	1	{					
	2	"imports": [
	3	{					
	4	"importTime": 1644648245,					
	5	"importValue": 40.02,					
	6	"importCost": 64.32,					
	7	"importCostVat": 76.84,					
	8	"importCostPerLiter": 1.60,					
	9	"importCostPerLiterVat": 1.92,					
	10	"importVat": 20,					
	11	"importObjectIdentifier": "AA000AA",					
	12	"importStatus": 1,					
	13	"importObjectId": 0,					
	14	"importRideId": 0					
	15	}					
	16],					
	17	"page": 1,					
	18	"count": 1,					
	19	"totalPages": 1,					
	20	"totalCount": 1					
	21	}					

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

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/jsonAuthorization : 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	Resp	Response data		
200	1	{		
	2	"contracts": [
	3	{		
	4	"contractId": 1,		
	5	"contractName": "Contract 1"		
	6	},		
	7	{		
	8	"contractId": 1,		
	9	9 "contractName": "Contract 2"		
	10	10 }		
	11	11],		
	12	"page": 1,		
	13 "count": 2 ,			
	14	"totalPages": 1,		
	15	"totalCount": 2		
	16	}		

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

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 /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/jsonAuthorization : 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.

Status	Response data		
200	1 {		
	2 "costCenters": [
	3 {		
	4 "costCenterId": 5,		

_	
5	"costCenterNumber": "001",
6	"costCenterName": "Office",
7	"costCenterDeleted": 0
8	
9	},
10	{
11	"costCenterId": 6,
12	"costCenterNumber": "002",
13	"costCenterName": "Office 2",
14	"costCenterDeleted": 1
15	}
16],
17	"page": 1,
18	"count": 2,
19	"totalPages": 1,
20	"totalCount": 2
21	}

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	L {		
	2 "status":"error",		
	3 "message":"error message"		
	4 }		

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/jsonAuthorization : 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	1 {"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	1	{	
	2	"status":"error",	
	3	"message":"error message"	
	4	}	

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