

Personeel WEB API

Implementation guide

Document version	V2.12
Release date	May 12, 2026
Last modified	May 7, 2026
Last modified by	Pierre-Marie Mahieu

Contents

Document description.....	4
API Basics.....	5
Endpoint.....	5
Versioning.....	5
Character sets.....	5
Request Format and Responses.....	6
JSON Basics.....	6
HTTP Headers.....	7
Request Headers.....	7
Response Headers.....	8
HTTP Status Codes.....	8
Usage restrictions.....	9
Ground rules.....	9
Privacy concerns.....	9
Disclaimer.....	9
Security.....	10
Transport security.....	10
Secure Socket Layer.....	10
Authentication endpoint.....	10
Request authentication.....	10
Calling resources without Access Token.....	10
Getting an Access Token by client credentials.....	11
Getting access for 1 or multiple scopes.....	11
Request token for 1 scope.....	11
Request token for multiple scopes.....	12
Use of access token.....	13
Request.....	13
Resources.....	14
Employees.....	14
GET /employees?schoolyear={schoolYear}&structure={structure}.....	14
GET /employees/{personId}?schoolyear={schoolYear}&structure={structure}.....	23
Assignments.....	24
GET /employees/assignments? schoolyear={schoolYear}&structure={structure}&inclNonSubHours={inclNonSubHours}.....	24
GET /employees/{personId}/assignments? schoolyear={schoolYear}&structure={structure}&inclNonSubHours={inclNonSubHours}.....	29
Interruptions.....	31
GET /employees/interruptions?schoolyear={schoolYear}&structure={structure}.....	31
GET /employees/{personId}/interruptions?schoolyear={schoolYear}&structure={structure}.....	34
PUT /employees/{personId}/interruptions/{interruptionId}.....	36
DELETE /employees/interruptions/{interruptionId}.....	41

Diplomas.....	43
GET /employees/diplomas?schoolyear={schoolYear}&structure={structure}.....	43
GET /employees/{personId}/diplomas?schoolyear={schoolYear}&structure={structure}.....	44
Appointments.....	45
GET /employees/appointments?schoolyear={schoolYear}&structure={structure}.....	45
GET /employees/{personId}/appointments?structure={structure}.....	46
Photos.....	47
GET /employees/photos?schoolyear={schoolYear}&structure={structure}&changedSince= {changedSince}.....	47
GET /employees/{personId}/photos.....	48
Own fields.....	49
GET /employees/ownfields?schoolyear={schoolYear}&structure={structure}.....	49
GET /employees/{personId}/ownfields?schoolyear={schoolYear}&structure={structure}.....	51
Attachments.....	53
PUT /employees/interruptions/{interruptionId}/attachments/{attachmentId}.....	53
DELETE /employees/interruptions/{interruptionId}/attachments/{attachmentId}.....	55
Communication numbers.....	57
PUT /employees/{personId}/communication-number.....	57
DELETE /employees/{personId}/communication-number/{communicationNold}.....	61
Documents.....	63
GET /employees/documents?schoolyear={schoolYear}.....	63
GET /employees/{personId}/documents?schoolyear={schoolYear}.....	65
GET /employees/{personId}/documents/{documentId}/pdf.....	66
POST /employees/{personId}/documents/{documentId}/signeddocument.....	67
APPENDIX Error-codes and descriptions.....	69
Document version history.....	73
References.....	76
List of attachments.....	77
Sample code (.NET).....	78

Document description

This document describes the “Personeel” Web API and provides details required for a technical implementation. It defines the communication model to use when talking to the Web API, including the required security algorithms. It describes all of the available methods in detail and is illustrated with examples.

Ownership

This document is written and maintained by Informat.

API Basics

The provided webservice is implemented as a RESTful service, based on the principles defined by Roy Fielding in his doctoral dissertation at UC Irvine in 2000.

Representational state transfer (REST) is an abstraction of the architecture of the World Wide Web; more precisely, REST is an architectural style consisting of a coordinated set of architectural constraints applied to components, connectors, and data elements, within a distributed hypermedia system. REST ignores the details of component implementation and protocol syntax in order to focus on the roles of components, the constraints upon their interaction with other components, and their interpretation of significant data elements.

This chapter describes the basic information you need to interact with the services provided by Informat within the Personeel context.

Endpoint

The **production environment** is available via:

```
endpoint = personeelsapi.informatsoftware.be
```

Versioning

The Personeel WEB API uses header embedded versioning to maintain backwards compatibility. This custom header is called "Api-Version".

Note that header names are case-insensitive.

C# coding example

```
Request.Headers.Add("Api-Version", "1");
```

The version number isn't mandatory. If not provided, it'll take it's default value which is 1. Providing a version outside the accepted version-range will result in a 400 error (Bad request).

Using this allows users to upgrade their implementations at their own pace. If a release creates a change that will break the workings of previous versions, e.g. a change in the underlying data structure, or if a significant change in the internal logic is added, this will be announced at least 1 month in advance, to allow time for client implementations to upgrade. These changes will be kept to a minimum, and where possible bundled, to guarantee consistent availability.

Character sets

All service response objects will be formatted with UTF-8 encoding to ensure compatibility with most languages.

Request Format and Responses

JSON Basics

The majority of requests and responses to the WEB API use the JavaScript Object Notation (JSON) for formatting the content and structure of the data and responses.

JSON is used because it is the simplest and easiest solution for working with data within a web browser, as JSON structures can be evaluated and used as JavaScript objects within the web browser environment.

JSON supports the same basic types as supported by JavaScript, these are:

- **Number** (either integer or floating-point).
- **String**; this should be enclosed by double-quotes and supports Unicode characters and backslash escaping. For example: "A String"
- **Boolean** - a true or false value. You can use these strings directly. For example: { "value": true }
- **Array** - a list of values enclosed in square brackets. For example: ["one", "two", "three"]
- **Object** - a set of key/value pairs (i.e. an associative array, or hash). The key must be a string, but the value can be any of the supported JSON values. For example:

```
{  
  "servings" : 4,  
  "subtitle" : "Easy to make in advance, and then cook when ready",  
  "cooktime" : 60,  
  "title" : "Chicken Coriander"  
}
```

Handling dates (and time)

Dates should always be sent to the server in either UTC format or using the yyyy-MM-dd (hh:mm:ss) notation.

Example:

```
2019-09-30T15:30:22.000Z           // Valid UTC Format
2019-09-30T15:30:22+0100         // Valid UTC Format
2019-09-30
2019-09-30 15:30:22
```

If you're using javascript Date() objects in your request body, your browser will automatically convert these to the UTZ format.

Example:

```
// javascript
var someDate = new Date(2019,05,21) // will be sent as 2019-05-21T00:00:00.000Z
```

If you need to send a date and time object, use the UTC format or use yyyy-MM-dd hh:mm:ss

Example:

```
2019-09-30T22:30:00.000Z           // UTC Format
2019-09-30 22:30:00
```

Boolean

Booleans should always be formatted as true/false (lower case, no quotation marks).

Example:

```
{
  "isAdmin": true,
  "canAccess": false
}
```

HTTP Headers

Because the WEB API uses HTTP for all communication, you need to ensure that the correct HTTP headers are supplied (and processed on retrieval) so that you get the right format and encoding. Different environments and clients will be more or less strict on the effect of these HTTP headers (especially when not present). Where possible you should be as specific as possible.

Request Headers

Content-type

Specifies the content type of the information being supplied within the request. The specification uses MIME type specifications. For the majority of requests this will be JSON (*application/json*). For some settings the MIME type will be plain text. When uploading attachments it should be the corresponding MIME type for the attachment or binary (*application/octet-stream*).

The use of the correct *Content-type* header on a request is required, unless the body is empty.

Accept

Specifies the list of accepted data types to be returned by the server (i.e. that are accepted/understandable by the client). The format should be a list of one or more MIME types, separated by colons.

For the majority of requests the definition should be for JSON data (*application/json*). For attachments you can either specify the MIME type explicitly, or use **/** to specify that all file types are supported. If the *Accept* header is not supplied, then the **/** MIME type is assumed (i.e. client accepts all formats).

The use of *Accept* in queries for Personnel is not required, but is recommended as it helps to ensure that the data returned can be processed by the client.

Response Headers

Response headers are returned by the server when sending back content and include a number of different header fields, many of which are standard HTTP response header and have no significance to Personeel operation. The list of response headers important to Personeel are listed below.

HTTP Status Codes

With the interface to Leerlingen working through HTTP, error codes and statuses are reported using a combination of the HTTP status code number, and corresponding data in the body of the response data.

A list of the error codes returned by the WEB API, and generic descriptions of the related errors are provided below. The meaning of status codes for specific request types is provided in the corresponding API call reference.

Code	Text	Description
200	OK	Request completed successfully.
400	Bad Request	Bad request structure. The error can indicate an error with the request URL, path or headers. Differences in the supplied MD5 hash and content also trigger this error, as this may indicate message corruption.
401	Unauthorized	The item requested was not available using the supplied authorization, or authorization was not supplied.
403	Forbidden	The requested item or operation is forbidden.
404	Not Found	The requested content could not be found.
405	Resource Not Allowed	A request was made using an invalid HTTP request type for the URL requested. For example, you have requested a PUT when a POST is required. Errors of this type can also triggered by invalid URL strings.
406	Not Acceptable	The requested content type is not supported by the server.
409	Conflict	Request resulted in an update conflict.
415	Bad Content Type	The content types supported, and the content type of the information being requested or submitted indicate that the content type is not supported.
416	Requested Range Not Satisfiable	The range specified in the request header cannot be satisfied by the server.
500	Internal Server Error	The request was invalid, either because the supplied JSON was invalid, or invalid information was supplied as part of the request.

Usage restrictions

To be able to balance the load on our servers and the impact on our overall performance, this service requires client implementations to comply with some basic rules. In case a specific resource or method requires additional rules, these will be added to the corresponding chapters.

Ground rules

- Cache retrieved data as much as possible, use the provided ID's and references to maintain data consistency.
- The privacy and security of the private key is the responsibility of the end user. In case the private key is compromised, the end user is required to inform Informat as soon as possible. We will then take the necessary steps to ensure the safety of the data. A new private key will then be issued.

Privacy concerns

The user is responsible for the safe and correct processing of all personal data, conform privacy regulations, as is stipulated in the contract between the user and the supplier, Informat.

Disclaimer

All requests to the service are logged and monitored to aid in the improvement of the service, to help troubleshoot issues and to avoid abuse.

If Informat finds that the user is found guilty of abuse, Informat reserves the right to terminate the users' access to the service if the abuse continues after multiple warnings.

Security

Transport security

Secure Socket Layer

To ensure the safety of the transferred data and to prevent data-theft, this service operates only via HTTPS.

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols designed to provide communication security over the Internet. They use X.509 certificates and hence asymmetric cryptography to authenticate the counterparty with whom they are communicating, and to exchange a symmetric key. This session key is then used to encrypt data flowing between the parties. This allows for data/message confidentiality, and message authentication codes for message integrity and as a by-product, message authentication.

Authentication endpoint

The **production environment** is available via:

```
identityEndpoint = https://www.identityserver.be
```

Request authentication

Modern secure applications often use access tokens to ensure a user has access to the appropriate resources, and these access tokens typically have a limited lifetime. This is done for various security reasons: for one, limiting the lifetime of the access token limits the amount of time an attacker can use a stolen token. Also, the information contained in or referenced by the access token could become stale.

Calling resources without Access Token

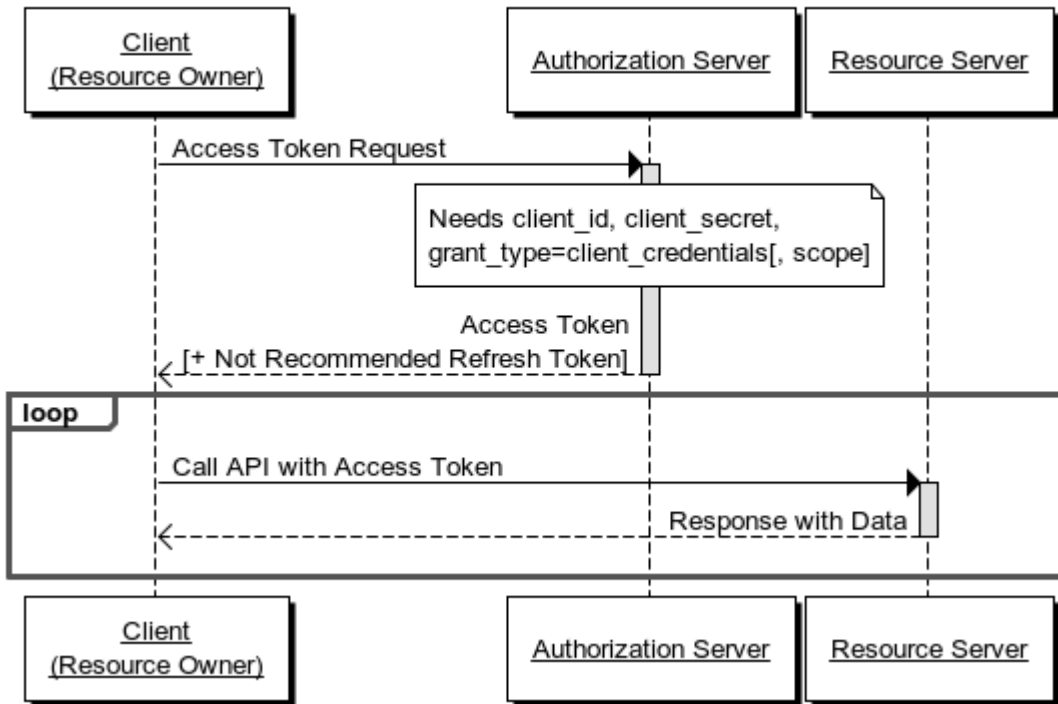
When trying to access the API without Access Token, an Unauthorized response will be returned.

```
HTTP/1.1 401 Unauthorized  
WWW-Authenticate: Bearer
```

Getting an Access Token by client credentials

The Client Credentials grant type is used by clients to obtain an access token by using a clientId and clientSecret.

Client Credentials Grant Flow



Getting access for 1 or multiple scopes

A scope is a permission that is set on a token, a context in which that token may act.

For example, a token with the **api_informat_sas_personeel.personeel.123456** scope is permitted to read or write data to the personeelsapi for the specified institute number, 123456 in this example. Otherwise you would be denied access.

You can request access for 1 scope but it's also possible to request a token for multiple scopes. See the following examples.

Request token for 1 scope

Request

```
POST Error! Hyperlink reference not valid. HTTP/1.1
...
Content-Type: application/x-www-form-urlencoded

grant_type: client_credentials
client_id: <received client id>
client_secret: <received client secret>
scope: api_informat_sas_personeel.personeel.123456
```


Resources

This chapter lists all the resources available on the webservice. A resource can be thought of as an entity which you can get data for.

Employees

This chapter describes how to:

- Get all employees for an institute-number, school year and structure (2 versions);
- Get an employee by personId.

It describes all the available methods, their request/response format and provides input/output examples.

GET /employees?schoolyear={schoolYear}&structure={structure}

Gets all the employees for the combination institute number, school year and structure. The institute number is defined via the mandatory header "InstituteNo" and the school year & structure can be provided via the URL.

As already mentioned, there are **2 versions** of this resource. Switching between them can be done via the version number in the header.

Request version 1

Headers

Besides the mandatory headers, an optional [version number](#) can be provided.

URL

Field	Type	Required/optional	Description
structure	string	required	Limits the output results to employees who have ever had an assignment within the given structure (hoofdstructuur) and institute number. Format: "211" If not provided a bad request (400) error will be returned.
schoolYear	string	optional	Limits the output results to employees with an assignment within the given schoolyear. Format: "2022-23"

Body

This request has an empty body.

Response version 1

The response contains a list of registrations.

Field	Type	Description
personId	integer	Unique integer-value for the employee within the database.
naam	string	Employee's last name.
voornaam	string	Employee's first name.

stamnr	string	Employee's stamnummer. Format: 11 characters (digits)
ambtcode	string	Code of employee's main function (hoofdambt) Format: 8 characters (digits)
isActive	bool	Indicates whether this employee is marked as active.
email	array	List of email addresses, currently limited to those of type school.

Example request version 1

Request

```
GET https://<endpoint>/employees?structure=211 HTTP/1.1
InstituteNo: 999999
Api-Version: 1
Authorization: BEARER
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1bm1xdWVfbmFtZSI6IklkluZm9ybWV0IiwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdCIsImF1dG8iOiJ1b3RlZjAwMjd1bnZmOTU0MzVmNmZc4MWEwY2ZkZDU5ZTVkIiwiaXZlbnVlIjoiaXNTc0NTIiwiaXN0IjoiODM3LCJ1b3RlZjE1NzQzNDkwMzd9.WOXKKj_a1-BaS3syFf-3k-ZT9JGS1_MLdHo4PqnE8eY
Accept: application/json
Content-Type: application/json
```

Example response version 1

Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Access-Control-Allow-Origin: *
Date: Mon, 03 Jul 2023 16:52:42 GMT
[
  {
    "personId": 4074,
    "naam": "Porter",
    "voornaam": "Tommy",
    "stamnr": "12358132134",
    "ambtcode": null,
    "isActive": true,
    "email": []
  },
  {
    "personId": 4481,
    "naam": "Luten",
    "voornaam": "Lotte",
    "stamnr": "55891442333",
    "ambtcode": "0000236",
    "isActive": true,
    "email": [
      "lotte.luten@school.be"
    ]
  },
  {...}
]
```

Request version 2

Headers

Besides the mandatory headers, an optional [version number](#) can be provided. But in this case Api-Version 2 must be added, otherwise the path we always lead to version 1.

URL

Field	Type	Required/optional	Description
schoolYear	string	required	For current & future school years: It limits the output results to employees with an assignment within the given schoolyear (and instituteNo) or which are marked as active for your instituteNo. For passed school years: It limits the output results to employees with an assignment within the given schoolyear (and instituteNo). Also used to determine the instituteNo's for the properties "EersteDienstScholengroep" and "EersteDienstScholengemeenschap". Format: "2023-24" If not provided a bad request (400) error will be returned.
structure	string	optional	This is an actually an additional restriction on the school year filter. Also used to determine - The properties "HoofdAmbt", "EersteDienstSchool" & "IsActive", - The list of "staff groups"; in a more detailed way, by means of the combination instituteNo & structure. Note: Only structures "311" & "312" are taken into account. Format: "311"

Body

This request has an empty body.

Response version 2

The response contains a list of registrations.

Field	Type	Description
pPersoon	integer	Unique integer-value for the employee within the database.
personId	Guid	Unique identifier for the employee. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50}
personald	Guid	Unique identifier for the employee within the persona platform. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50}
naam	string	Employee's last name.
voornaam	string	Employee's first name.
bijkomendeVoornamen	string	Employee's additional names.
stamnr	string	Employee's stamnummer. Format: 11 characters (digits)

sortName	string	Employee's sortable name.
nickname	string	Employee's nickname.
geslacht	string	Employee's sex. Possible values: "M" or "V"
initialen	string	Employee's initials.
geboortedatum	date	Employee's date of birth. Format: yyyy-MM-dd
geboorteplaats	string	Employee's place of birth.
geboortelandCode	string	Employee's official country of birth code.
nationaliteitCode	string	Employee's official nationality code.
rijksregisternr	string	Employee's national registration number for Belgium residents. Format: "yymmddxxxxx" if provided
bisnr	string	Employee's national registration number for a foreign person. Format: 11 characters (digits) if provided
bank	object	Single object representing employee's private bank account. Returns a "null" if not provided.
iban	string	Employee's international bank account number.
bic	string	Employee's Bank Identification Code. Format: 8 to 11 characters if provided
hoofdAmbt	object	Single object representing employee's main function within the provided instituteNo and optional structure. Only "311" & "312" are taken into account. Returns a "null" if the value could not be determined.
code	string	Code of employee's main function. Format: 8 characters (digits)
naam	string	Description of employee's main function.
eersteDienstSchool	date	Date of first employment within the school (provided instituteNo and optional structure).
eersteDienstScholengroep	date	Date of first employment within the school group.
eersteDienstScholengemeenschap	date	Date of first employment within the school community.
isActive	bool	Indicates whether this employee is marked as active for the provided instituteNo and optional structure.
effectiefInDienst	bool	Indicates whether this employee is marked as 'actually employed' for the provided instituteNo and optional structure.
pensioendatum	date	Employee's possible retirement date.
isMindervalide	bool	Indicates whether this employee is a person with disabilities.
isOverleden	bool	Indicates whether this employee is deceased.
datumOverleden	date	Employee's date of death.

burgStand	int	Employee's civil status. Possible values: 1 = Ongehuwd 2 = Gehuwd of wettelijk samenwonend 3 = Weduwnaar/weduwe 4 = Gescheiden 5 = Feitelijk gescheiden 6 = Onbekend
aantalKinderen	int	Number of children
mindervalideKinderen	int	Number of disabled children
aantalZorgbehoevenden	int	Number of care-dependent relatives above the age of 66
aantalAndereTenLaste	int	Number of other dependents
aantalMindervalideAndereTenLaste	int	Number of other disabled dependents
partner	object	Single object representing employee's partner. Returns a "null" if not provided.
id	Guid	Unique identifier for the partner (as relationship). Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50}
naam	string	Partner's last name.
voornaam	string	Partner's first name.
geboortedatum	date	Partner's date of birth. Format: yyyy-MM-dd or null if empty
nationaliteitCode	string	Partner's official nationality code.
isOverleden	bool	Indicates whether the partner is deceased.
datumOverleden	date	Partner's date of death.
inkomen	int	Partner's Income. Possible values: 0 = Geen 1 = Ja (hoger inkomen) 2 = Beperkt inkomen andere dan pensioenen 3 = Beperkt inkomen: pensioenen
adressen	array	List of domicile or residence addresses.
pAdres	int	Unique integer-value for the address within the database.
id	Guid	Unique identifier for the address. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50}
aanspreekTitel	string	Addressable title.
aanspreekNaam	string	Addressable name.
straat	string	Street name.
nummer	string	House number & Alpha number.
bus	string	Bus number.
postcode	string	Main postal code.
gemeente	string	Submunicipality.
landCode	string	Official country code.
isDomicilie	bool	Indicates whether this address is a domicile address.

isVerblijf	bool	Indicates whether this address is a residence address.
comnrs	array	List of communication numbers.
pComnr	int	Unique integer-value for the communication number within the database.
id	Guid	Unique identifier for the communication number. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50}
type	string	Communication number type. Examples: Domicilie, Verblijf, Alternatief nummer, Noodnummer, Privé nummer, Werk, Partner, ...
soort	string	Communication number sort. Possible values: Telefoon, Fax or Gsm
nr	string	Communication number.
emailadressen	array	List of email addresses
pEmail	int	Unique integer-value for the email address within the database.
id	Guid	Unique identifier for the email address. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50}
type	string	Email address type. Examples: Eigen, School, Privé, ...
email	string	Email address.
kinderen	array	List of children.
id	Guid	Unique identifier for the child (as relationship). Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50}
naam	string	Child's last name.
voornaam	string	Child's first name.
geslacht	string	Child's sex. Possible values: "M" or "V"
geboortedatum	date	Child's date of birth. Format: yyyy-MM-dd
isOverleden	bool	Indicates whether the child is deceased.
datumOverleden	date	Child's date of death.
fiscaalTenLaste	bool	Indicates whether this child is fiscally dependent on this employee.
fiscaalTenLasteTot	date	Indicates until when this child is fiscally dependent. Note: If this child isn't fiscally dependent, the value will always be null .
personeelsgroepen	array	List of staff groups to which the employee is a member. Note: Only groups for the provided instituteNo will be included in this list. As for the optional structure parameter, only "311" & "312" are taken into account for further filtering.
id	Guid	Unique identifier for the staff group. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50}
naam	string	Name of the staff group.

Example request version 2

Request

```
GET https://<endpoint>/employees?schoolyear=2023-24 HTTP/1.1
InstituteNo: 999999
Api-Version: 2
Authorization: BEARER
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1bm1xdWVfbmFtZSI6Ikluc2m9ybWF0IiwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdCIsImF1ZC16ImF1ZjAwMjd1NzNmOTU0MzVmNzc4MWEwY2ZhZDU5ZTVkIiwiaXZlIjoiJjoxNTc0NTIxODM3LCJ1YmYiOiJlE1NzQzNDkwMzd9.WOXKKj_a1-BaS3syFf-3k-ZT9JGS1_MLdHo4PqnE8eY
Accept: application/json
Content-Type: application/json
```

Example response version 2

Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Access-Control-Allow-Origin: *
Date: Mon, 22 Jan 2024 15:11:42 GMT
[
  {
    "pPersoon": 263,
    "personId": "67e4c91d-0407-4759-a6a7-c427a4784f47",
    "personaId": null,
    "naam": "spropger",
    "voornaam": "Mia Mella",
    "bijkomendeVoornamen": "",
    "stamnr": "25702161637",
    "sortName": "SPROPGER,MIAMELLA,",
    "nickname": "spropger Mia Mella",
    "geslacht": "V",
    "initialen": "STRIN",
    "geboortedatum": "1957-02-16T00:00:00",
    "geboorteplaats": "Westkapelle",
    "geboortelandCode": "00129",
    "nationaliteitCode": "000",
    "rijksregisternr": "57021612345",
    "bisnr": null,
    "bank": {
      "iban": "BE33343536370048",
      "bic": "ARSPBE22"
    },
    "hoofdAmbt": {
      "code": "00000006",
      "naam": "Godsdienstleraar"
    },
    "eersteDienstSchool": "2011-09-01T00:00:00",
    "eersteDienstScholengroep": "1900-01-01T00:00:00",
    "eersteDienstScholengemeenschap": "1900-01-01T00:00:00",
    "isActive": true,
    "effectiefInDienst": true,
    "pensioendatum": null,
    "isMindervalide": false,
    "isOverleden": false,
    "datumOverleden": null,
    "burgStand": 4,
    "aantalKinderen": 2,
    "mindervalideKinderen": 1,
    "aantalZorgbehoevenden": 1,
    "aantalAndereTenLaste": 0,
    "aantalMindervalideAndereTenLaste": 0,
```

```

"partner": {
  "id": "5d70a9c8-1526-42e5-8614-f313b9eb154b",
  "naam": "Merckx",
  "voornaam": "Jozef",
  "geboortedatum": "1958-03-09T00:00:00",
  "nationaliteitCode": "00150",
  "isOverleden": false,
  "datumOverleden": null,
  "inkomen": 1
},
"adressen": [
  {
    "pAdres": 12080,
    "id": "a22b2980-d9e8-4449-8e0d-7ae2f722c78e",
    "aanspreekTitel": "Aan",
    "aanspreekNaam": "spropger Mia Mella",
    "straat": "Lakenmakerssteenweg",
    "nummer": "157",
    "bus": "",
    "postcode": "2800",
    "gemeente": "MECHELEN",
    "landCode": "00150",
    "isDomicilie": true,
    "isVerblijf": true
  }
],
"comnrs": [
  {
    "pComnr": 6825,
    "id": "e2d2fa0c-9d1b-4193-9d50-493b5a6e3be1",
    "type": "Domicilie",
    "soort": "Telefoon",
    "nr": "025 55 52 52"
  }
],
"emailadressen": [
  {
    "pEmail": 9500,
    "id": "b4b4e246-a61a-4499-8f9e-dd0827b19435",
    "type": "Privé",
    "email": "spropger.miamella@informat.be"
  },
  {
    "pEmail": 10263,
    "id": "ad4b6ec9-b9d5-489d-88fb-318a838e56d5",
    "type": "School",
    "email": "andere.spropger.miamella@informat.be"
  }
],
"kinderen": [
  {
    "id": "707da31a-44fd-45ba-904b-80ad0f2052c4",
    "naam": "Merckx ",
    "voornaam": "Wouter ",
    "geslacht": "M",
    "geboortedatum": "1986-12-23T00:00:00",
    "isOverleden": false,
    "datumOverleden": null,
    "fiscaalTenLaste": false,
    "fiscaalTenLasteTot": null
  }
],

```

```
"personeelsgroepen": [  
  {  
    "id": "a31577ff-2dda-426a-b689-1634b4545985",  
    "naam": "Mechelen"  
  },  
  {  
    "id": "53ce3cf5-cdb9-4dc4-b2ba-66b52bb51340",  
    "naam": "leerkrachten 1ste graad"  
  }  
]  
,  
{...}  
]
```


Example response

Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Access-Control-Allow-Origin: *
Date: Mon, 22 Jan 2024 15:11:42 GMT
{
  "pPersoon": 13050,
  "personId": "e7fe8b72-1dd8-4bf8-a90a-0666e040f1ee",
  "personaId": null,
  "naam": "Vondenwiengoirt",
  "voornaam": "Jaime Fernando",
  ...
  SIMULAR TO THE GLOBAL GET EMPLOYEES CALL
  ...
}
```

Assignments

This chapter describes how to:

- Get all assignments for an institute-number, school year and structure;
- Get an employee's assignments by personId.

It describes all the available methods, their request/response format and provides input/output examples.

GET /employees/assignments?

schoolyear={schoolYear}&structure={structure}&inclNonSubHours={inclNonSubHours}

Gets all the assignments for the combination institute number, school year and structure. An additional filter can be applied for including/excluding non- subsidisable hours. The institute number is defined via the mandatory header "InstituteNo" and the school year must be send via the url. The optional structure and inclNonSubHours can also be provided via the URL.

Request

Headers

Besides the mandatory headers, an optional [version number](#) can be provided. This call is also version neutral, which means that using version 1 or 2 will result in the same response.

URL

Field	Type	Required/optional	Description
schoolYear	string	required	Limits the output results to assignments within the given schoolyear (and instituteNo). Format: "2023-24" If not provided a bad request (400) error will be returned.
structure	string	optional	This is an actually an additional restriction on the school year filter. It limits the output results to assignments for a given structure. Note: All structures are taken into account. Format: "311"
inclNonSubHours	bool	optional	Default value is true. So, assignments with non-subsidisable hours will only be excluded when a false is provided.

Body

This request has an empty body.

Response

The response contains a list of assignments.

Field	Type	Description
pOpdracht	int	Unique integer-value for the assignment within the database.
id	Guid	Unique identifier for the assignment. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50}
personId	Guid	Unique identifier for the employee. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50}
begindatum	date	Assignment's begin date.
einddatum	date	Assignment's end date.
hfd	string	Assignment's structure. Format: "211"
vervangingen	array	List of substitutions.
opdrachtId	Guid	Unique identifier of the substitute assignment (child).
dold	Guid	Unique identifier of the linked interruption.
prestatieUren	decimal	Number of hours of the assignment.
prestatieNoemer	integer	Number of hours to perform for a full-time position.
urenTbsOB	decimal	Number of hours that cannot be performed. e.g. due to too few classes.
ato	string	Administrative situation Possible values: 1 = Aanstelling in niet-vacante uren (vervanging) 2 = Aanstelling in vacante uren 3 = Proeftijd 4 = Vast benoemd
afstandWachtgeld	decimal	Number of hours in which no "Reaffectatie/Wedertewerkstelling" is possible.
reaffWedertew	string	Reaffectatie status Possible values: R = Reaffectatie W = Wedertewerkstelling VT = Voorlopige toewijzing NV = Niet vatbaar voor reaffectatie Note: As this is not a mandatory field, the value may be empty ("").
bekwaamheidsbewijs	string	To what extent the employee's diploma is suitable to perform the assignment/task. Possible values: VE = Vereist bekwaamheidsbewijs VO = Voldoende AND = Andere Note: As this is not a mandatory field, the value may be empty ("0").

graad	string	Grade. Only applicable for secondary education & HBO5. Possible values: 1, 2 & 3 for secondary education H for HBO5 0 means not applicable
leerjaar	string	Year. Possible values: 1 through 6 Note: As this is not a mandatory field, the value may be empty ("0").
onderwijsvorm	string	Educational form. Possible values: SO, BSO, KSO, TSO, ASO, HBO Note: As this is not always a mandatory field, the value may be empty ("0"). The value SO is only applicable for the first grade; "BSO, KSO, TSO & ASO" for the 2 nd and 3 rd grade.
opleidingsvorm	string	Student's level of special needs. Only applicable for BUSO. Possible values: 'OV1' through 'OV4', and '-' if not applicable.
ambtCode	string	Official code of the function.
ambt	string	Official description of the function.
vakCode	string	Official code of the subject associated to the function.
vak	string	Official description of the subject.
vakRubriek	string	Official code of the section under which the subject is classified.
gelijkgesteldVakCode	string	Official code of the "equivalent subject" (part within the subject).
gelijkgesteldVak	string	Official description of the "equivalent subject".
gelijkgesteldVakRubriek	string	Official code of the section under which the "equivalent subject" is classified.
isAmbtvak	boolean	Indicates whether there is no subject for the specified function.
aanstellingscode	integer	Type of appointment Possible values: 1 = TADD (tijdelijk aanstelling van doorlopende duur) 4 = LIO (leerkracht in opleiding) 0 = Gewone aanstelling
isTao	boolean	Indicates whether this assignment is a temporarily different assignment (an assignment that deviates from the permanent position).
isVacature	boolean	The assignment is open to permanent appointment.
mandaatCode	string	To indicate that someone is a teacher specialist. Possible values: 00 = Niet van toepassing 01 = Leraar specialist
omkaderingCode	string	Official code for funds allocated to the school.
omkadering	string	Official description for funds allocated to the school.
opmerking	string	Remark.
vestiging	string	Location where the assignment takes place.
baremaCode	string	Official code of the wage scale.
barema	string	Official description of the wage scale.

wvoStudiegebiedCode	string	Official study area code, only applicable to adult education .
wvoStudiegebied	string	Official study area, only applicable to adult education .
wvoOpleidingCode	string	Official training area code, only applicable to adult education .
wvoOpleiding	string	Official training area, only applicable to adult education .
wvoModuleCode	string	Official module code, only applicable to adult education .
wvoModule	string	Official module, only applicable to adult education .
urenVanScholengem	boolean	Indicates whether a number of hours is allocated to the school by the school community.
urenVanAndereSchool	boolean	Indicates whether a number of hours is donated by another school.
opdrachtType	integer	Assignment type (hours) Possible values: 1 = Gewone subsidieerbare uren 2 = Niet-subsidieerbare uren

Example request

Request

```
GET https://<endpoint>/employees/appointments?schoolyear=2023-24&structure=311 HTTP/1.1
InstituteNo: 999999
Api-Version: 1
Authorization: BEARER
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1bm1xdWVfbmFtZSI6Ikluc2m9ybWF0IiwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdCIsImF1ZCI6ImF1ZjAwMjd1NzNmOTU0MzVmNzc4MWEwY2ZhZDU5ZTVkIiwiaXNzIjoxNTc0NTIxODM3LCJ1YmYiOiJ0E1NzQzNDkwMzd9.WOXKKj_a1-BaS3syFf-3k-ZT9JGS1_MLdHo4PqnE8eY
Accept: application/json
Content-Type: application/json
```

Example response

Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Access-Control-Allow-Origin: *
Date: Mon, 26 Aug 2024 15:11:42 GMT
[
```

```

{
  "pOpdracht": 201952,
  "id": "1a65090a-2c68-4759-ac04-82d6dcf3b90d",
  "personId": "17a5d224-e470-4482-8925-5311cc031c89",
  "begindatum": "2023-09-01T00:00:00",
  "einddatum": "2024-08-31T00:00:00",
  "hfd": "311",
  "vervangingen": [
    {
      "opdrachtId": "c3441d1e-3408-4129-ab36-f1dc85555ae0",
      "doId": "fd179d48-ebf3-4e0c-8111-4149bffa7757"
    },
    {
      "opdrachtId": "4379d9af-0998-4111-b186-76eb6acb3e39",
      "doId": "fd179d48-ebf3-4e0c-8111-4149bffa7757"
    }
  ],
  "prestatieUren": 4.000,
  "prestatieNoemer": 20,
  "urenTbsOB": 0.000,
  "ato": "2",
  "afstandWachtgeld": 0.000,
  "reaffWedertew": "",
  "bekwaamheidsbewijs": "AND",
  "graad": "3",
  "leerjaar": "0",
  "onderwijsvorm": "ASO",
  "opleidingsvorm": "-",
  "ambtCode": "00000007",
  "ambt": "Leraar",
  "vakCode": "00000309",
  "vak": "Wetenschappelijk werk",
  "vakRubriek": "AV",
  "gelijkgesteldVakCode": null,
  "gelijkgesteldVak": null,
  "vakRubriek": null,
  "isAmbtvak": false,
  "aanstellingscode": 1,
  "isTao": false,
  "isVacature": true,
  "mandaatCode": "01",
  "omkaderingCode": "00",
  "omkadering": "eigen omkadering (standaard)",
  "opmerking": "",
  "vestiging": null,
  "baremaCode": null,
  "barema": null,
  "vwoStudiegebiedCode": null,
  "vwoStudiegebied": null,
  "vwoOpleidingCode": null,
  "vwoOpleiding": null,
  "vwoModuleCode": null,
  "vwoModule": null,
  "urenVanScholengem": false,
  "urenVanAndereSchool": false,
  "opdrachtType": 1
},
{...}
]

```

GET /employees/{personId}/assignments? schoolyear={schoolYear}&structure={structure}&inclNonSubHours={inclNonSubHours}

Gets an employee's assignments by personId and for the combination institute number, school year and structure. An additional filter can be applied for including/excluding non- subsidisable hours. The institute number is defined via the mandatory header "InstituteNo" and the personId and school year must be send via the url. Optional structure and inclNonSubHours can be provided via the URL.

Request

Headers

Besides the mandatory headers, an optional [version number](#) can be provided. This call is also version neutral, which means that using version 1 or 2 will result in the same response.

URL

Field	Type	Required/optional	Description
personId	Guid	required	Limits the output results to an employee's assignments for the provided personId. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50} If not provided a bad request (400) will be returned.
schoolYear	string	required	Limits the output results to assignments within the given schoolyear (and instituteNo). Format: "2023-24" If not provided a bad request (400) error will be returned.
structure	string	optional	This is an actually an additional restriction on the school year filter. It limits the output results to assignments for a given structure. Note: All structures are taken into account. Format: "311"
inclNonSubHours	bool	optional	Default value is true. So, assignments with non-subsidisable hours will only be excluded when a false is provided.

Body

This request has an empty body.

Response

The response contains assignments for one employee instead of a list of employees. This makes the response similar to the call [GET /employees/assignments?
schoolyear={schoolYear}&structure={structure}&inclNonSubHours={inclNonSubHours}](#).

Example request

Request

```
GET https://<endpoint>/employees/e7fe8b72-1dd8-4bf8-a90a-0666e040f1ee/assignments?schoolyear=2023-24
HTTP/1.1
InstituteNo: 999999
Api-Version: 1
```

```
Authorization: BEARER  
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1bm1xdWVfbmFtZSI6IklmZm9yYm90IiwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdC  
IsImF1ZCI6ImF1ZjAwMjd1NzNmOTU0MzVmNzc4MWEwY2ZhZDU5ZTVkIiwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdCJmZm9zdC  
WOXKKj_a1-BaS3syFf-3k-ZT9JGS1_MLdHo4PqnE8eY  
Accept: application/json  
Content-Type: application/json
```

Example response

The response contains assignments for one employee (personId) instead of a list of employees. This makes the response example similar to [GET /employees/assignments?schoolyear={schoolYear}&structure={structure}](#).

Interruptions

This chapter describes how to:

- Get all interruptions for an institute-number, school year and structure;
- Get an employee's interruptions by personId;
- Add a new interruption or update an existing interruption;
- Remove an interruption.

It describes all the available methods, their request/response format and provides input/output examples.

GET /employees/interruptions?schoolyear={schoolYear}&structure={structure}

Gets all the interruptions for the combination institute number, school year and structure. The institute number is defined via the mandatory header "InstituteNo" and the school year must be send via the url. The optional structure can be provided via the URL.

Request

Headers

Besides the mandatory headers, an optional [version number](#) can be provided. This call is also version neutral, which means that using version 1 or 2 will result in the same response.

URL

Field	Type	Required/optional	Description
schoolYear	string	required	Limits the output results to interruptions within the given schoolyear (and instituteNo). Because an interruption can fall outside the school year boundaries, filtering is done like so: Enddate interruption >= start school year && Begindate interruption <= end of school Year. Format: "2023-24" If not provided a bad request (400) error will be returned.
Structure	string	optional	This is an actually an additional restriction on the school year filter. It limits the output results to interruptions for a given structure. Note: All structures are taken into account. Format: "311"

Body

This request has an empty body.

Response

The response contains a list of interruptions.

Field	Type	Description
pOnderbreking	int	Unique integer-value for the interruption within the database.
dold	Guid	Unique identifier for the interruption. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50}
personId	Guid	Unique identifier for the employee. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50}

hfd	string	Interruption's structure. Format: "211"
begindatum	date	Interruption's begin date.
einddatum	date	Interruption's end date.
code	string	Interruption's code.
omschrijving	string	Interruption's description.
soort	string	Interruption's RL-code. Note: Will be "0" for own defined interruptions.
verstuurdAgODi	bool	Indicates whether the last changes have been sent to AgODi. Note: Will be null for own defined interruptions.
opmSecretariaat	string	Remark entered by a secretarial staff member via Informat.
opmPersoneelslid	string	Remark sent along with the "/employees/{personId}/interruptions/{interruptionId}" call. Note: Not applicable for own defined interruptions.
staking	string	Indicates the time of day, the employee is participating in the strike. Possible values: Dag = the whole day VM = half a day, during the morning NM = half a day, during the afternoon Uren = a certain number of hours
stakingUren	decimal	When "staking" has value "Uren", then this field shows the number of hours for which the strike was participated.
ziekteverlofZonderBezoldiging	bool	Indicates whether the interruption is an unpaid sick leave.

Example request

Request

```
GET https://<endpoint>/employees/interruptions?schoolyear=2023-24&structure=311 HTTP/1.1
InstituteNo: 999999
Api-Version: 1
Authorization: BEARER
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1bmlxdWVfbmFtZSI6IklkluZm9ybWV0IiwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdCIsImF1ZCI6ImFlZjAwMjd1NzNmOTU0MzVmNzc4MWEwY2ZhZDU5ZTVkIiwiaXZlbnIjoixNTc0NTIxODM3LCJuaWYiOiJ0jE1NzQzNDkwMzd9.WOXKkj_a1-BaS3syFf-3k-ZT9JGS1_MLdHo4PqnE8eY
Accept: application/json
Content-Type: application/json
```

Example response

Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Access-Control-Allow-Origin: *
Date: Mon, 26 Aug 2024 15:11:42 GMT
[
  {
    "pOnderbreking": 85684,
    "doId": "10271100-ea9e-4227-82ed-2a3370962a8b",
    "personId": "ba587efe-25a2-40f3-8c62-3cf8a2599124",
    "hfd": "311",
    "begindatum": "2023-10-27T00:00:00",
    "einddatum": "2023-10-27T00:00:00",
    "code": "1",
    "omschrijving": "Ziekteverlof",
    "soort": "2",
    "verstuurdAgODi": false,
    "opmSecretariaat": "Voelde zich griepig",
    "opmPersoneelslid": null,
    "staking": null,
    "stakingUren": null,
    "ziekteverlofZonderBezoldiging": false
  },
  {
    "pOnderbreking": 86104,
    "doId": "3fd9f2d4-2797-45a6-a433-a5e43f9dd04c",
    "personId": "1c600b02-10ad-474b-8166-f5aa2c962254",
    "hfd": "311",
    "begindatum": "2023-11-15T00:00:00",
    "einddatum": "2023-11-16T00:00:00",
    "code": "144",
    "omschrijving": "Verlof wegens overmacht",
    "soort": "2",
    "verstuurdAgODi": false,
    "opmSecretariaat": "Staking openbaar vervoer",
    "opmPersoneelslid": null,
    "staking": null,
    "stakingUren": null,
    "ziekteverlofZonderBezoldiging": false
  },
  {
    "pOnderbreking": 86214,
    "doId": "30ae92eb-a4a2-4f3b-898b-6857fe9d34ad",
    "personId": "bc334b17-58af-48cc-9fd5-1344acf33ab5",
    "hfd": "311",
    "begindatum": "2024-06-06T00:00:00",
    "einddatum": "2024-06-06T00:00:00",
    "code": "45",
    "omschrijving": "Staking",
    "soort": "2",
    "verstuurdAgODi": false,
    "opmSecretariaat": "",
    "opmPersoneelslid": null,
    "staking": "Uren",
    "stakingUren": 2.00,
    "ziekteverlofZonderBezoldiging": false
  },
  {...}
]
```


Example response

Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Access-Control-Allow-Origin: *
Date: Mon, 26 Aug 2024 15:11:42 GMT
[
  {
    "pOnderbreking": 86211,
    "doId": "93520f0e-de7c-4da3-b35d-937fa635ec58",
    "personId": "bc334b17-58af-48cc-9fd5-1344acf33ab5",
    "hfd": "311",
    "begindatum": "2024-01-24T00:00:00",
    "einddatum": "2024-01-24T00:00:00",
    "code": "002",
    "omschrijving": "Te laat toegekomen",
    "soort": "0",
    "verstuurdAgODi": null,
    "opmSecretariaat": "Lekke autoband",
    "opmPersoneelslid": null,
    "staking": null,
    "stakingUren": null,
    "ziekteverlofZonderBezoldiging": false
  },
  {
    "pOnderbreking": 86212,
    "doId": "3fd9f2d4-2797-45a6-a433-a5e43f9dd04c",
    "personId": "bc334b17-58af-48cc-9fd5-1344acf33ab5",
    "hfd": "311",
    "begindatum": "2024-03-06T00:00:00",
    "einddatum": "2024-04-04T00:00:00",
    "code": "19",
    "omschrijving": "Verlof tijdelijk ander ambt/opdracht",
    "soort": "1",
    "verstuurdAgODi": false,
    "opmSecretariaat": "",
    "opmPersoneelslid": null,
    "staking": null,
    "stakingUren": null,
    "ziekteverlofZonderBezoldiging": false
  },
  {
    "pOnderbreking": 86214,
    "doId": "30ae92eb-a4a2-4f3b-898b-6857fe9d34ad",
    "personId": "bc334b17-58af-48cc-9fd5-1344acf33ab5",
    "hfd": "311",
    "begindatum": "2024-06-06T00:00:00",
    "einddatum": "2024-06-06T00:00:00",
    "code": "45",
    "omschrijving": "Staking",
    "soort": "2",
    "verstuurdAgODi": false,
    "opmSecretariaat": "",
    "opmPersoneelslid": null,
    "staking": "Uren",
    "stakingUren": 2.00,
    "ziekteverlofZonderBezoldiging": false
  },
  {...}
]
```

PUT /employees/{personId}/interruptions/{interruptionId}

This call can be used to:

- Add a new interruption into Informat for an employee (personId);
- Update an existing interruption based on the given interruptionId.

When adding or updating, a MNA001-MNA006 (Method Not Allowed) error may be thrown. An overview can be found over [here](#).

Request

Headers

Besides the mandatory headers, an optional [version number](#) can be provided. This call is also version neutral, which means that using version 1 or 2 will result in the same behavior.

URL

Field	Type	Required/optional	Description
personId	Guid	required	The mandatory unique identifier of an existing employee, to which the interruption will be linked. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50} If not provided in the correct format a Bad Request (400) will be returned. If not provided at all a Not Found (404) error will be returned.
interruptionId	Guid	required	A mandatory unique identifier which is used to identify the interruption. Based on this Id the decision will be made to add a new interruption or update an existing one. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50}

Body

Field	Type	Required/optional	Description
instelnr	string	required	Official institute number. Validation: required (BR003A)
schooljaar	string	required	The school year in which the employee's interruption takes place. Format: "yyyy-yy" Example: "2023-24" Validation: required (BR004A) and correct format (BR004B)
hfd	string	required	The school structure (111, 211,...) of the interruption. Validation: required (BR005A) and must have a length of 3 characters (BR005B)

code	string	required	<p>Official interruption code.</p> <p>Example: "1"</p> <p>Possible values at this moment:</p> <p>"1" = Ziekteverlof "45" = Staking "90" = Omstandigheidsverlof "144" = Verlof wegens overmacht "197" = Voltijds zorgkrediet kinderen <= 12 jaar "200" = Voltijds zorgkrediet medische bijstand "203" = Voltijds zorgkrediet palliatieve zorg "206" = Voltijds zorgkrediet kind met handicap "209" = Voltijds zorgkrediet opleiding</p> <p>Validation: required (BR006A) and valid value (BR006B)</p>
begindatum	date	required	<p>Interruption's start date.</p> <p>Format: valid UTC Date or yyyy-MM-dd</p> <p>Validation: required (BR007A)</p>
einddatum	date	required	<p>Interruption's end date.</p> <p>Format: valid UTC Date or yyyy-MM-dd</p> <p>Validation: required (BR008A) and cannot be before Begindatum (BR008B)</p>
opmerking	string	optional	<p>Remark given by the employee. Will be stored in a separate database field different from the one entered by the secretariat.</p> <p>See the GET interruption Api-calls as an illustration.</p> <p>Validation: max. length 500 (BR009)</p>
naamGeneesheer	string	optional	<p>Name of the doctor</p> <p>Validation: max. length 75 (BR010)</p>
telefoonGeneesheer	string	optional	<p>Doctor's Phone number.</p> <p>Validation: max. length 20 (BR011)</p>
magWoonstVerlaten	boolean	optional	<p>Indicates whether the employee may leave his/her home.</p> <p>Note: If not provided, value true will be stored as a default.</p>
afzender	string	required	<p>Name of the sender. Can be the name of an application, user, ...</p> <p>Validation: required (BR012A) and max. length 50 (BR012B)</p>
geverifieerd	boolean	optional	<p>Indicates whether the interruption is verified by the secretariat.</p> <p>Note: If not provided, value false will be stored as a default.</p>
ziekteverlofZonderBezoldiging	boolean	optional	<p>Indicates whether the interruption is an unpaid sick leave.</p> <p>Note: If not provided, value false will be stored as a default.</p>

dagDeel	string	optional	<p>Time of the day the interruption takes place.</p> <p>Possible values:</p> <p>"DAG" = the whole day "VM" = half a day, during the morning "NM" = half a day, during the afternoon "UREN" = a certain number of hours</p> <p>'Time of the day' does not apply to 'sick leave' but is mandatory for all the other valid interruption types.</p> <p>All possible values are valid for 'Staking', only the first 3 (DAG, VM & NM) are valid for 'Omstandigheidsverlof' & 'Verlof wegens overmacht'.</p> <p>Validation: required (BR029A) and valid value for (interruption) code (BR029B)</p> <p>Note: If provided when not applicable (i.e. sick leave), then the value will not be stored.</p>
urenStaking	decimal	optional	<p>The number of hours the employee participated in the strike.</p> <p>Mandatory with interruption 'Staking' in combination with dagDeel 'UREN'.</p> <p>Validation: required (BR030)</p> <p>Note: If provided when not applicable, then the value will not be stored.</p>

Additional validation checks are:

- BR013: Invalid combination for Institute no, School year and Structure
- BR014: No employee found for this PersonId

See the section "[Bad Request errors](#)" for an overview of all BR-codes.

Response

- A successful add/insert process results in a response status code 200 (Ok) with empty body.
- When there are validation errors, then a status code 400 (Bad Request) is returned, with a list of errors in the body.
- When a method not allowed error occurs, a status code 405 (Method Not Allowed) is returned, with again a list of errors in the body.

Example request

Request

```
PUT https://<endpoint>/employees/bc334b17-58af-48cc-9fd5-1344acf33ab5/interruptions/c5f21445-1180-42ef-832e-62a8d53044c2 HTTP/1.1
InstituteNo: 999999
Api-Version: 1
Authorization: BEARER
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1bm1xdWVfbmFtZSI6IklkluZm9ybWV0IiwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdCIsImF1ZCI6ImF1ZjAwMjd1NzNmOTU0MzVmNzc4MWEwY2ZhZDU5ZTVkIiwiaXZlbnVlIjoiaXNTc0NTIiwiaXN0IjoiODM3LCJ1YmYiOiJ0jE1NzQzNDkwMzd9.WOXKkj_a1-BaS3syFf-3k-ZT9JGS1_MLdHo4PqnE8eY
Accept: application/json
Content-Type: application/json
```


Response with MNA error:

```
HTTP/1.1 405 Method Not Allowed
Content-Type: application/json; charset=utf-8
Access-Control-Allow-Origin: *
Date: Mon, 26 Aug 2024 15:11:42 GMT
{
  "errors": [
    {
      "Code": "MNA001",
      "Message": "This interruption is already linked to another school (institute no, school
year & structure)"
    }
  ]
}
```


Example response

Response with validation error(s):

```
HTTP/1.1 400 Bad Request
Content-Type: application/json; charset=utf-8
Access-Control-Allow-Origin: *
Date: Wed, 18 Sept 2024 11:48:04 GMT
{
  "errors": [
    {
      "Code": "BR018",
      "Message": "No Interruption found for this Id"
    }
  ]
}
```

Response with MNA error:

```
HTTP/1.1 405 Method Not Allowed
Content-Type: application/json; charset=utf-8
Access-Control-Allow-Origin: *
Date: Wed, 18 Sept 2024 11:49:42 GMT
{
  "errors": [
    {
      "Code": "MNA009",
      "Message": "This interruption is already verified by the secretariat and can no longer be
removed"
    }
  ]
}
```

Diplomas

This chapter describes how to:

- Get all diplomas for an institute-number, school year and structure;
- Get an employee's diplomas by personId.

It describes all the available methods, their request/response format and provides input/output examples.

GET /employees/diplomas?schoolyear={schoolYear}&structure={structure}

Gets all the diplomas for the combination institute number, school year and structure. The institute number is defined via the mandatory header "InstituteNo" and the school year must be send via the url. The optional structure can be provided via the URL.

Request

Headers

Besides the mandatory headers, an optional [version number](#) can be provided. This call is also version neutral, which means that using version 1 or 2 will result in the same response.

URL

Field	Type	Required/optional	Description
schoolYear	string	required	For current & future school years: It limits the output results to diplomas for employees with an assignment within the given schoolyear (and instituteNo) or which are marked as active for your instituteNo. For passed school years: It limits the output results to diplomas for employees with an assignment within the given schoolyear (and instituteNo). Format: "2023-24" If not provided a bad request (400) error will be returned.
structure	string	optional	This is an actually an additional restriction on the school year filter. Note: Only structures "311" & "312" are taken into account. Format: "311"

Body

This request has an empty body.

Response

The response contains a list of diploma's.

GET /employees/{personId}/diplomas?schoolYear={schoolYear}&structure={structure}

Gets an employee's diplomas by personId and for the combination institute number, school year and structure. The institute number is defined via the mandatory header "InstituteNo" and the personId and school year must be send via the url. The optional structure can be provided via the URL.

Request

Headers

Besides the mandatory headers, an optional [version number](#) can be provided. This call is also version neutral, which means that using version 1 or 2 will result in the same response.

URL

Field	Type	Required/optional	Description
personId	Guid	required	Limits the output results to an employee's diplomas by the provided personId. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50} If not provided a bad request (400) will be returned.
schoolYear	string	required	For current & future school years: It limits the output results to diplomas for employees with an assignment within the given schoolyear (and instituteNo) or which are marked as active for your instituteNo. For passed school years: It limits the output results to diplomas for employees with an assignment within the given schoolyear (and instituteNo). Format: "2023-24" If not provided a bad request (400) error will be returned.
structure	string	optional	This is an actually an additional restriction on the school year filter. Note: Only structures "311" & "312" are taken into account. Format: "311"

Body

This request has an empty body.

Appointments

This chapter describes how to:

- Get all appointments for an institute-number, school year and structure;
- Get an employee's appointments by personId.

It describes all the available methods, their request/response format and provides input/output examples.

GET /employees/appointments?schoolyear={schoolYear}&structure={structure}

Gets all the appointments for the combination institute number, school year and structure. The institute number is defined via the mandatory header "InstituteNo" and the school year must be send via the url. The optional structure can be provided via the URL.

Request

Headers

Besides the mandatory headers, an optional [version number](#) can be provided. This call is also version neutral, which means that using version 1 or 2 will result in the same response.

URL

Field	Type	Required/optional	Description
schoolYear	string	required	For current & future school years: It limits the output results to appointments for employees with an assignment within the given schoolyear (and instituteNo) or which are marked as active for your instituteNo. For passed school years: It limits the output results to appointments for employees with an assignment within the given schoolyear (and instituteNo). Format: "2023-24" If not provided a bad request (400) error will be returned.
structure	string	optional	This is a two step filter: - An additional restriction on the school year filter for employees Note: Only structures "311" & "312" are taken into account. - A filter on the appointments itself. So, only appointments for this provided structure will be returned. Note: All structures are taken into account. Format: "311"

Body

This request has an empty body.

GET /employees/{personId}/appointments?structure={structure}

Gets an employee's appointments by personId and for the combination institute number and structure. The institute number is defined via the mandatory header "InstituteNo" and the personId must be send via the url. The optional structure can be provided via the URL.

Request

Headers

Besides the mandatory headers, an optional [version number](#) can be provided. This call is also version neutral, which means that using version 1 or 2 will result in the same response.

URL

Field	Type	Required/optional	Description
personId	Guid	required	Limits the output results to an employee's appointments for the provided personId. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50} If not provided a bad request (400) will be returned.
structure	string	optional	Limits the output results to an employee's appointments based on the provided structure. Note: All structures are taken into account. Format: "311"

Body

This request has an empty body.

Photos

This chapter describes how to:

- Get all photos for an institute-number, school year and structure;
- Get an employee's photo by personId.

It describes all the available methods, their request/response format and provides input/output examples.

GET /employees/photos?schoolYear={schoolYear}&structure={structure}&changedSince={changedSince}

Gets all the photos for the combination institute number, school year and structure. The institute number is defined via the mandatory header "InstituteNo" and the school year must be send via the url. The optional structure can also be provided via the URL.

Request

Headers

Besides the mandatory headers, an optional [version number](#) can be provided. This call is also version neutral, which means that using version 1 or 2 will result in the same response.

URL

Field	Type	Required/optional	Description
schoolYear	string	required	For current & future school years: It limits the output results to photos for employees with an assignment within the given schoolyear (and instituteNo) or which are marked as active for your instituteNo. For passed school years: It limits the output results to photos for employees with an assignment within the given schoolyear (and instituteNo). Format: "2023-24" If not provided a bad request (400) error will be returned.
structure	string	optional	This is an actually an additional restriction on the school year filter. Note: Only structures "311" & "312" are taken into account. Format: "311"
changedSince	date	optional	Limits the output to photos that have changed since the specified date.

Body

This request has an empty body.

GET /employees/{personId}/photos

Gets an employee's photo by personId. The institute number is defined via the mandatory header "InstituteNo".

Request

Headers

Besides the mandatory headers, an optional [version number](#) can be provided. This call is also version neutral, which means that using version 1 or 2 will result in the same response.

URL

Field	Type	Required/optional	Description
personId	Guid	required	Limits the output results to an employee's photo for the provided personId. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50} If not provided a bad request (400) will be returned.

Body

This request has an empty body.

Own fields

This chapter describes how to:

- Get all own fields for an institute-number, school year and structure;
- Get an employee's own fields by personId.

It describes all the available methods, their request/response format and provides input/output examples.

GET /employees/ownfields?schoolyear={schoolYear}&structure={structure}

Gets all the own fields for the combination institute number, school year and structure. The institute number is defined via the mandatory header "InstituteNo" and the school year must be send via the url. The optional structure can also be provided via the URL.

Request

Headers

Besides the mandatory headers, an optional [version number](#) can be provided. This call is also version neutral, which means that using version 1 or 2 will result in the same response.

URL

Field	Type	Required/optional	Description
schoolYear	string	required	<p>This is a two step filter:</p> <ul style="list-style-type: none">- For current & future school years: It limits the output results to own fields for employees with an assignment within the given schoolyear (and instituteNo) or which are marked as active for your instituteNo.or for passed school years: It limits the output results to own fields for employees with an assignment within the given schoolyear (and instituteNo).- A filter on the own fields itself. So, only own fields that are linked to a school with the provided instituteNo and school year are returned. <p>Format: "2023-24"</p> <p>If not provided a bad request (400) error will be returned.</p>
structure	string	optional	<p>This is a two step filter:</p> <ul style="list-style-type: none">- An additional restriction on the school year filter for employees Note: Only structures "311" & "312" are taken into account.- A additional restriction on the school year filter for the own fields itself. So, only own fields that are linked to a school with the provided structure are returned. Note: All structures are taken into account. <p>Format: "311"</p>

Body

This request has an empty body.

GET /employees/{personId}/ownfields?schoolyear={schoolYear}&structure={structure}

Gets an employee’s own fields by personId and for the combination institute number and structure. The institute number is defined via the mandatory header “InstituteNo” and the school year must be send via the url. The optional structure can also be provided via the URL.

Request

Headers

Besides the mandatory headers, an optional [version number](#) can be provided. This call is also version neutral, which means that using version 1 or 2 will result in the same response.

URL

Field	Type	Required/optional	Description
personId	Guid	required	Limits the output results to an employee’s own fields by the provided personId. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50} If not provided a bad request (400) will be returned.
schoolYear	string	required	Limits the output results to an employee’s own fields, linked to a school with the provided instituteNo and school year. Format: “2023-24” If not provided a bad request (400) error will be returned.
structure	string	optional	Is actually an additional restriction on the school year filter. So, only own fields that are linked to a school with the provided structure are returned. Note: All structures are taken into account. Format: “311”

Body

This request has an empty body.

Response

The response contains a list of own fields for a particular personId. The response similar to the call [“GET /employees/ownfields?schoolyear={schoolYear}&structure={structure}”](#).

Example request

Request

```
GET https://<endpoint>/employees/bc334b17-58af-48cc-9fd5-1344acf33ab5/ownfields?schoolyear=2023-24&structure=211 HTTP/1.1
InstituteNo: 999999
Api-Version: 1
Authorization: BEARER
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1bm1xdWVfbmFtZSI6Ikluc2m9ybWFW0IiwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdCIsImF1ZCI6ImF1ZjAwMjdlNzNmOTU0MzVmNzc4MWEwY2ZkZDU5ZTVkIiwiaXNzIjoiwNTc0NTIxdm3LCJuaWYiOiJ0e1NzQzNDkwMzd9.W0XKkj_a1-BaS3syFf-3k-ZT9JGS1_MLdHo4PqnE8eY
Accept: application/json
Content-Type: application/json
```

Example response

Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Access-Control-Allow-Origin: *
Date: Mon, 05 Feb 2024 16:52:42 GMT
[
  {
    "personId": "bc334b17-58af-48cc-9fd5-1344acf33ab5",
    "vvId": "91eb3158-10a9-41a6-ab7c-27ce0853c5b2",
    "naam": "kastnummer",
    "waarde": "16",
    "dataType": "Getal",
    "rubriek": 1
  },
  {
    "personId": "bc334b17-58af-48cc-9fd5-1344acf33ab5",
    "vvId": "b69e572e-9ab3-48c2-a0a8-36d22f87d6a8",
    "naam": "nummerplaat",
    "waarde": "1-ABC-001",
    "dataType": "Tekst",
    "rubriek": 1
  },
  {...}
]
```

Attachments

This chapter describes how to:

- Add a new interruption-attachment or update an existing one;
- Remove an interruption-attachment.

It describes all the available methods, their request/response format and provides input/output examples.

PUT /employees/interruptions/{interruptionId}/attachments/{attachmentId}

This call can be used to:

- Add a new attachment into Informat linked to an interruption (interruptionId);
- Update an existing interruption-attachment based on the given attachmentId. An attachment can be linked to multiple interruptions of the same employee.

When adding or updating, a MNA007 (Method Not Allowed) error may be thrown (status code 405). An overview can be found over [here](#).

Request

Headers

Besides the mandatory headers, an optional [version number](#) can be provided. This call is also version neutral, which means that using version 1 or 2 will result in the same behavior.

URL

Field	Type	Required/optional	Description
interruptionId	Guid	required	The mandatory unique identifier of an existing interruption, to which the attachment will be linked. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50} If not provided in the correct format a Bad Request (400) will be returned. If not provided at all a Not Found (404) error will be returned. Validation: required (BR002)
attachmentId	Guid	required	A mandatory unique identifier which is used to identify the attachment. Based on this Id the decision will be made to add a new attachment or update an existing one. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50} Validation: required (BR015)

Additional validation checks are:

- BR018: No interruption found for this Id

See the section "[Bad Request errors](#)" for an overview of all BR-codes.

Example response

Response with validation error(s):

```
HTTP/1.1 400 Bad Request
Content-Type: application/json; charset=utf-8
Access-Control-Allow-Origin: *
Date: Wed, 11 Sept 2024 10:18:49 GMT
{
  "errors": [
    {
      "Code": "BR016C",
      "Message": "Invalid file extension"
    }
  ]
}
```

Response with MNA error:

```
HTTP/1.1 405 Method Not Allowed
Content-Type: application/json; charset=utf-8
Access-Control-Allow-Origin: *
Date: Wed, 11 Sept 2024 10:28:42 GMT
{
  "errors": [
    {
      "Code": "MNA007",
      "Message": "Attachment already linked to another employee"
    }
  ]
}
```

DELETE /employees/interruptions/{interruptionId}/attachments/{attachmentId}

This call can be used to:

- Remove the link between an interruption and an attachment. If no other interruption is linked to this attachment then the attachment itself will also be removed.

Request

Headers

Besides the mandatory headers, an optional [version number](#) can be provided. This call is also version neutral, which means that using version 1 or 2 will result in the same behavior.

URL

Field	Type	Required/optional	Description
interruptionId	Guid	required	The mandatory unique identifier of an existing interruption, to which the attachment is linked. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50} If not provided in the correct format a Bad Request (400) will be returned. If not provided at all a Not Found (404) error will be returned. Validation: required (BR002)
attachmentId	Guid	required	A mandatory unique identifier of an existing attachment. Format: {42CD5A20-E22F-45D7-973C-FAB8734DEC50} Validation: required (BR015)

Additional validation checks are:

- BR019: Nothing found to delete for this interruptionId & attachmentId combination

See the section "[Bad Request errors](#)" for an overview of all BR-codes.

Body

This request has an empty body.

Response

- A successful delete process results in a response status code 200 (Ok) with empty body.
- When there are validation errors, then a status code 400 (Bad Request) is returned, with a list of errors in the body.

Example request

Request

```
DELETE https://<endpoint>/employees/interruptions/c5f21445-1180-42ef-832e-62a8d53044c2/attachments/{
54fea255-24d2-4a91-80db-7325d5dd3d5e} HTTP/1.1
InstituteNo: 999999
Api-Version: 1
Authorization: BEARER
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1bm1xdWVfbmFtZSI6Ikluc29ybnVfWF0iwiiaXNzIjoiaHR0cDovL2xvY2FsaG9zdC
IsImF1ZCI6ImF1ZjAwMjd1NzNmOTU0MzVmNmZc4MWEWY2ZhZDU5ZTVkIiwiaXNzIjoiwNTc0NTIwODM3LCJ1YmYiOiJlE1NzQzNDkwMzd9.
W0XKkj_a1-BaS3syFf-3k-ZT9JGS1_MLdHo4PqnE8eY
Accept: application/json
Content-Type: application/json
```

Example response

Response with validation error(s):

```
HTTP/1.1 400 Bad Request
Content-Type: application/json; charset=utf-8
Access-Control-Allow-Origin: *
Date: Wed, 11 Sept 2024 10:18:49 GMT
{
  "errors": [
    {
      "Code": "BR019",
      "Message": "Nothing found to delete for this interruptionId & attachmentId combination"
    }
  ]
}
```

Communication numbers

This chapter describes how to:

- Add a new communication number or update an existing one;
- Remove a communication number.

It describes all the available methods, their request/response format and provides input/output examples.

PUT /employees/{personId}/communication-number

This call can be used to:

- Add a new communication number into Informat, linked to an employee;
- Update an existing communication number.

Request

Headers

Besides the mandatory headers, an optional [version number](#) can be provided. This call is also version neutral, which means that using version 1 or 2 will result in the same behavior.

URL

Field	Type	Required/optional	Description
personId	Guid	required	The mandatory unique identifier of an existing employee, to which the communication number is or will be linked. Format: {c5f21445-1180-42ef-832e-62a8d53044c2} If not provided in the correct format a Bad Request (400) will be returned. If not provided at all a Not Found (404) error will be returned. Validation: required (BR001)

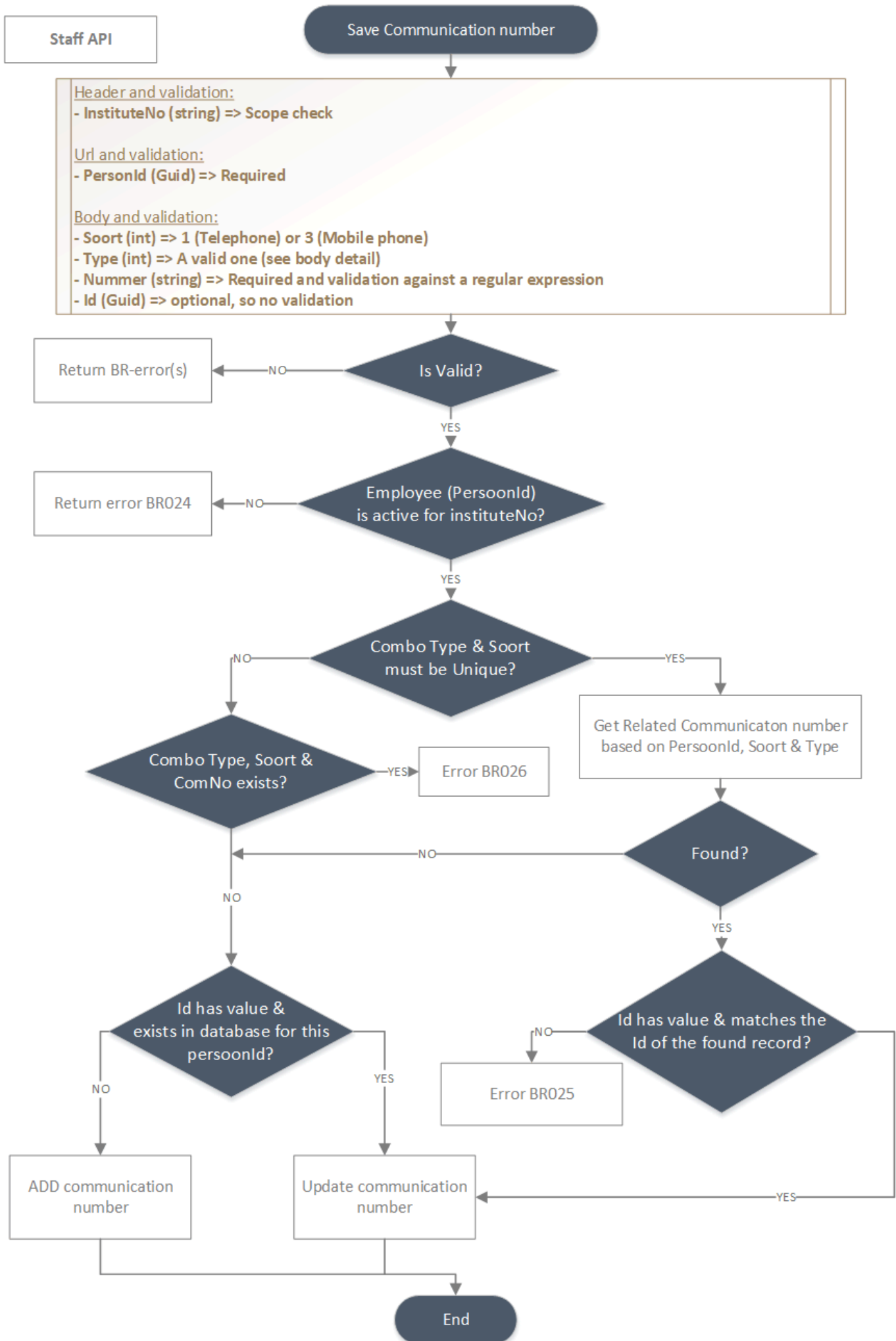
Body

Field	Type	Required/optional	Description
soort	int	required	Kind of Communication number. Possible values: 1 = Telephone 3 = Mobile phone Validation: must be one of the possible values (BR020)
type	int	required	Communication number type. Possible values: 1 = Domicile 2 = Residence 7 = Alternative Number 8 = Private Number 9 = Emergency Number 18 = Work 30 = Partner 46 = Neighbors Validation: type is invalid (BR021)


```
{
  "errors": [
    {
      "Code": "BR026",
      "Message": " A communication number already exists for this combination 'Soort, Type &
Nummer'; Id: 85c719d6-60eb-474a-a67d-059bba169249"
    }
  ]
}
```

Flow chart

See next page.



DELETE /employees/{personId}/communication-number/{communicationNold}

This call can be used to:

- Remove a communication number linked to an employee by communication number Id.

Request

Headers

Besides the mandatory headers, an optional [version number](#) can be provided. This call is also version neutral, which means that using version 1 or 2 will result in the same behavior.

URL

Field	Type	Required/optional	Description
personId	Guid	required	The mandatory unique identifier of an existing employee, to which the communication number is or will be linked. Format: {c5f21445-1180-42ef-832e-62a8d53044c2} If not provided in the correct format a Bad Request (400) will be returned. If not provided at all a Not Found (404) error will be returned. Validation: required (BR001)
comNold	Guid	required	A mandatory unique identifier of an existing communication number. Format: {a8050737-6004-459a-bc35-6028e15bd0a3} If not provided in the correct format a Bad Request (400) will be returned. If not provided at all a Not Found (404) error will be returned. Validation: required (BR027)

Body

This request has an empty body.

Response

- A successful delete process results in a response status code 200 (Ok) with empty body.
- When there are validation errors, then a status code 400 (Bad Request) is returned, with a list of errors in the body.

Example request

Request

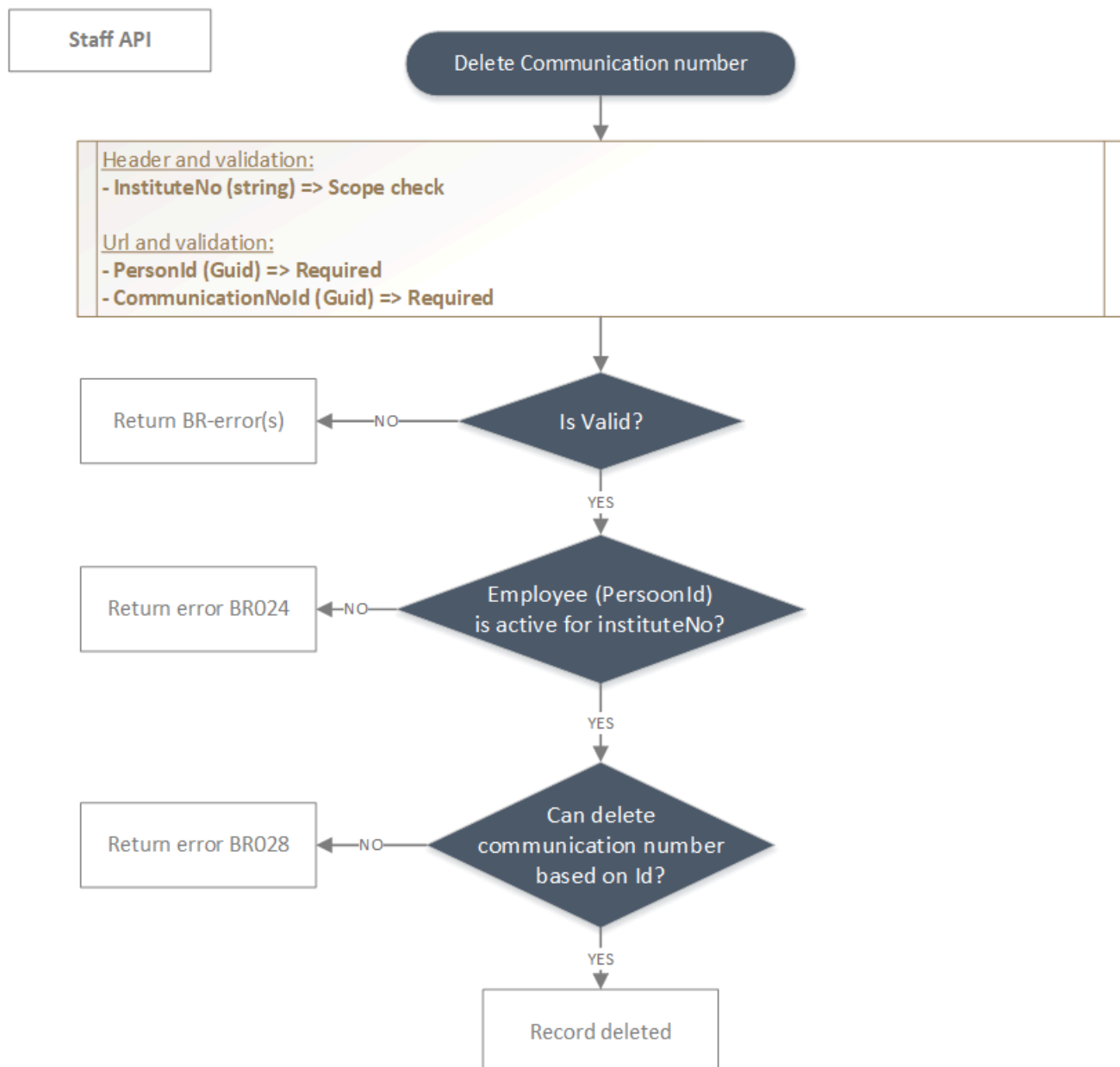
```
DELETE https://<endpoint>/employees/c5f21445-1180-42ef-832e-62a8d53044c2/communication-number/{
a8050737-6004-459a-bc35-6028e15bd0a3}
HTTP/1.1
InstituteNo: 999999
Api-Version: 1
Authorization: BEARER
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1bm1xdWVfbmFtZSI6IklkluZm9ybWV0IiwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdC
IsImF1ZC16ImFlZjAwMjd1NzNmOTU0MzVmNzc4MWEwY2ZlZDU5ZTVkIiwiaXZlbnV0IjoxNTc0NTIxODM3LCJyYmYiOiJlE1NzQzNDkwMzd9.
W0XKkj_a1-BaS3syFf-3k-ZT9JGS1_MLdHo4PqnE8eY
Accept: application/json
Content-Type: application/json
```

Example response

Response with validation error(s):

```
HTTP/1.1 400 Bad Request
Content-Type: application/json; charset=utf-8
Access-Control-Allow-Origin: *
{
  "errors": [
    {
      "Code": "BR028",
      "Message": "No Communication Number found for this Id"
    }
  ]
}
```

Flow chart



Documents

This section describes the calls related to RL documents and their digital signing. RL documents are official messages submitted by a school to the Flemish government (AGODI) through the Edison channel. Two types are supported:

RL1 – Opdrachtenpakket (work-package description for an employee).

RL2 – Notification, supplement or cancellation of a service interruption.

The typical integration flow is:

Fetch the list of available RL documents for a school year.

Store the highest “edisonStatusGewijzigdOp” value from the response as a watermark.

On subsequent calls, pass that watermark via “referencedate” to retrieve only documents whose Edison status has changed since that timestamp.

Download the PDF for each document via the dedicated PDF endpoint.

Present the PDF in your own signing environment (employee signs, refuses, or a technical failure occurs).

Submit the outcome (and, when signed, the signed PDF) back to Informat.

Important notes:

“referencedate” accepts ISO 8601 only (e.g. 2026-01-30 or 2026-01-30T09:15:00Z).

“rl1Ingangsdatum” is only present in the response for RL1 documents; for RL2 it is omitted.

“edisonStatusGewijzigdOp” may be null for older rows that pre-date this column. Such rows are NOT returned by calls that include a “referencedate” filter – for a complete initial sync, omit “referencedate”.

A document can be submitted as “Ondertekend” only once. A second attempt returns HTTP 409 (BR_DOC_006).

The PDF endpoint requires only the Bearer token – no temporary download URLs are needed. If the document has already been signed (“tekenStatus = Ondertekend”), the endpoint returns the signed PDF.

GET /employees/documents?schoolyear={schoolYear}

Returns the list of RL documents available for signing for all employees of the given institute.

Request

Headers

Besides the mandatory headers, an optional version number can be provided. This call is also versioned, although currently only version 1 exists.

URL

Field	Type	Mandatory	Description
schoolYear	string	Yes	School year in YYYY-YY format (e.g. 2025-26).
rlType	string[]	No	Filter by RL type. Allowed values: RL1, RL2. May be repeated to filter on multiple types (e.g. ?rlType=RL1&rlType=RL2). If omitted, both RL1 and

			RL2 are returned. An unsupported value returns HTTP 400 (BR_DOC_007).
referencedate	string (ISO 8601)	No	Watermark for incremental sync. Returns only documents whose edisonStatusGewijzigdOp is greater than or equal to this value. Rows where edisonStatusGewijzigdOp is null are not returned. Omit for a complete initial sync.

Body

n/a

Response

A successful call returns status code 200 (Ok) with an array of RLDocumentDto objects. An empty array is returned when no documents are available.

See the section "Bad Request errors" for an overview of all BR-codes.

Field	Type	Description
documentId	int32	Unique id of the document. Use this value as {documentId} in the PDF and SignedDocument endpoints.
personId	guid	Identifier of the employee this document belongs to.
rlType	string	RL1 or RL2.
tekenStatus	string (nullable)	Current signing status. Values: null (not yet downloaded), Opgehaald, Bekeken, Ondertekend, Geweigerd, Mislukt.
zendingnr	int32 (nullable)	Edison message id (from EdisonSend.MessageId).
referentie	string (nullable)	External shipment reference (from EdisonSend.Reference).
aangemaaktOp	datetime	When the document was created in Informat (UTC).
verstuurdNaarEdisonOp	datetime (nullable)	When the document was sent to Edison (UTC). null while not yet sent.
edisonStatusGewijzigdOp	datetime (nullable)	When the Edison status of this shipment was last changed. Use max(edisonStatusGewijzigdOp) from a response as the value of referencedate in the next call.
rl1Ingangsdatum	datetime (nullable)	RL1 only. The RL1 effective date. Omitted from the response for RL2 documents.

Example request

Request

```
GET https://<endpoint>/employees/documents?schoolyear=2025-26&r1Type=RL1&r1Type=RL2&referencedate=2026-01-30T00:00:00Z HTTP/1.1
InstituteNo: 999999
Api-Version: 1
Authorization: BEARER eyJ0eXAI0iJKV1QilCJhbGciOiJIUzI1NiJ9...
Accept: application/json
```

Example response

Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
[
  {
    "documentId": 42,
    "personId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "r1Type": "RL1",
    "tekenStatus": "Ondertekend",
    "zendingnr": 12300,
    "referentie": "REF-2025-08-007",
    "aangemaaktOp": "2026-01-25T08:30:00Z",
    "verstuurdNaarEdisonOp": "2026-01-26T14:20:00Z",
    "edisonStatusGewijzigdOp": "2026-01-28T10:00:00Z",
    "r1Ingangsdatum": "2025-09-01T00:00:00Z"
  },
  {
    "documentId": 43,
    "personId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "r1Type": "RL2",
    "tekenStatus": null,
    "zendingnr": 12345,
    "referentie": "REF-2025-09-001",
    "aangemaaktOp": "2026-01-30T09:15:00Z",
    "verstuurdNaarEdisonOp": null,
    "edisonStatusGewijzigdOp": "2026-01-30T09:15:00Z"
  }
]
```

GET /employees/{personId}/documents?schoolyear={schoolYear}

Same as the institute-wide call above, but scoped to a single employee. The list of available RL documents is returned for the given personId.

Request

Headers

Besides the mandatory headers, an optional version number can be provided. This call is also versioned, although currently only version 1 exists.

URL

Field	Type	Mandatory	Description
personId	guid	Yes	Unique identifier of the employee.
schoolYear	string	Yes	School year in YYYY-YY format (e.g. 2025-26).

rIType	string[]	No	Filter by RL type. Allowed values: RL1, RL2. May be repeated. If omitted, both types are returned.
referencedate	string (ISO 8601)	No	Watermark for incremental sync (see endpoint above).

Body

n/a

Response

A successful call returns status code 200 (Ok) with an array of RLDocumentDto objects (see the field table for the institute-wide call above). When the employee is not found for this institute, status code 404 is returned.

Example request

Request

```
GET https://<endpoint>/employees/3fa85f64-5717-4562-b3fc-2c963f66afa6/documents?schoolyear=2025-26
HTTP/1.1
InstituteNo: 999999
Api-Version: 1
Authorization: BEARER eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...
Accept: application/json
```

Example response

Response

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
[
  {
    "documentId": 42,
    "personId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "rIType": "RL2",
    "tekenStatus": null,
    "zendingnr": 12345,
    "referentie": "REF-2025-09-001",
    "aangemaaktOp": "2026-01-30T09:15:00Z",
    "verstuurdNaarEdisonOp": null,
    "edisonStatusGewijzigdOp": "2026-01-30T09:15:00Z"
  }
]
```

GET /employees/{personId}/documents/{documentId}/pdf

Downloads the PDF of an RL document. Authenticate with the Bearer token – no temporary download URLs are required. If the document has already been signed (tekenStatus = Ondertekend), the signed copy is returned.

After a successful download the document's tekenStatus is set to Opgehaald if it was previously null.

Request

Headers

Standard mandatory headers (Authorization, InstituteNo).

URL

Field	Type	Mandatory	Description
personId	guid	Yes	Unique identifier of the employee.
documentId	int32	Yes	Unique id of the RL document, taken from the documentId field in the GET /documents response.

Body

n/a

Response

A successful call returns status code 200 (Ok) with the PDF content as a binary application/pdf stream. The Content-Disposition header suggests a filename for saving.

When the document is not found, status code 404 is returned with code BR_DOC_003. When the PDF blob is missing in storage, code BR_DOC_004 is returned.

Example request

Request

```
GET https://<endpoint>/employees/3fa85f64-5717-4562-b3fc-2c963f66afa6/documents/42/pdf HTTP/1.1
InstituteNo: 999999
Authorization: BEARER eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...
```

Example response

Response

```
HTTP/1.1 200 OK
Content-Type: application/pdf
Content-Length: 204800
Content-Disposition: attachment; filename="RL2_Janssen_Jan_20260130.pdf"
<binary PDF bytes>
```

POST /employees/{personId}/documents/{documentId}/signeddocument

Submits the outcome of an external signing procedure to Informat. Call this endpoint as soon as your signing environment has completed the signing flow – regardless of whether the employee signed, refused, or a technical failure occurred.

A document can be submitted as “Ondertekend” only once. A second attempt for the same document returns HTTP 409 with code BR_DOC_006.

Request

Headers

Besides the standard headers, the request must use Content-Type: multipart/form-data. When status is “Ondertekend”, the bestand part must be a valid application/pdf payload.

URL

Field	Type	Mandatory	Description
-------	------	-----------	-------------

personId	Guid	Yes	Unique identifier of the employee.
documentId	int32	Yes	Unique id of the RL document.

Body

The request body is multipart/form-data with the following parts:

Field	Type	Mandatory	Description
status	String	Yes	Outcome of the signing procedure. Allowed values: Ondertekend, Geweigerd, Bekeken, Mislukt.
tijdstip	string (ISO 8601)	Yes	When the procedure was completed (UTC).
bestand	binary (application/pdf)	Conditional	The signed PDF file. Mandatory when status = Ondertekend. Must be a valid PDF, max 50 MB. Not present when status is Geweigerd, Bekeken or Mislukt.

When to submit which status:

Status	When to use	PDF required?
Ondertekend	The employee signed the document.	Yes
Geweigerd	The employee refused to sign.	No
Bekeken	The document was opened but no signature was placed yet.	No
Mislukt	A technical failure occurred in your signing environment.	No

Response

A successful call returns status code 204 (No Content). When status = Ondertekend, the supplied PDF is stored and subsequent calls to the PDF endpoint return the signed copy.

When validation fails, status code 400 (Bad Request) is returned with one or more BR_SIGN_xxx codes. See the section "Bad Request errors" for the complete list.

When the document is not found for this employee/institute, status code 404 is returned with code BR_DOC_003. When no source PDF is available to be signed, code BR_DOC_005 is returned.

When the document has already been signed, status code 409 (Conflict) is returned with code BR_DOC_006.

Example request

Request

```
POST https://<endpoint>/employees/3fa85f64-5717-4562-b3fc-2c963f66afa6/documents/42/signeddocument
HTTP/1.1
InstituteNo: 999999
Authorization: BEARER eyJ0eXAI0iJKV1QiLCJhbGciOiJIUzI1NiJ9...
Content-Type: multipart/form-data; boundary=----Boundary

----Boundary
Content-Disposition: form-data; name="status"

Ondertekend
----Boundary
Content-Disposition: form-data; name="tijdstip"

2026-02-01T10:32:00Z
----Boundary
Content-Disposition: form-data; name="bestand"; filename="signed.pdf"
Content-Type: application/pdf

<binary PDF bytes>
----Boundary--
```

Example response

Response:

```
HTTP/1.1 204 No Content
```

Response with validation error(s):

```
HTTP/1.1 400 Bad Request
Content-Type: application/json; charset=utf-8
{
  "errors": [
    {
      "Code": "BR_SIGN_004",
      "Message": "PDF file is required when status is 'Ondertekend'."
    }
  ]
}
```

Response when document is already signed:

```
HTTP/1.1 409 Conflict
Content-Type: application/json; charset=utf-8
{
  "errors": [
    {
      "Code": "BR_DOC_006",
      "Message": "Document 42 is already signed and cannot be overwritten."
    }
  ]
}
```

APPENDIX Error-codes and descriptions

Bad Request errors

Code	Description
BR001	Invalid PersonId

BR002	Invalid InterruptionId
BR003A	InstelNr is required
BR004A	Schooljaar is required
BR004B	Invalid Schooljaar format
BR005A	Hfd is required
BR005B	Hfd must have a length of 3 characters
BR006A	Code is required
BR006B	Invalid Code
BR007A	Begindatum is required
BR008A	Einddatum is required
BR008B	Einddatum cannot be before Begindatum
BR009	Opmerking doesn't comply with the validationrule: max. length 500
BR010	NaamGeneesheer doesn't comply with the validationrule: max. length 75
BR011	TelefoonGeneesheer doesn't comply with the validationrule: max. length 20
BR012A	Afzender is required
BR012B	Afzender doesn't comply with the validationrule: max. length 50
BR013	Invalid combination InstituteNo, School year and Structure
BR014	No employee found for this PersonId
BR015	Invalid AttachmentId
BR016A	Filename is required
BR016B	Filename doesn't comply with the validationrule: max. length 100
BR016C	Invalid file extension
BR017A	File is required
BR017B	Max. file size: 10MB
BR018	No interruption found for this Id
BR019	Nothing found to delete for this interruptionId & attachmentId combination
BR020	Soort must be 1 or 3
BR021	Type is invalid
BR022	Number is required
BR023	Number format is invalid
BR024	This employee is not active in institute {instituteNo}
BR025	A communication number already exists for this combination 'Soort & Type'; Id: {id}
BR026	A communication number already exists for this combination 'Soort, Type & Nummer'; Id: {id}
BR027	Invalid Communication Number Id
BR028	No Communication Number found for this Id
BR029A	Dag Deel is required
BR029B	Invalid Dag Deel for Code

BR030	Uren Staking is required
BR_DOC_001	'schoolyear' is required
BR_DOC_002	Invalid 'schoolyear' format (expected: YYYY-YY)
BR_DOC_003	No document found with the supplied id for this employee
BR_DOC_004	PDF file for the document was not found in blob storage
BR_DOC_005	No source PDF file is available to be signed for this document
BR_DOC_006	Document is already signed (conflict)
BR_DOC_007	Unsupported 'rlType' (only 'RL1' and 'RL2' are valid)
BR_DOC_008	Invalid 'referencedate' format (expected: ISO 8601)
BR_SIGN_001	'status' is required
BR_SIGN_002	Invalid status (allowed: Ondertekend, Geweigerd, Bekeken, Mislukt)
BR_SIGN_003	'tijdstip' is required and must be a valid datetime
BR_SIGN_004	PDF file is required when status is 'Ondertekend'
BR_SIGN_005	Supplied file is not a valid PDF document
BR_SIGN_006	PDF file is larger than 50 MB

Method Not Allowed errors

Code	Description
MNA001	This interruption is already linked to another school (institute no, school year & structure)
MNA002	This interruption is already linked to another employee
MNA003	Hfd, Code, Begin- and Einddatum can't be changed as the interruption has already been sent to AgODi
MNA004	Hfd, Begin- and Einddatum can't be changed as the interruption has already a substitution
MNA005	This interruption can't be added because an overlapping personal interruption already exists for this institute no.
MNA006	This interruption can't be adjusted because an overlapping personal interruption already exists for the institute no.
MNA007	Attachment already linked to another employee
MNA008	Only interruptions added via the API can be removed
MNA009	This interruption is already verified by the secretariat and can no longer be removed
MNA010	This interruption is already sent to AgODi and can no longer be removed
MNA011	This interruption has already a substitution and can no longer be removed

Document version history

The table below tracks the history of the different versions of this document and the respective release dates of the Api. The type can be initial / minor / major.

Doc Version	Release Date	Type	Changelog
1	July 3, 2023	Initial release	<ul style="list-style-type: none"> - First version containing the call: <ul style="list-style-type: none"> o GET /employees?structure={structure}&schoolYear={schoolYear}
2	January 30, 2024	Major	<ul style="list-style-type: none"> - A new version "Api-version 2" has been added for the call: <ul style="list-style-type: none"> o GET /employees?schoolYear={schoolYear}&structure={structure} - The following 11 new calls are implemented: <ul style="list-style-type: none"> o GET /employees/{personId}?schoolYear={schoolYear}&structure={structure} o GET /employees/assignments?schoolYear={schoolYear}&structure={structure}&inclNonSubHours={inclNonSubHours} o GET /employees/{personId}/assignments?schoolYear={schoolYear}&structure={structure}&inclNonSubHours={inclNonSubHours} o GET /employees/interruptions?schoolYear={schoolYear}&structure={structure} o GET /employees/{personId}/interruptions?schoolYear={schoolYear}&structure={structure} o GET /employees/diplomas?schoolYear={schoolYear}&structure={structure} o GET /employees/{personId}/diplomas?schoolYear={schoolYear}&structure={structure} o GET /employees/appointments?schoolYear={schoolYear}&structure={structure} o GET /employees/{personId}/appointments?structure={structure} o GET /employees/photos?schoolYear={schoolYear}&structure={structure} o GET /employees/{personId}/photos
2.1	February 13, 2024	Minor	<ul style="list-style-type: none"> - The following 2 new calls are implemented: <ul style="list-style-type: none"> o GET /employees/ownfields?schoolYear={schoolYear}&structure={structure} o GET /employees/{personId}/ownfields?schoolYear={schoolYear}&structure={structure}
2.2	September 10, 2024	Minor	<ul style="list-style-type: none"> - The response of the following calls is extended with "opmSecretariaat" and "opmPersoneelslid": <ul style="list-style-type: none"> o GET /employees/interruptions?schoolYear={schoolYear}&structure={structure} o GET /employees/{personId}/interruptions?schoolYear={schoolYear}&structure={structure} - One new call is implemented: <ul style="list-style-type: none"> o PUT /employees/{personId}/interruptions/{interruptionId}
2.3	September 17, 2024	Minor	<ul style="list-style-type: none"> - The following 2 new calls are implemented: <ul style="list-style-type: none"> o PUT /employees/interruptions/{interruptionId}/attachments/{attachmentId}

Doc Version	Release Date	Type	Changelog
			<ul style="list-style-type: none"> DELETE /employees/interruptions/{interruptionId}/attachments/{attachmentId}
2.4	September 20, 2024	Minor	<ul style="list-style-type: none"> One new call is implemented: <ul style="list-style-type: none"> DELETE /employees/interruptions/{interruptionId}
2.5	February 4, 2025	Minor	<ul style="list-style-type: none"> The body of the "PUT /employees/{personId}/interruptions/{interruptionId}" call is extended with the "verified" property.
2.6	April 24, 2025	Minor	<ul style="list-style-type: none"> For the calls "GET /employees?schoolYear={schoolYear}&structure={structure}" & "GET /employees/{personId}?schoolYear={schoolYear}&structure={structure}" <ul style="list-style-type: none"> The list of "personeelsgroepen" is now filtered based on the provided instituteNo (header) & optional structure parameter (see relevant paragraph for more details), The list of children is extended with the properties "fiscaalTenLaste" and "fiscaalTenLasteTot"; The response for the "GET assignments" calls is extended with "vakRubriek" and "gelijkgesteldVakRubriek". The response for the "GET interruptions" calls is extended with "staging" and "stagingUren".
2.7	September 29, 2025	Minor	<ul style="list-style-type: none"> The response of the following calls is extended with "aantalKinderen", "mindervalideKinderen", "aantalZorgbehoevenden", "aantalAndereTenLaste" and "aantalMindervalideAndereTenLaste": <ul style="list-style-type: none"> GET /employees?schoolyear={schoolYear}&structure={structure} GET /employees/{personId}?schoolyear={schoolYear}&structure={structure} The response of the following calls is extended with "ziekteverlofZonderBezoldiging": <ul style="list-style-type: none"> GET /employees/interruptions?schoolyear={schoolYear}&structure={structure} GET /employees/{personId}/interruptions?schoolyear={schoolYear}&structure={structure} The body of the "PUT /employees/{personId}/interruptions/{interruptionId}" call is extended with the "ziekteverlofZonderBezoldiging" property.
2.8	October 13, 2025	Minor	<p>The following call is extended with the parameter changedSince:</p> <ul style="list-style-type: none"> GET /employees/photos?schoolyear={schoolYear}&structure={structure}&changedSince={changedSince}
2.9	Januari 20, 2026	Minor	<ul style="list-style-type: none"> The following 2 new calls are implemented: <ul style="list-style-type: none"> PUT /employees/{PersonId}/communication-number DELETE /employees/{personId}/communication-number/{communicationNold}
2.10	March 17, 2026	Minor	<ul style="list-style-type: none"> The response of the following calls is extended with "effectiefInDienst": <ul style="list-style-type: none"> GET /employees?schoolyear={schoolYear}&structure={structure}

Doc Version	Release Date	Type	Changelog
			<ul style="list-style-type: none"> ○ GET /employees/{personId}?schoolyear={schoolYear}&structure={structure} <p>- The body of the “PUT /employees/{personId}/interruptions/{interruptionId}” call is extended with the “dagDeel” and “urenStaking” properties.</p>
2.11	April 28, 2026	Minor	<p>- The following 4 new calls are implemented for digital signing of RL documents:</p> <p>GET /employees/documents?schoolyear={schoolYear}</p> <p>GET /employees/{personId}/documents?schoolyear={schoolYear}</p> <p>GET /employees/{personId}/documents/{documentId}/pdf</p> <p>POST /employees/{personId}/documents/{documentId}/signeddocument</p>
2.12	May 12, 2026	Minor	<p>- For endpoint “PUT /employees/{personId}/interruptions/{interruptionId}”, interruption codes 197, 200, 203, 206, and 209 have been added as valid values. These codes are related to “voltijds zorgkrediet”; see the <i>Interruptions</i> section for more information.</p> <p>The overlapping rules MNA005 and MNA006 have been updated to allow code 1 (ziekteverlof) to overlap with any of the newly added codes, and vice versa.</p> <p>- Unique keys pAdres, pComnr & pEmail are added to the response of the GET /employees endpoints.</p> <p>- Unique key pOpdracht is added to the response of the GET Assignments endpoints.</p> <p>- Unique key pOnderbreking is added to the response of the GET Interruptions endpoints.</p> <p>- Unique key pDiploma is added to the response of the GET Diplomas endpoints.</p> <p>- Unique key pBenoeming is added to the response of the GET Appointments endpoints.</p>

References

- Wikipedia on RESTful services, http://en.wikipedia.org/wiki/Representational_state_transfer
- MSDN, <http://msdn.microsoft.com>

List of attachments

Attachment 1 – List of result codes:

Result code	Resources	Description
200	All	The request was received and processed correctly. The requested data has been attached.
400	All	The server was unable to parse the request because of a problem with the content of the request, this can be a problem with the headers / URL / body data that was received.
401	All	The request was sent without a valid signature. This could mean that the signature header was not present, that the signature value wasn't present or that the provided signature is not correct.
404	All	The request was received and processed correctly, but there is no data in the response set.
405	All	The request was sent using an invalid HttpMethod. Example: Sending a request as GET while only POST is allowed.
500	All	The request triggered a handled exception on the service. The exception has been logged. Please try again and/or contact Informat about this problem.
999	All	The request triggered an unexpected exception on the service. The exception has been logged. Please try again and/or contact Informat about this problem.

Sample code (.NET)

```
// ClientId & ClientSecret received from Informat
private const string ClientId = "informat_customer_{clientName}";
private const string ClientSecret = "mySecretCode";
// address to retrieve the access token
private const string IdentityServerBaseAddress = "https://www.identityserver.be";
// address leerlingenapi
private const string ApiBaseAddress = "https://personeelsapi.informatsoftware.be";

private static async Task<string> CallApi() {
    // the scopes for which the api can be called (multiple scopes possible)
    var scopes = new List<string>
    {
        "api_informat_sas_personeel.personeel.{123456}"
    };

    // get token from identityserver
    var token = await GetToken(scopes);

    // create HttpClient and use received Bearer token
    using (var apiClient = new HttpClient
    {
        BaseAddress = new Uri(ApiBaseAddress),
        DefaultRequestHeaders = { Authorization = new AuthenticationHeaderValue("Bearer", token) }
    })
    {
        // set InstituteNo in header for which the request is done
        apiClient.DefaultRequestHeaders.Add("InstituteNo", "123456");
        // call leerlingenapi
        var result = await apiClient.GetStringAsync("/{api method name}");
        return result;
    }
}

private static async Task<string> GetToken(IEnumerable<string> scopes) {
    using var tokenClient = new HttpClient { BaseAddress = new Uri(IdentityServerBaseAddress) };
    var response = await tokenClient.PostAsync("/connect/token", new FormUrlEncodedContent(
        new Dictionary<string, string> {
            { "client_id", ClientId },
            { "client_secret", ClientSecret },
            { "grant_type", "client_credentials" },
            { "scope", string.Join(" ", scopes) }
        }
    ));
    response.EnsureSuccessStatusCode();
    var content = await response.Content.ReadAsStringAsync();
    var json = JsonDocument.Parse(content);
    var token = json.RootElement.GetProperty("access_token").GetString();
    return token;
}
```