

TachoShare.connect

2.1.0

Reference Guide



Contents

- Getting started with TachoShare.connect..... 3**
- Welcome..... 4**
 - About this document.....4
 - About TachoShare.connect..... 4
- API authorisation..... 6**
- Terminology..... 7**
 - Webfleet Telematics Service Platform..... 7
 - WEBFLEET TachoShare..... 7
 - Vehicle unit.....7
 - Driver card..... 7
 - Remote download (RDL)..... 7
 - Company card.....7
 - Company card group..... 8
- Reference..... 9**
- Vehicles..... 10**
 - Getting vehicle information.....10
 - Getting vehicle information for a single vehicle..... 12
 - Collecting vehicle unit data.....13
 - Requesting a vehicle unit RDL.....14
- Drivers..... 16**
 - Getting driver card information..... 16
 - Collecting driver card data.....17
- Company cards.....20**
 - Getting company card information..... 20
 - Getting company card group information.....21
- Usage guidance.....23**
 - Vehicle flow.....23
 - Driver flow.....23
- Appendix..... 25**
- RDL status.....26**
- Company card status..... 27**
- Response codes.....28**
- Driver card country codes.....29**
- Resources..... 32**
- Revision history..... 33**
- Reference..... 35**

Getting started with TachoShare.connect

Welcome

TachoShare.connect is an API that allows customers and partners to access their data collected from digital tachographs and driver's cards which is stored in Webfleet Telematics Service Platform (WTSP). This document provides all the information you need to integrate driver's and vehicle data into your system or application using TachoShare.connect interface.

About this document

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

In order to access the TachoShare.connect service you need the consent of the relevant Webfleet Solutions customer with WEBFLEET TachoShare enabled. Otherwise you will not be able to test the integration for your application.

About TachoShare.connect

TachoShare.connect is an API that allows you to access the tachograph data belonging to Webfleet Solutions customers, which is stored in the [WTSP](#) archive. This tachograph data is periodically remotely downloaded when a customer has the WEBFLEET TachoShare service enabled in their vehicles. That periodicity depends on the WEBFLEET TachoShare service variant chosen by the customer.

This is an introduction to using TachoShare.connect, how to access the service and how to interpret the output that is returned from the REST endpoints.

At any time, the communication must be performed using the HTTPS protocol and the hostname of the API is tachoshareapi.webfleet.com.

Remote download periodicity

Service	Vehicle unit data	Driver card data
TACHOSHARE	28 days	7 days
TACHOSHARE PLUS	7 days	1 day

TachoShare.connect gives access to the following:

- **Vehicle unit data** - Every time TachoShare.connect downloads vehicle unit data remotely, it gets a combination of several types of data:
 - Overview
 - Vehicle activities
 - Speed details (only for the countries in which these details are mandatory)
 - Technical details
 - Events & faults

Vehicle unit data is offered as a single file containing all the blocks of data mentioned above and as many activity blocks as days requested.

- **Driver card data** - TachoShare.connect remotely downloads the data stored on a certain driver card. The driver card stores activities of the driver. Driver card data is offered in as many files as the number of remote downloads performed, where each of those files contains all the driver card data at the moment of the remote download.

Vehicle unit and driver card data must be requested with a date range specified in the request. If the date range is left empty in the request, a default date range value is used instead.

API authorisation

TachoShare.connect uses standard [OAuth2 Authorization Code Grant](#) for authorising access to the API.

Please refer to [Accessing Webfleet Solutions OAuth APIs](#) for documentation on the authorisation process.

Terminology

Webfleet Telematics Service Platform

Webfleet Telematics Service Platform (WTSP) embraces all the hardware, software and human engineering resources regarding the Webfleet Solutions engineering software stack.

WEBFLEET TachoShare

WEBFLEET TachoShare is a remote download module for digital tachographs inside the Webfleet Telematics Service Platform. It enables downloading tachograph data from remote vehicles and connecting third-party analysis software.

Vehicle unit

The vehicle unit (VU) is one of the parts of the digital tachographs. Together with card slots and the display, it forms what is known as digital tachograph. The vehicle unit can store data regarding their activities and duties for approximately 12 months. It also holds data about faults, speeding, calibration details, attempts to tamper with the system.

Driver card

The driver card is a smart card that is used with tachographs to identify the driver and to log their activities. It has an average storing capacity of 28 days.

Remote download (RDL)

Remote download (RDL) comprises remote download of the driver card data and/or vehicle unit data from the tachograph. The data is transmitted to the WTSP databases using the telematic devices, where all data are archived in compliance with the official legislation.

Additionally, the event of starting a download, manually or scheduled, is considered an RDL attempt. You can trigger a limited amount of manual RDL attempts.

The RDL status of driver cards and vehicle units indicates if the remote downloads are being completed as scheduled or their state with regards to compliance with legislation.

Company card

The company card is a smart card that is used with tachographs to identify the company. The company card contains information of the company and has an expiration date after which it cannot be used anymore.

The card allows to 'lock' data recorded in the vehicle unit (tachograph) to prevent another operator accessing the data. This locking must be done before starting to operate with the vehicle and can be 'locked out' once the vehicle will not be operated anymore by the company. It is not possible to 'lock in' data retrospectively and data generated without locking in would stay open for all to access.

The card allows to download the information from the vehicle unit of which it has been locked to. The company must send copies of their company cards that have been used to lock their vehicle units (tachographs) to Webfleet Solutions to be inserted in the slots of the company card readers within the WTSP for RDL.

Company card group

Each company card is identified by a 16 characters string. The first 13 characters identify the company card and are used to lock the vehicle units data. The last 3 characters identify the different copies of a same company card and can be used interchangeably to lock and access data from vehicle units.

The set of copies of a same company card is a company card group.

A company card group contains information about the level of occupation/the level of usage of the company card for remote download services. This is calculated by the number of vehicles locked with the company card group and the number of operational company card copies of the group available for RDL in WTSP.

Reference

Vehicles

Getting vehicle information

Description

Returns a list of all vehicles from which tachograph data can be downloaded. This list is limited to the visibility rights for user given by WEBFLEET user who has granted access during the authorisation process.

The list of vehicles returned by this action are identified by their Vehicle Identification Number (VIN) and include per vehicle information such as the date of last activity downloaded, the date of the next scheduled RDL, the RDL status and more.

Request details

This is an HTTP GET request with no extra parameters and access token needs to be used as bearer-token authorisation method.

REST endpoint	GET /vehicles
Response format	application/json
HTTP Header	Authorization: Bearer eyJhbGciOiJl...

Response details

Vehicle information is delivered in a JSON array, where every vehicle is defined as a JSON object with the following properties:

Parameter	Type	Description
vin	string	The Vehicle Identification Number.
licensePlate	string	The license plate number.
lastActivityDate	string(\$date)	The last activity downloaded from the tachograph. Date follows the yyyy-MM-dd format and is in UTC.
nextScheduledDate	string(\$date)	Scheduled date for the next download from the tachograph. Date follows the yyyy-MM-dd format and is in UTC.
lastRdlAttempt	string(\$date-time)	Date of the last attempted remote download from the tachograph. Date follows the yyyy-MM-ddTHH:mm:ssZ format and is in UTC.

Parameter	Type	Description
rdlStatus	string	<p>RDL status of the vehicle with regards to the remote download from the tachograph.</p> <p>Possible values:</p> <ul style="list-style-type: none"> • OK • DELAYED • OUTSIDE_LEGAL_COMPLIANCE • DISCONNECTED • NEVER_DOWNLOADED • EXPIRED
manualAttemptsRemaining	integer(\$int32)	Remaining manual attempts to download data remotely from the tachograph.
vehicleId	string	Unique vehicle identifier.

Below there is an example of a request and a response for GET /vehicles:

Request

```
GET /vehicles HTTP/1.1
Host: tachoshareapi.webfleet.com
Authorization: Bearer eyJhbGciOiJI...
```

Response

```
{
  "items": [
    {
      "vin": "ABCDEFGH0123456789",
      "nextScheduledDate": "2021-01-07",
      "lastRdlAttempt": "2020-01-22T14:55:15Z",
      "rdlStatus": "NEVER_DOWNLOADED",
      "manualAttemptsRemaining": 10,
      "vehicleId": "123"
    },
    {
      "vin": "HIJKLMN9876543210",
      "licensePlate": "1234ABC",
      "lastActivityDate": "2020-04-19",
      "nextScheduledDate": "2021-01-07",
      "lastRdlAttempt": "2020-04-20T15:04:10Z",
      "rdlStatus": "OUTSIDE_LEGAL_COMPLIANCE",
      "manualAttemptsRemaining": 3,
      "vehicleId": "531"
    },
    {
      "vin": "OPQRSTU0987612345",
      "licensePlate": "5678DEF",
      "lastActivityDate": "2020-07-15",
      "lastRdlAttempt": "2020-10-21T14:50:11Z",
      "rdlStatus": "DISCONNECTED",
      "manualAttemptsRemaining": 5,
      "vehicleId": "3567"
    }
  ]
}
```

As it is shown in the example above, the `lastActivityDate` and `licensePlate` parameters are not returned if no tachograph vehicle unit data is available for a certain vehicle. Similarly, the `nextScheduledDate` parameter is not returned if the link has been removed (`rdlStatus = DISCONNECTED`) or the service has expired (`rdlStatus = EXPIRED`).

Getting vehicle information for a single vehicle

This action returns information about a single vehicle identified by its Vehicle Identification Number (VIN) and includes information such as the date of last activity downloaded, the date of the next scheduled RDL, the RDL status and more.

Information is available only if the user has vehicle viewing rights given by WEBFLEET user who has granted access during the authorisation process.

Request details

This web service is an HTTP GET request with the VIN as parameter. Access token needs to be used as bearer-token authorisation method and the Accept header should also be sent as shown in the example below.

REST endpoint	GET /vehicles/{vin}
Response format	application/json
HTTP Header	Accept: application/json Authorization: Bearer eyJhbGciOiJI...
	Content-Type: application/json

Response details

The information is returned as a JSON with the properties described in [Getting vehicle information](#). If an error happens, then a JSON object is returned explaining the reason of the failure. Below there is an example of a request and a response for GET /vehicles/{vin} using HTTP Header Accept: application/json:

Request

```
<Headers>
GET /vehicles/ABCDEFGH0123456789 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJhbGciOiJI...
```

Response

```
{
  "items": [
    {
      "vin": "ABCDEFGH0123456789",
      "licensePlate": "0000ZXY",
      "nextScheduledDate": "2021-01-07",
      "lastRdlAttempt": "2020-01-22T14:55:15Z",
      "rdlStatus": "NEVER_DOWNLOADED",
      "manualAttemptsRemaining": 10,
      "vehicleId": "123"
    }
  ]
}
```

Collecting vehicle unit data

Description

This web service returns a single file containing all the vehicle unit data downloaded for a certain vehicle and according to a certain date range. The vehicle unit is identified by VIN and the maximum date range is 2 years.

Request details

This web service is an HTTP GET request with VIN as parameter, indicating the vehicle from which data shall be collected. Access token needs to be used as bearer-token authorisation method and the Accept header should also be sent as shown in the example below.

REST endpoint	GET /vehicles/{vin}/files
HTTP Header	Accept: application/octet-stream, application/json Authorization: Bearer eyJhbGciOiJl...
Response format	application/octet-stream

Query parameter	Type	Description
vin	string	Vehicle Identification Number. This parameter is required.
activityDate-from	string(\$date)	Initial activity date to be included in the response. The date should follow the yyyy-MM-dd format. This parameter is optional. If activityDate-from is empty, then it is considered to be 90 days before activityDate-to. If activityDate-to is empty, it is considered to be 90 days after activityDate-from. If both activityDate-from and activityDate-to are empty, the date range is the last 90 days.
activityDate-to	string(\$date)	Activities more recent or equal to this date are not included in the response. The date should follow the yyyy-MM-dd format. This parameter is optional. If activityDate-from is empty, then it is considered to be 90 days before activityDate-to. If activityDate-to is empty, it is considered to be 90 days after activityDate-from. If both activityDate-from and activityDate-to are empty, the date range is the last 90 days.
format	string	File format (DDD, V1B_V1C or TGD). DDD by default. This parameter is optional.

Response details

Vehicle unit data is returned in a file containing all the activities according to the date range indicated in the request. If an error happens, then a JSON object is returned explaining the reason of the failure.

Below there is an example of a request and a response for GET /vehicles/{vin}/files:

Request

```
GET
/vehicles/2GTEK13Z171168244/files?activityDate-from=2018-05-19&activity-
Date-to=2018-05-28
Host: tachoshareapi.webfleet.com
Accept: application/octet-stream, application/json
Authorization: Bearer eyJhbGciOiJI...
```

Response

```
<Headers>
Content-Disposition: attachment; file-
name="M_20180528_0715_9251QIA_2GTEK13Z171168244.DDD"
Content-Type: application/octet-stream
<Body>
octet-stream
```

Requesting a vehicle unit RDL

Description

This web service requests a new remote download for a specific vehicle identified by its VIN. The service checks if the preconditions for the RDL are met and if so, the tachograph will start a new download when the ignition of the vehicle is turned on. The service will also return the updated status of the vehicle.

The web service will deny the request in any of these conditions:

- There are no more remaining attempts, represented by `manualAttemptsRemaining`
- RDL is already in progress or waiting for execution
- RDL service is not enabled for this vehicle
- RDL service is not possible due to lack of telematics device
- RDL is already scheduled for today
- RDL is up-to-date

Request details

This web service is an HTTP POST request with the VIN as a path parameter. Access token needs to be used as bearer-token authorisation method and the Accept header should also be sent as shown in the example below.

REST endpoint	POST /vehicles/{vin}/rdl
HTTP Header	Accept: application/json Authorization: Bearer eyJhbGciOiJI... Content-Type: application/json
Response format	application/json

Query parameter	Type	Description
vin	string	Vehicle Identification Number. This parameter is required.

Response details

The unit state will be updated and returned as a JSON with the properties described in [Getting vehicle information](#). If an error occurs, a JSON object is returned explaining the reason for failure.

Below shows an example of a request and a response for POST `/vehicles/{vin}/rdl` using HTTP Header `Accept: application/json`:

Request

```
<Headers>
POST /vehicles/ABCDEFG0123456789/rdl HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJhbGciOiJI...
```

Response

```
{
  "vin": "ABCDEFG0123456789",
  "licensePlate": "0000ZXY",
  "nextScheduledDate": "2021-01-07",
  "lastRdlAttempt": "2020-01-22T14:55:15Z",
  "rdlStatus": "NEVER_DOWNLOADED",
  "manualAttemptsRemaining": 10
}
```

Drivers

Getting driver card information

Description

Delivers a list of driver cards where to obtain their data. This list is limited according to the access token used, and it is dependent on the visibility rights of the WEBFLEET user who granted the access to the third-party company.

The list of drivers returned by this action are identified by their Driver Card Number and their Country Code and include per driver information such as the date of last driver data downloaded, the date of the next scheduled RDL, the RDL status and more.

Request details

This is an HTTP GET request with no extra parameters and access token needs to be used as bearer-token authorisation method.

REST endpoint	GET /drivers
Response format	application/json
HTTP Header	Authorization: Bearer eyJhbGciOiJl...

Response details

Driver card information is provided in a JSON array, where every driver card is defined as a JSON object with the following properties:

Parameter	Type	Description
cardNumber	string	The driver card identifiers, consisting of the first 14 characters of the driver card.
countryCode	string	The country code of the driver in HEX format according to EC No 1360/2002.
driverId	string	Internal unique driver identifier.
lastDownloadDate	string(\$date)	The most recent date when driver card data was downloaded in the format yyyy-MM-dd and is in UTC.
name	string	The name of the driver from the driver card.
nextScheduledDate	string(\$date)	Scheduled date for the next download of the driver card. Date follows the yyyy-MM-dd format and is in UTC.

Parameter	Type	Description
rdlStatus	string	<p>RDL status of the driver with regards to the remote download from the driver card.</p> <ul style="list-style-type: none"> OK DELAYED OUTSIDE_LEGAL_COMPLIANCE NEVER_DOWNLOADED

Below there is an example of a request and a response for GET /drivers:

Request

```
GET /drivers HTTP/1.1
Host: tachoshareapi.webfleet.com
Authorization: Bearer eyJhbGciOiJI...
```

Response

```
{
  "items": [
    {
      "cardNumber": "31964257346125",
      "countryCode": "0x0F",
      "driverId": "354687",
      "lastDownloadDate": "2020-05-28",
      "nextScheduledDate": "2021-01-28",
      "rdlStatus": "OUTSIDE_LEGAL_COMPLIANCE",
      "name": "Driver Name 1"
    },
    {
      "cardNumber": "12345678901234",
      "countryCode": "0x12",
      "driverId": "907865",
      "nextScheduledDate": "2021-02-17",
      "rdlStatus": "NEVER_DOWNLOADED"
    }
  ]
}
```

As shown in the above example, the `lastDownloadDate` and `name` parameters are not returned, if no driver card data exists for a certain driver card.

Collecting driver card data

Description

This web service returns a ZIP file containing all the files downloaded from a certain driver card in a selected period. The driver card is identified by the first 14 characters of the driver card number and its [country code](#). The maximum date-range allowed is 2 years.

Request details

This web service is an HTTP GET request with `cardNumber` as parameter, indicating the driver card from which data shall be collected and also the `countryCode` of the card. Access token needs to be used as bearer-token authorisation method and the Accept header should also be sent as shown in the example below.

REST endpoint `GET /drivers/{cardNumber}/files`

Host: tachoshareapi.webfleet.com
Accept: application/octet-stream, application/json
Authorization: Bearer eyJhbGciOiJI...

Response

<Headers>

Content-Disposition: attachment; filename="C_20180604_0715_20180529_1742_L_MESSI_23196425826415.ZIP"
Content-Type: application/octet-stream

<Body>

octet-stream

Company cards

Getting company card information

Description

Delivers a list of company cards where to obtain their data. This list is limited according to the access token used, and it is dependent on the visibility rights of the WEBFLEET user who granted the access to the third-party company.

The company cards returned by this action are identified by their company card number and include per company card information such as the company name, the company card group, the expiration date of the card and its status.

Request details

This is an HTTP GET request with no extra parameters and access token needs to be used as bearer-token authorisation method.

REST endpoint	GET /companyCards
Response format	application/json
HTTP Header	Authorization: Bearer eyJhbGciOiJl...

Response details

Company card information is provided in a JSON array, where every company card is defined as a JSON object with the following properties:

Parameter	Type	Description
cardNumber	string	The company card full identifier, consisting of all 16 characters of the company card identifier. The last 3 characters identify the different copies of a same company card.
companyName	string	The name of the company from the company card.
expirationDate	string(\$date)	The date when the company card will expire and will not be able to authenticate and perform RDLs. Date follows the yyyy-MM-dd format and is in UTC.
status	string	Status of the company card with regards of its validity and readability to be used in the RDL process. <ul style="list-style-type: none">• OK• NEAR_EXPIRATION• EXPIRED• PENDING_INSERTION• REMOVED
companyCardGroup	string	The company card identifier, consisting of the first 13 characters of the company card.

Below there is an example of a request and a response for GET /companyCards:

Request

```
GET /companyCards HTTP/1.1
Host: tachoshareapi.webfleet.com
Authorization: Bearer eyJhbGciOiJI...
```

Response

```
{
  items: [
    {
      cardNumber: 'EB70361789020110',
      expirationDate: '2020-12-03',
      status: 'EXPIRED',
      companyName: 'Webfleet Solutions',
      companyCardGroup: 'EB70361789020'
    },
    {
      cardNumber: 'EB70361789020002',
      expirationDate: '2025-12-03',
      status: 'OK',
      companyName: 'Webfleet Solutions',
      companyCardGroup: 'EB70361789020'
    },
    {
      cardNumber: '42445642881F0501',
      status: 'PENDING',
      companyName: 'Solutions',
      companyCardGroup: '42445642881F0'
    }
  ]
}
```

As shown in the above example, the `expirationDate` parameter is not returned if the company card has not been used yet in an RDL.

Getting company card group information

Description

Delivers a list of company card groups where to obtain their data. This list is limited according to the access token used, and it is dependent on the visibility rights of the WEBFLEET user who granted the access to the third-party company.

The list of company card groups returned by this action are identified by their name and include per company card group, information of the level of occupation.

Request details

This is an HTTP GET request with no extra parameters and access token needs to be used as bearer-token authorisation method.

REST endpoint	GET /companyCardGroups
Response format	application/json
HTTP Header	Authorization: Bearer eyJhbGciOiJI...

Response details

Company card groups information is provided in a JSON array, where every company card group is defined as a JSON object with the following properties:

Parameter	Type	Description
name	string	The company card identifier, consisting of the first 13 characters of the company card identifier.
occupationPct	number	The occupation level of the company card group. Expressed as parts per unit, where 1.0 or above indicates full occupation.

Below there is an example of a request and a response for GET /drivers:

Request

```
GET /companyCardGroups HTTP/1.1
Host: tachoshareapi.webfleet.com
Authorization: Bearer eyJhbGciOiJI...
```

Response

```
{
  items: [
    {
      name: 'EB70361789020',
      occupationPct: 0.42
    },
    {
      name: '42445642881F0'
    }
  ]
}
```

As shown in the above example, the `occupationPct` parameter is not returned if no company card in the group is available to be used in the RDL.

Usage guidance

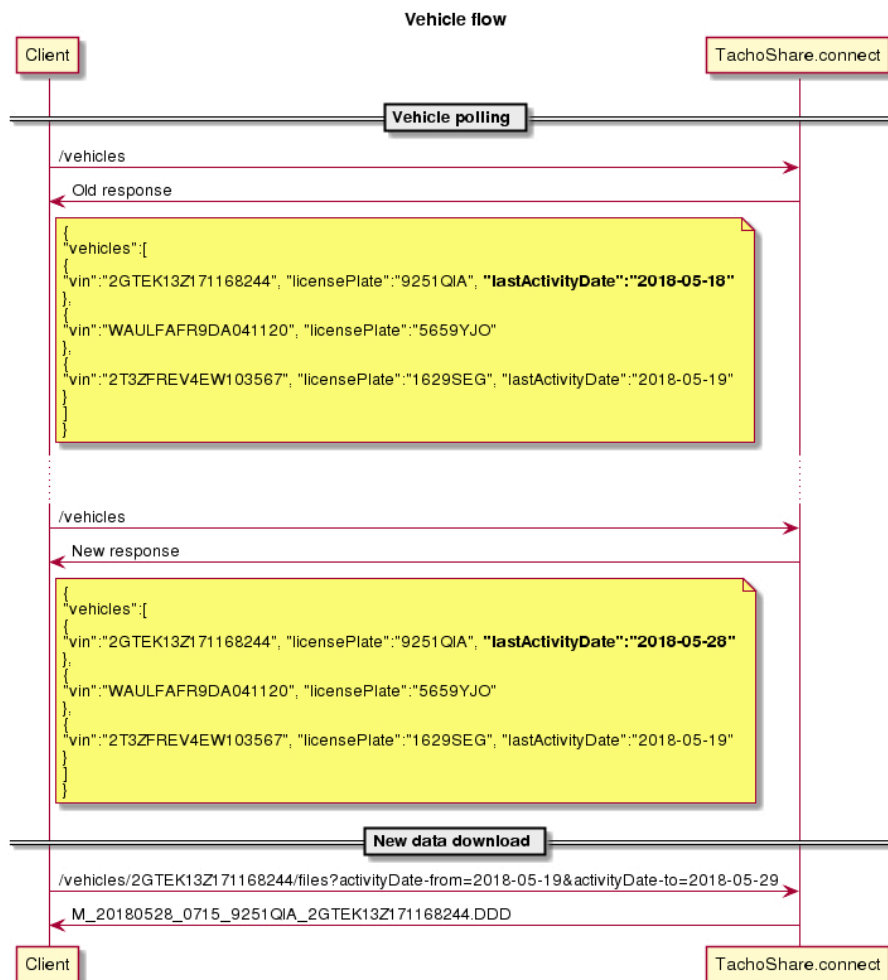
This section aims at providing guidelines on the recommended request flow.

Vehicle flow

The approach is intended to start polling from the general `getVehicles` endpoint to retrieve the available VINs and the latest data available date.

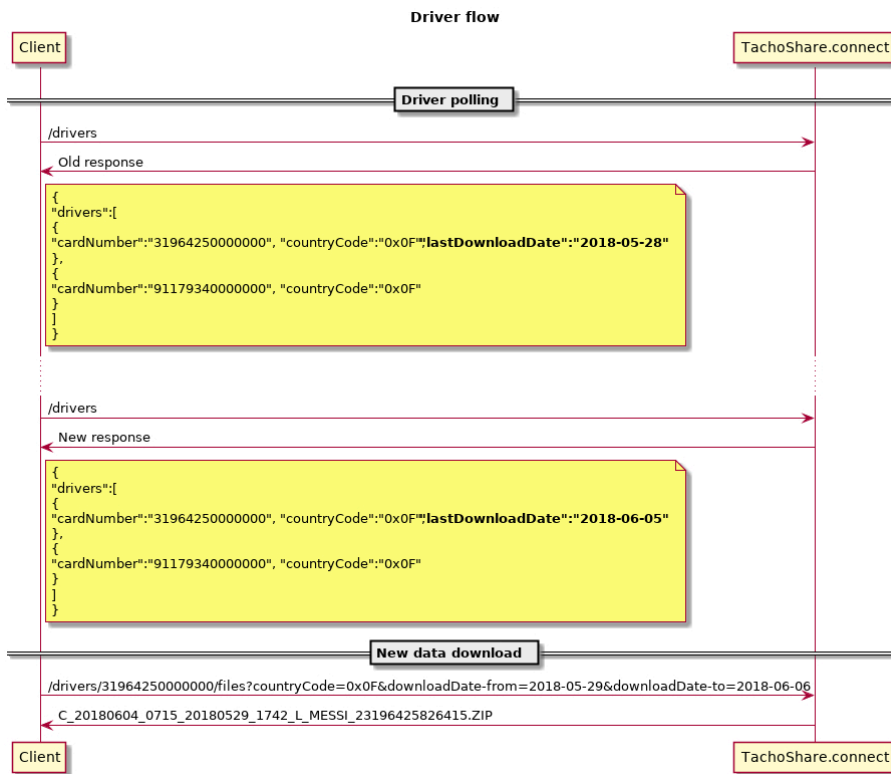
This latest data needs to be compared with the previously stored data of last activity downloaded by the client. When a new activity date is available, the client can start requesting the file for the given VIN, using the previously stored data of last activity downloaded date +1 day as `activityDate-from` (to exclude the previous last activity downloaded), and the newly available last activity date +1 day (to include the last activity downloaded) as `activityDate-to`. This operation can be performed for a single vehicle unit using the `/vehicles/{vin}/files` endpoint.

Note: The client is accountable to keep record of the last activity already downloaded.



Driver flow

Driver flow follows the same principles as the [vehicle approach](#).



Appendix

RDL status

RDL status indicates the status of the vehicles or drivers with regards to the remote download from the tachograph or the driver card respectively.

RDL status	Description
OK	The legal remote downloads are up to date.
DELAYED	The legal remote downloads are delayed.
OUTSIDE_LEGAL_COMPLIANCE	The legal remote downloads are delayed out of legal compliance.
NEVER_DOWNLOADED	Never has downloaded any data.
DISCONNECTED	LINK removed from the vehicle. <hr/> Note: This applies only to vehicles and NOT to drivers. <hr/>
EXPIRED	The legal remote download service is no longer valid. <hr/> Note: This applies only to vehicles and NOT to drivers. <hr/>

Company card status

The status of the company card indicates the validity and the readability of the card to be used for remote downloading tachograph data.

Company card status	Description
OK	The company card is inserted in the slot and available to be used to perform RDL.
NEAR_EXPIRATION	The company card is inserted in the slot and available to be used to perform RDL but its expiration date is within the next 90 days.
EXPIRED	The company card is inserted in the slot but has reached its expiration date. It's not valid anymore and cannot be used to perform RDL anymore.
PENDING_INSERTION	The company card has not been inserted in the slot yet. Some card information might be missing until it's inserted. It cannot be used to perform RDL until it is inserted.
REMOVED	The company card has been removed from the slot. It cannot be used to perform RDL. This status is usually the result of a client petition to remove the card when expired or the service has been terminated.

Response codes

Response code	Message	Description
200	The operation was successful	
204	The server successfully processed the request and is not returning any content	No data is available according to the parameters specified in the request.
400	The server cannot or will not process the request	A request parameter is missing or not properly specified.
401	Unauthorized	The access token used is not valid anymore. It needs to be refreshed with refresh token.
403	Forbidden	The access token is not valid for the required scopes.
404	The requested resource is not found	The vehicle or driver card requested doesn't exist in the system or the client has no access over it.
406	Not acceptable	The requested MIME types are not acceptable.
409	Conflict	The resource is in a state in which the service cannot be accepted.
429	Too many requests	The user has sent too many requests in a given amount of time.

Error response

All endpoints can throw an error with code 40x and the following structure:

Element	Type	Description
description	string	Detailed description of the error.
url	string	Requested URL.
traceId	string	unique identifier for the request.

Example:

Response

```
{  
  "description": "Brief description of error",  
  "url": "http://example.com/endpoint",  
  "traceId": "42iuh101f8723r"  
}
```

Driver card country codes

The table below shows the country codes used in driver cards according to the Commission Regulation (EC) No 1360/2002 of 13 June 2002. These codes are defined in the Appendix I of the regulations EU 1360/2002 and EU799/2016.

Country	Code
No information available	00
Austria	01
Albania	02
Andorra	03
Armenia	04
Azerbaijan	05
Belgium	06
Bulgaria	07
Bosnia and Herzegovina	08
Belarus	09
Switzerland	0A
Cyprus	0B
Czech Republic	0C
Germany	0D
Denmark	0E
Spain	0F
Estonia	10
France	11
Finland	12
Liechtenstein	13
Faeroe Islands	14
United Kingdom	15
Georgia	16
Greece	17

Hungary	18
Croatia	19
Italy	1A
Ireland	1B
Iceland	1C
Kazakhstan	1D
Luxembourg	1E
Lithuania	1F
Latvia	20
Malta	21
Monaco	22
Montenegro	34 (New nation not reported on (EC) 1360/2002 Appendix 1)
Republic of Moldova	23
Macedonia	24
Norway	25
Netherlands	26
Portugal	27
Poland	28
Romania	29
San Marino	2A
Serbia	35 (New nation not reported on (EC) 1360/2002 Appendix 1)
Russian Federation	2B
Sweden	2C
Slovakia	2D
Slovenia	2E
Tajikistan	37
Turkmenistan	2F
Turkey	30

Ukraine	31
Uzbekistan	36 (New nation not reported on (EC) 1360/2002 Appendix 1)
Vatican City	32
Yugoslavia	33 (Code no longer in use since 2003)
RFU	34-FC
European Community	FD
Reserved for Future Use (RFU)	38..FC
Rest of Europe	FE
Rest of the wold	FF

Resources

TachoShare.connect resources

- [Webfleet Solutions Technical Support](#)
Use the online support to access FAQs, user guides and to contact our customer support team.

Other resources

- [Commission Regulation \(EC\) No 165/2014](#)
- [JSON website](#)
- [List of nation codes \(alpha and numeric\)](#)

Revision history

TachoShare.connect

Webfleet Solutions, TachoShare.connect 2.1.0

© 2021 Webfleet Solutions. All rights reserved. WEBFLEET is a registered trademark of Webfleet Solutions B.V. or one of its Affiliates.

No part may be reproduced except as authorised by written permission. The copyright and the foregoing restriction extend to reproduction in all media.

Revision history

Revision	Date	Description	Author
1.0.0	2018-10-23	Initial release	XBQ
1.2.0	2019-01-10	Update driver card identifier length	
	2019-11-08	Introducing the new name and URLs with regards to change of company name Webfleet Solutions.	RH
1.6.0	2020-05-11	Miscellaneous edits to structure of the document.	RH

Revision history

Revision	Date	Description	Author
1.7.0	2021-03-09	<ul style="list-style-type: none"> • Updated maximum date-range for all endpoints from 90 days to 2 years. • Extended existing endpoints: <ul style="list-style-type: none"> ◦ GET /vehicles – new response parameters: nextScheduledDate, lastRdlAttempt, rdlStatus, manualAttemptsRemaining ◦ GET /drivers – new response parameters: nextScheduledDate, rdlStatus • Updated Resources, Driver Card country codes and Response Codes section with updated links and information. • Added RDL Status section and RDL Terminology. • Added new endpoint allowing to request an RDL for a vehicle unit. 	RH
1.8.0	2021-04-29	Added Getting vehicle information for a single vehicle	RH
1.9.0	2021-05-06	Added parameter <code>driverId</code> to the driver response data. Added the status <code>EXPIRED</code> for vehicle <code>rdlStatus</code> response parameter.	RH
2.0.0	2021-06-22	Changed parameter <code>licensePlate</code> from required to optional and added <code>vehicleId</code> to vehicle response data. Added <code>name</code> to driver response data.	AA
2.1.0	2021-09-16	Added new Company card endpoints with their reference, terminology and appendix sections.	AA

Revision history

Reference
