rFMS version 2.1 - API documentation

Table of Contents

Document History

Date	Changes/Additions
21.09.2016	Date of publish
08.11.2016	Correction of VehicleSpeedClass: Delete of the speed class [160; [Correction in XSD commend “accumulated_types_2.0.0.xsd”
21.08.2017	General remarks • added: Responses in XML are mandatory whereas JSON is optional API versioning • This version is related to the version of the specification. Vehicle - Base resource info: • deleted HTTP 415 : unsupported media type Vehicle position - Base resource info: • deleted HTTP 415 : unsupported media type Vehicle position - Request parameters: • added: If stoptime is used the response will have data with starttime inclusive and stoptime exclusive. Vehicle position - Response: • added in speed description: (e.g. GPS) Vehicle status - Base resource info: • deleted: HTTP 415 : unsupported media type Vehicle status - Parameter limitations: • added: If stoptime is used the response will have data with starttime inclusive and stoptime exclusive. Vehicle status - Accumulated data: • updated retarderTorqueClass: In percent (how the retarder is used as a positive value). Minimum 5 classes [0, 20[[20, 40[[40, 60[[60, 80[[80, 100] Vehicle status - Classes definition explanation: • correction of example: 2000 seconds between -0.099999… and 0.099999… Vehicle status - Snapshot data: • added in speed description: (e.g. GPS) • added: engineSpeed • deleted unit for TachoDriverIdentification: String • added: TachoDriverIdentificationType Error codes: • deleted HTTP 415 : unsupported media type

21.09.2017	Vehicle - Response • added: The vehicles are always returned in the same order. The ordering is OEM specific. Vehicle position – Request parameters • added: starttime: (i.e. >= starttime) • added: stoptime: (i.e. < stoptime) • changed: latestOnly description If this value is set to true only the ~~last~~ latest position, matching the trigger filter (if used), is returned for each vehicle in the response. • added: lastVin String **:**The response will return the next block of vehicles not including the one with the given VIN. When the last call to the vehicle position function had MoreDataAvailable-parameter set to true and the latestOnly=true in the request you must supply the VIN of the last item in the list as lastVin parameter in the next call. • added: If the times are sent according ISO 8601 e.g. with milliseconds they will be accepted. However milliseconds will be ignored • added: If the ***moreDataAvailable*** is set to true in the response and the ***latestOnly*** was set to true in the request then the next request shall state the last received VIN to get the full set. The vehicles are always returned in the same order, when using ***latestOnly*****.** The ordering is OEM specific. Otherwise use the last ***ReceivedDateTime*** **+ 1 second** of the last vehicle from the last response using it as the ***starttime*** parameter in the next request. Data received during the latest, not fully elapsed, second at the server cannot be sent to the client, i.e. if the current time at the server is 10:01:02.123, only data registered up to 10:01:01.999 can be returned. This to avoid duplicated and/or missing data Vehicle status – Request parameters • added: starttime: (i.e. >= starttime) • added: stoptime: (i.e. < stoptime) • changed: latestOnly description If this value is set to true only the ~~last~~ latest vehicle status event, matching the trigger filter (if used), is returned for each vehicle in the response. • added: If the times are sent according ISO 8601 e.g. with milliseconds they will be accepted. However milliseconds will be ignored • added: If the ***moreDataAvailable*** is set to true in the response and the ***latestOnly*** was set to true in the request then the next request shall state the last received VIN to get the full set. The vehicles are always returned in the same order, when using ***latestOnly*****.** The ordering is OEM specific. Otherwise use the last ***ReceivedDateTime*** **+ 1 second** of the last vehicle from the last response using it as the ***starttime*** parameter in the next request. Data received during the latest, not fully elapsed, second at the server cannot be sent to the client, i.e. if the current time at the server is 10:01:02.123, only data registered up to 10:01:01.999 can be returned. This to avoid duplicated and/or missing data • added: Examples for using the content filter Vehicle status - Accumulated data: • updated retarderTorqueClass: In percent (how the retarder is used as a positive value). Minimum 5 classes **]**0, 20[[20, 40[[40, 60[[60, 80[[80, 100]

	Vehicle status – content filter: • deleted:**:** If the content filter is used, only the vehicle status reports for which all the requested blocks are available will be included in the response. Vehicle status: • added: Examples for using the content filter and trigger filter • deleted: Example for starttime&latesOnly • added: Example for TellTale trigger&latestOnly

Introduction

The rFMS API is used to remotely access vehicle FMS data in a standardized way without installing any additional hardware to the vehicle by using the existing OEM hardware.

This document describes the Off-board API for the OEM back office.

Abbreviations

Abbreviation	Description
CAN	Controller Area Network
FMS	Fleet Management System
OBD	On-Board Diagnostics
OEM	Original Equipment Manufacturer
rFMS	Remote Fleet Management Standard
TF	Task Force
HW	Hardware
SW	Software
API	Application Programming Interface
REST	Representational State Transfer
OS	Operating System
HTTPS	Hypertext Transfer Protocol Secure
SSL	Secure Sockets Layer
TLS	Transport Layer Security
IP	Internet Protocol
XSD	XML Schema Definition
XML	Extensible Markup Language
JSON	JavaScript Object Notation
VIN	Vehicle Identification Number

Definitions

Term	Description
Mandatory	The API shall provide the mandatory parameters, provided that the vehicle is specified with rFMS version 2.
Optional	It is up to each OEM to decide if they want to implement this information or not in the API.

References

Document name
rFMS XML Schemas
SAE J1939 SPN 5837 **– (Fuel type definitions)**
fms_document_ver03_vers_14_09_2012.pdf

Contract and subscription

The contract and subscription is OEM specific.

General remarks

The following is valid for all services.

Parameters not supported by the vehicle will not be part of the response.

Parameters currently unavailable or invalid will not be part of the response.

Updates of data might be not possible during special conditions, e.g. ignition off, communication not possible, etc.

Responses in XML are mandatory whereas JSON is optional.

The timestamps in the responses are in UTC and always formatted as a string according to ISO8601, regardless of if it is an XML or JSON response.

Communication technology

The rFMS API is a HTTPS based REST API.

HTTP headers

There are a few mandatory HTTP headers for the rFMS API.

- Authorization
- Accept

Security

A Basic authentication security solution is used, where the username & password is set in the Authorization HTTP header encoded in Base64.

HTTP header field: **Authorization**

Format: **Basic [Base64 encoded(username:password)]**

If the Authorization header isn't supplied or empty a 401 error response will be returned.

Example:

Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==

API versioning

The resource URL's has no versions, but the client defines in the media type HTTP header which response version that is wanted. This version is related to the version of the specification.

This is done in the **Accept** HTTP header.

If the Accept header isn't set a 406 error response will be returned.

Example:

Accept: application/vnd.fmsstandard.com.VehicleStatuses.v2.1+json; UTF-8

In this example the clients expects to get a VehicleStatuses of version 2 in JSON format back from the vehicle statuses resource.

Rate Limitations

This is used to inform the client on the current rate limitations for the service.
Rate limits are divided into intervals, for each interval there is a pool of available requests.
The rate limitation is an optional feature and might not be available at all OEMs.

HTTP headers in response

Name	M/O	Description
X-Rate-Limit-Limit	M	The limit of requests within the given time frame
X-Rate-Limit-Remaining	M	The remaining number of requests until reset
X-Rate-Limit-Reset	M	The time in UTC until rate limit reset in second since 01.01.1970

When the limit is exceeded the API will return HTTP 429 “Too Many Requests”

Remark: Assume each request will be counted even if it is a request based on the “more data available”

Example:

This is returned at 2016-04-06T20:00:00 (1459972800)

Resource	X-Rate-Limit-Remaining	X-Rate-Limit-Reset	X-Rate-Limit-Limit
/vehiclepositions	55	1459973400	60
/vehicles	2	1459984500	4
/vehiclestatuses	28	1459973460	30

In this example the client can

- request vehiclepositions 55 times for the next 10 minutes.
- request vehicles 2 times for the next 3 hours and 15 minutes.
- request vehiclestatuses 28 times for the next 11 minutes.

Vehicle

The vehicle resource is used to get a list of all the vehicles the client credentials has access to.

The vehicle list is always returned in the same order.

Base resource info

Description	Vehicle resource
Sub URL	/rfms/vehicles/
Response Content-Type	Accept: application/vnd.fmsstandard.com.Vehicles.v2.1+json; UTF-8 Accept: application/vnd.fmsstandard.com.Vehicles.v2.1+xml; UTF-8
HTTP method	GET
Response codes (detailed info can be found here)	HTTP 200 : OK HTTP 400 : Bad Request HTTP 401 : Unauthorized HTTP 404 : Not Found HTTP 406 : Not acceptable HTTP 429 : Too many requests

Request parameters

Name	Type	Description
requestId	String	A client unique request id used for fault tracing at the API supplier. This shall be unique for each request if used.
lastVin	String	The response will return the next block of vehicles not including the one with the given VIN. When the last call to the vehicle list function had MoreDataAvailable-parameter set to true you must supply the VIN of the last item in the list as lastVin parameter in the next call.

Parameter limitations

It is optional to use the requestId.

If it is used then it should be ASCII min. 1 character and max. 40 characters.

Response

Vehicles

This is the response that will be sent back for a request to the Vehicles resource.
It will contain a list of vehicles matching the filter parameters supplied in the request.

If there are no vehicles in the account an empty list will be returned

Name	Type	M/O	Example	Description
vehicle	List of VehicleType objects	M		This is the list of the vehicles matching the supplied filter criteria
moreDataAvailable	Boolean	M		This will be set to true if the result set was too large to be sent back in one reply. A new request must be sent to get the rest of the vehicles, where the lastVin parameter must be supplied. The lastVin should be set to the VIN of the last vehicle in the result set of this message.

The vehicles are always returned in the same order.
The ordering is OEM specific.

VehicleType

The response object VehicleType

Name	Example	M/O	Description
vin		M	vehicle identification number See ISO 3779 (17 characters)
customerVehicleName		O	The customer's name for the vehicle.
brand	VOLVO TRUCKS, SCANIA, DAIMLER, IVECO, DAF, MAN, RENAULT TRUCKS, VDL, VOLVO BUS	M	The vehicle brand.
type	TRUCK, BUS	M	The type of vehicle
model		O	The model of the vehicle.
productionDate		O	Indicates when the vehicle was produced.
possibleFuelType	1A	O	The possible fuel types supported by this vehicle, formatted as the hex id number according to SPN 5837.
emissionLevel	EURO_V, EURO_VI	O	The emission level this vehicle supports
tellTaleCode		O	This parameter indicates how the tell tales shall be interpreted, the code is unique for each OEM. One OEM can have different interpretations depending on vehicle type.
chassisType		O	The chassis type of the vehicle. This is mainly used for busses.
noOfAxles		O	Number of axles on the vehicle This is mainly used for busses.
totalFuelTankVolume		O	Total fuel volume for all tanks in millilitres- tank volume only This is mainly used for busses.
tachographType	MTCO, DTCO, TSU, NONE	O	The type of tachograph in the vehicle This is mainly used for busses.
gearboxType	MANUAL, AUTOMATIC, SEMI_AUTOMATIC, NO_GEAR (e.g electrical)	O	The type of gearbox the vehicle is equipped with This is mainly used for busses.
bodyType	CITY_BUS, INTERCITY_BUS, COACH	O	The type of body on the chassis This is mainly used for busses.
doorConfiguration	1,2,2 means 1 front door, double doors for door 2 and 3	O	The door configuration. This is mainly used for busses.
hasRampOrLift		O	If the vehicle is equipped with a ramp or not. This is mainly used for busses.

Limitations

Limitations on how often the resource can be called is OEM specific.

The max number of items returned in one call is OEM specific.

Vehicle position

The vehicle position resource is used to get the positions for one or several vehicles.

The ***starttime***, ***stoptime*** & ***latestOnly*** parameters can be used to get all historical positions between a start and stop time or the latest known position only.

The ***vin*** parameter can be used to get all historical positions between starttime and stoptime or latest position for one individual vehicle.
If the ***vin*** parameter isn't set the response will contain all vehicles the client has access to.

If a request is made for data in a period where no data has been received, an empty list will be returned.

Base resource info

Description	Vehicle position resource
Sub URL	/rfms/vehiclepositions/
Response Content-Type	Accept: application/vnd.fmsstandard.com.VehiclePositions.v2.1+json; UTF-8 Accept: application/vnd.fmsstandard.com.VehiclePositions.v2.1+xml; UTF-8
HTTP method	GET
Response codes (detailed info can be found here)	HTTP 200 : OK HTTP 400 : Bad request HTTP 401 : Unauthorized HTTP 403 : Forbidden HTTP 404 : Not Found HTTP 406 : Not acceptable HTTP 429 : Too many requests

Request parameters

Name	Type	Default	Description
requestId	String		A client unique request id used for fault tracing at the API supplier. This shall be unique for each request if used.
datetype	Enum [created/ **received**]	received	The start/stop times are compared to the created or received time of the position reports. If this isn't supplied all times are received times.
starttime	String (ISO8601)		Only the data created/received after this time should be returned. (i.e. >= starttime)
stoptime	String (ISO8601)	Now	Only the data created/received before this time should be returned. (i.e. < stoptime)
vin	String (ISO3779)		Only the data created by the vehicle with this VIN should be returned
latestOnly	Boolean (true/false)	false	If this value is set to **true** only the ~~last~~ latest position, matching the trigger filter (if used), is returned for each vehicle in the response. If this value is set to **false (or not present)** all positions matching the starttime/stoptime criteria is returned for each vehicle in the response.
triggerFilter	String [A trigger name]		Will only return data reports that was triggered by the trigger defined by the filter. Detailed information for this parameter can be found in the Trigger filter chapter
lastVin	String		The response will return the next block of vehicles not including the one with the given VIN. When the last call to the vehicle status function had MoreDataAvailable-parameter set to true and the latestOnly=true in the request you must supply the VIN of the last item in the list as lastVin parameter in the next call.

Parameter limitations

It is optional to use the requestId.
If it is used then it should be ASCII min. 1 character and max. 40 characters.

It is mandatory to either supply the ***starttime*** (for historical requests) or ***latestOnly*** (for getting the latest data).

If none of these parameters are set, a HTTP 400 error will be returned indicating that the parameters supplied are invalid.

If ***latestOnly*** and ***starttime*** and/or ***stoptime*** are set, a HTTP 400 error will be returned indicating that the parameters supplied are invalid.

If stoptime is used the response will have data with starttime inclusive and stoptime exclusive.

If the times are sent according ISO 8601 e.g. with milliseconds they will be accepted. However milliseconds will be ignored

If the ***moreDataAvailable*** is set to true in the response and the ***latestOnly*** was set to true in the request then the next request shall state the last received VIN to get the full set.
The vehicles are always returned in the same order, when using latestOnly. The ordering is OEM specific.

Otherwise use the last ***ReceivedDateTime*** + 1 second of the last vehicle from the last response using it as the ***starttime*** parameter in the next request.

Data received during the latest, not fully elapsed, second at the server cannot be sent to the client, i.e. if the current time at the server is 10:01:02.123, only data registered up to 10:01:01.999 can be returned. This to avoid duplicated and/or missing data

Response

The response object VehiclePositions contains the following.

Name	Example	M/O	Description
triggerType		M	Indication of the type of trigger that triggered this event.
requestServerDateTime		M	Time to be used to ask for historical data at customers (for starttime), to solve the problem of having different times at server and clients. This is the time at the server when this request was received. To avoid losing any messages or get duplicates, this is the time that should be supplied in the startTime parameter in the next request.
receivedDateTime		M	Reception at Server To be used for handling of "more data available"
latitude		M	Latitude (WGS84 based)
longitude		M	Longitude (WGS84 based)
heading		O	The direction of the vehicle (0-359)
altitude		O	The altitude of the vehicle
speed		O	The GNSS(e.g. GPS)-speed in km/h
positionDateTime		M	The time of the position data
createdDateTime		M	When the data was retrieved in the vehicle
vin		M	vehicle identification number See ISO 3779 (17 characters)
wheelBasedSpeed		M	Wheel-Based Vehicle Speed in km/h (Speed of the vehicle as calculated from wheel or tailshaft speed)
tachographSpeed		O	Tachograph vehicle speed in km/h (Speed of the vehicle registered by the tachograph)
moreDataAvailable		M	This will be set to true if the result set was too large to be sent back in one reply. A new request must be sent to get the rest of the vehicle positions, where the starttime parameter must be supplied. The starttime should be set to the ReceivedDateTime of the last vehicle in the result set of this message.

Limitations

The refresh rate of data of the vehicle position for each vehicle is at least once every 15 minutes.
Storage period is minimum 2 weeks from when the position event was received from the vehicle.
The last received position is always available for the current requests (not available for historical requests if it is outside the storage period).
Limitations on how often the resource can be called is OEM specific. The max number of items returned in one call is OEM specific.

Vehicle status

The vehicle status resource is used to get the status reports for one or several vehicles.

Using the ***starttime***, ***stoptime*** & ***latestOnly*** parameters it can be used to get all historical status reports between a start and stop time or the latest known status only.

Using the ***vin*** parameter it can be used to get all historical vehicle reports or latest status for one individual vehicle.

If a request is made for data in a period where no data has been received, an empty list will be returned.

Base resource info

Description	Vehicle status resource
Sub URL	/rfms/vehiclestatuses/
Response Content-Type	Accept: application/vnd.fmsstandard.com.VehicleStatuses.v2.1+json; UTF-8 Accept: application/vnd.fmsstandard.com.VehicleStatuses.v2.1+xml; UTF-8
HTTP method	GET
Response codes (detailed info can be found here)	HTTP 200 : OK HTTP 400 : Bad Request HTTP 401 : Unauthorized HTTP 403 : Forbidden HTTP 404 : Not Found HTTP 406 : Not acceptable HTTP 429 : Too many requests

Request parameters

Name	Type	Default	Description
requestId	String		A client unique request id used for fault tracing at the API supplier. This shall be unique for each request if used.
datetype	Enum [created/**received**]	received	The start/stop times are compared to the created or received time of the position reports. If this isn't supplied all times are received times.
starttime	String (ISO8601)		Only the data created/received after this time should be returned. (i.e. >= starttime)
stoptime	String (ISO8601)	Now	Only the data created/received before this time should be returned. (i.e. < stoptime)
vin	String (ISO3779)		Only the data created by the vehicle with this VIN time should be returned
contentFilter	Enum [SNAPSHOT, ACCUMULATED, UPTIME]		Will only return data reports that contain information in the area defined by the filters. Detailed information for this parameter can be found in the Content chapter.
triggerFilter	String [A trigger name]		Will only return data reports that were triggered by the trigger defined by the filter. Detailed information for this parameter can be found in the Trigger filter chapter.
latestOnly	Boolean	false	If this value is set to **true** only the ~~last~~ latest vehicle status event, matching the trigger filter (if used), is returned for each vehicle in the response. If this value is set to **false (or not present)** all vehicle status events matching the starttime/stoptime criteria is returned for each vehicle in the response.
lastVin	String		The response will return the next block of vehicles not including the one with the given VIN. When the last call to the vehicle position function had MoreDataAvailable-parameter set to true and the latestOnly=true in the request you must supply the VIN of the last item in the list as lastVin parameter in the next call.

Parameter limitations

It is optional to use the requestId.
If it is used then it should be ASCII min. 1 character and max. 40 characters.

It is mandatory to either supply the ***starttime*** (for historical requests) or ***latestOnly*** (for getting the latest data).
If neither of these parameters is set, a HTTP 400 error will be returned indicating that the parameters supplied are invalid.

If ***latestOnly*** and ***starttime*** and/or ***stoptime*** are set, a HTTP 400 error will be returned indicating that the parameters supplied are invalid.
If ***stoptime*** is used the response will have data with ***starttime*** inclusive and ***stoptime*** exclusive.
If the times are sent according ISO 8601 e.g. with milliseconds they will be accepted. However milliseconds will be ignored

If the ***moreDataAvailable*** is set to true in the response and the ***latestOnly*** was set to true in the request then the next request shall state the last received VIN to get the full set.
The vehicles are always returned in the same order, when using latestOnly. The ordering is OEM specific.

Otherwise use the last ***ReceivedDateTime*** + 1 second of the last vehicle from the last response using it as the ***starttime*** parameter in the next request.

Data received during the latest, not fully elapsed, second at the server cannot be sent to the client, i.e. if the current time at the server is 10:01:02.123, only data registered up to 10:01:01.999 can be returned. This to avoid duplicated and/or missing data

Content filter

The content filter can be used to limit the data in the response to the requested blocks.

If several values are used in the filter they should be in a comma-separated list (E.g.: contentFilter=ACCUMULATED,SNAPSHOT).

If this filter parameter isn't supplied the returned reports contain all available blocks.

Possible values

- ACCUMULATED
- SNAPSHOT
- UPTIME

If the content filter is set and the latestOnly is set to true then the response will be the last received data containing the data set in the content filter

See Examples for using the content filter and trigger filter

Trigger filter

The trigger filter can be used to limit the response to contain only events that are triggered by the specified triggers (E.g. events triggered by a driver login).

If several values are used in the filter they should be in a comma-separated list (Ex: triggerFilter=DRIVER_LOGIN,TIMER).

If this filter parameter isn't supplied the returned data reports contain data reports triggered by any trigger.

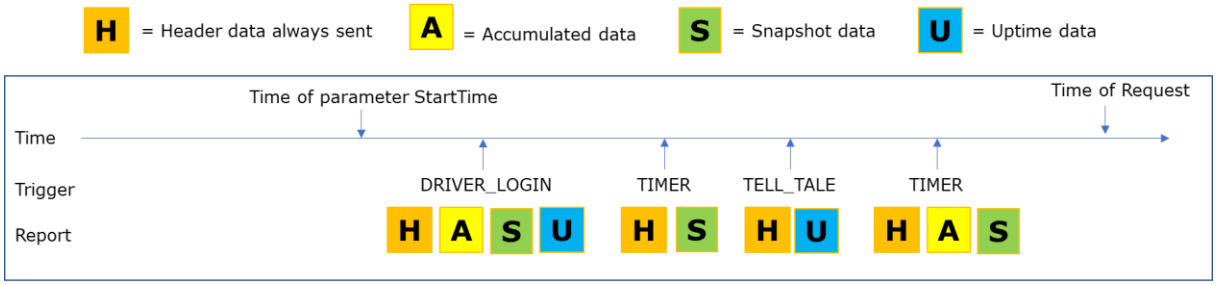
Possible values

The following list of triggers is the ones in the rFMS standard.

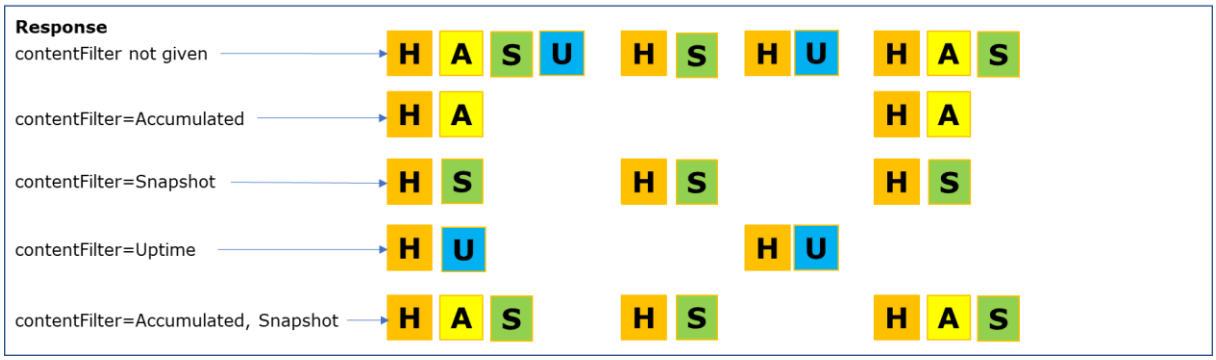
Complementing this list the different OEMs can also have their own triggers, the list of these can be retrieved by each OEM.

- TIMER
- TELL_TALE
- DRIVER_LOGIN
- DRIVER_LOGOUT
- IGNITION_ON
- IGNITION_OFF
- ENGINE_ON
- ENGINE_OFF
- PTO_ENABLED
- PTO_DISABLED
- DISTANCE_TRAVELLED
- DRIVER_1_WORKING_STATE_CHANGED
- DRIVER_2_WORKING_STATE_CHANGED

Examples for using the content filter and trigger filter

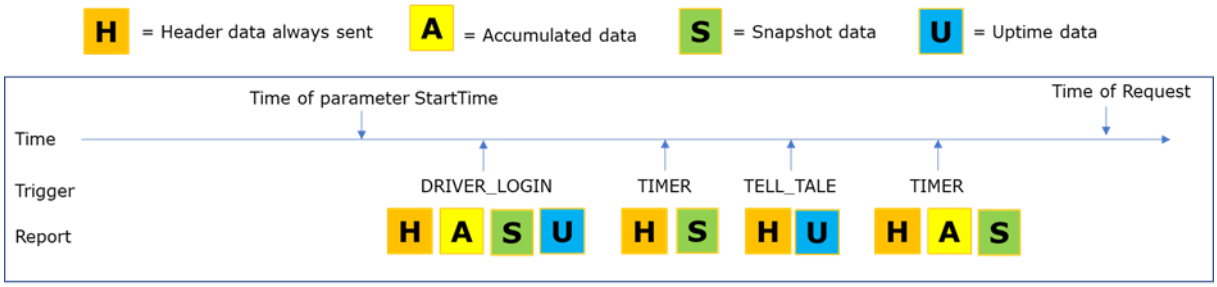


Request using parameter starttime



Request using parameter latestOnly

Request using parameter starttime & Trigger=TELL_TALE



Request using parameter Trigger=TELL_TALE & latestOnly

Response

contentFilter not given → H U

contentFilter=Accumulated →

contentFilter=Snapshot →

contentFilter=Uptime → H U

contentFilter=Accumulated, Snapshot →

Response

VehicleStatuses

This is the response that will be sent back for a request to the vehicle statuses resource.

It will contain a list of vehicle statuses matching the filter parameters supplied in the request.

Name	Example	M/O	Description
vehicleStatus		M	This is the list of the vehicle status types matching the supplied filter criteria
requestServerDateTime		M	Time in UTC to be used to ask for historical data (for starttime), to solve the problem of having different times at server and clients. This is the time at the server when this request was received. To avoid losing any messages or get duplicates, this is the time that should be supplied in the startTime parameter in the next request.
moreDataAvailable		M	This will be set to true if the result set was too large to be sent back in one reply. A new request must be done to get the rest of the vehicle statuses, where the starttime parameter must be supplied. The starttime should be set to the ReceivedDateTime of the last vehicle in the result set of this message.

VehicleStatusType

The response object VehicleStatusType

Name	Example	M/O	Description
vin		M	vehicle identification number See ISO 3779 (17 characters)
triggerType		M	Indication of the type of trigger that triggered this event.
createdDateTime		M	When the data was generated in the vehicle.
receivedDateTime		M	When the data was received at the OEM backend systems.
hrTotalVehicleDistance		M	Accumulated distance travelled by the vehicle during its operation in meter
totalEngineHours		M	The total hours of operation for the vehicle engine.
engineTotalFuelUsed		M	The total fuel the vehicle has used during its lifetime in MilliLitres.
grossCombinationVehicleWeight	kg	O	The full vehicle weight in kg
driver1Id		M	The driver identity
accumulatedData		O	A list of accumulated vehicle data
snapshotData		O	A list of snapshot vehicle data
uptimeData		O	A list of uptime vehicle data
status2OfDoors		M (bus)	Bus specific parameter
doorStatus		O	Bus specific parameter

Accumulated data

The content of the accumulated data block.

Name	Unit	M/O	Description
durationWheelbaseSpeedOverZero	Seconds	M	The time the vehicle speed has been over zero.
distanceCruiseControlActive	Meter	M	The distance the vehicle has been driven with cruise control active
durationCruiseControlActive	Seconds	M	The time the vehicle has been driven with cruise control active
fuelConsumptionDuringCruiseActive	MilliLitres	M	The fuel consumption the vehicle has consumed while driven with cruise control active
durationWheelbaseSpeedZero	Seconds	M	The time the vehicle speed has been equal to zero. Engine on (RPM>0) and no PTO active
fuelDuringWheelbaseSpeedZero	MilliLitres	M	The fuel consumption the vehicle has consumed while the vehicle speed has been equal to zero. Engine on (RPM>0) and no PTO active
fuelWheelbaseSpeedOverZero	MilliLitres	M	The fuel consumption the vehicle has consumed while the vehicle speed has been over zero. Engine on (RPM>0)
ptoActiveClass		M	2 Classes: wheelbased speed >0, wheelbased speed =0 at least one PTO on during wheelbased speed =0 counter in seconds and millilitres at least one PTO on during wheelbased speed>0 counter in seconds, millilitres (includes consumption of driving) and meter driven
brakePedalCounterSpeedOverZero		M	The total number of times the brake pedal has been used while the vehicle was driving.
distanceBrakePedalActiveSpeedOverZero	Meter	M	The total distance the vehicle has driven where the brake pedal has been used.
accelerationPedalPositionClass		M	In percent. Minimum 5 classes [0, 20[[20, 40[[40, 60[[60, 80[[80, 100]

Name	Unit	M/O	Description
accelerationClass		M	In m/s^2 Minimum 13 classes.], -1.1]]-1.1, -0.9]]-0.9, -0.7]]-0.7, -0.5]]-0.5, -0.3]]-0.3, -0.1]]-0.1, 0.1[[0.1, 0.3[[0.3, 0.5[[0.5, 0.7[[0.7, 0.9[[0.9, 1.1[[1.1, [
retarderTorqueClass		M	In percent (how the retarder is used as a positive value). Minimum 5 classes]0, 20[[20, 40[[40, 60[[60, 80[[80, 100]
HighAccelerationInClasses		O	In m/s2 Minimum 11 classes], -3.0]]-3.0, -2.5]]-2.5, -2.0]]-2.0, -1.5]]-1.5, -1.1]]-1.1, 1.1[[1.1, 1.5[[1.5, 2.0[[2.0, 2.5[[2.5, 3.0[[3.0, [
drivingWithoutTorqueClass		M	The total number of seconds/meters while driving without torque. With gear (clutch is engaged)
engineTorqueClass		M	In percent based on EEC1 value (Actual Engine-Percent Torque). Minimum 10 classes [0, 10[[10, 20[[20, 30[[30, 40[[40, 50[[50, 60[[60, 70[[70, 80[[80, 90[[90, 100]
engineTorqueAtCurrentSpeedClass		O	In percent based on EEC2 value (Engine Percent Load At Current Speed). Minimum 10 classes [0, 10[[10, 20[[20, 30[[30, 40[[40, 50[[50, 60[[60, 70[[70, 80[[80, 90[[90, 100]
vehicleSpeedClass		M	In km/h Minimum 40 classes. [0, 4[[4, 8[[8, 12[[12, 16[[16, 20[[20, 24[... [156, [Engine on (RPM>0)
engineSpeedClass		M	In RPM Minimum 10 classes [0, 400[[400, 800[[800, 1200[[1200, 1600[[1600, 2000[[2000, 2400[[2400, 2800[[2800, 3200[[3200, 3600[[3600, [
accelerationDuringBrakeClass		O	In m/s^2 Minimum 13 classes.], -1.1]]-1.1, -0.9]]-0.9, -0.7]]-0.7, -0.5]]-0.5, -0.3]]-0.3, -0.1]]-0.1, 0.1[[0.1, 0.3[[0.3, 0.5[[0.5, 0.7[[0.7, 0.9[[0.9, 1.1[[1.1, [

selectedGearClass		O	The currently selected gear One class per gear. Neutral is also a gear. Park is also a gear. This is formatted according to SPN 524, supplied as a decimal value. Example: 0 = Neutral, 1 = 1:st gear... This is mainly used for Busses
currentGearClass		O	The currently used gear One class per gear. Neutral is also a gear. Park is also a gear. This is formatted according to SPN 523, supplied as a decimal value. Example: 0 = Neutral, 1 = 1:st gear... This is mainly used for Busses
chairliftCounter		O	The total number of times the chairlift has been outside the bus. This is mainly used for Busses
stopRequestCounter		O	The total number of stop requests made. This is mainly used for Busses
kneelingCounter		O	The total number of times the bus has knelt.
pramRequestCounter		O	The total number of pram requests made. This is mainly used for Busses

The class parameter values are incremented continuously. Reset is depending on OEM and can be done e.g. in case of changing the owner of the vehicle.
The starting point of the class parameters are OEM specific.

The described classes are minimum classes. If there are more classes delivered they can be aggregated to the minimum classes

Example: value with 3 classes from 0-20, 20-40, 40-60
can be delivered in 6 classes from 0-10, 10-20, 20-30, 30-40, 40-50, 50-60
or in 5 classes from 0-20, 20-25, 25-30, 35-40, 40-60 as it can be aggregated to the minimum classes
and will be not delivered in classes from 5-15,15-30,30-45,45-60 as it cannot be aggregated to the minimum 3 classes

Classes definition explanation

The minimum classes are represented using a bracket notation indicating if a value is part of the block or not.
If the following would be a class definition with minimum 7 classes defined as follows

], -2]]-2, -1]]-1, -0.1]]-0.1, 0.1[[0.1, 1[[1, 2[[2, [

It would mean the following.

], -2]: This class represents the number of seconds in interval -2 to any value under -2
]-2, -1]: This class represents the number of seconds in interval -1 to but not including -2
]-1, -0,1]: This class represents the number of seconds in interval -0.1 to but not including -1
]-0.1, 0.1[: This class represents the number of seconds in interval from but not including 0.1 to but not including -0.1
[0.1, 1[:This class represents the number of seconds in interval 0.1 to but not including 1
[1, 2[:This class represents the number of seconds in interval 1 to but not including 2
[2,[:This class represents the number of seconds in interval 2 to any value over 2

If we exemplify this with receiving this list of seconds (7 values):

40, 10, 20, 2000, 30, 31, 60

The vehicle would have spent :

40 seconds between -2 and anything below that
10 seconds between -1 and -1.99999…
20 seconds between -0.1 and -0.99999…
2000 seconds between -0.099999… and 0.099999…
30 seconds between 0.1 and 0.99999…
31 seconds between 1 and 1.99999…
60 seconds between 2 and anything above this

Snapshot data

The content of the snapshot data block.

Name	Unit	M/O	Description
latitude	WGS84	M	Latitude
longitude	WGS84	M	Longitude
heading	Degrees	O	The direction of the vehicle (0-359)
altitude	meter	O	The altitude of the vehicle
speed	km/h	O	The GNSS (e.g. GPS)-speed in km/h
positionDateTime	Time	M	The time of the position data
wheelBasedSpeed	Km/h	M	The vehicle wheelbased speed
tachographSpeed	Km/h	O	The Tacho speed
engineSpeed	rpm	O	The engine speed in rev/min
fuelLevel1	%	M	The fuel level percentage
catalystFuelLevel	%	O	The adblue level percentage
driver1WorkingState		O	The working state of driver 1
driver2Id		O	The Id of driver 2
driver2WorkingState		O	The working state of driver 2
ambientAirTemperature	Degrees Celsius	O	The Ambient air temperature in Celsius

Driver Id

The id of the driver.

The Id can either be an EU tacho driver id or an OEM specific driver Id.

Name	Unit	M/O	Description
TachoDriverIdentification		O	The EU standard driver tachograph id.
OemDriverIdentification	String	O	An OEM specific driver id.
IdType	String	O	It can also contain an optional id type (ex: pin, USB, encrypted EU id…)

TachoDriverIdentificationType

Name	Description
TachoDriverIdentificationType	The EU standard driver tachograph id. The fields in this struct are formatted according to: COMMISSION REGULATION (EC) No 1360/2002 Annex 1b http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2002:207:0001:0252:EN:PDF
DriverIdentification	The unique identification of a driver in a Member State. This fields is formatted according the definition for driverIdentification in: COMMISSION REGULATION (EC) No 1360/2002 Annex 1b
CardIssuingMemberState	The country alpha code of the Member State having issued the card. This fields is formatted according the definition for NationAlpha in: COMMISSION REGULATION (EC) No 1360/2002 Annex 1b
DriverAuthenticationEquipment	Code to distinguish different types of equipment for the tachograph application. See description of the field 'DriverAuthenticationEquipment' in: COMMISSION REGULATION (EC) No 1360/2002 Annex 1b
CardReplacementIndex	A card replacement index. This fields is formatted according the definition for CardReplacementIndex (chap 2.26) in: COMMISSION REGULATION (EC) No 1360/2002 Annex 1b
CardRenewalIndex	A card renewal index. This fields is formatted according the definition for CardRenewalIndex (chap 2.25) in: COMMISSION REGULATION (EC) No 1360/2002 Annex 1b

Uptime data

The content of the uptime data block.

Name	Unit	M/O	Description
tellTaleInfo		M	List of tell tales with the actual status for each tell tale.
serviceDistance	Meter	O	The distance in meter to the next service
engineCoolantTemperature	Celsius	O	The temperature of the coolant liquid.
serviceBrakeAirPressureCircuit1	Pascal	O	The air pressure in circuit 1 in Pascal.
serviceBrakeAirPressureCircuit2	Pascal	O	The air pressure in circuit 2 in Pascal.
durationAtLeastOneDoorOpen	Seconds	O	The total time at least one door has been opened in the bus. Used mainly for busses.
alternatorStatus		M (bus)	The alternator status of the up to 4 alternators. Used mainly for busses.
bellowPressureFrontAxleLeft	Pascal	O	The bellow pressure in the front axle left side in Pascal. Used mainly for busses.
bellowPressureFrontAxleRight	Pascal	O	The bellow pressure in the front axle right side in Pascal. Used mainly for busses.
bellowPressureRearAxleLeft	Pascal	O	The bellow pressure in the rear axle left side in Pascal. Used mainly for busses.
bellowPressureRearAxleRight	Pascal	O	The bellow pressure in the rear axle right side in Pascal. Used mainly for busses.

Limitations

The refresh rate of data of the vehicle status reports for each vehicle is at least once every 60 minutes.

Storage period is minimum 2 weeks from when the vehicle status event was received from the vehicle.
The last received vehicle status event is always available for the current requests (not available for historical requests if it is outside the storage period)

Limitations on how often the resource can be called is OEM specific.

The max number of items returned in one call is OEM specific.

Response Codes

Success codes

This class of status codes indicates the action requested by the client was received, understood, accepted and processed successfully

Code	Meaning	Comment
200	Ok	The actual response will depend on the request method used. In a GET request, the response will contain an entity corresponding to the requested resource.

Errors codes

The 4xx class of status code is intended for cases in which the client seems to have erred. Except when responding to a HEAD request, the server should include an entity containing an explanation of the error situation, and whether it is a temporary or permanent condition. These status codes are applicable to any request method. User agents should display any included entity to the user.

Code	Meaning	Possible reason	Comment
400	Bad Request	mandatory field missing, e.g. Authentication Header empty or missing	The server cannot or will not process the request due to an apparent client error (e.g., malformed request syntax, invalid request message framing, or deceptive request routing)
401	Unauthorized	Wrong credentials	Similar to 403 Forbidden, but specifically for use when authentication is required and has failed or has not yet been provided. The response must include a WWW-Authenticate header field containing a challenge applicable to the requested resource. See Basic access authentication and Digest access authentication.
		Login credentials expired	
403	Forbidden	Insufficient rights for the service	The request was a valid request, but the server is refusing to respond to it. Unlike a 401 Unauthorized response, authenticating will make no difference. On servers where authentication is required, this commonly means that the provided credentials were successfully authenticated but that the credentials still do not grant the client permission to access the resource (e.g. a recognized user attempting to access restricted content)
		no rights on any service of this vehicle	
		Response is too large	
404	Not Found	vehicle unknown	The requested resource could not be found but may be available again in the future. Subsequent requests by the client are permissible
		rFMS-Version not supported	
406	Not acceptable	unsupported Accept parameter sent	
429	Too Many Requests	Request sent too often	The user has sent too many requests in a given amount of time. Intended for use with rate limiting schemes
		Max concurrent calls	

Triggers

The triggers that can be used for triggering of positions or vehicle statuses events.

Name	Context	M/O	Description
Time	rFMS	M	Data was sent due to a timer trigger. (Timer value set outside rFMS scope)
Ignition on	rFMS	O	Data was sent due to an ignition on
Ignition off	rFMS	O	Data was sent due to an ignition off
PTO enabled	rFMS	O	Data was sent due to that a PTO was enabled, will be sent for each PTO that gets enabled
PTO disabled	rFMS	O	Data was sent due to that a PTO was disabled, will be sent for each PTO that gets disabled.
Driver login	rFMS	M	Data was sent due to a successful driver login.
Driver logout	rFMS	M	Data was sent due to a driver logout
Tell tale	rFMS	M	Data was sent due to that at least one tell tale changed state
Engine on	rFMS	O	Data was sent due to an engine on
Engine off	rFMS	O	Data was sent due to an engine off
Driver 1 working state change	rFMS	O	Data was sent due to that driver 1 changed working state
Driver 2 working state change	rFMS	O	Data was sent due to that driver 2 changed working state
Distance travelled	rFMS	O	Data was sent due to that a set distance was travelled. (Distance set outside rFMS scope)
OEM defined	OEM brand (E.g.: VOLVO TRUCKS, RENAULT TRUCKS, SCANIA, DAIMLER, MAN, IVECO, DAF, VOLVO BUS)	O	OEM specific trigger

Tell tale triggers

These are the tell tales possible in the rFMS API.

For the triggers, only the ***any -> red*** and ***red -> any*** transitions are mandatory.

Trucks

The following tell tales are mandatory as triggers for trucks:

Id	Tell tale description
8	Brake failure/brake system malfunction
13	Engine oil
18	Engine / Mil indicator
22	Anti-lock brake system failure
26	Malfunction/general failure
34	Engine Emission system failure

The following tell tales are mandatory as data elements for trucks:

Id	Tell tale description
8	Brake failure/brake system malfunction
11	Engine coolant temperature
13	Engine oil
18	Engine / Mil indicator
21	Transmission failure/malfunction
22	Anti-lock brake system failure
26	Malfunction/general failure
27	Engine oil temperature
28	Engine oil level
29	Engine coolant level
34	Engine Emission system failure
50	EBS

Busses and Coaches

The following tell tales are mandatory as triggers for busses:

Id	Tell tale description
8	Brake failure/brake system malfunction
12	Battery charging condition
13	Engine oil
18	Engine / Mil indicator
22	Anti-lock brake system failure
26	Malfunction/general failure
34	Engine Emission system failure

The following tell tales are mandatory as data elements for busses:

Id	Tell tale description
8	Brake failure/brake system malfunction
11	Engine coolant temperature
12	Battery charging condition
13	Engine oil
18	Engine / Mil indicator
21	Transmission failure/malfunction
22	Anti-lock brake system failure
26	Malfunction/general failure
27	Engine oil temperature
28	Engine oil level
29	Engine coolant level
50	EBS
34	Engine Emission system failure

XML Schemas

The API is defined in XML Schemas.
If there is a discrepancy between this specification and the XML Schema the XML Schema is the relevant definition.

XSD files

This chapter contains a list of all XML schema files and their purposes.
The XSD files can be found on fms-standard.com.

XSD file name	Purpose
common_types_2.x.y.xsd	Common definitions not related to a specific domain used in several other xsd's.
vehicle_types_2.x.y.xsd	Common definitions in the vehicle domain used in several other xsd's.
driver_types_2.x.y.xsd	Common definitions in the driver domain used in several other xsd's.
position_types_2.x.y.xsd	Common definitions in the position domain used in several other xsd's.
tell_tale_types_2.x.y.xsd	Common definitions in the tell tale domain used in several other xsd's.
trigger_types_2.x.y.xsd	Common definitions in the trigger domain used in several other xsd's.
vehicle_2.x.y.xsd	A definition of the messages used for the vehicle resource.
vehicle_position_2.x.y.xsd	A definition of the messages used for the vehicle position resource.
vehicle_status_2.x.y.xsd	A definition of the messages used for the vehicle status resource.
accumulated_types_2.x.y.xsd	A definition of the type containing all accumulated parameters for the vehicle status.
snapshot_types_2.x.y.xsd	A definition of the type containing all snapshot parameters for the vehicle status.
uptime_types_2.x.y.xsd	A definition of the type containing all uptime parameters for the vehicle status.