

movizon CONTROL REST API

Base URL

Scheme	[SERVER_ADRESS]/api/v1
Example A	https://demo.movizon.de/api/v1
Example B	http://localhost:8080/movizoncontrol/api/v1

Authentication

Each REST API request is assigned to a registered user, which must authenticate itself using HTTP basic authentication. The credentials are transmitted as Base64 encoded string in the form [USERNAME]:[PASSWORD] inside the authorization header. If authentication fails, error code 401 is returned.

Authorization

Each REST API request requires permission. If permission is not granted, error code 403 is returned. This happens in one of the following cases:

- The administrator settings for the requesting user does not allow resource access in combination with the requested CRUD operation.
- The result of the requested operation would exceed the responsibility limit for this resource type set for the requesting user by the administrator. For example, if a user is allowed to be responsible for up to 10 jobs, this user won't be able to create an 11th job - but deletion requests would be granted.
- The user requests a modification or deletion of a resource that is locked by another user.
- The user requests an operation that is forbidden by the system, e.g. the modification of a read-only resource or resource property.

Standard Request Types

Several request types are available to perform standard CRUD (Create, Read, Update, Delete) operations on mC's resources:

Method	Path	Request	Success Response			Description
		Body	Code	Location Header	Body	
POST	[BASE_URL]/[RT]	resource as JSON String	201	ID of the created resource		create new resource; ID will be set by server
GET	[BASE_URL]/[RT]	-	200	-	resource array as JSON String	retrieve all resources of type [RT]
GET	[BASE_URL]/	-	200	-	resource	retrieve resource

	[RT]/[ID]				as JSON String	
GET	[BASE_URL]/[RT]/[ID]/[KEY]	-	200	-	object as JSON String	retrieve value of a single field of resource
PUT	[BASE_URL]/[RT]/[ID]	resource as JSON String	204 or 201	if code 201: ID of the created resource		create or overwrite a resource with specified ID
PUT	[BASE_URL]/[RT]/[ID]/[KEY]	object as JSON String	204	-		set value of a single field of resource
DELETE	[BASE_URL]/[RT]/[ID]	-	204	-		delete a resource
DELETE	[BASE_URL]/[RT]/[ID]/[KEY]	-	204	-		delete a single field of a resource

Filter & Search Queries

GET requests can add filter- and search queries. Both queries also can be combined in a single request. For details please see the following table:

GET request with single key filter	
Path Scheme	[BASE_URL]/[RT]/[ID]?keys=[KEY]
Path Example	https://demo.movizon.de/api/v1/vehicles?keys=id
Description	response will contain NOT an array of objects, but only an array with values of the queried key, e.g. ["TCVKS001", "TCVKS002", "TCVKS003"]
GET request with multiple key filter	
Path Scheme	[BASE_URL]/[RT]/[ID]?keys=[KEY],[KEY],[...]
Path Example	https://demo.movizon.de/api/v1/vehicles?keys=id,energy
Description	response will contain array of objects with values of the queried keys, e.g. [{"id": "TCVKS001", "energy": "46"}, {"id": "TCVKS002", "energy": "13"}, {"id": "TCVKS003", "energy": "85"}]
GET request with single search query	
Path Scheme	[BASE_URL]/[RT]/[ID]?[KEY][OPERATOR]=[VALUE]
Path Example	https://demo.movizon.de/api/v1/vehicles?id[sw]=TCVKS
Description	response will contain array of resources that match the search query; available operators: [gt] → greater than; for numbers only [lt] → less than; for numbers only [ge] → greater than or equals; for numbers only [le] → less than or equals; for numbers only [sw] → starts with; for strings only [ew] → ends with; for strings only

	<p>[co] → contains; for strings only [eq] → equals; uses JAVA equals method [ne] → not equals; uses JAVA equals method</p>
GET request with multiple search queries	
Path Scheme	[BASE_URL]/[RT]/[ID]?[KEY][OPERATOR]=[VALUE]&[KEY][OPERATOR]=[VALUE]
Path Example	https://demo.movizon.de/api/v1/vehicles?id[sw]=TCVKS&energy[gt]=75
Description	response will contain array of resources that match ALL search queries (boolean AND, not OR), e.g. [{"id": "TCVKS003", "energy": "85", "speed": "100", [...]}]
GET request with key filter and search query	
Path Scheme	[BASE_URL]/[RT]/[ID]?keys=[KEY]&[KEY][OPERATOR]=[VALUE]
Path Example	https://demo.movizon.de/api/v1/vehicles?keys=id&energy[gt]=40
Description	response will contain array of values of the queried key that match the appended search query, e.g. ["TCVKS001", "TCVKS003"]

Response Codes

Code	Name	Description
200	OK	request successfully processed; resource as JSON object returned in response body
201	CREATED	resource created; no response body; ID of the newly created resource inside location header
204	NO_CONTENT	request successfully processed; no response body
207	MULTI_STATUS	response to a batch request; individual feedback inside the response body as JSON array
400	BAD_REQUEST	unprocessible request, e.g. resource JSON can't be parsed
401	UNAUTHORIZED	unknown user (wrong username or password)
403	FORBIDDEN	resource access not permitted
404	NOT_FOUND	resource or method not found
413	PAYLOAD_TOO_LARGE	string size of body exceeds limit in settings; default limits: 10.000 chars for single field, 25.000 chars for entire resource
422	UNPROCESSABLE_ENTITY	resource in the request body can't be initialized
500	INTERNAL_SERVER_ERROR	resource can't be updated, e.g. threading exception

Batch Service

In order to minimize the overhead generated by HTTP requests, several individual requests can be combined into a single request. The following table shows the details:

	Scheme	Example
Request method	POST	
Request path	[BASE_URL]/batch	
Request body	<pre>[{ "method": [METHOD_1], "path": [RELATIVE_PATH_1], "body": [BODY_1] }, { "method": [METHOD_2], "path": [RELATIVE_PATH_2], "body": [BODY_2] }, { "method": [METHOD_N], "path": [RELATIVE_PATH_N], "body": [BODY_N] }]</pre>	<pre>[{ "method": "DELETE", "path": "waypoints/77", "body": "" }, { "method": "PUT", "path": "vehicles/TCVP60_7/stop", "body": "true" }, { "method": "GET", "path": "settings/system/logLevelFile", "body": "" }]</pre>
Response code	207	
Response body	<pre>[RESPONSE_1, RESPONSE_2, RESPONSE_N]</pre>	<pre>[204, 204, "FINE"]</pre>

Subscription Service

mC's REST server can only react to requests. For clients who want to be automatically informed about changes to certain resources - e.g. vehicle movement or job progress and completion - a subscription service via WebSocket is available. To subscribe please follow these steps:

1. Subscribe to the desired resources via REST API request
Method: POST; Path: [BASE_URL]/subscription; Body: [RT1, RT2, ...]
mC creates the authentication token required for step 2 and transmits this in the location header of the response
2. Open a WebSocket connection with the following URL
`wss://[BASE_URL]/[token]`
3. [OPTIONAL] Subsequent change of subscription
Method: PUT; Path: [BASE_URL]/subscription/[token]; Body: [RT1, RT2, ...]

After a successful connection, updates of the subscribed resources are sent to the client in time with the core loop (default every 100 ms). The first update contains the complete lists, all other updates contain only the changes to the previous packet.

Delta Update Structure

<pre>{ "serverTime": 1655883489713, "serverClock": 100, "vehicles": [{ "id": "TCVKS001", "posX": -2035.0, "posY": -420.0, "posH": 88.441, "conDist": 1830.0, "mileage": 2869789.682, "speedLimit_": null }, { "id": "TCVKS002" }], "scripts": [{ "id": "jamInspector", "load": 4.0, "tasks": 1.0 }] }</pre>	<p>current serverTime in milliseconds since January 1, 1970 UTC current serverClock in ms; field always present list of all changed resources of named type will follow</p> <p>ID field is always present object contains all changed fields of this resource the vehicle is obviously moving...</p> <p>null value indicates that the corresponding field was deleted</p> <p>object with only ID field indicated that resource was deleted</p> <p>resource changes of next subscribed type are following...</p>
---	---

Common Resource Fields

Resources in mC v4 are structured like maps that can have any number of key-value-pairs. Despite this flexibility, logical modules must agree on a common understanding of the properties of each resource type. Each module / script setup can have its own key system, but many fields are constant for most setups. These common fields are described in the table below:

Usage	Key	Example Value	Description
mandatory system fields for all resource types	id	TCVKATE1	ID of the resource; can't be changed
	timeCreate	1653899750781	time of creation in milliseconds since January 1, 1970 UTC
	timeUpdate	1653916635483	last modification time in milliseconds since January 1, 1970 UTC
	responsibility	GSF_SBE	ID of user who last modified the resource
fields of resources to be displayed on the map of the UI	model	KATE	ID of model resource whose shape definition will be used for map display
	posX	1000	x coordinate of the absolute position of the center of the resource's shape in mm
	posY	2000	y coordinate of the absolute position of the center of the resource's shape in mm
	posZ	30	z coordinate (height above floor) of the center of the resource's shape in mm
	posH	45	heading angle (compass direction) of the resource's shape in degrees (0-360); 0° means shape pointing towards positive x-axis; angle increases according to math quadrants (counterclockwise)
	posRef	TCVFORK1	if set, map position will be calculated relative to position of referenced resource
	hidden	false	selects if resource's shape is visible or hidden on the map
	points	[{ "x": -1000, "y": 1000 }, { "x": -1000, "y": -1000 }, { "x": 1000, "y": -1000 }, { "x": 1000, "y": 1000 }]	array of polygon points with coordinates in mm; coordinates are relative to pos values (posX/posY/posZ/posH) of resource if set or absolute if pos values are not set

	pointerEvents	stroke	works like CSS property "pointer-events" for resource shapes
vehicle resource: user input	sizeX	770	length of vehicle in mm
	sizeY	450	width of vehicle in mm
	offsetPivot	-250	offset from center of vehicle to pivot position in mm
	offsetProgress	250	offset from center of vehicle to transmitted progress point in mm
	offsetOdometry	-250	offset from center of vehicle to transmitted odometry point in mm
	rangeOfSightMin	1000	minimal range of sight = length of envelope in front of vehicle at 0 % speed in mm
	rangeOfSightMax	2000	maximal range of sight = length of envelope in front of vehicle at 100 % speed in mm
	speedMax	1	maximum speed of vehicle in m/s
	speedAverage	0.5	average speed of vehicle in m/s
	routeFreeze	2	number of route segments that remain unchanged in case of route/job changes; default value 2
	envelopeStepSize	200	number of calculation steps of envelope (= resolution) in mm; default value 200 mm
	maxErrorsLength	10	maximum number of events in error history
vehicle resource: readonly fields	lastWaypointId	7	ID of last read/passed waypoint of vehicle
	nextWaypointId	8	next waypoint in route of vehicle
	conDist	1235	traveled distance of connection between last and next waypoint in mm
	chainDrive	true	if true, vehicle drives the specified route
	speed	77	current speed of vehicle relative to speedMax in percent
	speedLimit	100	current speed limit for this vehicle calculated from all sources in percent
	actionCode	-1	current action code of vehicle; default value -1 if vehicle does no action
	actionValue	0	current action value of vehicle; default value 0 if vehicle does no action
	mileage	2833862.281	total mileage in meters
	energy	75	energy of battery in percent; value range 0 to 100
	errors	[0,1,0,0]	current errors of vehicle as array of 1-byte-bitmasks
	errorHistory	[{[EVENT_1]},	error history of vehicle as array of event objects; size limited by maxErrorsLength

		<code>{[EVENT_2]}, {[EVENT_n]}</code>]	
	↳ timestamp	1655795953447	timestamp of error event in milliseconds since January 1, 1970 UTC
	↳ key	dynStopError	name of error event
	↳ value	true	value of error event at this time
	isOnline	true	if true, vehicle currently sends data
	isValidated	true	if true, vehicle sends valid checksum
	deltaTx	99	delta between last two telegram input timestamps in ms (input into mC)
	deltaRx	98	delta between last two telegram output timestamps in ms (output from vehicle)
	rawInput	<code>{ "data": [...], "ip": ..., "port": ..., "timestamp": ... }</code>	object that contains raw input data from vehicle; format depends on protocol
	rawOutput	<code>{ "data": [...] }</code>	object that contains raw output data to vehicle; format depends on protocol
	zoneHits	<code>[{[ZONE_HIT_1]}, {[ZONE_HIT_2]}, {[ZONE_HIT_n]}]</code>	array of area intersections between vehicle base + envelope and zone polygons
	↳ id	zones/75	path (RESOURCE_TYPE/ID) of zone resource
	↳ model	zone_red	ID of model of zone
	↳ center	<code>{ "x": -6609, "y": -2630 }</code>	center coordinate of intersection area
	↳ base	true	if true, vehicle base intersects zone; if false only vehicle envelope intersects zone
job resource: user input	targetKey	TCVKATE	assignment pool filtering: ID of candidates must start with this string; "TCV" will usually catch all vehicles
	targetFilter	energy[gt]=80	assignment pool filtering: see Filter & Search Queries for details
	priority	50	value range 1 to 100; lowest value means highest priority
	endEvent	ARCHIVE	REMOVE (job will be removed on completion) or RESTART (order index will set to 0 and assignment will be removed on completion) or ARCHIVE (job will stay in job list with COMPLETED status)

	appointmentMode	FIXED	FIXED (user defined appointment time) or ASAP (job will be started as early as possible considering the costs; appointment time will be set to ETA at time of assignment)
	appointmentTime	1654770160914	in milliseconds since January 1, 1970 UTC
	appointmentIndex	2	index of order whose completion should be at the defined appointment time
	initRequest	true	if true, job will be resetted including assignment and current progress/orderIndex
	supersedable	true	if true, job can be superseded by another job with property superseding ; assigned target will be transferred to superseding job; status of this job will switch to SUPERSEDED
	superseding	true	if true, job can supersede jobs with property supersedable ; assigned target will be transferred from supersedable job
	orderList	[{[ORDER_1]}, {[ORDER_2]}, {[ORDER_n]}]	list of orders; no length limit; orders will be processed sequentially; each order typically contains the values below
	↳ location	17	ID of destination waypoint
	↳ actionCode	1	nominal code of action; will be transmitted to vehicle; vehicle has to understand its meaning; range and meaning depends on vehicle type
	↳ actionValue	100	nominal value of action; range depends on action code and vehicle type, e.g. value 100 could mean fully charged if code 1 stands for charging action
	↳ actionTime	40	[OPTIONAL] duration of action in seconds; defaults to 0
job resource: readonly fields	targetId	TCVKATE1	full vehicle ID of assigned vehicle or empty string if job is not assigned
	status	ASSIGNED	current job status; possible value: SCHEDULED → cost calculation ongoing ASSIGNED → vehicle is doing this job DELAYED → ETA later than expected COMPLETED → all orders done DEFERRED → nothing to do right now SUSPENDED → job out of assignment limit SUPERSEDED → job superseded by other job NO_TARGET → can't find vehicle NO_ROUTE → assigned vehicle lost route
	statusHistory	[{[ENTRY_1]}, {[ENTRY_2]}, {[ENTRY_n]}]	array of states with timestamp; contains up to 10 entries
	↳ time	1654777485560	timestamp of switching to this status in milliseconds since January 1, 1970 UTC

	↳ status	COMPLETED	value of status
	orderIndex	1	current order; value range 0 to order list length minus 1
	progress	0.935	current progress of job (all orders) as percentage; value range 0 to 1
	eta	1654771180418	estimated time of arrival (completion of last order including action) in milliseconds since January 1, 1970 UTC
	duration	93388	duration of job (all orders) in milliseconds
	deviation	-9084	deviation of job (all orders); delta between appointment time and ETA
	deviationRatio	-0.097	current deviation divided by duration; job status switches to DELAYED if ratio greater than 0.25
	forecasts	[{{FORECAST_1}}, {{FORECAST_2}}, {{FORECAST_n}}]	array of forecasts; contains one forecast for each vehicle; forecasts will be calculated only before assignment
waypoint resource: user input	penalty	1000	length in millimeters that is added to the length of the link that leads to this waypoint in the cost calculation of the routing algorithm
link resource: user input	penalty	7500	length in millimeters that is added to the length of this link in the cost calculation of the routing algorithm
	closure	true	if true, link is closed = not available for route finding
	speed	30	speed limit for this link in percent; default value 100 if field is not present
	fulcrums	[{ "x": 48995, "y": 21710, "h": 180 }, { "x": 47995, "y": 22710, "h": 90 }]	list of points (coordinates in mm) with heading angle that define fulcrums of the link between two waypoints; these points are not control points of a spline but positioned directly on the link and have their own pair of control points
route resource: readonly fields	jobId	22	ID of job resource whose current order is linked to this route
	waypointIds	["1", "9", "8", "7"]	list of waypoint IDs; vehicles must find its current lastWaypointId inside this array in order to synchronize to route
	length	[4000, 1621, 2000]	list of link lengths in mm
	costs	[8000, 1621, 2000]	list of costs per link; costs = length * (100 / speed) + penalties;

			e.g. 50 % speed limit equals double length
event resource: user input	type	WARNING	type of message; type determines color and display order (more severe events will displace less severe ones); types can be defined in settings/tableColumns/events; default types: WARNING → yellow ERROR → red CORE → pink
	message	TCSLP002 is offline	message of event; should be not too long
	popup	true	if true, event's message will show as popup in UI if unchecked
	checked	false	if true, event will be marked as read