webfleet

WEBFLEET.connect

1.76.0

Reference Guide

BRIDGESTONE Fleet Care

1586372083200 - 1.32 - 2026-10-08

Contents

What's new? 8

Welcome 10

Terminology 11

Webfleet components 12
Webfleet 12
LINK 12
Connected Driver Terminals 12
Non-connected Driver Terminals 12
Remote LINK 13
LINK 105/ecoPLUS 13
Object 14
Tracking and tracing 15
Tracking 15
Tracing 16
Trip 17
Trip with LINK device 17
Trip with navigation device only 17
Standstill 18
Idle time 19
Event 20
Driving event 21
Order 22
Working time 23
Working time reporting across vehicles or devices 23
Vehicle maintenance 24
Maintenance schedule 24
Maintenance task 24
Additional features 25
HD Tracking I 25
HD Tracking II 25
HD Tracking III 25
Direct FMS 25
Webfleet-compatibility upgrade (TomTom PRO 5150/5250) 26
Webfleet additional reporting package 26
Webfleet for Salescloud 26
Webfleet Plugin 26
Webfleet Video 26
Webfleet TPMS 27
Professional (Truck) Navigation 28
Order optimisation 28
Webfleet Cold Chain 28
LINK.connect 29
Webfleet Tachograph Manager services 29
Remaining driving times (TACHO.RDT) 30
Webfleet TachoShare services 31
Webfleet Hours of Service 32
Webfleet Road Usage Charges (TRUC) 33

Programming Guide 34

Introduction to WEBFLEET.connect 35

Preparing for WEBFLEET.connect 36
Access to WEBFLEET.connect with API Key 36
Checking requirements 36
Creating a user and assigning rights using Webfleet 36
Making requests to WEBFLEET.connect 38
Making HTTP requests 38
Getting started with HTTP requests 40
HTTP request encoding 41
Making SOAP requests 42
Getting started with SOAP requests 44
SOAP parameters 46
Request limits 49
Common parameters 49
Authentication parameters 49
General parameters 50
Date range filter parameters 52

Reference 54

Message queues 55
Overview of queues 55
createQueueExtern 55
deleteQueueExtern 56
popQueueMessagesExtern 57
ackQueueMessagesExtern 68
Message types 69
JSON objects 82
Parameters specific to acc_events 83
Parameters specific to app_data_in 84
Parameters specific to app_data_out_status 84
Parameters specific to aux_device_data_quota 84
Parameters specific to app_data_quota 84
Parameters specific to app_data_reference 84
Parameters specific to aux_device_data_in 85
Parameters specific to aux_device_data_out_status 85
Parameters specific to aux_device_data_reference 85
Parameters specific to ep_paireddata 85
Parameters specific to ep_snapshotdata 85
Parameters specific to aux_device_data_in 86
Parameters specific to fms_fuel_difference 86
Parameters specific to fms_snapshotdata 86
Parameters specific to fms_tripdata 86
Parameters specific to troublecodes 87
Parameters specific to er_tripdata 87
Objects 88
showObjectReportExtern 88
showVehicleReportExtern 94
showNearestVehicles 99
showContracts 101
updateVehicle 103
showObjectGroups 109
showObjectGroupObjects 110
attachObjectToGroup 111
detachObjectFromGroup 112
insertObjectGroup 113
deleteObjectGroup 113
updateObjectGroup 114
switchOutput 114
showWakeupTimers 115
updateWakeupTimers 117

getObjectFeatures....118
updateContractInfo....121
getObjectCanSignals....122
getObjectCanMalfunctions....124
getElectricVehicleData....126
getActiveAssetCouplings....128
getTelemetryDeviceBatteryState....130
getObjectDTCs....131
Orders....133
About orders....133
sendOrderExtern....133
sendDestinationOrderExtern....135
updateOrderExtern....144
updateDestinationOrderExtern....145
insertDestinationOrderExtern....151
cancelOrderExtern....156
assignOrderExtern....156
reassignOrderExtern....158
deleteOrderExtern....159
clearOrdersExtern....160
showOrderReportExtern....161
showOrderWaypoints....166
Parameters in wp....168
getOrderStatusPageURL....169
removeOrderStatusPageURL....170
downloadOrderAttachment....171
downloadOrderAttachmentThumbnail....171
uploadOrderAttachment....172
getOrderAttachmentsMetadata....173
deleteOrderAttachment....174
getOrderEpod....175
downloadOrderEpodSignature....176
downloadOrderEpodPhoto....177
createOrderLoadedGoods....178
updateOrderLoadedGoods....179
deleteOrderLoadedGoods....180
getOrderLoadedGoods....181
Messages....183
sendTextMessageExtern....183
clearTextMessagesExtern....183
showMessages....184
sendBinaryMessage....186
resetBinaryMessages....187
clearBinaryMessages....188
Drivers....190
Synchronising driver lists....190
showDriverReportExtern....190
insertDriverExtern....194
updateDriverExtern....197
deleteDriverExtern....200
showOptiDriveIndicator....201
showDriverGroups....205
showDriverGroupDrivers....206
attachDriverToGroup....207
detachDriverFromGroup....208
insertDriverGroup....209
deleteDriverGroup....209
updateDriverGroup....210
attachDriverToVehicle....210
detachDriverFromVehicle....211

getDriverRdtRules ... 212
updateDriverRdtRules ... 215
getOptiDriveProfiles ... 219
Addresses ... 222
showAddressReportExtern ... 222
showAddressGroupReportExtern ... 224
showAddressGroupAddressReportExtern ... 225
insertAddressExtern ... 226
updateAddressExtern ... 229
deleteAddressExtern ... 231
attachAddressToGroupExtern ... 232
detachAddressFromGroupExtern ... 232
insertAddressGroupExtern ... 233
deleteAddressGroupExtern ... 234
Events ... 235
showEventReportExtern ... 235
acknowledgeEventExtern ... 238
resolveEventExtern ... 238
getEventForwardConfigs ... 239
getEventForwardConfigRecipients ... 240
insertEventForwardConfig ... 241
updateEventForwardConfig ... 242
deleteEventForwardConfig ... 244
Trips and working times ... 245
showTripReportExtern ... 245
showTripSummaryReportExtern ... 252
showTracks ... 253
updateLogbook ... 255
showLogbook ... 256
showLogbookHistory ... 261
updateLogbookMode ... 263
updateLogbookDriver ... 264
showWorkingTimes ... 264
showStandStills ... 268
showIdleExceptions ... 271
getObjectKPIs ... 274
getDriverKPIs ... 276
getRemainingDrivingTimesEU ... 278
KPI names ... 284
Miscellaneous reports ... 287
showIOReportExtern ... 287
showAccelerationEvents ... 288
showSpeedingEvents ... 293
showDigitalInputStateMileage ... 296
getChargerConnections ... 297
getCrashLog ... 298
getLoadData ... 301
Geocoding and routing ... 302
geocodeAddress ... 302
calcRouteSimpleExtern ... 304
Configuration and security ... 308
showSettings ... 308
createSession ... 311
terminateSession ... 312
showAccountOrderStates ... 313
updateAccountOrderState ... 315
showAccountOrderAutomations ... 317
updateAccountOrderAutomation ... 318
getAccountStatusMessages ... 319
getStatusMessages ... 320

setVehicleConfig ... 321
getVehicleConfig ... 322
setStatusMessages ... 323
setAccountStatusMessages ... 325
User management ... 327
showUsers ... 327
changePassword ... 329
insertUser ... 329
Profile and interface style ... 331
updateUser ... 332
deleteUser ... 334
getUserRights ... 335
setUserRight ... 336
resetUserRights ... 337
removeUserRight ... 338
Supported right levels ... 339
Vehicle Maintenance ... 342
insertMaintenanceSchedule ... 342
updateMaintenanceSchedule ... 345
deleteMaintenanceSchedule ... 348
showMaintenanceSchedules ... 348
showMaintenanceTasks ... 352
resolveMaintenanceTask ... 356
getDiagnosticFaults ... 357
Reporting ... 359
getArchivedReportList ... 359
getArchivedReport ... 360
deleteArchivedReport ... 360
getReportList ... 361
createReport ... 362
sendReportViaMail ... 363
Areas ... 365
getAreas ... 365
insertArea ... 367
deleteArea ... 370
updateArea ... 371
getAreaPoints ... 374
getAreaAssignments ... 376
insertAreaAssignment ... 377
deleteAreaAssignment ... 379
getAreaSchedules ... 379
insertAreaSchedule ... 381
deleteAreaSchedule ... 382
LINK.connect ... 384
API key mandatory ... 384
sendAuxDeviceData ... 384
getLocalAuxDeviceConfig ... 385
configureLocalAuxDevice ... 386
getRemoteAuxDeviceConfig ... 388
configureRemoteAuxDevice ... 389
removeRemoteAuxDeviceConfig ... 391
clearAuxDeviceDataQueue ... 391
resetAuxDeviceData ... 392
Plugins ... 394
insertExternalEvent ... 394
setExternalObjectData ... 395
Cold Chain ... 396
getCurrentColdChainSensorData ... 396
getHistoricalColdChainSensorData ... 398
getCurrentColdChainStateData ... 401

getHistoricalColdChainStateData....404
getCurrentTemperatureData....407
getHistoricalTemperatureData....409
getCurrentRefrigeratedDoorStatusData....411
getHistoricalRefrigeratedDoorStatusData....413

Appendix A Operation response codes....416

Response codes....417
Response codes - Queues....439

Appendix B: Resources....441

WEBFLEET.connect resources....442
Other resources....443

Appendix C: Superseded functionality....444

Appendix D: Solving known issues....447

Using WEBFLEET.connect SOAP with Visual Basic....448
SOAP message size quota in Visual Studio....449

Appendix E: Using MTOM with SOAP....450

MTOM in the WEBFLEET.connect SOAP API....451
Defining binary data in WS parameters....451
Java Client Code Generation....451
Activating MTOM depending on data size....453
Java – server side....453
Java – client side....453
.NET....453
Testing with soapUI....453

Appendix F CAN signal types....454

CAN signal types....455

Revision history....463

What's new?

Here you find an overview of what is new or has changed in WEBFLEET.connect 1.76.0.

Please find the full revision history at the end of this document.

Deprecation of URL credentials

Support for credentials (username, password) in URL query parameters has been removed **in July 2026**. **Requests with URL credentials will not work anymore.**

Use HTTP Basic Auth instead. This is supported since September 2022. See Making HTTP requests

This only affects the CSV/JSON API. SOAP is unaffected.

Speed limit settings

Action showSettings now returns separate speed limits `speedlimit_urban`, `speedlimit_extraurban` and `speedlimit_motorway`. These are the same values as available in the Webfleet UI.

The old setting `speedlimit` is still returned by the API for backward compatibility. But this value is not used anymore.

Electric vehicle driving KPIs

A new KPI ev_driving_indicator is available for actions showOptiDriveIndicator and showTripReportExtern.

Action getOptiDriveProfiles returns the configured weight for EV driving behaviour in new response field `ev_driving_kpi_weight`.

Welcome

Welcome to the WEBFLEET.connect Reference Guide.

This document provides all the information you need to integrate Webfleet into your applications using the WEBFLEET.connect interface.

The documentation is divided into the following sections:

- Terminology – In this chapter you learn understanding terms used in the context of Webfleet and WEBFLEET.connect products and features.
- Programming Guide – The programming guide contains a detailed description of how to submit requests to WEBFLEET.connect and the data that is returned by the service, including an explanation of how to enable access to the service.
- Reference – The Reference is a detailed description of all available operations, including their parameters and the data returned by these operations.

More information

Updated versions of this documentation and other resources are available online at https://www.webfleet.com/webfleet/partners/integration/developer-resources/

Terminology

Here is a description of the terms used in the context of fleet management and Webfleet.

Webfleet components

There are three variations of Webfleet solutions:

- Webfleet connected to a LINK device.
- Webfleet connected to a LINK device and a Driver Terminal.
- Webfleet connected to a Driver Terminal.

Webfleet

Webfleet is Bridgestone Mobility Solutions' innovative Software as a Service (SaaS) fleet management software solution. It gives you secure access to all the information you need to manage your fleet operation in the most effective way. Webfleet is the web-based application for managing vehicles and orders and accessing Webfleet tracking and tracing functionality.

LINK

The LINK is a black box installed in the vehicle. The LINK is connected to the vehicle's power supply and the ignition. It registers ignition events, vehicle movement, and position information and sends all this information to the office. Connected via GPRS to Webfleet, it also enables communication between the office and a Driver Terminal. This device enables tracking and tracing.

Different models of the LINK are supported in Webfleet:

- LINK 7xx/6xx
- LINK 5xx/4xx
- LINK 3xx
- LINK (first generation)

Connected Driver Terminals

The term **Connected** refers to all Driver Terminals that come with a built-in mobile network modem. Driver Terminals do not need to establish a connection to a LINK but directly connect to Webfleet instead. There are different generations of connected Driver Terminals currently supported in Webfleet:

- PRO M
- TomTom PRO 5350/5250 (purchase of Webfleet-compatibility upgrade required)

Non-connected Driver Terminals

The term **Non-connected** refers to all Driver Terminals that need a LINK device for communication with Webfleet and the office. There are different generations of non-connected devices currently in use:

- PRO 84xx/83xx
- PRO M
- PRO 2020
- TomTom PRO 8275/8270/7350/7250

Remote LINK

Remote LINK is a remote control. It needs to be connected using Bluetooth® to an associated LINK 510/300/310 installed in a vehicle.

There are two variations:

- Remote LINK Working Time

  With Remote LINK Working time multiple employees are able to register their working times via the LINK 5xx/4xx/3xx.
- Remote LINK Logbook

  With Remote LINK Logbook, drivers are able to maintain a logbook via the LINK 5xx/4xx/3xx.

LINK 105/ecoPLUS

LINK 105/ecoPLUS™ retrieves fuel consumption information over the vehicle's OBD-II connector and sends this information to Webfleet in real time. LINK 105/ecoPLUS works together with the LINK 5xx/4xx using Bluetooth®.

Object

An object is the central entity in Webfleet. An object can be a vehicle like a car, truck or trailer. The vehicle or trailer is installed with a Webfleet unit. The Webfleet unit delivers all relevant vehicle/trailer-related information to Webfleet such as position, time, speed etc. The object is then shown on the map and in the respective lists in Webfleet. It is created in the Webfleet database as soon as it is activated after a successful installation.

Tracking and tracing

Tracking and tracing is the general term used for tracking the position of an object, either in real time or historically. Usually this includes functionality to communicate with vehicles and their drivers.

Tracking and tracing requires dedicated hardware and software. Webfleet uses both built-in and portable devices.

Built-in devices

24/7 monitoring and security is enabled by using built-in LINK devices.

The LINK device delivers position, movement and ignition information as well as input events, for example engine is running etc., related to the specific vehicle and cannot be switched off.

The LINK device normally sends six positions per minute as long as the vehicle is operated in countries covered by your Webfleet subscription. The countries and the position report interval vary per subscription. For details refer to the Webfleet contract. Using the application programming interface WEBFLEET.connect one position per minute is standard in these countries. You can extend the position report interval for WEBFLEET.connect integrations to more positions per minute by booking the Additional Service HD Tracking. Additional positions are then available in the 'surplus_data' result parameter as 'hd_tracking-positions' incl. geo coordinates, speed, course and time. For more details how to book this feature please refer to the Additional Service form.

A LINK device installed in the vehicle is required for additional devices such as the following:

- Non-connected navigation devices
- Remote LINK
- LINK 105/ecoPLUS
- Digital tachograph

For compatibility see www.webfleet.com.

These devices have basic functions like sending and accepting orders and messages, reporting working times, logbook-keeping, reporting on driver behaviour, driving times of driver and co-driver and fuel consumption.

Portable devices

Part-time tracking and tracing can be achieved with Connected navigation devices that directly communicate with Webfleet.

Tracking

Tracking is monitoring the position, movement, driving and other events of a vehicle in real time. The vehicle reports its current speed and the direction in which it is moving to Webfleet. Additionally you can see if a vehicle is stuck in a traffic jam, if it is parked and/or if it is idling the engine. This kind of information is available in Webfleet, for example on the map and in the vehicle details panel.

Tracing

Tracing is the ability to see the vehicles' positions, movements, driving and other events historically. The information is stored to Webfleet for up to 90 days, and can be evaluated during this time.

Trip

A trip is a movement of a vehicle from a start position to an end position.

After a trip has ended a trip report is sent to Webfleet containing all trip-relevant information such as start time, end time, stand stills, idling times, and fuel consumption.

Trip with LINK device

A trip start is reported when the vehicle's ignition is switched on and the vehicle moves or changes its GPS position for a minimum configured duration (default five min.), retroactive to vehicle ignition time.

The end of a trip is reported when one of the following conditions applies:

- The vehicle's ignition is switched off and remains so for a minimum configured duration (default 15 min.), and the vehicle does not move or change its GPS position.
- The logbook trip mode changes from private trip to business trip or vice versa.
- The driver changes, reported either through the Driver Terminal or the digital tachograph.

Trip with navigation device only

When using a connected navigation device, trips are reported under the following conditions:

The start of a trip is reported when all the following conditions apply to the navigation device:

- It is properly connected to the dock in the vehicle.
- It has GPS reception.
- It changes its GPS position.

The end of a trip is reported when one of the following conditions applies to the navigation device:

- It is removed from the dock.
- It has been switched off.
- It stands still for a minimum of ten minutes.
- The logbook mode has changed from business to private mode or vice versa.
- The driver changes, reported through the Driver Terminal (applicable for GO and PRO navigation devices with the software version 8.390.1916 or higher only).

Standstill

A standstill is reported when the vehicle stops moving or changing its GPS position for minimum configured duration (default five min.). The end of a stand still is determined when the vehicle starts moving or changing its GPS position. The complete standstill interval is reported to Webfleet when the standstill has ended.

Idle time

Idle time is the time a vehicle stands still with the engine running or the time the vehicle is moving very slowly. Idling is recorded after five minutes.

Event

Events are incidents recorded on the Driver Terminal that are reported as messages to Webfleet. Typical examples include the following:

- Predefined general status messages.
- Predefined order-related status messages.
- Being within geographical zones.
- Information on disconnection of peripherals, undesired functions, digital input events, etc.

Status messages and geo-zone events can be classified as notice, warning or alarm in three levels according to their importance.

Driving event

A driving event is reported to Webfleet when a driver exceeds a certain level of acceleration. Acceleration is recorded during braking, cornering and steering and exceptional acceleration, such as car crashes.

Order

An order is a message containing instructions about the job to be done. There are three types of orders:

- Delivery order.
- Pickup order.
- Service order.

An order contains an order number and a free text message up to 500 bytes.

Optionally, the following can be attached to an order:

- A contact.
- A phone number for the contact.
- The planned date of execution.
- A destination address to be displayed to the driver on the Driver Terminal.
- Destination coordinates to be used by the Driver Terminal for route calculation to the job location.
- The desired time of arrival.
- A time tolerance for the start of the execution to generate an alarm message if the tolerance is exceeded.
- A lead time for an arrival message.

The order instructions appear on the navigation device and can be accepted, rejected or simply saved for later. The (co-)driver can accept or refuse an order and start and complete activities on the order by using the corresponding order steps. If the driver rejects or cancels the order, he is asked to type in a reason.

All actions on the order taken by the driver generate messages sent to Webfleet.

Working time

Working time can be reported to Webfleet using:

- A connected navigation device; the driver can report working time, by identifying himself to the connected navigation device.
- A navigation device in combination with a LINK device; the driver can report working time, by identifying himself to the navigation device.
- A Remote LINK Working Time together with a LINK device; multiple employees can report working time via one installation, as they identify themselves through a personalised ID Key.

The following working time states can be reported to Webfleet:

- Commute to work.
- Start break.
- Start work, also indicates end of a break if the last reported status was start break.
- End work, also indicates end of a break if last reported status was start break.

Working time reporting across vehicles or devices

Generally, when reporting working time, the start and the end are performed on the same device, but if drivers and workers use various vehicles and thus devices, the following rules should be considered:

- If a driver moves between PRO navigation devices or Remote LINK devices working times are driver-oriented.
- If a driver moves between GO navigation devices, working times are vehicle-oriented.
- Working sessions have to be closed, with **End work**, before pairing any device with a different device.
- Closing working sessions on a different device than the one with which you have started the working session is only possible, if you use Remote LINK with an ID Key or PRO navigation devices.

Vehicle maintenance

Maintenance schedule

A maintenance schedule describes a maintenance task that has to be carried out on a vehicle. It defines when the task is scheduled and whether it has to be carried out once at a certain date or vehicle mileage, or if it needs to start at a certain date or vehicle mileage and be repeated at regular intervals.

In accordance with the maintenance schedule, Webfleet creates a maintenance task when the task is due. A reminder can also be issued a certain time or distance before the task is due, giving advanced notice of the maintenance task.

Maintenance task

A maintenance task for a vehicle is created from the respective maintenance schedule. The task appears in Webfleet when it is due or the task's reminder is issued.

Additional features

WEBFLEET.connect comes with a large variety of features and functionality that help you drive your business out of the box. For industry specific needs Webfleet offers a range of additional features/additional services. These can be booked on top of your WEBFLEET.connect subscription.

HD Tracking I

Charged per: Device

Due upon activation: Monthly

Minimum contract period in months: 1

Extension for WEBFLEET.connect integrations to include all positions sent within the tracking message.

HD Tracking II

Charged per: Device

Due upon activation: Monthly

Minimum contract period in months: 1

Change of position report interval from 6 positions/minute (default) to 5 positions/15 seconds for the device.

HD Tracking III

Charged per: Device

Due upon activation: Monthly

Minimum contract period in months: 1

Change of position report interval from 6 positions/minute (default) to 5 positions/15 seconds for the device; and additionally extension for WEBFLEET.connect integrations to include all positions sent within the tracking message (combination of HD Tracking I & II).

Not supported by the following devices: LINK 300, LINK 310

Direct FMS

Charged per: Device

Due upon activation: Monthly

Direct FMS enables the required data for Webfleet FMS features (for example OptiDrive, fuel consumption, cruise control, AdBlue level) by connecting a LINK 710 directly to the CAN bus of a compatible heavy commercial vehicle.

Webfleet-compatibility upgrade (TomTom PRO 5150/5250)

Charged per: Device

Due upon activation: Once

Upgrades the TomTom PRO 5150/5250 LIVE TRUCK to connect to WEBFLEET.connect.

Webfleet additional reporting package

Charged per: Webfleet account

Due upon activation: Monthly

Extension of the number of reports that can be stored in the Webfleet reports Archive by 10 additional reports. The period of validity of the service is determined by the contract period of the corresponding Webfleet account.

Webfleet for Salescloud

Charged per: Device

Due upon activation: Monthly

Seamless integration with Salesforce enables you to compare driven mileage with sales performance. Other insights include: single trip data, purpose, number of trips, driving performance information and arrival times. By syncing Webfleet to your Salesforce calendar, you can send appointments to your Driver Terminals in real time.

Webfleet Plugin

Charged per: Device

Due upon activation: Monthly

With a Webfleet Plugin data from an external source can be displayed in Webfleet on an existing Webfleet object. A Webfleet.connect integration is required.

Webfleet Video

Webfleet Video Plus

Charged per: Device

Due upon activation: Monthly

Minimum contract period in months: 1

Service as Webfleet Video with increased functionality of monitoring driver and their behaviour with the use of AI technology and the ability to request video on demand.

Available with Webfleet LINK subscription or higher. Excludes TomTom PRO 5350 Live subscription

Webfleet Video Live

Charged per: Device

Due upon activation: Monthly

Minimum contract period in months: 1

Service as Webfleet Video Plus with increased functionality of the ability to Live stream video content of the device via Webfleet for both the road and driver facing views. Also supports the use of auxiliary cameras for monitoring side and rear views of a vehicle.

Available with Webfleet LINK subscription or higher. Excludes TomTom PRO 5350 Live subscription

Functionality	Video Plus	Video Live
LINK based events	✓	✓
Dual video view (street- and cabin-facing)	✓	✓
Alert button events	✓	✓
Video events inbox (notifications)	✓	✓
Deleting video events	✓	✓
Video tab in the Vehicles view	✓	✓
On-demand request for video events	✓	✓
Camera based events (AI events)	✓	✓
Live streaming video		✓
Auxiliary camera support		✓

Webfleet TPMS

Webfleet TPMS

Charged per: Vehicle (for example: truck, bus, trailer)

Due upon activation: Monthly

Minimum contract period in months: 1

Enables Tyre Pressure Monitoring (TPMS) functionality in Webfleet, supporting live tyre pressure updates, notifications and maintenance tasks. Available with Webfleet ADM or Webfleet AST subscription for trailers and with Webfleet ECO subscription or higher for all other vehicles. A connection to a LINK 7x0 is required for all vehicles except trailers. For trailers, the LINK 7x0 in the towing vehicle is required for connectivity, unless the trailer is fitted with the LINK 350 and associated TPMS sensors.

IMPORTANT! TPMS for trailers – Live TPMS functionality is currently only supported for trailers when the trailer is used together with a towing vehicle fitted with Webfleet TPMS, unless the trailer is fitted with the LINK 350 and associated TPMS sensors.

Professional (Truck) Navigation

Charged per: Subscription

Due upon activation: Monthly

Minimum contract period in months: 1

Enables the use of TomTom GO Fleet in combination with Webfleet Work App. It includes best-in-class (truck) navigation, LIVE traffic, an automated reporting of ETA and destination to Webfleet, and more.

Only available for NAV and LIVE subscriptions. Can only be used in combination with Webfleet Work App. Requires an active data con-nection (SIM or Wi-Fi). Data consumption depends on usage and region, among other factors. The app should only be used with unlimited data plans or plans with automatic cost limits to avoid unexpected costs due to increased data consumption.

Order optimisation

Charged per: Device

Due upon activation: Monthly

Minimum contract period in months: 1

Order optimisation features allow you to increase the productivity of your field force by calculating the optimal sequence of orders per vehicle.

Only available for Webfleet subscriptions NAV and LIVE.

Can be used with:

- PRO 8475/8375/8275/8270/7350/7250/5350/5250.
- Webfleet Work App.

Note: Some of the above listed devices may not be available in your country.

Webfleet Cold Chain

Charged per: Device

Due upon activation: Monthly

Minimum contract period in months: 1

Webfleet Cold Chain provides real-time temperature monitoring in Webfleet, offering a new dedicated module, with specific notifications based on temperature thresholds that can be completely configured by Webfleet users. Temperature data is archived in Webfleet for 1 year and can be accessible from the new Cold Chain module in Webfleet or via the new temperature report.

The customer must choose either of the following 2 options as data source for temperature monitoring:

- Connecting 1-Wire (DS1820) sensors to the LINK device*
- Integrating temperature data from Thermo King Tracking

This new service is supported by LINK 340, LINK 710 and LINK 740 devices.

Available for Webfleet LINK subscription or higher (LINK 7X0) and Webfleet AST-M or higher (LINK 340).

**1-Wire sensors are not included in the Webfleet Cold Chain subscription and must be purchased separately. Up to 6 sensors can be connected in one LINK device. Only sensors purchased from Webfleet must be used.*

***The customer must have a valid Thermo King Tracking subscription available directly from Thermo King. Webfleet does not arrange Thermo King Tracking subscriptions on behalf of the customer, and is not responsible for the Thermo King Tracking service.*

LINK.connect

Charged per: Device

Due upon activation: Monthly

Supports the LINK to connect to third party devices using Bluetooth. Only available for Webfleet subscription ECO or higher. Can be used with LINK 410/510/530.

The particular fee includes the costs for the booked GPRS data volume using LINK.connect.

Webfleet Tachograph Manager services

Webfleet Tachograph Manager (TACHOMANAGER)

Charged per: Device

Due upon activation: Monthly

Minimum contract period in months: 1**

Support of downloading, archiving and analysis of tachograph information from driver card and vehicle mass storage. Includes Remote download and Manual download. Remote download frequency: weekly for driver card and monthly for vehicle unit mass storage. Manual download is free of charge for the object with TACHOMANAGER subscription.

Available for Webfleet ECO subscription or higher.

Webfleet Tachograph Manager PLUS (TACHOMANAGER PLUS)

Charged per: Device

Due upon activation: Monthly

Minimum contract period in months: 1**

Service as TACHOMANAGER with increased frequency remote download schedule and Remaining Driving Times. Remote download schedule; daily for driver cards and weekly for the vehicle unit mass storage.

Available for Webfleet ECO subscription or higher.

Manual download (TACHOMANAGER.MDL)

Charged per: Device

Minimum contract period in months: 1***

Manual download from digital tachographs that do not support remote download or are not connected to a LINK device. Does not include remote download.****

The prices listed in the table do not include the purchase of any hardware or the usage of the Webfleet service. The current terms and conditions for the Webfleet service and for the additional feature Webfleet Tachograph Manager apply.

Terms and conditions for the additional feature TACHOMANAGER services

All TACHOMANAGER subscriptions are determined by the Webfleet subscription for the respective object. For example, when you purchase TACHOMANAGER for a Webfleet object that has a WF-DE-EU-LIVE subscription, the TACHOMANAGER price for EU will be charged. Changing the Webfleet subscription consequently results in a change of the TACHOMANAGER/TACHOMANAGER PLUS price.

To use TACHOMANAGER services, please send your company card to Webfleet Sales Support to the address shown in the bottom right corner of your Webfleet service contract.

If you are located in Germany, Austria, Switzerland, Poland or Czech Republic please send your company card to:

Webfleet, Inselstraße 22, 04103 Leipzig, Germany

IMPORTANT! Company cards that are only valid for three months or less cannot be accepted and will be sent back. Please make sure that your company card is valid for a minimum of three more months.

** Cancellation of any TACHOMANAGER additional features for this article is required in writing sent to Webfleet Sales Support.

*** The additional feature TACHOMANAGER.MDL for manual download is invoiced on a pay-per-activation base - Every vehicle that has been active within a month will be invoiced for the entire month. Determining for the invoicing is the activation status of a vehicle in a month not whether a user uploaded a file or not. Deactivation of TACHOMANAGER for manual download does not require a written cancellation. The vehicle can be deactivated by deselecting the vehicle in the list in the Webfleet Tachograph Manager interface. The deactivation between two download dates is not supported - If a vehicle is re-activated, the full period in between will be charged.

**** Manual download (TACHOMANAGER.MDL) will be automatically charged for vehicles that have been manually created in the Webfleet Tachograph Manager interface by uploading tachograph files.

Remaining driving times (TACHO.RDT)

Charged per: Device

Due upon activation: Monthly

Minimum contract period in months: 1

Assists fleet managers to determine remaining driving times.

Terms and conditions for the additional feature TACHO.RDT

Cancellation of TACHO.RDT is required in writing sent to Webfleet Sales Support. Every unit for which TACHO.RDT has been booked will be invoiced for the entire month.

Disclaimer: The remaining driving times supplied by Webfleet are indicative and are reliant upon the information being obtained from the tachograph and sent to Webfleet via the onboard device installed in a vehicle. The algorithms used to calculate the indicative remaining driving times are based on European driving time legislation and it is your responsibility to verify remaining driving times and any applicable national legislation.

Webfleet TachoShare services

Webfleet TachoShare (TACHOSHARE)

Charged per: Device

Due upon activation: Monthly

Minimum contract period in months: 1**

Remote downloading and archiving of tachograph information and the option to share this data with selected analysis providers. Remote download frequency: weekly for driver card and monthly for vehicle unit mass storage.

Available for Webfleet ECO subscription or higher.

Webfleet TachoShare PLUS (TACHOSHARE PLUS)

Charged per: Device

Due upon activation: Monthly

Minimum contract period in months: 1**

Service as TACHOSHARE with increased frequency remote download schedule and Remaining Driving Times***. Remote download schedule; daily for driver cards and weekly for the vehicle unit mass storage.

Available for Webfleet ECO subscription or higher.

Webfleet TachoShare TIS Web Connect (TACHOSHARE.TIS-WEB)

Charged per: Device

Due upon activation: Monthly

Minimum contract period in months: 1**

Remote downloading and archiving of tachograph information and the option to share this data with VDO TIS-Web. Remote download frequency: weekly for driver card and monthly for vehicle unit mass storage.

Available for Webfleet ECO subscription or higher.

Webfleet TachoShare TIS Web Connect PLUS (TACHOSHARE.TIS-WEB PLUS)

Charged per: Device

Due upon activation: Monthly

Minimum contract period in months: 1**

Service as TACHOSHARE.IS-WEB with increased frequency remote download schedule and Remaining Driving Times***. Remote download schedule; daily for driver cards and weekly for the vehicle unit mass storage.

Available for Webfleet ECO subscription or higher.

Terms and conditions for the additional features TACHOSHARE services

All TACHOSHARE subscriptions are determined by the Webfleet subscription for the respective object. For example, when you purchase TACHOSHARE for a Webfleet object that has a WF-DE-EU-LIVE subscription, the TACHOSHARE price for EU will be charged. Changing the Webfleet subscription consequently results in a change of the TACHOSHARE price.

To use TACHOSHARE services, please send your company card to Webfleet Sales Support to the address shown in the bottom right corner of your Webfleet service contract.

If you are located in Germany, Austria, Switzerland, Poland or Czech Republic please send your company card to:

Webfleet, Inselstraße 22, 04103 Leipzig, Germany

IMPORTANT! Company cards that are only valid for three months or less cannot be accepted and will be sent back. Please make sure that your company card is valid for a minimum of three more months.

** Cancellation of the additional feature Webfleet TachoShare for this article is required in writing sent to Webfleet Sales Support.

*** Remain Driving Times is subject to fair use. The subscription to the TACHOSHARE PLUS service is rendered to a Client on a per vehicle basis. It includes remote downloading for two active drivers (driver cards) per month. An active driver is any driver inserting a Driver Card into the tachograph of a vehicle for which the TACHOSHARE PLUS subscription has been purchased. Additional charges may be applied by Webfleet for use outside this fair use policy. For the sake of clarity, we have included the following calculation example concerning the application of the fair use by Webfleet: A Client that has purchased the TachoShare Plus for 20 vehicles may use it up to 40 active drivers per month.

Webfleet Hours of Service

Charged per: Device

Due upon activation: Monthly

Enables Hours of Service registration for the respective object. Drivers with Webfleet Hours-of-Service app installed on a smartphones driving this object can register their duty hours.

Disclaimer:

THE APPLICATION IS PROVIDED TO LICENSEE „AS IS“ AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, Webfleet, ON ITS OWN BEHALF AND ON BEHALF OF ITS AFFILIATES AND ITS AND THEIR RESPECTIVE LICENSORS AND SERVICE PROVIDERS, EXPRESSLY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, WITH RESPECT TO THE APPLICATION, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND WARRANTIES THAT MAY ARISE OUT OF COURSE OF DEALING, COURSE OF PERFORMANCE, USAGE OR TRADE PRACTICE. WITHOUT LIMITATION TO THE FOREGOING, Webfleet PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE APPLICATION WILL MEET YOUR REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF OR LIMITATIONS ON IMPLIED WARRANTIES OR THE LIMITATIONS ON THE APPLICABLE STATUTORY RIGHTS OF A CONSUMER, SO SOME OR ALL OF THE ABOVE EXCLUSIONS AND LIMITATIONS MAY NOT APPLY TO YOU.

Webfleet Road Usage Charges (TRUC)

Charged per: Device

Due upon activation: Monthly

Support of road usage charge calculations within Webfleet.

Programming Guide

This programming guide is an introduction to using the WEBFLEET.connect interface, how to access the service and how to interpret the output that is returned.

In order to access the WEBFLEET.connect service you need a Webfleet account that has WEBFLEET.connect enabled. Otherwise you will not be able to test the integration for your application.

Please talk to your Webfleet sales contact if you do not have access to a WEBFLEET.connect-enabled account.

Introduction to WEBFLEET.connect

WEBFLEET.connect is an API that allows you to access the Webfleet service through a web-enabled application. These are the primary features accessible through WEBFLEET.connect:

- **Reports** — Retrieve data that correspond to the information contained in the reports generated within Webfleet
- **Messaging** — Send text to mobile units and retrieve incoming messages.

  See Message queues and Orders.
- **Addresses** — Insert, update and delete addresses and address groups as well as relations between addresses and address groups.

  See Addresses.
- **Orders** — Insert, send, update and delete orders and retrieve order status information.

  See Orders.
- **Drivers** — Insert, update and delete drivers and retrieve driver status information.

  See Drivers.

Note: As WEBFLEET.connect impersonates as the user that is provided as part of the service request authentication, access restrictions set up within Webfleet apply. This affects all elements of the WEBFLEET.connect interface. For instance, addresses can only be updated if the user has the respective access right to do so (for example: 'Edit access' for 'All addresses').

Preparing for WEBFLEET.connect

WEBFLEET.connect can be made available to every customer with a valid Webfleet account. There should be at least one active object to make full use of the functionality offered by WEBFLEET.connect.

Access to WEBFLEET.connect with API Key

To enable API access for your application, obtain an API key by doing the following:

For .connect partners

If you are a .connect partner, you will receive your API key during your partner application process. To request more API keys, complete the online request form on https://uk.support.webfleet.com/app/ask.

- In the **Refine search by product model** drop-down menu select **Integration**, then select the desired API.
- In the **Refine search by category** drop-down menu select **API key request**.

For customers

If you are a customer and would like to request an API key, complete the online request form on https://uk.support.webfleet.com/app/ask.

- In the **Refine search by product model** drop-down menu select **Integration**, then select the desired API.
- In the **Refine search by category** drop-down menu select **API key request**.

Checking requirements

Geographic coordinates

Some functions require geographic coordinates such as longitude and latitude. This includes inserts and updates of addresses and sending orders.

Make sure that you are able to provide valid coordinates, otherwise you won't be able to fully leverage all functionality that WEBFLEET.connect offers. Geographic coordinates used by Webfleet always refer to the WGS84 coordinate system and have different representations.

Creating a user and assigning rights using Webfleet

In order to access WEBFLEET.connect, you must first create a user within your Webfleet account and give this user the right to use WEBFLEET.connect.

This procedure is outlined below:

1. Select the **Users** icon in the main menu.

2. Click the **New user** button in the lower right.
   The **New user** view opens.

3. In the **User data** section enter the user data.

   Enter the user's name for **Name**, the Webfleet username for the new user under **Username**, and type in a contact email address in the field for **Email**.

   Note: Name, **Username** and **Email** are required.

4. In the section for **Active time period**, select **Unlimited** or define a period by selecting **Temporary**, and then select a start date and an end date using the date picker.
5. Select a user profile in the **Profile** section.

   A good starting point is to use the profile **Standard**.

   Based on the user profile that you select, default settings for access rights are used for this user. You can adjust these settings in the following steps.
6. Select **WEBFLEET.connect** in the **Interfaces** section.
7. In the **System rights** section, select the user access rights regarding **Orders**, **Areas**, and **Reports**.

   You can choose between **Full access**, **View access**, and **No access** for each module. For **Areas**, you can additionally choose the option **View access**, and for **Reports** the option **Download reports only** is available.

   In the following sections you can refine the access rights to specific elements and features within Webfleet.
8. Click **Save** to save your changes.

Making requests to WEBFLEET.connect

Making HTTP requests

This section explains how to use HTTP to issue requests to WEBFLEET.connect.

Important: Only HTTPS requests are accepted. Requests using unencrpyted HTTP are rejected.

WEBFLEET.connect generally uses HTTP GET requests as the underlying transport mechanism for requests (POST is supported for few actions only, see below). All requests are made using specific URLs, passing parameter names and values as URL parameters. Responses are returned as character-separated values (CSV).You can experiment with WEBFLEET.connect-specific HTTPS requests by entering the request URL into the browser's address bar and submitting the request.

WEBFLEET.connect uses the standardized transport protocol HTTPS 1.1 for which compliance with IETF RFC 2616 is very important. It includes proper evaluation and handling of all HTTP response header values, e.g. character set, content and transfer encoding including chunked transfer encoding.

The sequence of incoming messages may differ from the sequence of messages sent from the device. Use their timestamp to restore the sequence, if the sequence is significant for the application.

We highly recommend to use ISO8601 for all date and time values even if its use is optional with some functions. Date and time values carry timezone information where appropriate.

Character sets and date/time values need translation to local configurations, for example. UTF-8 to ISO-8859-10, UTC to CET. We do not guarantee the character encoding (currently: UTF-8) and timezone (currently: UTC for queue service, else time zone of the account) of the web service response, as all information to properly convert this to local requirements is provided as per the above mentioned standards.

For details about time zones when using ISO8601 in CSV read General parameters.

Do not pass parameters with an empty value in a request, if you do not explicitely want to delete the parameter's value.

Note: If the format of the HTTP request is not valid you will get a corresponding error.

Authentication

Authentication method is HTTP Basic Auth according to RFC 7617.

The old way of providing user name and password in the request URL is deprecated and **will be removed end of June 2026**.

For Basic Auth concatenate the Webfleet user name and password with a colon as separator and apply Base64 encoding afterwards. Use the result in the `Authorization` HTTP header. The Webfleet account name is provided as a request parameter as usual. It's not part of the HTTP authorisation.

Example

```
GET /extern?lang=en&account=wfcdevaccountt&apikey=yourapikey&action=showOb-
jectGroups&outputformat=json&useUTF8=true&useISO8601=true HTTP/1.1
Host: csv.webfleet.com
Authorization: Basic d2ZjdXNlcjp5b3VycHdk
```

The base URL

Every HTTP request to WEBFLEET.connect begins with constant elements for

- *host:*
  `csv.webfleet.com`
- *path:*
  `extern`

Therefore, the base URL with the https scheme used is:

https://csv.webfleet.com/extern

Handling the response

In case of an error, an error message is returned as plain text. The error message has the following layout:

```
id, description
```

`id` is a numeric value and `description` provides a reason text. The message is either in the language defined by the `lang` parameter or in English if no localised translation is available.

All methods that return data, provide the data as quoted character-separated values (CSV) with one record per line. Those methods' names typically contain verbs indicating data retrieval such as *show...* or *pop...* The Quoting character is ' **‘** ' - if this character is part of the data, it is quoted with **’**, appearing as **‘’**.The ordering of result columns might not always match that of the documentation and is subject to change without notice. It is therefore advisable to use the column names returned in the first response line to identify the data columns by their name. If there is no data to return, an error message is returned, for example:

```
63,document is empty
```

All methods that transmit data, e.g. all *send...* methods, return nothing on successful completion, that is the response is empty.

Error codes and descriptions are also returned in two HTTP response header fields:

- X-Webfleet-Errorcode: *<Error code>*
- X-Webfleet-Errormessage: *<Error message>*

If there is no error, the header fields are omitted.

Making HTTP POST requests

In addition to GET some actions allow to use HTTP POST to handle large payload. Examples are order related actions like

- sendDestinationOrderExtern
- insertDestinationOrderExtern
- updateDestinationOrderExtern

WEBFLEET.connect accepts POST requests with Content-Type '*application/x-www-form-urlencoded*'. The parameters and values are transferred in the body of the request. The parameter name is separated from the value by ' **=** ' and name/value pairs are separated from each other by ' **&** '. Special characters have to be encoded like query strings in URLs, see RFC 1738 - Uniform Resource Locators (URL).

The default character set is ISO-8859-1. To use UFT-8, you have to specify the character set in the HTTP header 'Content-Type'.

Example:

```
Content-Type: application/x-www-form-urlencoded;charset=utf-8
```

Code sample 3-1: Example of a complete POST request

```
POST/extern HTTP/1.1
Host:csv.webfleet.com
Connection:keep-alive
Content-Type:application/x-www-form-urlencoded;charset=UTF-8
Content-Length:177
Authorization: Basic d2ZjdXNlcjp5b3VycHdk

lang=en&account=wfcdevaccount&apikey=yourapikey&action=sendDestina-
tionOrder&objectno=0094&orderid=itn32&ordertext=Clean%20streets&longi-
tude=12399200&latitude=51364460&wp=51363230,12392520,Hamburger%20Str.
%2012,1,1
```

Using JSON

To use JSON, add the additional parameter *outputformat=json* to the request URL. This will return JSON instead of CSV. All other parameters and functionality stays the same.

The field names in the JSON output are identical to the column names in the CSV format. The returned JSON is an array with a flat representation of the data, which is not grouped nor structured. But there is one exception from the 'flat rule': The *surplus_data* member in the result of popQueueMessagesExtern is structured JSON.

Empty data is omitted in the result – no 'null' members. JSON data types, such as string, number and boolean are used where applicable. If there is no data to return, an empty JSON array is returned.

The HTTP Content-Type is '*application/json;charset=UTF-8*'.

Line breaks

To add line breaks in order text or message text you need to add the ASCII LF (line feed) character. For example: firstline%0Asecondline%0Athirdline

Getting started with HTTP requests

For trying out HTTP requests, you need a tool like cURL, wget, Postman, etc.

Preconditions

- Valid API key and credentials.

Making an HTTP request with cURL

Type in the following command on the command line. Remember to replace your specific credentials.

```
curl --user YOUR_USER:YOUR_PASSWORD "https://csv.webfleet.com/extern?lang=en&account=YOUR_ACCOUNT&apikey=YOUR_APIKEY&action=geocodeAddress&outputformat=json&freetext=Leipzig"
```

The response looks like:


```
[{'addrstreet':'','addrstreetnumber':'','addrzip':'04','addrcity':'Leipzig','addrcountry':'DE','postext':'04 Leipzig, Germany','formatted_longitude':'12°22'23.0\' E","formatted_latitude':'51°20'22.9\' N","longitude':12373074,'latitude':51339695,'additional_information':'accuracy=4'}]
```


HTTP request encoding

Every HTTP/HTTPS request must be a valid URL. That means that only ASCII characters are valid characters inside the URL and every other character, such as the German 'ß' character, or a special signs, such as the '@' sign, must be properly encoded inside the URL. There are two different types of encoding available depending on what kind of characters or special signs are needed.

Percent encoding

This is the basic version of character encoding. It contains only some special characters like for instance the 'ß' character (encoded as %DF) or the '@' sign (encoded as %40).

Find below an example that uses 'Sußanne@work' as WEBFLEET.connect account name and how the special characters and characters inside this user name are correctly percent encoded:

```
https://csv.webfleet.com/extern?lang=de&account=Su%DFanne
%40work&apikey=***&action=geocodeAddress&outputformat=json&freetext=Berlin
```

UTF-8 encoding

For more complex character encoding the UTF-8 encoding is needed as it contains nearly all characters and special signs.

Here is another example. For example, to geocode the Polish town 'Łódź' you have to enable the UTF-8 encoding in the request first by using the following extra parameter:

```
&useUTF8=true
```

All special characters and signs inside the URL must now be encoded using the UTF-8 format.

Here is how the URL should finally look like:

```
https://csv.webfleet.com/extern?lang=de&account=***&apikey=***&action=geocodeAddress&outputformat=json&useUTF8=true&freetext=%C5%81%C3%B3d
%C5%BA
```

Please note that the word Łódź has been converted to the following UTF-8 encoded string:

```
%C5%81%C3%B3d%C5%BA
```

Making SOAP requests

This section explains how to use SOAP to issue requests to WEBFLEET.connect. In order to ensure transmission security, it is required to use HTTPS to access the service via SSL.

WEBFLEET.connect supports the SOAP message protocol for issuing requests over an HTTPS connection. The easiest way to use the SOAP interface with your application is to use a SOAP toolkit appropriate for your programming platform. SOAP toolkits are available for most popular languages and platforms.

The files describing the operations and the data types are available at https://soap.webfleet.com/. Most SOAP toolkits support the automatic generation of routines and classes based on the description.

WEBFLEET.connect uses the MTOM extension to SOAP in order to provide an optimised transmission of data. Although most modern SOAP toolkits support this extension, your specific toolkit might need an additional support library to enable proper handling of MTOM.

Using .NET with the WEBFLEET.connect SOAP API

If you are using the .NET to integrate with WEBFLEET.connect we recommend to using C# as the main programming language. If however you are forced to use Visual Basic this requires to apply the additional steps described in Using WEBFLEET.connect SOAP with Visual Basic before importing the web service references into your development project.

Enabling the MTOM encoding support in .NET

.NET 3.5 (and higher) and the underlying Windows Communication Framework support SOAP with the MTOM extension. If you create a default web service project in Visual studio, the MTOM support is not automatically enabled for a new project. To enable it edit the `app.config` file in your code project.

Rename every occurrence of the `textMessageEncoding` element to `mtomMessageEncoding`. See 'before' and 'after' samples below.

Note: Only rename the element. The attributes and their values **must not be removed**. Do not forget to **close and re-open the project** and **possibly Visual Studio** after this change, otherwise Visual Studio will not recognise the changes.

The following code samples show when MTOM encoding is supported.

Code sample 3-2: Example: Before (MTOM encoding NOT supported):

```
<textMessageEncoding maxReadPoolSize="64" maxWritePoolSize="16" messageVer-
sion="Soap12" writeEncoding="utf-8">
<readerQuotas maxDepth="32" maxStringContentLength="8192" maxAr-
rayLength="16384" maxBytesPerRead="4096" maxNameTableCharCount="16384" />
</textMessageEncoding>
```

Code sample 3-3: Example - After (MTOM encoding supported):

```
<mtomMessageEncoding maxReadPoolSize="64" maxWritePoolSize="16" messageVer-
sion="Soap12" writeEncoding="utf-8">
 <readerQuotas maxDepth="32" maxStringContentLength="8192" maxAr-
rayLength="16384" maxBytesPerRead="4096" maxNameTableCharCount="16384" />
</mtomMessageEncoding>
```

Disabling MTOM

It's possible to enforce non-MTOM responses in the SOAP API. This allows to use the SOAP API also with frameworks or toolkits that don't support MTOM.

To ensure backward compatibility, the default response is still using MTOM. But you can disable MTOM per request by using one of these 2 methods:

- Append `/disable-mtom` to the URL of the SOAP service.
- Add a SOAP header with name `disable-mtom` and value `true` to the request.
  The namespace is `http://connect.webfleet.tomtomwork.com/services`

Using URL appendix

For example, instead of using `https://soap.webfleet.com/addressService` use `https://soap.webfleet.com/addressService/disable-mtom` when calling the API.

C# example code snippet

```
...
var endpointAddressBuilder = new EndpointAddressBuilder(client.Endpoint.Ad-
dress);
endpointAddressBuilder.Uri = new Uri(endpointAddressBuilder.Uri + "/dis-
able-mtom");
client.Endpoint.Address = endpointAddressBuilder.ToEndpointAddress();
...
```

Using SOAP header

Example SOAP message

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
 xmlns:ser="http://connect.webfleet.tomtomwork.com/services">
    <soap:Header>
        <ser:disable-mtom>true</ser:disable-mtom>
    </soap:Header>
    <soap:Body>
       ... details omitted ...
    </soap:Body>
</soap:Envelope>
```

C# example code snippet

```
...
var endpointAddressBuilder = new EndpointAddressBuilder(client.Endpoint.Ad-
dress);
endpointAddressBuilder.Headers.Add(AddressHeader.CreateAddressHeader("dis-
able-mtom", "http://connect.webfleet.tomtomwork.com/services", true));
client.Endpoint.Address = endpointAddressBuilder.ToEndpointAddress();
...
```

Using time zones with SOAP requests

With SOAP requests indicate a time zone known to the SOAP service in the general parameters (`gParm`) `timeZone` element. All date time values returned by the output of a SOAP function call will use the time zone thus indicated. Time zones known to the SOAP web service are enumerated in KnownTimeZones (see WSDL). The general parameters (`gParm`) `timeZone` element does not influence the interpretation of date time input data. This means, if a date time value is sent in a SOAP request, the time zone information specified in this value will be used.

Getting started with SOAP requests

For making SOAP requests, you need a special tool.

We recommend using SoapUI (https://www.soapui.org), which also offers a basic open source version.

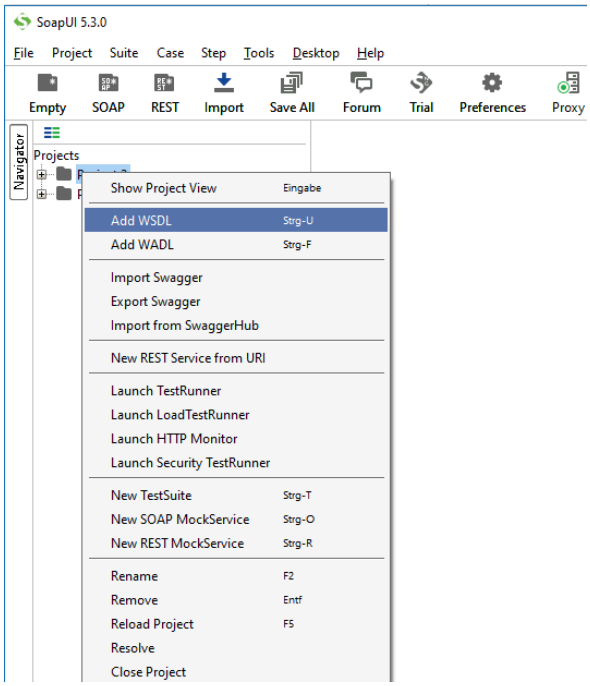
Preconditions

- SoapUI is installed.
- Valid WSDL URL or file.
- Valid API key and credentials.

Making a SOAP request with SoapUI

To make a SOAP request, do the following:

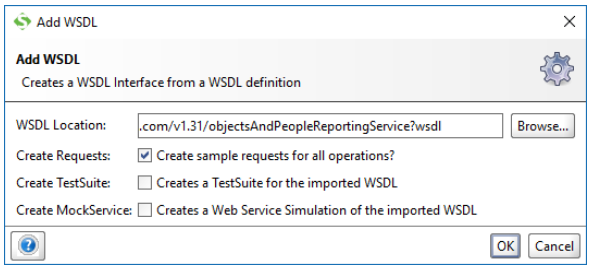
1. Create an empty project.
2. Right-click on the project
3. Click **Add WSDL**.

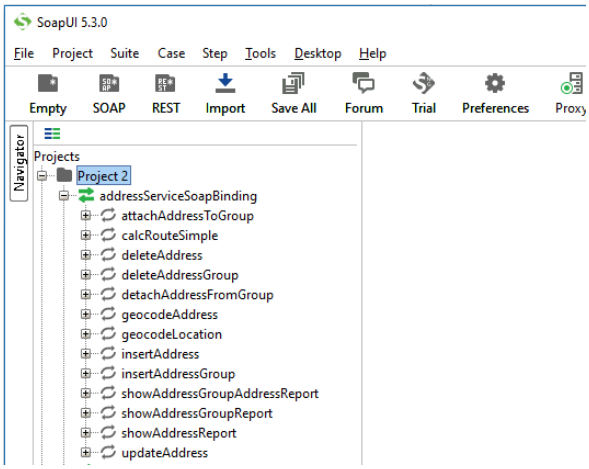


The **Add WSDL** dialogue opens.

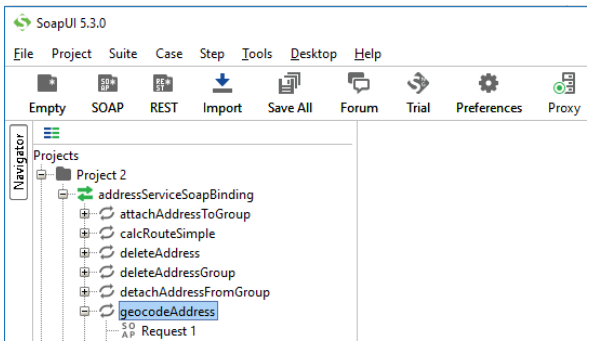
4. Enter a valid WSDL URL that points to a WEBFLEET.connect SOAP service.
   All available actions for this SOAP service are displayed.

   Example: `https://soap.webfleet.com/v1.33/objectsAndPeopleReportingSer-vice?wsdl`

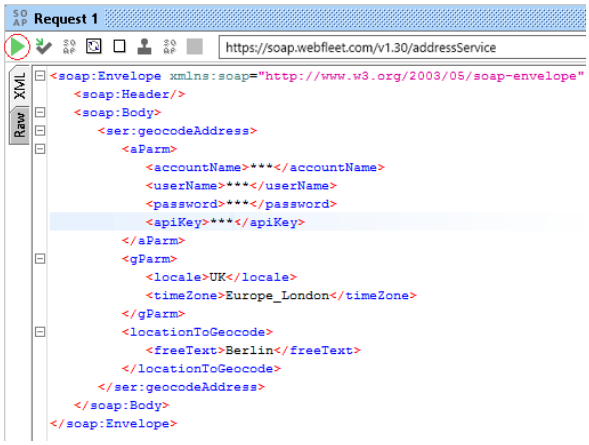




5. Click to select an action, for example **geocodeAddress**.
6. Create a new SOAP request.
   The request window opens.



7. Enter all required parameters.
8. Click the submit button.



Here is the result for the example SOAP request:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <ns3:geocodeAddressResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ns3=
      <return>
        <statusCode>0</statusCode>
        <statusMessage>The operation *geocodeAddress* processed succesfully.</statusMessage>
        <errors xsi:nil="true"/>
        <resultSize>1</resultSize>
        <results>
          <resultItem xsi:type="ns4:CompleteLocationWithAdditionalInformation" xmlns:ns4="htt
            <street xsi:nil="true"/>
            <postcode xsi:nil="true"/>
            <city xsi:nil="true"/>
            <country>Germany</country>
            <geoPosition>
              <latitude>52523404</latitude>
              <longitude>13411400</longitude>
              <mapcode xsi:nil="true"/>
              <latitudeHumanReadable>52°31'24.2" N</latitudeHumanReadable>
              <longitudeHumanReadable>13°24'41.0" E</longitudeHumanReadable>
            </geoPosition>
            <streetNumber xsi:nil="true"/>
            <positionText>Berlin, DE</positionText>
            <additionalInformation key="score" value="1.0"/>
            <additionalInformation key="type" value="state"/>
            <countryCode>DE</countryCode>
          </resultItem>
        </results>
      </return>
    </ns3:geocodeAddressResponse>
  </soap:Body>
</soap:Envelope>
```


SOAP parameters

The WEBFLEET.connect SOAP parameters and values slightly differ from the parameters and values that are used in the CSV/JSON interface. All valid SOAP parameters and values can always be found in the WSDL of the corresponding WEBFLEET.connect SOAP service.

The WSDLs for all our WEBFLEET.connect SOAP services can be found on https://soap.webfleet.com/

The following table shows some examples of what can differ between SOAP and JSON/CSV.

Table: Differences between SOAP and JSON/CSV

	SOAP example	CSV/JSON example
Spelling of parameters	apiKey	apikey
Parameter names	accountName	account
Parameter structure	<object objectNo='xxx'/>	objectno=xxx
Parameter values	<msgClass filter='ALL'/>	msgclass=0
Parameter structure and values	<gParm> <locale>UK</locale> <timeZone>Europe_London</timeZone> </gParm>	lang=en

Another good example of different parameter values between the SOAP and the CSV/JSON interface are the SOAP date range parameter values. Find below the parameter values that can be found in the WSDL of the WEBFLEET.connect SOAP service.

Example: SOAP date range parameters in the WSDL

```
<xs:simpleType name="DateRangePattern">
 <xs:restriction base="xs:string">
  <xs:enumeration value="UD"/>
  <xs:enumeration value="D0"/>
  <xs:enumeration value="Dm1"/>
  <xs:enumeration value="Dm2"/>
```

```
  <xs:enumeration value="Dm3"/>
  <xs:enumeration value="Dm4"/>
  <xs:enumeration value="Dm5"/>
  <xs:enumeration value="Dm6"/>
  <xs:enumeration value="W0"/>
  <xs:enumeration value="Wm1"/>
  <xs:enumeration value="Wm2"/>
  <xs:enumeration value="Wm3"/>
  <xs:enumeration value="WF0"/>
  <xs:enumeration value="WFm1"/>
  <xs:enumeration value="WFm2"/>
  <xs:enumeration value="WFm3"/>
  <xs:enumeration value="M0"/>
  <xs:enumeration value="Mm1"/>
  <xs:enumeration value="Mm2"/>
  <xs:enumeration value="Mm3"/>
  <xs:enumeration value="Mm4"/>
  <xs:enumeration value="Mm5"/>
  <xs:enumeration value="Mm6"/>
  <xs:enumeration value="Mm7"/>
  <xs:enumeration value="Mm8"/>
  <xs:enumeration value="Mm9"/>
  <xs:enumeration value="Y0"/>
  <xs:enumeration value="Ym1"/>
  <xs:enumeration value="Ym2"/>
  <xs:enumeration value="Ym3"/>
  <xs:enumeration value="Ym4"/>
  <xs:enumeration value="Ym5"/>
  <xs:enumeration value="Ym6"/>
  <xs:enumeration value="Ym7"/>
  <xs:enumeration value="Ym8"/>
  <xs:enumeration value="Ym9"/>
 </xs:restriction>
</xs:simpleType>
```

The following table provides a detailed description for every SOAP date range parameter value. They correspond to the CSV/JSON date range filter parameters too.

Table: SOAP date range parameter value descriptions

Parameter	Description
UD	User-defined range
D0	Today
Dm1	Yesterday
Dm2	Two days ago
Dm3	Three days ago
Dm4	Four days ago
Dm5	Five days ago
Dm6	Six days ago
W0	Current week

Parameter	Description
Wm1	Last week
Wm2	Two weeks ago
Wm3	Three weeks ago
WF0	Floating week, current day and previous seven days
WFm1	Floating week, the seven calendar days before WF0
WFm2	Floating week, the seven calendar days before WFm1
WFm3	Floating week, the seven calendar days before WFm2
M0	Current month
Mm1	Last month
Mm2	Two months ago
Mm3	Three months ago
Mm4	Four months ago
Mm5	Five months ago
Mm6	Six months ago
Mm7	Seven months ago
Mm8	Eight months ago
Mm9	Nine months ago
Y0	Current year
Ym1	Last year
Ym2	Two years ago
Ym3	Three years ago
Ym4	Four years ago
Ym5	Five years ago
Ym6	Six years ago
Ym7	Seven years ago
Ym8	Eight years ago
Ym9	Nine years ago

Request limits

The number of requests that can be issued is limited. If the number of requests executed exceeds this limit, WEBFLEET.connect will return an error message and not process requests again until there were no further requests within the limit monitoring interval. Limits are defined by a maximum number of requests allowed in a certain time period. For more information about request limits, read the chapter for the function you want to use.

Extensions of limits are available upon request on a case by case basis, but require a certain period of proper client operation within the existing limits set.

Should an application using WEBFLEET.connect cause too much load on the system, the limit may be reduced at any time without prior notice and eventually access to WEBFLEET.connect might be revoked completely if the problems are not fixed within a reasonable amount of time.

Common parameters

WEBFLEET.connect requests take a variety of parameters.

Find the parameters that are required on all requests below:

- Authentication parameters
- General parameters

Find the parameters that are not required but are shared between various requests below:

- Date range filter parameters

Authentication parameters

Parameter	Type	Description
account	string	A valid account name.
username	string	Username within the account that is allowed to access the service. Only for SOAP API.
password	string	Password for the user name. Only for SOAP API.
sessiontoken	string (32)	A session identifier, that has been fetched from the server using createSession. `sessiontoken` can be used as an alternative to `account`, `username` or `password` to authorise further requests for a limited time (see session lifetime).
apikey	string (36 characters)	This value is required.

The CSV/JSON API uses HTTP Basic Auth for user authentication. See Making HTTP requests

General parameters

Parameter	Type	Description
separator *(CSV only)*	positiveInteger	A delimiter character identifier that indicates the delimiter to be used for the output columns when using CSV format: • `1` - a *tab* character • `2` - a *space* character • `3` - a *comma* character If no value is specified, a semicolon used to as the delimiter for the output columns. This parameter is optional.
lang *(CSV/JSON only)*	string	Valid values are: • `de` - German • `en` - English Language to be used for output formatting. Expressed as a two-letter language code. The list of available languages is subject to change. Please ask your support contact if you want to use other values than those mentioned here.
locale *(SOAP only)*	enum	The value specified in the general parameters element (`genPam`) `locale` determines how the the result of a SOAP request will be localised. For locales known to the SOAP web service see WSDL.
action *(CSV/JSON only)*	string	Specifies the operation to be executed. See chapter Reference for a list of available operations and more information.
useISO8601 *(CSV/JSON only)*	boolean	If set to true, all date/time-relevant parameters are expected to be ISO8601-formatted and all date/time result fields will be ISO8601-formatted. See the ISO8601 specification or learn more here. The preferred ISO 8601 compliant notation for strings that represent dates which are to be passed to WEBFLEET.connect is the compact full notation with dashes and colons, optional milli seconds and time zone, for example: yyyy-MMdd'T'hh:mm:ss[(.|,)S](((+|-)hh:mm)|Z) *Example*: 2007-02-02T12:12:12.500+01:00 If set to `true`, the result will return date time values in UTC. If set to `false`, the result returns the time zone configured in the specific Webfleet account.

Parameter	Type	Description
columnfilter *(CSV/JSON only)*	string	This parameter reduces the result of a request to the specified columns. The possible values for this parameter are column names that can be returned by the function. If *columnfilter* is not specified, is empty or contains only invalid column names, then the full data set requested by the function is returned. The requested column names are case-sensitive and must be indicated correctly. Multiple column names must be separated with commas. Misspelled column names are disregarded in the resulting data set with no error notification.
timeZone *(SOAP only)*	enum	The value specified in the general parameters element (`genPam`) `timeZone` defines the time zone to which date and time related result data of the SOAP request will be matched. For valid values see WSDL.
useUTF8 *(CSV/JSON only)*	boolean	Controls how WF.connect interprets the character encoding of URL request parameters. If set to `true`, all parameters are expected to be UTF-8-encoded. If set to `false`, all parameters are interpreted as ISO-8859-1. Default is `false`.
outputformat *(CSV/JSON only)*	string	Defines the response format to be used by WEBFLEET.connect. Valid values: • `csv` (default) • `json`

Date range filter parameters

Parameter	Type	Description
range_pattern	string	Specifies a relative date range for the data to be returned. Valid values: • `d0` — Today • `d-1` — Yesterday • `d-2` — Two days ago • `d-3` — Three days ago • `d-4` — Four days ago • `d-5` — Five days ago • `d-6` — Six days ago • `w0` — Current week • `w-1` — Last week • `w-2` — Two weeks ago • `w-3` — Three weeks ago • `wf0` — Floating week, current day and previous seven days • `wf-1` — Floating week, the seven calendar days before `wf0` • `wf-2` — Floating week, the seven calendar days before `wf-1` • `wf-3` — Floating week, the seven calendar days before `wf-2` • `m0` — Current month • `m-1` — Last month • `m-2` — Two months ago • `m-3` — Three months ago • `ud` — User-defined range **Note**: Use of this parameter rules out the usage of `rangefrom_string` and `rangeto_string` and vice versa except for `range_pattern=ud`. Requires `rangefrom_string` and `rangeto_string` to be present.

Parameter	Type	Description
rangefrom_string	string	Lower bound for a date range (start date). The required format depends on the value for the lang parameter: • lang = de *Format*: dd.MM.yyyy HH:mm:ss *Example*: 13.07.2007 23:59:59 • lang = en *Format*: dd/MM/yyyy HH:mm:ss *Example*: 13/07/2007 23:59:59 • lang = nl *Format*: d-M-yyyy H:mm:ss *Example*: 13-07-2007 23:59:59
rangeto_string	string	Upper bound for a date range (end date).
year	nonNegativeInteger(4)	A range that covers the entire year, from 1st of January 00:00:00 o'clock to 31st of December 23:59:59 o'clock. Can be used as an alternative to `range_pattern`, `rangefrom_string`, `rangeto_string`.
month	nonNegativeInteger(2)	A range that covers the entire month that is indicated from first day 00:00:00 o'clock to last day 23:59:59 o'clock of the same month. **Note**: This parameter has to be used together with with `year`. Can be used as an alternative to `range_pattern`, `rangefrom_string`, `rangeto_string`.
day	nonNegativeInteger(2)	A range that covers the entire day that is indicated from 00:00:00 o'clock to 23:59:59 o'clock. **Note**: This parameter has to be used together with `year` and `month`.

Reference

Message queues

Overview of queues

An alternative way to collect data is offered via the message queues. All messages received by Webfleet are made available via message queues that can be accessed using a message queue. A message queue is a temporary repository for all messages associated with that subscription's filter condition (message class). Once a message has been retrieved from a queue by using popQueueMessagesExtern and subsequently acknowledged by using ackQueueMessagesExtern, the message is deleted from that queue and no longer available.

In order to prevent the message queues from overflowing, messages are stored a limited amount of time (time to live, TTL) and are automatically deleted once the TTL has expired, even if they have not been retrieved. The default TTL is set to 14 days.

The following points are a violation of the intended use of WEBFLEET.connect queues:

- Repeatedly creating queues and then leaving them inactive.
- Repeatedly calling popQueueMessagesExtern without subsequently calling ackQueueMessagesExtern.

Note: Queues that have not been used for 30 consecutive days may be removed without prior notice.

Outbound messages

Outbound messages in the text and order messages classes are available in the message queue system. Consider that outbound messages in the message queue provide a reduced data set, e.g. no position or odometer information. Outbound messages that are in the queue already have not necessarily been delivered to the device.

createQueueExtern

Description

Creates a new queue.

A queue is always bound to the user creating it and collects messages matching the message class provided.

Technical details

SOAP endpoint address / function name	messagesService#createQueueExtern
Request limits	10 requests / 24 hours

Parameters

createQueueExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to createQueueExtern

Parameter	Type	Description
msgclass	nonNegativeInteger	Valid values are: • `0` - All messages Includes tracking-only messages and 4,5,7,8 • `2` - All except position messages Includes 4,5,7,8, does not include tracking-only messages. • `4` - Order related messages • `5` - Driver related messages • `7` - Status messages • `8` - Text messages • `15` - Third party messages (LINK.connect, ...)

Result

Result for createQueueExtern:

Parameter	Type	Description
action	string	The name of the operation that has been executed.
result	boolean	Always `true`, errors will be reported using the standard error reporting mechanism.

deleteQueueExtern

Description

Deletes an existing queue.

Technical details

SOAP endpoint address / function name	messagesService#deleteQueueExtern
Request limits	10 requests / 24 hours

Parameters

deleteQueueExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to deleteQueueExtern:

Parameter	Type	Description
msgclass	nonNegativeInteger	Valid values are: • `0` - All messages Includes tracking-only messages and 4,5,7,8. • `2` - All except position messages Includes 4,5,7,8, does not include tracking-only messages. • `4` - Order related messages • `5` - Driver related messages • `7` - Status messages • `8` - Text messages • `15` - Third party messages (LINK.connect, ...)

Result

Result for createQueueExtern:

Parameter	Type	Description
action	string	The name of the operation that has been executed.
result	boolean	Always `true`. Errors will be reported using the standard error reporting mechanism.

popQueueMessagesExtern

Description

Retrieves outstanding messages for a given subscription. This includes most important and most often occurring inbound and outbound messages.

Before using `popQueueMessagesExtern` to retrieve outstanding messages, you need to create a queue using createQueueExtern using the same `message class` parameter that you are going to provide with calls to `popQueueMessagesExtern`.

Once you have successfully processed (and stored) all of the retrieved messages, you need to use ackQueueMessagesExtern to acknowledge completion of the message transfer to your application. Otherwise, the messages will be kept and returned again during the next call to `popQueueMessagesExtern`. Calls to `popQueueMessagesExtern` and `ackQueueMessagesExtern` must be serialised.

In order to prevent your system from being flooded with oversized responses, the number of messages that will be returned on a single response is limited to 500. This limit can be adjusted per account on request.

The resulting data set is delivered in the language you have chosen in the Webfleet account and not on the language you have indicated in the `lang` parameter.

Technical details

SOAP endpoint address / function name	messagesService#popQueueMessagesExtern
Request limits	10 requests / minute

Parameters

popQueueMessagesExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to popQueueMessagesExtern

Parameter	Type	Description
msgclass	nonNegativeInteger	Valid values: • `0` - All messages Includes tracking-only messages and 4,5,7,8. • `2` - All except position messages Includes 4,5,7,8, does not include tracking-only messages. • `4` - Order related messages • `5` - Driver related messages • `7` - Status messages • `8` - Text messages • `15` - Third party messages (LINK.connect, ...)

Result

Result for popQueueMessagesExtern

Parameter	Type	Description
msgid	string	`mgsid` identifies a message and is system wide unique. Valid value range: 0 # msgid # 2^{64}-1.
msg_time	dateTime	ISO 8601-formatted date and time in the UTC time-zone, combined representation in the extended format. *Example: 2007-12-24T16:00:00Z*

Parameter	Type	Description
msg_class	nonNegativeInteger	Valid values: • `0` - unknown • `1` - system • `2` - text • `3` - polling • `4` - timer event • `5` - GPS event • `6` - input event • `7` - set output • `8` - data • `9` - configuration • `10` - trip • `11` - order
msg_type	nonNegativeInteger	A 8 or 9 digit number representing class and type of the message. Counted from right to left, digits 0-3 indicate the message type, digits 4-6 are reserved and digits 7 and 8 indicate the message class. The message class is equal to the field `msg_class`. For a comprehensive list of possible message types see Messages types.
msg_text	string	The actual text of the message.
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
driverno	string	Identifying number of a driver. Unique within an account. See Drivers to learn more about drivers.
ign	int	
inputs	string	This property is deprecated. Please use `ioname` in `surplus_data` instead of `inputs`.
odometer	int	The current odometer value **in 100 meter**.
outputs	string	
pndconn	boolean	Indicates the connection state of the Driver Terminal.
trip_mode	nonNegativeInteger	Valid values: • `0` - Unknown trip type • `1` - Private trip • `2` - Business trip • `3` - Commute trip • `4` - Correction trip (user changed the odometer value manually)

Parameter	Type	Description
pos_latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd).
pos_longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd).
pos_text	string	
pos_country	string (2)	ISO 3166-1 alpha-2 code for the country (capital letters).
pos_postcode	string	
pos_params	string	**Note**: This parameter is not supported anymore. Use `location_params` instead.
pos_time	dateTime	ISO 8601-formatted date and time in the UTC timezone, combined representation in the extended format. *Example: 2007-12-24T16:00:0*
pos_addrno	string (10)	Refers to the nearest address if the position is within the address radius. See Addresses to learn more.
speed	int	in km/h
course	nonNegativeInteger	Compass direction in degrees (0° ... 360°):
status	string	GPS status indicator. Valid values: • `L` - No fix available. Position data refer to the last known position. • `V` - No fix data available. • `A` - Sufficient quality for a valid fix.

Parameter	Type	Description
direction	nonNegativeInteger	Cardinal and intercardinal compass directions, derived from `course`. Valid values: • `1` - North • `2` - Northeast • `3` - East • `4` - Southeast • `5` - South • `6` - Southwest • `7` - West • `8` - Northwest
orderno	string (20)	Account-unique order id, case-sensitive. Maximum capacity is 20 bytes, actual number of characters depends on the byte count consumed by the UTF-8 encoding of those characters. See Orders to learn more about orders.
order_state	nonNegativeInteger	Valid values: • `0` - Not yet sent • `100` - Sent • `101` - Received • `102` - Read • `103` - Accepted • `201` - Service order started • `202` - Arrived at destination • `203` - Work started • `204` - Work finished • `205` - Departed from destination • `221` - Pickup order started • `222` - Arrived at pick up location • `223` - Pick up started • `224` - Pick up finished • `225` - Departed from pick up location • `241` - Delivery order started • `242` - Arrived at delivery location • `243` - Delivery started • `244` - Delivery finished • `245` - Departed from delivery location • `298` - Resumed • `299` - Suspended • `301` - Cancelled • `302` - Rejected • `401` - Finished

Parameter	Type	Description
order_type	nonNegativeInteger	Type of order. Valid values: • `1` - service order • `2` - pickup order • `3` - delivery order
dest_latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd).
dest_longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd).
dest_text	string	
eta	dateTime	ISO 8601-formatted date and time in the UTC timezone, combined representation in the extended format. *Example: 2007-12-24T16:00:00Z*
distance	nonNegativeInteger	Distance to destination (in meters)
user_status	int	The number of a predefined status message or a predefined order message.
user_text	String (60)	An additional text that refers to a predefined status message or a predefined order message.
dt_cardid	string (18)	A driver card id as used with the Digital Tachograph, prefixed with the ISO 3166-1 alpha-2 code for the country (capital letters). Contains the card id for driver cards that are not assigned to a driver or for any other card type. Only available on card insertion and removal messages.
dt_cardaction	int	Valid values: • `0` - inserted • `1` - removed
dt_cardtype	nonNegativeInteger	Valid values: • `0` - Unknown • `1` - Driver card • `2` - Workshop card • `3` - Control card • `4` - Company card

Parameter	Type	Description
dt_slot	nonNegativeInteger	Valid values: • `1` - Driver • `2` - Co-Driver
dt_state	int	Valid values: • 0 - connected • 1 - disconnected
workstate	nonNegativeInteger	Valid values: • `0` - Unknown • `1` - Free time (PND only) • `2` - Pause (PND and digital tachograph) • `3` - Standby (digital tachograph only) • `4` - Working (PND only) • `5` - Driving (digital tachograph only) • `6` - Other work (digital tachograph only)
source_device	string	The external equipment this message originated from: • `0` - Driver Terminal • `1` - Digital tachograph • `2` - Remote LINK Working Time • `3` - Remote LINK Logbook • 4 - ecoPLUS • 6 - LINK 105 • 11 - PRO 2020
rll_btaddress	string (17)	Remote LINK Bluetooth address. The IEEE 802 MAC-48/EUI-48 address of a Bluetooth device, formatted as six groups of two hexadecimal digits, separated by colons (:). *Example: 00:13:6C:88:26:0B*
rll_buttonid	string (16)	Remote LINK/ID key identifier
surplus_data	string	A JSON-encoded string that carries additional data items that augment certain messages. • `dt_timeoffset`: time difference between `event_time` and the Digital Tachograph's internal clock, in milliseconds. • `ep_paireddata`: ecoPLUS initiating message, containing parameters specific to ep_paireddata. • `ep_tripdata`: Additional trip information reported by ecoPLUS, containing parameters specific to ep_tripdata.

Parameter	Type	Description
		• `ep_snapshotdata`: ecoPLUS update recording, containing parameters specific to ep_snapshotdata. • `invalid_waypoints`: An array of the numbers of the waypoints that are not valid for a route calculation on a Driver Terminal. • `fms_snapshotdata`: FMS live data, containing parameters specific to fms_snapshotdata.
		• `fms_fuel_difference`: Additional data for fuel level increase/decrease messages, containing parameters specific to fms_fuel_difference. • `fms_tripdata`: Additional trip information reported by FMS, containing parameters specific to fms_tripdata. • `acc_events`: additional details on acceleration events, containing parameters specific to acc_events. **Note**: This parameter is not available by default. Please contact Customer Support to activate it. Additional cost will apply.
		• `aux_device_data_in`: information about third party Bluetooth device (LINK.connect), containing parameters specific to aux_device_data_in. • `aux_device_data_quota`: information about allowed and used quota (LINK.connect), containing parameters specific to aux_device_data_quota. • `aux_device_data_out_status`: status of sent messages (LINK.connect), containing parameters specific to aux_device_data_out_status.
		• `aux_device_data_reference`: indentifiers for outgoing and incoming messages (LINK.connect), containing parameters specific to aux_device_data_reference. • `troublecodes`: Information about diagnostic trouble codes. See parameters specific to troublecodes.

Parameter	Type	Description
		• `app_data_in`: information about application (small binary message) data, containing parameters specific to app_data_in. • `app_data_quota`: information about allowed and used quota, containing parameters specific to app_data_quota. • `app_data_out_status`: status of sent small binary messages, containing parameters specific to app_data_out_status. • `app_data_reference`: indentifiers for outgoing and incoming small binary messages, containing parameters specific to app_data_reference. • `er_tripdata`: Additional trip information reported by eco reporting, containing parameters specific to er_tripdata. • `ioname`: input/output name, for example the name of a digital input. This element is appended if an input/output was switched to ON or OFF. `msg_type` gives information about switch type.
event_time	dateTime	ISO 8601-formatted date and time in the UTC timezone, combined representation in the extended format. *Example: 2007-12-24T16:00:00Z* Contains local time of the device on which the event occurred and/or the time the message was generated.
start_time	string	The time recorded when this event started.
start_latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
start_longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
start_odometer	int	The odometer reading when this event started.
end_time	string	The time recorded when this event ended.

Parameter	Type	Description
end_latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
end_longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
end_odometer	int	The odometer reading when this event ended.
trip_type	string	Valid values: • `G` - trip (vehicle changed GPS position) • `S` - standstill
order_addrno	string (10)	The address number of the order destination, if available.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.
driverkey_deviceaddress	string	Address of the driver key reading device which is associated with the vehicle.
driverkey	string	Composed by driver key type and driver key value.

Parameter	Type	Description
location_params	string	A list of name-value pairs that describe the position. Name and value are separated by '**=**'. Pairs are separated by '**;**'. Possible names are: • countryCode • countryName • adminLevel1Code • adminLevel1Name • adminLevel2Name • adminLevel3Name • adminLevel4Name • adminLevel5Name • locality • subLocality1 • subLocality2 • cityBlock • namedArea • neighbourHood • street • route • postalCode • extendedPostalCode • houseNumber • roadNumbers • speedLimit • roadUse Please note that the result does not always contain all position parameters. The parameter names are subject to modification. Example: `countryCode=DE;countryName=Deutschland;adminLevel1Code=SN; adminLevel1Name=Sachsen;locality=Leipzig;subLocality1=Zentrum-Ost; street=Inselstraße;postalCode=04103;houseNumber=22;speedLimit=30; roadUse=LocalStreet;`
odometer_long	int	The current odometer value **in meter**.
start_battery_level	int	Electric Vehicle's battery level in percent at the start of the trip. Values are within [0, 100]
end_battery_level	int	Electric Vehicle's battery level in percent at the end of the trip. Values are within [0, 100]

ackQueueMessagesExtern

Description

Acknowledges outstanding messages retrieved with a previous call to popQueueMessagesExtern.

Technical details

SOAP endpoint address / function name	messagesService#ackQueueMessagesExtern
Request limits	10 requests / minute

Parameters

ackQueueMessagesExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to ackQueueMessagesExtern:

Parameter	Type	Description
msgclass	nonNegativeInteger	Valid values: • `0` - All messages Includes tracking-only messages and 4,5,7,8. • `2` - All except position messages Includes 4,5,7,8, does not include tracking-only messages. • `4` - Order related messages • `5` - Driver related messages • `7` - Status messages • `8` - Text messages • `15` - Third party messages (LINK.connect, ...)

Result

Result for ackQueueMessagesExtern:

Parameter	Type	Description
action	string	The name of the operation that has been executed.
result	boolean	Always `true`. Errors will be reported using the standard error reporting mechanism.

Parameter	Type	Description
acknowledgedMessages	nonNegativeInteger	This value is a pointer value that does not always represent the actually acknowledged messages. **Note**: This value is used for internal purposes only.
outstandingMessages	nonNegativeInteger	An estimate for the number of messages that are still waiting in the queue.

Message types

msg_type	Available in message queue	Description
10000110	0, 2	Async command completion successful (outbound)
10000111	0, 2	Async command completion failed (outbound)
20000100	0, 2, 8	Sent text message (outbound)
20000101	0, 2, 8	Incoming text message
20000103	0, 2, 8	All messages deleted (outbound)
21100100	0, 2, 8	Sent text message (outbound)
21100101	0, 2, 8	Incoming text message
30000200	0, 2	Position query
31100200	0, 2	Position query
31100202	0, 2	Position reply - error
40000220	0, 2	Position message
40000230	0, 2	Distance message
41100300	0, 2	Position message
41100310	0, 2	Position/total position number
41100320	0, 2	Position message
41100330	0, 2	Distance message
41100340	0, 2	Position message
41100350	0, 2	Position/total position number
41100351	0, 2	Position message - empty

msg_type	Available in message queue	Description
41100353	0, 2	Tracking data indicating the number of positions delivered
50000420	0, 2	Standstill begin
50000421	0, 2	Standstill end
50000424	0, 2	Standstill end - indicating the duration of the standstill event
51100420	0, 2	Standstill begin
51100421	0, 2	Standstill end
51100426	0, 2	Movement violation
51100427	0, 2	Position lock alarm
60000510	0, 2	Ignition on
60000511	0, 2	Ignition off
60000520	0, 2, 7	Unit switched on
60000521	0, 2, 7	Unit switched off
60000545	0, 2, 7	Input switched on
60000546	0, 2, 7	Input switched off
60000547	0, 2, 7	GPS antenna connected (specific to LINK 300/310)
60000548	0, 2, 7	GPS antenna disconnected (specific to LINK 300/310)
60000551	0, 2, 7	External GSM antenna connected (specific to LINK 510)
60000552	0, 2, 7	External GSM antenna disconnected/not connected (specific to LINK 510)
60000553	0, 2, 7	External GSM antenna error/defective/does not comply with technical requirements from Webfleet
60000831	0, 2, 7	PTO (input) switched on
60000832	0, 2, 7	PTO (input) switched off
61100510	0, 2	Ignition on
61100511	0, 2	Ignition off
61100512	0, 2, 7	Trip data

msg_type	Available in message queue	Description
61100530	0, 2, 7	Power connected
61100531	0, 2, 7	GPS antenna failure
61100532	0, 2, 7	Power disconnected
61100540	0, 2, 7	Input signal changed
61100542	0, 2, 7	Operating data
61100543	0, 2, 7	Operating data not available
61100544	0, 2, 7	Operating data: input name, number of switch events, duration, distance
61100545	0, 2, 7	Input switched on
61100546	0, 2, 7	Input switched off
61100547	0, 2, 7	Request operating data - error (specific to LINK classic)
66000510	0, 2, 7	Device on (specific to TomTom GO 715) **Note**: This device is no longer in use.
66000511	0, 2, 7	Device off (specific to TomTom GO 715) **Note**: This device is no longer in use.
70000600	0, 2	Switch indicated output on
70000601	0, 2	Switch indicated output off
70000602	0, 2	Acknowledgment for message 600
70000603	0, 2	Acknowledgment for message 601
71100600	0, 2	Switch indicated output on
71100601	0, 2	Switch indicated output off
71100602	0, 2	Acknowledgment for message 600
71100603	0, 2	Acknowledgment for message 601
80000705	0, 2, 5, 7	Status message
80000706	0, 2, 5, 7	Text message
80000710	0, 2, 5, 7	Logon driver (including tachograph driver card inserted)

Message types

msg_type	Available in message queue	Description
80000711	0, 2, 5, 7	Logoff driver (including tachograph driver card removed)
80000712	0, 2, 5, 7	Begin of work, driver indicated
80000713	0, 2, 5, 7	End of work, driver indicated
80000714	0, 2, 5, 7	Begin of break, driver indicated
80000715	0, 2, 5, 7	End of break, driver indicated
80000716	0, 2, 5, 7	Work status change, indicating driver role, old and new work status (applies to all PRO 71xx, 91xx devices when used with identified driver and Remote LINK Working Time)
80000717	0, 2, 5, 7	Work started (driver role and driver name indicated)
80000718	0, 2, 5, 7	Work ended (driver role and driver name indicated)
80000719	0, 2, 5, 7	Work time event, indicating driver
80000750	0, 2, 5, 7	Message read
80000751	0, 2, 5, 7	Destination reached
80000752	0, 2, 5, 7	Navigation started (destination indicated)
80000753	0, 2, 5, 7	Navigation cancelled
80000754	0, 2, 5, 7	Message received
80000755	0, 2, 5, 7	Message discarded
80000756	0, 2, 5, 7	Connection established to device
80000770	0, 2, 5, 7	Digital tachograph connected
80000771	0, 2, 5, 7	Digital tachograph disconnected
80000772	0, 2, 5, 7	Device signals disconnection of unknown digital tachograph
80000773	0, 2, 5, 7	Tachograph driver card not assigned to driver
80000774	0, 2, 5, 7	Digital tachograph card (company card, workshop card, control card) inserted into digital tachograph
80000775	0, 2, 5, 7	Digital tachograph card (company card, workshop card, control card) removed from digital tachograph

msg_type	Available in message queue	Description
80000776	0, 2, 5, 7	Time difference between digital tachograph and LINK device
80000777	0, 2, 5, 7	Working state changed for digital tachograph card that is not assigned to a driver in Webfleet
80000778	0, 2, 5, 7	Driving without appropriate card started
80000779	0, 2, 5, 7	Driving without appropriate card ended
80000780	0, 2, 5, 7	Remote LINK is paired
80000781	0, 2, 5, 7	Remote LINK low battery (specific to LINK 5xx/4xx/3xx)
80000782	0, 2, 5, 7	Driver logon
80000783	0, 2, 5, 7	Unknown driver logon
80000784	0, 2, 5, 7	Driver logoff
80000785	0, 2, 5, 7	Unknown driver logoff
80000786	0, 2, 5, 7	Working time event of driver
80000787	0, 2, 5, 7	Working time event of unknown employee
80000788	0, 2, 5, 7	Remote LINK pairing rejected as not supported by subscription
80000789	0, 2, 5, 7	Remote LINK battery replaced
80000790	0, 2, 5, 7	Remote LINK setup request per installation tool
80000791	0, 2, 5, 7	Remote LINK setup response for installation tool
80000792	0, 2, 5, 7	Remote LINK pairing rejected as there is already another device paired
80000800	0, 2, 5, 7	ecoPLUS paired (specific to LINK 5xx/4xx/3xx)
80000801	0, 2, 5, 7	ecoPLUS pairing rejected as not supported by subscription
80000802	0, 2, 5, 7	ecoPLUS pairing rejected as there is already another device assigned (specific to LINK 5xx/4xx/3xx)
80000805	0, 2, 5, 7	ecoPLUS power status (specific to LINK 5xx/4xx/3xx)

Message types

msg_type	Available in message queue	Description
80000806	0, 2, 5, 7	ecoPLUS fuel type changed (specific to LINK 5xx/4xx/3xx)
80000807	0, 2, 5, 7	ecoPLUS connection lost (specific to LINK 5xx/4xx/3xx)
80000808	0, 2, 5, 7	ecoPLUS plugged into vehicle (specific to LINK 5xx/4xx/3xx)
80000809	0, 2, 5, 7	ecoPLUS power reconnected
80000810	0, 2, 5, 7	Crash sensor triggered
80000811	0, 2, 5, 7	Acceleration event
80000820	0, 2, 5, 7	Fuel level in the fuel tank decreased during driving
80000821	0, 2, 5, 7	Unexpected fuel loss - Significant drop of fuel level in the fuel tank detected
80000822	0, 2, 5, 7	Refuelled vehicle - Significant increase of fuel level in the fuel tank detected
80000830	0, 2, 5, 7	Fuel level in fuel tank is low
80000840	0, 2, 5, 7	FMS connected
80000841	0, 2, 5, 7	FMS disconnected (specific to LINK 510/710)
80000860	0, 2, 5, 7, 15	Mobile-terminated - LINK.connect/Binary message was sent to LINK/device. This message contains an outgoing actual opaque payload message. The payload is not repeated as the integration server back-end has previously sent it. Integrations have to store this message when following up on the delivery status of outbound opaque payload to a device.

Message types

msg_type	Available in message queue	Description
80000861	0, 2, 5, 7, 15	Mobile-originated - Status response for aux_device_data_out/app_device_data_out This message acknowledges device data sent from the integration server backend to an device. After using sendAuxDeviceData/sendBinaryMessage the integrator should wait for this acknowledge message to come through on the Queue Service and check the status of the opaque payload data sent out. Before this acknowledge message is received back from the device by Webfleet/integration server backend, all further calls to sendAuxDeviceData/sendBinaryMessage will be rejected by Webfleet with a specific error message, as it is assumed that the previous opaque payload transmission is still in progress. JSON structures in `surplus_data`: • `aux_device_data_out_status/app_data_out_status` • `aux_device_data_quota/app_data_quota` • `aux_device_data_reference/app_data_reference`
80000862	0, 2, 5, 7, 15	Mobile-originated - Contains the payload that was sent by the device. This message contains incoming actual opaque payload data from a device. Integrations have to store this data as there is no other way for the integration server backend to retrieve the opaque payload data. JSON structures in `surplus_data`: • `aux_device_data_in/app_data_in` • `aux_device_data_quota/app_data_quota` • `aux_device_data_reference/app_data_reference`
80000863	0, 2, 5, 7, 15	Mobile-terminated - Response sent to the device after receiving third party data. This message acknowledges data received from the device.Integrations can use this message for tracking the message flow. JSON structures in `surplus_data`: • `aux_device_data_status/app_data_status` • `aux_device_data_reference/app_data_reference`

msg_type	Available in message queue	Description
80000864	0, 2, 5, 7, 15	Mobile-originated/mobile-terminated - Control message. This is a multi-purpose control message sent to the device or received from the device.
80000880	0, 2, 5, 7	List of current vehicle Diagnostic Trouble Codes has changed. JSON structure in `surplus_data`: • `troublecodes`
80000913	0, 2, 5, 7	LINK 105 pairing rejected (there is already another device assigned)
80000914	0, 2, 5, 7	LINK 105 paired
80000915	0, 2, 5, 7	LINK 105 pairing rejected (not supported by subscription)
80000916	0, 2, 5, 7	LINK 105 setup response for installation tool
80000917	0, 2, 5, 7	LINK 105 setup request via installation tool
80000918	0, 2, 5, 7	LINK 105 connection lost
80000919	0, 2, 5, 7	LINK 105 connection re-established
80000927	0, 2, 5, 7	Service token request/response message
80000930	0, 2	generic CAN data
80000931	0, 2, 5, 7	Periodic alive message
80000934	0, 2	Vehicle battery voltage measurement
80000945	0, 2, 5, 7	Driver/worker worktime event (specific to PRO 2020)
80000946	0, 2, 5, 7	Driver key signals driver logon but driver is undefined because key value is unknown or not assigned
80000947	0, 2, 5, 7	Driver key signals driver logoff but driver is undefined because key value is unknown or not assigned
80000948	0, 2, 5, 7	Driver key signals driver logoff
80000949	0, 2, 5, 7	Driver key signals driver logon
80000950	0, 2, 5, 7	Driver key signals driver logoff but driver is not assigned to this unit

Message types

msg_type	Available in message queue	Description
80000971	0, 2	Signals a door's state has changed
80000981	0, 2, 5, 7	Number of days driven in roaming area
80000982	0, 2, 5, 7	Exceeded maximum days driven in roaming area
80001000	0, 2	Device Update: Network check request
80001001	0, 2	Device Update: Network check response
80001002	0, 2	Device Update: Key request
80001003	0, 2	Device Update: Key response
80001004	0, 2	Device Update: Start
80001005	0, 2	Device Update: Result
80001006	0, 2	Device Update: Progress
80001040	0, 2, 5, 7	Driver key signals driver logon but driver is undefined because key value is unknown or not assigned
80001041	0, 2, 5, 7	Driver key signals driver logoff but driver is undefined because key value is unknown or not assigned
80001042	0, 2, 5, 7	Driver key signals driver logoff
80001043	0, 2, 5, 7	Driver key signals driver logon
80001044	0, 2, 5, 7	Driver key signals driver logoff but driver is not assigned to this unit
80001045	0, 2	Driver Key Identification 2: Driver Key Identification Result
80001046	0, 2	Driver Key Identification 2: Driver logon rejected
80001050	0, 2	Device reports periodic eco data without trips
80001051	0, 2	Device reports periodic eco data without trips, fuel information missing or invalid
80001060	0, 2	Vehicle Data Reporting: Announcement of available data items
80001061	0, 2, 5, 7	Vehicle Data Reporting: Cruise control switched on

Message types

msg_type	Available in message queue	Description
80001062	0, 2, 5, 7	Vehicle Data Reporting: Cruise control switched off
80001064	0, 2, 5, 7	Vehicle Data Reporting: Engine switched on
80001065	0, 2, 5, 7	Vehicle Data Reporting: Engine switched off
80001066	0, 2	Contains the length of the engine on interval
80001067	0, 2	Contains the length of the engine on interval, but no total engine hours data
80001068	0, 2	Contains the length of the engine on interval, but no fuel related data
80001069	0, 2	Contains the length of the engine on interval, but no fuel and no total engine hour related data
80001080	0, 2	Driver key list request
80001081	0, 2	Driver key list
80001091	0, 2	Tire status changed message
80001092	0, 2	Tire status request
80001093	0, 2	External tire issue
80001100	0, 2	Exhaust fluid warning
80001200	0, 2, 5, 7	Engine oil temperature too high
80001201	0, 2, 5, 7	Engine coolant temperature too high
81100700	0, 2, 5, 7	Event message, status
81100703	0, 2, 5, 7	Status log - no data available
81100704	0, 2, 5, 7	Request status log - error
81100751	0, 2, 5, 7	Odometer manual input
81100760	0, 2, 5, 7	Arriving at customer (specific to LINK classic)
81100761	0, 2, 5, 7	Departing from customer (specific to LINK classic)
81100763	0, 2, 5, 7	Trip begin (specific to LINK classic)
81100764	0, 2, 5, 7	Trip end (specific to LINK classic)
81100765	0, 2, 5, 7	Begin waiting

msg_type	Available in message queue	Description
81100766	0, 2, 5, 7	End waiting
90000550	0, 2	Query for status of the external GPS antenna (outbound) (specific to LINK 510/710)
90000801	0, 2	Configuration acknowledge (specific to LINK classic)
90000808	0, 2	Firmware release information
90000811	0, 2	Reset unit - ok
90000814	0, 2	CAN configuration request
90000815	0, 2	CAN config
90000816	0, 2	CAN config acknowledge
90000826	0, 2	Device successfully activated
90000827	0, 2	Device has been reset
90000843	0, 2	PRO 5150 activation request. This message is created when a TomTom PRO 5150 that is not yet connected to Webfleet pairs with a LINK device for the first time and requests for the additional service PRO 5150 Webfleet compatibility upgrade.
90000847	0, 2	Set initial engine hours
90000848	0, 2	Setting of engine hours is rejected, the LINK has another source for the total engine hours
90000849	0, 2	Acknowledgment to set initial hours
90000912	0, 2	Reset LINK 105
90000990	0, 2	CAN3 configuration request
90000991	0, 2	CAN3 config announcement
90000992	0, 2	CAN3 signals config
90000993	0, 2	CAN3 trigger config
90000994	0, 2	CAN3 signals config acknowledge
90000995	0, 2	CAN3 trigger config acknowledge
90001010	0, 2	CAN bus installation: Device request to store its configuration

Message types

msg_type	Available in message queue	Description
90001011	0, 2	CAN bus installation: Set response
91100802	0, 2	For LINK classic: Configuration acknowledge - error (specific to LINK classic)
91100803	0, 2	Version information (specific to LINK classic)
91100806	0, 2	Version information - error (specific to LINK classic)
91100841	0, 2	Device reset completed
91100842	0, 2	Device reset error
91100871	0, 2	Odometer change acknowledged
91100872	0, 2	Odometer change error
91100874	0, 2	Odometer function disabled
91100876	0, 2	Odometer
91100877	0, 2	Odometer query - error
91100878	0, 2	Odometer query - error due to private mode
91100879	0, 2	Odometer change not allowed as a digital tachograph is connected
91100881	0, 2	Ignition time counter set to ignition on time and duration
91100882	0, 2	Set ignition time counter - error
91100884	0, 2	Ignition time counter disabled
91100886	0, 2	Ignition time counter is ignition on time and duration
91100887	0, 2	Query ignition time counter - error
91100888	0, 2	Query ignition time counter - error (private mode)
101100550	0, 2	Trip distance, duration, idling information, maximum speed, average speed, GPS coverage
101100551	0, 2	This message is delivered when either no GPS signal or partly no GPS signal was available during the trip.

Message types

msg_type	Available in message queue	Description
101100552	0, 2	Minor vehicle action (gives a summary of fuel usage and distance that do not qualify as a trip)
101100554	0, 2	Trip, distance, duration, idling information, fuel usage, maximum speed, GPS coverage
106010550	0, 2	Trip distance, duration, idling information, maximum speed, average speed, GPS coverage
106010551	0, 2	This message is delivered when either no GPS signal or partly no GPS signal was available during the trip.
106010552	0, 2	Minor vehicle action (gives a summary of fuel usage and distance that do not qualify as a trip)
110000729	0, 2, 4	Order sent (outbound)
110000730	0, 2, 4	Order accepted
110000731	0, 2, 4	Order rejected
110000732	0, 2, 4	Order started
110000733	0, 2, 4	Order completed
110000734	0, 2, 4	Order cancelled
110000735	0, 2, 4	Order modified (outbound)
110000736	0, 2, 4	Cancel order (outbound)
110000737	0, 2, 4	Order deleted (outbound)
110000738	0, 2, 4	All orders deleted (outbound)
110000760	0, 2, 4	Order state
110000761	0, 2, 4	Estimated time of arrival to order destination
110000762	0, 2, 4	Time left until order destination is reached
110000763	0, 2, 4	Estimated time of arrival without order information
110000764	0, 2, 4	List of invalid order waypoints
110000765	0, 2, 4	Appended order waypoints
110000766	0, 2, 4	Estimated time of arrival without distance to destination information

msg_type	Available in message queue	Description
110000767	0, 2, 4	Estimated time of arrival without order and without distance information
111100730	0, 2, 4	Order accepted
111100731	0, 2, 4	Order rejected
111100732	0, 2, 4	Order started
111100733	0, 2, 4	Order completed
111100734	0, 2, 4	Order cancelled
111100735	0, 2, 4	Arrived at loading site (specific to LINK classic)
111100736	0, 2, 4	Begin loading (specific to LINK classic)
111100737	0, 2, 4	End loading (specific to LINK classic)
111100738	0, 2, 4	Departure from loading site (specific to LINK classic)
111100739	0, 2, 4	Arrival at unloading site
111100741	0, 2, 4	End unloading
111100742	0, 2, 4	Departure from unloading site
111100743	0, 2, 4	Delay announcement indicating time
111100744	0, 2, 4	Begin waiting
111100745	0, 2, 4	End waiting
111100746	0, 2, 4	Additional costs (type, amount, currency)
111100747	0, 2, 4	Arriving at depot
111100749	0, 2, 4	Statustext

Message types

JSON objects

Parameters specific to acc_events

Parameter	Type	Description
type	int	The type of acceleration. Valid values: • `0` - Acceleration • `1` - Braking • `2` - Steering left • `3` - Steering right
maxacc	int	Maximum acceleration, in milli-*g*.
avgacc	int	Average acceleration, in milli-*g*.
dir	int	Direction of the acceleration, 0 ... 360.
startspeed	float	Speed at the beginning of the acceleration event, in km/h.
maxaccspeed	float	Maximum acceleration speed.
endspeed	float	Speed at the end of the acceleration event, in km/h.
lat	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
lon	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
st	dateTime	Time at which the acceleration event started.
maxacct	dateTime	Maximum acceleration time.
et	dateTime	Time at which the acceleration event ended.
dur	int	Duration of the acceleration event in milliseconds. The maximum precision is 100ms.

Parameters specific to app_data_in

Parameter	Type	Description
app_id	string	An identifier to mark the application on the device.
app_data	string	Base64 encoded aux device data received from device.

Parameters specific to app_data_out_status

Parameter	Type	Description
status	int	Result status of sending operation. Valid values: • `0` - Data was sent successfully. • `1` - Device has rejected the message, because the internal buffer is full. • `2` - The last sent but rejected message would now fit into the buffer of the device.

Parameters specific to aux_device_data_quota

Parameter	Type	Description
max	int	Maximum allowed data volume for a period.
used	int	Used volume in current period.

Parameters specific to app_data_quota

Parameter	Type	Description
max	int	Maximum allowed data volume for a period.
used	int	Used volume in current period.

Parameters specific to app_data_reference

Parameter	Type	Description
correlation_id	int	User provided correlation ID of outgoing messages.
reference_id	int	Internal ID provided by Webfleet and device. Available for outgoing and incoming messages.

Parameters specific to aux_device_data_in

Parameter	Type	Description
device_id	string	Identifies the sending third party device. In the case of Bluetooth this is the MAC address.
aux_data	string	Base64 encoded aux device data received from LINK.

Parameters specific to aux_device_data_out_status

Parameter	Type	Description
status	int	Result status of sending operation. Valid values: • `0` - data was sent successfully. • `1` - LINK has rejected the message, because the internal buffer is full. • `2` - the last sent but rejected message would now fit into the buffer of the LINK.

Parameters specific to aux_device_data_reference

Parameter	Type	Description
correlation_id	int	User-provided correlation ID of outgoing messages.
reference_id	int	Internal ID provided by Webfleet and LINK. Available for outgoing and incoming messages.

Parameters specific to ep_paireddata

Parameter	Type	Description
ep_btaddress	String	The Bluetooth address of the ecoPLUS.
version_info	String	

Parameters specific to ep_snapshotdata

Parameter	Type	Description
avgfuelusage_distance	int	in ml/100km, only returned if distance is greater than 0.
avgfuelusage_time	int	in ml/h

Parameter	Type	Description
rpm	int	revolutions per minute
gear	int	the current gear

Parameters specific to aux_device_data_in

Parameter	Type	Description
distance	int	The difference between `start_odometer` and `end_odometer`. Unit of measurement is 'meters'.
fuel_usage	float	Fuel consumption in litres **Note**: Max. three decimals allowed.
co2	int	Carbon dioxide emissions in grams.

Parameters specific to fms_fuel_difference

Parameter	Type	Description
slopetype	int	Fuel jump or continuous event. Valid values: • `0` - leap • `1` - continuous
timespan	int	Duration of event if continuous, in seconds.
fuellevel_start	int	Fuel level at start of the event in per mill.
fuellevel_end	int	Fuel level at end of the event in per mill.

Parameters specific to fms_snapshotdata

Parameter	Type	Description
fuel_level	int	Current fuel level in per mill.

Parameters specific to fms_tripdata

Parameter	Type	Description
fuel_usage	int	Fuel usage in ml.

Parameter	Type	Description
rpm_violation_duration	int	The duration during the trip the engine speeds over a threshold, in seconds.

Parameters specific to troublecodes

Parameter	Type	Description
dtclist	Array of string	Contains all current Diagnostic Trouble Codes. Optional.
ptclist	Array of sting	Contains all current Pending Trouble Codes. Optional.
milon	Boolean	Flag indicating whether the malfunction indicator is lit. Optional.

Parameters specific to er_tripdata

Parameter	Type	Description
fuel_usage	float	Fuel usage in ml.
co2	int	Carbon dioxide emissions in grams. Optional.
energy_consumption_driving	float	Energy used for driving in kWh.
energy_consumption_other	float	Energy used for heating, climate, etc. in kWh.
energy_recovered	float	Energy recovered from braking in kWh.

Objects

showObjectReportExtern

Description

Delivers a list of all objects.

Provides access to object master data and the most recent position information available for each object.

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#showObjectReport
Request limits	6 requests / minute

Use Message queues instead of `showObjectReportExtern`, if you need all tracking data for the objects in your account.

Object types – An object in Webfleet always has a certain type (vehicle, person, asset, etc.). Every type defines special properties. These properties are listed by special actions (`showVehicleReportExtern` etc.), whereas this action lists all objects with type independent properties like number, name and description.

Code sample 4-1: Request

CSV request

```
https://csv.webfleet.com/extern?account=xxx&apikey=xxx&lang=de&action=showObjectReportExtern&filterstring=Demo

SOAP request
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
 xmlns:ser="http://connect.webfleet.tomtomwork.com/services">
 <soap:Header/>
  <soap:Body>
   <ser:showObjectReport>
    <aParm>
     <accountName>xxx</accountName>
     <userName>xxx</userName>
     <password>xxx</password>
     <apiKey>xxx</apiKey>
    </aParm>
    <gParm>
     <locale>DE</locale>
     <timeZone>Europe_Berlin</timeZone>
    </gParm>
    <objectFilter>
     <filterCriterion>Demo</filterCriterion>
    </objectFilter>
   </ser:showObjectReport>
  </soap:Body>
</soap:Envelope>
```

Parameters

showObjectReportExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to showObjectReportExtern:

Parameter	Type	Description
filterstring	string	An arbitrary pattern that is located in the object data.
objectgroupname	string	A name of an object group.
ungroupedonly	string	Allows to filter on objects that are currently not assigned to a certain object group. Only valid value is `true`. This parameter overrules the group name filter parameter. This parameter is optional.
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
externalid	string	For future use. This parameter is optional.

Parameters specific to showObjectReportExtern

Result

Result for showObjectReportExtern:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectname	string	Display name of an object.
objectclassname	string	
objecttype	string	Can be empty or contain the valid values listed in `vehicletype`, see updateVehicle.
description	string (500)	Vehicle description

Result for showObjectReportExtern

Parameter	Type	Description
lastmsgid	string	The ID of the last message received from or sent to this object. `mgsid` identifies a message and is system-wide unique. Possible values are in the range 0 # msgid # 2^{64}-1.
deleted	boolean	
msgtime	string	
longitude	string	Geographic longitude in the form GGG° MM′ SS.S″ E/W in the WGS84 coordinate system.
latitude	string	Geographic latitude in the form GGG° MM′ SS.S″ N/S in the WGS84 coordinate system.
postext	string	
postext_short	string	
pos_time	dateTime	Time that is related to the last known position of the vehicle.
speed	string	
course	nonNegativeInteger	Compass direction in degrees (0° … 360°).
direction	nonNegativeInteger	Cardinal and intercardinal compass directions, derived from `course`. Valid values: • `1` - North • `2` - Northeast • `3` - East • `4` - Southeast • `5` - South • `6` - Southwest • `7` - West • `8` - Northwest
quality	string	Not supported anymore. This was only specific to LINK classic.
satellite	string	Not supported anymore. This was only specific to LINK classic.

Result for showObjectReportExtern

Parameter	Type	Description
status	string	GPS signal status indicator, possible values are: • `A` - ok: a current GPS fix is available with this message. • `V` - warn: a current GPS fix is available, but it might be inaccurate. • `L` - last: the GPS position is from the last known good fix, a current fix is not (yet) available. • `0` - invalid: no GPS fix available or GPS has no meaning for this message.
dest_latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd).
dest_longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd).
dest_text	string	
dest_eta	dateTime	Date and time of the estimated time of arrival (eta) as reported from the navigation system.
dest_isorder		
dest_addrnr		
orderno	string (20)	Account-unique order id, case sensitive. Maximum capacity is 20 bytes, actual number of characters depends on the byte count consumed by the UTF-8 encoding of those characters.
driver	string (20)	Account-unique driver number, case-sensitive.
drivername	string	Display name of a driver.
drivertelmobile	string	Mobile phone number of the driver.
codriver	string (20)	Account-unique driver number, case-sensitive.
codrivername	string	Display name of a co-driver.
codrivertelmobile	string	Mobile phone number of the co-driver.

Result for showObjectReportExtern

Parameter	Type	Description
driver_currentworkstate	nonNegativeInteger	Valid values: • `0` - Unknown • `1` - Free time (PND only) • `2` - Pause (PND and digital tachograph) • `3` - Standby (digital tachograph only) • `4` - Working (PND only) • `5` - Driving (digital tachograph only) • `6` - Other work (digital tachograph only) When no driver has signed on, `driver` is empty, the state shown indicates the current working state of the vehicle.
codriver_currentworkstate	nonNegativeInteger	Valid values: • `0` - Unknown • `1` - Free time (PND only) • `2` - Pause (PND and digital tachograph) • `3` - Standby (digital tachograph only) • `4` - Working (PND only) • `5` - Driving (digital tachograph only) • `6` - Other work (digital tachograph only)
rll_btaddress	string (17)	Remote LINK Bluetooth address. The IEEE 802 MAC-48/EUI-48 address of a Bluetooth device, formatted as six groups of two hexadecimal digits, separated by colons (:). *Example: 00:13:6C:88:26:0B*
odometer	string	The current odometer value in **100 meter**.
ignition	nonNegativeInteger	The current state of the ignition. • `0` - Ignition off • `1` - Ignition on
ignition_time	string	The time the last ignition state was reported.
dest_distance	string	The distance to the current navigation destination in meters. If empty, no navigation destination was specified.

Result for showObjectReportExtern

Parameter	Type	Description
tripmode	nonNegativeInteger	Valid values: • `0` - Unknown trip type • `1` - Private trip • `2` - Business trip • `3` - Commute trip • `4` - Correction trip (user changed the odometer value manually)
standstill	nonNegativeInteger	Indicates if the vehicle is standing still. Valid values: • `0` - not standing still • `1` - standing still
pndconn	nonNegativeInteger	Indicates if a navigation device is connected to the LINK device. Valid values: • `0` - not connected • `1` - connected
latitude_mdeg	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
longitude_mdeg	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Result for showObjectReportExtern

Parameter	Type	Description
fuellevel	int	The current fuel level in per mill. **Note**: This information is only available for LINK 510/710 devices connected to the vehicle's FMS.
externalid	string	For future use.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.
codriveruid	string	A unique, unchangeable identifier for the indicated codriver, automatically generated. Can be used alternatively to `codriver`.
driverkey_deviceaddress	string	JSON encoded string with an array of strings. Each string contains the address of the driver key reading device which is associated with the vehicle. *Example:* `['00:21:3E:B3:8D:EA']`
fuellevel_milliliters	string	The current fuel level in milliliters. **Note**: This information is only available for LINK 204.
engine_operation_time	nonNegativeInteger	Returns the accumulated time of operation of the engine in seconds. **Note**: This information is only available for some LINK devices e.g. LINK 7x0 (with input configured for 'Engine state') or LINK devices that are connected to FMS.
odometer_long	int	The current odometer value **in meter**.

Result for showObjectReportExtern

showVehicleReportExtern

Description

Lists all objects of type 'Vehicle'.

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#showVehicleReport

Request limits	10 requests / minute

Parameters

showVehicleReportExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to showVehicleReportExtern:

Parameter	Type	Description
filterstring	string	An arbitrary pattern that is located in the object data.
objectgroupname	string	A name of an object group.
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
externalid	string	For future use. This parameter is optional.

Result

Result for showVehicleReportExtern:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectname	string	Display name of an object.
licenseplatenum-ber	string	
vehicletype	string	Can be empty or contain the valid values listed in `vehicletype`, see updateVehicle.
width	string	
length	string	
height	string	
maxweight	string	
netweight	string	

Parameter	Type	Description
netload	string	
power	string	
numaxes	string	
identnumber	string (20)	If connected to FMS or Tachograph or when using a LINK 105 or ecoPLUS this field contains the VIN (Vehicle Identification Number) and cannot be changed.
registrationdate	string	
vh_avgfuelusage	int	Vehicle-based fuel consumption reference value, in ml/100 km
ep_avgfuelusage	int	ecoPLUS-based fuel consumption reference value, in ml/100 km
fl_avgfuelusage	int	Fleet-based fuel consumption reference value, in ml/100km.
vh_fueltype	int	Vehicle configured fuel type. Valid values: • `0` - unknown • `1` - diesel • `2` - gasoline • `3` - lpg • `4` - hybrid petrol • `5` - hybrid diesel • `6` - electric • `7` - cng • `8` - lng • `9` - hydrogen • `10` - plug-in hybrid electric • `11` - bioethanol • `12` - biodiesel_fame (Fatty acid methyl esters) • `13` - biodiesel_fame_uco (Fatty acid methyl esters, Used cooking oil) • `14` - hvo (Hydrotreated vegetable oil) • `15` - commercial_diesel • `16` - cbg (Bio-CNG) • `17` - lbg (Bio-LNG)
ep_fueltype	int	Fuel type determined by ecoPLUS. Valid values: • `1` - diesel • `2` - gasoline

Parameter	Type	Description
fl_fueltype	int	Fleet based fuel type. Valid values: • `0` - unknown • `1` - diesel • `2` - gasoline • `3` - lpg • `4` - hybrid petrol • `5` - hybrid diesel • `6` - electric • `7` - cng • `8` - lng • `9` - hydrogen • `10` - plug-in hybrid electric • `11` - bioethanol • `12` - biodiesel_fame (Fatty acid methyl esters) • `13` - biodiesel_fame_uco (Fatty acid methyl esters, Used cooking oil) • `14` - hvo (Hydrotreated vegetable oil) • `15` - commercial_diesel • `16` - cbg (Bio-CNG) • `17` - lbg (Bio-LNG)
enginesize	nonNega-tiveInteger	Engine size in ccm.
ep_btaddress	string (17)	ecoPLUS Bluetooth address. The IEEE 802 MAC-48/EUI-48 address of a Bluetooth device, formatted as six groups of two hexadecimal digits, separated by colons (:). *Example: 00:13:6C:88:26:0B*
speedlimit	float	Speed limit in km/h, min: 0, max: 300
vehiclecolor	string	Valid values: • `white` • `grey` • `black` • `red` • `orange` • `yellow` • `green` • `blue`
description	string (500)	Vehicle description.

Parameter	Type	Description
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
vh_rpmlimit	int	Indicates the engine's maximum allowed revolutions per minute. This value is used to report RPM violations. Supported by LINK 510/710 connected to FMS.
fl_rpmlimit	int	Indicates the engine's maximum allowed revolutions per minute defined for the whole vehicle fleet. This value is used to report RPM violations. Supported by LINK 510/710 connected to FMS.
externalid	string	For future use.
obu_btaddress	string (17)	The Bluetooth address of the LINK device. This information is currently available for LINK 510/710/410 with firmware version 3.4 or higher. The IEEE 802 MAC-48/EUI-48 address of a Bluetooth device, formatted as six groups of two hexadecimal digits, separated by colons. *Example: 00:13:6C:88:26:0B*
fueltanksize	int	Size of the fuel tank in litres.
ep_type	nonNegativeInteger	Type of the OBDII dongle (eco periphery). Valid values: • `1` - ecoPLUS • `2` - LINK 105
manufacturedyear	int	The year in which the vehicle was manufactured. This parameter is optional.
fuelreference	nonNegativeInteger	Indicates to which type of fuel reference to refer to. Valid values: • `0` - The reference value defined for the fleet. • `1` - The reference value definded for the individual vehicle. • `2` - ecoPLUS

Parameter	Type	Description
accelerationvehicletype	string	The 'acceleration vehicle type' is used to calibrate the acceleration event handling. By default it is determined by the vehicle type and can be overwritten via updateVehicle Valid values: • `heavy_weight` • `medium_weight` • `light_weight`

showNearestVehicles

Description

showNearestVehicles retrieves a list of vehicles that are within a defined radius around the indicated location. This action delivers details on the routed distance and the routed time for the 8 nearest vehicles. The list is sorted by:

1. Object state.
2. Linear distance to the location.
3. Object number.

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#showNearestVehicles
Request limits	10 requests / minute

Parameters

showNearestVehicles requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to showNearestVehicles:

Parameter	Type	Description
objectgroupname	string	A name of an object group.
addrno	string (10)	Identifying number of an address. Unique within an account. Is optional if you are using `latitude` and `longitude` to specify the destination.

Parameter	Type	Description
latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia. This parameter is optional if `addrno` is indicated.
longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia. This parameter is optional if `addrno` is indicated.
priority	int	Defines the priority of attributes by which the results shall be sorted. Valid values: • `1` - Object state / linear distance / object number • `2` - Linear distance / object number The default is `1`. This parameter is optional. **Note**: The first 8 results will be returned including routed distance and estimated travel time.
maxdistance	int	Defines the maximum linear distance from the indicated location, in meters. Vehicles that are further away from the location than the defined distance will not be included in the result list. This parameter is optional.

Result

Result for showNearestVehicles:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Parameter	Type	Description
objectstate	int	The status the vehicle currently has. Valid values: • `1` - available • `2` - has order • `3` - offline
latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia. This parameter is optional if `addrno` is indicated.
longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia. This parameter is optional if `addrno` is indicated.
lineardistance	int	The linear distance between the vehicle and the defined location, in meters.
routedistance	int	The distance calculated based on the roads to take from the actual position to the defined location considering HD Traffic information, in meters.
routetime	int	The time that is needed to get to the defined location using the calculated route, see `routedistance`, in seconds.

showContracts

Description

This action shows contract and device data for selected devices.

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#showContracts
Request limits	6 requests / minute

Parameters

showContracts requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to showContracts:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. This parameter is optional.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Result

Result for showContracts:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
contractno	string	Contract unit number
begin_date	dateTime	ISO 8601-formatted date and time in the UTC time-zone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00
end_date	dateTime	ISO 8601-formatted date and time in the UTC time-zone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00
tariff_name	string	
activation_key	string	
iccid	string	SIM card number
roaming_enabled	boolean	
obu_type	string	
obu_serialno	string	
obu_osversion	string	

Parameter	Type	Description
obu_appversion	string	
pnd_type	string	
pnd_serialno	string	
pnd_ncversion	string	
pnd_appversion	string	
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
referenceno	string	Reference number for the object individually assigned by the customer.
pnd_mapversion	string	Name and version of the map, that is currently installed on the navigation device. This column can be empty, if there is no navigation device in use for this contract. This column may contain multiple values separated by comma if there is more than one map installed on the navigation device.
msisdn	string	The MSISDN of the SIM card. This value is optional and might not be used for certain device types.

updateVehicle

Description

This action allows the update of vehicle details.

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#updateVehicle
Request limits	10 requests / minute

Parameters

updateVehicle requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to updateVehicle:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`. If both parameters `objectuid` and `objectno` are given, then `objectno` will be updated. `objectno` is required to have between 1 and 10 characters and must not already exist.
vehicletype	string	Case-sensitive. For valid values, see the List of vehicle types at the end of this section.
vehiclecolor	string	Case-sensitive. Valid values: • `white` • `grey` • `black` • `red` • `orange` • `yellow` • `green` • `blue`
identnumber	string (20)	If connected to FMS or Tachograph or when using a LINK 105 or ecoPLUS this field contains the VIN (Vehicle Identification Number) and cannot be changed.
registrationdate	string (20)	
denotation	string (30)	Changes the value of `objectname`.
licenseplatenumber	string (20)	
speedlimit	float	Speed limit in km/h. • min: 0 • max: 300 This parameter is optional.
fuelconsumption	float	Fuel consumption in l/100km. • min: 0 • max: 100 This parameter is optional.

Parameter	Type	Description
fueltype	int	Case-sensitive. Valid values: • `0` - unknown • `1` - diesel • `2` - gasoline • `3` - lpg • `4` - hybrid petrol • `5` - hybrid diesel • `6` - electric • `7` - cng • `8` - lng • `9` - hydrogen • `10` - plug-in hybrid electric • `11` - bioethanol • `12` - biodiesel_fame (Fatty acid methyl esters) • `13` - biodiesel_fame_uco (Fatty acid methyl esters, Used cooking oil) • `14` - hvo (Hydrotreated vegetable oil) • `15` - commercial_diesel • `16` - cbg (Bio-CNG) • `17` - lbg (Bio-LNG)
netweight	float	in kg
maxweight	float	in kg
netload	float	in kg
maxload	float	in kg
numaxles	nonNegativeInteger	
length	nonNegativeInteger	in mm
width	nonNegativeInteger	in mm
height	nonNegativeInteger	in mm
description	string (500)	Vehicle description
power	float	in kW
enginesize	nonNegativeInteger	in ccm

Parameter	Type	Description
reset_fuelavg	boolean	Reset switch for measured or calculated average fuel consumption. • `1` - reset
odometer	nonNegativeInteger	Updated odometer value in meters.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`. Can also be used to update `objectno`, please refer to the description of parameter `objectno`.
rpmlimit	int	Indicates the engine's maximum allowed revolutions per minute. This value is used to report RPM violations. Supported by LINK 510/710 connected to FMS. Valid values: • `<int value>` sets the vehicle specific RPM limit • '`null`' or empty parameter value removes the vehicle specific RPM limit. **Note**: If the vehicle-specific value is removed, the fleet-based value will be used for RPM violation reporting.
ep_btaddress	string (17)	ecoPLUS Bluetooth address. The IEEE 802 MAC-48/EUI-48 address of a Bluetooth device, formatted as six groups of two hexadecimal digits, separated by colons. *Example: 00:13:6C:88:26:0B* **Note**: If empty, the existing value will be deleted. When omitted, the existing value will not change. This parameter is optional.

Parameter	Type	Description
rll_btaddress	string (17)	Remote LINK Bluetooth address. The IEEE 802 MAC-48/EUI-48 address of a Bluetooth device, formatted as six groups of two hexadecimal digits, separated by colons (:). *Example: 00:13:6C:88:26:0B* **Note**: If empty the existing value will be deleted. When omitted the existing value will not change. This parameter is optional.
externalid	string	For future use.
fueltanksize	int	Size of the fuel tank in litres. Maximum allowed value is 65535.
ep_type	nonNegativeInteger	Type of the OBDII dongle (eco periphery). Valid values: • `1` - ecoPLUS • `2` - LINK 105
manufacturedyear	int	The year in which the vehicle was manufactured. This parameter is optional.
fuelreference	nonNegativeInteger	Indicates to which type of fuel reference to refer to. Valid values are: • `0` - The reference value defined for the fleet. • `1` - The reference value defined for anindividual vehicle. • `2` - ecoPLUS
accelerationvehicletype	string	The 'acceleration vehicle type' is used to calibrate the acceleration event handling. By default it is determined by the vehicle type. It can be overwritten with this action, but will be set back, when the vehicle type is changed. Case-sensitive. Valid values: • `heavy_weight` • `medium_weight` • `light_weight`

List of vehicle types

The following list shows valid values for parameter `vehicletype` including vehicle types, special vehicle types and devices:

- **Vehicle types** with icons that have a separate colouring option in the user interface; all eight colors are supported:
    - ape (shown as 'Three wheeler' in the UI)
    - barge
    - bike (shown as 'Motor bike' in the UI)
    - boom_lift
    - boom_lift_truck
    - bus
    - caddy (shown as 'Small van' in the UI)
    - car
    - car_station_wagon
    - concrete_lorry
    - containership (Shown as 'Cargo ship' in the UI)
    - crane
    - deposit_tipper
    - dump_truck
    - dump_truck_open
    - excavator
    - garbage_truck
    - heavy_truck
    - heavytruck_us
    - heavyweight_truck
    - heavyweight_truck_trailer
    - lighttruck
    - light_truck_pallet
    - loader (shown as 'Wheel loader' in the UI)
    - medium_truck
    - multicar (shown as 'Snow plough' in the UI)
    - multivan
    - pallet_truck
    - pickup
    - smart
    - sprinter (shown as 'Van / Light truck' in the UI)
    - street_sweeper
    - suv
    - tanker_truck
    - telescopic_handler
    - tractor
    - trailer
    - train
    - truck
    - truck_loader_arm
    - truck_trailer
    - truck_with_trailer
    - truck_wrecker

    - van
    - vespa
    - wrecker
    - yacht

- **Special vehicle types** – not all colours are supported; allowed colours are given in brackets:
    - ambulance (yellow, blue, red, white)
    - firetruck (red)
    - police (blue, green)
    - taxi (green, blue, yellow, orange, red, black, grey, white)
    - taxi_uk (black)

- **Devices** with separate colouring option in the user interface, all eight colours are supported:
    - link
    - ecoPLUS
    - link100
    - link510
    - link610
    - link710
    - phone
    - tablet

showObjectGroups

Description

This actions retrieves a list of all object groups.

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#showObjectGroupReport
Request limits	10 requests / minute

Parameters

showObjectGroups requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to showObjectGroups:

Parameter	Type	Description
objectgroupname	string	The given entry filters the result list on the object groups with matching names. Can be used alternatively to `objectgroupuid`.

Parameter	Type	Description
objectgroupuid	string	A unique, unchangeable identifier for the indicated object group, automatically generated. Can be used alternatively to `objectgroupname`.

Result

Result for showObjectGroups:

Parameter	Type	Description
objectgroupname	string	Name of the object group.
objectcount	string	The number of objects assigned to the indicated group. The number of objects that are not assigned to a group is not returned.
objectgroupuid	string	A unique, unchangable identifier for the indicated object group, automatically generated.

showObjectGroupObjects

Description

Lists the assignments of objects to object groups. Each object can be in no, one or more than one group.

Using both parameters `objectgroupname` and `onbjectno`, only according matches are returned.

Technical details:

SOAP endpoint address / function name	objectsAndPeopleReportingService#showObjectGroupObjectReport
Request limits	10 requests / minute

Parameters

showObjectGroupObjects requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to showObjectGroupObjects:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. This parameter is optional.

Parameter	Type	Description
objectgroupname	string	The given entry filters the result list on the object groups with matching names, case-sensitive. This parameter is optional.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Result

Result for showObjectGroupObjects:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account. Case-sensitive.
objectname	string	Display name of an object.
objectgroupname	string	
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
objectgroupuid	string	A unique, unchangable identifier for the indicated object group, automatically generated.

attachObjectToGroup

Description

This action assigns an object to a specific object group.

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#attachObjectToGroup
Request limits	1000 requests / hour

Parameters

attachObjectToGroup requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to attachObjectToGroup:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectgroupname	string	Name of the object group, case-sensitive. This parameter is required.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

detachObjectFromGroup

Description

This action detached an object from a specific object group.

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#detachObjectFromGroup
Request limits	1000 requests / hour

Parameters

detachObjectFromGroup requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to detachObjectFromGroup:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectgroupname	string	Name of the object group, case-sensitive. This parameter is required.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

insertObjectGroup

Description

This action creates an object group. The group name must not start with '**sys$**'.

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#insertObjectGroup
Request limits	100 requests / 30 minutes

Parameters

insertObjectGroup requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to insertObjectGroup:

Parameter	Type	Description
objectgroupname	string	Name of the object group, case-sensitive. This parameter is required.

deleteObjectGroup

Description

This action deletes an object group and the assignments of all objects assigned to that group. The objects detached through this action are not being deleted.

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#deleteObjectGroup
Request limits	100 requests / 30 minutes

Parameters

deleteObjectGroup requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to deleteObjectGroup:

Parameter	Type	Description
objectgroupname	string	Name of the object group, case-sensitive. This parameter is required.

updateObjectGroup

Description

This action allows to update the name of the object group while retaining the assignment of objects to that group. The group name must not start with '**sys$**' .

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#updateObject-Group
Request limits	100 requests / 30 minutes

Parameters

updateObjectGroup requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to updateObjectGroup:

Parameter	Type	Description
objectgroupname	string	Name of the object group, case-sensitive. This parameter is required.
objectgroup-name_old	string	Name of the object group, case-sensitive. This parameter is required.

switchOutput

Description

Using `switchOutput` you can control the digital output of the LINK classic and the LINK 510/710 to be on or off (1 or 0). You first need to enable the digital output of the LINK classic/LINK 510/LINK 710 in the device configuration dialogue in the Webfleet user interface.

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#switchOutput
Request limits	10 requests / minute

Parameters

switchOutput requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific for switchOuput:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
status	int	Valid values: • `1` • `0` This parameter is required.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
output_name	string (50)	Identifies the output to be switched. The name can be configured in the Webfleet user interface. This parameter is optional. If omitted, the first configured output is used. It is recommended to always use the `output_name` parameter instead of relying on configuration order.

showWakeupTimers

Description

This action returns a list of days and times that are configured on the LINK 7xx/6xx/5xx/4xx/3xx to wake up the device.

The LINK device can be set to wake up at several times per day. The times defined apply to all the days that are enabled for wake-up.

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#showWakeupTimers
Request limits	10 requests / minute

Parameters

showWakeupTimers requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific for showWakeupTimers:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Result

Result for showWakeupTimers:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated.
objectname	string (200)	Display name of an object.
mon	boolean	Indicates whether the LINK device is set to wake up at that day. Valid values: • `0` - LINK set to not wake up • `1` - LINK set to wake up
tue	boolean	See `mon`.
wed	boolean	See `mon`.
thu	boolean	See `mon`.
fri	boolean	See `mon`.
sat	boolean	See `mon`.
sun	boolean	See `mon`.
time1	String (5)	Wake up time in `format hh:mm`, in the local time zone of the user. If empty, the wake up time is not set.
time2	String (5)	See `time1`.
time3	String (5)	See `time1`.
time4	String (5)	See `time1`.

Parameter	Type	Description
time5	String (5)	See `time1`.

updateWakeupTimers

Description

Using `updateWakeupTimers` you can set and change the days and times the specified LINK 7xx/6xx/5xx/4xx/3xx device shall automatically wake up.

The LINK device can be set to wake up at several times per day. The times defined apply to all the days that are enabled for wake-up.

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#updateWakeup-Timers
Request limits	10 requests / minute

Parameters

updateWakeupTimers requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific for updateWakeupTimers:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
mon	boolean	Indicates whether the LINK device is set to wake up at that day. The parameter can be omitted to leave its value unchanged. Valid values: • `0` - LINK set to not wake up • `1` - LINK set to wake up This parameter is optional.
tue	boolean	See `mon`.
wed	boolean	See `mon`.

Parameter	Type	Description
thu	boolean	See `mon`.
fri	boolean	See `mon`.
sat	boolean	See `mon`.
sun	boolean	See `mon`.
time1	string (5)	Wake up time in format `hh:mm`, in the local time zone of the user. If empty the existing value will be deleted. When omitted, its existing value will not change.
time2	string (5)	See `time1`.
time3	string (5)	See `time1`.
time4	string (5)	See `time1`.
time5	string (5)	See `time1`.

getObjectFeatures

Description

`getObjectFeatures` returns all additional features that are booked for the object. If you specify a feature, `getObjectFeatures` returns all objects that have the indicated additional feature booked.

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#getObjectFeatures
Request limits	10 requests / minute

Parameters

getObjectFeatures requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific for `getObjectFeatures`:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.

Parameter	Type	Description
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
externalid	string	For future use. This parameter is optional.
feature	string	Identifies the feature (additional service): • `linkconnect` • `hdtracking` • `waypoints` • `tachodownload` • `drivingtime` • `plugins` • `orderscheduleoptimization` • `fmsdirect` • `tpms` • `eld` • `video` • `vehiclechecklist` • `proappnav` • `coldchain` • `oemconnect`

Result

Result for `getObjectFeatures`:

```
feature
```

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated.

Parameter	Type	Description
feature	string	Identifies the feature (additional service): • `linkconnect` • `hdtracking` • `waypoints` • `tachodownload` • `drivingtime` • `plugins` • `orderscheduleoptimization` • `fmsdirect` • `tpms` • `eld` • `video` • `oemconnect` • `proappnav` • `coldchain` • `oemconnect`
volume_max	int	Quota limit (if applicable).
volume_used	int	Currently used quota (if applicable).

Parameter	Type	Description
feature_detail	string	The name of the exact variant of an additional feature. This parameter is only available for additional features with multiple variants. Possible values, depending on `feature` hdtracking • `hdtracking_1` • `hdtracking_2` • `hdtracking_3` • `hdtracking_4` • `hdtracking_5` • `e_toll` • `e_toll_pl` linkconnect • `linkconnect_xs` • `linkconnect_s` • `linkconnect_m` • `linkconnect_l` • `linkconnect_xl` tachodownload • `tachoshare_tis_web_plus` • `tachoshare_tis_web` • `tachoshare_plus` • `tachoshare` • `tachomanager` • `tachomanager_plus` video • `video_basic` • `video_plus` • `video_live` • `video_basic_int` • `video_plus_int` • `video_live_int` eld • `eld_mex` • `eld_usa`

updateContractInfo

Description

Using `updateContractInfo` you can update data that refer to the contract.

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#updateContractInfo
Request limits	10 requests / minute

Parameters

updateContractInfo requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific for updateContractInfo:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
referenceno	string (20)	Reference number for the object individually assigned by the customer. This parameter is required.

getObjectCanSignals

Description

`getObjectSignals` returns the latest received CAN signal values for an object in an account. This action will also be useful for FMS when LINK 610/710 are used.

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#getObjectSignals
Request limits	6 requests / minute

Parameters

getObjectCanSignals requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getObjectCanSignals:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Result

Result for getObjectCanSignals:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
cansignaltime	dateTime	The time the CAN signal was read from the CAN bus of the vehicle.
cansignaltype-name	string	Corresponds to a defined list of CAN signal types. All available signals are listed in Appendix F.
cansignaltype-description	string	Description of CAN signal type
cansignaldatatype	nonNega-tiveInteger	• 1 = Integer • 2 = Decimal • 3 = Boolean
cansignalvalue	float	The value of the CAN signal. Either 0 = false, 1 = true for boolean values or a decimal number for other values like fuel level or battery voltage.

Parameter	Type	Description
cansignalunit	nonNegativeInteger	Indicates the unit of measurement in which `cansignalvalue` is provided. Valid value are: • 0 - Special use (such as bit masks etc.) • 1 - Kilometer per hour • 2 - One per minute • 3 - Kilometer • 4 - Degree Celsius • 5 - Volt • 6 - On/Off • 7 - Percent • 8 - Ampere • 9 - Decimeter • 10 - Hours • 11 - Kilogram • 12 - Kilopascal • 13 - Milliliter • 14 - Liter • 15 - Liter per hour • 16 - Meter • 17 - Minutes • 18 - Newton meter • 19 - Rounds per minute • 20 - Seconds • 21 - Watt-hours

getObjectCanMalfunctions

Description

`getObjectCanMalfunctions` returns the current state of CAN based malfunctions of an object. This action will also be useful for FMS when a LINK 610/710 are used.

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#getObjectCanMalfunctions
Request limits	6 requests / minute

Parameters

getObjectCanMalfunctions requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getObjectCanMalfunctions:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`. This parameter is optional.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`. This parameter is optional.
objectgroupname	String	The given entry filters the result list on the object groups with matching names. Can be used alternatively to `objectgroupuid`. This parameter is optional.
objectgroupuid	String	A unique, unchangeable identifier for the indicated object group, automatically generated. Can be used alternatively to `objectgroupname`. This parameter is optional.
malfunctionlevel	nonNega-tiveInteger	Filter by malfunction level. All CAN based malfunctions of the same or higher level are returned. This parameter is optional.

Result

Result for getObjectCanMalfunctions:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Parameter	Type	Description
malfunctiontype	nonNega-tiveInteger	Malfunction types. Valid value are: • 10 = Battery • 20 = Battery voltage • 30 = Lights • 40 = Exhaust system • 50 = Tyre pressure • 60 = Brake fluid • 70 = Coolant level • 80 = Engine oil level • 90 = Engine oil pressure • 100 = Diesel particulate filter • 110 = Coolant temperature • 120 = Engine cooling • 130 = ABS • 140 = Brake system • 150 = Airbag • 160 = EPC • 170 = ESP • 180 = Preheat or engine malfunction • 190 = Brake pads • 200 = Engine/malfunction indicator lamp • 210 = Transmission failure • 220 = Low coolant level • 230 = Brake failure • 240 = Airbag • 250 = AdBlue level • 260 = Engine oil • 270 = Worn brake linings • 280 = Tyre failure • 290 = Engine oil level low • 300 = Engine emission system failure • 310 = Battery charging condition failure
malfunctionlevel	nonNega-tiveInteger	• 1 = OK • 2 = WARNING • 3 = ERROR

getElectricVehicleData

Description

`getElectricVehicleData` returns data for all electric vehicles in the account.

This action is supported by the following devices:

- Webfleet LINK 740
- TomTom LINK 710
- Webfleet LINK 640
- TomTom LINK 610
- TomTom LINK 710B
- Webfleet LINK 240
- Webfleet LINK 245
- TomTom LINK 210

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#getElectricVehicleData
Request limits	6 requests / minute

Parameters

getElectricVehicleData requires the following common parameters:

- Authentication parameters
- General parameters

Result

Result for `getElectricVehicleData`:

Parameter	Type	Description
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated.
batteryLevel	float	Current battery level in percent. Values are within [0.00, 100.00].
estimatedTimeToCharge	dateTime	The moment at which the battery will be fully charged. ISO 8601-formatted date and time.
isCharging	boolean	Status of charging. `True` when the vehicle is being charged, `false` otherwise.
range	int	Total range in meters. For hybrid vehicles this is electric range + fuel range.
lastModified	dateTime	Date and time of last signal received from the electric vehicle.
batteryCapacity	float	User defined battery capacity of the electric vehicle in kWh. Values are within [0.00, +infinity]

Parameter	Type	Description
batteryHealth	float	Current battery health in percentages. Values are within [0.00, 100.00]
connectorTypes	array[string]	A list of user defined charger connector types supported by the electric vehicle (may include additional adapters). The list is initially empty. Connector types can be configured in the Webfleet UI. Possible values: • `StandardHouseholdCountrySpecific` • `IEC62196Type1` • `IEC62196Type1CCS` • `IEC62196Type2CableAttached` • `IEC62196Type2Outlet` • `IEC62196Type2CCS` • `IEC62196Type3` • `Chademo` • `GBT20234Part2` • `GBT20234Part3` • `IEC60309AC3PhaseRed` • `IEC60309AC1PhaseBlue` • `IEC60309DCWhite` • `Tesla`

getActiveAssetCouplings

Description

`getActiveAssetCouplings` returns the active couplings of vehicles and assets. Optionally also the last couplings can be retrieved.

Technical details

SOAP endpoint address / function name	assetCoupling#getActiveAssetCouplings
Request limits	10 requests / minute

Parameters

getActiveAssetCouplings requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to `getActiveAssetCouplings`:

Parameter	Type	Description
objectuid	string (30)	A unique, unchangeable identifier for the indicated vehcile or asset, automatically generated. This parameter is optional. If not provided, all active couplings of the account are returned. If view access is not granted, the result is empty.
includeLastCoupling	boolean	If set to `true`, also the last coupling will be returned in the response. This parameter is optional.

Result

Result for `getActiveAssetCouplings`:

Parameter	Type	Description
vehicleuid	string (30)	The coupled vehicle. A unique, unchangeable identifier, automatically generated.
assetuid	string (30)	The coupled asset. A unique, unchangeable identifier, automatically generated.
couplingtime	dateTime	The detection time of the coupling event. ISO 8601-formatted date and time in the UTC timezone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00
longitude	Integer	The longitude of the the coupling event location. Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.

Parameter	Type	Description
latitude	Integer	The latitude of the the coupling event location. Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.

getTelemetryDeviceBatteryState

Description

`getTelemetryDeviceBatteryState` returns the battery state of a telematics device.

This is only supported for devices which can operate in battery-powered mode (e.g. LINK 340, LINK 350).

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#getTelemetryDeviceBatteryState
Request limits	10 requests / minute

Parameters

getTelemetryDeviceBatteryState requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to `getTelemetryDeviceBatteryState`:

Parameter	Type	Description
objectuid	string (30)	A unique, unchangeable identifier for the indicated vehicle or asset, automatically generated.

Result

Result for `getTelemetryDeviceBatteryState`:

Parameter	Type	Description
objectUid	string (30)	A unique, unchangeable identifier for the indicated vehicle or asset, automatically generated.
levelPercentage	int	Current battery level in percentages. Values are within [0, 100].
chargingState	string	Charging state. Values are `CHARGING_ON` or `CHARGING_OFF`.
lastGpsTime	dateTime	Date and time of last GPS update received from the device.

getObjectDTCs

Description

`getObjectDTCs` returns diagnostic trouble code (DTC) information for an object. It allows to retrieve diagnostic data related to specific vehicles or assets, including fault codes, their classification and recommended actions.

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#getObjectDTCs
Request limits	10 requests / minute

Parameters

getObjectDTCs requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getObjectCanSignals:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Result

Result for getObjectDTCs:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
trouble_code_id	string	Unique identifier for the trouble code.
code	string	Vehicle trouble code.
odometer	long	Vehicle odometer.
trou- ble_code_counter	int	Trouble code counter.
vin	string	Vehicle Identification Number.
event_time	dateTime	Event time.
resolve_time	dateTime	Timestamp when the trouble was resolved.
driver_uid	string	Unique identifier of the driver associated with the event.
trouble_classifica- tion	string	Classification of the trouble code (e.g., info, warning, critical).
trouble_title	string	Title or short description of the trouble.
trouble_cause	string	Identified or suspected cause of the trouble.
trouble_effect	string	Effect or impact of the trouble on the vehicle or oper- ation.
trouble_recom- mendation	string	Recommended action or steps to resolve the trouble.
latitude	int	Latitude coordinate where the event occurred.
longitude	int	Longitude coordinate where the event occurred.
address_desc	string	Human-readable description of the location (e.g., street address or area).
address_uid	string	Unique identifier for the address or location.

Orders

About orders

Asynchronous processing

Many order actions involve sending messages to devices. These messages are processed and sent asynchronously. As a consequence, there's no guarantee that a message was already sent out when the API call returns.

There are a few scenarios where this needs to be considered. For example, if you delete an order from a device and immediately create a new order with the same order number.

Asynchronous processing applies to the following actions:

- cancelOrderExtern
- sendOrderExtern
- sendDestinationOrderExtern
- updateOrderExtern
- updateDestinationOrderExtern
- deleteOrderExtern
- clearOrdersExtern
- assignOrderExtern
- reassignOrderExtern

sendOrderExtern

Description

The `sendOrderExtern` operation allows you to send an order message to an object. The message is sent asynchronously and therefore a positive result of this operation does not indicate that the message was sent to the object successfully.

Technical details

SOAP endpoint address / function name	ordersService#sendOrder
Request limits	300 requests / 30 minutes

Note: On the Driver Terminal, the most recent order is shown at the top of the list of orders. If you tap the **New order** button in the Driving View, the oldest order is listed first.

Important: Orders without an order destination are not displayed in Webfleet. As soon as an order destination is added with the action `updateDestinationOrderExtern`, the order will also be visible in the Webfleet user interface.

Parameters

sendOrderExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to sendOrderExtern:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
orderid	string (20)	Account-unique order id, case-sensitive. Maximum capacity is 20 bytes, actual number of characters depends on the byte count consumed by the UTF-8 encoding of those characters. This parameter is required.
ordertext	string (1000)	This parameter is required.
useorderstates	string	Specifies the order states that appear in the workflow of the indicated order. You can specify multiple values for this parameter. Separate the values with a comma when using CSV interface. Valid values: • `202` - Arrived at destination • `203` - Work started • `204` - Work finished • `205` - Departed from destination • `null` - The states do not appear in the workflow, even if specified differently in the account settings. If you do not specify this parameter, the account-specific settings are used. This parameter is optional.

Parameter	Type	Description
orderautomations	int	Specifies the workflow step(s) when starting an order that can be automated. You can specify multiple values for this parameter. Separate the values with a comma when using CSV interface. *Example: orderautomations=1,2* Valid values: • `1` - accept the order • `2` - start the order • `3` - navigate to the order destination • `4` - skip displaying the route summary screen • `5` - delete the order after it has been finished • `6` - suppress the 'continue with next order' screen If you do not specify this parameter, the account-specific settings are used. This parameter is optional. Available on PRO devices with firmware version 10.533 or higher.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

sendDestinationOrderExtern

Description

The `sendDestinationOrderExtern` operation allows you to send an order message together with target coordinates for a navigation system connected to the in-vehicle unit. The message is sent asynchronously and therefore a positive result of this operation does not indicate that the message was sent to the object successfully.

Technical details

SOAP endpoint address / function name	ordersService#sendDestinationOrder
Request limits	300 requests / 30 minutes

Note: On the Driver Terminal, the most recent order is shown at the top of the list of orders. If you tap the **New order** button in the Driving View, the oldest order is listed first.

Use with ISO8601 date and time formats

If used in conjunction with `useISO8601=true`, you need to take care of specifying the time zone, because otherwise UTC will be assumed by definition. Therefore, you should always provide a time zone definition with `orderdate`, e.g. `2009-01-20T+01:00` and provide your local time part to `ordertime`.

Example for minimal set of parameters

Sends a destination order referring to an address that has been created before with insertAddressExtern. The address must be geocoded.

Code sample 4-2: Minimal set of parameters

CSV

```
https://csv.webfleet.com/extern?account=xxx&apikey=xxx&lang=de&action=send-
DestinationOrderExtern&objectno=001&orderid=S-1234.001AF&ordertext=Test&ad-
drnr=401
```

SOAP

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
 xmlns:ser="http://connect.webfleet.tomtomwork.com/services">
<soap:Header/>
 <soap:Body>
  <ser:sendDestinationOrder>
   <aParm>
    <accountName>xxx</accountName>
    <userName>xxx</userName>
    <password>xxx</password>
    <apiKey>xxx</apiKey>
   </aParm>
   <gParm>
    <locale>DE</locale>
    <timeZone>Europe_Berlin</timeZone>
   </gParm>
   <dstOrderToSend orderNo="S-1234.001AF" orderType="" scheduledCompletion-
DateAndTime="2018-09-21T00:00:00.000+02:00">
    <orderText>Test</orderText>
    <addrNoToUseAsDestination>401</addrNoToUseAsDestination>
   </dstOrderToSend>
   <object objectNo="001"/>
  </ser:sendDestinationOrder>
 </soap:Body>
</soap:Envelope>
```

Example for ISO-formatted order date

Sends a destination order with an scheduled arrival time of 4:30pm on June 5th 2009 in the CEST timezone (Central European Summer Time is 2 hours ahead of UTC). Make sure to quote the + as %2B, otherwise the timezone specification will not parse correctly and the response will contain an error message.

Code sample 4-3: ISO-formatted order date

CSV

```
https://csv.webfleet.com/extern?account=xxx&apikey=xxx&lang=de&action=send-
DestinationOrderExtern&objectno=001&orderid=S-1234.002AF&ordertext=Test&ad-
drnr=401&useISO8601=true&orderdate=2009-06-05T+02:00&ordertime=16:30:00
```

SOAP

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
 xmlns:ser="http://connect.webfleet.tomtomwork.com/services">
 <soap:Body>
  <ser:sendDestinationOrder>
   <aParm>
    <accountName>xxx</accountName>
    <userName>xxx</userName>
    <password>xxx</password>
    <apiKey>xxx</apiKey>
   </aParm>
   <gParm>
    <locale>DE</locale>
```

```
    <timeZone>Europe_Berlin</timeZone>
   </gParm>
   <dstOrderToSend orderNo="S-1234.002AF" scheduledCompletionDateAnd-
Time="2009-06-05T16:30:00">
    <orderText>Test</orderText>
    <addrNoToUseAsDestination>401</addrNoToUseAsDestination>
   </dstOrderToSend>
   <object objectNo="001"/>
  </ser:sendDestinationOrder>
 </soap:Body>
</soap:Envelope>
```

Example for destination with latitude and longitude

Sends a destination order; the destination is given by geographical coordinates (51° 21′55,2″ N - 12° 23′55,6″ E) and a location description is provided for the driver.

Code sample 4-4: Destination with latitude and longitude

CSV

```
https://csv.webfleet.com/extern?account=xxx&apikey=xxx&lang=de&ac-
tion=sendDestinationOrderExtern&objectno=001&orderid=S-1234.003AF&order-
text=Test&latitude=51365338&longitude=12398799&country=DE&zip=04129&c-
ity=Leipzig&street=Maximilianallee 4
```

SOAP

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
 xmlns:ser="http://connect.webfleet.tomtomwork.com/services">
 <soap:Body>
  <ser:sendDestinationOrder>
   <aParm>
    <accountName>xxx</accountName>
    <userName>xxx</userName>
    <password>xxx</password>
    <apiKey>xxx</apiKey>
   </aParm>
   <gParm>
    <locale>DE</locale>
    <timeZone>Europe_Berlin</timeZone>
   </gParm>
   <dstOrderToSend orderNo="S-1234.003AF" scheduledCompletionDateAnd-
Time="">
    <orderText>Test</orderText>
    <explicitDestination>
     <street>Maximilianallee 4</street>
     <postcode>04129</postcode>
     <city>Leipzig</city>
     <country>DE</country>
     <geoPosition latitude="51365338" longitude="12398799"/>

     <contact contactName=""/>
    </explicitDestination>
   </dstOrderToSend>
   <object objectNo="001"/>
  </ser:sendDestinationOrder>
 </soap:Body>
</soap:Envelope>
```

Parameters

sendDestinationOrderExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to sendDestinationOrderExtern:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
orderid	string (20)	Account-unique order id, case-sensitive. Maximum capacity is 20 bytes, actual number of characters depends on the byte count consumed by the UTF-8 encoding of those characters.
ordertext	string (1000)	This parameter is required.
ordertype	nonNegativeInteger	Type of order. Valid values: • `1` - service order • `2` - pickup order • `3` - delivery order If empty or not present, the default is service order. This parameter is optional.
orderdate	date	Scheduled date of fulfilment for this order. If empty or not present, the default is the current day (calculated using the time zone defined on the Webfleet account). This parameter is optional.
ordertime	time	Scheduled time of fulfilment for this order. If empty or not present, the default is 00:00. This parameter is optional.

Parameter	Type	Description
arrivaltolerance	nonNegativeInteger	If there is an active order, the Driver Terminal constantly monitors the ETA and sends notifications messages if the ETA changes significantly. Differences between ETA and `ordertime` are interpreted as follows: • show as standard if: $t_{eta} < t_{order}$ • show as warning if: $t_{order} < t_{eta} < t_{order}$+`arrivaltolerance` • show as alarm if: t_{order}+`arrivaltolerance` $< t_{eta}$ Unit of measurement is minutes. Valid values: • 0 • 15 • 30 • 60 • 120 Requires `ordertime` and `orderdate`. This parameter is optional.
notify_enabled	boolean	If set to `1` enables generation of an event (Events) as soon as the ETA is lower than `notify_leadtime`. If not set, the default will be 0 (no notification created). This parameter is optional.
notify_leadtime	nonNegativeInteger	Time in minutes before arrival at destination at which an event (Events) is generated if `notify_enabled` is set to `1`. If not set, the default will be 15min. This parameter is optional.
contact	string (50)	Name of a contact person for this order. This parameter is optional.
contacttel	string (20)	Phone number of the contact person for this order. Must conform to the following format: `\+?([\d /\(\)\-])+` (a sequence of numbers and `()/-` that may be proceeded by a `+`). This parameter is optional.
contactemail	string (50)	The order contact email address. This parameter is optional.

Parameter	Type	Description
addrnr	string (10)	Use an address identified by `addrnr` as returned by showAddressReportExtern. The address must be geocoded. Case-sensitive. **Note**: If `addrnr` is provided, other parameters describing the destination (`longitude, latitude, country, zip, city, street, contact, contectTel`) are ignored and replaced by the respective values from the address record identified by `addrnr`. This parameter is optional.
longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
country	string (2)	ISO 3166-1 alpha-2 code for the country (capital letters). Country of the destination address; see navigation hints below.
zip	string (10)	Post code of the destination address; see navigation hints below. **Please note:** A zip is mandatory, if you want to make use of sharing the ePOD (electronic proof of delivery) data via order status page.
city	string (50)	City of the destination address; see navigation hints below.
street	string (50)	Street and house number of the destination address; see navigation hints below.

Parameter	Type	Description
useorderstates	string	Specifies the order states that appear in the workflow of the indicated order. You can specify multiple values for this parameter. Separate the values with comma when using CSV interface. Valid values: `null` - Below states do not appear in workflow, even if specified differently in the account settings. For `ordertype` = `1`: • `202` - Arrived at destination • `203` - Work started • `204` - Work finished • `205` - Departed from destination • `207` - Electronic proof of delivery (ePOD) For `ordertype` = `2`: • `222` - Arrived at pick up location • `223` - Pick up started • `224` - Pick up finished • `225` - Departed from pick up location • `227` - Electronic proof of delivery (ePOD) For `ordertype` = `3`: • `242` - Arrived at delivery location • `243` - Delivery started • `244` - Delivery finished • `245` - Departed from delivery location • `247` - Electronic proof of delivery (ePOD) **Note**: If you do not specify the parameter, the account specific settings are used. This parameter is optional.

Parameter	Type	Description
orderautomations	int	Specifies the workflow step(s) when starting an order that can be automated. You can specify multiple values for this parameter. Separate the values with a comma when using CSV interface. *Example: orderautomations=1,2* Valid values: • `1` - accept the order • `2` - start the order • `3` - navigate to the order destination • `4` - skip displaying the route summary screen • `5` - delete the order after it has been finished • `6` - suppress the 'continue with next order' screen **Note**: If you do not specify the parameter, the account specific settings are used. This parameter is optional. Available on PRO devices with firmware version 10.533 or higher.

Parameter	Type	Description
wp	string	Contains details for a waypoint for orders with predefined routes (itinerary) in the following format. `<latitude>,<longitude>,[<description>],[<notify>],[<visible>]` For details of the parameters contained, see Parameters in wp. The number of waypoints per order is limited: • The following devices support up to **1000** waypoints per order: ° TomTom PRO 7100/7150 ° TomTom PRO 9100/9150 ° TomTom PRO 5150 • The following devices support up to **250** waypoints per order: ° PRO 8475 ° PRO 8375 ° TomTom PRO 8270/8275 ° PRO 7350 ° PRO 5350 • The following devices **do not** support waypoints: ° TomTom PRO 5250, TomTom PRO 7250 The number of waypoints/vehicle/month is limited to 20,000. If you exceed this limit this action will return with response code 25001. This parameter is optional.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
mapcode	string (20)	A mapcode represents a location. Either `mapcode` or `latitude/longitude` coordinates can be used, but not both. Learn more about mapcodes.

Parameter	Type	Description
epod_required	boolean	If set to `true`, the order requires to be signed after completion (ePOD - electronic proof of delivery). Note that ePOD is only supported on the Work App. Other driver terminals will ignore this step in the workflow.

Hints for navigation

The latitude and longitude parameters are mandatory, if `addrnr` is not used. Optional parameters `country`, `zip`, `city`, `street` have additional purposes:

On classical Driver Terminals not using Work App (like PRO 5350, 7350, 8475):

- If existing, the parameters making up the description are solely used as a label for the destination as displayed in the navigation app.

On Work App or driver terminals running the Work App (like PRO M, PRO X):

- If existing, the parameters making up the description are always used as a label for the destination as displayed in the navigation app.
- **By default**, the parameters are used in combination with the coordinates to improve the routing. This can improve routing and final approaches to destinations especially by properly considering the context like the street name and house number. It's suggested to use this in most circumstances.
- **Optionally**, this behavior can be changed, so that the system behaves as on classical Driver Terminals where the parameters are ignored for the actual route planning. This can be useful in situations where the destination is off the road or on company grounds. **Please contact customer support to change the behavior**.

Note: If `addrnr` is provided, other parameters describing the destination (`longitude, latitude, country, zip, city, street, contact, contacttel, contactemail`) are ignored and replaced by the respective values from the address record identified by `addrnr`.

updateOrderExtern

Description

Updates an order that was submitted with `sendOrderExtern`.

Technical details

SOAP endpoint address / function name	ordersService#updateOrder
Request limits	300 requests / 30 minutes

Important: Orders without an order destination are not displayed in Webfleet. As soon as an order destination is added with the action `updateDestinationOrderExtern`, the order will also be visible in the Webfleet user interface.

Parameters

updateOrderExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to updateOrderExtern:

Parameter	Type	Description
orderid	string (20)	Account-unique order id, case-sensitive. Maximum capacity is 20 bytes, actual number of characters depends on the byte count consumed by the UTF-8 encoding of those characters. This parameter is required.
ordertext	string (1000)	This parameter is required.
orderautomations	int	Specifies the workflow step(s) when starting an order that can be automated. You can specify multiple values for this parameter. Separate the values with a comma when using CSV interface. **Important**: Updating the order automation exclusively affects orders that are being sent to the device. *Example: orderautomations=1,2* Valid values: • `1` - accept the order • `2` - start the order • `3` - navigate to the order destination • `4` - skip displaying the route summary screen • `5` - delete the order after it has been finished • `6` - suppress the 'continue with next order' screen **Note**: If you do not specify this parameter, the account-specific settings are used. This parameter is optional. Available on PRO devices with firmware version 10.533 or higher.

updateDestinationOrderExtern

Description

Updates an order that was submitted with `sendDestinationOrderExtern`, with `insertDestinationOrderExtern` or with `sendOrderExtern`.

Technical details

SOAP endpoint address / function name	ordersService#updateDestinationOrder
Request limits	300 requests / 30 minutes

Parameters

updateDestinationOrderExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to updateDestinationOrderExtern:

Parameter	Type	Description
orderid	string (20)	Account-unique order id, case-sensitive. Maximum capacity is 20 bytes, actual number of characters depends on the byte count consumed by the UTF-8 encoding of those characters. This parameter is required.
ordertext	string (1000)	This parameter is optional.
orderdate	date	Scheduled date of fulfilment for this order. If empty or not present, the default is the current day (calculated using the time zone defined on the Webfleet account). This parameter is optional.
ordertime	time	Scheduled time of fulfilment for this order. If empty or not present, the default is 00:00. This parameter is optional.

Parameter	Type	Description
arrivaltolerance	nonNegativeInteger	If there is an active order, the Driver Terminal constantly monitors the ETA and sends notifications messages if the ETA changes significantly. Differences between ETA and `ordertime` are interpreted as follows: • show as standard if: $t_{eta} < t_{order}$ • show as warning if: $t_{order} < t_{eta} < t_{order}$+`arrivaltolerance` • show as alarm if: t_{order}+`arrivaltolerance` $< t_{eta}$ Unit of measurement is minutes. Valid values: • 0 • 15 • 30 • 60 • 120 Requires `ordertime` and `orderdate`. This parameter is optional.
contact	string (50)	Name of a contact person for this order. This parameter is optional.
contacttel	string (20)	Phone number of the contact person for this order. Must conform to the following format: `\+?([\d /\(\)\-])+` (a sequence of numbers and `()/-` that may be proceeded by a `+`). This parameter is optional.
contactemail	string (50)	The order contact email address. This parameter is optional.

Parameter	Type	Description
addrnr	string (10)	Use an address identified by `addrnr` as returned by showAddressReportExtern. The address must be geocoded. Case-sensitive. **Note**: If addrnr is provided, other parameters describing the destination (`longitude, latitude, country, zip, city, street, contact, contectTel`) are ignored and replaced by the respective values from the address record identified by `addrnr`. This parameter is optional.
longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
country	string (2)	ISO 3166-1 alpha-2 code for the country (capital letters). Country of the destination address; see navigation hints below.
zip	string (10)	Post code of the destination address; see navigation hints below.
city	string (50)	City of the destination address; see navigation hints below.
street	string (50)	Street and house number of the destination address; see navigation hints below.

Parameter	Type	Description
orderautomations	int	Specifies the workflow step(s) when starting an order that can be automated. You can specify multiple values for this parameter. Separate the values with a comma when using CSV interface. **Important**: Updating the order automation exclusively affects orders that are being sent to the device. *Example: orderautomations=1,2* Valid values: • `1` - accept the order • `2` - start the order • `3` - navigate to the order destination • `4` - skip displaying the route summary screen • `5` - delete the order after it has been finished • `6` - suppress the 'continue with next order' screen **Note**: If you do not specify this parameter, the account-specific settings are used. This parameter is optional. Available on PRO devices with firmware version 10.533 or higher.PRO 8475

Parameter	Type	Description
wp	string	Contains details for a waypoint for orders with predefined routes (itinerary) in the following format. This parameter is optional. `<latitude>,<longitude>,[<description>],[<notify>],[<visible>]` For details of the parameters contained, see Parameters in wp in sendDestinationOrderExtern. The number of waypoints per order is limited: • The following devices support up to **1000** waypoints per order: ° TomTom PRO 7100/7150 ° TomTom PRO 9100/9150 ° TomTom PRO 5150 • The following devices support up to **250** waypoints per order: ° PRO 8475 ° PRO 8375 ° TomTom PRO 8270/8275 ° PRO 7350 ° PRO 5350 • The following devices **do not** support waypoints: ° TomTom PRO 5250, TomTom PRO 7250 **Note**: The number of waypoints/vehicle/month is limited to 20,000.
mapcode	string (20)	A mapcode represents a location. Either `mapcode` or `latitude/longitude` coordinates can be used but not both. Learn more about mapcodes.

Hints for navigation

The latitude and longitude parameters are mandatory, if `addrnr` is not used. Optional parameters `country`, `zip`, `city`, `street` have additional purposes:

On classical Driver Terminals not using Work App (like PRO 5350, 7350, 8475):

- If existing, the parameters making up the description are solely used as a label for the destination as displayed in the navigation app.

On Work App or driver terminals running the Work App (like PRO M, PRO X):

- If existing, the parameters making up the description are always used as a label for the destination as displayed in the navigation app.
- **By default**, the parameters are used in combination with the coordinates to improve the routing. This can improve routing and final approaches to destinations especially by properly considering the context like the street name and house number. It's suggested to use this in most circumstances.
- **Optionally**, this behavior can be changed, so that the system behaves as on classical Driver Terminals where the parameters are ignored for the actual route planning. This can be useful in situations where the destination is off the road or on company grounds. **Please contact customer support to change the behavior**.

Note: If `addrnr` is provided, other parameters describing the destination (`longitude, latitude, country, zip, city, street, contact, contacttel, contactemail`) are ignored and replaced by the respective values from the address record identified by `addrnr`.

insertDestinationOrderExtern

Description

The `insertDestinationOrderExtern` operation allows you to transmit an order message to Webfleet. The message is not sent and must be manually dispatched to an object within Webfleet.

Technical details

SOAP endpoint address / function name	ordersService#insertDestinationOrder
Request limits	300 requests / 30 minutes

Parameters

insertDestinationOrderExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to insertDestinationOrderExtern:

Parameter	Type	Description
orderid	string (20)	Account-unique order id, case-sensitive. Maximum capacity is 20 bytes, actual number of characters depends on the byte count consumed by the UTF-8 encoding of those characters.
ordertext	string (1000)	This parameter is required.

Parameter	Type	Description
ordertype	nonNegativeInteger	Type of order. Valid values: • `1` - service order • `2` - pickup order • `3` - delivery order If empty or not present, the default is service order. This parameter is optional.
orderdate	date	Scheduled date of fulfilment for this order. If empty or not present, the default is the current day (calculated using the time zone defined on the Webfleet account). This parameter is optional.
ordertime	time	Scheduled time of fulfilment for this order. If empty or not present, the default is 00:00. This parameter is optional.
arrivaltolerance	nonNegativeInteger	If there is an active order, the Driver Terminal constantly monitors the ETA and sends notifications messages if the ETA changes significantly. Differences between ETA and `ordertime` are interpreted as follows: • show as standard if: $t_{eta} < t_{order}$ • show as warning if: $t_{order} < t_{eta} < t_{order}$+`arrivaltolerance` • show as alarm if: t_{order}+`arrivaltolerance` $< t_{eta}$ Unit of measurement is minutes. Valid values: • 0 • 15 • 30 • 60 • 120 Requires `ordertime` and orderdate. This parameter is optional.
notify_enabled	boolean	If set to `1` enables generation of an event (Events) as soon as the ETA is lower than `notify_leadtime`. If not set, the default will be 0 (no notification created). This parameter is optional.

Parameter	Type	Description
notify_leadtime	nonNegativeInteger	Time in minutes before arrival at destination at which an event (Events) is generated if `notify_enabled` is set to `1`. If not set, the default will be 15min. This parameter is optional.
contact	string (50)	Name of a contact person for this order. This parameter is optional.
contacttel	string (20)	Phone number of the contact person for this order. Must conform to the following format: `\+?([\d /\(\)\-])+` (a sequence of numbers and `()/-` that may be proceeded by a `+`). This parameter is optional.
contactemail	string (50)	The order contact email address. This parameter is optional.
addrnr	string (10)	Use an address identified by `addrnr` as returned by showAddressReportExtern. The address must be geocoded. Case-sensitive. **Note**: If `addrnr` is provided, other parameters describing the destination (`longitude, latitude, country, zip, city, street, contact, contectTel`) are ignored and replaced by the respective values from the address record identified by `addrnr`. This parameter is optional.
longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.

Parameter	Type	Description
latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
country	string (2)	ISO 3166-1 alpha-2 code for the country (capital letters). Country of the destination address; see navigation hints below.
zip	string (10)	Post code of the destination address; see navigation hints below.
city	string (50)	City of the destination address; see navigation hints below.
street	string (50)	Street and house number of the destination address; see navigation hints below.

Parameter	Type	Description
wp	string	Contains details for a waypoint for orders with predefined routes (itinerary) in the following format. This parameter is optional. `<latitude>,<longitude>,[<description>],[<notify>],[<visible>]` For details of the parameters contained, see Parameters in wp in sendDestinationOrderExtern. The number of waypoints per order is limited: • The following devices support up to **1000** waypoints per order: ° TomTom PRO 7100/7150 ° TomTom PRO 9100/9150 ° TomTom PRO 5150 • The following devices support up to **250** waypoints per order: ° PRO 8475 ° PRO 8375 ° TomTom PRO 8270/8275 ° PRO 7350 ° PRO 5350 • The following devices **do not** support waypoints: ° TomTom PRO 5250, TomTom PRO 7250 **Note**: The number of waypoints/vehicle/month is limited to 20,000.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
mapcode	string (20)	A mapcode represents a location. Either `mapcode` or `latitude/longitude` coordinates can be used but not both. Learn more about mapcodes.

Hints for navigation

The latitude and longitude parameters are mandatory, if `addrnr` is not used. Optional parameters `country`, `zip`, `city`, `street` have additional purposes:

On classical Driver Terminals not using Work App (like PRO 5350, 7350, 8475):

- If existing, the parameters making up the description are solely used as a label for the destination as displayed in the navigation app.

On Work App or driver terminals running the Work App (like PRO M, PRO X):

- If existing, the parameters making up the description are always used as a label for the destination as displayed in the navigation app.
- **By default**, the parameters are used in combination with the coordinates to improve the routing. This can improve routing and final approaches to destinations especially by properly considering the context like the street name and house number. It's suggested to use this in most circumstances.
- **Optionally**, this behavior can be changed, so that the system behaves as on classical Driver Terminals where the parameters are ignored for the actual route planning. This can be useful in situations where the destination is off the road or on company grounds. **Please contact customer support to change the behavior**.

Note: If `addrnr` is provided, other parameters describing the destination (`longitude, latitude, country, zip, city, street, contact, contacttel, contactemail`) are ignored and replaced by the respective values from the address record identified by `addrnr`.

cancelOrderExtern

Description

Cancels orders that were submitted using one of `sendDestinationOrderExtern`, `insertDestinationOrderExtern` or `sendOrderExtern`.

Technical details

SOAP endpoint address / function name	ordersService#cancelOrder
Request limits	300 requests / 30 minutes

Parameters

cancelOrderExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to cancelOrderExtern:

Parameter	Type	Description
orderid	string (20)	Account-unique order id, case-sensitive. Maximum capacity is 20 bytes, actual number of characters depends on the byte count consumed by the UTF-8 encoding of those characters.

assignOrderExtern

Description

Assigns an existing order to an object and can be used to accomplish the following:

- send an order that was inserted before using `insertDestinationOrderExtern`
- resend an order that has been rejected or cancelled

Technical details

SOAP endpoint address / function name	ordersService#assignOrder
Request limits	300 requests / 30 minutes

Parameters

assignOrderExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to assignOrderExtern:

Parameter	Type	Description
orderid	string (20)	Account-unique order id, case-sensitive. Maximum capacity is 20 bytes, actual number of characters depends on the byte count consumed by the UTF-8 encoding of those characters. This parameter is required.
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. This `objectno` names the object that will receive the order. Required if the order is not linked to an object yet. This parameter is optional.

Parameter	Type	Description
orderautomations	int	Specifies the workflow step(s) when starting an order that can be automated. You can specify multiple values for this parameter. Separate the values with a comma when using CSV interface. *Example: orderautomations=1,2* Valid values: • `1` - accept the order • `2` - start the order • `3` - navigate to the order destination • `4` - skip displaying the route summary screen • `5` - delete the order after it has been finished • `6` - suppress the 'continue with next order' screen **Note**: If you do not specify this parameter, the account-specific settings are used. This parameter is optional. Available on PRO devices with firmware version 10.533 or higher.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

reassignOrderExtern

Description

Reassigns an order that was submitted using one of sendDestinationOrderExtern, insertDestinationOrderExtern or sendOrderExtern to another object. This is done by cancelling the order on the old object that is currently assigned to this order and assigning the new object to the order. The order is then sent to the new object.

Technical details

SOAP endpoint address / function name	ordersService#reassignOrder
Request limits	300 requests / 30 minutes

Parameters

reassignOrderExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to reassignOrderExtern:

Parameter	Type	Description
orderid	string (20)	Account-unique order id, case-sensitive. Maximum capacity is 20 bytes, actual number of characters depends on the byte count consumed by the UTF-8 encoding of those characters. This parameter is required.
objectid	string (10)	Identifying number of an object. Unique within an account, case-sensitive. This `objectid` (which is equal to `objectno`) names the object that will receive the order. This parameter is required.
orderautomations	int	Specifies the workflow step(s) when starting an order that can be automated. You can specify multiple values for this parameter. Separate the values with a comma when using CSV interface. *Example: orderautomations=1,2* Valid values: • `1` - accept the order • `2` - start the order • `3` - navigate to the order destination • `4` - skip displaying the route summary screen • `5` - delete the order after it has been finished • `6` - suppress the 'continue with next order' screen If you do not specify this parameter, the account-specific settings are used. This parameter is optional. Available on PRO devices with firmware version 10.533 or higher.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectid`.

deleteOrderExtern

Description

Deletes an order from a device and optionally marks it as deleted in Webfleet too.

Supported for the stand-alone Driver Terminals connected to Webfleet and the Driver Terminals connected to LINK 7xx/5xx/4xx/3xx.

Technical details

SOAP endpoint address / function name	ordersService#deleteOrder
Request limits	300 requests / 30 minutes

Parameters

deleteOrderExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to deleteOrderExtern:

Parameter	Type	Description
orderid	string (20)	Account-unique order id, case-sensitive. Maximum capacity is 20 bytes, actual number of characters depends on the byte count consumed by the UTF-8 encoding of those characters.
mark_deleted	boolean	If set to `1`, the order is deleted in Webfleet and the order ID can be used again. Defaults to `0`.

clearOrdersExtern

Description

Removes all orders from the device and optionally deletes them all in Webfleet too.

Technical details

SOAP endpoint address / function name	ordersService#clearOrders
Request limits	300 requests / 30 minutes

Parameters

clearOrdersExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to clearOrdersExtern:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case sensitive.

Parameter	Type	Description
mark_deleted	boolean	If set to `1`, the orders are deleted in Webfleet too and the order IDs can be used again. Defaults to `0`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

showOrderReportExtern

Description

Shows a list of orders that match the search parameters. Each entry shows the order details and current status information.

Technical details

SOAP endpoint address / function name	ordersService#showOrderReport
Request limits	6 requests / minute

Example - Retrieve all orders for the previous week

Uses a `range_pattern` to access all orders with an `orderdate` within the last week.

Code sample 4-5: Retrieve all orders for the previous week

CSV

```
https://csv.webfleet.com/extern?account=xxx&apikey=xxx&lang=de&ac-
tion=showOrderReportExtern&range_pattern=w-1
```

SOAP

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
 xmlns:ser="http://connect.webfleet.tomtomwork.com/services">
<soap:Header/>
 <soap:Body>
  <ser:showOrderReport>
   <aParm>
    <accountName>xxx</accountName>
    <userName>xxx</userName>
    <password>xxx</password>
    <apiKey>xxx</apiKey>
   </aParm>
   <gParm>
    <locale>DE</locale>
    <timeZone>Europe_Berlin</timeZone>
   </gParm>
   <queryFilter>
    <object objectNo="" objectUid="" externalId=""/>
    <dateRange rangePattern="Wm1"></dateRange>
   </queryFilter>
  </ser:showOrderReport>
 </soap:Body>
</soap:Envelope>
```

Example - Retrieve orders for a specific object and day

Uses a user-defined (`ud`) `range_pattern` with ISO-formatting to access all orders with a specific `orderdate` assigned to a specific object.

Code sample 4-6: Retrieve orders for a specific object and day

CSV

```
https://csv.webfleet.com/extern?account=xxx&apikey=xxx&lang=de&ac-
tion=showOrderReportExtern&objectno=001&useISO8601=true&range_pat-
tern=ud&rangefrom_string=2009-06-11T00:00:00&range-
to_string=2009-06-11T23:59:59
```

SOAP

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
 xmlns:ser="http://connect.webfleet.tomtomwork.com/services">
<soap:Header/>
 <soap:Body>
  <ser:showOrderReport>
   <aParm>
<accountName>xxx</accountName>
<userName>xxx</userName>
<password>xxx</password>
<apiKey>xxx</apiKey>
   </aParm>
   <gParm>
<locale>DE</locale>
<timeZone>Europe_Berlin</timeZone>
   </gParm>
   <queryFilter>
<object objectNo="001"/>
<dateRange rangePattern="UD">
<from>2009-06-11T00:00:00.000+01:00</from>
<to>2009-06-11T23:59:59.000+01:00</to>
</dateRange>
   </queryFilter>
  </ser:showOrderReport>
 </soap:Body>
</soap:Envelope>
```

Parameters

showOrderReportExtern requires the following common parameters:

- Authentication parameters
- General parameters

The following other parameters are required if `orderid` is not indicated:

- Date range filter parameters

Parameters specific to showOrderReportExtern:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. This parameter is optional.

Parameter	Type	Description
objectgroupname	string	Case-sensitive. This parameter is optional.
orderid	string (20)	Account-unique order id, case-sensitive. Maximum capacity is 20 bytes, actual number of characters depends on the byte count consumed by the UTF-8 encoding of those characters. This parameter is optional.
ordertype	nonNegativeInteger	Type of order. Valid values: • `1` - service order • `2` - pickup order • `3` - delivery order This parameter is optional.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Result

Result for showOrderReportExtern:

Parameter	Type	Description
orderdate	date	Scheduled date of fulfilment for this order. If empty or not present, the default is the current day (calculated using the time zone defined on the Webfleet account).
orderid	string (20)	Account-unique order id, case-sensitive. Maximum capacity is 20 bytes, actual number of characters depends on the byte count consumed by the UTF-8 encoding of those characters.
ordertext	string (1000)	
ordertype	nonNegativeInteger	Type of order. Valid values: • `1` - service order • `2` - pickup order • `3` - delivery order
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectname	string	Display name of an object.

Parameter	Type	Description
longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
destination	string	
orderstate	nonNegativeInteger	Valid values: • `0` - Not yet sent • `100` - Sent • `101` - Received • `102` - Read • `103` - Accepted • `201` - Service order started • `202` - Arrived at destination • `203` - Work started • `204` - Work finished • `205` - Departed from destination • `207` - Proof of delivery (service order type) • `221` - Pickup order started • `222` - Arrived at pick up location • `223` - Pick up started • `224` - Pick up finished • `225` - Departed from pick up location • `227` - Proof of delivery (pick up oder type) • `241` - Delivery order started • `242` - Arrived at delivery location • `243` - Delivery started • `244` - Delivery finished • `245` - Departed from delivery location • `247` - Proof of delivery (delivery order type) • `298` - Resumed • `299` - Suspended • `301` - Cancelled • `302` - Rejected • `401` - Finished

Parameter	Type	Description
orderstate_time	string	
orderstate_longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
orderstate_latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
orderstate_postext	string	
orderstate_msgtext	string	String containing the message text of the last order related message.
planned_arrival_time	time	Scheduled time of fulfillment for this order. If empty or not present, the default is 00:00.
estimated_arrival_time	time	Scheduled time of fulfillment for this order. If empty or not present, the default is 00:00.
arrivaltolerance	time	
notify_enabled	boolean	
notify_leadtime	nonNegativeInteger	
delay_warnings	nonNegativeInteger	Warning state based on planned and estimated arrival time and the arrival time tolerance value. Possible values are: • `1` - Estimated delay within tolerance. • `2` - Estimated delay above tolerance.
contact	string	Name of the contact person.
contacttel	string (20)	Phone number of the contact person for this order.
contactemail	string (50)	The order contact email address.

Parameter	Type	Description
driverno	string	Identifying number of a driver. Unique within an account. See Drivers to learn more about drivers. Assigned to the driver who reported working on the order most recently.
drivername	string	Display name of a driver. Assigned to the driver who reported working on the order most recently.
drivertelmobile	string	Mobile phone number of the driver. Associated to the order. Assigned to the driver who reported working on the order most recently.
waypointcount	nonNegativeInteger	The number of waypoints in an itinerary order.
addrnr	string (10)	Identifying number of an address. Unique within an account. Number of the destination address. Only returned when the order destination is identified by the address number.
country	string (2)	ISO 3166-1 alpha-2 code for the country (capital letters). Country of the destination address.
zip	string (10)	Post code of the destination address.
city	string (250)	City of the destination address.
street	string (250)	Street and house number of the destination address.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.
mapcode	string (20)	A mapcode represents a location. Learn more about mapcodes.

showOrderWaypoints

Description

This action retrieves the waypoints for an itinerary order with additional information on the validity and state. The waypoints are sorted in the same order which was used when creating the itinerary.

Itinerary orders (predefined routes over the air) are supported on all Driver Terminals with software version 10.537 or higher.

Technical details

SOAP endpoint address / function name	ordersService#showOrderWaypoints
Request limits	10 requests / minute

Parameters

showOrderWaypoints requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to showOrderWaypoints:

Parameter	Type	Description
orderid	string (20)	Account-unique order id, case-sensitive. Maximum capacity is 20 bytes, actual number of characters depends on the byte count consumed by the UTF-8 encoding of those characters. This parameter is mandatory.

Result

Result for showOrderWaypoints:

Parameter	Type	Description
waypointnumber	string	Automatically generated ordering number.
latitude	int	
longitude	int	
formatted_latitude	string	Human readable formatted latitude. Geographic latitude in the form GGG° MM′ SS.S″ N/S in the WGS84 coordinate system.
formatted_longitude	string	Human readable formatted longitude Geographic longitude in the form GGG° MM′ SS.S″ E/W in the WGS84 coordinate system.
description	string (20)	Short waypoint description.
notify	Boolean	Indicates whether the driver gets a notification when the waypoint is passed. Default is `1`. Valid values: • `0` - false • `1` - true (driver will be notified when passing this waypoint)

Parameter	Type	Description
visible	Boolean	Indicates whether the waypoint is visible on the navigation device. Default is `1`. Valid values: • `0` - false • `1` - true (waypoint is visible in the list of waypoints on the navigation device)
state	int	Indicates whether the waypoint was valid for a route calculation on the navigation device. Valid values: • `0` - normal • `1` - rejected

Parameters in wp

Parameter	Type	Description
latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia. This parameter is required.
longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia. This parameter is required.
description	string (20)	Short waypoint description. This parameter is optional.

Parameter	Type	Description
notify	Boolean	**Note**: This parameter is not supported any more on newer devices. The use is discouraged. Indicates whether the driver gets a notification when the waypoint is passed. Default is `1`. Valid values: • `0` - false • `1` - true (driver will be notified when passing the waypoint) This parameter is optional.
visible	Boolean	**Note**: This parameter is not supported any more on newer devices. The use is discouraged. Indicates whether the waypoint is visible on the navigation device. Default is `1`. Valid values: • `0` - false • `1` - true (waypoint is visible in the list of waypoints on the navigation device) This parameter is optional.

getOrderStatusPageURL

Description

`getOrderStatusPageURL` returns a link which can be used to track the status of the specified order. The returned link is a public URL and can be used by everyone who knows the URL.

Calling `getOrderStatusPageURL` multiple times for the same order will always return the same existing link, except if you invalidate an existing link using `removeOrderStatusPageURL`.

Please note that the status of the order, including its planned time of arrival and estimated time of arrival, is an approximation. Information relating to the status of the order relies on factors including, but not limited to the timely update of progress by the dispatcher and driver, stops such as for mandatory breaks or refueling, or route deviations.

Technical details

SOAP endpoint address / function name	ordersService#getOrderStatusPageURL
Request limits	300 requests / 30 minutes

Parameters

getOrderStatusPageURL requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to `getOrderStatusPageURL`:

Parameter	Type	Description
orderno	string (20)	Account-unique order number, case-sensitive. This parameter is mandatory.

Result

Result for `getOrderStatusPageURL`:

Parameter	Type	Description
orderno	string (20)	Account-unique order number, case-sensitive.
statuspageurl	string	A URL for order status page.

removeOrderStatusPageURL

Description

`removeOrderStatusPageURL` invalidates the link created by `getOrderStatusPageURL`.

Usually you don't need to remove order status links, as they expire automatically some days after order completion. Using `removeOrderStatusPageURL` you can invalidate a link before its automatic expiration, for example if you gave the link to someone by accident. Calling `getOrderStatusPageURL` after invalidation will create a new link with a different URL.

Technical details

SOAP endpoint address / function name	ordersService#removeOrderStatusPageURL
Request limits	300 requests / 30 minutes

Parameters

removeOrderStatusPageURL requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to `removeOrderStatusPageURL`:

Parameter	Type	Description
orderno	string (20)	Account-unique order number, case-sensitive. This parameter is mandatory.

downloadOrderAttachment

Description

`downloadOrderAttachment` fetches a specific order attachment.

Technical details

SOAP endpoint address / function name	ordersService#downloadOrderAttachment
Request limits	10 requests / minute

Parameters

downloadOrderAttachment requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to `downloadOrderAttachment`:

Parameter	Type	Description
orderno	string (20)	Account-unique order number, case-sensitive. This parameter is mandatory.
attachment_id	string	A unique, unchangeable identifier for the order attachment. Required.

Result

The file content is returned in the response body. The `Content-Type` HTTP header contains the MIME type.

The SOAP API returns a data structure containing these values.

downloadOrderAttachmentThumbnail

Description

`downloadOrderAttachmentThumbnail` fetches a thumbnail of a specific order attachment.

Technical details

SOAP endpoint address / function name	ordersService#downloadOrderAttachmentThumbnail

Request limits	10 requests / minute

Parameters

downloadOrderAttachmentThumbnail requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to `downloadOrderAttachmentThumbnail`:

Parameter	Type	Description
orderno	string (20)	Account-unique order number, case-sensitive. This parameter is mandatory.
attachment_id	string	A unique, unchangeable identifier for the order attachment. Required.

Result

The thumbnail content is returned in the response body. The `Content-Type` HTTP header contains the MIME type.

The SOAP API returns a data structure containing these values.

uploadOrderAttachment

Description

`uploadOrderAttachment` uploads an order attachment, such as a picture or document, for the given order number.

This action requires to use **HTTP POST** due to the size limitation of URLs. See Making HTTP requests

A maximum of 10 attachments can be uploaded for each order. 5 of them via WEBFLEET.connect or the Webfleet user interface. 5 more attachments can be uploaded by the driver using the Webfleet Work App.

The maximum file size of each attachment is 5 MB.

Technical details

SOAP endpoint address / function name	ordersService#uploadOrderAttachment
Request limits	10 requests/minute

Parameters

uploadOrderAttachment requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to `uploadOrderAttachment`:

Parameter	Type	Description
orderno	string (20)	Account-unique order number, case-sensitive. This parameter is mandatory.
file_name	string	The file name of the attachment. Required, must have at least one character and extension e.g. 'test.txt'. The following file types may be uploaded: .jpg, .p-ng, .bmp, .pdf, .txt, .rtf, .doc, .docx, .gif, .heic and .heif.
file_content	string	Order attachment as base64 encoded string. Re-quired. Maximum allowed file size is 5 MB.

Result

Result for uploadOrderAttachment:

Parameter	Type	Description
attachment_id	string	A unique, unchangeable identifier for the created or-der attachment.
orderno	string (20)	Account-unique order number, case-sensitive.

getOrderAttachmentsMetadata

Description

`getOrderAttachmentsMetadata` provides information of all the order attachments of the given order number.

Technical details

SOAP endpoint address / func-tion name	ordersService#getOrderAttachmentsMetadata
Request limits	10 requests / minute

Parameters

getOrderAttachmentsMetadata requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to `getOrderAttachmentsMetadata`:

Parameter	Type	Description
orderno	string (20)	Account-unique order number, case-sensitive. This parameter is mandatory.

Result

Result for getOrderAttachmentsMetadata:

Parameter	Type	Description
attachment_id	string	A unique, unchangeable identifier for the order attachment.
orderno	string (20)	Account-unique order number, case-sensitive.
file_name	string	File name of the attachment.
file_size_in_bytes	int	File size of the order attachment in bytes.
file_mime_type	string	MIME type of the stored file.
uploader_user_uid	string	ID of the user who uploaded the attachment for the order.
has_thumbnail	boolean	`true` if a thumbnail for the attachment can be requested.
uploaded_by_driver	boolean	`true` if the upload was done by a driver.
creation_time	dateTime	The date and time the order attachment was uploaded.
expiration_time	dateTime	The date and time the order attachment will be deleted due to an expired retention duration.

deleteOrderAttachment

Description

`deleteOrderAttachment` deletes a specific order attachment.

Technical details

SOAP endpoint address / function name	ordersService#deleteOrderAttachment
Request limits	10 requests / minute

Parameters

deleteOrderAttachment requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to `deleteOrderAttachment`:

Parameter	Type	Description
orderno	string (20)	Account-unique order number, case-sensitive. This parameter is mandatory.
attachment_id	string	A unique, unchangeable identifier for order attachment. Required.

getOrderEpod

Description

`getOrderEpod` returns the ePOD (electronic proof of delivery) metadata of an order. An ePOD for an order is available only if the ePOD workflow state was included in the workflow states when the order was sent. (e.g. ServiceEpod 207, PickUpEpod 227, DeliveryEpod 247)

Technical details

SOAP endpoint address / function name	ordersService#getOrderEpod
Request limits	10 requests / minute

Parameters

getOrderEpod requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to `getOrderEpod`:

Parameter	Type	Description
orderno	string (20)	Account-unique order number, case-sensitive. This parameter is mandatory.

Result

Result for getOrderEpod:

Parameter	Type	Description
state	string	State of the electronic proof of delivery (ePOD). • PENDING • REJECTED • COMPLETED • SKIPPED This property is always present in response.
state_change_time	dateTime	The date and time, when the ePOD state has changed. This property is always present in response.
signatory	string (50)	The name of the person who should sign the receipt, pre-filled with the contact name from the order. Maximum length is 50 characters. This property is optional and can be empty.
comment	string	A free-text comment regarding the delivery (e.g., quality of service, damage to the package), provided by the driver or signatory. Maximum length is 4000 characters. This property is optional and can be empty.
rating	string	A rating allowing clients to give feedback on their satisfaction of the delivery or service. If there was no rating the `NO_RATING` value is used. • NO_RATING • VERY_DISSATISFIED • DISSATISFIED • NEUTRAL • SATISFIED • VERY_SATISFIED This property is always present in response.
photo_uids	string	JSON encoded string with an array of strings. Each string contains the UIDs of photos related to the ePOD. Example: ‘[’‘1-6712264’‘, ’‘1-28EDD4D’‘]’. This property is optional and can be empty.

downloadOrderEpodSignature

Description

`downloadOrderEpodSignature` fetches a signature (JPEG) attached to an ePOD.

Technical details

SOAP endpoint address / function name	ordersService#downloadOrderEpodSignature
Request limits	10 requests / minute

Parameters

downloadOrderEpodSignature requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to `downloadOrderEpodSignature`:

Parameter	Type	Description
orderno	string (20)	Account-unique order number, case-sensitive. This parameter is mandatory.

Result

The file content is returned in the response body. The Content-Type HTTP header contains the MIME type 'application/octet-stream'. The Content-Disposition HTTP header contains the type 'attachment' and the file name, which is currently always 'signature.jpg'.

The SOAP API returns a data structure containing these values.

downloadOrderEpodPhoto

Description

`downloadOrderEpodPhoto` fetches a photo (JPEG) attached to an ePOD.

Technical details

SOAP endpoint address / function name	ordersService#downloadOrderEpodPhoto
Request limits	10 requests / minute

Parameters

downloadOrderEpodPhoto requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to `downloadOrderEpodPhoto`:

Parameter	Type	Description
orderno	string (20)	Account-unique order number, case-sensitive. This parameter is mandatory.
photo_uid	string	A required, unique, identifier for a photo.

Result

The file content is returned in the response body. The Content-Type HTTP header contains the MIME type 'application/octet-stream'. The Content-Disposition HTTP header contains the type 'attachment' and the file name, which is currently always 'epod_photo.jpg'.

The SOAP API returns a data structure containing these values.

createOrderLoadedGoods

Description

`createOrderLoadedGoods` creates a loaded goods entry associated with a specific order.

Loaded goods information is available for drivers when using the Work App.

Note that loaded goods might be created and updatedf only if the order is not yet sent.

Technical details

SOAP endpoint address / function name	ordersService#createOrderLoadedGoods
Request limits	10 requests / minute

Parameters

createOrderLoadedGoods requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to `createOrderLoadedGoods`:

Parameter	Type	Description
orderno	string (20)	Account-unique order number, case-sensitive. This parameter is mandatory.
productId	string (50)	User defined identifier. Unique per order. This parameter is mandatory.
contents	string (50)	Description. This parameter is mandatory.
loadingMedium	string (50)	Name of the loading medium. This parameter is mandatory.

Parameter	Type	Description
quantity	int	Number of mediums. This parameter is mandatory.
weightInGrams	int	Weight of all mediums. This parameter is mandatory.

Result

Result for createOrderLoadedGoods:

Parameter	Type	Description
productId	string (50)	User defined identifier.
contents	string (50)	Description.
loadingMedium	string (50)	Name of the loading medium.
quantity	int	Number of mediums.
weightInGrams	int	Weight of all mediums.

updateOrderLoadedGoods

Description

`updateOrderLoadedGoods` updates an existing loaded goods entry associated with a specific order.

Loaded goods might be created and updated only if the order is not yet sent.

Technical details

SOAP endpoint address / function name	ordersService#updateOrderLoadedGoods
Request limits	10 requests / minute

Parameters

updateOrderLoadedGoods requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to `updateOrderLoadedGoods`:

Parameter	Type	Description
orderno	string (20)	Account-unique order number, case-sensitive. This parameter is mandatory.

Parameter	Type	Description
productId	string (50)	User defined identifier. Unique per order. This parameter is mandatory.
contents	string (50)	Description. This parameter is mandatory.
loadingMedium	string (50)	Name of the loading medium. This parameter is mandatory.
quantity	int	Number of mediums. This parameter is mandatory.
weightInGrams	int	Weight of all mediums. This parameter is mandatory.

Result

Result for updateOrderLoadedGoods:

Parameter	Type	Description
productId	string (50)	User defined identifier.
contents	string (50)	Description.
loadingMedium	string (50)	Name of the loading medium.
quantity	int	Number of mediums.
weightInGrams	int	Weight of all mediums.

deleteOrderLoadedGoods

Description

`deleteOrderLoadedGoods` deleted an existing loaded goods entry associated with a specific order.

Loaded goods might be created and updated only if the order is not yet sent.

Technical details

SOAP endpoint address / function name	ordersService#deleteOrderLoadedGoods
Request limits	10 requests / minute

Parameters

deleteOrderLoadedGoods requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to `deleteOrderLoadedGoods`:

Parameter	Type	Description
orderno	string (20)	Account-unique order number, case-sensitive. This parameter is mandatory.
productId	string (50)	User defined identifier. Unique per order. This parameter is mandatory.

getOrderLoadedGoods

Description

`getOrderLoadedGoods` retrieves existing loaded goods entries associated with a specific order.

Technical details

SOAP endpoint address / function name	ordersService#getOrderLoadedGoods
Request limits	10 requests / minute

Parameters

getOrderLoadedGoods requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to `getOrderLoadedGoods`:

Parameter	Type	Description
orderno	string (20)	Account-unique order number, case-sensitive. This parameter is mandatory.

Result

Result for getOrderLoadedGoods:

Parameter	Type	Description
productId	string (50)	User defined identifier.
contents	string (50)	Description.
loadingMedium	string (50)	Name of the loading medium.
quantity	int	Number of mediums.

Parameter	Type	Description
weightInGrams	int	Weight of all mediums.

Messages

sendTextMessageExtern

Description

The `sendTextMessageExtern` operation allows you to send a text message to an object.The message is sent asynchronously and therefore a positive result of this operation does not indicate that the message was sent to the object successfully.

Technical details

SOAP endpoint address / function name	messagesService#sendTextMessage
Request limits	300 requests / 30 minutes

Parameters

sendTextMessageExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to sendTextMessageExtern:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
messagetext	string (500)	
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

clearTextMessagesExtern

Description

Removes all text messages from the device.

Technical details

SOAP endpoint address / function name	messagesService#clearTextMessages
Request limits	300 requests / 30 minutes

Parameters

clearTextMessagesExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to clearTextMessagesExtern:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

showMessages

Description

Using `showMessages` you can retrieve exclusively text, order, or text status messages without using the queue service. The maximum result size is limited to 500 entries. To get an additional batch of messages the date range parameter has to be changed.

Technical details

SOAP endpoint address / function name	messagesService#showMessages
Request limits	10 requests / minute

Parameters

showMessages requires the following common parameters:

- Authentication parameters
- General parameters
- Date range filter parameters (lower limit is 14x24h back)

Parameters specific to showMessages:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. This parameter is optional

Parameter	Type	Description
category	int	Valid values: • `1` - text messages • `2` - order status messages • `3` - text status messages This parameter is optional.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`. This parameter is optional.

Result

Result for showMessages:

Parameter	Type	Description
msgid	int	Identifies a message and is unique within the system. Possible values are in the range 0 # msgid # 2^{64}-1
msg_time	dateTime	ISO 8601-formatted date and time in the UTC time-zone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00
msg_flag	string	• Valid values: • `S` = sent • `R` = received
msg_category	int	• Valid values: • `1` - text messages • `2` - order status messages • `3` - text status messages
msg_text	string	The text of the message.
msg_refid	int	The message ID reference to the original message (if available).
msg_reftype	int	• Valid values: • `1` - received • `2` - read • `3` - discarded Only in use if `msg_refid` exists, otherwise empty.
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.

Parameter	Type	Description
pos_time	dateTime	ISO 8601-formatted date and time in the UTC time-zone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00
pos_latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
pos_longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
pos_text	string	A detailed textual description of the location.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

sendBinaryMessage

Description

`sendBinaryMessage` sends a maximum of 2560 byte raw data to a the driver terminal. The base64 encoding required for transmission of binary messages through the web service API does not count as raw payload data size. Be aware that there can only be one pending binary message for a driver terminal at a time.

Further calls of `sendBinaryMessage` for the same driver terminal will be rejected until the data is transferred to the driver terminal.

To track the status of the binary message, please use the WEBFLEET.connect Queue Service.

Note: To run sendBinaryMessage, an API key and the LINK.connect feature is required.

This action is supported on PRO 83xx/TomTom PRO 82xx devices only.

To use this action, you need to have the additional service LINK.connect booked. The data volume caused through `sendBinaryMessage` should be covered by the data volume of the selected LINK.connect tariff.

Technical details

SOAP endpoint address / function name	messagesService#sendBinaryMessages
Request limits	10 requests / minute

Parameters

sendBinaryMessages requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to sendBinaryMessage:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
appid	string (128)	An identifier to mark the application on the destination device. **Note**: Use package style for example com.vendor.application.part This parameter is required.
correlationid	int	An arbitrary identifier that helps to identify Webfleet messages in the message queue service that belong to this request. Should be unique in the integration server backend. This paramter is optional.
data	base64 (raw (2560))	Base64 encoded user payload, max. 2.5 KB. The data is not interpreted by Webfleet. This parameter is required.

resetBinaryMessages

Description

If the driver terminal does not respond to outgoing data sent using sendBinaryMessage the transfer status is incomplete. `resetBinaryMessages` resets the outgoing data transfer and gets back to a consistent state between integration server backend and device. This is important because you cannot send new data until the current transfer is completed or cancelled/reset.

If the binary data to be delivered with the original send is still important from the view of the PRO.connect integration solution, the integration server backend should repeat the `sendBinaryMessages` operation after the reset.

Note: This action is supported on PRO 83xx/TomTom PRO 82xx devices only. To use this action, you need to have the additional service LINK.connect booked.

Technical details

SOAP endpoint address / function name	messagesService#resetBinaryMessages
Request limits	10 requests / minute

Parameters

resetBinaryMessages requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to resetBinaryMessage:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

clearBinaryMessages

Description

`clearBinaryMessages` removes data from the outgoing queue on the triver terminal. This can be used to remove stale data for applications on the driver terminal that do not exist anymore or that are unlikely to connect anymore.

`clearBinaryMessages` is needed because the driver terminal does not remove outgoing data by itself as it cannot decide if an application will connect sometime.

Note: This action is supported on PRO 83xx/TomTom PRO 82xx devices only. To use this action, you need to have the additional service LINK.connect booked.

Technical details

SOAP endpoint address / function name	messagesService#clearBinaryMessages
Request limits	10 requests / minute

Parameters

clearBinaryMessages requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to clearBinaryMessages:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
appid	string (128)	An identifier to mark the application on the destination device. **Note**: Use package style, for example *com.vendor.application.part* If not used, all binary messages on the device will be removed. This parameter is optional.

Drivers

Synchronising driver lists

Manually

Using the Webfleet user interface, you can send the complete driver list to a vehicle.

1. Select your vehicle.
2. Go to **Contract/Details**.
3. Select **Configure**.
4. Select the **Basic settings** tab.
   You can find the section **Synchronize driver list**.
5. Click **Send**.
   The current driver list is transmitted.

Automatically

Webfleet will send the latest changes of the driver list to a currently connected LINK 7xx/5xx/4xx/3xx/Driver Terminal combination when the device:

- Sends a working time related message.
- Sends a text message.
- Sends a predefined general status message.
- Sends an order related predefined status message.
- Changes the status of the active order.
- Activates in Webfleet.
- Starts navigating.

Specifically for connected navigation devices, the driver list is updated when the user accesses the driver list and the device is connected to Webfleet.

showDriverReportExtern

Description

Lists all active drivers matching the indicated parameters and filters.

Technical details

SOAP endpoint address / function name	objectsAndPeopleReportingService#showDriverReport
Request limits	10 requests / minute

Parameters

showDriverReportExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to showDriverReportExtern:

Parameter	Type	Description
filterstring	string	An arbitrary pattern that is located in the object data.
driverno	string	Identifying number of a driver. Unique within an account. Search for the specified driver number. Can be used alternatively to `driveruid`.
ungroupedonly	string	Allows to filter on drivers that are currently not assigned to a certain driver group. Only valid value is `true`. This parameter overrules the group name filter parameter. This parameter is optional.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.

Result

Result for showDriverReportExtern:

Parameter	Type	Description
driverno	string (15)	Account-unique driver number, case-sensitive.
name1	string (50)	
name2	string (50)	
name3	string (50)	
addrno	string (10)	Identifying number of an address. Unique within an account.
state	string (2)	ISO 3166-1 alpha-2 code for the country (capital letters).
zip	string (10)	
city	string (250)	
street	string (250)	
telmobile	string (50)	
telprivate	string (50)	
email	string (50)	
description		

Parameter	Type	Description
company	string (250)	
objectno	string (10)	Identifying number of an object. Unique within an account, case sensitive. Identifies the driver currently signed on to this object.
signontime	dateTime	The time when the driver signed on to this object. ISO 8601-formatted date and time in the UTC time-zone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00
dt_cardid	string (16)	A driver card id as used with the digital tachograph.
dt_cardcountry	string (2)	ISO 3166-1 alpha-2 code for the country (capital letters).
rll_buttonid	string (16)	Remote LINK/ID key identifier
signonrole	nonNegativeInteger	Valid values: • `1` - Driver • `2` - Co-Driver • `3` - Worker
current_workstate	nonNegativeInteger	Valid values: • `0` - Unknown • `1` - Free time (PND only) • `2` - Pause (PND and digital tachograph) • `3` - Standby (digital tachograph only) • `4` - Working (PND only) • `5` - Driving (digital tachograph only) • `6` - Other work (digital tachograph only)
current_workingtimestart	dateTime	ISO 8601-formatted date and time in the UTC time-zone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00
current_workingtimeend	dateTime	ISO 8601-formatted date and time in the UTC time-zone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00

Parameter	Type	Description
current_working-time	duration	ISO 8601-formatted duration. *Example: PT362S represents a duration of 362 seconds, as does PT6M2S.* Accumulated working time if `current_workstate` indicates a non-working state.
addr_latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
addr_longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
manualassignment	int	Indicates whether the driver is assigned to the current vehicle manually by using `attachDriverToVehicle` or automatically by the driver who has logged on using a navigation device or digital tachograph. Valid values: • `0` - automatically • `1` - manually • `<empty>` - if no vehicle is assigned or the information is not available
pin	nonNegativeInteger (20) ([1-9]\d*)	The PIN that a driver uses to log in to the Driver Terminal.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.

Parameter	Type	Description
driver_keys	string	JSON encoded string with an array of JSON objects composed by the driver keys, associated with the driver. The JSON objects have following format: `{ "driver_key_type": <number>, "driver_key_value": <string> }` Valid values: • `1` - Remote LINK • `2` - Generic identifier *Usage example*: RFID used with PRO 2020 Driver Terminal.

driving_license	string	JSON encoded string with a JSON object composed by the driving licence details of the driver.
		The JSON object has following format:
		`{ "number": <string>, "country": <string>, "state": <string>, "issue_date": <date>, "expiry_date": <date>, "types": <array> }`
		Field `types` is an array of JSON objects which have the following format:
		`{ "code": <string> "valid_from": <date>, "valid_to": <date> }`
		Valid values of licence type code: A, A1, A2, AM, B, B1, BE, C, C1, C1E, CE, D, D1, D1E, DE, F, G, L, S, T

insertDriverExtern

Description

This action creates a driver.

Technical details

SOAP endpoint address / function name	driverManagementService#insertDriver

Request limits	10 requests / minute

Parameters

insertDriverExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to insertDriverExtern:

Parameter	Type	Description
driverno	string (15)	Account-unique driver number, case-sensitive. This parameter is required.
name	string (50)	This parameter is required.
name2	string (50)	Additional driver name. This parameter is optional.
name3	string (50)	Additional driver name. This parameter is optional.
company	string (250)	
description	string (4000)	
addrno	string (10)	Identifying number of an address. Unique within an account. The following conditions apply: • If `addrno` is indicated and it shows a valid address number, the corresponding address will be assigned to the driver. • If `addrno` is indicated and the value is empty, the relation between an assigned address and the driver will be deleted. • If `addrno` is empty and at least one of the free text address components `country`, `zip`, `city` and `street` is indicated, the stored address relation for this driver will be removed and the value of the address component will be attached to the driver.
country	string (2)	ISO 3166-1 alpha-2 code for the country (capital letters).
zip	string (10)	
city	string (250)	
street	string (250)	
telmobile	string (50)	
telprivate	string (50)	

Parameter	Type	Description
pin	nonNegativeInteger (20) ([1-9]\d*)	Identifying number used for driver identification. To be used for all devices except LINK (classic) and TomTom GO 715. This parameter is optional.
email	string (50)	
dt_cardid	string (16)	A driver card id as used with the digital tachograph. This parameter is optional.
dt_cardcountry	string (2)	ISO 3166-1 alpha-2 code for the country (capital letters). This parameter is optional.
rll_buttonid	string (16)	Remote LINK/ID key identifier. Usage details: • the ID key provided must be 'available', that is, not assigned to any other driver • to remove the ID key assignment, provide an empty parameter • to assign a new ID key, first remove the old assignment • if the parameter is not part of the request, the current assignment is left unchanged This parameter is optional.
driver_key	string	Driver key identifier composed by driver key type and driver key value. The value must follow the format: `driverkeytype,driverkeyvalue` Usage details: • the driver key provided must not be assigned to any other driver • to remove a driver key assignment, `driverkeytype` followed by an empty `driverkeyvalue` must be provided • to change multiple driver keys you can specify the `driver_key` parameter multiple times Valid values for `driverkeytype`: • `1` - Remote LINK • `2` - Generic Identifier, which is used for example for RFID used with PRO 2020 Driver Terminal
license_number	string (20)	Number of the driving licence. This parameter is required if any driving licence field is filled.

Parameter	Type	Description
license_country	string (2)	Country code of the driving licence. ISO 3166-1 alpha-2 for the country (capital letters). This parameter is required if any driving license field is filled.
license_state	string (30)	Issuing state of the driving licence. This parameter is required if any driving licence field is filled.
license_issue_date	date	Issue date of the driving licence. This parameter is required if any driving licence field is filled.
license_expiry_date	date	Expiry date of the driving licence. This parameter is optional.

license_type	string	Driving licence type composed by licence type code, valid from date and optional valid to date.
		The value must follow one of these formats:
		`licenseTypeCode,validFrom,validTo`
		`licenseTypeCode,validFrom`
		Valid values of licence type code: A, A1, A2, AM, B, B1, BE, C, C1, C1E, CE, D, D1, D1E, DE, F, G, L, S, T.
		This parameter is required if any driving licence field is filled.

updateDriverExtern

Description

This action updates driver details.

Technical details

SOAP endpoint address / function name	driverManagementService#updateDriver
Request limits	10 requests / minute

Parameters

updateDriverExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to updateDriverExtern:

Parameter	Type	Description
driverno	string (15)	New driver number to be set for the driver identified by `driverno_old` or `driveruid`, account-unique, case-sensitive. This paramater is optional.
name	string (50)	
name2	string (50)	Additional driver name. This parameter is optional.
name3	string (50)	Additional driver name. This parameter is optional.
company	string (250)	
description	string (4000)	
addrno	string (10)	Identifying number of an address. Unique within an account. The following conditions apply: • If `addrno` is indicated and it shows a valid address number, the corresponding address will be assigned to the driver. • If `addrno` is indicated and the value is empty, the relation between an assigned address and the driver will be deleted. • If `addrno` is empty and at least one of the free text address components `country`, `zip`, `city` and `street` is indicated, the stored address relation for this driver will be removed and the value of the address component will be attached to the driver.
country	string (2)	ISO 3166-1 alpha-2 code for the country (capital letters).
zip	string (10)	
city	string (250)	
street	string (250)	
telmobile	string (50)	
telprivate	string (50)	
pin	nonNegativeInteger (20) ([1-9]\d*)	Identifying number used for driver identification. To be used for all devices except LINK (classic) and TomTom GO 715.
email	string (50)	

Parameter	Type	Description
dt_cardid	string (16)	A driver card id as used with the digital tachograph. This parameter is optional.
dt_cardcountry	string (2)	ISO 3166-1 alpha-2 code for the country (capital letters). This parameter is optional.
rll_buttonid	string (16)	Remote LINK/ID key identifier Usage details: • the ID key provided must be 'available', that is, not assigned to any other driver • to remove the ID key assignment, provide an empty parameter • to assign a new ID key, first remove the old assignment • if the parameter is not part of the request, the current assignment is left unchanged This parameter is optional.
driverno_old	string (10)	Account-unique driver number, case-sensitive. Either `driverno_old` or `driveruid` is required.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Either `driverno_old` or `driveruid` is required.
driver_key	string	Driver key identifier composed of driver key type and driver key value. The value must follow the format: `driverkeytype,driverkeyvalue` Usage details: • the driver key provided must not be assigned to any other driver • to remove a driver key assignment, `driverkeytype` followed by an empty `driverkeyvalue` must be provided • to change multiple driver keys you can specify the `driver_key` parameter multiple times Valid values for `driverkeytype`: • `1` - Remote LINK • `2` - Generic Identifier, which is used for example for RFID used with PRO 2020 Driver Terminal
license_number	string (20)	Number of the driving licence. This parameter is required if any driving licence field is filled.

Parameter	Type	Description
license_country	string (2)	Country code of the driving licence. ISO 3166-1 alpha-2 for the country (capital letters). This parameter is required if any driving licence field is filled.
license_state	string (30)	Issuing state of the driving licence. This parameter is required if any driving licence field is filled.
license_issue_date	date	Issue date of the driving licence. This parameter is required if any driving licence field is filled.
license_expiry_date	date	Expiry date of the driving licence. This parameter is optional.

license_type	string	Driving license type composed by licence type code, valid from date and optional valid to date.
		The value must follow one of these formats:
		`licenseTypeCode,validFrom,validTo`
		`licenseTypeCode,validFrom`
		Valid values of licence type code: A, A1, A2, AM, B, B1, BE, C, C1, C1E, CE, D, D1, D1E, DE, F, G, L, S, T.
		This parameter is required if any driving licence field is filled.

Additional notes for driver license data

To delete existing driver license data use empty field in license_number parameter.

deleteDriverExtern

Description

This action deactivates the indicated driver.

Technical details

SOAP endpoint address / function name	driverManagementService#deleteDriver
Request limits	10 requests / minute

Parameters

deleteDriverExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to deleteDriverExtern:

Parameter	Type	Description
driverno	string (15)	Account-unique driver number, case-sensitive. This parameter is required. Can be used alternatively to `driveruid`.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.

showOptiDriveIndicator

Description

`showOptiDriveIndicator` shows the OptiDrive indicator per driver, the values of all its influencing variables and the factors each of them is calculated from for a defined period. The start and end of this period can be adjusted up to a maximum time frame of seven days within the last three months.

The OptiDrive indicator for the defined period delivered by `showOptiDriveIndicator` is calculated from pre-aggregated values per calendar day. The aggregated results produced by `showOptiDriveIndicator` are based on information for each driver across multiple vehicles.

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#showOptiDriveIndicator
Request limits	10 requests / minute

Parameters

showOptiDriveIndicator requires the following common parameters:

- Authentication parameters
- General parameters

  Make use of the `range_pattern` for floating weeks.

  `rangefrom_string` and `rangeto_string` are treated as pure date values, there is no time indication needed. The specified day is considered in the currently valid time zone of the account. The date range must not be greater than 8 days ranging from the beginning of the start day to the ending of the current day. The date range must not be greater than 7 days when defined for an earlier period.

Parameters specific to showOptiDriveIndicator:

Parameter	Type	Description
driverno	string (15)	Identifying number of a driver. Unique within an account, case-sensitive. This parameter is optional.

Parameter	Type	Description
drivergroupname	string	Name of the group of drivers the driver is assigned to, case-sensitive. This parameter is optional.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. This parameter is optional.

Result

Result for showOptiDriveIndicator:

Parameter	Type	Description
driverno	string	Identifying number of a driver. Unique within an account.
drivername	string	Display name of a driver.
totaldrivingtime	integer	The total driving time of the driver in the defined period in seconds.
optidrive_indicator	float	The overall OptiDrive indicator.
trend	float	Value ranges from `-1` to `1`. Indicates the development of the OptiDrive value of the past showing whether the driver's driving style was increasing, stable or decreasing. Increase of the OptiDrive value is indicated by positive numbers. Decrease of the OptiDrive value is indicated by negative numbers. Values around 0 indicate the OptiDrive indicator stays stable. *Example 1: A value of `0.1` indicates only a slight increase of the OptiDrive indicator.* *Example 2: A value of `-0.9` indicates a heavy decrease of the OptiDrive indicator.*
totaldistance	integer	Total distance driven in defined period in meters.
speeding_indicator	float	Value of influencing variable for speeding events. Value ranges from 0 to 1.
speeding_time	integer	Total time of speeding events in seconds.

Parameter	Type	Description
speeding_avgex-cess	float	The average speed limit excess of all speeding events within defined period, in km/h. *Example: During the first speeding event the speed limit was broken by 10 km/h. During the second speeding event the speed limit was broken by 20 km/h. The resulting* `speeding_avgexcess` *is 15 km/h.*
speeding_avglimit	float	The average of effective speed limits during all speeding events in km/h. *Example: If the speed limit during the first speeding event was 50 km/h and the speed limit during the second speeding event was 70 km/h, the resulting* `speeding_avglimit` *is 60 km/h.*
speeding_count	integer	Number of speeding events registered during the defined period.
drivingevents_in-dicator	float	Value of influencing variable for driving events. Value ranges from `0` to `1`.
dri-vingevents_count	integer	Number of driving events registered during defined period.
dri-vingevents_avg-severity	float	The average severity of driving events in percent.
idling_indicator	float	Value of influencing variable for idling. Value ranges from `0` to `1`.
idling_time	integer	Time spent idling during defined period, in seconds.
idling_wastedfuel	integer	Total amount of fuel wasted while idling within the defined period.
fuelusage_indica-tor	float	Value of influencing variable for fuel consumption. Value ranges from `0` to `1`.
fuelusage_total	integer	Total amount of fuel used by driver within the defined period in millilitres.
fuelusage_refer-ence	integer	The amount of fuel the driver should have used according to the fuel consumption reference value for the covered distance within the defined period.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.
coasting_indicator	float	Value of influencing variable for coasting. Value ranges from `0` to `1`.

Parameter	Type	Description
coasting_reference_time	nonNegativeInteger	Potential coasting time in seconds, i.e. the sum of coasting and braking time.
coasting_time	nonNegativeInteger	Actual coasting time in seconds.
coasting_distance	nonNegativeInteger	Actual coasting distance in meters.
constant_speed_indicator	float	Value of influencing variable for constant speed. Value ranges from `0` to `1`.
constant_speed_reference_time	nonNegativeInteger	Potential driving time in seconds at which constant speed was evaluated.
constant_speed_time	nonNegativeInteger	Actual driving time with constant speed, in seconds.
constant_speed_reference_time_urban	nonNegativeInteger	Potential driving time with constant speed in urban areas, in seconds.
constant_speed_reference_time_extra_urban	nonNegativeInteger	Potential driving time with constant speed in extra urban areas, in seconds
constant_speed_reference_time_motorway	nonNegativeInteger	Potential driving time with constant speed on motorway, in seconds.
constant_speed_time_urban	nonNegativeInteger	Actual driving time with constant speed in urban areas, in seconds.
constant_speed_time_extra_urban	nonNegativeInteger	Actual driving time with constant speed in extra urban areas, in seconds.
constant_speed_time_motorway	nonNegativeInteger	Actual driving time with constant speed on motorways, in seconds.
green_speed_indicator	float	Value of influencing variable for green speed. Value ranges from `0` to `1`.
green_speed_reference_time	nonNegativeInteger	The actual time of a trip the green speed reference value would apply, in seconds. *Example: The time where the speed limit was greater than > 80 km/h.*

Parameter	Type	Description
green_speed_time	nonNegativeInteger	Driving time with speed lower than/equal to current green speed limit (never greater than `green_speed_reference_time`), in seconds.
high_revving_indicator	float	Value of influencing variable for highrevving. Value ranges from `0` to `1`.
high_revving_reference_time	nonNegativeInteger	Driving time in which RPM data was reported, in seconds.
high_revving_time	nonNegativeInteger	Actual time in which the high-revving was reported, in seconds.
energy_consumption_indicator	float	Value of influencing variable for energy consumption of electric vehicles. Value ranges from `0` to `1`.
cruise_control_indicator	float	Value of influencing variable for cruise control. Value ranges from `0` to `1`.
video_indicator	float	Value of influencing variable video events. Value ranges from `0` to `1`.
ev_driving_indicator	float	Value of influencing variable for EV driving style. Value ranges from `0` to `1`.

showDriverGroups

Description

This action retrieves a list of all driver groups.

Technical details

SOAP endpoint address / function name	driverManagementService#showDriverGroups
Request limits	10 requests / minute

Parameters

showDriverGroups requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to showDriverGroups:

Parameter	Type	Description
drivergroupname	string	The given entry filters the result list on driver groups with matching names. This parameter is optional.

Result

Result forshowDriverGroups:

Parameter	Type	Description
drivergroupname	string	
drivergroupuid	string	A unique, unchangable identifier for the indicated driver group, automatically generated.

showDriverGroupDrivers

Description

This action lists the assignment of drivers to driver groups. Each driver can be in no, one or more than one group.

Using one or both of the parameters `driverno` and `drivergroupname`, only according matches are returned.

Technical details

SOAP endpoint address / function name	driverManagementService#showDriverGroupDrivers
Request limits	10 requests / minute

Parameters

showDriverGroupDrivers requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to showDriverGroupDrivers:

Parameter	Type	Description
driverno	string	Identifying number of a driver. Unique within an account, case-sensitive. Can be used alternatively to `driveruid`.
drivergroupname	string	The given entry filters the result list on driver groups with matching names, case-sensitive. This parameter is optional.

Parameter	Type	Description
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.

Result

Result for showDriverGroupDrivers:

Parameter	Type	Description
driverno	string	Identifying number of a driver. Unique within an account.
drivername	string	Display name of a driver.
drivergroupname	string	
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.
drivergroupuid	string	A unique, unchangeable identifier for the indicated driver group, automatically generated.

attachDriverToGroup

Description

This action assigns a driver to a specific group.

Technical details

SOAP endpoint address / function name	driverManagementService#attachDriverToGroup
Request limits	10 requests / minute

Parameters

attachDriverToGroup requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to attachDriverToGroup:

Parameter	Type	Description
drivergroupname	string	Name of the group the driver shall be assigned to, case-sensitive. This parameter is required.

Parameter	Type	Description
driverno	string	Identifying number of a driver. Unique within an account, case-sensitive. This parameter is required. Can be used alternatively to `driveruid`.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.

detachDriverFromGroup

Description

This action detaches a driver from a specific driver group.

Technical details

SOAP endpoint address / function name	driverManagementService#detachDriverGroup
Request limits	10 requests / minute

Parameters

detachDriverFromGroup requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to detachDriverFromGroup:

Parameter	Type	Description
drivergroupname	string	Name of the group the driver group from which the driver shall be detached, case-sensitive. This parameter is required.
driverno	string	Identifying number of a driver. Unique within an account, case-sensitive. This parameter is required. Can be used alternatively to `driveruid`.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.

insertDriverGroup

Description

This action creates a driver group. The group name must not start with '**sys$**'.

Technical details

SOAP endpoint address / function name	driverManagementService#insertDriverGroup
Request limits	10 requests / minute

Parameters

insertDriverGroup requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to insertDriverGroup:

Parameter	Type	Description
drivergroupname	string	Name of the driver group, case-sensitive. This parameter is required.

deleteDriverGroup

Description

This action deletes a driver group and the assignments of all drivers to that group. The drivers detached through this action are not being deleted.

Technical details

SOAP endpoint address / function name	driverManagementService#deleteDriverGroup
Request limits	10 requests / minute

Parameters

deleteDriverGroup requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to deleteDriverGroup:

Parameter	Type	Description
drivergroupname	string	Name of the driver group, case-sensitive. This parameter is required.

updateDriverGroup

Description

This action allows to update the name of a driver group while retaining the assignment of drivers to that group. The group name must not start with '**sys$**'.

Technical details

SOAP endpoint address / function name	driverManagementService#updateDriverGroup
Request limits	10 requests / minute

Parameters

updateDriverGroup requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to updateDriverGroup:

Parameter	Type	Description
drivergroupname	string	Name of the driver group, case-sensitive. This parameter is required.
drivergroup-name_old	string	Name of the driver group, case-sensitive. This parameter is required.

attachDriverToVehicle

Description

Using `attachDriverToVehicle` you can manually assign a driver to a specific vehicle.

If the driver was previously assigned to another vehicle, they will be automatically detached from that vehicle.

This action cannot be executed when the driver is not manually assigned to another vehicle – that means by not using `attachDriverToVehicle`. Additionally, this action cannot be executed when a different driver is not manually assigned to the respective vehicle. This can be the case if the driver for example logs on to a vehicle by using a Driver Terminal or a digital tachograph.

Note: Keep in mind that this action should only be used as a kind of exception (e.g. one-time driver assignment/detachment) and not for the regular and reoccurring driver assignment. Generally the driver assignment should only be done by using a Driver Terminal from Webfleet, PRO 2020, Remote LINK Working Time or a digital tachograph.

Technical details

SOAP endpoint address / function name	driverManagementService#attachDriverToVehicle
Request limits	10 requests / minute

Parameters

`attachDriverToVehicle` requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to attachDriverToVehicle:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
driverno	string	Identifying number of a driver. Unique within an account, case-sensitive. This parameter is required. Can be used alternatively to `driveruid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.

detachDriverFromVehicle

Description

Using `detachDriverFromVehicle` you can remove the manual assignment of a driver from a vehicle.

This action cannot be executed if the driver was not manually assigned by using `attachDriverToVehicle`. This can be the case when the driver for example logged on to the vehicle using a Driver Terminal or the digital tachograph.

Note: Keep in mind that this action should only be used as a kind of exception (e.g. one-time driver assignment/detachment) and not for the regular and reoccurring driver assignment. Generally the driver assignment should only be done by using a Driver Terminal from Webfleet, PRO 2020, Remote LINK Working Time or a digital tachograph.

Technical details

SOAP endpoint address / function name	driverManagementService#detachDriverFromVehicle
Request limits	10 requests / minute

Parameters

detachDriverFromVehicle requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to detachDriverFromVehicle:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
driverno	string (15)	Identifying number of a driver. Unique within an account, case-sensitive. This parameter is required. Can be used alternatively to `driveruid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.

getDriverRdtRules

Description

`getDriverRdtRules` returns the rules for calculating remaining driving times, working times and infringements for a specific driver or all drivers of the account.

These rules are applied

- when retrieving remaining driving times for the driver
- when determining infringements reported by showEventReportExtern

The rules can be changed by the action `updateDriverRdtRules`.

`getDriverRdtRules` is available for accounts that have at least one vehicle for which one of the following additional services is booked:

- Remaining Driving Times
- Webfleet Tachograph Manager Plus
- Webfleet TachoShare Plus

Related actions

- getRemainingDrivingTimesEU – calculates remaining driving times
- updateDriverRdtRules – changes the rules for calculating driving and working times and infringements
- showEventReportExtern – returns event notifications, including infringement notifications

Technical details

SOAP endpoint address / function name	driverManagementService#getDriverRdtRules
Request limits	10 requests / minute

Parameters

getDriverRdtRules requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getDriverRdtRules:

Parameter	Type	Description
driverno	string (15)	Identifying number of a driver. Unique within an account, case-sensitive. Can be used alternatively to `driveruid`.
driveruid	string (30)	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.

Result

Result for getDriverRdtRules:

Parameter	Type	Description
driverno	string (15)	Identifying number of a driver. Unique within an account.
driveruid	string (30)	A unique, unchangeable identifier for the indicated driver, automatically generated.

Parameter	Type	Description
country	string	The country determines which regulation and which rules are available for configuration. Refer to updateDriverRdtRules for possible combinations of country, regulation and rules. Format: ISO 3166-1 alpha-2 or NONE Allowed values: • DE • IT • DK • NL • AT • GB • IE • CH • FR • SI • PL • BE • ES • SE • NO • FI • PT • NONE
regulation_type	string	The regulation that should be applied when calculating (remaining) driving and working times and infringements. Allowed values: • EU_561 Regulation (EC) No 561/2006 – Regulation of rules on driving times, breaks and rest periods. Applicable to all EU countries. • ARV1_2011 Chauffeurverordnung, ARV 1 – Verordnung über die Arbeits- und Ruhezeit der berufsmässigen Motorfahrzeugführer und -führerinnen. Applicable to Swizerland (CH) only. • FPERSV Fahrpersonalverordnung – Verordnung zur Durchführung des Fahrpersonalgesetzes. Applicable to Germany (DE) only.
bus_rule	boolean	If `true` the 12-day rule is considered in the calculation, which is regulated by (EC) No 561/2006, Article 8(6a).
working_time_rules	boolean	If `true` country specific working time rules are considered in the calculation.

Parameter	Type	Description
employment_type	string	The type of employment. Only applicable if `regulation_type` ARV1_2011 is selected. Allowed values: • EMPLOYEE • SELF_EMPLOYED If EMPLOYEE is selected, the Swiss working time regulations are considered in the calculation. Otherwise, if SELF_EMPLOYED is selected, they are not considered.
isDefault	boolean	If `true` then these are default rules. As long as no specific rule has been saved for a driver, predefined default rules are automatically used. Only after setting the rules using `updateDriverRdtRules` the first time, the value `isDefault` is always `false`.

updateDriverRdtRules

Description

`updateDriverRdtRules` changes the rules for calculating remaining driving times, working times and infringements for a specific driver of the account. The action is available for accounts that have at least one vehicle for which one of the following additional features is booked:

- Remaining Driving Times
- Webfleet Tachograph Manager Plus
- Webfleet TachoShare Plus

These rules are applied

- when retrieving remaining driving times for the driver
- when determining infringements reported by showEventReportExtern

The currently stored rules can be retrieved by the action `getDriverRdtRules`.

Related actions

- getRemainingDrivingTimesEU – calculates remaining driving times
- getDriverRdtRules – retrieves the rules for calculating driving and working times and infringements
- showEventReportExtern – returns event notifications, including infringement notifications

Technical details

SOAP endpoint address / function name	driverManagementService#updateDriverRdtRules
Request limits	10 requests / minute

Parameters

updateDriverRdtRules requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to updateDriverRdtRules:

Parameter	Type	Description
driverno	string (15)	Identifying number of a driver. Unique within an account, case-sensitive. Can be used alternatively to `driveruid`.
driveruid	string (30)	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.
country	string	The country determines which regulation and which rules are available for configuration. Refer to section **Allowed parameter combinations** for possible combinations of `country` and `regulation_type`. Format: ISO 3166-1 alpha-2 or NONE Allowed values: • DE • IT • DK • NL • AT • GB • IE • CH • FR • SI • PL • BE • ES • SE • NO • FI • PT • NONE This parameter is mandatory.

Parameter	Type	Description
regulation_type	string	The regulation that should be applied when calculating (remaining) driving and working times and infringements. Allowed values: • EU_561 Regulation (EC) No 561/2006 – Regulation of rules on driving times, breaks and rest periods. Applicable to all EU countries. • ARV1_2011 Chauffeurverordnung, ARV 1 – Verordnung über die Arbeits- und Ruhezeit der berufsmässigen Motorfahrzeugführer und -führerinnen. Applicable to Swizerland (CH) only. • FPERSV Fahrpersonalverordnung – Verordnung zur Durchführung des Fahrpersonalgesetzes. Applicable to Germany (DE) only. This parameter is mandatory.
bus_rule	boolean	If `true` the 12-day rule is considered in the calculation, which is regulated by (EC) No 561/2006, Article 8(6a).
working_time_rules	boolean	If `true` country specific working time rules are considered in the calculation.
employment_type	string	The type of employment. Only applicable if `regulation_type` ARV1_2011 is selected. Allowed values: • EMPLOYEE • SELF_EMPLOYED If EMPLOYEE is selected, the Swiss working time regulations are considered in the calculation. Otherwise, if SELF_EMPLOYED is selected, they are not considered.

Allowed parameter combinations

Driving and working times rules are defined by the country, the regulation and rules. In each country only a certain regulation and a certain set of rules can be applied. The corresponding parameters `country`, `regulation_type`, `bus_rules`, `working_time_rules` and `employment_type` depend on each other and only specific combinations are allowed as shown in the table below.

Note: Country `NONE` is a special case that, if stored in combination with `EU_561`, allows to reset the rules of a driver to the default. This default rule configuration is the same that is automatically applied to those drivers, for which no own configuration has yet been saved.

Country	Regulation	Type
DE	EU_561	• bus_rule • working_time_rules
DE	FPersV	• bus_rule
IT	EU_561	• bus_rule • working_time_rules
DK	EU_561	• bus_rule
NL	EU_561	• bus_rule • working_time_rules
AT	EU_561	• bus_rule • working_time_rules
GB	EU_561	• bus_rule • working_time_rules
IE	EU_561	• bus_rule • working_time_rules
CH	ARV1_2011	• bus_rule • employment_type
FR	EU_561	• bus_rule • working_time_rules
SI	EU_561	• bus_rule
PL	EU_561	• bus_rule
BE	EU_561	• bus_rule
ES	EU_561	• bus_rule
SE	EU_561	• bus_rule
NO	EU_561	• bus_rule
FI	EU_561	• bus_rule
PT	EU_561	• bus_rule
NONE	EU_561	

getOptiDriveProfiles

Description

This action returns a list of OptiDrive Profiles available for the account. Optionally, the result can be restricted to a single driver only.

OptiDrive Profiles are an extension of what was previously known as OptiDrive KPI weights which were configured in the account settings. The main difference is that OptiDrive Profiles are assigned per driver, which means that on the same account multiple drivers can have different configurations on how their OptiDrive score is calculated. Each account has at least one, default OptiDrive profile and each driver is always assigned to one and only one profile.

Technical details

SOAP endpoint address / function name	optiDriveProfileService#getOptiDriveProfiles
Request limits	10 requests / minute

Parameters

getOptiDriveProfiles requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getOptiDriveProfiles:

Parameter	Type	Description
driverno	string	Identifying number of a driver. Unique within an account. Can be used alternatively to `driveruid`.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.

Result

Result for getOptiDriveProfiles:

Parameter	Type	Description
profile_name	string (50)	A name of the OptiDrive profile. In case of the default OptiDrive profile it is translated. The names of other profiles are chosen by users and are not translated.
speeding_kpi_weight	int	The Speeding KPI weight in the OptiDrive score, represented as an unbounded, non-normalized value, used as a proportion in the weighted average of the overall score.

Parameter	Type	Description
driving_kpi_weight	int	The Driving KPI weight in the OptiDrive score, represented as an unbounded, non-normalized value, used as a proportion in the weighted average of the overall score.
fuel_kpi_weight	int	The Fuel KPI weight in the OptiDrive score, represented as an unbounded, non-normalized value, used as a proportion in the weighted average of the overall score.
coasting_kpi_weight	int	The Coasting KPI weight in the OptiDrive score, represented as an unbounded, non-normalized value, used as a proportion in the weighted average of the overall score.
high_revving_kpi_weight	int	The High Revving KPI weight in the OptiDrive score, represented as an unbounded, non-normalized value, used as a proportion in the weighted average of the overall score.
constant_speed_kpi_weight	int	The Constant Speed KPI weight in the OptiDrive score, represented as an unbounded, non-normalized value, used as a proportion in the weighted average of the overall score.
green_speed_kpi_weight	int	The Green Speed KPI weight in the OptiDrive score, represented as an unbounded, non-normalized value, used as a proportion in the weighted average of the overall score.
idling_kpi_weight	int	The Idling KPI weight in the OptiDrive score, represented as an unbounded, non-normalized value, used as a proportion in the weighted average of the overall score.
cruise_control_kpi_weight	int	The Cruise Control KPI weight in the OptiDrive score, represented as an unbounded, non-normalized value, used as a proportion in the weighted average of the overall score.
video_kpi_weight	int	The Video KPI weight in the OptiDrive score, represented as an unbounded, non-normalized value, used as a proportion in the weighted average of the overall score.
energy_consumption_kpi_weight	int	The Energy Consumption KPI weight in the OptiDrive score, represented as an unbounded, non-normalized value, used as a proportion in the weighted average of the overall score.
ev_driving_kpi_weight	int	The EV Driving Style KPI weight in the OptiDrive score, represented as an unbounded, non-normalized value, used as a proportion in the weighted average of the overall score.

Parameter	Type	Description
driver_distracted_video_event_weight	int	The Driver Distracted video event type weight in the Video KPI score. Represents the proportional impact of the video events of this type on the Video KPI score.
eating_video_event_weight	int	The Eating video event type weight in the Video KPI score. Represents the proportional impact of the video events of this type on the Video KPI score.
smoking_video_event_weight	int	The Smoking video event type weight in the Video KPI score. Represents the proportional impact of the video events of this type on the Video KPI score.
seatbelts_video_event_weight	int	The Seatbelts video event type weight in the Video KPI score. Represents the proportional impact of the video events of this type on the Video KPI score.
mobile_phone_video_event_weight	int	The Mobile Phone video event type weight in the Video KPI score. Represents the proportional impact of the video events of this type on the Video KPI score.
driverUids	string array	JSON encoded string with an array of driverUids representing drivers assigned to this OptiDrive profile. Those drivers have their scores calculated according to configuration present in this profile.

Addresses

showAddressReportExtern

Description

This action returns a list of addresses matching the parameters and filters.

Technical details

SOAP endpoint address / function name	addressService#showAddressReport
Request limits	6 requests / minute

Parameters

showAddressReportExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to showAddressReportExtern:

Parameter	Type	Description
filterstring	string	An arbitrary pattern that is located in the object data.
addressgroup-name	string	A name of an address group.
ungroupedonly	string	Allows to filter on addresses that are currently not assigned to a certain address group. Only valid value is `true`. This parameter overrules the group name filter parameter. This parameter is optional.
addrnr	string (10)	Identifying number of an address. Unique within an account.
addruid	string	A unique, unchangable identifier for the indicated address, automatically generated.

Result

Result for showAddressReportExtern:

Parameter	Type	Description
addrnr	string (10)	Identifying number of an address. Unique within an account.
addrname1	string	
addrname2	string	
addrname3	string	
addrstate	string (2)	ISO 3166-1 alpha-2 code for the country (capital letters).
addrzip	string	
addrcity	string	
addrstreet	string	
addrinfo	string	
addrregion	string	
contact	string	
teloffice	string	
telmobile	string	
telprivate	string	
fax	string	
mailaddr	string	
latitude	string	Geographic latitude in the form GGG° MM′ SS.S″ N/S in the WGS84 coordinate system.
longitude	string	Geographic longitude in the form GGG° MM′ SS.S″ E/W in the WGS84 coordinate system.
radius	int	Radius of the circle around the address position. Object positions within the circle are associated to this address. The nearest address is selected in case of object positions within overlapping circles of different addresses. Unit of measurement is meters.
visible	boolean	Controls visibility of this address on the map. If set to `1`, the address will be shown on the map.

Parameter	Type	Description
color	string	Valid values: • `brightblue` • `dullorange` • `brightorange` • `brightpalegreen` • `grassgreen` • `brightpurple` • `darkred` • `dullazure` • `deeppurple` • `brightazure` • `orangehighlight` • `brightgreen` • `brightorangehighlight` • `khaki` • `paleblue` • `turquoise`
positiony	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
positionx	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
addruid	string	A unique, unchangable identifier for the indicated address, automatically generated.
mapcode	string (20)	A mapcode represents a location. Learn more about mapcodes.

showAddressGroupReportExtern

Description

Returns a list of address groups within the account matching the filter parameter.

Technical details

SOAP endpoint address / function name	addressService#showAddressGroupReport
Request limits	900 requests / hour

Parameters

showAddressGroupReportExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to showAddressGroupReportExtern:

Parameter	Type	Description
filterstring	string	An arbitrary pattern that is located in the address group name.

Result

Result for showAddressGroupReportExtern:

Parameter	Type	Description
addressgroupname	string	The name of the address group as it appears in Webfleet.
addressgroupuid	string	A unique, unchangable identifier for the indicated address group, automatically generated.

showAddressGroupAddressReportExtern

Description

Shows a list with all associations between addresses and address groups. Each address can be in more than one address group, but must not necessarily belong to a group. The relationship between addresses and address groups is of m:n cardinality.

Technical details

SOAP endpoint address / function name	addressService#showAddressGroupAddressReport
Request limits	900 requests / hour

Parameters

showAddressGroupAddressReportExtern requires the following common parameters:

- Authentication parameters
- General parameters

Table 4-151: Parameters specific to showAddressGroupAddressReportExtern:

Parameter	Type	Description
filterstring	string	An arbitrary pattern that is located in the address group name.

Result

Result for showAddressGroupAddressReportExtern:

Parameter	Type	Description
addressgroup-name	string	The name of the address group as it appears in Webfleet.
addrnr	string (10)	Identifying number of an address. Unique within an account.
addruid	string	A unique, unchangable identifier for the indicated address, automatically generated.
addressgroupuid	string	A unique, unchangable identifier for the indicated address group, automatically generated.

insertAddressExtern

Description

Inserts an address record.

Technical details

SOAP endpoint address / function name	addressService#insertAddress
Request limits	900 requests / hour

Parameters

insertAddressExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to insertAddressExtern:

Parameter	Type	Description
addrnr	string (10)	Identifying number of an address, case-sensitive. Unique within an account. This parameter is mandatory.
addrname1	string (50)	This parameter is mandatory.
addrname2	string (50)	
addrname3	string (50)	
addrstreet	string (50)	
addrzip	string (10)	
addrcity	string (50)	
addrcountry	string (2)	ISO 3166-1 alpha-2 code for the country (capital letters).
addrregion	string	Name (or abbreviation) of the county, district or federal state this locality belongs to.
contact	string (50)	
teloffice	string (20)	
telmobile	string (20)	
telprivate	string (20)	
fax	string (20)	
mailaddr	string (254)	
radius	int	Radius of the circle around the address position. Object positions within the circle are associated to this address. The nearest address is selected in case of object positions within overlapping circles of different addresses. Unit of measurement is meters. Default value is 250.
addrinfo	string (1000)	
positiony	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.

Parameter	Type	Description
positionx	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
visible	boolean	Controls visibility of this address on the map. If set to `1`, the address will be shown on the map.
color	string	The colour associated with the address. Case-sensitive. Valid values: • `brightblue` • `dullorange` • `brightorange` • `brightpalegreen` • `grassgreen` • `brightpurple` • `darkred` • `dullazure` • `deeppurple` • `brightazure` • `orangehighlight` • `brightgreen` • `brightorangehighlight` • `khaki` • `paleblue` • `turquoise`
addrgrpname	string (30)	Name of an address group, case-sensitive. Unique within an account. The address will be automatically associated with this group. This parameter is optional.
addrgrpuid	string	A unique, unchangable identifier for the indicated address group, automatically generated.
mapcode	string (20)	A mapcode represents a location. Either `mapcode` or `latitude`/`longitude` coordinates can be used, but not both. Learn more about mapcodes.

updateAddressExtern

Description

Updates an existing address record.

Note: The following parameters are deleted when using this action if they are not specified:

addrname2, *addrname3*, *addrstreet*, *addrzip*, *addrcity*, *addrcountry*, *addrregion*, *contact*, *teloffice*, *telmobile*, *telprivate*, *fax*, *mailaddr*, *addrinfo*, *color*

Technical details

SOAP endpoint address / function name	addressService#updateAddress
Request limits	900 requests / hour

Parameters

updateAddressExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to updateAddressExtern:

Parameter	Type	Description
addrnr	string (10)	Identifying number of an address. Unique within an account. **Note**: The parameter `addrnr` or `addruid` is required.
addrname1	string (50)	This parameter is required.
addrname2	string (50)	
addrname3	string (50)	
addrstreet	string (50)	
addrzip	string (10)	
addrcity	string (50)	
addrcountry	string (2)	ISO 3166-1 alpha-2 code for the country (capital letters).
addrregion	string	Name (or abbreviation) of the county, district or federal state this locality belongs to.
contact	string (50)	
teloffice	string (20)	

Parameter	Type	Description
telmobile	string (20)	
telprivate	string (20)	
fax	string (20)	
mailaddr	string (254)	
radius	int	Radius of the circle around the address position. Object positions within the circle are associated to this address. The nearest address is selected in case of object positions within overlapping circles of different addresses. Unit of measurement is meters. Default value is 250.
addrinfo	string (1000)	
positiony	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
positionx	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
visible	boolean	Controls visibility of this address on the map. If set to `1`, the address will be shown on the map.

Parameter	Type	Description
color	string	The colour associated with the address. Case-sensitive. Valid values: • `brightblue` • `dullorange` • `brightorange` • `brightpalegreen` • `grassgreen` • `brightpurple` • `darkred` • `dullazure` • `deeppurple` • `brightazure` • `orangehighlight` • `brightgreen` • `brightorangehighlight` • `khaki` • `paleblue` • `turquoise`
addruid	string	A unique, unchangable identifier for the indicated address, automatically generated. **Note**: The parameter `addrnr` or `addruid` is required.
mapcode	string (20)	A mapcode represents a location. Either `mapcode` or `latitude`/`longitude` coordinates can be used but not both. Learn more about mapcodes.

deleteAddressExtern

Description

Deletes the specified address.

Technical details

SOAP endpoint address / function name	addressService#deleteAddress
Request limits	900 requests / hour

Parameters

deleteAddressExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to deleteAddressExtern:

Parameter	Type	Description
addrnr	string (10)	Identifying number of an address. Unique within an account.
addruid	string	A unique, unchangeable identifier for the indicated address, automatically generated.

attachAddressToGroupExtern

Description

Assigns an address to a specific address group.

Technical details

SOAP endpoint address / function name	addressService#attachAddressToGroup
Request limits	900 requests / hour

Parameters

attachAddressToGroupExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to attachAddressToGroupExtern:

Parameter	Type	Description
addrnr	string (10)	Identifying number of an address. Unique within an account.
addrgrpname	string (30)	Name of an address group, case-sensitive. Unique within an account.
addruid	string	A unique, unchangable identifier for the indicated address, automatically generated.
addrgrpuid	string	A unique, unchangable identifier for the indicated address group, automatically generated.

detachAddressFromGroupExtern

Description

Deletes the assignment of an address to an address group.

Technical details

SOAP endpoint address / function name	addressService#detachAddressFromGroup
Request limits	900 requests / hour

Parameters

detachAddressFromGroupExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to detachAddressFromGroupExtern:

Parameter	Type	Description
addrnr	string (10)	Identifying number of an address, case-sensitive. Unique within an account.
addrgrpname	string (30)	Name of an address group, case-sensitive. Unique within an account.
addruid	string	A unique, unchangable identifier for the indicated address, automatically generated.
addrgrpuid	string	A unique, unchangable identifier for the indicated address group, automatically generated.

insertAddressGroupExtern

Description

This action creates an address group in an account.

Technical details

SOAP endpoint address / function name	addressService#insertAddressGroup
Request limits	900 requests / hour

Parameters

insertAddressGroupExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to insertAddressGroupExtern:

Parameter	Type	Description
addrgrpname	string (30)	Name of an address group. Unique within an account.

deleteAddressGroupExtern

Description

This action deletes an address group and the assignments of all addresses assigned to that group. With this action you can also delete all addresses that are assigned to the address group.

Technical details

SOAP endpoint address / function name	addressService#deleteAddressGroup
Request limits	900 requests / hour

Parameters

deleteAddressGroupExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to deleteAddressGroupExtern:

Parameter	Type	Description
addrgrpname	string (30)	Name of an address group, case-sensitive. Unique within an account.
deleteaddresses	boolean	If set to `1` all addresses attached to the indicated address group will be deleted in addition to the address group itself and the address assignments. **IMPORTANT**: Please use carefully. All addresses assigned will be deleted without further confirmation. This parameter is optional.
addrgrpuid	string	A unique, unchangable identifier for the indicated address group, automatically generated.

Events

showEventReportExtern

Description

Provides a list of event notifications.

Technical details

SOAP endpoint address / function name	eventReportingService#showEventReport
Request limits	1 request / minute

Parameters

showEventReportExtern requires the following common parameters:

- Authentication parameters
- General parameters

`showEventReportExtern` requires the following other parameters:

- Date range filter parameters

Parameters specific to showEventReportExtern:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. This parameter is optional.
eventlevel_cur	nonNegativeInteger	Event level. Valid values: • `0` - Message • `1` - Notice/Information • `2` - Warning • `3` - Alarm 1 • `4` - Alarm 2 • `5` - Alarm 3 This parameter is optional.

Parameter	Type	Description
resolved	boolean	Flags a resolved event. Valid values: • `0` - Unresolved • `1` - Resolved This parameter is optional.
acknowledged	boolean	Flags an acknowledged event. Valid values: • `0` - Unacknowledged • `1` - Acknowledged This parameter is optional.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Result

Result for showEventReportExtern:

Parameter	Type	Description
eventid	string	
msgtime	string	
eventtime	string	
objectno	string (10)	Identifying number of an object. Unique within an account, case sensitive.
msgtext	string	
postext	string	
latitude	string	Geographic latitude in the form GGG° MM′ SS.S″ N/S in the WGS84 coordinate system.
longitude	string	Geographic longitude in the form GGG° MM′ SS.S″ E/W in the WGS84 coordinate system.
restime	string	
acktime	string	

Parameter	Type	Description
eventlevel	string	The original event level this event had when it was created. Valid values: • `I` - Information • `W` - Warning • `A` - Alarm
alarmlevel	string	
eventlevel_cur	string	The current event level for this event. Valid values: • `I` - Information • `W` - Warning • `A` - Alarm
resuser	string	Identifies the user who resolved this event. Can be empty if user does not have the relevant rights to access user information.
ackuser	string	Identifies the user who acknowledged this event. Can be empty if user does not have the relevant rights to access user information.
latitude_mdeg	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
longitude_mdeg	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
pos_time	string	The time that is related to the last known position of the vehicle when the event occured. Can be empty if user does not have the relevant rights to see positions of the respective object.

acknowledgeEventExtern

Description

Flags an event at an alarmed level as acknowledged. The event level will be set to the next lower level.

Technical details

SOAP endpoint address / function name	eventReportingService#acknowledgeEvent
Request limits	300 requests / 30 minutes

Parameters

acknowledgeEventExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to acknowledgeEventExtern:

Parameter	Type	Description
eventid	String	

resolveEventExtern

Description

Flags an event below an alarmed level as resolved.

Technical details

SOAP endpoint address / function name	eventReportingService#resolveEvent
Request limits	300 requests / 30 minutes

Parameters

resolveEventExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to resolveEventExtern:

Parameter	Type	Description
eventid	string	

getEventForwardConfigs

Description

This action returns event forwarding configurations.

Technical details

SOAP endpoint address / function name	eventReportingService#getEventForwardConfigs
Request limits	10 requests / minute

Parameters

getEventForwardConfigs requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getEventForwardConfigs:

Parameter	Type	Description
objectgroupname	string (50)	Name of the group of objects. This parameter is optional.
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`. **Note**: Make sure that the object you indicate using `objectno` belongs to the object group you have indicated using `objectgroupname`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`. **Note**: Make sure that the object you indicate using `objectuid` belongs to the object group you have indicated using `objectgroupname`.
eventlevel	int	The level of an event. Valid values: • `1` - Notice/Information • `2` - Warning • `3` - Alarm 1 • `4` - Alarm 2 • `5` - Alarm 3 This parameter is optional.

Result

Results for getEventForwardConfigs:

Parameter	Type	Description
eventforwardconfiguid	string (30)	A unique, unchangeable identifier for the indicated event forwarding configuration, automatically generated.
eventlevel	int	The level of an event. Valid values: • `1` - Notice/Information • `2` - Warning • `3` - Alarm 1 • `4` - Alarm 2 • `5` - Alarm 3
objectgroupname	string (50)	Name of the group of objects.
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated.
recipientcount	int	The number of recipients.

getEventForwardConfigRecipients

Description

`getEventForwardConfigRecipients` returns the primary and all additional recipients of an event forwarding configuration.

Technical details

SOAP endpoint address / function name	eventReportingService#getEventForwardConfigRecipients
Request limits	10 requests / minute

Parameters

getEventForwardConfigRecipients requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getEventForwardConfigRecipients:

Parameter	Type	Description

eventforwardconfiguid	string (30)	A unique unchangeable identifier for the indicated event forwarding configuration, automatically generated. This parameter is mandatory.

Result

Results for getEventForwardConfigRecipients:

Parameter	Type	Description
eventforwardconfiguid	string (30)	A unique, unchangeable identifier for the indicated event forwarding configuration, automatically generated.
recipientname	string (200)	Name of recipient.
recipientaddress	string (200)	Email address of recipient.

insertEventForwardConfig

Description

Using `insertEventForwardConfig` you can create a new event forwarding configuration.

Technical details

SOAP endpoint address / function name	eventReportingService#insertEventForwardConfig
Request limits	10 requests / minute

Parameters

insertEventForwardConfig requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to insertEventForwardConfig:

Parameter	Type	Description
eventlevel	int	The level of an event. Valid values: • `1` - Notice/Information • `2` - Warning • `3` - Alarm 1 • `4` - Alarm 2 • `5` - Alarm 3 This parameter is mandatory.

Parameter	Type	Description
objectgroupname	string (50)	Name of the group of objects. This parameter is optional.
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`. **Note**: Make sure that the object you indicate using `objectno` belongs to the object group you have indicated using `objectgroupname`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`. **Note**: Make sure that the object you indicate using `objectuid` belongs to the object group you have indicated using `objectgroupname`.
recipient	string (400)	Name and email address of recipient. You can separate first name from last name by using a comma. The email address must be a valid email address according to RFC5322 and RFC2606. Separate name from email by using a semicolon. You can indicate multiple recipients by repeating the `recipient` parameter. This parameter is required.

Result

Results for insertEventForwardConfig:

Parameter	Type	Description
eventforwardconfiguid	string (30)	A unique, unchangeable identifier for the indicated event forwarding configuration, automatically generated.

updateEventForwardConfig

Description

Using `updateEventForwardConfig` you can update an existing event forwarding configuration.

Technical details

SOAP endpoint address / function name	eventReportingService#updateEventForwardConfig
Request limits	10 requests / minute

Parameters

updateEventForwardConfig requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to updateEventForwardConfig:

Parameter	Type	Description
eventforwardconfiguid	string (30)	A unique, unchangeable identifier for the indicated event forwarding configuration, automatically generated. This parameter is mandatory.
eventlevel	int	The level of an event. Valid values: • `1` - Notice/Information • `2` - Warning • `3` - Alarm 1 • `4` - Alarm 2 • `5` - Alarm 3 This parameter is required.
objectgroupname	string (50)	Name of the group of objects. This parameter is optional.
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`. **Note**: Make sure that the object you indicate using `objectno` belongs to the object group you have indicated using `objectgroupname`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`. **Note**: Make sure that the object you indicate using `objectuid` belongs to the object group you have indicated using `objectgroupname`.

Parameter	Type	Description
recipient	string (400)	Name and email address of recipient. You can separate first name from family name by using a comma. The email address must be a valid email address according to RFC5322 and RFC2606. Separate name from email by using a semicolon. You can indicate multiple recipients by repeating the `recipient` parameter. This parameter is required.

deleteEventForwardConfig

Description

Using `deleteEventForwardConfig` you can delete an existing event forwarding configuration.

Technical details

SOAP endpoint address / function name	eventReportingService#deleteEventForwardConfig
Request limits	10 requests / minute

Parameters

deleteEventForwardConfig requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to deleteEventForwardConfig:

Parameter	Type	Description
eventforwardconfiguid	string (30)	A unique, unchangeable identifier for the indicated event forwarding configuration, automatically generated. This parameter is mandatory.

Trips and working times

showTripReportExtern

Description

Provides a list of trips of an object. Trips of deleted objects cannot be shown.

In order to prevent your system from being flooded with oversized responses, the result is limited to 10000 entries, if the continuous replication is used. The limit can be adjusted per account on request.

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#showTripReport
Request limits	1 request / minute

Usage patterns

`showTripReportExtern` supports two alternative usages patterns for different application integration scenarios:

- **Continuous replication of trip records** into a local application; the replication is done by referring to a tripid retrieved from a previous call to `showTripReportExtern`.
- **On-demand download of trip records** created within a variable time period using a date range filter (Date range filter parameters). The date range filter applies to the end time of a trip.

For both usage patterns, either all trip records or a subset of trip records for a specific object (selected by providing filter conditions) can be downloaded.

- **Continuous replication**

  If your application requires access to all or specific trip records, a periodic call of `showTripReportExtern` can be used to retrieve all new trip records received since the last call.The recommended frequency of calls depends on the distribution of trip reports over any specific period of time as well as business requirements, but should not be more frequent than one call every 15 minutes.

  Each trip record is identified by a `tripid` that is strictly (monotonically) increasing. While processing the downloaded results of a call to `showTripReportExtern`, store the highest `tripid` and provide this as a parameter to the next call.

  In order to prevent your system from being flooded with oversized responses, the result is limited to 10000 entries, if you are using continuous replication. The limit can be adjusted per account on request.
- **Date range filter behaviour**

  The date range filter applies to the trip end date/time, not the date/time when the trip record was created. Due to this behaviour, there is no guarantee that all trips can be consistently downloaded by e.g. daily calls to `showTripReportExtern` (for example, trip records sent late due to unavailable GPRS coverage might be missed). Continuous replication is therefore the recommended method of retrieving trip records in near real-time.

Parameter restrictions

Before processing a request, all parameters are checked for invalid combinations and an error message is returned if an unacceptable parameter combination is detected.

- If `tripid` is provided, it must be valid, that is, must have been returned by previous calls to `showTripReportExtern`. In order to retrieve an initial value for `tripid`, use a one-time download covering a small time period.
- If no `tripid` is provided, a date range filter (Date range filter parameters) compliant to more restrictions (see below) must be present.
- The time period covered by date range filter (Date range filter parameters) may not be greater than 1 year.
- If the time period covered by date range filter (Date range filter parameters) is greater than 1 month (and less than or equal to 1 year), an `objectno` must be provided to constrain the result set.

Parameters

`showTripReportExtern` requires the following common parameters:

- Authentication parameters
- General parameters

`showTripReportExtern` requires the following other parameters:

- Date range filter parameters

Parameters specific to showTripReportExtern:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. This parameter is optional.
tripid	nonNegativeInteger	This parameter is optional.
driverno	string	Identifying number of a driver. Unique within an account, case-sensitive. Can be used alternatively to `driveruid`.
drivergroupname	string	Name of the group of drivers the driver is assigned to, case-sensitive. This parameter is optional.
objectgroupname	string	Name of the group of objects the object is assigned to, case-sensitive. This parameter is optional.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
externalid	string	For future use. This parameter is optional.

Parameters specific to showTripReportExtern

Parameter	Type	Description
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.

Parameters specific to showTripReportExtern

Result

Result for showTripReportExtern:

Parameter	Type	Description
tripid	string	
tripmode	nonNegativeInteger	Valid values: • `0` - Unknown trip type • `1` - Private trip • `2` - Business trip • `3` - Commute trip • `4` - Correction trip (user changed the odometer value manually)
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectname	string	Display name of an object.
start_time	string	The time recorded when this event started.
start_odometer	long	The odometer reading when this event started.
start_postext	string	A detailed textual description of the location where this event started.
end_time	string	The time recorded when this event ended.
end_odometer	long	The odometer reading when this event ended.
end_postext	string	A detailed textual description of the location where this event ended.
duration	int	The difference between `start_time` and `end_time`. Unit of measurement is 'seconds'.
idle_time	int	The idle time during this trip. Unit of measurement is 'seconds'.
distance	long	The difference between `start_odometer` and `end_odometer`. Unit of measurement is 'meters'.

Result for showTripReportExtern

Parameter	Type	Description
avg_speed	int	The average speed during this trip.
max_speed	int	The maximum speed during this trip.
fuel_usage	float	Fuel consumption in litres, max. three decimals allowed.
start_longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
start_latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
start_formatted_longitude	string	Geographic longitude in the form GGG° MM′ SS.S″ E/W in the WGS84 coordinate system.
start_formatted_latitude	string	Geographic latitude in the form GGG° MM′ SS.S″ N/S in the WGS84 coordinate system.
end_longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.

Result for showTripReportExtern

Parameter	Type	Description
end_latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
end_formatted_longitude	string	Geographic longitude in the form GGG° MM′ SS.S″ E/W in the WGS84 coordinate system.
end_formatted_latitude	string	Geographic latitude in the form GGG° MM′ SS.S″ N/S in the WGS84 coordinate system.
driverno	string	Identifying number of a driver. Unique within an account.
drivername	string	Display name of a driver.
start_addrno	string	Identifying number of an address. Unique within an account.
end_addrno	string	Identifying number of an address. Unique within an account.

Result for showTripReportExtern

Parameter	Type	Description
fueltype	int	Valid values: • `0` - unknown • `1` - diesel • `2` - gasoline • `3` - lpg • `4` - hybrid petrol • `5` - hybrid diesel • `6` - electric • `7` - cng • `8` - lng • `9` - hydrogen • `10` - plug-in hybrid electric • `11` - bioethanol • `12` - biodiesel_fame (Fatty acid methyl esters) • `13` - biodiesel_fame_uco (Fatty acid methyl esters, Used cooking oil) • `14` - hvo (Hydrotreated vegetable oil) • `15` - commercial_diesel • `16` - cbg (Bio-CNG) • `17` - lbg (Bio-LNG)
co2	int	Carbon dioxide emissions in grams.
ep_distance	int	Distance measured by ecoPLUS in meters.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
externalid	string	For future use.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.
optidrive_indicator	float	The overall OptiDrive indicator.
speeding_indicator	float	Value of influencing variable for speeding events. Value ranges from 0 to 1.
drivingevents_indicator	float	Value of influencing variable for driving events. Value ranges from 0 to 1.
idling_indicator	float	Value of influencing variable for idling. Value ranges from 0 and 1.

Result for showTripReportExtern

Parameter	Type	Description
fuelusage_indicator	float	Value of influencing variable for fuel consumption. Value ranges from 0 and 1.
coasting_indicator	float	Value of influencing variable for coasting. Value ranges from `0` to `1`.
constant_speed_indicator	float	Value of influencing variable for constant speed. Value ranges from `0` to `1`.
green_speed_indicator	float	Value of influencing variable for green speed. Value ranges from `0` to `1`.
high_revving_indicator	float	Value of influencing variable for high-revving. Value ranges from `0` to `1`.
energy_usage	float	Energy consumption in kWh. Max. three decimals allowed.
start_battery_level	int	Electric vehicle's battery level in percent at the start of the trip. Values range from 0 to 100.
end_battery_level	int	Electric vehicle's battery level in percent at the end of the trip. Values range from 0 to 100.
start_battery_energy	float	Electric vehicle's battery energy in Wh at the start of the trip.
end_battery_energy	float	Electric vehicle's battery energy in Wh at the end of the trip.
energy_consumption_indicator	float	Value of influencing variable for energy consumption of electric vehicles. Value ranges from `0` to `1`.
cruise_control_indicator	float	Value of influencing variable for cruise control. Value ranges from `0` to `1`.
video_indicator	float	Value of influencing variable for video events. Value ranges from `0` to `1`.
energy_consumption_driving	float	Electric vehicles: Energy used for driving in kWh.
energy_consumption_other	float	Electric vehicles: Energy used for heating, climate, etc. in kWh.

Result for showTripReportExtern

Parameter	Type	Description
enery_recovered	float	Electric vehicles: Energy recovered from braking in kWh.
ev_driving_indicator	float	Value of influencing variable for EV driving style. Value ranges from `0` to `1`.

Result for showTripReportExtern

showTripSummaryReportExtern

Description

This action provides daily accumulated trip information.

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#showTripSummaryReport
Request limits	10 requests / minute

Parameters

showTripSummaryReportExtern requires the following common parameters:

- Authentication parameters
- General parameters

`showTripSummaryReportExtern` requires the following other parameters:

- Date range filter parameters - Maximum range is one year (but only if `objectno` is set.)

Parameters specific to showTripSummaryReportExtern:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. This paramater is mandatory if `objectuid` is not defined.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. This parameter is mandatory if `objectno` is not defined.

Result

Result for showTripSummaryReportExtern:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectname	string	Display name of an object.
start_time	string	The time recorded when this event started.
start_odometer	long	The odometer reading when this event started.
end_time	string	The time recorded when this event ended.
end_odometer	long	The odometer reading when this event ended.
end_postext	string	A detailed textual description of the location where this event ended.
distance	long	The difference between start_odometer and end_odometer. Unit of measurement is 'meters'.
triptime	string	Cumulated trip durations in seconds.
operatingtime	string	Time between start_time and end_time in seconds.
standstill	string	
tours	string	
fuel_usage	float	Fuel consumption in litres, max. three decimals allowed.
co2	int	Carbon dioxide emissions in grams.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
energy_usage	float	Energy consumption in kWh. Max. three decimals allowed.

showTracks

Description

This action retrieves a list of positions of a vehicle for a defined period.

If certain information was not available the corresponding results can be empty.

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#showTracks

Request limits	10 requests / minute

Parameters

showTracks requires the following common parameters:

- Authentication parameters
- General parameters
- Date range filter parameters, periods longer than 2 days are not permitted.

Parameters specific to showTracks:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
externalid	string	For future use. This parameter is optional.

Result

Result for showTracks:

Parameter	Type	Description
pos_time	dateTime	Date and time when the position was registered in Webfleet.
receivetime	dateTime	Date and time when the position was received in Webfleet.
latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.

Parameter	Type	Description
speed	nonNegativeInteger	in km/h
course	nonNegativeInteger	Compass direction in degrees (0° ... 360°):
fuelusage	float	Fuel consumption in litres, max. three decimals allowed. This is a snapshot measurement. It's the same value as shown in the trace view in Webfleet UI.
odometer	int	Odometer value at the position. Only available for LINK 7xx/6xx/5xx/4xx.
country	string	Country code.
state	string	Code for federal state.
energy_usage	float	Energy consumption (snapshot) in kWh. Max. three decimals allowed.

updateLogbook

Descriptions

This action updates some parameters of an existing logbook entry.

When using `updateLogbook`, you only need to include the parameters for which you want to change or delete the values. If you include a parameter and do not indicate a value, the existing value will be deleted.

In addition to the specific parameters listed below the parameters `modifiedby and modifiedon` are being automatically stored.

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#updateLogbook
Request limits	10 requests / minute

Parameters

updateLogbook requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to updateLogbook:

Parameter	Type	Description
tripid	nonNegativeInteger	This parameter is required.
logflag	string	Indicates the trip mode. Valid values: • `1` - Private • `2` - Business • `3` - Commute This parameter is optional.
logpurpose	string (max. 254 characters)	A description of the purpose. This parameter is mandatory when changing the trip mode to business, otherwise optional.
logcontact	string (max. 254 characters)	Contact details for this trip; this is typically at the visited party. This parameter is mandatory when changing the trip mode to business, otherwise optional.
logcomment	string (max. 254 characters)	Any additional annotation for this trip. This parameter is optional.
reason	string (max. 254 characters)	Reason for this trip. This parameter is optional.

showLogbook

Descriptions

This action returns the most recent state of logbook information for an indicated object or trip - this applies to finished trips only.

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#showLogbook
Request limits	10 requests / minute

Parameters

`showLogbook` requires the following common parameters:

- Authentication parameters
- General parameters

`showLogbook` requires the following other parameters:

- Date range filter parameters - The maximum range is one year. The date range filter parameter is optional if `tripid` is specified.

Date range filter behaviour

The date range filter applies to the trip end date/time, not the date/time when the trip record was created.

Parameters specific to showLogbook:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
tripid	nonNegativeInteger	This parameter precedes `objectno`. This parameter is optional.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
modified_since	string	Returns all trips that have been manually edited after the indicated date. The format depends on the value of the General parameters `lang` and `useISO8601`. • lang = de *Format*: `dd.MM.yyyy HH:mm:ss` *Example: 13.07.2007 23:59:59* • lang = en *Format*: `dd/MM/yyyy HH:mm:ss` *Example: 13/07/2007 23:59:59* • lang = nl *Format*: `d-M-yyyy H:mm:ss` *Example: 13-07-2007 23:59:59* This parameter is optional.

Parameters specific to showLogbook

Result

Result for showLogbook:

Parameter	Type	Description
tripid	string	

Result for showLogbook

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectname	string	Display name of an object.
logflag	string	Indicates the trip mode. Valid values: • `1` - Private • `2` - Business • `3` - Commute • `4` - Odometer correction
start_time	string	The time recorded when this event started.
start_odometer	long	The odometer reading when this event started.
start_postext	string	A detailed textual description of the location where this event started.
end_time	string	The time recorded when this event ended.
end_odometer	long	The odometer reading when this event ended.
end_postext	string	A detailed textual description of the location where this event ended.
distance	long	The difference between start_odometer and end_odometer. Unit of measurement is 'meters'.
logpurpose	string (max. 254 characters)	A description of the purpose.
logcontact	string (max. 254 characters)	Contact details for this trip; this is typically at the visited party.
logcomment	string (max. 254 characters)	Any additional annotation for this trip.
modifiedon	dateTime	ISO 8601-formatted date and time in the UTC timezone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00
modifiedby	string	The name of the user who issues the last change.
driverno	string	Identifying number of a driver. Unique within an account.

Result for showLogbook

Parameter	Type	Description
drivername	string	Display name of a driver.
start_addrno	string	Identifying number of an address. Unique within an account.
end_addrno	string	Identifying number of an address. Unique within an account.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
start_latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
start_longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
end_latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.

Result for showLogbook

Parameter	Type	Description
end_longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
avg_speed	int	The average speed during this trip.
max_speed	int	The maximum speed during this trip.
fuel_usage	float	Fuel consumption in litres. Max. three decimals allowed.
co2	int	Carbon dioxide emissions in grams.
end_addrname1	string	A suggested address based on end position information.
end_addrname2	string	A suggested address based on end position information.
end_addrname3	string	A suggested address based on end position information.
end_addrcontact	strings	A suggested address contact based on end position information.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.
energy_usage	float	Energy consumption in kWh. Max. three decimals allowed.
start_battery_level	int	Electric vehicle's battery level in percent at the start of the trip. Values range from 0 to 100.
end_battery_level	int	Electric vehicle's battery level in percent at the end of the trip. Values range from 0 to 100.
start_battery_energy	float	Electric vehicle's battery energy in Wh at the start of the trip.

Result for showLogbook

Parameter	Type	Description
end_battery_energy	float	Electric vehicle's battery energy in Wh at the end of the trip.
energy_consumption_driving	float	Electric vehicles: Energy used for driving in kWh.
energy_consumption_other	float	Electric vehicles: Energy used for heating, climate, etc. in kWh.
energy_recovered	float	Electric vehicles: Energy recovered from braking in kWh.

Result for showLogbook

showLogbookHistory

Descriptions

This action returns a logbook history showing logbook edits including the old and new values.

Change history records are available as of 1st of January 2011.

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#showLogbookHistory
Request limits	10 requests / minute

Parameters

showLogbookHistory requires the following common parameters:

- Authentication parameters
- General parameters

`showLogbookHistory` requires the following other parameters:

- Date range filter parameters - The maximum range is one year. The date range filter parameter is optional if `tripid` is specified.

Parameters specific to showLogbookHistory:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
tripid	nonNegativeInteger	This parameter precedes `objectno`.

Parameter	Type	Description
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Result

Result for showLogbookHistory:

Parameter	Type	Description
tripid	string	
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectname	string	Display name of an object.
logpurposeold	string	The old description of the purpose.
logcontactold	string	The old contact details for this trip; this is typically at the visited party.
logcommentold	string	The old additional annotation for this trip.
logflagold	string	The previous trip mode.
logpurposenew	string	The new description of the purpose.
logcontactnew	string	The new contact details for this trip; this is typically at the visited party.
logcommentnew	string	The new additional annotation for this trip.
logflagnew	string	The new trip mode
modifiedon	dateTime	ISO 8601-formatted date and time in the UTC time-zone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00
modifiedby	string	The name of the user who issued the last change.
reason	string	The reason for the edits.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

updateLogbookMode

Descriptions

This action changes the driver's logbook mode on the LINK 7xx/6xx/5xx/4xx/3xx. The change of the logbook mode will become effective with the next trip. If `updateLogbook-Mode` is executed during a trip, this trip will be ended and a new trip using the new logbook mode will be created.

Limitations

- For LINK 3xx - firmware version 1.92 or higher
- For LINK 5xx/4xx - firmware version 2.0 or higher
- For LINK 6xx/7xx - firmware version 4.0 or higher
- The Operational mode for the LINK device must be set to Logbook.
  You can set the Operational mode for your object (LINK device) in Webfleet as follows:

  1. Select your object.
  2. Under **Contract / Device** click **Configure** in the details panel on the right.
  3. Go to the **Basic settings** tab and select **Logbook** under **Function**.

Note: You cannot change the logbook mode if a digital tachograph is connected to the LINK device.

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#updateLogbookMode
Request limits	10 requests / minute

Parameters

updateLogbookMode requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to updateLogbookMode:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Parameter	Type	Description
logbook_mode	int	The logbook mode to set on the LINK device. Valid values: • `1` - private • `2` - business • `3` - commute This parameter is required.

updateLogbookDriver

Descriptions

`updateLogbookDriver` can be used to change the driver for an existing trip.

Changing the driver after a trip can cause data inconsistencies, for example when managing orders or for some reports. Therefore, this action is disabled by default. Please contact Webfleet Customer Support to enable this action for your Webfleet account.

This action cannot be used for trips of vehicles which have a tachograph connected.

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#updateLogbookDriver
Request limits	10 requests / minute

Parameters

`updateLogbookDriver` requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to `updateLogbookDriver`:

Parameter	Type	Description
tripid	nonNegativeInteger	The ID of the trip to which the driver is assigned. This parameter is required.
driverno	string	The driver number of the driver to be assigned to the trip. This parameter is required.

showWorkingTimes

Description

This report shows work time states changes of the (co-)driver, the locations and vehicle.

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#showWorkingTimes
Request limits	6 requests / minute

Parameter restrictions

Before processing a request, all parameters are checked for invalid combinations and an error message is returned if an unacceptable parameter combination is detected.

- The time period covered by date range filter (Date range filter parameters) may not be greater than 1 month.

Parameters

showWorkingTimes requires the following common parameters:

- Authentication parameters
- General parameters

`showWorkingTimes` requires the following other parameters:

- Date range filter parameters

Parameters specific to showWorkingTimes:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
driverno	string	Identifying number of a driver. Unique within an account, case-sensitive. This parameter is required. Can be used alternatively to `driveruid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.
drivergroupname	string	Name of the group of drivers the driver is assigned to, case-sensitive. This parameter is optional.
objectgroupname	string	Name of the group of objects the driver is assigned to, case-sensitive. This parameter is optional.

Parameter	Type	Description
timelineType	integer	This parameter filters working times intervals by device specific timelines. • 1 - 'Any'. Return activities for all timelines. Default if filter parameter is not provided • 2 - 'Tachograph'. Return activities only from tachograph timeline. • 3 - 'PRO device / Work App'. Return activities only from non-tachograph timeline. • 4 - 'Verified Tachograph'. Return activities only from tachograph timeline, which are verified by Remote Download. This parameter is optional.

Result

Result for showWorkingTimes:

Parameter	Type	Description
driverno	string	Identifying number of a driver. Unique within an account.
drivername	string	Display name of a driver.
role	nonNegativeInteger	Valid values: • `1` - Driver • `2` - Co-Driver • `3` - Worker
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectname	string	Display name of an object.
start_time	string	The time recorded when this event started.
start_odometer	long	The odometer reading when this event started.
start_postext	string	A detailed textual description of the location where this event started.
start_addrno	string	Identifying number of an address. Unique within an account.
start_latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.

Parameter	Type	Description
start_longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
end_time	string	The time recorded when this event ended.
end_odometer	long	The odometer reading when this event ended.
end_postext	string	A detailed textual description of the location where this event ended.
end_addrno	string	Identifying number of an address. Unique within an account.
end_latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
end_longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
workstate	nonNegativeInteger	Valid values: • `0` - Unknown • `1` - Free time (PND only) • `2` - Pause (PND and digital tachograph) • `3` - Standby (digital tachograph only) • `4` - Working (PND only) • `5` - Driving (digital tachograph only) • `6` - Other work (digital tachograph only)
end_objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
end_objectname	string	

Parameter	Type	Description
externaldevice-timeoffset	string	The offset in system time between external device (currently only digital tachograph) and LINK, in seconds.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
end_objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.
isManual	boolean	Whether the activity was introduced manually by the driver or not.
source	string	Indicates the source of the working time with the following possible values: • `TACHOGRAPH_LIVE` - Activity obtained from the Tachograph D8 port • `TACHOGRAPH_DRIVER_FILE` - Activity obtained from a Driver card DDD file • `NON_TACHOGRAPH` - Activity from Non-tachograph device like PRO device / Work App
licensePlate	string	The license plate of the vehicle where the activity took place.

showStandStills

Description

This actions shows a list of all stops for a certain vehicle for a specified period.

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#showStandStills
Request limits	6 requests / minute

Parameters

showStandStills requires the following common parameters:

- Authentication parameters
- General parameters

`showStandStills` requires the following other parameters:

- `range_pattern`, see Date range filter parameters.

Parameters specific to showStandStills:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
externalid	string	For future use. This parameter is optional.

Result

Result for showStandStills:

Parameter	Type	Description
start_time	string	The time recorded when this event started. ISO 8601-formatted date and time in the UTC timezone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00
end_time	string	The time recorded when this event ended. ISO 8601-formatted date and time in the UTC timezone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00
duration	int	The difference between `start_time` and `end_time`. Unit of measurement is 'seconds'.
latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.

Parameter	Type	Description
longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
formatted_latitude	string	Geographic latitude in the form GGG° MM′ SS.S″ N/S in the WGS84 coordinate system.
formatted_longitude	string	Geographic longitude in the form GGG° MM′ SS.S″ E/W in the WGS84 coordinate system.
addrno		Identifying number of an address. Unique within an account.
postext	string	
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
fuel_usage	float	Fuel consumption in litres, max. three decimals allowed.
co2	int	Carbon dioxide emissions in grams.
fueltype	int	Valid values: • `0` - unknown • `1` - diesel • `2` - gasoline • `3` - lpg • `4` - hybrid petrol • `5` - hybrid diesel • `6` - electric • `7` - cng • `8` - lng • `9` - hydrogen • `10` - plug-in hybrid electric • `11` - bioethanol • `12` - biodiesel_fame (Fatty acid methyl esters) • `13` - biodiesel_fame_uco (Fatty acid methyl esters, Used cooking oil) • `14` - hvo (Hydrotreated vegetable oil) • `15` - commercial_diesel • `16` - cbg (Bio-CNG) • `17` - lbg (Bio-LNG)

Parameter	Type	Description
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
externalid	string	For future use.

showIdleExceptions

Description

This report shows a list of idle exceptions detected by the LINK or ecoPLUS of an object or of all objects of a object group. An idle event occurs when an object is still for more than five minutes with engine running.

The report shows all idle exceptions that have ended within the period defined with the `start_time` and `end_time`.

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#showIdleExceptions
Request limits	10 requests / minute

Parameter restrictions

Before processing a request, all parameters are checked for invalid combinations and an error message is returned if an unacceptable parameter combination is detected.

- As historical data is available for 90 days only a historical time period covered by date range filter (Date range filter parameters) may not start earlier than 90 days in the past.
- A future time period covered by date range (Date range filter parameters) may not start in the future.
- If no object is specified, the time period covered by date range filter (Date range filter parameters) may not be greater than 1 day.
- If an object group is specified, the time period covered by date range filter (Date range filter parameters) may not be greater than 1 day.
- If an object is specified, the time period covered by date range filter (Date range filter parameters) may not be greater than 1 month.

Note: Only those idle exceptions are reported that end in the requested date and time range. The start time of the idle exception can be outside this range.

Parameters

showIdleExceptions requires the following common parameters:

- Authentication parameters
- General parameters

`showIdleExceptions` requires the following other parameters:

- Date range filter parameters

Parameters specific to showIdleExceptions:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectgroupname	string	Case-sensitive. This parameter is optional.
driverno	string	Identifying number of a driver. Unique within an account, case-sensitive. Can be used alternatively to `driveruid`.
drivergroupname	string	Name of the group of drivers the driver is assigned to, case-sensitive. This parameter is optional.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.

Result

Result for showIdleExceptions:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectname	string	Display name of an object.
driverno	string	Identifying number of a driver. Unique within an account.
drivername	string	Display name of a driver.
start_time	string	The time recorded when this event started.
end_time	string	The time recorded when this event ended.
idle_duration	int	Duration of idle exception interval in seconds.

Parameter	Type	Description
latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
formatted_latitude	string	Geographic latitude in the form GGG° MM′ SS.S″ N/S in the WGS84 coordinate system.
formatted_longitude	string	Geographic longitude in the form GGG° MM′ SS.S″ E/W in the WGS84 coordinate system.
addrno		Identifying number of an address. Unique within an account.
postext	string	
fuel_usage	float	Fuel consumption in litres, max. three decimals allowed.
co2	int	Carbon dioxide emissions in grams.

Parameter	Type	Description
fueltype	int	Valid values: • `0` - unknown • `1` - diesel • `2` - gasoline • `3` - lpg • `4` - hybrid petrol • `5` - hybrid diesel • `6` - electric • `7` - cng • `8` - lng • `9` - hydrogen • `10` - plug-in hybrid electric • `11` - bioethanol • `12` - biodiesel_fame (Fatty acid methyl esters) • `13` - biodiesel_fame_uco (Fatty acid methyl esters, Used cooking oil) • `14` - hvo (Hydrotreated vegetable oil) • `15` - commercial_diesel • `16` - cbg (Bio-CNG) • `17` - lbg (Bio-LNG)
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.

getObjectKPIs

Description

Using `getObjectKPIs` you can retrieve one or more KPIs (Key Performance Indicators) specific to an individual object.

Note: `getObjectKPIs` is the successor action of getKPIs. All calls to `getKPIs` will be redirected to `getObjectKPIs`.

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#getObjectKPIs
Request limits	10 requests / minute

Parameters

getObjectKPIs requires the following common parameters:

- Authentication parameters
- General parameters

`getObjectKPIs` requires the following other parameters:

- Date range filter parameters

Parameters specific to getObjectKPIs

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
kpinames	string	A comma-separated list of KPI-group names. This parameter is required. Valid values: • `tripstats` • `ecostats` • `optidrive` • `speedingevents` • `drivingevents` • `orders` *Example CSV request parameter:* `kpinames=tripstats,optidrive,orders`
level	string	The level of temporal aggregation of the calculated KPIs. Defaults to `range`. Valid values: • `day` • `month` • `range` When set to `day`, the result contains a KPI for each day within the requested date range. When set to `month`, the result contains one KPI per month (totalised for each month). When set to `range`, the whole result consists of a single KPI (summarized over the whole date range). This parameter is optional.
externalid	string	For future use. This parameter is optional.

Result

Result for getObjectKPIs

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
objectname	string	Display name of an object.
year	string (4)	The year the KPIs are calculated for. Format: `YYYY` Returned if specified in request parameter `level`.
month	string (2)	The month, the KPIs are calculated for. Format: `MM` Returned if specified in request parameter `level`.
day	string (2)	The day of month, the KPIs are calculated for. Format: `DD` Returned if specified in request parameter `level`.
externalid	string	For future use.
<kpi-list>		The following columns depend on the requested KPI-group names. See table KPI names.

getDriverKPIs

Description

Using `getDriverKPIs` you can retrieve one or more KPIs (Key Performance Indicators) specific to an individual driver.

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#getDriverKPIs
Request limits	10 requests / minute

Parameters

getDriverKPIs requires the following common parameters:

- Authentication parameters
- General parameters

`getDriverKPIs` requires the following other parameters:

- Date range filter parameters

Parameters specific to getDriverKPIs

Parameter	Type	Description
driverno	string (15)	Identifying number of a driver. Unique within an account, case-sensitive. Can be used alternatively to `driveruid`.
driveruid	string (30)	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.
kpinames	string	A comma-separated list of KPI-group names. This parameter is required. Valid values: • `tripstats` • `ecostats` • `optidrive` • `speedingevents` • `drivingevents` • `orders` *Example CSV request parameter:* `kpinames=tripstats,optidrive,orders`
level	string	The level of temporal aggregation of the calculated KPIs. Defaults to `range`. Valid values: • `day` • `month` • `range` When set to `day`, the result contains a KPI for each day within the requested date range. When set to `month`, the result contains one KPI per month (totalised for each month). When set to `range`, the whole result consists of a single KPI (summarized over the whole date range). This parameter is optional.

Result

Table 4-221: Result for getDriverKPIs

Parameter	Type	Description
driverno	string (15)	Identifying number of a driver. Unique within an account, case-sensitive.
driveruid	string (30)	A unique, unchangeable identifier for the indicated driver, automatically generated.
drivername	string	Display name of a driver.
year	string (4)	The year the KPIs are calculated for. Format: `YYYY` Returned if specified in request parameter `level`.
month	string (2)	The month the KPIs are calculated for. Format: `MM` Returned if specified in request parameter `level`.
day	string (2)	The day of month the KPIs are calculated for. Format: `DD` Returned if specified in request parameter `level`.
<kpi-list>		The following columns depend on the requested KPI-group names. See table KPI names.

getRemainingDrivingTimesEU

Description

`getRemainingDrivingTimesEU` returns several Remaining Driving Times for one or multiple drivers according to the rules of the EU. The respective driver must be assigned to a vehicle for which one of the following additional features is booked:

- Remaining Driving Times
- Webfleet Tachograph Manager Plus
- Webfleet TachoShare Plus

The calculcation of Remaining Driving Times is based on the rules for calculating remaining driving times, working times and infringements. These rules can be configured for each driver by the action updateDriverRdtRules. The action getDriverRdtRules returns the currently configured rules for a driver.

Disclaimer: The remaining driving times supplied by Webfleet are indicative and are reliant upon the information being obtained from the tachograph and sent to Webfleet via the onboard device installed in a vehicle. The algorithms used to calculate the indicative remaining driving times are based on European driving time legislation and it is you responsibility to verify remaining driving times and any applicable national legislation.

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#getRemainingDrivingTimesEU

Request limits	10 requests / minute

Parameters

`getRemainingDrivingTimesEU` requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to `getRemainingDrivingTimesEU`:

Parameter	Type	Description
driverno	string (15)	Identifying number of a driver. Unique within an account, case-sensitive. Can be used alternatively to `driveruid`.
driveruid	string (30)	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.
drivergroupname	string	Name of the group of drivers the driver is assigned to, case-sensitive. Can be used alternatively to `drivergroupuid`.
drivergroupuid	string (30)	A unique, unchangable identifier for the indicated driver group, automatically generated. Can be used alternatively to `drivergroupname`.

Result

Result for getRemainingDrivingTimesEU:

Parameter	Type	Description
driverno	string (15)	Identifying number of a driver. Unique within an account.
driveruid	string (30)	A unique, unchangeable identifier for the indicated driver, automatically generated.
drivername	string (50)	Display name of a driver.
reference_time	datetime	Timestamp as reference for all remaining driving time values. This value can be used to calculate the timestamp at the end of a remaining time. ISO 8601-formatted date and time in the UTC time-zone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00

Parameter	Type	Description
next_break	integer	Remaining time until the latest possible start of next break in milliseconds. This is the latest possible end of the current driving or working period, limited by the remaining driving and working time of the current shift as well as the end of the shift. The value is negative if the break should have already been started. The value is not available if the driver is on **Break**.
next_break_change	integer	Indicator whether the remaining driving time until the start of next break is constant or in which direction it is currently changing. This information can be used for continuous calculation of remaining time on client side without frequent polling. Valid values: • `-1` - Remaining time is decreasing. • `0` - Remaining time is constant. • `1` - Remaining time is increasing.
next_rest	integer	Remaining time until the latest possible start of next rest period in milliseconds. This is the latest possible end of the current shift, limited by the resting time requirements and the availability of daily reduced rest periods. The value is negative if the rest should have already been started. The value is not available if the driver is on **Break**.
next_rest_change	integer	Indicator whether `next_rest` is constant or in which direction it is currently changing. This information can be used for continuous calculation of remaining time on client side without frequent polling. Valid values: • `-1` - Remaining time is decreasing. • `0` - Remaining time is constant. • `1` - Remaining time is increasing.
remain_driving_today	integer	Remaining driving time of the current shift in milliseconds. This value is limited by the weekly remaining driving time, the availability of daily driving time extensions and the end of the shift. If **Working hours legislation** is enabled for the driver, it is also limited by the daily working time. The value is negative if the daily maximum allowed driving time has been exceeded.

Parameter	Type	Description
remain_driving_today_change	integer	Indicator whether `remain_driving_today` is constant or in which direction it is currently changing. This information can be used for continuous calculation of remaining driving time on client side without frequent polling. Valid values: • `-1` - Remaining time is decreasing. • `0` - Remaining time is constant. • `1` - Remaining time is increasing.
total_driving_today	integer	Total driving time for today in milliseconds.
total_driving_today_change	integer	Indicator whether `total_driving_today` is constant or in which direction it is currently changing. This information can be used for continuous calculation of the driving time on client side without frequent polling. Valid values: • `0` - Total time is constant. • `1` - Total time is increasing.
max_driving_week	integer	Maximum weekly driving time within the current week in milliseconds. The 90 hours rule during two consecutive weeks is considered. If **Working hours legislation** is enabled, the value is also limited by the weekly maximum working time.
remain_driving_week	integer	Remaining weekly driving time for the current week in milliseconds which is limited by maximum weekly driving time. The value is negative if the weekly driving time exceeds the maximum weekly driving time.
remain_driving_week_change	integer	Indicator whether `remain_driving_week` is constant or in which direction it is currently changing. This information can be used for continuous calculation of remaining driving time on client side without frequent polling. Valid values: • `-1` - Remaining time is decreasing. • `0` - Remaining time is constant. • `1` - Remaining time is increasing.
remain_driving_next_week	integer	Remaining driving time for the next week in milliseconds. The value can be negative.

Parameter	Type	Description
remain_driving_next_week_change	integer	Indicator whether `remain_driving_next_week` is constant or in which direction it is currently changing. This information can be used for continuous calculation of remaining driving time on client side without frequent polling. Valid values: • `-1` - Remaining time is decreasing. • `0` - Remaining time is constant. • `1` - Remaining time is increasing.
max_extended_shifts	integer	Maximum number of allowed daily driving time extensions per week.
extended_shifts	integer	Number of used daily driving time extensions in current week.
max_reduced_rests	integer	Maximum number of allowed reduced rests per week.
reduced_rests	integer	Number of used reduced rest periods in current week.
rest_balances	string	A string that carries a list of balances for reduced rest periods. One rest balance consists of comma separated values: • `deadline` (datetime, timestamp until the balance has to be fulfilled) • `duration` (integer, balance amount in milliseconds) Multiple rest balances are separated by semicolon. *Example: 2014-07-01T16:00:00Z,120000;2014-07-02T17:00:00Z,500000*
infringements	boolean	`true` if the driver has already committed social infringements.
max_driving_today	integer	Maximum driving time of the current shift in milliseconds. This value is limited by the weekly remaining driving time and the availability of daily driving time extensions. If **Working hours legislation** is enabled for the driver, it is also limited by the daily working time.
max_working_today	integer	Maximum daily working time of the current shift in milliseconds. This value is limited by the weekly remaining working time. This value is only available if **Working hours legislation** is enabled for the driver.

Parameter	Type	Description
remain_working_today	integer	Remaining working time of the current shift in milliseconds, limited by the weekly remaining working time and the end of the shift. This value is only available if **Working hours legislation** is enabled for the driver.
total_otherwork_today	integer	The total duration of activity **Other work** of the current shift in milliseconds.
total_availability_today	integer	The total duration of activity **Availability** of the current shift in milliseconds.
last_activity_from_drivercard	datetime	The time at which the last activity included in the downloaded driver card data ended, including manual corrections. ISO 8601-formatted date and time in the UTC timezone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00
next_break_min_duration	integer	Duration of minimum uninterrupted break in milliseconds to finish the current driving or working period.
next_rest_min_duration	integer	Required minimum duration of the next rest in milliseconds.
current_shift_start	datetime	Start of the current shift. This value is not available if no shift has started. ISO 8601-formatted date and time in the UTC timezone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00
latest_end_current_shift	datetime	Latest end of the current shift. This value is not available if no shift has started. ISO 8601-formatted date and time in the UTC timezone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00

Parameter	Type	Description
earliest_start_next_shift	datetime	The earliest possible start of the next shift, if the driver would immediately start the daily rest. ISO 8601-formatted date and time in the UTC timezone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00
max_working_week	integer	The maximum weekly working time of the current week in milliseconds. Limited by a weekly maximum and the average weekly working time. This value is only available if **Working hours legislation** is enabled for the driver.
remain_working_week	integer	Remaining working time of the current week in milliseconds limited by the maximum weekly working time. This value is only available if **Working hours legislation** is enabled for the driver.

KPI names

KPI group	Returned column	Description
tripstats	tripstats_count	Number of trips that ended within the defined period.
	tripstats_mileage	Total mileage of all trips that ended within the defined period (in meters).
	tripstats_mileage-business	Total mileage of all trips that ended within the defined period in business or unknown mode (in meters).
	tripstats_mileageprivate	Total mileage of all trips that ended within the defined period in private or commute mode (in meters).
	tripstats_drivingtime	Total trip time minus the total idling time of all trips that ended within the defined period (in seconds). **Note**: The first trip may start before the defined period and thus the total driving time may exceed the length of the defined period.

KPI group	Returned column	Description
	tripstats_usagetime	Latest end time minus earliest start time of all trips that ended within a day. If the defined period covers more than one day, `usagetime` is the sum of the daily usage times of all covered days. **Note**: The earliest start time may be on a previous day and thus the daily usage time may exceed 24 hours.
ecostats	ecostats_fuelusage	Total amount of fuel consumed within the defined period (in milliliters).
	ecostats_co2emission	Total amount of carbon dioxide emitted within the defined period (in gram).
	ecostats_wastedfuel	Total amount of fuel wasted while idling within the defined period (in milliliters).
	ecostats_idletime	Total idling time within the defined period (in seconds).
	ecostats_overrevvingtime	Total driving time with high RPM within the defined period (in seconds).
optidrive	optidrive_indicator	The value of the OptiDrive indicator for the defined period. Its value ranges from 0.0 (bad) to 1.0 (good).
	optidrive_indicator_speeding	Value of influencing variable for speeding events, ranging from 0.0 to 1.0 (good).
	optidrive_indicator_driving	Value of influencing variable for driving events, ranging from 0.0 to 1.0.
	optidrive_indicator_idling	Value of influencing variable for idling events, ranging from 0.0 to 1.0.
	optidrive_indicator_fuel	Value of influencing variable for fuel consumtion, ranging from 0.0 to 1.0.
	optidrive_indicator_energy_consumption	Value of influencing variable for energy consumption of electric vehicles, ranging from 0.0 to 1.0.
speedingevents	speedingevents_count	Total number of all speeding-exception intervals that ended within the defined period. A speeding exception is reported if the vehicle speed was continuously over the speed limit, with a tolerance given by an account setting, for at least 30 seconds. The speed limit is the minimum of road-speed limit and vehicle-speed limit.
	speedingevents_speedingtime	Average length of all speeding-exception intervals that ended within the defined period (in seconds).

KPI group	Returned column	Description
	speedingevents_avg_violation	Average violation of the speed limit of all speed-exception intervals that ended within the defined period (in km/h).
drivingevents	drivingevents_count	Total number of driving events (braking, steering) registered within the defined period.
	drivingevents_avg_severity	Average severity of the driving events. The severity ranges from 1 (weak) to 5 (strong) and is dependent on the event type (braking or steering), the vehicle type, and the speed at the time when the event happened.
	drivingevents_braking_count	Total number of braking events registered within the defined period.
	drivingevents_braking_avg_severity	Average severity of the braking events.
	drivingevents_steering_count	Total number of steering events registered within the defined period.
	drivingevents_steering_avg_severity	Average severity of the steering events.
orders	orders_completed	Total number of all orders that have been completed within the defined period.
	orders_cancelled	Total number of all orders that have been cancelled or rejected within the defined period.
	orders_ontime	Total number of completed orders (as defined above), whose event time is less than or equal to the planned arrival time.

Miscellaneous reports

showIOReportExtern

Description

This action returns a list of events recorded with the inputs and outputs of the LINK device.

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#showIOReport
Request limits	10 requests / minute

Parameters

showIOReportExtern requires the following common parameters:

- Authentication parameters
- General parameters

`showIOReportExtern` requires the following other parameters:

- Date range filter parameters

Parameters specific to showIOReportExtern:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
iofilter	string	Name of a digital input. Can be specified multiple times to get combinations of all the specified inputs.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Result

Result for showIOReportExtern:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Maximum capacity is 40 bytes, actual number of characters depends on the byte count consumed by the UTF-8 encoding of those characters.
objectname	string	Display name of an object.
start_time	string	The time recorded when this event started.
start_odometer	int	The odometer reading when this event started.
start_postext	string	A detailed textual description of the location where this event started.
end_time	string	The time recorded when this event ended.
end_odometer	int	The odometer reading when this event ended.
end_postext	string	A detailed textual description of the location where this event ended.
reference	string	Name of the input or output channel.
distance	int	The difference between start_odometer and end_odometer. Unit of measurement is 'meters'.
duration	int	The difference between start_time and end_time. Unit of measurement is 'seconds'.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

showAccelerationEvents

Description

This action shows unwanted driver behaviour. It shows a list of events with information on excessive acceleration, breaking or cornering, based on a threshold defined by the user. These events only cover short time periods, e.g. two to four seconds for a sharp turn.

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#showAccelerationEvents
Request limits	10 requests / minute

Parameter restrictions

Before processing a request, all parameters are checked for invalid combinations and an error message is returned if an unacceptable parameter combination is detected.

One of the following parameters is mandatory:

- `objectno`
- `objectuid`
- `objectgroupname`
- `driverno`
- `driveruid`
- `drivergroupname`

Parameters

showAccelerationEvents requires the following common parameters:

- Authentication parameters
- General parameters

`showAccelerationEvents` requires the following other parameters:

- Date range filter parameters

Parameters specific to showAccelerationEvents:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
driverno	string	Identifying number of a driver. Unique within an account. Search for the specified driver number. Can be used alternatively to `driveruid`.
drivergroupname	string	Name of the group of drivers the driver is assigned to, case-sensitive. This parameter is optional.

Parameter	Type	Description
avg_accel	int	Acceleration in milli-g. This value is the lower threshold used for including acceleration events with an average acceleration higher than indicated. Use this value to limit the size of the result list and receive events with more relevant severity. Varies between a minimum of 0 and a maximum of approx.1100. • # 449 - very low severity • 450 to 549 - low severity • 550 to 649 - medium severity • 650 to 749 - high severity • # 750 - very high severity The result may not include average acceleration in the very low severity range. This parameter is optional.
accelerationtypes	int	Specifies which acceleration types will be reported. You can specify multiple values for this parameter. Separate the values by comma when using the CSV interface. Valid values: • `0` - acceleration (can be enabled on request) • `1` - brake • `2` - steering left • `3` - steering right When you do not indicate the parameter the types 1,2 and 3 will be returned.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
externalid	string	For future use. This parameter is optional.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.

Result

Result for showAccelerationEvents:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectname	string	Display name of an object.

Parameter	Type	Description
driverno	string	Identifying number of a driver. Unique within an account.
drivername	string	Display name of a driver.
start_time	Date	Event start. ISO 8601-formatted date and time in the UTC timezone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00
end_time	Date	Event end. ISO 8601-formatted date and time in the UTC timezone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00
start_speed	float	The speed measured at the start of the acceleration event in km/h.
end_speed	float	The speed measured at the end of the acceleration event in km/h.
max_accelspeed	float	The speed at which the maximum acceleration during the acceleration event was measured in km/h.
avg_accel	int	Acceleration in milli-*g*. Varies between a minimum of 0 and a maximum of 2000. • # 449 - very low severity • 450 to 549 - low severity • 550 to 649 - medium severity • 650 to 749 - high severity • # 750 - very high severity
max_accel	int	Acceleration in milli-*g*.

Parameter	Type	Description
accel_dir	int	Direction of the acceleration in relation to the vehicle, in degrees. • `0` - min • `360` - max Understand the values as follows: • `315–45` - forward acceleration (can be enabled on request) • `45–135` - cornering to the left • `135–225` - braking • `225–315` - cornering to the right
accel_type	nonNega-tiveInteger	Possible values are: • `0` - acceleration (can be enabled on request) • `1` - brake • `2` - steering left • `3` - steering right
latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
postext	string	Position at event end time.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
duration	int	The duration of the event, in milliseconds.
severity	float	Severity of the event. Dependent on event type (braking or steering), vehicle type, and speed. Ranges from1.0 to 5.0.
roadspeedlimit	float	The actual road speed limit taken from the map data in km/h.

Parameter	Type	Description
externalid	string	For future use.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.

showSpeedingEvents

Description

This action shows unwanted driver behaviour. It shows a list of trips or part of trips with information on excessive speeding.

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#showSpeedingEvents
Request limits	10 requests / minute

Parameter restrictions

Before processing a request, all parameters are checked for invalid combinations and an error message is returned if an unacceptable parameter combination is detected.

One of the following parameters is mandatory:

- `objectno`
- `objectuid`
- `objectgroupname`
- `driverno`
- `driveruid`
- `drivergroupname`

Parameters

showSpeedingEvents requires the following common parameters:

- Authentication parameters
- General parameters

`showSpeedingEvents` requires the following other parameters:

- Date range filter parameters

Parameters specific to showSpeedingEvents:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.

Parameter	Type	Description
driverno	string	Identifying number of a driver. Unique within an account. Search for the specified driver number. Can be used alternatively to `driveruid`.
objectgroupname	string	A name of an object group.
drivergroupname	string	Name of the group of drivers the driver is assigned to, case-sensitive. This parameter is optional.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
externalid	string	For future use. This parameter is optional.
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.

Result

Result for showSpeedingEvents:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectname	string	Display name of an object.
driverno	string	Identifying number of a driver. Unique within an account.
drivername	string	Display name of a driver.
start_time	string	The time recorded when this event started.
start_longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.

Parameter	Type	Description
start_latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
start_postext	string	A detailed textual description of the location where this event started.
end_time	string	The time recorded when this event ended.
end_longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
end_latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
end_postext	string	A detailed textual description of the location where this event ended.
max_speed	int	Maximum speed driven during the speeding event.
min_speed	string	Minimum speed driven during the speeding event.
avg_speed	int	The average speed driven during the speeding event.
object_speedlimit	float	The speed limit defined for the vehicle in km/h.
road_speedlimit	float	Information about actual speed limits for the specific road taken from map data in km/h.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
duration	int	The duration of the event, in milliseconds.
distance	int	The distance the vehicle was speeding, in meters.
externalid	string	For future use.

Parameter	Type	Description
driveruid	string	A unique, unchangeable identifier for the indicated driver, automatically generated. Can be used alternatively to `driverno`.
area_speedlimit	float	Information about speed limits for Webfleet areas in km/h.

showDigitalInputStateMileage

Description

This action returns all possible switching state combinations of digital inputs along with the mileage for each combination.

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#showDigitalInputS-tateMileage
Request limits	10 requests / minute

Parameters

`showDigitalInputStateMileage` requires the following common parameters:

- Authentication parameters
- General parameters

`showDigitalInputStateMileage` requires the following other parameters:

- Date range filter parameters

Parameters specific to showDigitalInputStateMileage:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
iofilter	string	Name of a digital input. Can be specified multiple times to get combinations of all the specified inputs. This parameter is mandatory, i.e. at least one iofilter must be specified.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Parameters specific to showDigitalInputStateMileage

Result

Result for showDigitalInputStateMileage:

Parameter	Type	Description
names	string	Names of the digital inputs. Separated by comma.
statuses	string	States of the inputs (ON/OFF), separated by comma.
distance	int	The distance driven with the specific input combination. Unit of measurement is 'meters'.

Result for showDigitalInputStateMileage

getChargerConnections

Description

`getChargerConnections` returns information about charger connections for a single object or for all objects in the account for a selected time period.

Parameters

getChargerConnections requires the following common parameters:

- Authentication parameters
- General parameters

`getChargerConnections` requires the following other parameters:

- Date range filter parameters

Parameters specific to `getChargerConnections`:

Parameter	Type	Description
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.

Result

Result for getChargerConnections:

Parameter	Type	Description
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
driveruid	string (30)	Identifier of the driver who has started charging.
start_time	string	Date and time of the charging event start.

Parameter	Type	Description
end_time	string	Date and time of the charging event end.
longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10-6 grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 106 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10-6 grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 106 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
location	string	
start_battery_level	float	Battery level in percentages at which the charging started.
end_battery_level	float	Battery level in percentages at which the charging finished.
odometer	int	Odometer value (in meters) at which the charging started.
distance	int	Distance since last charge (in meters).
range_start	int	Vehicle range when the charger is connected.
range_end	int	Vehicle range when the charger is disconnected.
cycle_no	int	Counter of charger connections since activation in Webfleet electric vehicles service.
kwh_charged	float	Amount of energy charged. Unit is KWh.

getCrashLog

Description

Using this action you can retrieve all crash log data reported by a LINK device.

Supported devices:

- LINK 410
- LINK 510/530
- LINK 610/640
- LINK 710/740
- LINK 210/240/245

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#getCrashLog
Request limits	10 requests / minute

Parameter restrictions

Before processing a request, all parameters are checked for invalid combinations and an error message is returned if an unacceptable parameter combination is detected.

One of the following parameters is mandatory:

- `objectno`
- `objectuid`

Parameters

`getCrashLog` requires the following common parameters:

- Authentication parameters
- General parameters

`getCrashLog` requires the following other parameters, maximum range is 2 days:

- Date range filter parameters

Parameters specific to getCrashLog:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Result

Result for getCrashLog:

Parameter	Type	Description
sampletime	datetime	Sample time.
accx	integer	Acceleration on X axis.
accx	integer	Acceleration on Y axis.
accz	integer	Acceleration on Z axis.
speed	integer	Speed in km/h. Optional.
enginerpm	integer	Revolutions per minute. Optional.

Parameter	Type	Description
engineload	integer	Engine load in %. Optional.
throttlepos	integer	Throttle position in %. Optional.
latitude	integer	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia. Optional.
longitude	integer	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia. Optional.
country	string	Country code, e.g. “NL”, “AT”, “GB”. Optional.
state	string	Federal state (US only). Optional.
roadspeedlimit	integer	The actual road speed limit taken from the map data in km/h. Optional.
roadtype	integer	Road type classification, can be one of the following. Optional. • 0 - Motorway • 1 - Major road of high importance • 2 - Major road • 3 - Secondary road • 4 - Connecting road • 5 - Local road of major importance • 6 - Local road • 7 - Destination road • 8 - Other road • 9 – Ferry • 10 - Ferry railroad • 11 - Railroad • 12 - Reference line
incity	boolean	Indicates if the vehicle was driving on an urban road (true) or not (false). Optional.

Note: Parameters `roadspeedlimit`, `roadtype`, `incity` are not available by default. Please contact Customer Support to activate it. Additional cost will apply.

getLoadData

Description

`getLoadData` retrieves load data from a trailer for a defined period.

Receving trailer data is supported by LINK 350.

Technical details

SOAP endpoint address / function name	tripAndTimeReportingService#getLoadData
Request limits	10 requests / minute

Parameters

getLoadData requires the following common parameters:

- Authentication parameters
- General parameters
- Date range filter parameters, periods longer than 3 months are not permitted.

Parameters specific to `getLoadData`:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Result

Result for getLoadData:

Parameter	Type	Description
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated.
load	long	Trailer's load in kilograms.
speed	long	Speed in km/h.
timestamp	dateTime	Timestamp of registered load event.

Geocoding and routing

geocodeAddress

Description

Geocodes the address provided as request parameters and returns all possible matches, one per line.

Technical details

SOAP endpoint address / function name	addressService#geocodeAddress
Request limits	900 requests / hour

Validating the response

If the request returns one or multiple matches, the result needs to be validated, e.g. by comparing it to the input parameters, because the result might not be as expected (e.g. due to typing errors) and could result in driving to the wrong destination if used without validation.

Parameters

geocodeAddress requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to geocodeAddress:

Parameter	Type	Description
addrstreet	string (50)	
addrstreetnumber	string (10)	
addrzip	string (10)	
addrcity	string (50)	
addrcountry	string (2)	ISO 3166-1 alpha-2 code for the country (capital letters).

Parameter	Type	Description
freetext	string (512)	Address to geocode in unstructured format. This string can contain all parts of the address separated using spaces. *Example:* `Oosterdoksstraat 114 Amsterdam NL` **Note**: When you use freetext search we recommend not to use the structured parameters. The closer a freetext query is to a structured formatted address the better results you will achieve. Using freetext search you can also search for POIs. This depends on the country and the availablility of POI information.

Result

Result for geocodeAddress:

Parameter	Type	Description
addrstreet	string (50)	
addrstreetnumber	string (10)	
addrzip	string (10)	
addrcity	string (50)	
addrcountry	string (2)	ISO 3166-1 alpha-2 code for the country (capital letters). This parameter is required.
postext	string	
formatted_longitude	string	Geographic longitude in the form GGG° MM′ SS.S″ E/W in the WGS84 coordinate system.
formatted_latitude	string	Geographic latitude in the form GGG° MM′ SS.S″ N/S in the WGS84 coordinate system.
longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.

Parameter	Type	Description
latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
additional_information		Contains the following, separated by a semicolon: `score` - Value between `1` and `0`, where `1` is best score and `0` is worst. It is possible that multiple records are returned with the same score. The score is not a measure of absolute accuracy. If a query returns only approximate matches, the best of those will be given a score of 1.0. `type` - The information returned can give you an idea of which type of match the geocoder has found. Important: Don't rely on the possible value, as they can change without further notice.

calcRouteSimpleExtern

Description

Determines the route from a start location to an end location and calculates the resulting estimated time of arrival for a specific route-type. Optionally IQ Routes and/or HD Traffic information can be included.

Note: If you use this action together with our SOAP interface, please make sure you specify the start time to avoid it being set to the beginning of the day, which is usually already in the past.

Technical details

SOAP endpoint address / function name	addressService#calcRouteSimple
Request limits	6 requests / minute

Parameters

calcRouteSimpleExtern requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to calcRouteSimpleExtern:

Parameter	Type	Description
route_type	nonNegativeInteger	Defines type of route type for calculation of estimated time of arrival. Valid values: • `0` - Quickest • `1` - Shortest • `2` - Avoid motorway • `3` - Walk (deprecated) • `4` - Bicycle (deprecated) Defaults to `0`. This parameter is optional.
use_traffic	nonNegativeInteger	Defines if traffic information needs to be taken into account for calculating the route and estimated time of arrival. Default: • if `start_datetime` is used - `0` • if `start_datetime` is not used - `1` Valid values: • `0` - don't use traffic • `1` - use traffic **Note**: If `use_traffic` is set to 1, the current traffic situation is used to calculate the best route and the estimated time of arrival. This works only if the departure time is less than 30 minutes in the future. This parameter is optional.
start_datetime	dateTime	Defines the departure date and time. If specified, over rules `start_day` and `start_time`. Defaults to the current day and time.
start_day	string	Defines the weekday of departure. If `start_day` is specified it is used together with `start_time`, overruling the default of `start_date-time`. Case-sensitive. Valid values: • `mon` • `tue` • `wed` • `thu` • `fri` • `sat` • `sun` • `today` Defaults to the current day.

Parameter	Type	Description
start_time	string	Defines the departure time. If `start_time` is specified, it is used together with `start_day` overruling the default of `start_datetime`. Defaults to the current time.
use_tollroads	nonNega-tiveInteger	Defines if toll roads should be avoided for and the route calculation. Possible values are: • `0` - Avoid toll roads • `1` - Include toll roads Defaults to `1`.
start_latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia. This parameter is required.
start_longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia. This parameter is required.
end_latitude	int	Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia. This parameter is required.
end_longitude	int	Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia. This parameter is required.

Parameter	Type	Description
travel_mode	string	Determines the travel mode for the calculation of the estimated time of arrival. Valid values: • `car` • `truck` • `bicycle` • `pedestrian` This parameter is optional. Defaults to `car`.

Result

Result for calcRouteSimpleExtern:

Parameter	Type	Description
end_datetime	string	The calculated arrival date and time. Empty if no `start_datetime` is specified and `start_day` or `start_time` is specified.
end_day	string	Valid values: • `mon` • `tue` • `wed` • `thu` • `fri` • `sat` • `sun` Empty if `start_datetime` is specified.
end_time	string	The calculated arrival time. Empty if `start_datetime` is specified.
distance	int	Distance to destination (in meters).
time	string	The calculated duration of a trip from the departure location to the destination location. The unit of measurement is ‘seconds’.
delay	string	Estimated delay for a trip due to traffic. The unit of measurement is ‘seconds’.
timezone_offset	string	Timezone offset. The unit of measurement is ‘seconds’.

Configuration and security

showSettings

Description

`showSettings` shows a number of Webfleet settings.

Currently `showSettings` returns OptiDrive indicator related account settings only. The settings indicate the weight given to each influencing variable used for the calculation of the OptiDrive indicator.

In Webfleet, when you move a slider of a variable under **Settings** in the **Reporting** tab, the given weight in relation to the weight of the other three variables is indicated in brackets.

Technical details

SOAP endpoint address / function name	configurationAndSecurityService#showSettings
Request limits	10 requests / minute

Parameters

showSettings requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to showSettings:

Parameter	Type	Description
target_username	string	The username of the Webfleet user for whom the user specific settings shall be retrieved in addition to the account-wide settings. **Note**: If not specified, settings will be returned on account level. This parameter is optional.

Result

Result for showSettings:

Parameter	Type	Description
od_weight_idling	float	The weight given to idling information used for the calculation of the OptiDrive indicator. The value indicates the total influence of the available information in relation to the other three variables influencing the OptiDrive indicator.

Parameter	Type	Description
od_weight_speeding	float	The weight given to speeding information used for the calculation of the OptiDrive indicator. The value indicates the total influence of the available information in relation to the other three variables influencing the OptiDrive indicator.
od_weight_drivingevents	float	The weight given to information about driving events used for the calculation of the OptiDrive indicator. The value indicates the total influence of the available information in relation to the other three variables influencing the OptiDrive indicator.
od_weight_fuelusage	float	The weight given to fuel consumption information used for the calculation of the OptiDrive indicator. The value indicates the total influence of the available information in relation to the other three variables influencing the OptiDrive indicator.
distanceunit	string	The preferred distance unit set in the application, for example Webfleet user interface. Valid values: • `km` • `mi` **Note**: All values returned by WEBFLEET.connect actions deliver metric values no matter which `distanceunit` is set.
timezone	string	Valid values are known time zones, for example: • `Europe/Berlin` • `Europe/London` • `America/Toronto` • …
block_pwdattack	int	Valid values: • `1` - enabled • `0` - disabled **Note**: If set to `1`, the user's login credential will be blocked after having entered a wrong password multiple times. This is a security feature that protects credentials against password attacks.
order_destination-notification	int	Indicates if a notification is created when the object approaches the order destination. Valid values: • `1` - enabled • `0` - disabled

Parameter	Type	Description
order_destination-notification_min-utes	int	The time before arrival when a notification is created automatically, in minutes.
order_desti-nationnotifica-tion_eventlevel	int	The `eventlevel` of the notification, see [Parameters specific to showEventReportExtern].
order_defaultype	int	Valid values: • `1` - service order • `2` - pickup order • `3` - delivery order
order_rejectional-lowed	int	Valid values: • `1` - enabled • `0` - disabled
fuelconsumption	float	This is the fleet's reference value for fuel consumption reporting in l/100km. May be empty if not used.
fueltype	int	Valid values: • `0` - unknown • `1` - diesel • `2` - gasoline • `3` - lpg
speedlimit	float	This setting is not used anymore.
speedlimit_urban	int	Urban road speed limit on account level used for re-ports, in km/h.
speedlimit_ex-traurban	int	Extra-urban road speed limit on account level used for reports, in km/h.
speedlimit_motor-way	int	Mororway speed limit on account level used for re-ports, in km/h.
speedlimit_toler-ance	int	This value indicates the percentage above the speed limit that vehicles are allowed to travel be-fore Webfleet records that the vehicle is breaking the speed limit. The values range from 0 ... 100.
od_weight_coast-ing	float	The weight given to coasting information used for the calculation of the OptiDrive indicator. The value indicates the total influence of the available information in relation to the other variables influenc-ing the OptiDrive indicator.

Parameter	Type	Description
od_weight_constantspeed	float	The weight given to constant speed information used for the calculation of the OptiDrive indicator. The value indicates the total influence of the available information in relation to the other variables influencing the OptiDrive indicator.
od_weight_greenspeed	float	The weight given to green speed information used for the calculation of the OptiDrive indicator. The value indicates the total influence of the available information in relation to the other variables influencing the OptiDrive indicator.
od_weight_highrevving	float	The weight given to gear shifting (high revving) information used for the calculation of the OptiDrive indicator. The value indicates the total influence of the available information in relation to the other variables influencing the OptiDrive indicator.
energy_consumption	float	This is the fleet's reference value for energy consumption reporting in kWh/100km. May be empty if not used.

createSession

Description

Using createSession a session on the server will be created. The result structure contains a `sessiontoken`, also see Authentication parameters. For the lifetime of the session that is usually ~60 minutes, this token can be used alternatively to `account`, `username` or `password` to authorise further requests to WEBFLEET.connect.

Technical details

SOAP endpoint address / function name	configurationAndSecurityService#createSession
Request limits	10 requests / hour

Parameters

createSession requires the following common parameters:

- Authentication parameters
- General parameters

Result

Result for createSession:

Parameter	Type	Description
sessiontoken	string (32)	A server-generated alphanumeric random string. Within the session lifetime, the Authentication parameters `account`, `username` and `password` can be omitted for further requests. Instead, only the new request parameter `sessiontoken` is needed to identify the user's session. **Note**: `sessiontoken` is bound to the IP address of the client, that requested this `sessiontoken`. Therefore the `sessiontoken` cannot be used from other computers.
timetolive	nonNegativeInteger	Maximum lifetime of the session in seconds. After this time, the `sessiontoken` becomes invalid, no matter if requests happened in the meantime. Typical value: `3600` (60 minutes)
timetoidle	nonNegativeInteger	Maximum time between two requests in seconds. The idle-timeout is used by the server to detect unused sessions. Typical value: `3600` (60 minutes)
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. This value is returned, when the user is assigned to one object only. Can be used alternatively to `objectno`.
externalid	string	For future use.
passwordexpiration	dateTime	The expiration time of the password. ISO 8601-formatted date and time in UTC timezone, combined representation in the extended format. *Example: 2013-06-24T16:00:00Z* Empty if the setting **password expiration** is disabled or no timestamp of the last password change exists.

terminateSession

Description

Using `terminateSession` you can invalidate a session that was created using createSession. After that you can create a new session.

Technical details

SOAP endpoint address / function name	configurationAndSecurityService#terminateSession

Request limits	10 requests / minute

Parameters

terminateSession requires the following common parameters:

- Authentication parameters, `sessiontoken` is required
- General parameters

Note: `terminateSession` does not require any specific parameters. You only need to indicate sessiontoken of the session that you want to invalidate, see Authentication parameters.

showAccountOrderStates

Description

`showAccountOrderStates` retrieves the list of order states and their properties. The properties indicate if the respective order state appears in the order workflow and whether a notification shall be created when the respective order state was reported. These settings apply to all newly created orders within the Webfleet account.

The settings of properties of order states correspond to the settings you can make in the Webfleet user interface.

Technical details

SOAP endpoint address / function name	configurationAndSecurityService#showAccountOrderStates
Request limits	10 requests / minute

Parameters

showAccountOrderStates requires the following common parameters:

- Authentication parameters
- General parameters

Result

Result for showAccountOrderStates:

Parameter	Type	Description
orderstate	int	The state of the progress of an order. Valid values: • `101` - Received • `102` - Read • `103` - Accepted • `201` - Service order started • `202` - Arrived at destination • `203` - Work started • `204` - Work finished • `205` - Departed from destination • `207` - Electronic proof of delivery (ePOD) • `222` - Arrived at pick up location • `223` - Pick up started • `224` - Pick up finished • `225` - Departed from pick up location • `227` - Electronic proof of delivery (ePOD) • `242` - Arrived at delivery location • `243` - Delivery started • `244` - Delivery finished • `245` - Departed from delivery location • `247` - Electronic proof of delivery (ePOD) • `298` - Resumed • `299` - Suspended • `301` - Cancelled • `302` - Rejected • `401` - Finished The following order states are handled according to the settings made for order state `201`: • `221` - Pickup order started • `241` - Delivery order started

Parameter	Type	Description
enabled	int	Indicates whether the respective order state appears in the order workflow on the navigation device. Valid values: • `0` - false • `1` - true The following order states can be enabled or disabled: • `202` - Arrived at destination • `203` - Work started • `204` - Work finished • `205` - Departed from destination • `207` - Electronic proof of delivery (ePOD) • `222` - Arrived at pick up location • `223` - Pick up started • `224` - Pick up finished • `225` - Departed from pick up location • `227` - Electronic proof of delivery (ePOD) • `242` - Arrived at delivery location • `243` - Delivery started • `244` - Delivery finished • `245` - Departed from delivery location • `247` - Electronic proof of delivery (ePOD)
eventlevel	int	Event level. Valid values: • `<empty>` - No event is created. • `2` - Warning

updateAccountOrderState

Description

`updateAccountOrderState` updates the properties of an order state. This update applies to all newly created orders within an account. Previously created orders are not affected by this change.

Note: If you want to update the properties of multiple order states for the whole account you have to update each order state separately.

Technical details

SOAP endpoint address / function name	configurationAndSecurityService#updateAccountOrderState
Request limits	50 requests / hour

Parameters

updateAccountOrderState requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to updateAccountOrderState:

Parameter	Type	Description
orderstate	int	The state of the progress of an order. Valid values: • `101` - Received • `102` - Read • `103` - Accepted • `201` - Service order started • `202` - Arrived at destination • `203` - Work started • `204` - Work finished • `205` - Departed from destination • `207` - Electronic proof of delivery (ePOD) • `222` - Arrived at pick up location • `223` - Pick up started • `224` - Pick up finished • `225` - Departed from pick up location • `227` - Electronic proof of delivery (ePOD) • `242` - Arrived at delivery location • `243` - Delivery started • `244` - Delivery finished • `245` - Departed from delivery location • `247` - Electronic proof of delivery (ePOD) • `298` - Resumed • `299` - Suspended • `301` - Cancelled • `302` - Rejected • `401` - Finished The following order states are handled according to the settings made for order state `201`: • `221` - Pickup order started • `241` - Delivery order started This parameter is required.

Parameter	Type	Description
enabled	int	Indicates whether the respective order state appears in the order workflow on the navigation device. Valid values are: • `0` - false • `1` - true The following order states can be enabled or disabled: • `202` - Arrived at destination • `203` - Work started • `204` - Work finished • `205` - Departed from destination • `207` - Electronic proof of delivery (ePOD) • `222` - Arrived at pick up location • `223` - Pick up started • `224` - Pick up finished • `225` - Departed from pick up location • `227` - Electronic proof of delivery (ePOD) • `242` - Arrived at delivery location • `243` - Delivery started • `244` - Delivery finished • `245` - Departed from delivery location • `247` - Electronic proof of delivery (ePOD) This parameter is required.
eventlevel	int	Event level. Valid values: • `<empty>` • `2` - Warning This parameter is optional.

showAccountOrderAutomations

Description

`showAccountOrderAutomations` retrieves the list of order automation steps. These properties indicate if an order shall be automatically accepted, started, navigated to etc. or if the driver shall be asked to actively confirm the steps. These settings apply to all newly created orders within the Webfleet account.

The settings of the order workflow control correspond to the settings you can make in the Webfleet user interface.

Technical details

SOAP endpoint address / function name	configurationAndSecurityService#showAccountOrderAutomations

Request limits	10 requests / minute

Parameters

showAccountOrderAutomations requires the following common parameters:

- Authentication parameters
- General parameters

Result

Result for showAccountOrderAutomations:

Parameter	Type	Description
automation	int	Workflow step that can be automated. Valid values: • `1` - accept the order • `2` - start the order • `3` - navigate to the order destination • `4` - skip displaying the route summary screen • `5` - delete the order after it has been finished • `6` - suppress the 'continue with next order' screen
enabled	int	Indicates whether the respective step is automated. Valid values: • `0` - false • `1` - true, the step is automated

updateAccountOrderAutomation

Description

`updateAccountOrderAutomation` updates the properties of the workflow steps of orders. This update applies to all newly created orders within an account. Previously created orders are not affected by this change. You can update multiple workflow steps by running the action for each step separately.

Technical details

SOAP endpoint address / function name	configurationAndSecurityService#updateAccountOrderAutomation
Request limits	10 requests / minute

Parameters

updateAccountOrderAutomation requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to updateAccountOrderAutomation:

Parameter	Type	Description
orderautomation	int	Workflow step that can be automated. Valid values: • `1` - accept the order • `2` - start the order • `3` - navigate to the order destination • `4` - skip displaying the route summary screen • `5` - delete the order after it has been finished • `6` - suppress the 'continue with next order' screen Available on PRO devices with firmware version 10.533 or higher. This parameter is required.
enabled	int	Indicates whether the respective step is automated. Valid values: • `0` - false • `1` - true, workflow step is automated This parameter is required.

getAccountStatusMessages

Description

`getAccountStatusMessages` returns the predefined status messages for the Webfleet account. This includes both free text and order related status messages which are visible as predefined messages on the navigation device. By default the units synchronise these account status messages.

For a configuration for individual objects refer to getStatusMessages.

Technical details

SOAP endpoint address / function name	configurationAndSecurityService#getAccountStatusMessages
Request limits	10 requests / minute

Parameters

getAccountStatusMessages requires the following common parameters:

- Authentication parameters
- General parameters

Result

Result for getAccountStatusMessages:

Parameter	Type	Description
type	string	Indicates the type of status message. Valid values: • `text` • `order`
index	nonNegativeInteger	The index of the predefined message Range: 1 ... 15
text	string	The text of the predefined status message.
eventlevel	nonNegativeInteger	The level of the reported event. Valid values: • <empty> - just the message • `1` - Notice/information • `2` - Warning • `3` - Alarm 1 • `4` - Alarm 2 • `5` - Alarm 3

getStatusMessages

Description

`getStatusMessages` returns the predefined status messages for a single object. This applies to both free text and order related status messages which are visible as predefined messages on the navigation device. By default these messages are synchronised with the status messages in the Webfleet account.

For account-wide configuration please refer to getAccountStatusMessages.

Technical details

SOAP endpoint address / function name	configurationAndSecurityService#getStatusMessages
Request limits	10 requests / minute

Parameters

getStatusMessages requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getStatusMessages:

Parameter	Type	Description
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. If `objectno` is not indicated this parameter is mandatory.
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. If `objectuid` is not indicated this parameter is mandatory.

Result

Result for getStatusMessages:

Parameter	Type	Description
type	string	Indicates the type of status message. Valid values: • `text` • `order`
index	nonNegativeInteger	The index of the predefined message. Range: 1 ... 15
text	string	The text of the predefined status message.
eventlevel	nonNegativeInteger	The level of the reported event. Valid values: • `<empty>` - just the message • `1` - Notice/information • `2` - Warning • `3` - Alarm 1 • `4` - Alarm 2 • `5` - Alarm 3

setVehicleConfig

Description

Using `setVehicleConfig` you can change one or more LINK specific configurations at once.

Technical details

SOAP endpoint address / function name	configurationAndSecurityService#setVehicleConfig

Request limits	30 requests / hour

Parameters

setVehicleConfig requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to setVehicleConfig:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Indicate either `objectno` or `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Indicate either `objectno` or `objectuid`.
ign_trip-start_timeout	int	Timeout before starting the trip automatically after the device detects the ignition is switched on, in seconds. Valid values are: 0, 30, 60, 120, 180, 240, 300 This parameter is optional.
ign_tripstop_timeout	int	Timeout before ending the trip automatically when the device detects standstill or is undocked. Valid values are: 0, 60, 120, 180, 240, 300, 600, 900, 1200, 1800, 2700, 3600 This parameter is optional.
standstill_timeout	int	Minimal standstill duration. Valid values are: 60, 120, 180, 240, 300, 600, 900, 1200, 1800, 2700, 3600 This parameter is optional.

getVehicleConfig

Description

Using `getVehicleConfig` you can retrieve LINK specific configuration settings for individual vehicles or all vehicles in the Webfleet account.

Technical details

SOAP endpoint address / function name	configurationAndSecurityService#getVehicleConfig
Request limits	10 requests / minute

Parameters

getVehicleConfig requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getVehicleConfig:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. This parameter is optional.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. This parameter is optional.

Result

Result for getVehicleConfig:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated.
ign_trip-start_timeout	int	Timeout before starting the trip automatically after the device detects the ignition is switched on, in seconds. Valid values are: 0, 30, 60, 120, 180, 240, 300
ign_tripstop_timeout	int	Timeout before ending the trip automatically when the device detects standstill or is undocked. Valid values are: 0, 60, 120, 180, 240, 300, 600, 900, 1200, 1800, 2700, 3600
standstill_timeout	int	Minimal standstill duration. Valid values are: 60, 120, 180, 240, 300, 600, 900, 1200, 1800, 2700, 3600

setStatusMessages

Description

`setStatusMessages` sets predefined text and order status messages for a specific object. These messages are visible as predefined messages on the driver terminal.

Setting predefined status messages for the Webfleet account can be done using setAccountStatusMessages.

Technical details

SOAP endpoint address / function name	configurationAndSecurityService#setStatusMessages
Request limits	50 request / 8 hours

Parameters

setStatusMessages requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to setStatusMessages:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Indicate either `objectno` or `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Indicate either `objectno` or `objectuid`.
resettoaccount	boolean	Valid values: • `1` - The object specific status messages are reset to the account defaults and the status messages will be used from the Webfleet account • `0` - Status messages remain object specific. This parameter is optional.
ts_txt_1 ts_txt_2 ... ts_txt_15	string (60)	The text of the text status message on index 1, 2 ... 15. These parameters are optional.
ts_el_1 ts_el_2 ... ts_el_15	nonNegativeInteger	The text of the event level on index 1, 2 ... 15. Valid values: • `1` - Notice/Information • `2` - Warning • `3` - Alarm 1 • `4` - Alarm 2 • `5` - Alarm 3 These parameters are optional. **Note**: If not specified, the status message is not associated with any event level.

Parameter	Type	Description
os_txt_1 os_txt_2 ... os_txt_15	string (60)	The order status message text on index 1, 2 ... 15. These parameters are optional.
os_el_1 os_el_2 ... os_el_15	nonNegativeInteger	The event level of the order status message on index 1, 2 ... 15. Valid values: • `1` - Notice/Information • `2` - Warning • `3` - Alarm 1 • `4` - Alarm 2 • `5` - Alarm 3 These parameters are optional. **Note**: If not indicated the status message is not associated with any event level.

setAccountStatusMessages

Description

`setAccountStatusMessages` sets the predefined order and text status messages for a whole Webfleet account. For object specific configurations use setStatusMessages.

Technical details

SOAP endpoint address / function name	configurationAndSecurityService#setAccountStatusMessages
Request limits	1 request / 4 hours

Parameters

setAccountStatusMessages requires the following common parameters:

- **Authentication parameters**
- **General parameters**

Parameters specific to setAccountStatusMessages:

Parameter	Type	Description
ts_txt_1 ts_txt_2 ... ts_txt_15	string (60)	The text of the text status message on index 1, 2 ... 15. These parameters are optional.

Parameter	Type	Description
ts_el_1 ts_el_2 ... ts_el_15	nonNega- tiveInteger	The text of the event level on index 1, 2 ... 15. Possible values for event levels are: • `1` - Notice/Information • `2` - Warning • `3` - Alarm 1 • `4` - Alarm 2 • `5` - Alarm 3 These parameters are optional. If not indicated the status message is not associated with any event level.
os_txt_1 os_txt_2 ... os_txt_15	string (60)	The order status message text on index 1, 2 ... 15. These parameters are optional.
os_el_1 os_el_2 ... os_el_15	nonNega- tiveInteger	The event level of the order status message on index 1, 2 ... 15. Possible values for event levels are: • `1` - Notice/Information • `2` - Warning • `3` - Alarm 1 • `4` - Alarm 2 • `5` - Alarm 3 These parameters are optional. If not indicated the status message is not associated with any event level.

User management

showUsers

Description

This actions returns a list of all existing users within the account along with the last recorded login time.

Technical details

SOAP endpoint address / function name	userManagementService#showUsers
Request limits	10 requests / minute

Parameters

showUsers requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to showUsers:

Parameter	Type	Description
username_filter	string (max. 50 characters)	Filter used to match any user name in the account containing the indicated string, also as substring. This parameter is optional.
realname_filter	string (max. 50 characters)	Filter used to match any real name of a user in the account containing the indicated string, also as substring. This parameter is optional.
company_filter	string (max. 50 characters)	Filter used to match any company name in the account containing the indicated string, also as substring. This parameter is optional.

Result

Result for showUsers:

Parameter	Type	Description
username	string (50 characters)	

Parameter	Type	Description
realname	string (50 characters)	The real name of a user
company	string (50 characters)	
email	string (255)	The e-mail address of the user. This parameter is optional.
validfrom	dateTime	ISO 8601-formatted date and time in the UTC time-zone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00 Empty if no start date is set.
validto	dateTime	ISO 8601-formatted date and time in the UTC time-zone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00 Empty if no end date is set.
lastlogin	dateTime	ISO 8601-formatted date and time in the UTC time-zone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00 Empty if no end date is set.
profile	string (50 characters)	The profile of the user. Valid values: • `guest_login_only` • `standard_login_only` • `guest` • `standard` • `expert` • `admin`
profilename	string	The label used for the value of profile. Returned in the language of the request.
userinfo	string (4000 characters)	Additional information about the user.

Parameter	Type	Description
passwordexpiration	dateTime	The expiration time of the password. ISO 8601-formatted date and time in the UTC time-zone, combined representation in the extended format. Example: 2007-12-24T16:00:00+02:00 Empty if the setting **password expiration** is disabled or no timestamp of the last password change exists.
useruid	string (30)	A unique unchangeable identifier for the user which is automatically generated by Webfleet.

changePassword

Description

Using `changePassword` you can change the password of your own user account.

Technical details

SOAP endpoint address / function name	userManagementService#changePassword
Request limits	10 requests / hour

Parameters

changePassword requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to changePassword:

Parameter	Type	Description
oldpassword	string	The current password of your user account. This parameter is mandatory.
newpassword	string	The new password. This parameter is mandatory.

insertUser

Description

Using `insertUser` you can create a new Webfleet user within the current account.

Note: This action can only be executed by users that have the 'Administrator' profile.

Technical details

SOAP endpoint address / function name	userManagementService#insertUser
Request limits	10 requests / minute

Parameters

insertUser requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to insertUser:

Parameter	Type	Description
new_username	string (50)	User name of the newly created user. This parameter is mandatory. It must be unique within the Webfleet account.
new_password	string (255)	The password of the newly created user. You can leave this empty to let Webfleet create a new random password. The generated password will be sent to the email address of the newly created user. This parameter is optional.
require_password_change	boolean	If this parameter is set to `true`, the new user is forced to change the password when they log on to Webfleet for the first time. This parameter affects the Webfleet user interface only, but not WEBFLEET.connect. Default is `true`. This parameter is optional.
realname	string (50)	The full name of the user. This parameter is mandatory.
info	string (500)	A descriptive text. This parameter is optional.
company	string (50)	The name of the company. This parameter is optional.
email	string (255)	The e-mail address of the user. This parameter is mandatory.

Parameters specific to insertUser

Parameter	Type	Description
validfrom	datetime	Defines the start date of the validity of the user account. You can omit this parameter if not needed. Default is null. This parameter is optional.
validto	datetime	Defines the end date of the validity of the user account. You can omit this parameter if not needed. Default is null. This parameter is optional.
profile	string (50)	The profile of the user, see showUsers. Valid values: • `guest_login_only` • `guest` • `standard_login_only` • `standard` • `expert` • `admin` • `invoices_user` • `report_recipient` This parameter is required. See below table for possible combinations with `interfacestyle`.
interfacestyle	string (50)	The interface style of the user which is used by the Webfleet user interface. The default value is 'standard'. Valid values are: • `standard` • `bigmap` • `invoices` • `reportrecipient` This parameter is optional. See below table for possible combinations with `profile`.

Parameters specific to insertUser

Profile and interface style

The following combinations of `profile` and `interfacestyle` are possible:

	standard	bigmap	invoices	reportrecipient
guest_login_only	x			
guest	x			

	standard	bigmap	invoices	reportrecipient
standard_login_only	x	x		
standard	x			
export	x			
admin	x			
invoices_user			x	
report_recipient				x

updateUser

Description

Using `updateUser` you can update the details of a Webfleet user within the current account.

Note: This action can only be executed by users that have the 'Administrator' profile.

Technical details

SOAP endpoint address / function name	userManagementService#updateUser
Request limits	10 requests / minute

Parameters

updateUser requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to updateUser:

Parameter	Type	Description
target_useruid	string (30)	A unique, unchangeable identifier for the user, automatically generated by Webfleet. The `useruid` is returned by showUsers. It is mandatory to specify `target_useruid` and/or `target_username`.
target_username	string (50)	User name of the user. It is mandatory to specify `target_useruid` and/or `target_username`.
new_username	string (50)	User name of the newly created user. It is required only if the user name shall be changed. This parameter is optional.

Parameter	Type	Description
new_password	string (255)	Defines a new password for the user. You can omit this parameter to not change the password. This parameter is optional.
generate_password	boolean	Generates a password for the user. If set to `true` a new random password will be generated by Webfleet. The password will be sent to the email address of the user. Omit this parameter to not change the password. This parameter is optional.
require_password_change	boolean	If this parameter is set to `true`, the user is forced to change the password when they log on to Webfleet for the first time. This parameter affects the Webfleet user interface only, but not WEBFLEET.connect. This parameter is optional.
realname	string (50)	The full name of the user. This parameter is optional.
info	string (500)	A descriptive text. This parameter is optional.
company	string (50)	The name of the company. This parameter is optional.
email	string (255)	The e-mail address of the user. This parameter is optional.
validfrom	datetime	Defines the start date of the validity of the user account. You can omit this parameter if not needed. Default is null. This parameter is optional.
validto	datetime	Defines the end date of the validity of the user account. You can omit this parameter if not needed. Default is null. This parameter is optional.

Parameter	Type	Description
profile	string (50)	The profile of the user, see showUsers. Valid values: • `guest_login_only` • `guest` • `standard_login_only` • `standard` • `expert` • `admin` • `invoices_user` • `report_recipient` This parameter is optional. See table for possible combinations with `interfaces-tyle`, insertUser.
interfacestyle	string (50)	The interface style of the user which is used by the Webfleet user interface. The default value is 'standard'. Valid values are: • `standard` • `bigmap` • `invoices` • `reportrecipient` This parameter is optional. See table for possible combinations with `profile`, insertUser.

deleteUser

Description

Using `deleteUser` you can delete a Webfleet user within the current account.

Note: This action can only be executed by users that have the 'Administrator' profile.

Technical details

SOAP endpoint address / function name	userManagementService#deleteUser
Request limits	10 requests / minute

Parameters

deleteUser requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to deleteUser:

Parameter	Type	Description
target_username	string (50)	User name of the user. It is mandatory to specify `target_useruid` or `target_username`.
target_useruid	string (30)	A unique, unchangeable identifier for the user, automatically generated by Webfleet. The `useruid` is returned by showUsers. It is mandatory to specify `target_useruid` or `target_username`.

getUserRights

Description

This action returns the currently configured access right levels for a specified user. The result contains profile default rights and individually configured rights.

Note: This action can only be executed by users that have the 'Administrator' profile.

Technical details

SOAP endpoint address / function name	userManagementService#getUserRights
Request limits	10 requests / minute

Parameters

getUserRights requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getUserRights:

Parameter	Type	Description
target_username	string (50)	User name of the user. It is mandatory to indicate target_useruid `and/or` `target_username`.
target_useruid	string (30)	A unique, unchangeable identifier for the user, automatically generated by Webfleet. The `useruid` is returned by showUsers. It is mandatory to specify `target_useruid` and/or `target_username`.

Result

Result for getUserRights:

Parameter	Type	Description
rightlevel	string	The name of the right level, see List of supported right levels.
entityuid	string	The UID of an entity the right applies to. Can be empty for global rights.
entitytype	string	The type of entity the right applies to. Valid values: • `object` • `objectgroup` • `driver` • `drivergroup` • `address` • `addressgroup`

setUserRight

Description

This action adds a right level to an individual user. Right levels can only be changed for users with the `standard` or `bigmap` interface style (`interfacestyle`). For users with the `bigmap` interface style only the right levels `address_read_access` and `object_tracking` can be set/removed in combination with an `entityuid`.

Note: This action can only be executed by users that have the 'Administrator' profile.

Technical details

SOAP endpoint address / function name	userManagementService#setUserRight
Request limits	10 requests / minute

Parameters

setUserRight requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to setUserRight:

Parameter	Type	Description
target_username	string (50)	User name of the user. It is mandatory to specify `target_useruid` and/or `target_username`.

Parameter	Type	Description
target_useruid	string (30)	A unique, unchangeable identifier for the user, automatically generated by Webfleet. The `useruid` is returned by showUsers. It is mandatory to specify `target_useruid` and/or `target_username`.
rightlevel	string	The name of the right level, see List of supported right levels.
entityuid	string	The UID of the entity the right applies to. **Note**: You can only assign right levels to entities when they are supported by the `entitytype` of the indicated `entityuid`. For example, you cannot assign the right level `object_full_access` to an entity of the entity type `driver`. Omit this parameter for global rights.

resetUserRights

Description

Using `resetUserRights` you can reset the user access right levels to the profile defaults. All individual configured rights will be lost after executing this function.

Note: This action can only be executed by users that have the 'Administrator' profile.

Technical details

SOAP endpoint address / function name	userManagementService#resetUserRights
Request limits	10 requests / minute

Parameters

resetUserRights requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to resetUserRights:

Parameter	Type	Description
target_username	string (50)	User name of the user. It is mandatory to specify `target_useruid` and/or `target_username`.

Parameter	Type	Description
target_useruid	string (30)	A unique, unchangeable identifier for the user, automatically generated by Webfleet. The `useruid` is returned by showUsers. It is mandatory to specify `target_useruid` and/or `target_username`.

removeUserRight

Description

This action removes a right level from an individual user. Right levels can only be removed from users with the `standard` or `bigmap` interface style (`interfacestyle`). For users with the `bigmap` interface style only the right levels `address_read_access` and `object_tracking` can be removed in combination with an `entityuid`.

Note: This action can only be executed by users that have the 'Administrator' profile.

Technical details

SOAP endpoint address / function name	userManagementService#removeUserRight
Request limits	10 requests / minute

Parameters

removeUserRight requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to removeUserRight:

Parameter	Type	Description
target_username	string (50)	User name of the user. It is mandatory to specify `target_useruid` and/or `target_username`.
target_useruid	string (30)	A unique, unchangeable identifier for the user, automatically generated by Webfleet. The `useruid` is returned by showUsers. It is mandatory to specify `target_useruid` and/or `target_username`.
rightlevel	string	The name of the right level, see List of supported right levels.

Parameter	Type	Description
entityuid	string	The UID of the entity the right applies to. **Note**: You can only remove right levels from entities when they are supported by the `entitytype` of the indicated `entityuid`. For example, you cannot remove the right level `object_full_access` from an entity of the entity type `driver`. Omit this parameter for global rights.

Supported right levels

List of supported right levels

Some of the right levels can be set only on global basis as where other right levels can be set on individual basis, per entity, or on a global basis. In this context, 'entity' means a vehicle, address, or driver (or associated group). If a right level it set to global, all associated entities are affected.

List of supported right levels

Parameter	Basis	Description
external_access	Global	Allowed to access WEBFLEET.connect API.
full_access_accountsettings	Global	Allowed to change account settings.
edit_usersettings	Global	User can change own settings.
change_password	Global	User can change his own password.
trip_data_access	Global	Access to trip related data.
areas_full_access	Global	Read/Write access to areas.
areas_read_access	Global	Read access to areas.
orders_full_access	Global	Full access to order management functions.
orders_read_access	Global	Read access to orders.
read_access_orders_when_vehicle_assigned	Global	Read access to orders which have a vehicle assigned.
full_access_orders_when_vehicle_assigned	Global	Full access to orders which have a vehicle assigned.
external_tachograph_interface	Global	Access to the tachograph page

Parameter	Basis	Description
object_full_access	Entity Global	Full read/write access to vehicles, messaging, tracking, change group association etc.
object_expert_access	Entity Global	Full read/write access to vehicles, messaging, tracking
object_standard_access	Entity Global	Read access to vehicles, messaging, tracking
object_tracking	Entity Global	Read access to vehicles, tracking, read messages
object_locating	Entity Global	Read access to vehicles, view position
object_locating_and_messaging	Entity Global	Read access to vehicles, view position, messaging
object_locating_and_communication	Entity Global	Read access to vehicles, view position, messaging
object_messaging	Entity Global	Read access to vehicles, messaging, no position info
object_read_access	Entity Global	Read access to vehicles, view position
address_admin_access	Entity Global	Full edit access to addresses including address group assignments
address_edit_access	Entity Global	Edit access to addresses
address_read_access	Entity Global	Read access to addresses
driver_admin_access	Entity Global	Full edit access to drivers, including driver group assignments
driver_edit_access	Entity Global	Edit access to drivers
driver_read_access	Entity Global	Read access to drivers
enable_device_configuration_management	Global	Allows to manage configurations for mobile devices. This right level is only available for specific users.
report_read_access	Global	Grants read access to reports.

Parameter	Basis	Description
report_edit_access	Global	Allows to edit and create reports. Only available when the right level `report_read_access` is set.
camera_read_access	Global	Grants read access to Webfleet Video.
camera_full_access	Global	Grants full access to Webfleet Video.

Vehicle Maintenance

insertMaintenanceSchedule

Description

This action creates a new maintenance schedule and task for a specific object. When you have created a new maintenance schedule, an ID for it will be returned.

Technical details

SOAP endpoint address / function name	vehicleMaintenanceService#insertMaintenanceSchedule
Request limits	10 requests / minute

Parameters

insertMaintenanceSchedule requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to insertMaintenanceSchedule:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
schedulename	string (100)	The name of the maintenance schedule. This parameter is required.
scheduledescription	string (2000)	Description of the maintenance schedule. This parameter is optional.

Parameter	Type	Description
scheduletype	nonNegativeInteger	Indicates the icon for the specific maintenance schedule shown in Webfleet. This parameter is required. Valid values: • `0` - Renew Oil • `1` - Change tires • `2` - Check engine • `3` - General • `4` - License • `5` - Brake disc • `6` - Gas • `7` - Lights • `8` - Air condition • `9` - Battery • `10` - Inspection • `11` - Car lift • `12` - Steering wheel • `13` - Key • `14` - Air bag • `15` - Car jack • `16` - Gears • `17` - Water • `18` - Voltage • `19` - ABS • `20` - Heater • `21` - Handbrake • `22` - Parking • `23` - Temperature • `24` - Wiper • 25 - Diagnostic trouble code • 26 - CAN warning

Parameter	Type	Description
ruletype	nonNegativeInteger	Describes the rules if and how a maintenance task shall be repeated. This parameter is required. Valid values: • `0` - Single shot - Task will not be repeated. • `1` - Continuous fixed interval - Task will be repeated in a fixed interval that was reported at the end of the previous task, independent from mileage or time when the task was carried out. • `2` - Continuous sequential interval - Task will be repeated in an interval depending on the state of the last completed task. *Example: A Task will be repeated after 2000 km driving after the previous maintenance task has ended.*
intervaltime	nonNegativeInteger	Indicates the interval in months. Valid values: `1` … `120` This parameter is required if `intervalodometer` is missing and `ruletype` is `1` or `2`.
intervalodometer	nonNegativeInteger	Indicates the interval in meters. Valid values: `1` … `1,000,000,000` This parameter is required if `intervaltime` is missing and `ruletype` is `1` or `2`.
remindingtime	nonNegativeInteger	Indicates the time a reminder shall show up in Webfleet previous to the due date, in days. Valid values: `1` … `180` This parameter is optional.
remindingodometer	nonNegativeInteger	Indicates the mileage a reminder shall show up in Webfleet previous to the mileage when the task is due, in meters. Valid values: `1` … `100,000,000` This parameter is optional.
plannedexectime	date	Indicates the next due date of the maintenance work. This parameter is required if `plannedexecodometer` is missing.
plannedexecodometer	nonNegativeInteger	Indicates the mileage after which the maintenance is due, in meters. Valid values: 1 … 10,000,000,000 This parameter is required if `plannedexectime` is missing.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Parameter	Type	Description
intervaltimetype	nonNegativeInteger	The type of time interval. Valid values: • `0` - days • `1` - weeks • `2` - months • `3` - years

Result

Result for insertMaintenanceSchedule:

Parameter	Type	Description
scheduleid	nonNegativeInteger	Maintenance schedule specific identifier. Unique within an account.

updateMaintenanceSchedule

Description

This action updates an existing maintenance schedule by using `scheduleid`.

When using `updateMaintenanceSchedule`, you only need to include the parameters for which you want to change or delete the values. If you include a parameter and do not indicate a value, the existing value will be deleted.

To delete a numeric value in SOAP use the `setNull` member.

Technical details

SOAP endpoint address / function name	vehicleMaintenanceService#updateMaintenanceSchedule
Request limits	10 requests / minute

Parameters

updateMaintenanceSchedule requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to updateMaintenanceSchedule:

Parameter	Type	Description
scheduleid	nonNegativeInteger	Maintenance schedule specific identifier. Unique within an account. This parameter is required.

Parameter	Type	Description
schedulename	string (100)	The name of the maintenance schedule. This parameter is optional.
scheduledescrip-tion	string (2000)	Description of the maintenance schedule. This parameter is optional.
scheduletype	nonNega-tiveInteger	Indicates the icon for the specific maintenance sched-ule shown in Webfleet. This parameter is optional. Valid values: • `0` - Renew Oil • `1` - Change tires • `2` - Check engine • `3` - General • `4` - License • `5` - Brake disc • `6` - Gas • `7` - Lights • `8` - Air condition • `9` - Battery • `10` - Inspection • `11` - Car lift • `12` - Steering wheel • `13` - Key • `14` - Air bag • `15` - Car jack • `16` - Gears • `17` - Water • `18` - Voltage • `19` - ABS • `20` - Heater • `21` - Handbrake • `22` - Parking • `23` - Temperature • `24` - Wiper • 25 - Diagnostic trouble code • 26 - CAN warning

Parameter	Type	Description
ruletype	nonNegativeInteger	Describes the rules if and how a maintenance task shall be repeated. This parameter is optional. Valid values: • `0` - Single shot - Task will not be repeated. • `1` - Continuous fixed interval - Task will be repeated in a fixed interval that was reported at the end of the previous task, independant from mileage or time when the task was carried out. • `2` - Continuous sequential interval - Task will be repeated in an interval depending on the state of the last completed task. *Example: Task will be repeated after 2000 km driving after the previous maintenance task has ended.*
intervaltime	nonNegativeInteger	Indicates the interval in months. Valid values: `1` … `120` This parameter is optional. You can only delete the value of the parameter when there is a valid value for `intervalodometer`.
intervalodometer	nonNegativeInteger	Indicates the interval in meters. Valid values: `1` … `1,000,000,000` You can only delete the value of the parameter when there is a valid value for `intervaltime`.
remindingtime	nonNegativeInteger	Indicates the time a reminder shall show up in Webfleet previous to the due date, in days. Valid values: `1` … `180` This parameter is optional.
remindingodometer	nonNegativeInteger	Indicates the mileage a reminder shall show up in Webfleet previous to the mileage when the task is due, in meters. Valid values: `1` … `100,000,000` This parameter is optional.
plannedexectime	date	Indicates the next due date of the maintenance work. This parameter is optional. You can only delete the value of the parameter when there is a valid value for `plannedexecodometer`. **Note**: The parameter must include a time as well as the date.

Parameter	Type	Description
plannedexe-codometer	nonNega-tiveInteger	Indicates the mileage after which the maintenance is due, in meters. Valid values: 1 ... 10,000,000,000 This parameter is optional. You can only delete the value of the parameter when there is a valid value for `plannedexectime`.
intervaltimetype	nonNega-tiveInteger	The type of time interval. Valid values: • `0` - days • `1` - weeks • `2` - months • `3` - years

deleteMaintenanceSchedule

Description

This action deletes an existing maintenance schedule and all related tasks. Tasks that have already been read or completed are not being deleted.

Technical details

SOAP endpoint address / func-tion name	vehicleMaintenanceService#deleteMaintenanceSched-ule
Request limits	10 requests / minute

Parameters

deleteMaintenanceSchedule requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to deleteMaintenanceSchedule:

Parameter	Type	Description
scheduleid	nonNega-tiveInteger	Maintenance schedule specific identifier. Unique within an account. This parameter is required.

showMaintenanceSchedules

Description

This action retrieves a list of maintenance schedules.

Technical details

SOAP endpoint address / function name	vehicleMaintenanceService#showMaintenanceSchedules
Request limits	10 requests / minute

Parameters

showMaintenanceSchedules requires the following common parameters:

- **Authentication parameters**
- **General parameters**

Parameters specific to showMaintenanceSchedules:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. This parameter is optional.
objectgroupname	string	Case-sensitive. This parameter is optional.
scheduleid	nonNegativeInteger	Maintenance schedule specific identifier. Unique within an account. This parameter is optional.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Result

Result for showMaintenanceSchedules:

Parameter	Type	Description
scheduleid	nonNegativeInteger	Maintenance schedule specific identifier. Unique within an account.
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
schedulename	string (100)	The name of the maintenance schedule.
scheduledescription	string (2000)	Description of the maintenance schedule.

Parameter	Type	Description
scheduletype	nonNegativeInteger	Indicates the icon for the specific maintenance schedule shown in Webfleet. Valid values: • `0` - Renew Oil • `1` - Change tires • `2` - Check engine • `3` - General • `4` - License • `5` - Brake disc • `6` - Gas • `7` - Lights • `8` - Air condition • `9` - Battery • `10` - Inspection • `11` - Car lift • `12` - Steering wheel • `13` - Key • `14` - Air bag • `15` - Car jack • `16` - Gears • `17` - Water • `18` - Voltage • `19` - ABS • `20` - Heater • `21` - Handbrake • `22` - Parking • `23` - Temperature • `24` - Wiper • 25 - Diagnostic trouble code • 26 - CAN warning

Parameter	Type	Description
ruletype	nonNegativeInteger	Describes the rules if and how a maintenance task shall be repeated. This parameter is required. Valid values: • `0` - Single shot - Task will not be repeated. • `1` - Continuous fixed interval - Task will be repeated in a fixed interval that was reported at the end of the previous task, independant from mileage or time when the task was carried out. • `2` - Continuous sequential interval - Task will be repeated in an interval depending on the state of the last completed task. *Example: Task will be repeated after 2000 km driving after the previous maintenance task has ended.*
intervaltime	nonNegativeInteger	Indicates the interval in months.
intervalodometer	nonNegativeInteger	Indicates the interval in meters.
remindingtime	nonNegativeInteger	Indicates the time a reminder shall show up in Webfleet previous to the due date, in days.
remindingodometer	nonNegativeInteger	Indicates the mileage a reminder shall show up in Webfleet previous to the mileage when the task is due, in meters.
plannedexectime	date	Indicates the next due date of the maintenance work.
plannedexecodometer	nonNegativeInteger	Indicates the mileage after which the maintenance is due, in meters.
firstexectime	date	Indicates the date when the maintenance task was carried out first.
firstexecodometer	nonNegativeInteger	Indicates the mileage at which the maintenance task was carried out first, in meters.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
creationtype	int	Describes how the maintenance schedule was created. Possible values are: • `0` - manually by user • `1` - automatically by vehicle diagnostic system

Parameter	Type	Description
intervaltimetype	nonNega-tiveInteger	The type of time interval. Valid values: • `0` - days • `1` - weeks • `2` - months • `3` - years

showMaintenanceTasks

Description

This action retrieves a list of maintenance tasks.

Technical details

SOAP endpoint address / function name	vehicleMaintenanceService#showMaintenanceTasks
Request limits	10 requests / minute

Parameters

showMaintenanceTasks requires the following common parameters:

- Authentication parameters
- General parameters

showMaintenanceTasks requires the following other parameters:

- Date range filter parameters
  The date range applies to:
  - `plannedexectime` if `status` has the value 0 or is not set.
  - `readtime` if `status` has the value 1 or is not set.
  - `completedtime` if `status` has the value 2 or is not set.

Parameters specific to showMaintenanceTasks:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. This parameter is optional.
objectgroupname	string	Case-sensitive. This parameter is optional.
scheduleid	nonNega-tiveInteger	Maintenance schedule specific identifier. Unique within an account. This parameter is optional.

Parameter	Type	Description
taskid	nonNegativeInteger	Maintenance task specific identifier. Unique within an account. This parameter is optional.
status	nonNegativeInteger	Status of the task. Valid values: • `0` - new • 1 - read • 2 - completed This parameter is optional.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
duetype	int	Filter for tasks that are overdue or tasks with a exceeded reminder. Valid values: • `0` - no filter • 1 - tasks with exceeded reminder • 2 - tasks that are overdue This parameter is optional.

Parameter	Type	Description
scheduletype	Integer	Indicates the icon for the specific maintenance schedule shown in Webfleet. Valid values: • `0` - Renew Oil • `1` - Change tires • `2` - Check engine • `3` - General • `4` - License • `5` - Brake disc • `6` - Gas • `7` - Lights • `8` - Air condition • `9` - Battery • `10` - Inspection • `11` - Car lift • `12` - Steering wheel • `13` - Key • `14` - Air bag • `15` - Car jack • `16` - Gears • `17` - Water • `18` - Voltage • `19` - ABS • `20` - Heater • `21` - Handbrake • `22` - Parking • `23` - Temperature • `24` - Wiper • 25 - Diagnostic trouble code • 26 - CAN warning

Result

Result for showMaintenanceTasks:

Parameter	Type	Description
taskid	nonNegativeInteger	Maintenance task specific identifier. Unique within an account.
scheduleid	nonNegativeInteger	Maintenance schedule specific identifier. Unique within an account.
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
plannedexectime	date	Indicates the next due date of the maintenance work.

Parameter	Type	Description
plannedexe-codometer	nonNega-tiveInteger	Indicates the mileage after which the maintenance is due, in meters.
remindingtime	date	Indicates the date a reminder shall show up in Webfleet previous to the due date.
remindingodome-ter	nonNega-tiveInteger	Indicates the mileage a reminder shall show up in Webfleet previous to the mileage when the task is due, in meters.
readusername	string	Username of who read the maintenance task.
readtime	date	Time when the maintenance task was read.
readodometer	nonNega-tiveInteger	Vehicle mileage when the maintenance task was read, in meters.
completeduser-name	string	Username of who completed the maintenance task.
completedtime	date	Time when the maintenance task was completed.
completedodome-ter	nonNega-tiveInteger	Vehicle mileage when the maintenance task was com-pleted, in meters.
status	nonNega-tiveInteger	Status of the task. Valid values: • `0` - new • `1` - read • `2` - completed
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
schedulename	string(100)	The name of the maintenance schedule.

Parameter	Type	Description
scheduletype	nonNegativeInteger	Indicates the icon for the specific maintenance schedule shown in Webfleet. Valid values: • `0` - Renew Oil • `1` - Change tires • `2` - Check engine • `3` - General • `4` - License • `5` - Brake disc • `6` - Gas • `7` - Lights • `8` - Air condition • `9` - Battery • `10` - Inspection • `11` - Car lift • `12` - Steering wheel • `13` - Key • `14` - Air bag • `15` - Car jack • `16` - Gears • `17` - Water • `18` - Voltage • `19` - ABS • `20` - Heater • `21` - Handbrake • `22` - Parking • `23` - Temperature • `24` - Wiper
schedulecreationtype	int	Describes how the maintenance schedule was created. Valid values: • `0` - manually by user • `1` - automatically by vehicle diagnostic system

resolveMaintenanceTask

Description

This action resolves a maintenance task depending on the specified status.

Technical details

SOAP endpoint address / function name	vehicleMaintenanceService#resolveMaintenanceTask
Request limits	10 requests / minute

Parameters

resolveMaintenanceTask requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to resolveMaintenanceTask:

Parameter	Type	Description
taskid	nonNegativeInteger	Maintenance task specific identifier. Unique within an account. This parameter is required.
status	nonNegativeInteger	Status of the task. Valid values: • `1` - read • `2` - completed This parameter is required.

getDiagnosticFaults

Description

The `getDiagnosticFaults` action retrieves diagnostic event data for a specified object (e.g., a vehicle or asset). These events may include various technical issues, system warnings, or other relevant conditions affecting the operational status of the object.

Technical details

SOAP endpoint address / function name	vehicleMaintenanceService#getDiagnosticFaults
Request limits	10 requests / minute

Parameters

getDiagnosticFaults requires the following common parameters:

- Authentication parameters
- General parameters
- Date range filter parameters, periods longer than 24 hours are not permitted.

Parameters specific to getDiagnosticFaults:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. This parameter is optional.

Parameter	Type	Description
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Result

Result for getDiagnosticFaults:

Parameter	Type	Description
objectUid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated.
objectName	string	Name of the object
severity	string	Severity level (e.g., info, warning, critical)
event	string	Description of the diagnostic event
eventType	string	Type of event
category	string	Event category
status	string	Status of the event (e.g., active, resolved)
eventTime	dateTime	Timestamp when the event occurred
lastUpdateTime	dateTime	Timestamp of the last update
resolveTime	dateTime	Timestamp when the event was resolved
odometer	long	Vehicle odometer reading
occurrenceCount	long	Number of occurrences
latitude	int	Latitude coordinate of the event
longitude	int	Longitude coordinate of the event
posText	string	Human-readable location description
signalId	string	Identifier of the signal associated with the event

Reporting

getArchivedReportList

Description

`getArchivedReportList` lists information about the reports stored in the Webfleet Reports Archive of a specific user. It returns meta information of the reports such as ID, name, file size, creation time, etc.

It does not fetch the report data (PDF or CSV). Use getArchivedReport to retrieve the actual PDF or CSV files.

Technical details

SOAP endpoint address / function name	reportingService#getArchivedReportList
Request limits	10 requests / minute

Parameters

getArchivedReportList requires the following common parameters:

- Authentication parameters
- General parameters
- Date range filter parameters

Result

Result for getArchivedReportList:

Parameter	Type	Description
reportid	nonNegativeInteger	Identifying number of the report file.
reportname	string (500)	Display name of the report file.
format	string	Name of the file format. Valid values: • `pdf` • `csv`
creationtime	dateTime	Date and time, the report file has been created.
expirytime	dateTime	Date and time of when the report will be deleted automatically.

Parameter	Type	Description
size	nonNegativeInteger	The file size of the report file, in kB.

getArchivedReport

Description

Using `getArchivedReport` you can retrieve a PDF or CSV report that is stored in the Webfleet Reports Archive of a specific user.

Technical details

SOAP endpoint address / function name	reportingService#getArchivedReport
Request limits	10 requests / minute

Parameters

getArchivedReport requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getArchivedReport:

Parameter	Type	Description
reportid	nonNegativeInteger	Identifying number of the report file as returned by getArchivedReportList. This parameter is mandatory.

deleteArchivedReport

Description

Using `deleteArchivedReport` you can delete an archived report file from the Webfleet Reports Archive of a specific user.

Technical details

SOAP endpoint address / function name	reportingService#deleteArchivedReport
Request limits	10 requests / minute

Parameters

deleteArchivedReport requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to deleteArchivedReport:

Parameter	Type	Description
reportid	nonNegativeInteger	Identifying number of the report file as returned by getArchivedReportList. This parameter is required.

getReportList

Description

`getReportList` returns a list of reports, that can be created on demand and that are assigned to the current Webfleet user. Reports can be created on demand using `sendReportViaMail` or `createReport`.

Technical details

SOAP endpoint address / function name	reportingService#getReportList
Request limits	10 requests / minute

Parameters

getReportList requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getReportList:

Parameter	Type	Description
reporttype	string (100)	A type that helps to filter for the report names to be returned. Each report can be associated to several types of reports. This parameter is optional. Omit this parameter to retrieve the complete list of available reports.

Result

Result for getReportList:

Parameter	Type	Description
reportname	string (500)	Display name of the report file.

Parameter	Type	Description
reporttypes	string	Comma-separated list of report types the report is associated with.

createReport

Description

Using `createReport` you can create a new PDF or CSV report on demand.

Technical details

SOAP endpoint address / function name	reportingService#createReport
Request limits	5 requests / 10 minutes

Note: The returned CSV file uses a different style than returned by other WEBFLEET.connect actions. This is because the reports are not created by WEBFLEET.connect but by the Webfleet user interface. Therefore, the general parameter “separator” cannot be used for the CSV separator here either.

Parameters

createReport requires the following common parameters:

- Authentication parameters
- General parameters
- Date range filter parameters, the maximum period length is 12 months.

Parameters specific to createReport:

Parameter	Type	Description
reportname	string (500)	Display name of the report file. This parameter is required.
format	string	Name of the file format. Valid values: • `pdf` • `csv` Defaults to `pdf` if not specified.
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`. This parameter is optional.

Parameter	Type	Description
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`. This parameter is optional.
objectgroupname	string	A name of an object group. This parameter is optional.
driverno	string	Identifying number of a driver. Unique within an account. This parameter is optional.
drivergroupname	string	Name of the group of drivers the driver is assigned to, case-sensitive. This parameter is optional.
orderno	string (20)	Account-unique order id, case-sensitive. This parameter is optional.

sendReportViaMail

Description

Using `sendReportViaMail` you can create a new PDF or CSV report on demand and send it to an indicated email address.

Technical details

SOAP endpoint address / function name	reportingService#sendReportViaMail
Request limits	5 requests / 10 minutes

Parameters

sendReportViaMail requires the following common parameters:

- Authentication parameters
- General parameters
- Date range filter parameters, the maximum period length is 12 months.

Parameters specific to sendReportViaMail:

Parameter	Type	Description
reportname	string (500)	Display name of the report file. This parameter is mandatory.

Parameter	Type	Description
format	string	Name of the file format. Valid values: • `pdf` • `csv` Defaults to `pdf` if not indicated.
email	string	The email address to which the created report will be sent. This parameter is required.
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`. This parameter is optional.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`. This parameter is optional.
objectgroupname	string	A name of an object group. This parameter is optional.
driverno	string	Identifying number of a driver. Unique within an account. This parameter is optional.
drivergroupname	string	Name of the group of drivers the driver is assigned to, case-sensitive. This parameter is optional.
orderno	string (20)	Account-unique order id, case-sensitive. This parameter is optional.

Areas

getAreas

Description

This actions returns a list of all areas that are currently configured within the account.

Technical details

SOAP endpoint address / function name	areaService#getAreas
Request limits	10 requests / minute

Parameters

getAreas requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getAreas

Parameter	Type	Description
areano	string (50)	Identification number of an area. Unique within an account, case-sensitive. **Note**: Indicate either `areano` or `areauid`; do not use both parameters in combination.
areauid	string (30)	A unique, unchangeable identifier for the area, automatically generated by Webfleet. **Note**: Indicate either `areano` or `areauid`; do not use both parameters in combination.
active	boolean	Indicator for whether an area is enabled. Valid values: • `0` —area disabled • `1` —area enabled Omit this parameter to retrieve all areas. This parameter is optional.

Result

Result for getAreas:

Parameter	Type	Description
areano	string (50)	Identification number of an area. Unique within an account, case-sensitive.
areauid	string (30)	A unique, unchangeable identifier for the area, automatically generated by Webfleet.
areaname	string (254)	Display name of the area.
type	int	Classifies the geometric shape of this area. Valid values: • `1` - rectangle • `2` - circle • `3` - polygon • `4` - corridor
active	boolean	Indicator for whether an area is enabled. Valid values are: • `0` - area disabled • `1` - area enabled
validfrom	date	Start date of a period to limit the validity of the area.
validto	date	End date of a period to limit the validity of the area.
notificationmode	int	Describes the type of activity for which events shall be created. Valid values are: • `0` - inside/outside • `1` - enter/leave **Note**: If the area is configured as 'inside/outside', recurrent events will be created as long as the vehicle is located inside/outside the area. If the area is configured as 'enter/leave', only a single event will be created when the vehicle passes the border of the area.
eventlevel_inside	int	Notification level (urgency) of the created event, when the vehicle is located inside of the area. Valid values: • <empty> - No message • `1` - Notice • `2` - Warning • `3` - Alarm 1 • `4` - Alarm 2 • `5` - Alarm 3 Value is only set if `notificationmode` = 0.

Parameter	Type	Description
eventlevel_outside	int	Notification level (urgency) of the created event, when the vehicle is located outside of the area. For valid values refer to the list for `eventlevel_inside`. Value is only set if `notificationmode` = 0.
eventlevel_enter	int	Notification level (urgency) of the created event, when the vehicle enters the area. For valid values refer to the list for `eventlevel_inside`. Value is only set if `notificationmode` = `1`.
eventlevel_leave	int	Notification level (urgency) of the created event, when the vehicle leaves the area. For valid values refer to the list for `eventlevel_inside`. Value is only set if `notificationmode` = `1`.
latitude	int	Geographic latitude of the geometric centre of the shape. Only set if area is a circle or rectangle. For polygon and corridor, please use getAreaPoints to retrieve a list of all coordinates. Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
longitude	int	See description for `latitude` for analogue use for longitude.
radius	int	Radius of the area in meters. Only set if the area is a circle.
width	int	Width of the area in meters. Only set if area is a rectangle or corridor.
height	int	Height of the area in meters. Only set if area is a rectangle.
pointcount	int	The number of points of the area. Only set if area is a polygon or corridor.
color	string (6)	The color of the area as a 6-digit hex RGB-color code.

insertArea

Description

`insertArea` creates a new geographic area.

Technical details

SOAP endpoint address / function name	areaService#insertArea
Request limits	10 requests / minute

Parameters

insertArea requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to insertArea:

Parameter	Type	Description
areano	string (50)	Identification number of an area. Unique within an account, case-sensitive. This parameter is required.
areaname	string (254)	Display name of the area. This parameter is required.
type	int	Classifies the geometric shape of this area. Valid values: • `1` – rectangle • `2` – circle • `3` – polygon • `4` – corridor This parameter is required.
active	boolean	Indicator for whether an area is enabled. Valid values: • `0` – area disabled • `1` – area enabled Default value is `1`. This parameter is optional.
validfrom	date	Start date of a period to limit the validity of the area. This parameter is optional.
validto	date	End date of a period to limit the validity of the area. This parameter is optional.
notificationmode	int	Describes the type of activity for which events shall be created. The value defaults to 1 (enter/leave). When the vehicle passes the border of area one event is created. This parameter is optional.

Parameter	Type	Description
eventlevel_enter	int	Notification level (urgency) of the created event, when the vehicle enters the area. Valid values: • `<empty>` – No message • `1` – Notice • `2` – Warning • `3` – Alarm 1 • `4` – Alarm 2 • `5` – Alarm 3 Value is only set if `notificationmode` = `1`. Default value is <empty>. This parameter is optional.
eventlevel_leave	int	Notification level (urgency) of the created event, when the vehicle leaves the area. Valid values: • `<empty>` – No message • `1` – Notice • `2` – Warning • `3` – Alarm 1 • `4` – Alarm 2 • `5` – Alarm 3 Value is only set if `notificationmode` = `1`. Default value is <empty>. This parameter is optional.
latitude	int	Geographic latitude of the geometric centre of the shape (rectangle or circle). Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia. Mandatory for rectangle and circle. For polygon and corridor, please use the parameter `point` multiple times to submit each point of the complex shape.
longitude	int	See description for `latitude` for analogue use for longitude.
radius	int	Radius of the area in meters. Only set if the area is a circle. This parameter is mandatory for circle.

Parameter	Type	Description
width	int	Width of the area in meters. Only set if area is a rectangle or corridor. This parameter is mandatory for rectangle and corridor.
height	int	Height of the area in meters. Only set if area is a rectangle. This parameter is mandatory for rectangle.
point	string	Comma-separated `latitude/longitude` pair, which describes a single point of the polygon or corridor (in WGS84 notation). `<latitude>,<longitude>` Repeat this parameter for each point in the shape. **Note**: Use HTTP-POST if the generated request URL is longer than 2000 characters. This parameter is mandatory for polygon and corridor.
color	string (6)	The color of the area as a 6-digit hex RGB-color code. This parameter is optional. The default value is 3366FF. The following colors are available within the Webfleet GUI:

deleteArea

Description

`deleteArea` deletes an area. All possibly existing vehicle assignments and schedules for this area will be deleted, too.

Technical details

SOAP endpoint address / function name	areaService#deleteArea
Request limits	10 requests / minute

Parameters

deleteArea requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to deleteArea:

Parameter	Type	Description
areano	string (50)	Identification number of an area. Unique within an account, case-sensitive. **Note**: Specifiy either `areano` or `areauid`; do not use both parameters in combination.
areauid	string (30)	A unique, unchangeable identifier for the area, automatically generated by Webfleet. **Note**: Specify either `areano` or `areauid`; do not use both parameters in combination.

updateArea

Description

`updateArea` updates the details of an existing geographic area.

Technical details

SOAP endpoint address / function name	areaService#updateArea
Request limits	10 requests / minute

Parameters

updateArea requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to updateArea

Parameter	Type	Description
areano	string (50)	Identification number of an area. Unique within an account, case-sensitive. **Note**: It is mandatory to indicate at least areano `or` `areauid` to identify the area that needs to be updated. Indicate both when `areano` shall be updated.
areauid	string (30)	A unique, unchangeable identifier for the area, automatically generated by Webfleet. **Note**: It is mandatory to indicate at least `areano` or `areauid` to identify the area that needs to be updated. Indicate both when `areano` shall be updated.
areaname	string (254)	Display name of the area. This parameter is optional.
type	int	Classifies the geometric shape of this area. Valid values: • `1` - rectangle • `2` - circle • `3` - polygon • `4` - corridor This parameter is optional.
active	boolean	Indicator for whether an area is enabled. Valid values: • `0` - area disabled • `1` - area enabled This parameter is optional.
validfrom	date	Start date of a period to limit the validity of the area. This parameter is optional.
validto	date	End date of a period to limit the validity of the area. This parameter is optional.
notificationmode	int	Describes the type of activity for which events shall be created. The value defaults to 1 (enter/leave). When the vehicle passes the border of area one event is created. This parameter is optional.

Parameter	Type	Description
eventlevel_enter	int	Notification level (urgency) of the created event, when the vehicle enters the area. Valid values: • `<empty>` – No message • `1` – Notice • `2` – Warning • `3` – Alarm 1 • `4` – Alarm 2 • `5` – Alarm 3 Value is only set if `notificationmode` = `1`. Default value is <empty>. This parameter is optional.
eventlevel_leave	int	Notification level (urgency) of the created event, when the vehicle leaves the area. Valid values: • `<empty>` – No message • `1` – Notice • `2` – Warning • `3` – Alarm 1 • `4` – Alarm 2 • `5` – Alarm 3 Value is only set if `notificationmode` = `1`. Default value is <empty>. This parameter is optional.
latitude	int	Geographic latitude of the geometric centre of the shape (rectangle or circle). Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia. This parameter is optional. For polygon and corridor, please use the parameter `point` multiple times to submit each point of the complex shape.
longitude	int	See description for `latitude` for analogue use for longitude.
radius	int	Radius of the area in meters. Only set if the area is a circle. This parameter is optional.

Parameter	Type	Description
width	int	Width of the area in meters. Only set if area is a rectangle or corridor. This parameter is optional.
height	int	Height of the area in meters. Only set if area is a rectangle. This parameter is optional.
point	string	Comma-separated `latitude/longitude` pair, which describes a single point of the polygon or corridor (in WGS84 notation). `<latitude>,<longitude>` Repeat this parameter for each point in the shape. **Note**: Use HTTP-POST if the generated request URL is longer than 2000 characters. This parameter is optional.
color	string (6)	The color of the area as a 6-digit hex RGB-color code. This parameter is optional. The following colors are available within the Webfleet GUI:

getAreaPoints

Description

This action returns a list of coordinates describing the geometric form and location of an area. `getAreasPoints` is only useful for areas in the shape of polygons or corridors.

Technical details

SOAP endpoint address / function name	areaService#getAreasPoints
Request limits	10 requests / minute

Parameters

getAreaPoints requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getAreasPoints:

Parameter	Type	Description
areano	string (50)	Identification number of an area. Unique within an account, case-sensitive. **Note**: Specify either `areano` or `areauid`; do not use both parameters in combination.
areauid	string (30)	A unique, unchangeable identifier for the area, automatically generated by Webfleet. **Note**: Specify either `areano` or `areauid`; do not use both parameters in combination.
active	boolean	Indicator for whether an area is enabled. Valid values: • `0` - area disabled • `1` - area enabled Omit parameter to retrieve all areas. This parameter is optional.

Result

Result for getAreasPoints

Parameter	Type	Description
areano	string (50)	Identification number of an area. Unique within an account, case-sensitive.
areauid	string (30)	A unique, unchangeable identifier for the area, automatically generated by Webfleet.

Parameter	Type	Description
latitude	int	Geographic latitude of a single point of the corridor or polygon. Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
longitude	int	See description for `latitude` for analogue use for longitude.

getAreaAssignments

Description

`getAreaAssignments` returns a list of vehicles and object groups, that are assigned to an area. Assignments can be used to limit the validity of the area to specific vehicles or object groups.

Technical details

SOAP endpoint address / function name	areaService#getAreaAssignments
Request limits	10 requests / minute

Parameters

getAreaAssignments requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getAreaAssignments:

Parameter	Type	Description
areano	string (50)	Identification number of an area. Unique within an account, case-sensitive. Indicate either `areano` or `areauid`; do not use both parameters in combination.
areauid	string (30)	A unique, unchangeable identifier for the area, automatically generated by Webfleet. Indicate either `areano` or `areauid`; do not use both parameters in combination.

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. This parameter is optional. Indicate either `objectno` or `objectuid`; do not use both parameters in combination.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. This parameter is optional. Indicate either `objectno` or `objectuid`; do not use both parameters in combination.
objectgroupname	string (50)	A name of an object group. This parameter is optional.

Result

Result for getAreaAssignments:

Parameter	Type	Description
areano	string (50)	Identification number of an area. Unique within an account, case-sensitive.
areauid	string (30)	A unique, unchangeable identifier for the area, automatically generated by Webfleet.
assignmentuid	string (30)	A unique, unchangeable identifier for the assignment, automatically generated by Webfleet.
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Value is set if the area is assigned to this vehicle, `objectgroupname` is empty in this case.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Value is set if the area is assigned to this vehicle, `objectgroupname` is empty in this case.
objectgroupname	string (50)	A name of an object group. Value is set if the area is assigned to this group. `objectno` and `objectuid` are empty in this case.

insertAreaAssignment

Description

`insertAreaAssignment` adds a new assignment of a vehicle or object group to a specified area. Assignments can be used to limit the validity of the area to individual vehicles or object groups.

Technical details

SOAP endpoint address / function name	areaService#insertAreaAssignment
Request limits	10 requests / minute

Parameters

insertAreaAssignment requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to insertAreaAssignment:

Parameter	Type	Description
areano	string (50)	Identification number of an area. Unique within an account, case-sensitive. This parameter is optional. **Note**: Indicate either `areano` or `areauid`; do not use both parameters in combination.
areauid	string (30)	A unique, unchangeable identifier for the area, automatically generated by Webfleet. This parameter is optional. **Note**: Indicate either `areano` or `areauid`; do not use both parameters in combination.
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. **Note**: It is required to indicate either `objectno`, `objectuid`, or `objectgroupname`; do not combine two or all of these parameters.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. **Note**:It is required to indicate either `objectno`, `objectuid`, or `objectgroupname`; do not combine two or all of these parameters.
objectgroupname	string (50)	A name of an object group. **Note**:It is required to indicate either `objectno`, `objectuid`, or `objectgroupname`; do not combine two or all of these parameters.

deleteAreaAssignment

Description

`deleteAreaAssignment` deletes one or more vehicle or objectgroup assignments of an area.

Technical details

SOAP endpoint address / function name	areaService#deleteAreaAssignment
Request limits	10 requests / minute

Parameters

deleteAreaAssignment requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to deleteAreaAssignment:

Parameter	Type	Description
areano	string (50)	Identification number of an area. Unique within an account, case-sensitive. **Note**: Indicate either `areano` or `areauid`; do not use both parameters in combination.
areauid	string (30)	A unique, unchangeable identifier for the area, automatically generated by Webfleet. **Note**: Indicate either `areano` or `areauid`; do not use both parameters in combination.
assignmentuid	string (30)	A unique, unchangeable identifier for the assignment, automatically generated by Webfleet. Repeat this parameter for each assignment that shall be deleted. Use HTTP-POST if the generated request URL is longer than 2000 characters. This parameter is required.

getAreaSchedules

Description

`getAreaSchedules` returns a list of days and times for which the area shall be effective. Schedules can be used to limit the validity of areas to specific days or time periods.

Technical details

SOAP endpoint address / function name	areaService#getAreaSchedules
Request limits	10 requests / minute

Parameters

getAreaSchedules requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getAreaSchedules:

Parameter	Type	Description
areano	string (50)	Identification number of an area. Unique within an account, case-sensitive. **Note:** Indicate either `areano` or `areauid`; do not use both parameters in combination.
areauid	string (30)	A unique, unchangeable identifier for the area, automatically generated by Webfleet. **Note**: Indicate either `areano` or `areauid`; do not use both parameters in combination.

Result

Result for getAreaSchedules:

Parameter	Type	Description
areano	string (50)	Identification number of an area. Unique within an account, case-sensitive.
areauid	string (30)	A unique, unchangeable identifier for the area, automatically generated by Webfleet.
scheduleuid	string (30)	A unique, unchangeable identifier for the schedule, automatically generated by Webfleet.
mon	boolean	Indicates if the area is activated on this day. Valid values: • `0` - area disabled • `1` - area enabled
tue	boolean	See `mon`.

Parameter	Type	Description
wed	boolean	See `mon`.
thu	boolean	See `mon`.
fri	boolean	See `mon`.
sat	boolean	See `mon`.
sun	boolean	See `mon`.
start_time	string (5)	Time of day, the activation of the area starts on each of the days above. Format `hh:mm`, in the local time zone of the user.
end_time	string (5)	Time of day, the activation of the area ends on each of the days above. Format `hh:mm`, in the local time zone of the user.

insertAreaSchedule

Description

`insertAreaSchedule` adds a list of weekdays and times for which the area shall be effective. Schedules can be used to limit the validity of areas in a temporal way.

Technical details

SOAP endpoint address / function name	areaService#insertAreaSchedule
Request limits	10 requests / minute

Parameters

insertAreaSchedule requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to insertAreaSchedule:

Parameter	Type	Description
areano	string (50)	Identification number of an area. Unique within an account, case-sensitive. **Note**: Indicate either `areano` or `areauid`; do not use both parameters in combination.

Parameter	Type	Description
areauid	string (30)	A unique, unchangeable identifier for the area, automatically generated by Webfleet. **Note**: Indicate either `areano` or `areauid`; do not use both parameters in combination.
mon	boolean	Indicates if the area is activated on this day. Valid values: • `0` - area disabled • `1` - area enabled This parameter is optional.
tue	boolean	See `mon`.
wed	boolean	See `mon`.
thu	boolean	See `mon`.
fri	boolean	See `mon`.
sat	boolean	See `mon`.
sun	boolean	See `mon`.
start_time	string (5)	Time of day, the activation of the area starts on each of the days above. Format `hh:mm`, in the local time zone of the user. Default value is `00:00`. This parameter is optional.
end_time	string (5)	Time of day, the activation of the area ends on each of the days above. Format `hh:mm`, in the local time zone of the user. Default value is `23:59`. This parameter is optional.

deleteAreaSchedule

Description

`deteleAreaSchedule` deletes one or more schedules of an area.

Technical details

SOAP endpoint address / function name	areaService#deleteAreaSchedule
Request limits	10 requests / minute

Parameters

deleteAreaSchedule requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to deleteAreaSchedule:

Parameter	Type	Description
areano	string (50)	Identification number of an area. Unique within an account, case-sensitive. **Note**: Specify either `areano` or `areauid`; do not use both parameters in combination.
areauid	string (30)	A unique, unchangeable identifier for the area, automatically generated by Webfleet. **Note**: Specify either `areano` or `areauid`; do not use both parameters in combination.
scheduleuid	string (30)	A unique, unchangeable identifier for the schedule, automatically generated by Webfleet. Repeat this parameter for each assignment that shall be deleted. Use HTTP-POST if the generated request URL is longer than 2000 characters. This parameter is required.

LINK.connect

For documentation of the LINK.connect functionality and integration between third party devices and LINK devices, terms and abbreviations used please refer to the LINK.connect Reference document at https://www.webfleet.com/webfleet/partners/integration/developer-resources/

API key mandatory

Note: All LINK.connect-related functions require a dedicated API key.

Always use a dedicated API for the LINK.connect integration solution that you will develop.

sendAuxDeviceData

Description

`sendAuxDeviceData` sends the given opaque payload data to the specified thrid party device using the indicated Webfleet unit/LINK device.

Maximum allowed data size is 2560 byte raw opaque payload data. The base64 encoding required for transmission of opaque payload through the web service API does not count as raw payload data size.

There can only be one pending aux device data message for a LINK at a time. Further calls to `sendAuxDeviceData` for the same LINK will be rejected until the data is transferred to the LINK.

To track the status of the opaque payload message once sent, please use the WEBFLEET.connect Queue Service.

Technical details

SOAP endpoint address / function name	auxDeviceDataService#sendAuxDeviceData
Request limits	10 requests / minute

Parameters

sendAuxDeviceData requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to sendAuxDeviceData:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. This parameter is required.

Parameter	Type	Description
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. This parameter is required.
data	base64(raw(256))	Base64 encoded user payload. the data is not interpreted by Webfleet.
deviceid	string	Identifier of the aux device. In case of Bluetooth, this is the Bluetooth adapter MAC address (Bluetooth device address) in the 6 byte notation separated by colon, e.g. 11:22:33:44:55:66
correlationid	int	An arbitrary identifier which helps to identify the Webfleet messages in the message queue service that belong to this request. Should be unique in the integration server backend. Optional.

LINK to aux device connection handling when sending opaque payload

The LINK will try to connect to the specified aux device, if the device is not already connected. If the LINK is not paired to the aux device yet, it will initiate a Bluetooth pairing if needed. The configured PIN is used for pairing.

Data will be transferred to the aux device when a Bluetooth connection exists. If the LINK is unable to establish this connection the opaque payload will remain in the LINK's opaque payload data message queue, reducing the size of this queue.

getLocalAuxDeviceConfig

Description

`getLocalAuxDeviceConfig` returns the stored and applicable configuration of a Webfleet unit regarding Bluetooth connectivity.

Technical details

SOAP endpoint address / function name	auxDeviceDataService#getLocalAuxDeviceConfig
Request limits	10 requests / minute

Parameters

getLocalAuxDeviceConfig requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getLocalAuxDeviceConfig:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. This parameter is optional.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. This parameter is optional.

Result

Result for getLocalAuxDeviceConfig:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated.
servicename	string (32)	Local Bluetooth service name of the LINK.
serviceuuid	UUID(128)	Local Bluetooth service UUID in 128 bit UUID format.
authentication	boolean	Flag indicating whether authentication for local SPP service is enabled.
addgpsfix	boolean	Flag indicating whether the LINK adds GPS data to every opaque payload data message.
addtimestamp	boolean	Flag indicating whether the LINK adds a time stamp to every opaque payload data message.
addodometer	boolean	Flag indicating whether the LINK adds the current odometer value to every opaque payload data message.

configureLocalAuxDevice

Description

`configureLocalAuxDevice` allows the integration server backend to change configuration settings relating to aux devices. Currently the settings are specific to Bluetooth.

Note: All configuration parameters are optional. Parameters not specified in the request will not be changed.

Technical details

SOAP endpoint address / function name	auxDeviceDataService#configureLocalAuxDevice

Request limits	10 requests / minute

Parameters

configureLocalAuxDevice requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to configureLocalAuxDevice:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. **Note**: Either `objectno` or `objectuid` is required.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. **Note**: Either `objectno` or `objectuid` is required.
servicename	string (32)	Local Bluetooth service name of the LINK.
serviceuuid	UUID(128)	Local Bluetooth service UUID in 128 bit UUID format.
authentication	boolean	Flag indicating whether authentication for local SPP service is enabled. When set to `1` authentication for local SPP service is enabled.
addgpsfix	boolean	Flag indicating whether the LINK adds GPS data to every opaque payload data message. When set to `1` the LINK will add GPS data to every opaque payload data message.
addtimestamp	boolean	Flag indicating whether the LINK adds a time stamp to every opaque payload data message. When set to `1` the LINK will add a time stamp to every opaque payload data message.
addodometer	boolean	Flag indicating whether the LINK adds the current odometer value to every opaque payload data message. When set to `1` the LINK will add the current odometer value to every opaque payload data message.

getRemoteAuxDeviceConfig

Description

`getRemoteAuxDeviceConfig` returns the stored and applicable configuration of a Webfleet unit regarding Bluetooth remote aux devices.

Technical details

SOAP endpoint address / function name	auxDeviceDataService#getRemoteAuxDeviceConfig
Request limits	10 requests / minute

Parameters

getRemoteAuxDeviceConfig requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getRemoteAuxDeviceConfig:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Either of `objectno` or `objectuid` is mandatory.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Either of `objectno` or `objectuid` is mandatory.

Result

Result for getRemoteAuxDeviceConfig:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated.
configid	int	Identifies the remote aux device configuration. Every remote device configuration for a LINK must have a unique identifier.
deviceid	string	Identifier of the remote aux device. In the case of Bluetooth, this is the Bluetooth adapter MAC address (Bluetooth device address) in the 6 byte notation. Separated by colon, e.g. 11:22:33:44:55:66.

Parameter	Type	Description
friendlyname	string (248)	A name for the remote device that is easier to read for humans. Maximum length is 248 bytes UTF-8.
servicename	string (32)	Remote device SPP service name.
serviceuuid	UUID(128)	Remote device SPP service UUID in 128 bit UUID format.
channel	int	RFCOMM channel ID (1..30).
pin	string(16)	Bluetooth PIN to be used for pairing. When set to an empty value, no PIN will be used. Maximum length is 16 bytes UTF-8.
sppbuffersize	int	SPP buffer threshold of the LINK for conversion of SPP data streams (from aux device to LINK) to opaque payload messages. Maximum 1024 Byte.
sppflushtimeout	int	SPP buffer timeout (ms) of the LINK for conversion of SPP data streams (from aux device to LINK) to opaque payload messages. Maximum 5000 ms.

configureRemoteAuxDevice

Description

`configureRemoteAuxDevice` allows the integration server backend to change configuration settings relating to a remote aux device.

Currently the settings are specific to Bluetooth. All configuration parameters are optional. Parameters not specified in the request will not be changed. To remove a value, specify `null` or use an empty parameter value.

Technical details

SOAP endpoint address / function name	auxDeviceDataService#configureRemoteAuxDevice
Request limits	10 requests / minute

Parameters

configureRemoteAuxDevice requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to configureRemoteAuxDevice:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. **Note**: Either `objectno` or `objectuid` is required.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. **Note**: Either `objectno` or `objectuid` is required.
configid	int	Identifies the remote aux device configuration. Every remote device configuration for a LINK must have a unique identifier. The maximum allowed number of remote device configurations per LINK is 5. Valid values: 0 … 4
deviceid	string	Identifier of the remote aux device. In the case of Bluetooth, this is the Bluetooth adapter MAC address (Bluetooth device address) in the 6 byte notation. Separated by colon, e.g. 11:22:33:44:55:66 **Note**: Either `deviceid` or `friendlyname` is required.
friendlyname	string (248)	Remote device Friendly name. Maximum length is 248 bytes UTF-8. **Note**: Either `deviceid` or `friendlyname` is required.
servicename	string (32)	Remote device SPP service name.
serviceuuid	UUID(128)	Remote device SPP service UUID in 128 bit UUID format.
channel	int	RFCOMM channel ID (1..30).
pin	string(16)	Bluetooth PIN to be used for pairing. When set to an empty value, no PIN will be used. Maximum length is 16 bytes UTF-8.
sppbuffersize	int	SPP buffer threshold of the LINK for conversion of SPP data streams (from aux device to LINK) to opaque payload messages. Maximum 1024 Byte.
sppflushtimeout	int	SPP buffer timeout (ms) of the LINK for conversion of SPP data streams (from aux device to LINK) to opaque payload messages. Maximum 5000 ms.

removeRemoteAuxDeviceConfig

Description

`removeRemoteAuxDeviceConfig` removes a remote aux device configuration.

Technical details

SOAP endpoint address / function name	auxDeviceDataService#removeRemoteAuxDeviceConfig
Request limits	10 requests / minute

Parameters

removeRemoteAuxDeviceConfig requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to removeRemoteAuxDeviceConfig:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Either of `objectno` or `objectuid` is mandatory.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Either of `objectno` or `objectuid` is mandatory.
configid	int	Identifies the remote aux device configuration. Every remote device configuration for a LINK must have a unique identifier. The maximum allowed number of remote device configurations per LINK is 5. Valid values: `0` … `4`. This parameter is required.

clearAuxDeviceDataQueue

Description

`clearAuxDeviceDataQueue` removes data from the outgoing queue of the LINK. This can be used to remove stale data for third party devices that do not exist anymore or that are unlikely to connect anymore. The LINK does not remove outgoing data by itself because it cannot decide if a third party device will connect sometime.

Technical details

SOAP endpoint address / function name	auxDeviceDataService#clearAuxDeviceDataQueue

Request limits	10 requests / minute

Parameters

clearAuxDeviceDataQueue requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to clearAuxDeviceDataQueue:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. **Note**: Either `objectno` or `objectuid` is mandatory.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. **Note**: Either `objectno` or `objectuid` is mandatory.
deviceid	string	Identifier of the remote aux device. In the case of Bluetooth, this is the Bluetooth adapter MAC address (Bluetooth device address) in the 6 byte notation. Separated by colon, e.g. 11:22:33:44:55:66 This parameter is optional.

resetAuxDeviceData

Description

If the LINK does not respond to outgoing aux device data sent with sendAuxDeviceData the transfer status is incomplete. In such cases `resetAuxDeviceData` must be used to reset the outgoing data transfer and get back to a consistent state between integration server backend and aux device. This is important because you cannot send new data until the current transfer is completed or cancelled/reset. If the opaque payload to be delivered with the original send is still important from the view of LINK.connect integration solution, the integration server backend should repeat the `sendAuxDeviceData` operation after the reset.

Technical details

SOAP endpoint address / function name	auxDeviceDataService#resetAuxDeviceData
Request limits	10 requests / minute

Parameters

resetAuxDeviceData requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to resetAuxDeviceData:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. **Note**: Either `objectno` or `objectuid` is mandatory.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. **Note**: Either `objectno` or `objectuid` is mandatory.

Plugins

insertExternalEvent

Description

`insertExternalEvent` allows integrators to create own events for vehicles. The events are handled in the same way like internal events created by Webfleet. The user of the Webfleet user interface can see notifications for the events and is able to acknowledge and/or resolve them. Also notification forwarding and reporting works as usual.

Use cases are for example warnings for external temperature or tire pressure sensors. Gathering of the data and decisions about warnings or alarm levels is up to the 3rd party integrator.

Prerequisites

1. The integrator needs to register a Webfleet Plugin. The plugin is tied to the API key.
2. The Webfleet customer needs to purchase the additional feature **Webfleet Plugin** to have the plugin data available in Webfleet. This is booked per vehicle (object).

Please refer to the Webfleet Plugin developer guide for more information.

Technical details

SOAP endpoint address / function name	pluginsService#insertExternalEvent
Request limits	5 requests / minute

Parameters

insertExternalEvent requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to `insertExternalEvent` :

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
data	string	JSON containing the event payload. The content is validated against the JSON schema.

setExternalObjectData

Description

`setExternalObjectData` allows integrators to specify arbitrary data for a vehicle in a well-defined JSON format. The data is shown on the Webfleet user interface. The intended use is to display measured data from sensors or other equipment, such as temperature, tire pressure, cargo load etc. Gathering such data is up to the 3rd party integrator.

Prerequisites

1. The integrator needs to register a Webfleet Plugin. The plugin is tied to the API key.
2. The Webfleet customer needs to book the additional feature Webfleet Plugin to have the plugin data available in Webfleet. This is booked per vehicle (object).

Please refer to the Webfleet Plugin developer guide for more information.

Technical details

SOAP endpoint address / function name	pluginsService#setExternalObjectData
Request limits	10 requests / minute

Parameters

setExternalObjectData requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to `setExternatObjactData`:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
data	string	JSON containing the event payload. The content is validated against the JSON schema.

Cold Chain

getCurrentColdChainSensorData

Description

`getCurrentColdChainSensorData` returns current sensor data along with some related information.

Sensor data is not only related to temperature. It also includes reefer unit data like fuel level or engine hours.

This action deprecates getCurrentTemperatureData which will be removed in a later release.

Technical details

SOAP endpoint address / function name	coldChainDataService#getCurrentColdChainSensorData
Request limits	10 requests / minute

Parameters

getCurrentColdChainSensorData requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getCurrentColdChainSensorData:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`. This parameter is optional.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`. This parameter is optional.
objectgroupname	string	The given entry filters the result list on the object groups with matching names. Can be used alternatively to `objectgroupuid`. This parameter is optional.

Parameter	Type	Description
objectgroupuid	string	A unique, unchangeable identifier for the indicated object group, automatically generated. Can be used alternatively to `objectgroupname`. This parameter is optional.

Result

Result for getCurrentColdChainSensorData:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
timestamp	dateTime	Timestamp of the sensor measurement.
sensorType	string	The type of sensor. • TEMPERATURE • FUEL_LEVEL • ENGINE_HOURS • ELECTRICAL_HOURS
sensorName	string	Name of the sensor. This name can be changed in the Webfleet UI.
sensorCode	string	The content of the `sensorCode` depends on the sensor type and temperature sensor type. For `ONE_WIRE` it contains the ROM code of the sensor. This is a unique serial number of the hardware. For `BLE_TEMPERATURE` it contains the Bluetooth address of the sensor. For `THERMOGRAPH` it contains the sensor ID connected to the thermograph (e.g. '1', '2', etc.). For other sensor types this contains a generated identifier, which is unqiue for the object (e.g. 'GENERIC_1').
latitude	int	Latitude when the sensor value was obtained. Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.

Parameter	Type	Description
longitude	int	Longitude when the sensor value was obtained. Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
postext		A detailed textual description of the location of the measurement.
temperatureSensorType	string	• `ONE_WIRE` - One-Wire sensor • `THERMOGRAPH` - Sensor of a thermograph connected to LINK device • `BLE_TEMPERATURE` - Bluetooth Low Energy temperature sensor • `SUPPLY_AIR` - The air temperature after compressor, before going into the compartment • `RETURN_AIR` - The temperature of air leaving the compartment • `GENERIC` - Generic type Only available for data with `sensorType` TEMPERATURE.
temperature	float	Temperature measurement (in ºC). Only available for data with `sensorType` TEMPERATURE.
fuelLevel	float	Reefer unit fuel level in percent. Only available for data with `sensorType` FUEL_LEVEL.
engineHours	float	Total hours the engine of the reefer unit was running. Only available for data with `sensorType` ENGINE_HOURS.
electricalHours	float	Hours the engine of the reefer unit was running in electrical mode. Only available for data with `sensorType` ELECTRICAL_HOURS.

getHistoricalColdChainSensorData

Description

`getHistoricalColdChainSensorData` returns sensor data for an object within a specific time range.

Sensor data is not only related to temperature. It also includes reefer unit data like fuel level or engine hours.

This action deprecates getHistoricalTemperatureData which will be removed in a later release.

Technical details

SOAP endpoint address / function name	coldChainDataService#getHistoricalColdChainSensorData
Request limits	10 requests / minute

Parameters

getCurrentColdChainSensorData requires the following common parameters:

- Authentication parameters
- General parameters
- Date range filter parameters, periods longer than 7 days are not permitted.

Parameters specific to getHistoricalColdChainSensorData:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Result

Result for getHistoricalColdChainSensorData:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
timestamp	dateTime	Timestamp of the sensor measurement.
sensorType	string	The type of sensor. • TEMPERATURE • FUEL_LEVEL • ENGINE_HOURS • ELECTRICAL_HOURS

Parameter	Type	Description
sensorName	string	Name of the sensor. This name can be changed in the Webfleet UI.
sensorCode	string	The content of the `sensorCode` depends on the sensor type and temperature sensor type. For `ONE_WIRE` it contains the ROM code of the sensor. This is a unique serial number of the hardware. For `BLE_TEMPERATURE` it contains the Bluetooth address of the sensor. For `THERMOGRAPH` it contains the sensor ID connected to the thermograph (e.g. '1', '2', etc.). For other sensor types this contains a generated identifier, which is unqiue for the object (e.g. 'GENERIC_1').
latitude	int	Latitude when the sensor value was obtained. Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
longitude	int	Longitude when the sensor value was obtained. Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
postext		A detailed textual description of the location of the measurement.
temperatureSensorType	string	• `ONE_WIRE` - One-Wire sensor • `THERMOGRAPH` - Sensor of a thermograph connected to LINK device • `BLE_TEMPERATURE` - Bluetooth Low Energy temperature sensor • `SUPPLY_AIR` - The air temperature after compressor, before going into the compartment • `RETURN_AIR` - The temperature of air leaving the compartment • `GENERIC` - Generic type Only available for data with `sensorType` TEMPERATURE.
temperature	float	Temperature measurement (in ºC). Only available for data with `sensorType` TEMPERATURE.

Parameter	Type	Description
fuelLevel	float	Reefer unit fuel level in percent. Only available for data with `sensorType` FUEL_LEVEL.
engineHours	float	Total hours the engine of the reefer unit was running. Only available for data with `sensorType` ENGINE_HOURS.
electricalHours	float	Hours the engine of the reefer unit was running in electrical mode. Only available for data with `sensorType` ELECTRICAL_HOURS.

getCurrentColdChainStateData

Description

`getCurrentColdChainStateData` returns current state data along with some related information.

With this action you can get the current state of doors as well as reefer unit state data.

This action deprecates getCurrentRefrigeratedDoorStatusData which will be removed in a later release.

Technical details

SOAP endpoint address / function name	coldChainDataService#getCurrentColdChainStateData
Request limits	10 requests / minute

Parameters

getCurrentColdChainStateData requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getCurrentColdChainSensorData:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`. This parameter is optional.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`. This parameter is optional.

Parameter	Type	Description
objectgroupname	string	The given entry filters the result list on the object groups with matching names. Can be used alternatively to `objectgroupuid`. This parameter is optional.
objectgroupuid	string	A unique, unchangeable identifier for the indicated object group, automatically generated. Can be used alternatively to `objectgroupname`. This parameter is optional.

Result

Result for getCurrentColdChainStateData:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
timestamp	dateTime	Timestamp of the state change.
latitude	int	Latitude when the sensor value was obtained. Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
longitude	int	Longitude when the sensor value was obtained. Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
postext		A detailed textual description of the location of the measurement.

Parameter	Type	Description
stateDataType	string	The type of state data. • DOOR • FUEL_TANK_SIZE • POWER_MODE • REEFER_UNIT_STATE • SETPOINT • UNIT_OPERATIONAL_MODE • ZONE_OPERATIONAL_MODE
reeferUnitState	string	Current state of the reefer unit. • ON • OFF • UNKNOWN Only available for data with `stateDataType` REEFER_UNIT_STATE.
unitOperationalMode	string	Operational mode of the cooling unit. • CONTINUOUS • CYCLE_SENTRY • UNKNOWN Only available for data with `stateDataType` UNIT_OPERATIONAL_MODE.
powerMode	string	Power mode of the cooling unit engine. • DIESEL • ELECTRIC • CRYO • UNKNOWN Only available for data with `stateDataType` POWER_MODE.
setPointTemperature	float	Setpoint temperature for the zone specified in result parameter `zone`. Only available for data with `stateDataType` SETPOINT.
doorCode	string	The `doorcode` is a generated door identifier, unique for an object. Only available for data with `stateDataType` DOOR.
doorName	string	Name of the door. Only available for data with `stateDataType` DOOR.

Parameter	Type	Description
doorStatus	string	Door status. • OPEN • CLOSED Only available for data with `stateDataType` DOOR.
fuelTankSize	int	Fuel tank size in litres. Only available for data with `stateDataType` FUEL_TANK_SIZE.
zoneOperationalMode	string	Operational mode of the zone specified in result parameter `zone`. • COOLING • DEFROSTING • HEATING • OFF • UNKNOWN Only available for data with `stateDataType` ZONE_OPERATIONAL_MODE.
zone	int	The zone (compartment) which the state data belongs to. Only available for data with `stateDataType` ZONE_OPERATIONAL_MODE or SETPOINT.

getHistoricalColdChainStateData

Description

`getHistoricalColdChainStateData` returns current state data changes for an object within a specific time range.

With this action you can get the history state changes of doors as well as reefer unit state data.

This action deprecates getHistoricalRefrigeratedDoorStatusData which will be removed in a later release.

Technical details

SOAP endpoint address / function name	coldChainDataService#getHistoricalColdChainStateData
Request limits	10 requests / minute

Parameters

getHistoricalColdChainStateData requires the following common parameters:

- Authentication parameters
- General parameters
- Date range filter parameters, periods longer than 7 days are not permitted.

Parameters specific to getCurrentColdChainSensorData:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.

Result

Result for getHistoricalColdChainStateData:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
timestamp	dateTime	Timestamp of the state change.
latitude	int	Latitude when the sensor value was obtained. Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
longitude	int	Longitude when the sensor value was obtained. Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
postext		A detailed textual description of the location of the measurement.

Parameter	Type	Description
stateDataType	string	The type of state data. • DOOR • FUEL_TANK_SIZE • POWER_MODE • REEFER_UNIT_STATE • SETPOINT • UNIT_OPERATIONAL_MODE • ZONE_OPERATIONAL_MODE
reeferUnitState	string	Current state of the reefer unit. • ON • OFF • UNKNOWN Only available for data with `stateDataType` REEFER_UNIT_STATE.
unitOperationalMode	string	Operational mode of the cooling unit. • CONTINUOUS • CYCLE_SENTRY • UNKNOWN Only available for data with `stateDataType` UNIT_OPERATIONAL_MODE.
powerMode	string	Power mode of the cooling unit engine. • DIESEL • ELECTRIC • CRYO • UNKNOWN Only available for data with `stateDataType` POWER_MODE.
setPointTemperature	float	Setpoint temperature for the zone specified in result parameter `zone`. Only available for data with `stateDataType` SETPOINT.
doorCode	string	The `doorcode` is a generated door identifier, unique for an object. Only available for data with `stateDataType` DOOR.
doorName	string	Name of the door. Only available for data with `stateDataType` DOOR.

Parameter	Type	Description
doorStatus	string	Door status. • OPEN • CLOSED Only available for data with `stateDataType` DOOR.
fuelTankSize	int	Fuel tank size in litres. Only available for data with `stateDataType` FUEL_TANK_SIZE.
zoneOperationalMode	string	Operational mode of the zone specified in result parameter `zone`. • COOLING • DEFROSTING • HEATING • OFF • UNKNOWN Only available for data with `stateDataType` ZONE_OPERATIONAL_MODE.
zone	int	The zone (compartment) which the state data belongs to. Only available for data with `stateDataType` ZONE_OPERATIONAL_MODE or SETPOINT.

getCurrentTemperatureData

Description

`getCurrentTemperatureData` returns current temperature data along with some related information.

This action is deprecated. Please use getCurrentColdChainSensorData instead.

Technical details

SOAP endpoint address / function name	coldChainDataService#getCurrentTemperatureData
Request limits	10 requests / minute

Parameters

getCurrentTemperatureData requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getCurrentTemperatureData:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`. This parameter is optional.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`. This parameter is optional.
objectgroupname	string	The given entry filters the result list on the object groups with matching names. Can be used alternatively to `objectgroupuid`. This parameter is optional.
objectgroupuid	string	A unique, unchangeable identifier for the indicated object group, automatically generated. Can be used alternatively to `objectgroupname`. This parameter is optional.

Result

Result for getCurrentTemperatureData:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
sensorcode	string	The content of the `sensorcode` depends on the sensor type. For `ONE_WIRE` it contains the ROM code of the sensor. This is a unique serial number of the hardware. For `BLE_TEMPERATURE` it contains the Bluetooth address of the sensor. For `THERMOGRAPH` it contains the sensor ID connected to the thermograph (e.g. '1', '2', etc.). For other sensor types this contains a generated identifier, which is unqiue for the object (e.g. 'GENERIC_1').
sensorname	string	Name of the sensor. This name can be changed in the Webfleet UI.

Parameter	Type	Description
sensortype	string	• `ONE_WIRE` - One-Wire sensor • `THERMOGRAPH` - Sensor of a thermograph connected to LINK device • `BLE_TEMPERATURE` - Bluetooth Low Energy temperature sensor • `SUPPLY_AIR` - The air temperature after compressor, before going into the compartment • `RETURN_AIR` - The temperature of air leaving the compartment • `GENERIC` - Generic type
temperature	float	Temperature measurement (in ºC).
timestamp	dateTime	Timestamp of the temperature measurement.
latitude	int	Latitude when `temperature` was obtained. Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
longitude	int	Longitude when `temperature` was obtained. Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
postext		A detailed textual description of the location of the temperature measurement.

getHistoricalTemperatureData

Description

`getHistoricalTemperatureData` returns temperature data for an object within a specific time range.

This action is deprecated. Please use getHistoricalColdChainSensorData instead.

Technical details

SOAP endpoint address / function name	coldChainDataService#getHistoricalTemperatureData

Request limits	1000 requests / 24 hours

Parameters

getHistoricalTemperatureData requires the following common parameters:

- Authentication parameters
- General parameters
- Date range filter parameters, periods longer than 7 days are not permitted.

Parameters specific to getHistoricalTemperatureData:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
sensorcode	string	Can be used to restrict the result to a specific sensor. This parameter is optional.
sensorname	string	Can be used to restrict the result to a specific sensor. This parameter is optional.

Result

Result for getHistoricalTemperatureData:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
sensorcode	string	The content of the `sensorcode` depends on the sensor type. For `ONE_WIRE` it contains the ROM code of the sensor. This is a unique serial number of the hardware. For `BLE_TEMPERATURE` it contains the Bluetooth address of the sensor. For `THERMOGRAPH` it contains the sensor ID connected to the thermograph (e.g. '1', '2', etc.). For other sensor types this contains a generated identifier, which is unqiue for the object (e.g. 'GENERIC_1').

Parameter	Type	Description
sensorname	string	Name of the sensor. This name can be changed in the Webfleet UI.
sensortype	string	• `ONE_WIRE` - One-Wire sensor • `THERMOGRAPH` - Sensor of a thermograph connected to LINK device • `BLE_TEMPERATURE` - Bluetooth Low Energy temperature sensor • `SUPPLY_AIR` - The air temperature after compressor, before going into the compartment • `RETURN_AIR` - The temperature of air leaving the compartment • `GENERIC` - Generic type
temperature	float	Temperature measurement (in ºC).
timestamp	dateTime	Timestamp of the temperature measurement.
latitude	int	Latitude when `temperature` was obtained. Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
longitude	int	Longitude when `temperature` was obtained. Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
postext		A detailed textual description of the location of the temperature measurement.

getCurrentRefrigeratedDoorStatusData

Description

`getCurrentRefrigeratedDoorStatusData` returns the current status of refrigerated doors along with some related information.

This action is deprecated. Please use getCurrentColdChainStateData instead.

Technical details

SOAP endpoint address / function name	coldChainDataService#getCurrentRefrigeratedDoorStatusData
Request limits	10 requests / minute

Parameters

getCurrentRefrigeratedDoorStatusData requires the following common parameters:

- Authentication parameters
- General parameters

Parameters specific to getCurrentRefrigeratedDoorStatusData:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`. This parameter is optional.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`. This parameter is optional.
objectgroupname	string	The given entry filters the result list on the object groups with matching names. Can be used alternatively to `objectgroupuid`. This parameter is optional.
objectgroupuid	string	A unique, unchangeable identifier for the indicated object group, automatically generated. Can be used alternatively to `objectgroupname`. This parameter is optional.

Result

Result for getCurrentRefrigeratedDoorStatusData:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
doorcode	string	The `doorcode` is a generated door identifier, unique for an object.

Parameter	Type	Description
doorname	string	Name of the door.
status	string	The status of the door at the time specified by the `timestamp`. • `OPEN` - The door is open • `CLOSED` - The door is closed
timestamp	dateTime	Timestamp when door was opened or closed.
latitude	int	Latitude when door changed its state. Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
longitude	int	Longitude when door changed its state. Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
postext		A detailed textual description of the location of the door status change.

getHistoricalRefrigeratedDoorStatusData

Description

`getHistoricalRefrigeratedDoorStatusData` returns changes in the status of refrigerated doors for an object within a specific time range.

This action is deprecated. Please use getHistoricalColdChainStateData instead.

Technical details

SOAP endpoint address / function name	coldChainDataService#getHistoricalRefrigerated-DoorStatusData
Request limits	1000 requests / 24 hours

Parameters

getHistoricalRefrigeratedDoorStatusData requires the following common parameters:

- Authentication parameters
- General parameters
- Date range filter parameters, periods longer than 7 days are not permitted.

Parameters specific to getHistoricalRefrigeratedDoorStatusData:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive. Can be used alternatively to `objectuid`.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
doorcode	string	Can be used to restrict the result to a specific door. This parameter is optional.
doorname	string	Can be used to restrict the result to a specific door. This parameter is optional.

Result

Result for getHistoricalRefrigeratedDoorStatusData:

Parameter	Type	Description
objectno	string (10)	Identifying number of an object. Unique within an account, case-sensitive.
objectuid	string (30)	A unique, unchangeable identifier for the indicated object, automatically generated. Can be used alternatively to `objectno`.
doorcode	string	The `doorcode` is a generated door identifier, unique for an object.
doorname	string	Name of the door.
status	string	The status of the door at the time specified by the `timestamp`. • `OPEN` - The door is open • `CLOSED` - The door is closed
timestamp	dateTime	Timestamp when door was opened or closed.

Parameter	Type	Description
latitude	int	Latitude when door changed its state. Geographic latitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
longitude	int	Longitude when door changed its state. Geographic longitude in the WGS84 coordinate system. Unit of measurement is micro degrees (10^{-6} grd). Transformation from degree, minutes and seconds can be done with the following calculation: (degrees + minutes/60 + seconds/3600) * 10^6 and rounding to an integer. See Geographic coordinate conversion on Wikipedia.
postext		A detailed textual description of the location of the door status change.

Appendix A Operation response codes

Response codes

Response code	Message	Description
0	The operation was successful.	
1	Your last action could not be processed	A system error has occurred.
8	A constraint violation occurred. No detailed information available.	
9	One of the entered values is too large. Please check your input.	
20	Your last action could not be processed.	A system error has occurred.
40	Your last action could not be processed.	A system error has occurred.
45	Access denied.You have insufficient permissions to perform this function.	
47	The time period you have entered is invalid. Please enter a valid time period.	Invalid time period.
55	Action is not valid	Indicates that the API method supplied as the parameter `action` is not available. Check your parameter for typos.
60	General error.	
63	Document contains no data	Indicates that the result set is empty for the specific combination of parameters provided for the request.
123	Unknown macro command or unit not initialised.	
1100	User name already exists for this account.	
1101	User invalid (does not exist).	
1105	Password must be given.	Either the password is empty or the parameter was not part of your request. You need to provide a password.

Response code	Message	Description
1106	Authentication failed. check account/user-name/password.	Either the account or the username supplied does not exist or the password is wrong.
1112	The input field 'Username' must not be empty.	
1114	User profile missing or in-valid.	
1115	New password missing.	
1123	New password must not be the username.	
1124	New password too short (at least 5 characters re-quired).	
1125	Password not allowed (blacklisted).	
1126	New password must not be the account name.	
1127	Old password missing.	
1128	New password too simple.	
1129	New password must not be a part of the users real name.	
1130	The account is temporary blocked.	
1132	The account is not valid anymore.	
1140	The user is not valid any-more.	
1143	API key is invalid	
1144	Session invalid or expired.	
1146	Password expired	
1147	Interface style invalid.	The used interface style parameter is not a valid interface style.
1148	User profile for interface style invalid.	The user profile for the interface style of this user is invalid.

Response code	Message	Description
1180	URL credentials are not supported. Use BasicAuth instead. See Making HTTP requests	
1220	Right level does not fit to given entity.	
1221	Unknown right level.	
1223	Right level is not allowed for the user.	Changing the given right level for the given user is not allowed.
2109	The provided object number doesn't exist.	`objectno` is invalid.
2110	Invalid objectgroupname	`objectgroupname` is invalid.
2112	Specify either objectno or objectuid, not both.	
2210	Contract not valid	The contract for the object used in the request is not valid (e.g. has expired).
2307	The referenced event forwarding configuration does not exist within the account.	
2308	The event forward configuration is missing.	
2501	Please enter a date of execution.	Missing order date
2502	Please enter an order text.	Missing order text
2503	Deficient address data	The address format is invalid.
2504	The entered time of arrival has an invalid format.	Invalid arrival time format.
2505	Invalid duration format	The duration format is invalid.
2506	Please enter the coordinates of the order destination.	Missing order destination coordinates.
2507	The entered order destination coordinates have an invalid format.	Invalid format of order destination coordinates.
2508	Invalid format order number	The order number format is invalid.

Response code	Message	Description
2509	The provided order number doesn't exist.	
2510	Order was already sent.	
2511	Missing object number.	
2512	Invalid street address.	
2513	The entered telephone number has an invalid format.	
2514	Incomplete destination description.	
2515	Duplicate Order number.	Use sendDestinationOrderExtern or updateOrderExtern if you need to modify an existing order.
2516	The selected order number is invalid or not unique.	
2517	Updating the order type of an already existing order is not supported.	
2600	Maximum message text length (UTF-8 format) exceeded	
2601	Maximum order text length (UTF-8 format) exceeded	
2602	Maximum order number length exceeded	
2603	Feature not supported.	
2604	The feature sendText is not supported by this unit	The addressed device is either used with a tariff that doesn't include messaging or hasn't been configured properly (PND not paired during initialization).
2605	The feature sendOrder is not supported by this unit	The addressed device is either used with a tariff that doesn't include messaging or hasn't been configured properly (PND not paired during initialization).
2607	Order state is invalid or missing.	
2608	Order state cannot be disabled.	
2609	Event level not supported.	

Response code	Message	Description
2610	Order state has to be enabled or disabled.	
2611	Order state not supported.	
2614	Invalid waypoint parameter.	
2615	Itinerary orders not supported for this unit type.	The unit does not support orders with itineraries.
2616	Too many waypoints.	The amount of waypoints is too large.
2617	Itinerary distance is too long.	The overall straight line distance of the order's itinerary is too long.
2618	Waypoint description text is too long.	
2622	This order state is invalid for deletion.	
2623	Shared order status link not found.	
2624	The route feature is not available for this unit.	
26257	The order number length is invalid.	
2628	The order has exceeded the maximum allowable number of documents that can be added.	
2629	The route limit for the objects has been exceeded.	
2630	Clearing orders failed.	
2631	The order relation is invalid.	
2632	The order relation target is invalid.	
2633	The order relation assignment is missing.	
2634	The order relation is circular.	
2635	The order relation type is invalid.	

Response code	Message	Description
2636	The suggested execution date is invalid.	
2637	The arrival tolerance is invalid.	
2638	The execution date is invalid.	
2639	The order type is invalid.	
2640	The route parameters are invalid.	
2641	Cannot decode base64 encoded document file of the order.	
2642	Order document file content must be not empty or null.	
2643	Document filename of the order must be not empty or null.	
2644	Maximum allowed document file size of order is exceeded.	
2645	The given file extension is not supported.	
2646	Order document ID missing or has invalid UUID format.	
2647	The order document doesn't exist.	
2648	The order document thumbnail doesn't exist.	
32000	Feature plugins not booked for this vehicle.	
32001	Invalid JSON data.	
32002	No plugin registered for the API key.	
32003	Invalid schema version number in JSON data.	
32004	Invalid external URL in JSON element href.	

Response code	Message	Description
32005	Missing parameter data. It must contain a JSON string.	
32006	Object is currently tied to external data of another plugin. Try again later when those external data have expired.	
32007	The total size of all translation strings exceeds the limit.	
33022	Unknown output name	The output name used in the request is not configured for the LINK device.
4000	An address by the same key exists already in the account.	
4001	The address does not exist.	
4002	An address with the same identification does already exist.	
4003	The address group does not exist.	
4004	A group of the same name / identification does already exist.	
4006	A name must be provided for an address.	
4007	A number identifying an address within an account must be provided.	
4008	The name of an address group can not be NULL or empty	
4009	The micro degree value of a geo-coordinate (latitude, longitude) is outside the allowed range. The allowed range is a non-null value of - 180000000 to +180000000 micro degrees.	

Response code	Message	Description
4010	A unique, non-null Identifier for an address group within an account must be given!	
4107	Invalid accountname	
6210	areano already exists.	
6211	areano parameter missing or invalid.	
6212	areaname parameter missing or invalid.	
6213	radius parameter missing or invalid.	
6214	width parameter missing or invalid.	
6215	height parameter missing or invalid.	
6218	Please specify at least one weekday for scheduled activation of this area.	
6219	Format of time parameter invalid. please specify the time in HH:mm format.	
6221	Specify either areano or areauid to identify the area.	
6222	Area not found.	
6223	area type parameter missing or invalid.	
6224	validfrom parameter invalid.	
6225	validto parameter invalid.	
6226	area contains an invalid geo-position.	
6227	area contains an invalid event level.	
6228	assignmentuid parameter missing or invalid.	
6229	Specify one of objectno, objectuid or objectgroupname.	

Response code	Message	Description
6230	scheduleuid parameter missing or invalid.	
6231	End time must be greater than start time.	
6232	Area assignment already exists.	
6233	Specify either areano or areauid, not both.	
6234	area assignment does not exist.	
6235	Area schedule does not exist.	
6236	Maxiumum number of points exceeded.	
6237	Maximum count of available of areas exceeded.	
6238	Color parameter invalid.	The used color parameter is not a valid 6-digit hex RGB-color code.
8000	General system error.	
8011	Request quota reached. More than *n* requests in *t* seconds are not permitted	Your request exceeds the quota defined for that request type or that request group. See Request limits.
8014	External requests are not allowed for that account/user. Please make sure that an API key is used and the user has the WEBFLEET.connect access right.	See Access to WEBFLEET.connect with API Key.
8015	Account / username busy processing other request.	Requests using the same account-username pair must always be serialised and may not overlap. Occurs if you did not wait long enough for a response from the previous request and prematurely terminated the connection. The recommended timeout for requests is 300 seconds. Requests may be retried after a 10 second wait interval, but might fail again until the initial request has timed out.
8104	Action missing or invalid.	

Response code	Message	Description
8105	Language missing or invalid.	
9000	Your last WEBFLEET.connect action could not be processed.	
9001	invalid parameter (range_pattern)	The `range_pattern` is invalid for this operation, please see the operation's documentation for the allowed range.
9002	missing parameter (range_pattern)	The `range_pattern` is missing for this operation, please see the operation's documentation for the allowed range.
9003	missing parameter (objectno)	
9004	invalid parameters (rangefrom_string, rangeto_string)	The time period given by `rangefrom_string` and `rangeto_string` exceeds the allowed time range for this operation and parameter combination, please see the documentation for the allowed range.
9005	invalid parameter (`msgid`)	The `msgid` is not known to the system - the message might have been deleted after being outdated.
9006	missing parameter(s) (`range_pattern` and/or `msgid`)	You need to provide either a msgid or date range filter parameters (see [Date range filter parameters])
9007	invalid parameter (`rangefrom_string`)	The use of `range_pattern` requires values for `rangefrom_string` and `rangeto_string`.
9008	invalid parameter (`rangeto_string`)	The use of `range_pattern` requires values for `rangefrom_string` and `rangeto_string`.
9009	invalid parameters (`rangefrom_string` must be a date before `rangeto_string`)	
9010	invalid `orderstate` (must be `0|100|101|102|103|201|202|203|204|205|221|222|223|224|225|241|242|243|244|245|298|299|301|302|401`)	
9011	invalid `ordertype` (must be `1|2|3`)	

Response code	Message	Description
9012	missing parameter(s) (orderid, objectid)	
9013	wrong state, order not reassignable	
9014	no valid contract for object	
9015	no valid id given	
9016	no trip id, objectno and/or date range given	
9017	For the date range given an `objectno` also needs to be given.	
9018	driver does not exist	
9019	driver group does not exist	
9020	driver name is missing	
9021	driver number is missing or invalid	
9022	driver group name is missing or invalid	
9023	driver pin must be not empty and must be a number	
9024	driver code must be a number	
9025	driver number must be unique	
9026	driver group name must be unique	
9027	country of driver invalid	
9028	driver group not unique or not existing	
9029	driver not unique or not existing	
9030	driver already in group	
9031	driver not in group	
9032	driver company invalid	

Response code	Message	Description
9033	driver description invalid	
9034	driver zip invalid	
9035	driver city invalid	
9036	driver street invalid	
9037	driver mobile invalid	
9038	driver private telephone invalid	
9039	driver email invalid	
9040	driver code not unique	
9041	tachograph card id invalid	
9042	tachograph card country invalid	
9044	tachograph card is already assigned to other driver	
9047	driver is already assigned to other ibutton	The driver was already assigned a button id. Remove the assignment first by providing an empty button id.
9048	ibutton already assigned to other driver	The button id is already assigned to another driver. Remove this assignment first by updating the other driver with an empty button id.
9050	Driver is already assigned non-manually	
9051	There's a driver already assigned to the vehicle non-manually	
9052	Driver is not assigned to the specific vehicle.	
9053	Manual driver detach not possible for non-manual assignments.	
9054	Conflicting driver parameters (driverno vs. drivergroupname)	
9055	Specify either driverNo or driverUid, not both.	

Response code	Message	Description
9056	Specify either driverGroupName or driverGroupUid, not both	
9058	Remaining driving time feature not enabled for driver's unit.	Also occurres if the driver is not attached to any unit.
9060	event does not exist	
9061	event is not on alarm level	Use resolveEventExtern to flag this event as `resolved`.
9062	event is not below alarm level	Use acknowledgeEventExtern to flag this event as `acknowledged` before resolving it.
9063	eventid invalid	
9064	event already resolved	
9065	event already acknowledged	
9066	event level unknown	The event level parameter `eventlevel_cur` provided is not within the defined range.
9067	event state unknown	One of the event state parameters (`resolved`, `acknowledged`) provided is not within the defined range.
9069	The driver key is invalid	
9070	The driver key type is not assigned to this driver	
9071	The driver key is already assigned to another driver	
9080	order does not exist	
9081	order not yet assigned to unit	
9082	order is assigned to an other unit already	
9083	orderid parameter missing or invalid	
9084	Driver assignment is not allowed	The tariff for the object does not include driver-related functionality.
9100	object does not exist	
9101	`objectno` parameter missing or invalid	

Response code	Message	Description
9102	Invalid object group name	
9104	date-rage is unparsable	
9107	licence plate number parameter invalid	
9114	This operation is presently not supported.	
9122	Object group name is missing or invalid.	
9123	object-no is not part of this object group name	Both objectno and objectgroupname are given and the objectno and objectgroupname exist, but the objectno is not part of the objectgroup.
9124	Object already in group.	
9125	Object group name is missing or invalid or not unique.	
9126	Object no is missing or invalid or not unique.	
9127	Object group already exists.	
9130	Odometer parameter invalid.	
9131	Conflicting object parameters (objectno/objectuid vs. objectgroupname)	
9135	`manufacturedyear` parameter invalid	This message is returned if the indicated year is not an 4 digit integer value.
9150	This action is not applicable with an order in this state.	
9151	Invalid HTTP request method. Only HTTP-GET requests are supported.	
9152	Output status invalid.	
9155	Message category invalid.	
9160	The geo position (lat/long) is invalid.	
9161	Address geo position is invalid.	The geo-position of the address is invalid.

Response code	Message	Description
9162	The parameter for modified since time is invalid.	
9170	The email format is invalid.	
9171	Maximum data amount for an email is exceeded.	
9172	Filter parameter invalid.	
9173	No report configuration found.	
9174	No report template found.	
9176	Report generation failed.	
9177	Report name parameter invalid.	
9179	Report type parameter invalid.	
9180	Report format parameter invalid.	
9181	Report ID parameter missing or invalid.	
9182	Report does not exist.	
9183	Invalid fuel tank size.	
9184	Invalid parameter (`referenceno`).	
9195	The request parameters for this request are invalid. Please make sure you provide parameters that adhere to the required structure (see documentation or) and the minimum amount of request data as documented.	Invalid parameter(s).
9196	Some or all request parameters are missing or incomplete. The request can not be processed.	Missing or incomplete parameter(s).
9197	The current combination of parameters can not be processed as-is. Please provide fewer / differently combined parameters.	Invalid combination of parameter(s).

Response code	Message	Description
9198	The current request with the parameters given processed successfully, but with an empty result.	Indicates that the result set is empty for the specific combination of parameters provided for the request.
9199	An upstream system did not respond properly. At the moment we are unable to service this request.	
9200	Maximum period length exceeded (range_pattern)	
9205	Date range missing	
9300	Date invalid, since it lies in the future or >90 days in past.	
9301	Invalid day.	
9302	non-numeric value	
9303	value out of range	This error code is returned for SOAP errors. See also Operation response codes for queues
9304	no duplicates allowed.	
9305	geofence-area-id duplicate entry.	
9306	unknown entry.	
9310	invalid geographical coordinate, see WGS84 standards	
9311	invalid or unknown geofence-assignment-id	
9312	invalid or unknown geofence-area-id	
9500	Address street parameter invalid	
9501	Address street number parameter invalid.	
9504	Message: address country code parameter invalid or missing	

Response code	Message	Description
9514	Unable to calculate an appropriate route.	
9517	empty geocoding result	
9538	Address not valid	
9539	Address does not exist	
9543	`name1` parameter invalid	
9544	`name2` parameter invalid	
9545	`name3` parameter invalid	
9546	`addrregion` parameter invalid	
9547	`contact` parameter invalid	
9548	`fax` parameter invalid	
9549	`mailaddr` parameter invalid	
9550	`teloffice` parameter invalid	
9551	`telmobile` parameter invalid	
9552	`telprivate` parameter invalid	
9601	Driving license number is missing or it exceeds 20 characters	
9602	Driving license issuing state is missing or it exceeds 30 characters	
9603	Driving license country code is missing, invalid or it exceeds 5 characters	Country code invalid
9604	The issue date cannot exceed the expiry date for the driving license	
9605	Driving license issue date is missing or invalid	Wrong date format
9606	Driving license expiry date is invalid	Wrong date format

Response code	Message	Description
9607	The driving license must have at least one driving license type	
9608	The driving license type must be one of these: A,A1,B,B1,BE,C,C1,C1E,CE,D,D1,D1E,DE	License type code invalid
9609	The initial date cannot exceed the final date for the driving license type	
9610	Driving license type initial date is missing or invalid	Wrong date format
9611	Driving license type final date is invalid	Wrong date format
9612	One of the driving license types informed has invalid values	Any mandatory field of driving license type is missing or it has wrong fromat
9074	Remaining Driving Time configuration invalid. Please check possible configuration options.	
9075	Remaining Driving Time configuration: Country missing or invalid.	
9076	Remaining Driving Time configuration: Regulation type missing or invalid.	
9077	Remaining Driving Time configuration: Employment type missing or invalid.	
10000	Operation time-out, retry later.	Resource currently busy, try again later.
10020	Operation service error	Returned when calling downstream service.
10100	Request intercepted and blocked by application security manager (request id = <some id>).	This error is returned when the format of the HTTP request is invalid.
20003	User name missing or invalid.	
20004	Password missing or invalid.	
20005	Real name missing.	

Response code	Message	Description
20006	Company name missing or invalid.	
20007	Info parameter is invalid.	
20008	Own user cannot be modified or deleted.	
20009	Specify either username or useruid, not both.	
20010	Specify either username or useruid to identify the user.	
20101	Object no, object group name, driver no or driver group name missing or invalid.	
20100	driverno or objectno or both are missing or invalid	
21007	UID invalid	
22015	Logbook flag is invalid	
22016	Logbook purpose is invalid	
22017	Logbook contact is invalid	
22018	Logbook comment is invalid	
22020	Logbook modification reason invalid	
22023	Aggregation level invalid.	
22024	KPI name(s) invalid.	
22036	The new driver is already assigned to other trips for the same period	
23003	ecoPLUS BT address invalid.	
23004	Remote LINK Bluetooth address invalid.	
23005	Invalid config value.	
23007	The request parameter is invalid.	

Response code	Message	Description
23008	Wakeup-timers parameter missing or invalid.	
24000	Tracking feature disabled	
25000	Contract has expired	
25001	The allowed monthly quota has been exceeded	
26001	Schedule does not exist	
26002	Schedule type parameter missing or invalid	
26003	Rule type parameter missing or invalid	
26007	Resolution status parameter invalid	
26009	Schedule name parameter missing or invalid	
26010	Schedule description parameter invalid	
26013	Reminding time parameter missing or invalid	
26014	Reminding odometer parameter missing or invalid	
26015	Interval odometer parameter missing or invalid	
26016	Interval time parameter missing or invalid	
26018	Next execution odometer parameter missing or invalid	
26019	Task does not exist	
27001	Cannot decode binary message data.	
27002	Sending of binary message data not enabled for this unit.	
27003	Binary message data missing.	

Response code	Message	Description
27004	Maximum size of data exceeded.	
27005	Maximum length of application identifier exceeded.	
27006	The application ID contains invalid characters.	
27007	The application ID is missing.	
27008	Another binary mesage data transfer is already in progress.	
28000	The size of the aux data is too large.	
28002	There's already another aux data transfer in progress. Additional data can only be sent when the LINK has acknowledged the previously sent data.	
28003	Data parameter missing.	
28004	The SPP buffer size is invalid.	
28005	The SPP flush timeout is invalid.	
28006	Cannot decode auxiliary device data. The base64 encoding may be wrong.	
28007	Service name is too long.	
28008	Service UUID has an invalid format.	
28009	Aux device data feature is not enabled for this unit.	
28010	Config ID for remote device configuration is missing or invalid.	
28011	Config ID for remote device configuration too large.	
28012	Device ID missing.	

Response code	Message	Description
28013	Device ID has an invalid format.	
28014	RFCOMM channel number is invalid	
28015	Bluetooth PIN too long	
28016	Bluetooth PIN has invalid characters	
28017	Bluetooth device name (friendly name) invalid	
28018	Remote BD_ADDR or friendly name is required	
29000	Unable to retrieve unit configuration	
29001	No dedicated status message available	
29002	Object is already synchronized with account defaults	
29003	Message text parameter is too long	
29004	Message parameter is missing	
29005	Message in message parameter is invalid	
29006	Message parameter or reset flag missing	
29007	Specify either message parameter or reset flag, not both	
29008	Index is invalid	
29009	Multiple messages with the same index defined	

Response codes - Queues

Response code	Message	Description
WFCQ_E0001	queue database doesn't exist for {0}	
WFC-Q_E0006	skipped creation of queue, exists already	Returned by createQueueExtern if a queue for that msgclass already exists for that user. The creation has been skipped and no changes were made to the existing queue.
WFC-Q_E0007	queue to acknowledge doesn't exist	The operation ackQueueMessagesExtern could not be executed as there is no queue for the `msgclass` provided.
WFCQ_E0022	queue to pop doesn't exist	There is no queue for the `msgclass` provided. A queue needs be created with createQueueExtern before trying to retrieve messages from it.
WFCQ_E0033	Hint: can contain multiple error messages	Please check the returned error message for further details.
WFCQ_E0034	queue unknown, no subscription(s) found	
WFCQ_E0035	authentication failed	The username does not exist or the password does not validate.
WFCQ_E0037	queue doesn't exist, skipping deletion	A queue that has not been created (createQueueExtern) first cannot be deleted.
WFCQ_E0041	account unknown	The account name does not exist.
WFCQ_E0043	shutdown was already initiated, no more calls allowed	
WFCQ_E0053	error due to being finally unable to resolve a deadlock while processing tx	
WFC-QCS_E0000	'action' unknown	The operation named in the action is not available.
WFC-QCS_E0001	one or more of 'action parameter missing', 'account parameter missing', 'username parameter missing', 'password parameter missing', 'msgclass parameter missing'	
WFC-QCS_E0003	empty result	

Response code	Message	Description
WFC-QCSWS_UNKNOWN	problems processing your request, please try again later	There are (likely transient) problems, that prevent the operation from completing successfully.

Appendix B: Resources

WEBFLEET.connect resources

- WEBFLEET.connect developer resources

  Download the latest documentation, libraries, examples and localised activation forms.
- WEBFLEET.connect API-key request

  Request an API key to gain access to Webfleet APIs.
- WEBFLEET.connect discussion group

  Turn to this discussion group if you have general questions or if you want to get in touch with other developers.
- Webfleet Solutions technical support

  Use the online support to access FAQs, user guides and to contact our customer support team.

Other resources

- Wikipedia article about WGS84
- Wikipedia article about Geographic coordinate conversion
- RFC 2616 - Hypertext Transfer Protocol — HTTP/1.1
- RFC 1738 - Uniform Resource Locators (URL)
- Simple Object Access Protocol (SOAP)
- Web Services Description Language (WSDL)
- SOAP Message Transmission Optimization Mechanism (MTOM)
- ISO 8601
- Wikipedia article about ISO 8601
- JSON website
- Mapcode website
- ISO 3166-1 alpha-2 codes
- IEEE guidelines for unique identifiers

Appendix C: Superseded functionality

Table: Superseded functionality

Affected functionality	Announcement date End of Life date	Alternatives
SOAP version 1.9	2012-08-06 2012-09-30	All requests to the removed version will be forwarded to the latest compatible SOAP version automatically. No changes in existing integration should be required. Please stop new integrations on this version.
SOAP version 1.8	2012-04-20 2012-07-31	All requests to the removed version will be forwarded to the latest compatible SOAP version automatically. No changes in existing integration should be required. Please stop new integrations on this version.

Affected functionality	Announcement date End of Life date	Alternatives
SOAP version 1.6 and 1.7	2012-04-20 2012-07-12	All requests to the removed version will be forwarded to the latest compatible SOAP version automatically. No changes in existing integration should be required. Please stop new integrations on this version.
showLogbookReportExtern	2011-11-09 2013-11-09	showLogbook - Requests to `showLogbookReportExtern` will be forwarded to showLogbook as of announcement date.
SOAP version 1.5	2011-08-23 2011-10-13	All requests to the removed version will be forwarded to the latest compatible SOAP version automatically. No changes in existing integration should be required. Please stop new integrations on this version.
SOAP version 1.4	2011-08-23 2011-09-12	All requests to the removed version will be forwarded to the latest compatible SOAP version automatically. No changes in existing integration should be required. Please stop new integrations on this version.
SOAP version 1.1 and 1.2	2011-05-30 2011-06-15	All requests to the removed versions will be forwarded to the latest compatible SOAP version automatically. No changes in existing integration should be required. Please stop new integrations on these versions.
SOAP versions 1.0 and 1.3	2011-05-30 2011-07-01	All requests to the removed versions will be forwarded to the latest compatible SOAP version automatically. No changes in existing integration should be required. Please stop new integrations on these versions.
showObjectGroupReportExtern	2011-05-30 2013-05-30	showObjectGroups, requests to showObjectGroupReportExtern will be forwarded to showObjectGroups as of 2011-05-30.
showObjectGroupObjectReportExtern	2011-05-30 2013-05-30	showObjectGroupObjects, requests to showObjectGroupObjectReportExtern will be forwarded to showObjectGroupObjects as of 2011-05-30.
showObjectAccelerationEvents	201-05-30 2014-05-30	showAccelerationEvents, requests to showObjectAccelerationEvents will be forwarded to showAccelerationEvents as of 2014-05-30.

Affected functionality	Announcement date End of Life date	Alternatives
showObjectSpeedingEvents	2011-05-30 2014-05-30	showSpeedingEvents, requests to showObjectSpeedingEvents will be forwarded to showSpeedingEvents as of 2014-05-30.
geocodeAddressExtern geocodeLocation (SOAP)	2010-12-15 2012-12-15	geocodeAddress, requests to geocodeAddressExtern are currently handled separately from geocodeAddress. We are planning to forward those requests to geocodeAddress in future and will inform you about this change. The corresponding SOAP function geocodeLocation will expire accordingly.
Endpoint URLs: http(s)://connect.webfleet.tomtomwork.com https://soap.connect.webfleet.tomtomwork.com	2010-09-30 2013-06-30	Announced with rebranding of TomTom WORK to TomTom Telematics. Alternatives, see Making HTTP requests, Making SOAP requests
showTrailerReportExtern	2010-09-02 2011-09-02	No alternative, support for object category trailer was stopped several years ago
showPersonReportExtern	2010-09-02 2011-09-02	no alternative, support for tracking and locating people was stopped several years ago
showStoplistReportExtern	2010-06-17 2011-06-17	showStandStills, inverted showTripReportExtern
showWorkingTimeDriverReportExtern	2010-02-18 2012-02-18	showWorkingTimes
showWorkingTimeObjectReportExtern	2010-02-18 2012-02-18	showWorkingTimes
showUsageTimeDriverReportExtern	2010-02-18 2012-02-18	No direct alternative. Please contact customer support to discuss your use case to define alternative approaches.
showUsageTimeObjectReportExtern	2010-02-18 2012-02-18	No direct alternative. Please contact customer support to discuss your use case to define alternative approaches.
showOrderMessageReportExtern	2010-01-06 2013-01-06	Make use of the Message queues system.

Appendix D: Solving known issues

Using WEBFLEET.connect SOAP with Visual Basic

When you are using WEBFLEET.connect SOAP with Visual Basic you may encounter some naming conflicts. This is because Visual Basic is not case-sensitive with names. To prevent naming conflicts, you have to edit the WSDL file manually.

For the addressService:

1. Download the WSDL file for the addressService and store it locally.
2. Find the line `<:portType name="address">` and replace `address` with `address_if`.
3. Find this line `<:binding name='addressServiceSoapBinding' type="tns:address">` and replace '`address`' with '`address_if`'.

Now you can create a new service reference using this file without naming conflicts.

For the ordersService:

1. Download the WSDL file for the ordersService and store it locally.
2. Find the line `<xs:simpleType name="RouteType">` and replace `RouteType` by `RouteTypes`.
3. Find the line `<xs:element default='Quickest' minOccurs='0' name='routeType' nillable='true' type='ns1:RouteTypes'/>` and replace `ns1:RouteType` by `ns1:RouteTypes`.

Now you can create a new service reference using this file without naming conflicts.

SOAP message size quota in Visual Studio

By default, Visual Studio restricts the maximum size for SOAP messages to 65536 bytes. In many cases this is too small for production use.

You can increase the size by adjusting the appropriate values in the file **app.config** in your Visual Studio project:

- In element `httpsTransport`, change the values of the attributes `maxReceivedMessageSize` and `maxBufferSize` to a higher value.

  Note: Use the same value for both attributes.

- In element `mtomMessageEncoding`, insert attribute `maxBufferSize`.

  Note: Set it to the same value as for the attributes above.

Appendix E: Using MTOM with SOAP

MTOM is a technique to send binary data in SOAP messages. Binary data means XML data type `xs:base64Binary`.

Without MTOM, this data would be inlined as a Base64-encoded string. Using MTOM this data is sent as a binary attachment (multipart MIME). The SOAP message contains a reference to the attachment part.

There is a size overhead for the reference to the attachment and for the multipart MIME message. For binary content less than ~1kB per SOAP message it is not worth to use MTOM. It is better to inline base64 encoded data directly instead.

MTOM in the WEBFLEET.connect SOAP API

WEBFLEET.connect is configured to use SOAP 1.2 MTOM binding. This means that all clients must use MTOM binding, too — even if MTOM is not used for a specific request.

Defining binary data in WS parameters

On the server side using byte arrays in JAXB annotated classed will produce xs:base64Binary data type in WSDLs.

The WSDL then has impact on client code generation for some programming languages, such as Java. It is especially useful for very large data (Mbytes) because it allows streaming.

Java Client Code Generation

Given these examples from the server backend used to generate WSDLs:

Code sample: Example from server backend

1.

```
@XmlElement
public byte[] userData1;
```

2.

```
@XmlElement
@XmlMimeType("application/octet-stream")
public byte[] userData2;
```

Java

Java's wsimport will generate client code like this:

Code sample: Java generated code

1.

```
protected byte[] userData1;
```

2.

```
@XmlMimeType("application/octet-stream")
protected DataHandler userData2;
```

Visual Studio

Visual Studio will generate code like this (Property setter/getters omitted):

Code sample: Visual Studio generated code

1.

```
private byte[] userData1Field;
```

2.

```
private byte[] userData2Field;
```

Data handling

As you can see, Java will generate a `DataHandler` for the member annotated with a content type. This can be useful for very large data. In most cases it is more convenient to operate on simple byte arrays. Using a `DataHandler` has no impact on MTOM vs. inlined base64. The pure byte arrays can also be transported using MTOM.

.NET does not generate any streaming handler code.

> **Note**: Streaming is not covered here in this article. It would require to use `DataHandler` on the server side, too.

Activating MTOM depending on data size

Due to the message size overhead for SOAP attachments we advise to activate MTOM only for binary data larger than ~1Kbytes.

Java – server side

On the server side it is possible to control the threshold for MTOM attachments via annotations on the SEI. The following example will activate MTOM for outgoing binary data larger or equal than 1kB. Smaller binary content will be inlined as base64 encoded string.

Code sample: Activate MTOM for outgoing binary data larger than 1KB

```
@BindingType(value = javax.xml.ws.soap.SOAPBinding.SOAP12HTTP_MTOM_BINDING)
@MTOM(threshold = 1024)
public interface MessagesService {
}
```

Java – client side

Here is an example how to use a threshold on binary data sent to the server:

Code sample: Threshold on binary data sent to server

```
MessagesService service = new MessagesService();
MTOMFeature mtom = new MTOMFeature(true, 1024);
Messages messagesPort = service.getMessagesPort(mtom);
```

.NET

.NET has no adjustable threshold. .NET will enable MTOM automatically if the binary content is above a certain size. This seems to be around 800 or 900 bytes.

Testing with soapUI

MTOM must be enabled in soapUI.

For more information please see https://www.soapui.org/soap-and-wsdl/headers-and-attachments.html#2-Attachments-and-Inline-Files

Appendix F CAN signal types

CAN signal types

CAN signal types	Description
WARNING_BULB_DEFECT_GROUP_D	(On/Off)
SWITCHEDPOWERSUPPLY	(On/Off)
WARNING_BULB_DEFECT_GROUP_B	(On/Off)
WARNING_BULB_DEFECT_GROUP_C	(On/Off)
WARNING_BULB_DEFECT_GROUP_J	(On/Off)
WARNING_BULB_DEFECT_GROUP_A	(On/Off)
WARNING_BULB_DEFECT_GROUP_F	(On/Off)
WARNING_BULB_DEFECT_GROUP_H	(On/Off)
WARNING_BULB_DEFECT_GROUP_E	(On/Off)
WARNING_BULB_DEFECT_GROUP_G	(On/Off)
WARNING_BULB_DEFECT_GROUP_I	(On/Off)
FUEL_LEVEL	Amount of fuel available in the fuel tank (L)
FUEL_CONSUMPTION	Amount of fuel consumed by the engine (mL)
CHARGER_CONNECTED	Battery charger is connected (electrical vehicles) (On/Off)
CHARGING	Battery is charging (electric/hybrid vechicles) (On/Off)
SYN_WARNING_BRAKE_FLUID_LEVEL_LOW	Brake Fluid Level Low (debounced)
SYN_WARNING_COOLANT_LEVEL_LOW	Coolant Level Low (debounced)
SYN_MIL_ABS	Defect ABS (debounced)
SYN_MIL_AIRBAG	Defect Airbag (debounced)
SYN_MIL_BRAKING_SYSTEM	Defect Braking System (debounced)
SYN_WARNING_BULB_DEFECTIVE	Defect Bulb (debounced)
SYN_MIL_CHARGE_CIRCUIT	Defect Charge Circuit (debounced)
SYN_MIL_DIESEL_ENGINE	Defect Diesel Engine (debounced)
SYN_WARNING_DIESEL_PARTICULATE_FILTER	Defect Diesel Particulate Filter (DPF) (debounced)

CAN signal types	Description
SYN_MIL_ELECTRONIC_POWER_CONTROL	Defect Electronic Power Control (debounced)
SYN_MIL_ELECTRONIC_STABILITY_PROGRAM	Defect Electronic Stability Program (debounced)
SYN_MIL_EMISSION_CONTROL_SYSTEM	Defect Emission Control System (debounced)
SYN_MIL_ENGINE_COOLING	Defect Engine Cooling (debounced)
MILEAGE_TO_INSPECTION	Distance to go before the vehicle needs an inspection (km)
MILEAGE_TO_OIL_SERVICE	Distance to go before the vehicle needs oil service (km)
TRIP_METER	Distance travelled during trip (m)
WARNING_OIL_PRESSURE_LOW	Engine oil pressure is low (On/Off)
DOOR_FRONT_LEFT	Front left door is opened (On/Off)
DOOR_FRONT_RIGHT	Front right door is open (On/Off)
WARNING_FUEL_LEVEL_LOW	Fuel level is low (On/Off)
HOOD	Hood of the car is opened (On/Off)
DAYS_TO_INSPECTION	Number of days before the vehicle needs an inspection (-)
DAYS_TO_OIL_SERVICE	Number of days before the vehicle needs oil service (-)
SYN_WARNING_OIL_LEVEL_CHECK	Oil level check (debounced)
SYN_WARNING_OIL_LEVEL_HIGH	Oil level high (debounced)
SYN_WARNING_OIL_LEVEL_LOW	Oil level low (debounced)
SYN_WARNING_OIL_PRESSURE_LOW	Oil pressure low (debounced)
DOOR_LOCKED_REAR_LEFT	On = locked (On/Off)
DOOR_LOCKED_REAR_RIGHT	On = locked (On/Off)
TRUNK_LOCKED	On = locked (On/Off)
DOOR_LOCKED_FRONT_LEFT	On = locked (On/Off)
DOOR_LOCKED_FRONT_RIGHT	On = locked (On/Off)
WARNING_BULB_DEFECTIVE	One of the lighting bulbs is defective (On/Off)
PARKING_BRAKE	Parking brake is applied (On/Off)

CAN signal types	Description
SYN_SWITCHED_POWER	Powersupply is active (On/Off) (derived)
MIL_EMISSION_CONTROL_SYSTEM	Problem with the emission control system (On/Off)
RANGE_FUEL_UNIT	Range Fuel Unit
SYN_ELECTRIC_RANGE	Range until next recharge (km) (derived)
ELECTRIC_RANGE	Range until next recharge (km|miles)
SYN_FUEL_RANGE	Range until next refuel (km) (derived)
FUEL_RANGE	Range until next refuel (km|miles)
SYN_TOTAL_RANGE	Range until next refuel or recharge (km) (derived)
TOTAL_RANGE	Range until next refuel or recharge (km| miles)
DOOR_REAR_LEFT	Rear left door is opened (On/Off)
DOOR_RIGHT_REAR	Rear right door is opened (On/Off)
STATE_OF_CHARGE	State of charge of the battery pack (%)
ENGINE_COOLANT_TEMPERATURE	Temperature of the engines coolant fluid (°C)
WARNING_BRAKE_FLUID_LEVEL_LOW	The brake fluid level is low (On/Off)
WARNING_BRAKE_PAD	The brake pads are worn (On/Off)
ELECTRONIC_PARKING_BRAKE	The electronic parking brake is enabled (On/ Off)
WARNING_COOLANT_LEVEL_LOW	The engine coolant level is low (On/Off)
WARNING_OIL_LEVEL_LOW	The engine oil level is low (On/Off)
WARNING_TYRE_PRESSURE_LOW	The pressure of one of the tyres is low (On/ Off)
WARNING_WINDSCREEN_WASHER_FLUID_LEVEL_LOW	The windscreen washer fluid level is low (On/Off)
MIL_AIRBAG	There is a problem with one of the airbags (On/Off)
MIL_ABS	There is a problem with the ABS (On/Off)
MIL_BRAKING_SYSTEM	There is a problem with the braking system (On/Off)
MIL_CHARGE_CIRCUIT	There is a problem with the charge circuit (On/Off)

CAN signal types	Description
MIL_DIESEL_ENGINE	There is a problem with the diesel engine (On/Off)
WARNING_DIESEL_PARTICULATE_FILTER	There is a problem with the diesel particulate filter (On/Off)
MIL_ELECTRONIC_POWER_CONTROL	There is a problem with the electronic power control (On/Off)
MIL_ELECTRONIC_STABILITY_PROGRAM	There is a problem with the electronic stability program (On/Off)
MIL_ENGINE_COOLING	There is a problem with the engine cooling system (On/Off)
TOW_BAR_CONNECTED	Tow bar is connected (On/Off)
TRUNK	Trunk is open (On/Off)
SYN_WARNING_TYRE_PRESSURE_LOW	Tyre pressure low (debounced)
VEHICLE_LOCKED	Vehicle is locked (On/Off)
SPEED	Vehicle speed (km/h)
BATTERY_VOLTAGE	Voltage of the battery pack (in electric and/or hybrid vehicles) (V)
WARNING_OIL_LEVEL_CHECK	Warning Oil Level Check (On/Off)
WARNING_OIL_LEVEL_HIGH	Warning Oil Level High (On/Off)
SYN_ZERO_EMISSION_DISTANCE	Zero Emission Distance (km) (derived)
AMBIENT_AIR_TEMPERATURE	Temperature outside the vehicle (°C)
AVERAGE_ENERGYCONSUMPTION_FROM_RESET	Energy Consumption From Reset
AVERAGE_ENERGY_CONSUMPTION_FROM_START	Average Energy Consumption From Start
CHARGER_CONNECTED_AND_CHARGING	The charger for the electric vehicle is connected and charging (in electric and/or PHEV vehicles)
CHARGER_CONNECTED_LOCK	The charger for the electric vehicle is locked (On/Off)
DTC_BATTERY_OVERVOLTAGE	DTC - HV Battery overvoltage (on/off)
DTC_BATTERY_UNDERVOLTAGE	DTC - HV Battery undervoltage (on/off)
DTC_BMS_FAULT	DTC - HV Battery Management System fault (on/off)
DTC_BRAKING_PEDAL	DTC - Braking pedal (on/off)

CAN signal types	Description
DTC_CONTACTOR_DOWN	DTC - Contactor down (on/off)
DTC_CONTROLLER_OVERTEMPERATURE	DTC - Controller overtemperature (on/off)
DTC_CURRENT_SENSOR_FAULT	DTC - Current sensor fault (on/off)
DTC_MOTOR_OVERTEMPERATURE	DTC - Motor overtemperature (on/off)
DTC_OVERCURRENT	DTC - Overcurrent (on/off)
DTC_PRECHARGE	DTC - Precharge (on/off)
DTC_ROTATION_SENSOR_FAULT	DTC - Rotation sensor fault (on/off)
DTC_THROTTLE_FAULT	DTC - Throttle fault (on/off)
ELECTRIC_MOTOR_TEMPERATURE	Temperature of an electric motor (°C)
ELECTRIC_RANGE_MAX	Value shows maximum theoretical range. This value is always higher than real range
ERROR_CODE_ELEKTRAMOBILYS	Error code from Elektra Mobilys Scooter
ERROR_CODE_URBANMOBILITY	Error code from Urban Mobility Scooter
FMS_ADBLUE_LEVEL_INDICATOR	FMS dashboard warning indicator for low AdBlue level
FMS_AFTERTREATMENT_1_DIESEL_EXHAUST_FLUID_TANK_1_LEVEL	Ratio of volume of diesel exhaust fluid (%)
AFTERTREATMENT_SCR_CATALYST_TANK_LEVEL_LITERS	AdBlue level in liters
AFTERTREATMENT_SCR_CATALYST_TANK_RANGE	AdBlue range (km)
FMS_AMBIENT_AIR_TEMPERATURE	Temperature of air surrounding vehicle (degree Celsius)
FMS_BATTERY_CHARGING_CONDITION_INDICATOR	FMS dashboard warning indicator for battery charging condition failure
FMS_BELLOW_PRESSURE_FRONT_AXLE_LEFT	Bellow pressure information (kPa)
FMS_BELLOW_PRESSURE_FRONT_AXLE_RIGHT	Bellow pressure information (kPa)
FMS_BELLOW_PRESSURE_REAR_AXLE_LEFT	Bellow pressure information (kPa)
FMS_BELLOW_PRESSURE_REAR_AXLE_RIGHT	Bellow pressure information (kPa)

CAN signal types	Description
FMS_BRAKE_SYSTEM_FAILURE_INDICATOR	FMS dashboard warning indicator for brake failure/brake system malfunction
FMS_CRUISE_ACTIVE	Cruise control state (On/Off)
FMS_DEFECT_AIRBAG_INDICATOR	FMS dashboard warning indicator for defect airbag
FMS_DOOR_STATUS	Summary of all door status
FMS_EMISSION_SYSTEM_FAILURE_INDICATOR	FMS dashboard warning indicator for engine emission system failure
FMS_ENGINE_COOLANT_LEVEL_INDICATOR	FMS dashboard warning indicator for low coolant level
FMS_ENGINE_COOLANT_TEMPERATURE	Temperature of liquid found in engine cooling system (°C)
FMS_ENGINE_HOURS	Accumulated time of operation of engine (h)
FMS_ENGINE_MIL_INDICATOR	FMS dashboard warning indicator for engine/MIL indicator
FMS_ENGINE_OIL_INDICATOR	FMS dashboard warning indicator for engine oil
FMS_ENGINE_OIL_LEVEL_INDICATOR	FMS dashboard warning indicator for low engine oil level
FMS_FUEL_LEVEL_PERCENTAGE	Ratio of volume of fuel to the total volume of fuel storage container (%)
FMS_GROSS_COMBINATION_VEHICLE_WEIGHT	The total weight of the truck and all attached trailers (kg)
FMS_MILEAGE_TO_SERVICE	The distance which can be travelled before the next service inspection is required (km)
FMS_PARKING_BRAKE	Switch signal which indicates when the parking brake is set (True/False)
FMS_PTO_ENGAGEMENT	At least one PTO engaged
FMS_PTO_STATE	Indicate the current state or mode of operation by the power takeoff device (On/Off)
FMS_SERVICE_BREAK_AIR_PRESSURE_1	Pneumatic pressure in the service brake circuit or reservoir (kPa)
FMS_SERVICE_BREAK_AIR_PRESSURE_2	Pneumatic pressure in the service brake circuit or reservoir (kPa)
FMS_TIRE_FAILURE_INDICATOR	FMS dashboard warning indicator for tire failures

CAN signal types	Description
FMS_TRANSMISSION_FAILURE_INDICATOR	FMS dashboard warning indicator for transmission failures
FMS_VEHICLE_MOTION	Indicates whether motion of the vehicle is detected or not (Yes/No)
FMS_VEHICLE_OVERSPEED	Indicates whether the vehicle is exceeding the legal speed limit set in the tachograph (Yes/No)
FMS_WORN_BRAKE_LININGS_INDICATOR	FMS dashboard warning indicator for worn brake linings
FUEL_LEVEL_PERCENTAGE	The percentage of available fuel in the tank (%)
ISOLATION_RESISTANCE	Resistance of the insulation of a battery
MIL_BATTERY_CELL_TOO_HIGH	MIL Battery cell is too high (on/off)
MIL_BATTERY_CELL_TOO_LOW	MIL Battery cell is too low (on/off)
MIL_BMS_INTERNAL_FAILURE	MIL BMS internal communication failure (on/off)
MIL_CHARGER_BATTERY_CONNECTION_FAILURE	MIL Charger battery connection failure (on/off)
MIL_CHARGER_COMMUNICATION_FAILURE	MIL Charger communication failure (on/off)
MIL_CHARGER_HARDWARE_FAILURE	MIL Charger hardware failure (on/off)
MIL_CHARGER_INPUT_VOLTAGE_FAULT	MIL Charger input voltage fault (on/off)
MIL_CHARGER_OUTPUT_OVERCURRENT	MIL Charger output overcurrent (on/off)
MIL_CHARGER_START_ERROR	MIL Charger start error (on/off)
MIL_CHARGER_TEMPERATURE_FAILURE	MIL Charger temperature failure (on/off)
MIL_CHARGER_VOLTAGE_LIMIT_FAULT	MIL Charger voltage limit fault (on/off)
MIL_DIFFERENCE_IN_BATTERY_TEMPERATURE	MIL Difference in battery temperature (on/off)
MIL_DIFFERENCE_IN_BATTERY_VOLTAGE	MIL Difference in battery voltage (on/off)
MILEAGE_TO_SERVICE	Distance to go before the vehicle needs service (km)
MIL_HIGH_CELL_VOLTAGE	MIL High cell voltage (on/off)
MIL_HIGH_DISCHARGE_CURRENT	MIL High discharge current (on/off)

CAN signal types	Description
MIL_INSULATION_RESIS-TANCE_TOO_LOW	MIL Insulation resistance too low (on/off)
MIL_LARGE_DIFFERENCE_IN_BAT-TERY_TEMPERATURE	MIL Large difference in battery temperature (on/off)
MIL_LARGE_DIFFERENCE_IN_BAT-TERY_VOLTAGE	MIL Large difference in battery voltage (on/off)
MIL_LOW_CELL_VOLTAGE	MIL Low cell voltage (on/off)
MIL_OVERCOOLING_ALERT	MIL Overcooling alert (on/off)
MIL_OVERHEAT_ALERT	MIL Overheat alert (on/off)
MIL_PARK_ASSIST_REAR_OUT-ER_LEFT_SENSOR_FAIL	MIL Assist - Park Control - Rear outer left sensor fail (on/off)
MIL_PARK_ASSIST_REAR_OUT-ER_RIGHT_SENSOR_FAIL	MIL Assist - Park Control - Rear outer right sensor fail (on/off)
MIL_PRE_CHARGE_FAULT	MIL Pre-charge fault (on/off)
MIL_SEVERE_DISCHARGE_OVERCUR-RENT	MIL Severe discharge overcurrent (on/off)
MIL_SEVERE_OVER_TEMPERATURE	MIL Severe over temperature alarm (on/off)
MIL_STATE_OF_CHARGE_LOW	MIL State of Charge low (on/off)
MIL_STATE_OF_CHARGE_VERY_LOW	MIL State of Charge very low (on/off)
MIL_TOTAL_VOLTAGE_TOO_HIGH	MIL Total voltage too high (on/off)
MIL_TOTAL_VOLTAGE_TOO_LOW	MIL Total voltage too low (on/off)
REMAINING_CHARGE_TIME	Remaining Charge Time (in electric and/or plugin hybrid vehicles) (min)
SYN_BATTERY_VOLTAGE	Battery Voltage (debounced)
TRIP_AVERAGE_ENERGY_CONSUMPTION	Average energy consumption from a trip computer. Value is resetable.

Revision history

WEBFLEET.connect

Webfleet, WEBFLEET.connect 1.76.0

© 2026 Bridgestone Mobility Solutions B.V. All rights reserved. Webfleet is a registered trademark of Bridgestone Mobility Solutions B.V. or one of its Affiliates.

No part may be reproduced except as authorized by written permission. The copyright and the foregoing restriction extend to reproduction in all media.

Revision history

Revision	Date	Description	Author
1.0.0	2004-10-08	Initial release	UM
1.1.2	2005-08-22	Added showTripReportExtern and added the field fuel_usage to showTripSummaryReportExtern output	CS
1.1.3	2005-09-29	Added sendDestinationOrderExtern	IM
1.1.4	2005-09-30	Added methods to deal with addresses (Addresses)	ML
1.1.5	2005-09-30	Added parameter addrgrpname to insertAddressExtern, added updateAddressExtern.	ML
1.1.6	2006-04-05	Editorial changes.	IM

Revision	Date	Description	Author
1.1.7	2006-04-07	Initial release after conversion to DocBook.	IM
1.1.8	2006-06-29	Extended parameter set for sendDestinationOrderExtern - added support for GO x10.	IM
1.1.9	2006-08-22	Changed the size of the order text parameter for sendDestinationOrderExtern, Renamed parameter from radium to radius for insertAddressExtern (documentation fix)	IM
1.2.0	2006-11-07	Added methods for order management: insertDestinationOrderExtern, updateOrderExtern, updateDestinationOrderExtern and cancelOrderExtern. Added some notes regarding performance impacts to showMessageReportExtern.	IM
1.2.1	2006-11-24	Corrected the range_pattern for user-defined from du to ud. Changed the parameter addrstate to addrcountry for insertAddressExtern and updateAddressExtern.	IM
1.2.2	2006-12-08	Added value documentation for tripmode for showTripReportExtern. Added value documentation for status, msgstatus and msgtype for showMessageReportExtern.	IM
1.2.3	2007-01-23	Minor editorial changes.	IM
1.3.1	2007-05-07	Extensions to showObjectReportExtern, now returns most recent position data in addition to object details. Added geocodeAddress for geocoding.	IM
1.3.2	2007-07-17	Changed DNS hostname for WEBFLEET.connect to csv.business.tomtom.com. Added documentation of parameter constraints for showMessageReportExtern. Removed filter parameter objectgroupname from several report methods that return message-based data.	IM
1.3.3	2007-07-20	Added examples and corrections to the date format requirements for rangefrom_string and rangeto_string. Changed the result column name for showOrderReportExtern from orderno to orderid. Added some more explanations on how to make HTTP requests.	IM
1.3.4	2007-10-10	Added showOrderMessageReportExtern. Removed `userstate` from the result set documentation for showMessageReportExtern (for backward compatibility, the column might still be part of the result, but the contents – if any – should be considered as reserved). Minor additions to the documentation of some result columns.	IM
1.3.5	2007-10-12	Added reassignOrderExtern	IM

Revision	Date	Description	Author
1.3.6	2008-01-24	New request limits (Request limits) based on functional groups. Updated documentation for showMessageReportExtern with new explanations for msgid-based replication and parameter limits.	IM
1.3.7	2008-05-08	Updated documentation for showTripReportExtern with new explanations for tripid-based replication and interpretation of date range filters.	IM
1.3.8	2008-07-29	Updated documentation for insertAddressExtern and updateAddressExtern - added description for the parameter visible. Updated documentation for objectno/objectid and orderno/orderid with details about parameter size. Updated documentation for userstate result element for showOrderMessageReportExtern. Updated Creating a user and assigning rights for the new Webfleet UI. Added useISO8601 to the general parameters (General parameters).	IM
1.4.0	2008-10-28	Added new details to showObjectReportExtern on page 56: current destination navigated to, eta, current order (if any) and matched address number for the vehicle's current position (all subject to support by the connected device). Added a new section about queue-based access to messages (Overview of queues). Removed documentation for showMessageReportExtern because its use is deprecated and will not be available after 2008-12-31. Added appendixes with operation result codes and recommended resources.	IM
1.5.0	2009-02-16	Added support for address colours to insertAddressExtern, updateAddressExtern and showAddressReportExtern. Extended the filter parameter set for showEventReportExtern and added methods to flag events as acknowledged and as resolved. Added a set of methods for driver management, see Orders. Updated result columns for the object a driver is currently signed onto to showDriverReportExtern.	IM
1.5.1	2009-06-18	Added support for address regions to insertAddressExtern, updateAddressExtern and showAddressReportExtern. Added examples to sendDestinationOrderExtern and showOrderReportExtern.	IM
1.5.2	2009-07-02	Added deleteOrderExtern.	IM

Revision	Date	Description	Author
1.6.0	2010-01-06	Minor editorial changes. Added new groups to request limits for driver and event methods. Added section Making SOAP requests. Added documentation for parameter restrictions on showTripReportExtern. Added support for the digital tachograph to showObjectReportExtern, popQueueMessagesExtern and all driver management methods. Removed documentation for `showOrderMessageReportExtern` (its use is still supported but it should not be used in new work anymore).	IM
1.6.1	2010-02-18	Added new methods: clearOrdersExtern, clearTextMessagesExtern. Replaced working and usage time reports with showWorkingTimes.	IM
1.6.2	2010-02-25	Added a clarification regarding supported date/time formats to showWorkingTimes. Expanded the list of documented error codes in Appendix A, Operation Return Codes.	IM
1.7.0	2010-05-20	Updated the request limit table with new limits for queue access and queue management methods. Expanded the list of documented error codes in Appendix A: Operation response codes. Added new data elements for the TomTom Remote LINK (see Drivers, popQueueMessagesExtern). Added new method to support inventory of devices: showContracts. Added updateVehicle.	IM
1.7.1	2010-06-17	Removed documentation for `showOrderMessageReportExtern` and `showStoplistReportExtern` (their use is still supported but it should not be used in new work anymore). Clarification on the message classes that can be used for queue creation. Updated the user creation guide to point out that full permissions on orders might be necessary for certain methods to succeed.	IM
1.7.2	2010-09-02	Clarification on the data size for `objectno` (10 characters). Added working time information to showDriverReportExtern. Added `event_time` to popQueueMessagesExtern. Fixed and expanded the `source_device` enumeration for popQueueMessagesExtern. Added Bluetooth address of a paired Remote LINK device (`rll_btaddress`) to showObjectReportExtern. Clarification on the supported devices for deleteOrderExtern. Removed documentation for `showPersonReportExtern` and `showTrailerReportExtern` (their use is still supported but it should not be used in new work anymore).	IM

Revision	Date	Description	Author
1.7.3	2010-12-15	Removed function `geocodeAddressExtern` from the Reference document and added geocodeAddress. Added new action calcRouteSimpleExtern. Added new result parameter to showWorkingTimes. Added new section Trips and working times and new actions showSpeedingEvents, showAccelerationEvents, showStandStills, showIdleExceptions. Added new parameters to updateVehicle. Added new JSON objects to `surplus_data` and new message classes to popQueueMessagesExtern. Added parameters to showTripReportExtern. Added parameters to showTripSummaryReportExtern. Added parameters to showVehicleReportExtern. Corrected parameter namin in showDriverReportExtern. Renamed chapters and changed order. Other minor editorial changes.	RH
1.7.4	2011-02-24	Changed description for `avg_accel` and description of showAccelerationEvents. Updated result list for showTripSummaryReportExtern. Updated parameters list for updateDestinationOrderExtern. Added advise on the use of HTTP authentication in Making HTTP requests. Edited parameter names for showEventReportExtern. Added new parameters to showTripReportExtern. Added description for `pin` and `code` for Drivers commands. Extended list of Messages types for popQueueMessagesExtern. Extended list of Operation response codes. Extended information about case-sensitivity of inbound parameters of multiple commands. Added limits for calcRouteSimpleExtern to Default request limit configuration.	RH

Revision	Date	Description	Author
1.8.0	2011-05-24	New end point URLs see Making requests to WEBFLEET.connect. Added new authentication parameter `apikey` to Common parameters. Extended the list values for date range filter parameter `range_pattern` in Common parameters. Replaced old function `showObjectSpeedingEvents` with new function showSpeedingEvents. Replaced old function `showObjectAccelerationEvents` with new function showAccelerationEvents. Added query and result parameters to showIdleExceptions. Added parameter to showOrderReportExtern. Added result parameters to showObjectReportExtern. Added new functions showObjectGroups, showObjectGroupObjects, attachObjectToGroup, detachObjectFromGroup, insertObjectGroup, deleteObjectGroup, updateObjectGroup. Removed old functions `showObjectGroupReportExtern` and `showObjectGroupObjectReportExtern`. Added new chapter Configuration and security with new function showSettings. Added new function showOptiDriveIndicator. Exended list of Response codes. Added new results to showVehicleReportExtern. Added list of sperseded functionality Appendix C: Superseded functionality indicating the announcement date, the date of end of life and possible alternatives.	RH

Revision	Date	Description	Author
1.9.0	2011-08-23	New functions attachDriverToGroup, detachDriverFromGroup, deleteDriverGroup, insertDriverGroup, showDriverGroups, showDriverGroupDrivers, updateDriverGroup, showUsers in the newly introduced chapter User management. Removed `showLogbookReportExtern`. New parameter to filter on entities that are not assigned to certain entity groups for showObjectReportExtern, showDriverReportExtern and showAddressReportExtern. Extended result list for showObjectReportExtern with `drivername`, `codrivername` and `pos_time`. New result `objectcount` in showObjectGroups. Unformatted latitude/longitude results with showAddressReportExtern. Updated Appendix C: Superseded functionality, please read carefully. Added important notes to Making HTTP requests. Introduced new general parameter `columnfilter` to be used with all functions, see General parameters. Added technical details, such as SOAP function name and endpoint address and request limits to each function. Most important outbound messages relating to text messages and order messages are now available in the queue system, see Message queues. Other miscellaneous edits. Added note about how orders appear on the TomTom navigation device to sendOrderExtern and sendDestinationOrderExtern. Added instructions for integrators working with SOAP and Visual Basic with regards to WEBFLEET.connect addresses and MTOM encoding support, see Making SOAP requests. Added error codes to Response codes. Added a new section called Terminology explaining terms used in the context Webfleet, WEBFLEET.connect, and Webfleet solutions, products, and features.Edits to the list of Messages types. Information about Synchronising driver lists.	RH

Revision	Date	Description	Author
1.10.0	2011-11-09	New functions insertMaintenanceSchedule, updateMaintenanceSchedule, deleteMaintenanceSchedule, showMaintenanceSchedules, showMaintenanceTasks, resolveMaintenanceTask, updateLogbook, showLogbook, showLogbookHistory. Added `co2` to result list of showTripSummaryReportExtern. showWorkingTimes supports now all time/date values listed in Date range filter parameters. Contact and driver details that relate to an order are included in the result of showOrderReportExtern. Added `drivertelmobile` and `codrivertelmobile` to the result list of showObjectReportExtern. Added new parameter `addrno` to insertDriverExtern, updateDriverExtern and showDriverReportExtern. Added parameters `addrno`, `positiony` and `positionx` to showDriverReportExtern. The working state of a vehicle is indicated even when no driver has signed on using showObjectReportExtern. Added information about input and return date formats to Making HTTP requests, Making SOAP requests and General parameters. Distinction between SOAP and CSV General parameters.Superseded functionality - `geocodeLocation` will be superseded together with `geocodeAddressExtern`.Superseded functionality - `showLogbookReportExtern` will be superseded. Added new error codes to Appendix A: Operation return codes.	RH
1.10.1	2011-11-09	Adjusted maximum values for parameters `intervalodometer`, `remindingodometer`, `plannedexecodometer` of insertMaintenanceSchedule and updateMaintenanceSchedule. Minor other edits.	RH
1.11.0	2012-01-30	Added new functions showAccountOrderStates, updateAccountOrderState, switchOutput, showTracks, attachDriverToVehicle, detachDriverFromVehicle. Added new parameters `name2` and `name3` to insertDriverExtern and updateDriverExtern. Added new column `manualassignment` to result list of showDriverReportExtern. Added new parameter `useorderstates` that defines the order states that appear in the order workflow per order to sendOrderExtern and sendDestinationOrderExtern. Added two new parameters `latitude_mdeg` and `longitude_mdeg`, with unit of measurement in micro degrees, to the result list of showObjectReportExtern and showEventReportExtern. Added `useUTF8` parameter to General parameters. Added to the list of terms explained in the Terminology chapter the section Vehicle maintenance including the definitions of Maintenance schedule and Maintenance task. Other minor editorial changes.	RH

Revision	Date	Description	Author
1.12.0	2012-04-13	Added new parameter `target_username` and result columns to showSettings. Added showAccountOrderAutomations and updateAccountOrderAutomation. Added parameter `orderautomations` to sendOrderExtern, sendDestinationOrderExtern, assignOrderExtern, reassignOrderExtern, updateOrderExtern, updateDestinationOrderExtern. Added parameter `odometer` to updateVehicle. Added parameter `objectuid to` showObjectReportExtern, showVehicleReportExtern, showContracts, updateVehicle, showObjectGroupObjects, attachObjectToGroup, detachObjectFromGroup, switchOutput. Added result `objectuid` to showObjectReportExtern, showVehicleReportExtern, showContracts, showObjectGroupObjects. Added parameter `objectuid` to showTracks. geocodeAddress parameter `provider` can now also be the new TomTom Geo Coder Version 2.	RH
1.12.1	30.05.2012	Introduces the support of the LINK 510 (see Messages types and switchOutput) and other minor edits.	RH
1.12.2	15.06.2012	Introduction of itinerary orders: Added waypoint parameter `wp` (waypoint) to sendDestinationOrderExtern, updateDestinationOrderExtern, insertDestinationOrderExtern. New action showOrderWaypoints. Edits to list of Messages types and Response codes. Making HTTP requests - added HTTP POST support for a limited number of actions)	RH

Revision	Date	Description	Author
1.13.0	06.08.2012	Added FMS and acceleration `surplus_data` to popQueueMessagesExtern. New result `order_addrno` in popQueueMessagesExtern. showOrderReportExtern - Added destination address details to result list. showNearestVehicles delivers a list of vehicles that are within a defined distance to a specific location. showWakeupTimers delivers a list of days and times when the LINK 3xx/510 is set to automatically wake up. updateWakeupTimers let s you define new and change existing days and times when the LINK 3xx/510 device shall automatically wake up. Added new parameter `rpmlimit` to updateVehicle that sets a maximum number of revolutions per minute for the vehicle s engine that is being used for reporting on rpm violations. Added new results `vh_rpmlimit` and `fl_rpmlimit` to showVehicleReportExtern. Added new result `fuellevel` to showObjectReportExtern. showDriverReportExtern - Added new result `pin` that is used by the driver to logon to the Driver Terminal. updateLogbookMode changes the logbook mode on the LINK 3xx/510 device. Added new parameter `acceleration_types` to showAccelerationEvents. Added the parameter `objectuid as an alternative to objectno to`sendOrderExtern, insertDestinationOrderExtern, sendDestinationOrderExtern, assignOrderExtern, reassignOrderExtern, clearOrdersExtern, showOrderReportExtern, attachDriverToVehicle, detachDriverFromVehicle, showTripReportExtern, showTripSummaryReportExtern, showLogbook, showLogbookHistory, showWorkingTimes, showStandStills, showIdleExceptions, sendTextMessageExtern, clearTextMessagesExtern, insertMaintenanceSchedule, showMaintenanceSchedules, showMaintenanceTasks, showIOReportExtern, showAccelerationEvents, showSpeedingEvents, showEventReportExtern. Added new result `objectuid` to showOrderReportExtern, showDriverReportExtern, showTripReportExtern, showTripSummaryReportExtern, showLogbook, showLogbookHistory, showWorkingTimes, showStandStills, showIdleExceptions, popQueueMessagesExtern, showMaintenanceSchedules, showMaintenanceTasks, showIOReportExtern, showAccelerationEvents, showSpeedingEvents, showEventReportExtern. Added new error codes to Appendix A: Operation return codes.	RH

Revision	Date	Description	Author
1.14.0	06.11.2012	New action getObjectKPIs. New parameter `modified_since` and new results `start_latitude`, `start_longitude`, `end_latitude`, `end_longitude`, `avg_speed`, `max_speed`, `fuel_usage`, `co2`, `end_addrname1`, `end_addrname2`, `end_addrname3`, `end_addrcontact` for showLogbook. showTracks has the new result parameters `odometer`, `country`, `state`. Added information on limitations to the description of updateLogbookMode. New actions createSession, terminateSession, changePassword. Added new parameter `sessiontoken` to Authentication parameters. Extended `type`-list within in the result `additional_information` of geocodeAddress. showAccelerationEvents has new results `duration`, `severity` and `roadspeedlimit`. showSpeedingEvents has new results `duration` and `distance`. For showOrderReportExtern, Date range filter parameters are not required if `orderid` is indicated. New Date range filter parameters `sessiontoken`, 2020, `month`, `day`. Updated Operation response codes. Updated the description of `accel_dir` in showAccelerationEvents.	RH
1.15.1	13.03.2013	Added new section LINK.connect. Added new actions sendAuxDeviceData, getLocalAuxDeviceConfig, configureLocalAuxDevice, getRemoteAuxDeviceConfig, configureRemoteAuxDevice, removeRemoteAuxDeviceConfig, clearAuxDeviceDataQueue, resetAuxDeviceData. Added to popQueueMessagesExtern new LINK.connect related message types to Messages types and added new JSON objects. Added new section Reporting. Added new actions getArchivedReportList, getArchivedReport, deleteArchivedReport, getReportList, createReport, sendReportViaMail. Added new section Appendix E: Using MTOM with SOAP. Added new parameter deleteaddresses to deleteAddressGroupExtern, that deletes all addresses assigned to the indicated address group too. The `odometer` value can now be updated for navigation devices too using updateVehicle. Added new result `referenceno` to showContracts. Edits to Appendix A: Operation return codes.	RH

Revision	Date	Description	Author
1.16.0	10.04.2013	Added actions showMessages, getObjectFeatures. Added new property `dur` to table Parameters specific to acc_events in popQueueMessagesExtern. Updated updateDestinationOrderExtern - `ordertext` is optional. Introduced new parameter/result `driveruid/codriveruid` to popQueueMessagesExtern, showObjectReportExtern, showDriverReportExtern, updateDriverExtern, deleteDriverExtern, showOptiDriveIndicator, showDriverGroupDrivers, attachDriverToGroup, detachDriverFromGroup, attachDriverToVehicle, detachDriverFromVehicle, showTripReportExtern, showWorkingTimes, showIdleExceptions, showAccelerationEvents, showSpeedingEvents. Introduced new parameter/result `externalid` to createSession, showVehicleReportExtern, showObjectReportExtern, showTripReportExtern, showStandStills, showTracks, showAccelerationEvents, showSpeedingEvents, getObjectKPIs. Parameter `msgclass` has an additional value for LINK.connect messages in the following actions createQueueExtern, deleteQueueExtern, popQueueMessagesExtern, ackQueueMessagesExtern.	RH
1.17.0	21.08.2013	Introduced new action updateContractInfo which allows to assign and change an customer specific reference number per object. Introduced new parameter `freetext` to geocodeAddress. Introduced new parameter `ep_btaddess` and `rll_btaddress` to updateVehicle. Introduced new result `passwordexpiration` for showUsers, createSession. Introduced new result `creationtype` for showMaintenanceSchedules. Added `troublecodes` to `surplus_data` in popQueueMessagesExtern and JSON structure Parameters specific to troublecodes. New msg_type `80000880` indicating that the list of current vehicle Diagnostic Trouble Codes has changed. Introduced new result `obu_btaddress` to showVehicleReportExtern. Introduced `driveruid` to result list of showOptiDriveIndicator, showDriverGroupDrivers, showTripReportExtern, showLogbook, showWorkingTimes, showIdleExceptions, showAccelerationEvents, showSpeedingEvents, showOrderReportExtern. Added SOAP message size quota in Visual Studio to Appendix D: Solving known issues.	RH
1.18.0	21.08.2013	Introduced new chapter Areas with the following actions: getAreas, getAreaPoints, deleteArea, insertArea, updateArea, getAreaAssignments, insertAreaAssignment, deleteAreaAssignment, getAreaSchedules, insertAreaSchedule, deleteAreaSchedule.	RH

Revision	Date	Description	Author
1.19.0	15.02.2014	Introduced new functions insertUser, updateUser, deleteUser, getUserRights, setUserRight, removeUserRight, getStatusMessages, setVehicleConfig, getAccountStatusMessages. Added new result `useruid` to showUsers. Added new parameters `drivergroupname` and `objectgroupname` to showWorkingTimes. Added new parameters `addrnr` and `addruid` and result `addruid` to showAddressReportExtern. Introduced result `addressgroupuid` to showAddressGroupReportExtern. Added parameter `addruid` to updateAddressExtern. In insertDriverExtern the parameters `code` and `pin` are optional. showDriverGroups returns `drivergroupuid`. WEBFLEET.connect supports JSON throughout the whole range of functions. To define the response format the parameter `outputformat` was introduced to the list of General parameters.	RH
1.19.1	31.03.2014	TomTom Business Solutions was renamed to TomTom Telematics.	RH
1.20.0	02.04.2014	The following functions are new getEventForwardConfigs, getEventForwardConfigRecipients, insertEventForwardConfig, updateEventForwardConfig, deleteEventForwardConfig, resetUserRights. Added output parameters `addruid` and `addressgroupuid` to showAddressGroupAddressReportExtern. Added `addruid` to updateAddressExtern, deleteAddressExtern, attachAddressToGroupExtern, detachAddressFromGroupExtern. Added parameter `addrgrpuid` to insertAddressExtern, attachAddressToGroupExtern, detachAddressFromGroupExtern, deleteAddressGroupExtern. Parameters `notify_enabled` and `notify_leadtime` in updateDestinationOrderExtern are not supported anymore. Added new paramter `duetype` to showMaintenanceTasks. Added right level `edit_usersettings` to List of supported right levels	RH
1.21.0	01.08.2014	Webfleet error codes are now also returned in HTTP header fields, Making HTTP requests. Added new result `pnd_mapversion` to showContracts. Added new parameter `ep_type` to updateVehicle and showVehicleReportExtern.	RH
1.21.1	15.08.2014	The process of requesting access to APIs has changed, Access to WEBFLEET.connect with API Key	RH

Revision	Date	Description	Author
1.23.0	05.11.2014	Introduced sendBinaryMessage, resetBinaryMessages and clearBinaryMessages. Introduced sendBinaryMessage, resetBinaryMessages and clearBinaryMessages that allow to send and manage messages transmitting raw binary data that can be used by third party apps installed on the driver terminal. popQueueMessagesExtern: Added new `surplus_data` relevant for the transfer of binary data for third party apps installed on the driver terminal. Introduced getRemainingDrivingTimesEU that assists fleet managers to determine remaining driving times of their truck drivers. Introduced getDriverKPIs and getObjectKPIs to retrieve the Key Performance Indicators of individual drivers or objects. getObjectKPIs is the successor action of `getKPIs`. All calls to `getKPIs` will be redirected to getObjectKPIs. With the new parameter (and result) `color` in getAreas, insertArea, updateArea you can define and retrieve the colour of an area as a 6-digit hex RGB-color code. The actions insertUser and updateUser come with the new parameter `interfacestyle`, that lets you define which type of interface a user is supposed to see/use in Webfleet. The new `profile` values `invoices_user` and `report_recipient` have been added to the actions insertUser and updateUser. Extended the List of supported right levels to include the rights levels `report_read_access`, `enable_device_configuration_management` and `report_edit_access`. Added new result `msisdn` to showContracts. Added `objectgroupid` to the results of showObjectGroupObjects and to the list of parameters of showObjectGroups. The new getVehicleConfig lets you retrieve LINK specific configurations. Introduced setStatusMessages and setAccountStatusMessages to set the predefined text and order status messages for individual vehicles and for the Webfleet account. The newly introduced `mapcode` parameter offers an alternative to longitude and latitude when determining locations using sendDestinationOrderExtern, insertDestinationOrderExtern, updateDestinationOrderExtern, insertAddressExtern, updateAddressExtern. Added new result value `mapcode` to showOrderReportExtern, showAddressReportExtern.	RH
1.24.0	19.01.2015	Assigning the relevant WEBFLEET.connect user rights to Webfleet users in the Webfleet user interface has changed. The instructions in Creating a user and assigning rights have been amended to reflect these developments. showEventReportExtern: Added new result `pos_time` that indicates the time of the last known position when the event occured.	RH

Revision	Date	Description	Author
1.25.0	28.04.2015	We have extended the result list for showOptiDriveIndicator to include all relevant parameters for **OptiDrive 360**. We have extended the result list for showSettings to include the four new parameters that show the weights of coasting, constant speed, green speed and gear shift.	RH
1.26.0	16.12.2015	**Itinerary orders** - PRO 7100/7150, PRO 9100/9150 and PRO 5150 support up to 1000 waypoints/order. PRO 8270/8275 support up to 250 waypoints/order. PRO 5250, PRO 7250 do not support itinerary orders. The affected functions are: sendDestinationOrderExtern, updateDestinationOrderExtern, insertDestinationOrderExtern. showVehicleReportExtern and updateVehicle have the new optional parameter `manufacturedyear` that indicates the year a specific vehicle was manufactured. With the new parameter `intervaltimetype` in insertMaintenanceSchedule and updateMaintenanceSchedule you can define maintenance schedules timely more flexible. `intervaltimetype` lets you differentiate between days, weeks, months and year. `intervaltimetype` has been added to the result list of showMaintenanceSchedules too. Minor edits and improvements.	RH
1.27.0	31.03.2016	Using showMaintenanceTasks you can retrieve all relevant information on the next maintenance tasks as it is shown in the Webfleet user interface. There is no need to make additional calls using showMaintenanceSchedules anymore. For this we have added the parameter `scheduletype` and we have added `schedulename`, `scheduletype` and `schedulecreationtype` to the result list. Improved support of various driver authentication devices by Adding parameter `driver_key` to insertDriverExtern and updateDriverExtern; Adding `driver_keys` to result list of showDriverReportExtern; Adding `driverkey_deviceaddress` to the result list of showObjectReportExtern; Adding `driverkey_deviceaddress` and `driverkey` to the result list of popQueueMessagesExtern. Minor edits and improvements.	RH

Revision	Date	Description	Author
1.28.0	01.05.2016	Using showMaintenanceTasks you can retrieve all relevant information on the next maintenance tasks as it is shown in the Webfleet user interface. There is no need to make additional calls using showMaintenanceSchedules anymore. For this we have added the parameter `scheduletype` and we have added `schedulename`, `scheduletype` and `schedulecreationtype` to the result list. Improved support of various driver authetication devices by: • Adding parameter `driver_key` to insertDriverExtern and updateDriverExtern • Adding `driver_keys` to result list of showDriverReportExtern • Adding `driverkey_deviceaddress` to the result list of showObjectReportExtern • Adding `driverkey_deviceaddress` and `driverkey` to the result list of popQueueMessagesExtern • Minor edits and improvements	RH
1.29.0	30.06.2016	showUsers returns the `email` address of the user. Minor edits and improvements	RH
1.30.1	01.09.2016	showTripReportExtern returns the following OptiDrive 360 related parameters: `optidrive_indicator`, `speeding_indicator`, `drivingevents_indicator`, `idling_indicator`, `fuelusage_indicator`, `coasting_indicator`, `constant_speed_indicator`, `green_speed_indicator`, `high_revving_indicator` showObjectReportExtern returns the parameter `fuelLevelMilliliters`.	RH
1.31.0	30.11.2016	Introduced the new `fuelreference` parameter to both showVehicleReportExtern and updateVehicle. Using this parameter you select which source shall be used as fuel reference. geocodeAddress - parameter `provider` not supported anymore. insertDriverExtern and updateDriverExtern support both driver key types Remote LINK and RFID used with the PRO 2020 Driver Terminal. endpoints have changed from `soap.business.tomtom.com` to `soap.telematics.tomtom.com` and from `csv.business.tomtom.com` to `csv.telematics.tomtom.com`	RH
1.32.0	01.09.2017	Migrated document from legacy system. Various edits and improvements. **Not released** due to impending bug fix.	CH

Revision	Date	Description	Author
1.33.0	05.10.2017	Added chapters Getting started with HTTP requests and Getting started with SOAP requests to Programming Guide. Removed colour `ivory`. This affects showVehicleReportExtern and updateVehicle. The list of message types was updated. It now shows the full `msg_type` numbers. Additionally, you can see now on which message queues a message is available. Double entries in the parameter tables of popQueueMessagesExtern have been removed. Added note on usage for updateAddressExtern. If certain parameters are not specified in a request, these parameters are deleted. showDriverReportExtern: The parameter `driver_keys` only appears in a response. It was moved from the **Parameters** to the **Result** table. The colour `ivory` has been removed. This affects showVehicleReportExtern and updateVehicle. The GO 715 device has been discontinued. Information on this device was removed. Added the value '`3` – Worker' to the parameter `role` in the **Result** table of showWorkingTimes. Added the value '`3` – Worker' to the parameter `signonrole` in the **Result** table of showDriverReportExtern. Minor edits and improvements.	CH
1.35.0	26.02.2018	popQueueMessagesExtern - introduced the new parameter `location_params`, which is currently being tested. The list of message types was updated (Acceleration event and Working time event). showObjectReportExtern - renamed parameter `fuelLevelMilliliters` to `fuellevel_milliliters` Minor edits and improvements.	RH
1.37.0	12.07.2018	Added the new topic HTTP request encoding showOrderWaypoints - parameter `orderid` is mandatory Minor edit and improvements	RH
1.39.0	09.10.2018	Added the new chapter Plugins including the actions insertExternalEvent and setExternalObjectData to be used for the addtional feature **Webfleet Plugin**.	RH
1.40.1	08.01.2019	Enhanced SOAP documentation - added new SOAP parameters topic, added SOAP request examples to showObjectReportExtern, sendDestinationOrderExtern and showOrderReportExtern	RH
1.40.1 - update	01.02.2019	Minor editoral changes.	RH

Revision	Date	Description	Author
1.40.1 - update	19.02.2019	Added some message types.	RH
1.42.0	09.04.2019	Removed unsupported inside/outside value for `notificationmode` from insertArea and updateArea. Added new surplus data to popQueueMessageExtern. `er_tripdata` provides additional trip information for optimal eco reporting.	RH
1.43.0	20.05.2019	The handling of rights to see position time and user information for showEventReportExtern has changed and can return empty results for `pos_time`, `ackuser` and `resuser` paramters.	RH
1.44.0	16.07.2019	Minor changes.	RH
1.45.0	01.08.2019	Introducing two new actions getDriverRdtRules and updateDriverRdtRules Minor changes and edits.	RH
1.46.0	21.10.2019	Introducing new return value `engine_operation_time` to action showObjectReportExtern. updateVehicle introduces the option to update the vehicle `objectno`.	RH
1.47.0	29.01.2020	`location` parameter not supported anymore in all actions related to addresses. Introduced new message types on cruise control, engine status and engine oil and coolant temperature. The new surplus data `ioname` for popQueueMessegesExtern provides the name of the input or output when switched on or off. Introducing the two new actions getObjectCanSignals and getObjectCanMalfunctions, that let you retrieve the latest received CAN signal values for one or more objects in an account and the current state of CAN based malfunctions of an object. Using the newly introduced parameter `output_name` for action switchOutput you can specify the output instead of relying on the order that is pre-configured. URLs, endpoints and references to the company and products have been amended according to the migration from TomTom Telematics to Webfleet Soutions.	RH
1.48.0	01.04.2020	Introduced new action showDigitalInputStateMileage to retrieve all possible switching state combinations of digital inputs along with mileage information. Minor other edits.	RH

Revision	Date	Description	Author
1.49.0	27.07.2020	Disabling MTOM – You can enforce non-MTOM responses in the SOAP API. vehicleUpdate supports all vehicle types that are available in the vehicle icon gallery in Webfleet. Some vehicle types are not supported for all colours.	RH
1.50.0	28.08.2020	Introduced new action getElectricVehicleData. Introduced new action updateLogbookDriver. Added missing features for action getObjectFeatures.	HR
1.51.0	09.12.2020	Documented parameter restrictions for actions showAccelerationEvents and showSpeedingEvents. Removed documentation about how to create users using Webfleet classic. Added PRO 8375 and PRO 8475 to sendDestinationOrderExtern, updateDestinationOrderExtern, insertDestinationOrderExtern. Adapted description for response code 25001.	HR
1.53.0	25.05.2021	Documented new result values of action getRemainingDrivingTimesEU. Added the fuel type values for electric and hybrid vehicles to showVehicleReportExternand updateVehicle	HR
1.55.0	30.11.2021	Introduced new action getActiveAssetCouplings Documented new malfunction codes for getObjectCanMalfunctions Documented new return parameters for getObjectFeatures Documented new parameters for insertDriverExtern, updateDriverExtern and showDriverReportExtern	HR

Revision	Date	Description	Author
1.56.0	17.03.2022	Extended information about HD Tracking at Tracking and tracing Documented message types at Message types Extended description of parameter engine_operation_time of action showObjectReportExtern Added documentation of parameter `identnumber` for actions showVehicleReportExtern, updateVehicle Documented new parameter `accelerationvehicletype` for actions showVehicleReportExtern, updateVehicle Added PRO 7350 and PRO 5350 to documentation of parameter `WP` for actions sendDestinationOrderExtern, updateDestinationOrderExternand insertDestinationOrderExtern Documented that use of orders waypoint parameters `notify` and `visible` is discouraged at Parameters in wp Documented that LINK.connect feature is necessary to use action sendBinaryMessage Fixed typo in parameter `idling_wastedfuel` for action showOptiDriveIndicator Documented new result parameter `drivergroupuid` for action showDriverGroupDrivers Extended documentation of parameter `radius` for action insertAddressExternand updateAddressExtern Documented new result parameter `energy_usage` for actions showTripReportExtern, showTripSummaryReportExtern, showTracks and showLogbook Documented new result parameter `energy_consumption` for action showSettings Removed filter restriction for action showWorkingTimes Corrected name of parameter `resettoaccount` for action setStatusMessages Documented missing `scheduletype` 25 & 26 for actions insertMaintenanceSchedule, updateMaintenanceSchedule, showMaintenanceSchedules and showMaintenanceTasks Documented missing `response code` 9500 at Response codes	MO
1.57.0	22.07.2022	Added new action getChargerConnections New odometer properties for several endpoints Renamed licence parameters to correct values in insertDriverExtern and updateDriverExtern Added documentation on how to delete driver license in updateDriverExtern	
1.58.0	19.09.2022	Added documentation for HTTP Basic Auth. Streamlined documentation for driver license data Added missing CAN signals to documentation	

Revision	Date	Description	Author
1.62.0	07.06.2023	Documented new driver number length limit. Added getCrashLog documentation. Fixed getRemainingDrivingTimesEU documentation. Added notes regarding missing order destination to sendOrderExtern documentation.	
1.63.0	10.07.2023	Removed unsupported value `RPM` from CAN signal types list. Added new actions getOrderStatusPageURLand removeOrderStatusPageURL Removed documentation for unsupported `pos_params` parameter for action popQueueMessagesExtern. Added 3 new parameters to action getElectricVehicleData	
1.64.0	11.08.2023	Added new actions for order attachments, `downloadOrderAttachment` , `downloadOrderAttachmentThumbnail`, `uploadOrderAttachment`, `getOrderAttachmentsMetadata`, `deleteOrderAttachment`. Added new action `getLoadData`. New return value `area_speedlimit` for `showSpeedingEvents`. Added new access rights for Webfleet Video. Added new fuel types. Added some new error response codes. Minor adaptions and fixes.	HR
1.65.0	06.02.2024	New return values related to electric vehicles for showTripReportExtern and showLogbook. Added new parameter `travel_mode` to calcRouteSimpleExtern. New return value `kwh_charged` for getChargerConnections.	HR
1.66.0	18.07.2024	New energy consumption related KPI added to showTripReportExtern, getDriverKPIs, getObjectKPIs, showOptiDriveIndicator Action getObjectFeatures now also reports `hdtracking_4` and `hdtracking_5` in the feature details. Documented message for crash sensor triggered in Message types for `popQueueMessagesExtern`.	

Revision	Date	Description	Author
1.67.0	16.10.2024	New action getOptiDriveProfiles. Added feature names `vehiclechecklist`, `proappnav`, `coldchain`, `oemconnect` to action getObjectFeatures. Added `cruise_control_indicator` and `video_indicator` to actions showTripReportExtern and showOptiDriveIndicator. Added deprecation notice for credentials in URLs. Minor documentation corrections.	HR
1.67.1	27.01.2025	Added hints about asynchronous order processing.	HR
1.69.0	14.03.2025	Action sendDestinationOrderExtern has a new parameter for ePOD support. Added new order states for ePOD to result of showOrderReportExtern. A new parameter for getActiveAssetCouplings allows to return the previous coupling. Minor documentation changes. EOL notice for URL credentials Added new CAN signal types for AdBlue level and AdBlue range.	HR
1.70.0	20.05.2025	Added new actions getOrderEpodand downloadOrderEpodSignature. Added new order states for ePOD to sendDestinationOrderExtern. Clarify default behaviour for parameters notify_leadtime and notify_enabled for sendDestinationOrderExtern. Results of showTripReportExtern and showLogbook now contain energy consumption and recovery. Trip messages in message queues now contain energy consumption/recovery and battery levels. New action getTelemetryDeviceBatteryState returns the battery state of telematics devices which can operate in battery-powered mode (e.g. LINK 340, LINK 350).	HR
1.71.0	16.06.2025	Added new action downloadOrderEpodPhoto. Added ePOD related order states to updateAccountOrderState and showAccountOrderStates. Removed BETA test hints from ePOD related documentation. Added new Cold Chain related actions getCurrentTemperatureData, getHistoricalTemperatureData, getCurrentRefrigeratedDoorStatusData and getHistoricalRefrigeratedDoorStatusData. Added description about how certain parameters affect routing behaviour in Work App for sendDestinationOrderExtern, insertDestinationOrderExtern and updateDestinationOrderExtern.	HR

Revision	Date	Description	Author
1.72.0	07.10.2025	New action getObjectDTCs returns diagnostic trouble code (DTC) information for an object. New action getDiagnosticFaults retrieves diagnostic event data for an object. New actions to manage loaded goods: createOrderLoadedGoods, updateOrderLoadedGoods, deleteOrderLoadedGoods and getOrderLoadedGoods. New parameter `contactemail` in insertDestinationOrderExtern, updateDestinationOrderExtern and sendDestinationOrderExtern. New actions getCurrentColdChainSensorData, getCurrentColdChainStateData, getHistoricalColdChainSensorData, getHistoricalColdChainStateData. Documented under which conditions parameters are mandatory in updateLogbook. Adapted various code examples to reflect deprecation of URL credentials for CSV/JSON API.	HR
1.73.0	09.12.2025	New fuel types for showVehicleReportExtern, updateVehicle, showTripReportExtern, showStandStills, showIdleExceptions. The new access rights `read_access_orders_when_vehicle_assigned` and `full_access_orders_when_vehicle_assigned` can now be configured via setUserRight action. Adapted order text length limit to 1000. Added information about line breaks in order and message text. Fixed naming of `fuelusage` parameter for showTracks	HR
1.74.0	16.03.2026	Updated description about how certain parameters affect routing behaviour in Work App for sendDestinationOrderExtern, insertDestinationOrderExtern and updateDestinationOrderExtern. Added new driving license types to showDriverReportExtern, insertDriverExtern and updateDriverExtern. Added hint that `separator` parameter has no effect for action createReport. Added new result parameters for showWorkingTimes. Added new timeline type filter parameter for showWorkingTimes.	HR
1.75.0	04.06.2026	Added documentation for result parameter `source` for showWorkingTimes. Documented omitted result parameters for showOrderMessageReportExtern. Added error codes `1180` and `9205`.	HR

Revision	Date	Description	Author
1.76.0	08.10.2026	New speed limit settings available for showSettings New EV driving KPI available for showOptiDriveIndicator and showTripReportExtern getOptiDriveProfiles returns the configured weight for EV driving behaviour. Added error code `9539`.	