Internaat WEB API

Implementation guide

Document version	V1.1
Release date	September 24, 2024
Last modified	September 18, 2024
Last modified by	Pierre-Marie Mahieu

Contents

Document description 3
API Basics 4
Endpoint 4
Versioning 4
Character sets 4
Request Format and Responses 5
JSON Basics 5
HTTP Headers 6
Request Headers 6
Response Headers 7
HTTP Status Codes 7
Usage restrictions 8
Ground rules 8
Privacy concerns 8
Disclaimer 8
Security 9
Transport security 9
Secure Socket Layer 9
Authentication endpoint 9
Request authentication 9
Calling resources without Access Token 9
Getting an Access Token by client credentials 10
Getting access for 1 or multiple scopes 10
Request token for 1 scope 10
Request token for multiple scopes 11
Use of access token 12
Request 12
Resources 13
Registrations 13
GET /registrations?schoolYear={schoolYear} 13
Pupils 16
GET /pupils?schoolyear={schoolYear}&refdate={referenceDate} 16
GET /pupils/{pupilId}?schoolYear={schoolYear}&refdate={referenceDate} 22
Document version history 24
References 25
List of attachments 26
Sample code (.NET) 27

Document description

This document describes the “Internaat” 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 Internaat context.

Endpoint

The **production environment** is available via:

```
endpoint = internaatapi.informatsoftware.be
```

Versioning

The Internaat 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 Internaat 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 Internaat operation. The list of response headers important to Internaat are listed below.

HTTP Status Codes

With the interface to Internaat 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.

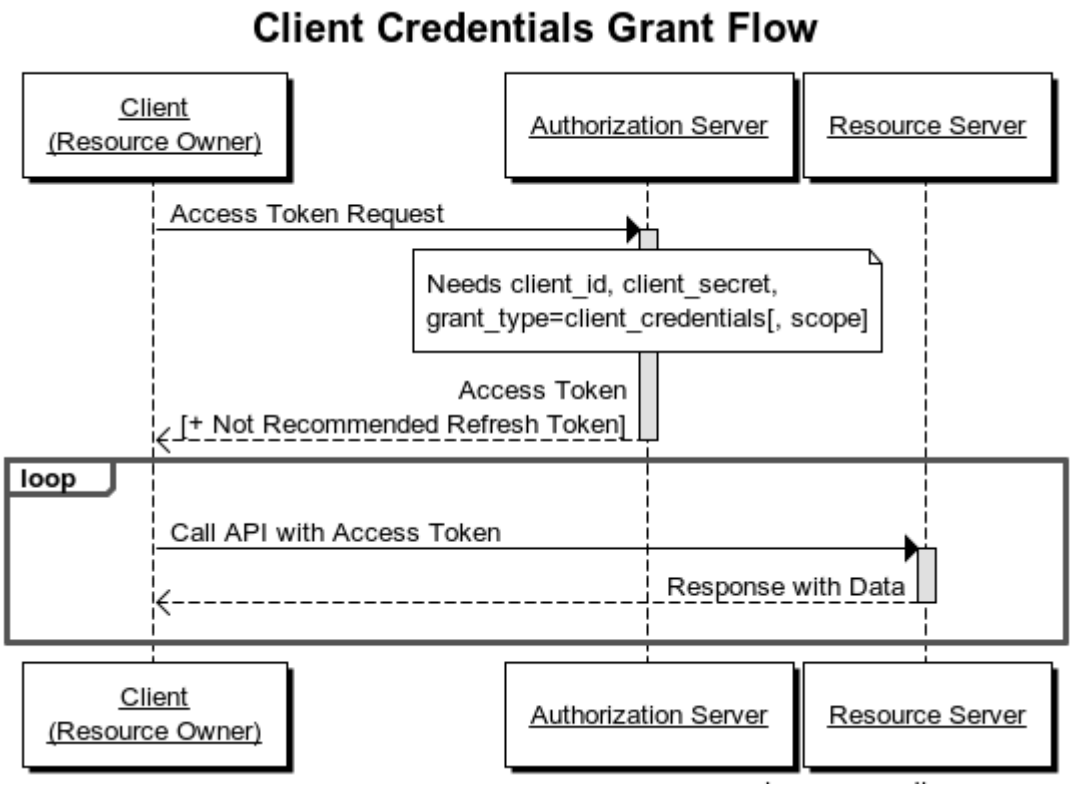
Calling resources without Access Token

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.



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_internaat.123456** scope is permitted to read or write data to the Internaatapi 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 https://<identityEndpoint>/connect/token    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_internaat.123456
```

Response

```
HTTP/1.1 200 OK
...

{
"access_token":"eyJhbGciOiJSUzI1NiIsImtpZCI6IjYyOEM3NzFEM0FCNEI0Rjk4MzRFODkyMTcyMTc2OTk5NjkyNTZFOEQiLCJ
0eXAiOiJKV1QiLCJ4NXQiOiJZb3gzSFRxMHRQbURUb2toY2hkcG1Xa2xibzAifQ.eyJuYmYiOjE1OTM2NzM1MDcsImV4cCI6MTU5MzY
3NzEwNywiaXNzIjoiaHR0cHM6Ly93d3cuaWRlbnRpdHlzZXJ2ZXIuYmUiLCJhdWQiOlsiaHR0cHM6Ly93d3cuaWRlbnRpdHlzZXJ2ZX
IuYmUvcmVzb3VyY2VzIiwiYXBpX2luZm9ybWF0X3Nhc19sZWVybGluZ2VuIl0sImNsaWVudF9pZCI6ImluZm9ybWF0X2N1c3RvbWVyX
2hoc2Nob2xlbiIsInNjb3BlIjpbImFwaV9pbmZvcm1hdF9zYXNfbGVlcmxpbmdlbi52b29yaW5zY2hyaWp2aW5nZW4uMDMyODQ3Il19
.vfTPZYPGGsbbl4Vy1S-FJdvPQTDlcT5m4E2mWFPqPMRBIQINqQBiNfxpH8DUu5cMRvcTFJxCeAsm33Rf-
my0jR1qVBYIlrKgmbIth_gyCNAycHUBAUsd-nYIJIv3Jz20_zM1L3gklUL48fCYc-
hwT1rbS4uRIpHQxDd_sdsNyvUjB9DgMCdxuoqBm0cVzmVvHlH6Fpez-
ttQ5Aymsj0dx0JvAIItVjxVHPqRwA33Sl1iSVogOjfjHP437d4ztFOjzAGUHglwO4eF10ALe_d8D1w3CqjAJGOaqP_W9ttPTATuPvwK
Q2xYaGiL1kPsejTpoRCd176vjv9i3b1dxPiw6FcVffeMUguetub8l8mHVEd0eYEDKUoTmZtp_bAhu5sUq-
eOC3GcJtGwC9TfjTQLtTdgniEfqhX81TZI3Kh2th4z5SOROKt5Sq5RtnN9U9WDrci2DQrfkt5pNg7leMIMqF8AsvuYp5rXGR-
HYWN2YB1uf2YTwIa-wOCshixjdzV4",
"expires_in": 3600,
"token_type": "Bearer",
"scope": "api_informat_sas_internaat.123456"
}
```

Request token for multiple scopes

Doing a request for multiple scopes can be done by defining all the scopes, separated by a single space.

Request

```
POST https://<identityEndpoint>/connect/token    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_internaat.123456 api_informat_sas_internaat.456789
```

Response

```
HTTP/1.1 200 OK
...

{
"access_token":"eyJhbGciOiJSUzI1NiIsImtpZCI6IjYyOEM3NzFEM0FCNEI0Rjk4MzRFODkyMTcyMTc2OTk5NjkyNTZFOEQiLCJ
0eXAiOiJKV1QiLCJ4NXQiOiJZb3gzSFRxMHRQbURUb2toY2hkcG1Xa2xibzAifQ.eyJuYmYiOjE1OTM2NzM1MDcsImV4cCI6MTU5MzY
3NzEwNywiaXNzIjoiaHR0cHM6Ly93d3cuaWRlbnRpdHlzZXJ2ZXIuYmUiLCJhdWQiOlsiaHR0cHM6Ly93d3cuaWRlbnRpdHlzZXJ2ZX
IuYmUvcmVzb3VyY2VzIiwiYXBpX2luZm9ybWF0X3Nhc19sZWVybGluZ2VuIl0sImNsaWVudF9pZCI6ImluZm9ybWF0X2N1c3RvbWVyX
2hoc2Nob2xlbiIsInNjb3BlIjpbImFwaV9pbmZvcm1hdF9zYXNfbGVlcmxpbmdlbi52b29yaW5zY2hyaWp2aW5nZW4uMDMyODQ3Il19
.vfTPZYPGGsbbl4Vy1S-FJdvPQTDlcT5m4E2mWFPqPMRBIQINqQBiNfxpH8DUu5cMRvcTFJxCeAsm33Rf-
my0jR1qVBYIlrKgmbIth_gyCNAycHUBAUsd-nYIJIv3Jz20_zM1L3gklUL48fCYc-
hwT1rbS4uRIpHQxDd_sdsNyvUjB9DgMCdxuoqBm0cVzmVvHlH6Fpez-
ttQ5Aymsj0dx0JvAIItVjxVHPqRwA33Sl1iSVogOjfjHP437d4ztFOjzAGUHglwO4eF10ALe_d8D1w3CqjAJGOaqP_W9ttPTATuPvwK
Q2xYaGiL1kPsejTpoRCd176vjv9i3b1dxPiw6FcVffeMUguetub8l8mHVEd0eYEDKUoTmZtp_bAhu5sUq-
eOC3GcJtGwC9TfjTQLtTdgniEfqhX81TZI3Kh2th4z5SOROKt5Sq5RtnN9U9WDrci2DQrfkt5pNg7leMIMqF8AsvuYp5rXGR-
HYWN2YB1uf2YTwIa-wOCshixjdzV4",
"expires_in": 3600,
"token_type": "Bearer",
"scope": "api_informat_sas_internaat.123456 api_informat_sas_internaat.456789"
}
```

Use of access token

Each API Request must send with the access token as follows

Authorization: BEARER <token>

Example

```
GET https://internaatapi.informatsoftware.be/... HTTP/1.1
Authorization: BEARER
eyJhbGciOiJSUzI1NiIsImtpZCI6IjYyOEM3NzFEM0FCNEI0Rjk4MzRFODkyMTcyMTc2OTk5NjkyNTZFOEQiLCJ0eXAiOiJKV1QiLCJ
4NXQiOiJZb3gzSFRxMHRQbURUb2toY2hkcG1Xa2xibzAifQ.eyJuYmYiOjE1OTM2NzM1MDcsImV4cCI6MTU5MzY3NzEwNywiaXNzIjo
iaHR0cHM6Ly93d3cuaWRlbnRpdHlzZXJ2ZXIuYmUiLCJhdWQiOlsiaHR0cHM6Ly93d3cuaWRlbnRpdHlzZXJ2ZXIuYmUvcmVzb3VyY2
VzIiwiYXBpX2luZm9ybWF0X3Nhc19sZWVybGluZ2VuIl0sImNsaWVudF9pZCI6ImluZm9ybWF0X2N1c3RvbWVyX2hoc2Nob2xlbiIsI
nNjb3BlIjpbImFwaV9pbmZvcm1hdF9zYXNfbGVlcmxpbmdlbi52b29yaW5zY2hyaWp2aW5nZW4uMDMyODQ3Il19.vfTPZYPGGsbbl4V
y1S-FJdvPQTDlcT5m4E2mWFPqPMRBIQINqQBiNfxpH8DUu5cMRvcTFJxCeAsm33Rf-my0jR1qVBYIlrKgmbIth_gyCNAycHUBAUsd-
nYIJIv3Jz20_zM1L3gklUL48fCYc-hwT1rbS4uRIpHQxDd_sdsNyvUjB9DgMCdxuoqBm0cVzmVvHlH6Fpez-
ttQ5Aymsj0dx0JvAIItVjxVHPqRwA33Sl1iSVogOjfjHP437d4ztFOjzAGUHglwO4eF10ALe_d8D1w3CqjAJGOaqP_W9ttPTATuPvwK
Q2xYaGiL1kPsejTpoRCd176vjv9i3b1dxPiw6FcVffeMUguetub8l8mHVEd0eYEDKUoTmZtp_bAhu5sUq-
eOC3GcJtGwC9TfjTQLtTdgniEfqhX81TZI3Kh2th4z5SOROKt5Sq5RtnN9U9WDrci2DQrfkt5pNg7leMIMqF8AsvuYp5rXGR-
HYWN2YB1uf2YTwIa-wOCshixjdzV4
```

Request

It is mandatory that you provide some information with every request:

- Your institute number

Institute number

Your institute is added as a header to **every** request. This header is called “InstituteNo”.

Note that header names are case-insensitive.

C# coding example

```
Request.Headers.Add("InstituteNo", "999999");
```

Please make sure that you only add this header once to the request, if you try to add it twice (or with multiple values), the header will be sent as a comma-separated list of values.

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.

Registrations

This chapter describes how to:

- Get all registrations for an institute number and school year.

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

GET /registrations?schoolYear={schoolYear}

Gets all the registrations for the combination institute number and school year. The institute number is defined via the mandatory header "InstituteNo" and the school year must be send via the URL.

Request

Headers

Besides the mandatory headers, an optional version number can be provided.
Accepted versions are: 1

URL

Field	Type	Required/optional	Description
schoolYear	string	required	Limits the output results to registrations within the given schoolyear. **Format:** `"2022-23"` If not provided a bad request (400) error will be returned.

Body

This request has an empty body.

Response

The response contains a list of registrations.

Field	Type	Description
inschrijvingsId	GUID	Unique identifier for the registration **Format**: `{42CD5A20-E22F-45D7-973C-FAB8734DEC50}`
pPersoon	integer	Unique integer-value for the pupil within the database
persoonId	GUID	Unique identifier for the pupil **Format**: `{42CD5A20-E22F-45D7-973C-FAB8734DEC50}`
instelnr	string	Official institute number of the school **Format**: `6 characters (digits)`
hfdstructuur	string	Structure of the institute (IA0, IB0,…)
internaat	string	Descriptive name of the boarding school
stamnr	string	Students' stamnummer **Format**: `9 characters (digits) or null if not available`
vestCode	string	Own defined location code
vestNr	string	Official Discimus location number

vestiging	string	Own descriptive name of the location
begindatum	date	Date the registration starts **Format**: : yyyy-MM-dd
einddatum	date	Date the registration ends **Format**: : yyyy-MM-dd or null if empty

fin	string	Finance ability
		Possible values:
		`Regelmatig financierbaar`
		`Vrije leerling`
		`Regelmatig niet-financierbaar`

afsprakenkaderEnOndersteuningsplan	boolean	Indicates whether there is an 'Agreement framework and support plan' drawn up between the support services Welfare and the boarding school.
kamers	array	List of rooms
blok	string	Block in which the room is located
verdiep	string	Floor where the room is located
nr	integer	Room number
begindatum	date	Date on which the stay in the room starts **Format**: : yyyy-MM-dd
einddatum	date	Date on which the stay in the room ends **Format**: : yyyy-MM-dd or null if empty

Example request

Request

```
POST https://<endpoint>/registrations?schoolYear=2022-23 HTTP/1.1
InstituteNo: 999999
Api-Version: 1
Authorization: BEARER
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1bmlxdWVfbmFtZSI6IkluZm9ybWF0IiwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdC
IsImF1ZCI6ImFlZjAwMjdlNzNmOTU0MzVmNzc4MWEwY2ZhZDU5ZTVkIiwiZXhwIjoxNTc0NTIxODM3LCJuYmYiOjE1NzQzNDkwMzd9.
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, 14 Aug 2023 16:06:30 GMT
```

```
[
    {
        "inschrijvingsId": "dbaacd24-43d8-4b83-a201-f2178867e08d",
        "pPersoon": 133075,
        "persoonId": "2e79366e-14de-4a5e-9d7b-797d01a07267",
        "instelnr": "999999",
        "hfdstructuur": "II0",
        "internaat": "Internaat Informat",
        "stamnr": "202200001",
        "vestCode": "001",
        "vestNr": 1,
        "vestiging": "Collegestraat",
        "begindatum": "2023-09-01T00:00:00",
        "einddatum": null,
        "fin": "Regelmatig financierbaar",
        "afsprakenkaderEnOndersteuningsplan": false,
        "kamers": [
            {
                "blok": "A",
                "verdiep": "3",
                "nr": "301",
                "begindatum": "2023-09-01T00:00:00",
                "einddatum": null
            }
        ]
    },
    {
        "inschrijvingsId": "52389972-1064-46ac-8aa2-29de9eb490a2",
        "pPersoon": 133094,
        "persoonId": "d4faf3ae-9515-4104-a73a-aa72d9a0a646",
        "instelnr": "999999",
        "hfdstructuur": "II0",
        "internaat": "Internaat Informat",
        "stamnr": "202300012",
        "vestCode": "001",
        "vestNr": 1,
        "vestiging": "Collegestraat",
        "begindatum": "2023-01-02T00:00:00",
        "einddatum": null,
        "fin": "Regelmatig financierbaar",
        "afsprakenkaderEnOndersteuningsplan": false,
        "kamers": []
    },
    {…}
]
```

Pupils

This chapter describes how to:

- Get all pupils for an institute number, school year and reference date ;
- Get a pupil by pupilId

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

GET /pupils?schoolyear={schoolYear}&refdate={referenceDate}

Gets all pupils with a registration for a given institute number, school year and reference date falling between registration's begin & end date.

The institute number is defined via the mandatory header "InstituteNo" and the school year must be send via the URL. The ReferenceDate is optional, but will be calculated automatically if not provided (see request for more information).

Request

Headers

Besides the mandatory headers, an optional version number can be provided.
Accepted versions are: 1

URL

Field	Type	Required/optional	Description
schoolYear	string	required	Limits the output results to students which have a registration within the given schoolyear. **Format:** "2022-23" If not provided a bad request (400) error will be returned.
referenceDate	date	optional	Limits the output results to students which have a registration where the reference date falls between registration's begin & end date. If the reference date is not provided, today's date will be taken. **In any case the reference date will be overridden as follows if it falls outside the boundaries of the provided school year:** - Reference date < beginning of school year => reference date will be overridden with begin date of school year; - Reference date > end of school year => reference date will be overridden with end date of school year.

Body

This request has an empty body.

Response

The response contains a list of pupils.

Field	Type	Description
pPersoon	integer	Unique integer-value for the pupil within the database.
persoonId	GUID	Unique identifier for the pupil. **Format:** {42CD5A20-E22F-45D7-973C-FAB8734DEC50}
naam	string	Pupil's last name.
voornaam	string	Pupil's first name.

geboortedatum	date	Pupil's date of birth. **Format:** yyyy-MM-dd
nickname	string	Pupil's nickname.
voornaam2	string	Pupil's additional names.
initialen	string	Pupil's initials.
geboorteland	string	Pupil's country of birth.
geboorteplaats	string	Pupil's place of birth.
nationaliteitCode	string	Pupil's official nationality codes.
rijksregisternr	string	Pupil's national registration number for Belgium residents. **Format:** "yymmddxxxxx" if provided
bisnr	string	Pupil's national registration number for a foreign person. **Format:** 11 characters (digits) if provided
geslacht	string	Pupil's sex. **Possible values:** "M" or "V"
huisdokter	string	Name of the pupil's doctor.
telefoonHuisdokter	string	Phone number of the pupil's doctor.
inschrijvingsId	GUID	Unique identifier of the actual registration. The **actual** registration is determined by the provided InstituteNo, school year, reference date (between registration's begin & end date). If no such registration is found, inschrijvingsId will be null.
adressen	array	List of domicile addresses (usually one).
pAdres	integer	Unique integer-value for the address within the database.
adresId	GUID	Unique identifier for the address. **Format:** {42CD5A20-E22F-45D7-973C-FAB8734DEC50}
aanspreekTitel	string	Addressable title.
aanspreekNaam	string	Addressable name.
straat	string	Street name.
nr	string	House number & Alpha number.
bus	string	Bus number
postcode	string	Postal code (main).
gemeente	string	Town (main).
isFacturatie	bool	Indicates whether this address is a invoice address.
isAanschrijf	bool	Indicates whether this address is a writing address.
isVerblijf	bool	Indicates whether this address is a residence address.
isOverige	bool	Indicates whether this address is of another type then the provided ones.
percentage	decimal	Cost allocation percentage.
relaties	array	List of relationships.
pRelatie	integer	Unique integer-value for the relationship within the database.
relatieId	GUID	Unique identifier for the relationship. **Format:** {42CD5A20-E22F-45D7-973C-FAB8734DEC50}
type	string	Relationship type **Examples:** Vader, Moeder, Plusvader, Plusmoeder, …

naam	string	Relation's last name
voornaam	string	Relation's first name
insz	string	Relation's national registration number. This is either the Bisnummer, for foreign pupil, or Rijksregisternummer for Belgium residents. **Format**: "yymmddxxxxx" **Example**: `"85073115025"`
geboortedatum	date	Relation's date of birth. **Format:** yyyy-MM-dd or null if empty
geslacht	string	Relation's sex. **Possible values:** "M" or "V"
nationaliteitCode	string	Relation's official nationality codes.
beroep	string	Relation's profession.
burgerlijkeStand	string	Relation's civil status.
lpv	integer	Indicates if the relation is marked as a school attendance officer and defines if it's the first or second one.
ophalen	bool	Indicates if the relation picks up the student from school.
isOverleden	bool	Indicates if the relation is deceased.
adressen	array	List of addresses linked to the relationship (usually one, and most likely a domicile address).
pAdres	integer	Unique integer-value for the address within the database.
adresId	GUID	Unique identifier for the address. **Format:** `{42CD5A20-E22F-45D7-973C-FAB8734DEC50}`
aanspreekTitel	string	Addressable title.
aanspreekNaam	string	Addressable name.
straat	string	Street name.
nr	string	House number & Alpha number.
bus	string	Bus number
postcode	string	Postal code (main).
gemeente	string	Town (main).
isFacturatie	bool	Indicates whether this address is a invoice address.
isAanschrijf	bool	Indicates whether this address is a writing address.
isVerblijf	bool	Indicates whether this address is a residence address.
isOverige	bool	Indicates whether this address is of another type then the provided ones.
percentage	decimal	Cost allocation percentage.
comnrs	array	List of communication numbers linked to the relationship.
pComnr	integer	Unique integer-value for the communication number within the database.
nr	string	Communication number.
type	string	Communication number type.
soort	string	Communication number sort. **Possible values:** Telefoon, Fax or Gsm
emails	array	List of email addresses linked to the relationship.
pEmail	integer	Unique integer-value for the email address within the database.

email	string	Email address.
type	string	Email address type. **Examples:** Eigen, School, Privé, Ouders, …
schoolcom	bool	Indicates if this email address can be used for school communication purposes.
factuurMailen	bool	Indicates if this email address can be used to mail invoices.
comnrs	array	List of communication numbers **not** linked to a relationship.
pComnr	integer	Unique integer-value for the communication number within the database.
nr	string	Communication number.
type	string	Communication number type. **Examples:** Eigen, Privé nummer, Ouders, Vader, Moeder, …
soort	string	Communication number sort. **Possible values:** Telefoon, Fax or Gsm
emails	array	List of email addresses **not** linked to a relationship
pEmail	integer	Unique integer-value for the email address within the database.
email	string	Email address.
type	string	Email address type. **Examples:** Eigen, School, Privé, Ouders, …
schoolcom	bool	Indicates if this email address can be used for school communication purposes.
factuurMailen	bool	Indicates if this email address can be used to mail invoices.

Example request

Request

```
POST https://<endpoint>/pupils?schoolYear=2022-23&refdate=2023-06-30 HTTP/1.1
InstituteNo: 999999
Api-Version: 1
Authorization: BEARER
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1bmlxdWVfbmFtZSI6IkluZm9ybWF0IiwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdC
IsImF1ZCI6ImFlZjAwMjdlNzNmOTU0MzVmNzc4MWEwY2ZhZDU5ZTVkIiwiZXhwIjoxNTc0NTIxODM3LCJuYmYiOjE1NzQzNDkwMzd9.
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, 14 Aug 2023 16:52:42 GMT
```

```
[
    {
        "pPersoon": 69780,
        "persoonId": "5b8f64c5-3968-4b13-b962-e4acda9601bc",
        "naam": "Andries",
        "voornaam": "Thomas",
        "geboortedatum": "2001-12-20T00:00:00",
        "nickname": "Andries Thomas",
        "voornaam2": "",
        "initialen": null,
        "geboorteland": "België",
        "geboorteplaats": "Diksmuide",
        "nationaliteitCode": "00150",
        "rijksregisternr": "01122018924",
        "bisnr": null,
        "geslacht": "M",
        "huisdokter": "Dr. Forever Young",
        "telefoonHuisdokter": "",
        "inschrijvingsId": "9de13451-1591-47d9-af25-97f323d35168",
        "adressen": [
            {
                "pAdres": 80000,
                "adresId": "67c7633a-f3d9-4f82-b600-8bc617cf851c",
                "aanspreekTitel": "Aan de vader van",
                "aanspreekNaam": "Andriessens Thomas",
                "straat": "Nijverheidstraat",
                "nr": "9",
                "bus": "",
                "postcode": "8600",
                "gemeente": "DIKSMUIDE",
                "isFacturatie": true,
                "isAanschrijf": true,
                "isVerblijf": true,
                "isOverige": false,
                "percentage": 100
            }
        ],
        "relaties": [
            {
                "pRelatie": 72936,
                "relatieId": "8a002367-18ac-45cd-a91c-493b2f317a69",
                "type": "Vader",
                "naam": "Andries",
                "voornaam": "Dieter",
                "insz": null,
                "geboortedatum": "1976-03-06T00:00:00",
                "geslacht": "M",
                "nationaliteitCode": "00150",
                "beroep": "Zelfstandige",
                "burgerlijkeStand": "Gescheiden",
                "lpv": 1,
                "ophalen": false,
                "isOverleden": false,
                "adressen": [
                    {
                        "pAdres": 80000,
                        "adresId": "67c7633a-f3d9-4f82-b600-8bc617cf851c",
                        "aanspreekTitel": "Aan de vader van",
                        "aanspreekNaam": "Andriessens Thomas",
                        "straat": "Nijverheidstraat",
                        "nr": "9",
                        "bus": "",
                        "postcode": "8600",
                        "gemeente": "DIKSMUIDE",
                        "isFacturatie": true,
                        "isAanschrijf": true,
                        "isVerblijf": true,
                        "isOverige": false,
```

```
                    "percentage": 100
                }
            ],
            "comnrs": [
                {
                    "pComnr": 156687,
                    "nr": "0404 95 22 52",
                    "type": "Vader",
                    "soort": "Gsm"
                }
            ],
            "emails": [
                {
                    "pEmail": 59294,
                    "email": "Dieter.Andries@informat.be",
                    "type": "Vader",
                    "schoolcom": true,
                    "factuurMailen": false
                }
            ]
        },
        {
            "pRelatie": 72937,
            "relatieId": "06124ed9-bec9-45e7-b5d3-0f553ca77d66",
            "type": "Moeder",
            "naam": "Somers",
            "voornaam": "Melissa",
            "insz": null,
            "geboortedatum": "1981-06-08T00:00:00",
            "geslacht": "V",
            "nationaliteitCode": "00150",
            "beroep": "Bediende",
            "burgerlijkeStand": "Gescheiden",
            "lpv": 2,
            "ophalen": false,
            "isOverleden": false,
            "adressen": [
                {
                    "pAdres": 80001,
                    "adresId": "67c7633a-f3d9-4f82-b600-8bc617cf851c",
                    "aanspreekTitel": "Aan de moeder van",
                    "aanspreekNaam": "Andriessens Thomas",
                    "straat": "Stationsstraat",
                    "nr": "100",
                    "bus": "",
                    "postcode": "8600",
                    "gemeente": "DIKSMUIDE",
                    "isFacturatie": true,
                    "isAanschrijf": true,
                    "isVerblijf": true,
                    "isOverige": false,
                    "percentage": 100
                }
            ],
            "comnrs": [
                {
                    "pComnr": 156688,
                    "nr": "0492 09 52 25",
                    "type": "Moeder",
                    "soort": "Gsm"
                }
            ],
            "emails": [
                {
                    "pEmail": 22720,
                    "email": "Melissa.Somers@informat.be",
                    "type": "Moeder",
                    "schoolcom": true,
```

```
                        "factuurMailen": true
                    }
                ]
            }
        ],
        "comnrs": [
            {
                "pComnr": 156689,
                "nr": "0492 25 55 29",
                "type": "Noodnummer",
                "soort": "Gsm"
            },
            {
                "pComnr": 188384,
                "nr": "025 04 09 52",
                "type": "Domicilie",
                "soort": "Telefoon"
            },
            {
                "pComnr": 188385,
                "nr": "0402 45 02 25",
                "type": "Eigen",
                "soort": "Gsm"
            },
            {
                "pComnr": 234237,
                "nr": "025 22 94 20",
                "type": "Alternatief nummer",
                "soort": "Telefoon"
            }
        ],
        "emails": [
            {
                "pEmail": 59295,
                "email": "Thomas.Andriessen@informat.be",
                "type": "Privé",
                "schoolcom": true,
                "factuurMailen": false
            }
        ]
    },
    {…}
]
```


GET /pupils/{pupilId}?schoolYear={schoolYear}&refdate={referenceDate}

Gets a pupil by it's pupilId. The institute number, school year and reference date are only used to determine the actual registration (inschrijvingsId property).
The institute number is defined via the mandatory header "InstituteNo" and the school year must be send via the URL. The ReferenceDate is optional, but will be calculated automatically if not provided (see request for more information).

Request

Headers

Besides the mandatory headers, an optional version number can be provided.
Accepted versions are: 1

URL

Field	Type	Required/optional	Description
pupilId	GUID	required	Limits the output results to a pupil with the provided pupilId. **Format:** `{42CD5A20-E22F-45D7-973C-FAB8734DEC50}` If not provided a bad request (400) error will be returned.

schoolYear	string	required	Is only used to determine the actual registration (inschrijvingsId property). So this parameter has no limiting effect on the output result. **Format:** "2022-23" If not provided a bad request (400) error will be returned.
referenceDate	date	optional	Is also only used to determine the actual registration (inschrijvingsId property). So this parameter has no limiting effect on the output result. If the reference date is not provided, today's date will be taken. **In any case the reference date will be overridden as follows if it falls outside the boundaries of the provided school year:** - Reference date < beginning of school year => reference date will be overridden with begin date of school year; - Reference date > end of school year => reference date will be overridden with end date of school year.

Body

This request has an empty body.

Response

The response contains one pupil instead of a list of pupils. This makes the response similar to the call "GET /pupils?schoolYear={schoolYear}&refdate={referenceDate}".

Example request

Request

```
POST https://<endpoint>/pupils/c5806777-4bce-4d50-afc5-9cb47f1a5f14?schoolYear=2022-23 HTTP/1.1
InstituteNo: 999999
Api-Version: 1
Authorization: BEARER
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1bmlxdWVfbmFtZSI6IkluZm9ybWF0IiwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdC
IsImF1ZCI6ImFlZjAwMjdlNzNmOTU0MzVmNzc4MWEwY2ZhZDU5ZTVkIiwiZXhwIjoxNTc0NTIxODM3LCJuYmYiOjE1NzQzNDkwMzd9.
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, 14 Aug 2023 09:36:29 GMT

{
   "pPersoon": 31236,
   "pPersoon": 69780,
   "persoonId": "5b8f64c5-3968-4b13-b962-e4acda9601bc",
   "naam": "Andries",
   "voornaam": "Thomas",
   "geboortedatum": "2001-12-20T00:00:00",
   …
   SIMULAR TO THE GLOBAL STUDENTS CALL
   …
}
```

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	August 22, 2023	Initial release	- First version containing the calls: o GET /registrations?schoolYear={schoolYear} o GET /pupils?schoolyear={schoolYear}&refdate={referenceDate} o GET /pupils/{pupilId}?schoolYear={schoolYear}&refdate={referenceDate}
1.1	September 24, 2024	minor	- The response of the following call is extended with 'afsprakenkaderEnOndersteuningsplan' o GET /registrations?schoolYear={schoolYear}

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://leerlingenapi.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_leerlingen.voorinschrijvingen.{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;
}
```