

get_personnel.php

Getting data for personnel set in personnel administration

Description

GET

`get_personnel.php?company=RTI_COMPANYKEY&import=RTI_VEHICLEKEY&format=csv`

Parameters

Parameter	Description	Type	Note	Mandatory
company	RTI company key	string		yes
import	RTI import key	string		yes
format	export format for list	string	supported output formats: <ul style="list-style-type: none">• csv - export as csv formatted file• json - export as json• xml - export as xml default: csv	no
history	show only latest personnel key for each person or all keys for each person	int	<ul style="list-style-type: none">• 0 - return only latest key• 1 - return all keys<ul style="list-style-type: none">◦ will return one entry per key default: 0	no
id	limit exported persons list to given id	string	see <i>strict</i> parameter	no
name	limit exported persons list to given name	string	see <i>strict</i> parameter	no
pkey	limit exported persons list to given personnel key	string	see <i>strict</i> parameter	no
group	limit exported persons list to given personnel group	string	see <i>strict</i> parameter	no
strict	switch search between finding the exact given value or return all results where the search string is part of the result	int	<ul style="list-style-type: none">• 0 - return all results containing the searched string• 1 - return only results matching exactly the searched string default: 1 this setting modifies the behaviour for parameters: <i>id</i> , <i>name</i> , <i>pkey</i> , <i>group</i>	no

Return

Returning a string in selected format. On error returning "**403 - Verboten: Zugriff verweigert.**" as HTTP-Header.

Column

CSV Column name	JSON /XML key name	Name in personnel administration	Description	Note	
ID	id	employee number	self defined employee number		
NAME	name	Name	complete name of the person		
KEY	key[id]	personnel key	personnel key (for bookings via YellowFox)		
VALID_FROM	key[valid_from]	personnel key valid from	validity start for the personnel key		format: YYYY-MM-DD HH:MM:SS
VALID_TO	key[valid_to]	personnel key valid to	validity end for the personnel key		format: YYYY-MM-DD HH:MM:SS

GROUPS	groups[group]	personnel groups	comma separated list with all personnel groups assigned to the person		
CARD_NR	card_nr	employee licence number	optional licence number to assign bookings to ReinerSCT		
DRV_LICENCE_NR	drv_licence[id]	driver licence	driver licence number	only filled, if tachograph active	
DRV_LICENCE_VALID_TO	drv_licence[valid_to]	driver licence valid to	expiration date of driver licence	only filled, if tachograph active	format: YYYY-MM-DD HH:MM:SS
IDENTITY_CARD_NR	identity_card[id]	identity card	identity card number	only filled, if tachograph active	
IDENTITY_CARD_VALID_TO	identity_card[valid_to]	identity card valid to	expiration date of identity card	only filled, if tachograph active	format: YYYY-MM-DD HH:MM:SS
PASSPORT_NR	passport[id]	passport	passport number	only filled, if tachograph active	
PASSPORT_VALID_TO	passport[valid_to]	passport valid to	expiration date of passport	only filled, if tachograph active	format: YYYY-MM-DD HH:MM:SS
ADR_CERTIFICATE_NR	adr_certificate[id]	ADR certificate	number of ADR certificate (carriage of dangerous goods by road)	only filled, if tachograph active	
ADR_CERTIFICATE_VALID_TO	adr_certificate[valid_to]	ADR certificate valid to	expiration date of ADR certificate	only filled, if tachograph active	format: YYYY-MM-DD HH:MM:SS
BKRFQG_VALID_TO	bkrfqg[valid_to]	qualification by keynumber 95 valid to	expiration date of german "Berufskraftfahrerkarifikationsgesetz"	only filled, if tachograph active	format: YYYY-MM-DD HH:MM:SS
BIRTHDATE	birthdate	date of birth	date of birth of person	only filled, if tachograph active If day and/or month are unknown, the birthdate is filled up with "XX" for the unknown values according to german passport law (example: 1980-06-XX or 1980-XX-XX)	format: YYYY-MM-DD HH:MM:SS
JOB_START_DATE	job_start_date	employee since	start date of being an employee at this company	only filled, if tachograph active	format: YYYY-MM-DD HH:MM:SS
DRV_LICENCE_CHECK_TAG_ID	drv_licence_check[tag_id]	NFC tag id	id of nfc tag for driver licence check		
DRV_LICENCE_CHECK_LAST_CHECK	drv_licence_check[last_check]	last [driver licence] check	date of latest driver licence check		format: YYYY-MM-DD HH:MM:SS
DRV_LICENCE_CHECK_NEXT_CHECK	drv_licence_check[next_check]	next [driver licence] check	date of next driver licence check		format YYYY-MM-DD
DRV_LICENCE_CHECK_INTERVAL	drv_licence_check[interval]	[driver licence] check interval	timerange between 2 driver licence checks in seconds		

CSV Format

Sending a correct request will return a csv formatted file with a headline as first line containing column names. The data itself will be returned in following rows.

Meta informations

Separator	semicolon (;)
text limiter	double quote (")
line brak	0x0D 0x0A (CarriageReturn LineFeed)
header	row 1

Header

header informationen

```
"ID";"NAME";"KEY";"VALID_FROM";"VALID_TO";"GROUPS";"CARD_NR";"DRV_LICENCE_NR";"DRV_LICENCE_VALID_TO";"
IDENTITY_CARD_NR";"IDENTITY_CARD_VALID_TO";"PASSPORT_NR";"PASSPORT_VALID_TO";"ADR_CERTIFICATE_NR";"
ADR_CERTIFICATE_VALID_TO";"BKRFQG_VALID_TO";"BIRTHDATE";"JOB_START_DATE"
```

Beispiel

example

```
"ID";"NAME";"KEY";"VALID_FROM";"VALID_TO";"GROUPS";"CARD_NR";"DRV_LICENCE_NR";"DRV_LICENCE_VALID_TO";"
IDENTITY_CARD_NR";"IDENTITY_CARD_VALID_TO";"PASSPORT_NR";"PASSPORT_VALID_TO";"ADR_CERTIFICATE_NR";"
ADR_CERTIFICATE_VALID_TO";"BKRFQG_VALID_TO";"BIRTHDATE";"JOB_START_DATE";"DRV_LICENCE_CHECK_TAG_ID";"
DRV_LICENCE_CHECK_LAST_CHECK";"DRV_LICENCE_CHECK_NEXT_CHECK";"DRV_LICENCE_CHECK_INTERVAL"
"0012";"Max Mustermann";"DF1234567890";"2018-01-01 06:43:00";"";"Fahrer,Dresden";"";"YFDD9876";"2021-11-10 23:
59:59";"";"";"";"2023-04-30 23:59:59";"";"2020-07-31 23:59:59";"2021-11-30 23:59:59";"1990-02-19 00:00:00";"
2014-09-01 00:00:00";""4729000A3341";"2020-03-01 10:43:00";"2020-03-15";"1209600"
```

JSON Format

Retrun a json formatted string with all the available data matching the request parameters. The entries will be returned as key - value relation. Some data such as personnel key or personnel groups are grouped in a sub object.

Beispiel

example

```
[
  {
    "id": "0013",
    "name": "Marion Musterfrau",
    "key": {
      "id": "DF9876543210",
      "valid_from": "2016-01-01 00:00:00",
      "valid_to": null
    },
    "card_nr": null,
    "groups": [
      "Fahrer",
      "Berlin"
    ],
    "deleted": null,
    "drv_licence": {
      "id": "asd09472845",
      "valid_to": "2017-02-24 23:59:59"
    },
    "identity_card": {
      "id": "hal2001",
      "valid_to": "2017-01-11 23:59:59"
    },
    "passport": {
      "id": null,
      "valid_to": null
    },
    "adr_certificate": {
      "id": null,
      "valid_to": "2019-05-23 23:59:59"
    },
    "bkfqq": {
      "valid_to": null
    },
    "birthdate": "1998-05-13 00:00:00",
    "job_start_date": "2016-09-01 00:00:00"
  }
]
```

```

    },
    {
        "id": "0014",
        "name": "Patrick Berger",
        "key": {
            "id": "00047392",
            "valid_from": "2017-07-05 13:20:00",
            "valid_to": null
        },
        "card_nr": "7937102",
        "groups": [
            "B\u00f0fcro",
            "Berlin"
        ],
        "deleted": null,
        "drv_licence": {
            "id": "jvg7820185",
            "valid_to": "2014-07-18 23:59:59"
        },
        "identity_card": {
            "id": "ppa4711",
            "valid_to": "2014-07-19 23:59:59"
        },
        "passport": {
            "id": null,
            "valid_to": null
        },
        "adr_certificate": {
            "id": null,
            "valid_to": null
        },
        "bkfqq": {
            "valid_to": "2019-07-18 25:59:59"
        },
        "birthdate": "1996-07-17 00:00:00",
        "job_start_date": "2014-07-17 00:00:00",
        "drv_licence_check": {
            "tag_id": "4729000A3341",
            "last_check": "2020-03-01 10:43:00",
            "next_check": "2019-03-15",
            "interval": 1209600
        }
    }
]

```

XML Format

Return an xml formatted string with main key "<persons>" and data for each single person then grouped in a "<person>" object. Some data such as personnel key or personnel groups are grouped in a sub object.

Beispiel

example

```

<?xml version="1.0"?>
<persons>
  <person>
    <id>0012</id>
    <name>Max Mustermann</name>
    <key>
      <id>DF1234567890</id>
      <valid_from>2018-01-01 06:43:00</valid_from>
      <valid_to/>
    </key>
    <card_nr/>
    <groups>
      <group>Fahrer</group>
    </groups>
  </person>
</persons>

```

```

        <group>Dresden</group>
    </groups>
    <deleted/>
    <drv_licence>
        <id>asd09472845</id>
        <valid_to>2020-04-23 23:59:59</valid_to>
    </drv_licence>
    <identity_card>
        <id/>
        <valid_to/>
    </identity_card>
    <passport>
        <id>yfdd01723</id>
        <valid_to>2023-09-17 23:59:59</valid_to>
    </passport>
    <adr_certificate>
        <id/>
        <valid_to>2018-09-30 23:59:59</valid_to>
    </adr_certificate>
    <bkfqq>
        <valid_to/>
    </bkfqq>
    <birthdate/>
    <job_start_date>2012-05-01 00:00:00</job_start_date>
</person>
<person>
    <id>0013</id>
    <name>Marion Musterfrau</name>
    <key>
        <id>DF9876543210</id>
        <valid_from>2016-01-01 00:00:00</valid_from>
        <valid_to>2019-01-01 00:00:00</valid_to>
    </key>
    <card_nr/>
    <groups>
        <group>Fahrer</group>
        <group>Berlin</group>
    </groups>
    <deleted/>
    <drv_licence>
        <id>asd09472845</id>
        <valid_to>2020-04-23 23:59:59</valid_to>
    </drv_licence>
    <identity_card>
        <id/>
        <valid_to/>
    </identity_card>
    <passport>
        <id>yfdd01723</id>
        <valid_to>2023-09-17 23:59:59</valid_to>
    </passport>
    <adr_certificate>
        <id/>
        <valid_to>2018-09-30 23:59:59</valid_to>
    </adr_certificate>
    <bkfqq>
        <valid_to/>
    </bkfqq>
    <birthdate/>
    <job_start_date>2012-05-01 00:00:00</job_start_date>
    <drv_licence_check>
        <tag_id>4729000A3341</tag_id>
        <last_check>2020-03-01 10:43:00</last_check>
        <next_check>2020-03-15</next_check>
        <interval>1209600</interval>
    </drv_licence_check>
</person>
</persons>

```

Examples

Abfrage aller Personen im CSV Format

```
https://map.yellowfox.de/rti/get_personnel.php?company=COMPANY_RTIKEY&import=IMPORT_RTIKEY&format=csv
```

Abfrage aller Personen der Gruppe "Fahrer" mit allen Schlüsseln

```
https://map.yellowfox.de/rti/get_personnel.php?  
company=COMPANY_RTIKEY&import=IMPORT_RTIKEY&format=xml&group=Fahrer&history=1
```

Abfrage der Daten einer Person mit einer bestimmten ID

```
https://map.yellowfox.de/rti/get_personnel.php?  
company=COMPANY_RTIKEY&import=IMPORT_RTIKEY&format=json&history=1&id=1234&strict=1
```