

Tachosys SaaS API



Document Change Record

Revision	Date	Owner	Details
0.1	25 th Sept 2019	James Scott-Evans	Initial Draft
1.0	20 th Dec 2019	James Scott-Evans	File upload credits
1.1	21 st Feb 2020	James Scott-Evans	Analysis returns categorisation of infringements
1.2	26 th Jun 2020	James Scott-Evans	Nominatim and Tacho Lookup added
1.3	19 th May 2021	James Scott-Evans	Infringement coding
1.4	4 th Mar 2022	James Scott-Evans	File GUID
1.5	10 th Mar 2023	James Scott-Evans	Smart 2 Tachographs
1.6	26 th Jun 2024	James Scott-Evans	Add Night work and Occasional Passenger service rules.

Table of Contents

Document Change Record	2
Table of Contents	2
1. Introduction	3
2. References	4
2.1. Joint Research Centre	4
2.2. Regulation (EU) No 165/2014	4
2.3. Implementing Regulation (EU) 2016/799	4
2.4. Regulation (EC) No. 561/2006	4
2.5. Directive 2002/15/EC	4
2.6. Regulation (EU) 2016/403	4
2.7. Nominatim	4
3. API	5
4. Error status codes	5
5. Token	5
6. Credit Balance	6
7. File upload	6
8. File parse	7
9. File name	8
10. File summary	8
11. File contents	10
12. File section	11
13. Analyse	11
14. Analyse data	15



14.1.	ActivityChangeInfo.....	15
14.2.	PlacesRecords	15
14.3.	SpecificConditionRecords	16
14.4.	PassengerServiceRecords	16
15.	Infringement table	17
16.	Nominatim	20
17.	Sample code.....	20

1. Introduction

In May 2006, digital tachographs became mandatory for use in new vehicles. In readiness, Tachosys had produced tools that enabled the merging of driving activity stored on analogue charts with the data stored on drivers' cards.

Our first 'Rules Engine' was taking this combined data and producing reports for a major freight business group. Soon after the introduction of the digital tachograph, in April 2007, the rules were changed and our 'Rules Engine' was ready producing reports under both regulations.

At the same time, we started our range of hardware products. One of the first was the digifob that brought the 'Rules Engine' to a device that enabled drivers' hours to be checked from the data held on their card. This was followed by the digifobpro and digipostpro devices that combined download tools with drivers' hours checking.

To help with our development of digital tachograph products we had created a Windows application that displayed the contents of the tachograph download files – commonly known as .DDD files. It became apparent that being able to see inside the .DDD files was useful for our resellers and even many of our end users so our Tachograph File Viewer was freely released. Little did we know at that time just how widely used and critical to many in the industry this tool would become.

Smart tachographs are the new generation of on-board mandatory digital recorders to enforce the EU legislation on professional drivers driving and resting times. They became mandatory in June 2019 but started to appear in the weeks leading up to this date.

Tachosys started work on meeting the demands of the Smart Tachographs in the summer of 2018.

Understanding the new download steps and file formats, the geo co-ordinates, and the elliptical curve signature validation, etc., that was required to work with the new equipment. We were one of the first to release compatible software and firmware; on our 'Gen Tuesday' - 2nd April 2019.

At the beginning of 2023, Tachosys began preparations for the new Smart Tachograph version 2 due for release later in the summer. These new tachographs will also need to be retrofitted to all vehicles involved in international transport by 21 August 2025.

Smart2 will record the position and time of border-crossings and will require the driver to record load/unload operations as well as load type (goods/passenger). It will also increase the record storage from 28 to 56 days to



allow for the changes to roadside enforcement of driving hours and rest periods. Additionally, they gain the ability to connect via Bluetooth and improved security and robustness against GNSS jamming and spoofing attacks.

Software as a service (SaaS) is a software distribution model in which a third-party provider hosts applications and makes them available to customers over the Internet. With the Tachosys SaaS API we are providing access to our 'Rules Engine' and 'File Viewer'.

In addition, we have our own Nominatim instance which is a tool that generates an address from a latitude and longitude. The Tachosys SaaS API provides a means to make use of this lookup.

The purpose of this document is to outline the use of the Tachosys SaaS API.

2. References

2.1. Joint Research Centre

The website of the European Commission – Joint Research Centre for Digital Tachographs has a lot of information regarding both Digital Tachographs and Smart Tachographs and the related equipment:-

<https://dtc.jrc.ec.europa.eu/index.php>

2.2. Regulation (EU) No 165/2014

Use of tachographs in road transport

<https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:02014R0165-20200820>

2.3. Implementing Regulation (EU) 2016/799

Technical specification of the Smart 2, Smart, and Digital tachographs - commonly known as Annex 1C

<https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:02016R0799-20230821>

2.4. Regulation (EC) No. 561/2006

The drivers' hours rules

<https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:02006R0561-20200820>

2.5. Directive 2002/15/EC

The working time rules

<https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32002L0015>

2.6. Regulation (EU) 2016/403

A list of categories, types and degrees of seriousness of serious infringements of drivers' hours and working time rules

<https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:02016R0403-20220523>

2.7. Nominatim

Nominatim (from the Latin, 'by name') is a tool to search OSM data by name and address (geocoding) and to generate synthetic addresses of OSM points (reverse geocoding).

<https://wiki.openstreetmap.org/wiki/Nominatim>



3. API

The API has been implemented as a secure REST service with a base address of:-

<https://saas.tachosys.com/api/>.

A content type of `application/json; charset=utf-8` will be used throughout, except when uploading files.

4. Error status codes

Status code	Use
200	OK – Success
204	No Content – File not uploaded.
401	Unauthorised – Login failed; Missing Bearer Authorization; Token has expired.
403	Forbidden – File upload credits have been exhausted.
422	Unprocessable Entity – File cannot be analysed.
500	Internal Server Error – Error occurred.

5. Token

The API uses JSON Web Tokens (JWT) bearer authentication to validate user access. To get the token to access the API the client application first needs to make a POST request with the below body to the URI of `token`.

```
{
  "Username": "<username>",
  "Password": "<password>"
}
```

The username and password will be provided for a per application use implementation.

The response will be in the following form:-

```
{
  "token": "<token>",
  "expiresIn": <minutes>
}
```

The client must send this token in the Authorization header when making requests to the protected resources of the API:-

Authorization: Bearer <token>

The token will expire after the number of minutes indicated, at which point the client will need to re-authenticate.



6. Credit Balance

Each upload of a file to Tachosys SaaS API costs one unit of credit. The current credit balance can be checked using a GET request to the URI of `creditbalance`.

The response will be in the following form:-

```
[{
  "uri": <uri>,
  "creditBalance": <creditBalance>,
  "allowance Limit": <allowanceLimit>
}, ... ]
```

`<creditBalance>` is the current balance of credits. `<allowanceLimit>` is the value at which the API will stop further file uploads.

The credit balance can be checked by visiting the following page on the SaaS site:-

<https://saas.tachosys.com/creditbalance>.

Once logged in you can see a graph showing a history of the credit usage. A settings area allows the setup of an allowance warning level and an email address to send the warning to.

7. File upload

Files are uploaded using a content-type of `multipart/form-data` POST to the URI of `file/upload`.

The response will be in the following form:-

```
{
  "fileGuid": "<fileGuid>"
}
```

`<fileGuid>` will be a globally unique identifier and should be used in later API calls that operate on the uploaded file.

If the response is 403 – Forbidden, this indicates that there are no further credits available for file uploads.



8. File parse

The uploaded file can be parsed and the signatures checked with the results returned using a POST request to the URI of `file/parse` with the below body:-

```
{
  "fileGuid": "<fileGuid>"
}
```

The response will be in the following form:-

```
{
  "isFileParseSuccessful": <isFileParseSuccessful>,
  "isVerifySignaturesSuccessful": <isVerifySignaturesSuccessful>,
  "fileParseException": "<fileParseException>",
  "verifySignatureResult": {
    "result": "<result>",
    "exception": "<verifySignatureException>",
    "sectionResults": [
      { <sectionResult> },
      { <sectionResult> }, ...
    ]
  }
}
```

`<isFileParseSuccessful>` and `<isVerifySignaturesSuccessful>` are boolean values indicating success. `<fileParseException>` and `<verifySignatureException>` return null or the text of the exception that has occurred.

`<result>` returns one of the following values:- Unsigned, Error, Failed, Success.

`<sectionResult>` is defined as follows:-

```
{
  "fileID": "<fileID>",
  "fileOccurrence": <fileOccurrence>,
  "success": <success>
}
```

For `<fileID>` and `<fileOccurrence>` refer to [11. File section](#) below.

`<success>` is a Boolean value indicating success.



9. File name

To get a formatted file name for the file using one of the known naming conventions make a POST request with the below body to the URI of `file/filename`.

```
{
  "fileGuid": "<fileGuid>",
  "downloadDate": "<downloadDate>",
  "namingConvention": <namingConvention>
}
```

The download date is required in the file name formatting of the naming conventions. See the below table for values for namingConvention.

Value	File type	Description
0	.ddd	Standard European
1	.c1b, .v1b	France
2	.tgd	Spain

The response will be in the following form:-

```
{
  "filename": "<filename>"
}
```

Note: The France and Spain naming conventions require the Vehicle Registration Nation alpha code. For Smart Tachographs (Generation 2) this means that the vehicle unit download must include Technical Data (TREP 25).

10. File summary

A summary of the contents of the file will be returned by making a POST request to the URI of `file/summary` with the below body:-

```
{
  "fileGuid": "<fileGuid>"
}
```

The response will be in the following form:-

```
{
  "type": <type>,
  "generation": <generation>,
  "driverCardSummary": <driverCardSummary>,
  "vehicleUnitSummary": <vehicleUnitSummary>,
  "workshopCardSummary": <workshopCardSummary>
}
```




See the below table for the values of <type>:-

Value	Description
0	Unknown
1	Driver card
2	Workshop card
3	Vehicle unit

<generation> will indicate which generation the tachograph file is.

<driverCardSummary> is defined as follows:-

```
{
  "cardHolderFirstNames": "<cardHolderFirstNames>",
  "cardHolderSurname": "<cardHolderSurname>",
  "cardHolderBirthDate": { <cardHolderBirthDate> },
  "cardValidityBegin": { <cardValidityBegin> },
  "cardExpiryDate": { <cardExpiryDate> },
  "cardIssuingMemberStateAlpha": "<cardIssuingMemberStateAlpha>",
  "cardNumber": {
    "driverIdentification": "<driverIdentification>",
    "ownerIdentification": "<ownerIdentification>",
    "cardConsecutiveIndex": "<cardConsecutiveIndex>",
    "cardReplacementIndex": "<cardReplacementIndex>",
    "cardRenewalIndex": "<cardRenewalIndex>" },
  "drivingLicenceNumber": "<drivingLicenceNumber>",
  "activitySummary": {
    "startDate": { <startDate> },
    "endDate": { <endDate> } }
}
```

<vehicleUnitSummary> is defined as follows:-

```
{
  "vehicleIdentificationNumber": "<vehicleIdentificationNumber>",
  "vehicleRegistrationNationAlpha": "<vehicleRegistrationNationAlpha>",
  "vehicleRegistrationNumber": "<vehicleRegistrationNumber>",
  "activitySummary": {
    "startDate": { <startDate> },
    "endDate": { <endDate> } },
  "vehicleUnitIdentificationSummary": {
    "vuManufacturerName": "<vuManufacturerName>",
    "vuPartNumber": "<vuPartNumber>",
    "vuSoftwareVersion": "<vuSoftwareVersion>" }
}
```



<workshopCardSummary> is defined as follows:-

```
{
  "cardHolderFirstNames": "<cardHolderFirstNames>"
  "cardHolderSurname": "<cardHolderSurname>",
  "cardValidityBegin": { <cardValidityBegin> },
  "cardExpiryDate": { <cardExpiryDate> },
  "cardIssuingMemberStateAlpha": "<cardIssuingMemberStateAlpha>",
  "cardNumber": {
    "driverIdentification": "<driverIdentification>",
    "ownerIdentification": "<ownerIdentification>",
    "cardConsecutiveIndex": "<cardConsecutiveIndex>",
    "cardReplacementIndex": "<cardReplacementIndex>",
    "cardRenewalIndex": "<cardRenewalIndex>" },
  "workshopName": "<workshopName>",
  "workshopAddress": "<workshopAddress>"
}
```

<startDate> and <endDate> of activitySummary are both of data type TimeReal.

The definitions of the above items can be found in Appendix 1 – 2. Data Type Definitions of the Annex 1C documentation.

11. File contents

This lists all the sections that make up the file. Can be called by making a POST request to the URI of `file/contents` passing the fileGuid as shown previously. This can be very useful when starting out using the API and with locating sections of the tachograph files. The combinations of FileID, RecordType and FileOccurrence can then be used with the File section call below.



12. File section

Sections of the uploaded file can be returned by making a POST request to the URI of `file/section` using the below body:-

```
{
  "fileGuid": "<fileGuid>",
  "FileID": "<fileID>",
  "RecordType": "<recordType>",
  "FileOccurrence": "<fileOccurrence>"
}
```

For Driver card and Workshop card files:-

`<fileID>` matches the FileID values defined in Appendix 2 - 4. Tachograph Cards Structure of Annex 1C.

`<recordType>` is un-used should be left blank.

`<fileOccurrence>` matches the FID value definition in Appendix 7 – 3.4.2 File format of Annex 1C.

For Vehicle unit files:-

`<fileID>` matches the TRTP values defined in Appendix 7 – 2.2.2.9 Transfer Data Request of Annex 1C.

`<recordType>` for Generation 1 sections this value is the Data element defined in Appendix 7 – 2.2.6.1 Positive Response Transfer Data Overview of Annex 1C. For Generation 2 sections see the RecordType values in Appendix 1 – 2. Data Type Definitions of Annex 1C.

`<fileOccurrence>` should be passed with a value of 0 (zero).

To retrieve the whole of the Activities section of a Vehicle unit file only `<fileID>` needs to be set, `<recordType>` and `<fileOccurrence>` should be left blank.

The returned data will match the file section definitions as per Annex 1C documentation.

13. Analyse

The Drivers Hours analysis of the uploaded Driver card file can be retrieved by making a POST request to the URI of `analyse` using the below body:-

```
{
  "fileGuid": "<fileGuid>",
  "HomeNation": "<homeNation>",
  "WorkingTimeAveragePeriod": "<workingTimeAveragePeriod>",
  "WorkingTimeNightWorkPeriodStarts": "<workingTimeNightWorkPeriodStarts>",
  "WorkingTimeNightWorkPeriodEnds": "<workingTimeNightWorkPeriodEnds>",
  "WorkingTimeNightWorkTimeOffset": "<workingTimeNightWorkTimeOffset>",
  "WorkingTimeNightWorkUseDaylightSaving": "<workingTimeNightWorkUseDaylightSaving>",
  "POAasBreak": "<poaAsBreak>", deprecated – should always be set to false.
  "MissingManualEntry": "<missingManualEntry>",
  "Language": "<Language>",
  "TimeZone": "<timeZone>"
}
```



<homeNation> set to a value of NationNumeric that can be found in Appendix 1 – 2. Data Type Definitions of the Annex 1C documentation.

<workingTimeAveragePeriod> set to the number of weeks to use for calculating Working Time averages. Defaults to 17.

<workingTimeNightWorkPeriodStarts> set to the number of seconds past midnight that the night work period starts. Defaults to midnight = 0 (zero).

<workingTimeNightWorkPeriodEnds> set to the number of seconds past midnight that the night work period ends. Defaults to 04:00 = 14400.

<workingTimeNightWorkTimeOffset> set to the number of seconds that the night work local time periods are ahead of UTC. For example, for UTC +01:00 set to 3600. Defaults to 0 (zero).

<workingTimeNightWorkUseDayLightSaving> set to true if the night work local time makes use of daylight saving. Defaults to true.

<poaAsBreak> set to true if POA can be assumed as Break in single driver mode. POAasBreak is deprecated as POA should not be considered BREAK in single driver mode. Only for multi-manned driving POA is considered as BREAK. Defaults to false.

<missingManualEntry> set to true if an infringement should be raised when manual entries are missing. Known to be required in the Netherlands and the UK. Use false otherwise.

<Language> a language tag of the form en-GB. If translations are not available in the chosen language, then en-GB text will be returned. Date and time formats will match the default for the language.

<timeZone> Date and times shown in the analysis text will be converted from UTC to the specified time zone that must match one of the subkey names of the HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Time Zones branch of the Windows registry.

The response will be in the following form:-

```
[
    { <analysisItem> },
    { <analysisItem> }, ...
]
```

<analysisItem> is defined as follows:-

```
{
    "type": "<type>",
    "subType": "<subType>",
    "infringement_code": "<infringement_code>"
    "infringement_level": "<infringement_level>"
    "seriousness": "<seriousness>", deprecated - use infringement code & level.
    "isOngoing": "<isOngoing>",
    "dateTime": "<dateTime>",
    "message": "<message>",
    "parameters": [ <parameter>, <parameter>, ... ]
}
```

See the below table for the values of <type> and <subtype>:-





Type	SubType
Infringement	DrivingBreakInfringement
Infringement	DailyDriveInfringement
Infringement	DailyRestInfringement
Infringement	WeeklyRestWithin6DaysInfringement
Infringement	WeeklyRestInfringement
Infringement	WeeklyRestCompensationInfringement
Infringement	WeeklyDriveInfringement
Infringement	FortnightlyDriveInfringement
Infringement	WorkingTimeBreakInfringement
Infringement	WorkingTimeBreak6Infringement
Infringement	WorkingTimeInfringement
Infringement	WorkingTimeAverageInfringement
Infringement	WorkingTimeNightWorkInfringement
Infringement	NetherlandsNightWorkShiftsInfringement
Infringement	NetherlandsFortnightlyNightWorkInfringement
Infringement	ManualEntryMissingInfringement
Infringement	PlacesRecordMissingInfringement
Information	InactivityInformation
Information	LongPeriodOfActivityInformation
Information	TimeOverlapInformation
Information	WeeklyRestInformation
Information	RestInformation
Information	OutstandingCompensationInformation
Summary	DailySummaryInformation
Summary	WeeklySummaryInformation
Summary	WorkingTimeAverageSummary
Latest	DriveAvailableNextBreakInformation
Latest	DriveAvailableNextRestInformation
Latest	NextDailyRestByInformation
Latest	NextWeeklyRestByInformation
Latest	LatestShiftSummary
Latest	WeeklyDriveAvailableInformation
Latest	FortnightlyDriveAvailableInformation

The Infringement Table, later in this document, lists the values used for *<infringement_code>*. The below table list the values of *<infringement_Level>*:-

Seriousness	Description
	When <i><type></i> is not an Infringement, this value is left blank.
NA	The Infringement has not been categorised.
MI	minor infringement
SI	serious infringement
VSI	very serious infringement
MSI	most serious infringement



<isOngoing> for Infringements this will be set to `true` when the details could change in the future, e.g. when driving is continuing when a break is overdue. This is likely when the download was taken during a driver's shift, rather than at the end.

<dateTime> in UTC when the item occurred.

<message> a description of the item.

<parameter> a value used in the message text.

14. Analyse data

The Drivers Hours analysis can also be run on uploaded `ActivityChangeInfo`, `PlacesRecords`, and `SpecificConditionRecords` data instead of an uploaded driver card file. These calls should be used when you have these tables held in a database. This has the advantage of enabling analysis across replacement cards.

To run analysis that would need to make use of derogations available during occasional passenger services journeys. These include driving breaks, daily rest extension, and 12-day rule. These require a journey plan to be included with the analysis. Use `PassengerServiceRecords` to enter the journey plan before running analysis.

14.1. ActivityChangeInfo

The data can be uploaded by using a POST request to the URI of `analyse/data/activityChangeInfo` using the below body:-

```
{
  "fileGuid": "<fileGuid>",
  "activityChangeInfo": [
    { <activityChangeInfo> },
    { <activityChangeInfo> }, ...
  ]
}
```

The definition of <activityChangeInfo> can be found in Appendix 1 – 2. Data Type Definitions of the Annex 1C documentation.

14.2. PlacesRecords

The data can be uploaded by using a POST request to the URI of `analyse/data/placesrecords` using the below body: -

```
{
  "fileGuid": "<fileGuid>",
  "placesRecords": [
    { <placesRecord> },
    { <placesRecord> }, ...
  ]
}
```

The definition of <placesRecord> can be found in Appendix 1 – 2. Data Type Definitions of the Annex 1C documentation.



14.3. SpecificConditionRecords

The data can be uploaded by using a POST request to the URI of `analyse/data/specificConditionRecords` using the below body:-

```
{
  "fileGuid": "<fileGuid>",
  "specificConditionRecords": [
    { <specificConditionRecord> },
    { <specificConditionRecord> }, ...
  ]
}
```

The definition of `<specificConditionRecord>` can be found in Appendix 1 – 2. Data Type Definitions of the Annex 1C documentation.

14.4. PassengerServiceRecords

The journey plan for occasional passenger services can be uploaded by using a POST request to the URI of `analyse/data/passengerServiceRecords` using the below body:-

```
{
  "fileGuid": "<fileGuid>",
  "passengerServiceRecords": [
    { <passengerServiceRecord> },
    { <passengerServiceRecord> }, ...
  ]
}
```

The definition of `<passengerServiceRecord>` is as follows:-

```
{
  "entryTime": {
    "ticks": <entryTime>
  },
  "passengerServiceType": <passengerServiceType>
}
```

`<entryTime>` is the date and time of the entry, expressed as number of seconds since midnight on 1 January 1970.

`<passengerServiceType>` sets whether the entry time represents start or end of the journey. Occasional Service Begin = 1, Occasional Service End = 2.

15. Infringement table

The following table lists all infringements reported by the Analyse method above.

SubType	Tachosys Code	EU 2016/403 Code	Seriousness	Legal Basis	Infringement text (English)
DrivingBreakInfringement	EU_DB1	C1	MI	Article 7 ¹	Exceeded driving time of 4h30 at {0:g} before taking a valid break of at least 45 min. Duration {1}.
		C2	SI		
		C3	VSI		
DailyDriveInfringement	EU_DD1	B1	MI	Article 6.1 ¹	Daily work period commencing {0:g} - Exceeded daily drive limit of {2}h. Duration: {1}.
		B2	SI		
		B3	VSI		Daily work period commencing {0:g} - Exceeded daily drive limit of {2}h by 50% or more. Duration: {1}.
		B4	MSI		
	EU_DD2	B5	MI		Daily work period commencing {0:g} - Exceeded daily drive limit of {2}h. Duration: {1}.
		B6	SI		
		B7	VSI		Daily work period commencing {0:g} - Exceeded daily drive limit of {2}h by 50% or more. Duration: {1}.
		B8	MSI		
DailyRestInfringement	EU_DR1	D1	MI	Article 8.2 ¹	Insufficient daily rest. {3} hours were required by {1:g}. Duration {2}.
		D2	SI		
		D3	VSI		
	EU_DR2	D4	MI		Insufficient daily rest. {3} hours were required by {1:g}. Duration {2}.
		D5	SI		
		D6	VSI		
	EU_DR3	D7	MI		Insufficient split daily rest. 3 hours + {3} hours were required by {1:g}. Duration: 3h+{2}.
		D8	SI		
		D9	VSI		
	EU_DR4	D10	MI	Article 8.5 ¹	Insufficient daily rest when double manned. {3} hours were required by {1:g}. Duration {2}.
		D11	SI		

SubType	Tachosys Code	EU 2016/403 Code		Seriousness	Legal Basis	Infringement text (English)	
		D12		VSI			
WeeklyRestWithin6DaysInfringement	EU_WR1	D19		MI	Article 8.6 ¹	No weekly rest started within 6 days of the previous weekly rest. Rest required by {0:g}. Duration: {1}.	
		D20		SI			
		D21		VSI			
WeeklyRestInfringement	EU_WR2	D13		MI	Article 8.6 ¹	Week commencing {0:d} - Insufficient weekly rest. {3} hours were required by {1:g}. Duration: {2}.	
		D14		SI			
		D15		VSI			
	EU_WR3	D16		MI			Week commencing {0:d} - Insufficient weekly rest. {3} hours were required by {1:g}. Duration: {2}.
		D17		SI			
		D18		VSI			
	EU_WR4			MI	Article 1.6 ³	Week commencing {0:d} - Insufficient weekly rest. {3} hours at home were required by {1:g}. Duration: {2}.	
				SI			
				VSI			
	EU_WR5			MI			Week commencing {0:d} - Insufficient weekly rest. {3} hours at home were required by {1:g}. Duration: {2}.
				SI			
				VSI			
WeeklyRestCompensationInfringement	EU_WC1				Article 8.6 ¹	Week commencing {0:d} - Failure to compensate for reduced weekly rest. {2} was required by {1:g}.	
WeeklyDriveInfringement	EU_WD1	B9		MI	Article 6.2 ¹	Week commencing {0:d} - Exceeded weekly driving limit of 56 hours. Duration: {1}.	
		B10		SI			
		B11		VSI			
		B12		MSI			
FortnightlyDriveInfringement	EU_FD1	B13		MI	Article 6.3 ¹	Fortnight commencing {0:d} - Exceeded fortnightly driving limit of 90 hours. Duration: {1}.	
		B14		SI			
		B15		VSI			
		B16		MSI			
WorkingTimeBreakInfringement	EU_WB1	MI		Article 5.1 ²	Insufficient working time breaks at {0:g}. Working more than {1} hours shall be interrupted by a break of at least {3} min. Duration: {2}.		
	SI						

SubType	Tachosys Code	EU 2016/403 Code		Seriousness	Legal Basis	Infringement text (English)
			VSI			
			MI			
	EU_WB2		SI			Insufficient working time breaks at {0:g}. Working more than {1} hours shall be interrupted by a break of at least {3} min. Duration: {2}.
			VSI			
WorkingTimeBreak6Infringement	EU_WB3			Article 5.1 ²		Exceeded working time of 6 hours at {0:g} before taking a valid break of at least 15 min. Duration: {1}.
			MI			
	EU_WT1		SI	Article 4 ²		Week commencing {0:d} - Exceeded weekly working time limit of 60 hours. Duration: {1}.
			VSI			
			MI			
WorkingTimeAverageInfringement	EU_WT2		SI	Article 4 ²		Period commencing {0:d} - Exceeded average working time limit of 48 hours over period of {2} weeks. Average: {1:0.00}h.
			VSI			
			MI			
WorkingTimeNightWorkInfringement	EU_NW1		SI	Article 7.1 ²		Night Work period commencing {0:g} - Exceeded daily working time of 10h in each 24h period when night work performed. Duration: {1}.
			VSI			
NetherlandsNightWorkShiftsInfringement	NL_NW1					Period commencing {0:d} - Exceeded limit on number of night shifts over period of 16 weeks. Count: {1}.
NetherlandsFortnightlyNightWorkInfringement	NL_FW1					Fortnight commencing {0:d} - Exceeded fortnightly night work limit of 20h. Duration: {1}.
ManualEntryMissingInfringement	EU_ME1	H16	VSI	Article 34.4 ⁴		Unknown activity starting at {0:g}. Duration {1}. A manual entry may be required.
PlacesRecordMissingInfringement	EU_CF1		SI	Article 34.7 ⁴		Daily work period commencing {0:g} - Location at both start and end of daily work period missing.

Note:

¹Infringement against Regulation (EC) no 561/2006.²Infringement against Directive 2002/15/EC.³Infringement against Regulation (EU) no 2020/1054.⁴Infringement against Regulation (EU) No 165/2014.



16. Nominatim

Given a latitude and longitude anywhere on the planet, Nominatim can find the nearest address using OpenStreetMap data. This lookup can be achieved by using a POST request with the below body to the URI of `nominatim/reverse`.

```
{
  "key": "<key>",
  "format": "place",
  "extratags": "0",
  "zoom": "16",
  "positions": [
    { <position> },
    { <position> }, ...
  ]
}
```

<key> will be provided for your use only and must not be shared.

<position> is defined as follows:-

```
{
  "lat": "<Latitude>",
  "lon": "<Longitude>"
}
```

<Latitude> and <Longitude> are passed as decimal degrees.

The response will be in the following form:-

```
[
  {
    "lat": "<Latitude>",
    "lon": "<Longitude>",
    "response": "<address>"
  }, ...
]
```

<address> is a string representing the address for the location.

If the response is 403 – Forbidden, this indicates that there are no further credits available.

17. Sample code

C# code for a Windows Console sample client to the API is available on GitHub from here:

<https://github.com/james-prosys/Tachosys-SaaS-Client>