

get_timerecording.php

Compatibility



This function is intended to be used with timerecording V3

If you are using the older V2 version, please use [get_timerecording.asp](#)

Returns all received timerecording bookings for the requested time range. This range may refer to the processing time in our computing center or to the booked date and time.

In case you are using bookings based on digitacho files, there is a delay of multiple hours between the booked time and when this booking is available via rti. The digitacho files with all bookings per (work) day are typically submitted at the next work day. Then this digitacho bookings will be translated into time bookings and are available after this is done. So working with digitacho bookings requires to use the insert date to get catch all bookings without a gap.

Description

GET

get_timerecording.php?company=COMPANY_RTIKEY&import=COMPANY_IMPORTKEY&start=YYYYMMDDHHMMSS&end=YY
YYMMDDHHMMSS&format=CSV&version=1&date_query=insert

Parameter

Parameter	Description	Type	Note	Mandatory
company	RTI company key	string		yes
import	RTI import key	string		yes
start	start time for output	string	the difference between start and end must not exceed 31 days	yes
end	end time for output	string	the difference between start and end must not exceed 31 days	yes
format	output format	string	valid output formats • csv • json • xml • buspronet • gfos	yes
version	version number for request	integer	currently there is only the version 1 on change of any output parameter this version number will change, too	no
date_query	set the date reference for request	string	'insert' (default) = get bookings by processing time 'gps' = get booking by booked date and time (if 'with_changes=1' active and a booked time was modified, this modified time is relevant) 'update' = ONLY for 'with_changes=1'; common result similar to option 'insert', for changed bookings, the change date will be used	no
group	Name of the personnel group to which the restriction is to be made	string	The exact name of a personnel group must be given	no
with_changes	indicates if changes, done in portal will be applied to returned bookings or not if so, bookings deleted in portal won't be returned anymore	integer	0 (default) = always return unmodified bookings 1 = apply changes so the resultset is similar to the visible result in web portal	no

Return values

This funtion returns a **string**. On error the return will be "**Error:Description**".

CSV Format

Returning a row per booking with all the data as defined below

Meta informations

Separator	Semicolon (;)
Delimiter	Double quote (")
Line break	0x0D 0x0A (CarriageReturn LineFeed)
Header	Row 1
Encoding	Windows-1252

Columns

Column	Description	
PERSON_ID	self defined person number	in personell administration listed as MA-Nr
PERSON_KEY	Booking key, used by person	
PERSON_NAME	name of person	
PERSON_CARD_NUMBER	self defineable additional booking number	in personell administration listed as Card-Nr
DATETIME	Date	format DD.MM.YYYY HH:MM:SS
ACTIVITY_ID	Short booging type definition	
ACTIVITY_NAME	Description text for booking as defined	
TERMINAL_ID	rti ident of booking terminal	as set via rti settings rti ident per vehicle
TERMINAL_NAME	name of the booking terminal	usually the vehicle sign
LAT	Latitude	
LON	Longitude	
ADDRESS	Aufgelöster Positionstext	
AREA	areas, as defined for this Position	delimiter: "% %"
DRIVERKEY	personal key of person who did the booking	
COST_CENTER_ID	technical ident of cost center	
COST_CENTER_NAME	name of used cost center	

Example

return in csv
"PERSON_ID";"PERSON_KEY";"PERSON_NAME";"PERSON_CARD_NR";"DATETIME";"ACTIVITY_ID";"ACTIVITY_NAME";"TERMINAL_ID";"TERMINAL_NAME";"LAT";"LON";""ADDRESS";"AREA";"COST_CENTER_ID";"COST_CENTER_NAME" "A23";"DF000000000123";"John Smith";"";"01.10.2019 08:00:00";"1023";"work";"CAR_17";"DD-YF 123";"51.4";"13.1";"Musterstraße 17, 01234 Musterstadt";"Hauptzentrale";"K-47";"common work" "A23";"DF000000000123";"John Smith";"";"01.10.2019 12:00:00";"1024";"break";"CAR_17";"DD-YF 123";"52.0";"11.3";"Testweg 23, 91234 Testhausen";"Müller GmbH";"";"

JSON Format

Returns a JSON formatted string with detailed data per booking

Example output

JSON output

```
[
  {
    "person": {
      "id": "A23",
      "key": "DF00000001234",
      "name": "John Smith",
      "card_nr": null
    },
    "datetime": "2019-04-01 05:26:00",
    "activity": {
      "id": "1003",
      "name": "work time"
    },
    "terminal": {
      "id": "C_32",
      "name": "DD-YF 456"
    },
    "position": {
      "lat": 51.4,
      "lon": 13.1,
      "address": "Am Wüsteberg 3, 01723 Kesselsdorf"
    },
    "area": null,
    "costcenter": {
      "id": null,
      "name": null
    }
  },
  {
    "person": {
      "id": "A23",
      "key": "DF00000001234",
      "name": "John Smith",
      "card_nr": null
    },
    "datetime": "2019-04-01 12:35:00",
    "activity": {
      "id": "1001",
      "name": "break time"
    },
    "terminal": {
      "id": "C-1",
      "name": "central"
    },
    "position": {
      "lat": 52.0,
      "lon": 11.9,
      "address": "Am Wüsteberg 3, 01723 Kesselsdorf"
    },
    "area": null,
    "costcenter": {
      "id": "K-47",
      "name": "common"
    }
  }
]
```

XML Format

XML output

```
<?xml version="1.0"?>
<bookings>
  <booking>
    <person>
      <id/>
      <key>DF00000001234</key>
      <name>John Smith</name>
      <card_nr/>
    </person>
    <datetime>2019-04-01 05:26:00</datetime>
    <activity>
      <id>1003</id>
      <name>work time</name>
    </activity>
    <terminal>
      <id>C_32</id>
      <name>DD-YF 456</name>
    </terminal>
    <position>
      <lat>51.0</lat>
      <lon>13.1</lon>
      <address>Am Wüsteberg 3, 01723 Kesselsdorf</address>
    </position>
    <area/>
    <costcenter>
      <id/>
      <name/>
    </costcenter>
  </booking>
  <booking>
    <person>
      <id/>
      <key>DF00000001234</key>
      <name>John Smith</name>
      <card_nr/>
    </person>
    <datetime>2019-04-01 12:35:00</datetime>
    <activity>
      <id>1001</id>
      <name>break</name>
    </activity>
    <terminal>
      <id>C_1</id>
      <name>central</name>
    </terminal>
    <position>
      <lat>52.3</lat>
      <lon>11.09</lon>
      <address>Am Wüsteberg 3, 01723 Kesselsdorf</address>
    </position>
    <area/>
    <costcenter>
      <id>K-47</id>
      <name>common</name>
    </costcenter>
  </booking>
</bookings>
```

Example call

call to output json

`https://map.yellowfox.de/rti/get_timerecording.php?
company=COMPANY_RTIKEY&import=COMPANY_IMPORTKEY&start=20190602000000&end=20190605235959&format=JSON`