

OGC API Features Guideline

Geonovum Guide

Editor's draft September 13, 2023



Latest editor's draft:

<https://geonovum.github.io/ogc-api-features-guideline/>

Editor:

Pieter Bresters ([Geonovum](#))

Author:

Pieter Bresters ([Geonovum](#))

Participate:

[GitHub Geonovum/ogc-api-features-guideline](#)

[File an issue](#)

[Commit history](#)

[Pull requests](#)

This document is also available in these non-normative format: [pdf](#)



This document is licensed under

[Creative Commons Attribution 4.0 International Public License](#)

Abstract

This document is a guideline for Dutch data providers who want to use OGC-API-Features for download services in a way that they comply with relevant standards. The guideline has been written on the bases of the experiences gained from implementations of OGC-API-Features of INSPIRE datasets on test beds. A second important bases is formed by the results of an open tender, held in the beginning of 2023. This tender aimed at adjusting tools for serving OGC-API-feature services to 3 standards: OGC, Dutch Application Design Rules and INSPIRE. Until this tender, no tools were known that comply with all these standards. This guideline describes the requirements, as set out by the standards and uses the demoservices from this open tender as an example how one can comply with the standards. The aim of setting up these test beds and this guideline, has been to stimulate the Dutch data providers and hosting organizations to start publishing data as OGC API Features. By doing this, a greater goal is reached: A better use of geospatial data.

A general recommendation to all parties involved is to adjust as much as possible to the existing requirements as stated in this document. The most important recommendations for the hosting organizations is to stimulate data providers to start with OGC API features to increase the use of geospatial data according to standards. Data providers are recommended to first orientate on the possible work of other data providers in this field. A roadmap is described with the steps needed to be taken by data providers.

Status of this document

This is a working draft that can be changed, removed or replaced by other documents at any time. It is not a stable document.

Table of Contents

Abstract

Status of this document

1. OGC API Features explained

2. Roadmap for data providers

3. Requirements

3.1 OGC API Features Core

3.2 INSPIRE-MIF document: Setting up an INSPIRE Download service based on the OGC API-Features standard

3.3 Multilinguality

3.4 Predefined download

3.5 GeoJSON

3.6 Bulk download

3.7 CRS ETRS89 and WGS84

3.8 GML

3.9 Dutch API design rules

3.10 Describing encoding

3.11 Filtering

3.12 Metadata links

3.13 Relevant documentation

4. Examples of implementations after tooladjustments

4.1 Pygeoapi versus requirements

4.2 Geoserver versus requirements

4.3 Deegree versus requirements

4.4 General findings

4.5 Resulting documentation

A. References

A.1 Informative references

§ 1. OGC API Features explained

OGC API Features (OAPIF) is a multi-part standard that offers the capability to create, modify, and query spatial data on the Web and specifies requirements and recommendations for APIs that want to follow a standard way of sharing feature data. The specification is a multi-part document. [PUB-1], [PUB-5], [PUB-6].

OAPIF is also the term used for a feature download service by means of an API (Application Program Interface) based on OGC standards. OAPIF has been considered as follow up for the OGC WFS standard, but that does not mean it will replace it in the near future, although it might happen in a later future. They are complementary to each other. Where WFS is mainly known and used in the GIS community, the OAPIF is aiming at the non GIS-community, like web developers. OAPIF easier to use and needs less knowledge in the spatial domain. Note as well that WFS adopts the Geography Markup Language (GML) as a default data format. In contrast, OAPIF includes recommendations to support HTML and GeoJSON as encodings. Implementations of OAPIF may also optionally support GML.

An OAPIF consist of resources that can be retrieved by typing the corresponding path after the landing page of the OAPIF in a web browser or web application.

Resource	Path	Purpose
Landing page	/	This is the top-level resource, which serves as an entry point.
Conformance declaration	/conformance	This resource presents information about the functionality that is implemented by the server.
API definition	/api	This resource provides metadata about the API itself. Note use of /api on the server is optional and the API definition may be hosted on completely separate server
Feature collections	/collections	This resource lists the feature collections that are offered through the API.
Feature collection	/collections/{collectionId}	This resource describes the feature collection identified in the path.
Features	/collections/{collectionId}/items	This resource presents the features that are contained in the collection.
Feature	/collections/{collectionId}/items/{featureId}	This resource presents the feature that is identified in the path.

In the API definition, one can find all the supported encodings (HTML, JSON) and parameters that can be given along with URL like a bounding box or a limit of the amount of features. By default, an OAPIF

service will provide access to a single dataset. Rather than sharing the data as a complete dataset, the OGC API Features standards offer direct, fine-grained access to the data at the feature (object) level.

The best way of understanding the concept is looking at the examples that are discussed in the chapter of [examples](#).

Since providing a download service is an INSPIRE requirement when responsible for an INSPIRE dataset, the use of OAPIF can be considered for this purpose. It is even seen as an endorsed [Good Practice](#) within the INSPIRE community.

§ 2. Roadmap for data providers

The following steps could be considered to follow:

1. Read this document and all the documents in <https://geonovum.github.io/OAPIF-PDOK-INSPIRE/#references>
2. Make a choice between publishing by yourself or contact a hosting organization that can help you publish the OAPIF services.
3. Look what other data providers have done in this field for the concerning INSPIRE themes and have a look at these examples: <https://github.com/INSPIRE-MIF/gp-ogc-api-features/tree/master/deployments>.
4. If you decided to serve the OAPIF by yourself, make a choice for the best tooling, while regarding the CRS options (see next step).
5. Figure out the best way of supporting more than one CRS, and at least [WGS84](#) and [ETRS89](#) since the last is the most common in INSPIRE and mostly mandatory, but also provide the [Dutch RD](#). If tooling is chosen that is not able to serve more than one CRS, a second INSPIRE download option should be provided that does give the required CRS.
6. Decide on the best input encoding for the OAPIF. It depends on the previous steps.
7. Decide on the best output encoding, which also depends on the previous steps. The tooling used in the examples did a simple 1 to 1 mapping between the input and output encoding.
8. If a previously published mapping to an encoding other than GML cannot be found for the concerned INSPIRE theme, research the mapping of your harmonized data to this other encodings together with other INSPIRE data providers in Europe and use the principles as stated in [\[PUB-4\]](#). Consider leaving out the empty fields to reduce the output size or use an option not to show them.
9. Execute the transformation into the chosen input encoding. This can be done with software like [HALE studio](#) or [FME](#)
10. If not described before, describe the mapping from the INSPIRE data model to the output encoding of the OAPIF and publish it in order to be INSPIRE compliant.
11. Adjust your metadata of the dataset with the addition of extra OAPIF service. As long as there is no official protocol defined in <https://inspire.ec.europa.eu/metadata-codelist/ProtocolValue:1>, use the

extended code list for the protocol in the Dutch metadata standard 2.1.0

(<https://docs.geostandaarden.nl/md/mdprofiel-iso19119/#codelist-protocol>) contains: "OGC:API features".

12. If you host your OAPIF by yourself, research how to make INSPIRE compliant metadata of the OAPIF service. It is probably similar to the metadata of a WFS, except for the protocol element.
13. Add all the links as mentioned in the chapter on [requirements](#) (metadata of dataset, INSPIRE feature concept dictionary, Licence, mapping description, bulk download).
14. The steps for final actual publishing of the OAPIF service, depend on the chosen tool, so there the tooling guidelines need to be followed:
 - Geonovum testbed: <https://github.com/Geonovum/ogc-api-testbed/tree/main/docs/docs/howto>
 - GOAF: <https://github.com/PDOK/goaf>
 - Pygeoapi: https://github.com/Geonovum/ogc-api-testbed/blob/main/docs/docs/howto/howto_pygeoapi.md or <https://docs.pygeoapi.io/en/stable/pdf/>
 - Geoserver: https://github.com/Geonovum/ogc-api-testbed/blob/main/docs/docs/howto/howto_geoserver.md or <https://docs.geoserver.org/latest/en/user/>
15. Validate the OAPIF service with the [INSPIRE validation tool](#) and adjust where possible to be compliant.
16. When a working INSPIRE OAPIF is published with a well described mapping to the output encoding, it could be shared at <https://github.com/INSPIRE-MIF/2017.2/issues> or <https://github.com/Geonovum/OAPIF-PDOK-INSPIRE/issues>.

§ 3. Requirements

Below the most relevant requirements (or requirement classes) for setting up an INSPIRE OAPIF are listed:

nr	requirement	priority	reference
1	OGC API Features Core	1	[PUB-1]
2	INSPIRE-MIF document: Setting up an INSPIRE Download service based on the OGC API-Features standard	1	[PUB-2]
3	multilinguality	3	[PUB-2] #82-requirements-class-inspire-multilinguality-

nr	requirement	priority	reference
4	predefined download	1	[PUB-2] #req-pre-defined
5	GeoJSON	1	[PUB-2] #req-oapif-json
6	bulk download	1	[PUB-2] #req-bulk-download
7	CRS ETRS89 and WGS84	2	[PUB-5] and [PUB-2] #req-crs
8	INSPIRE validated GML as input and output	3	https://inspire.ec.europa.eu/validator/about/ and [PUB-1] #_requirements_classes_for_encodings
9	Dutch API design rules	1	[PUB-3]
10	describing encoding	1	[PUB-4]
11	filtering	2	[PUB-6]
12	metadata links	1	[PUB-1] #rec_core_fc-md-descriptions and [PUB-2] #metadata-elements-of-the-data-set

§ 3.1 OGC API Features Core

[OGC API Features Core](#), [PUB-1] describes the basic requirements (50) and recommendations (17) according to OGC that one needs to follow, independent of INSPIRE. It describes which paths can be used and what responses one should receive. It does not make the use of [OpenAPI Specification 3.0](#) mandatory, but if it is used, it gives an extra [requirement class](#).

There is a INSPIRE validation on the OGC standards for OAPIF available. It test on OGC requirements, but it does not test the requirements as stated in [PUB-2].

INSPIRE Reference Validator - Test selection

[Home](#) [Test selection](#) [Test reports](#) [Get support](#) [More on the INSPIRE Reference Validator](#)

Configure your test

Select the INSPIRE resource you would like to test

- ☐ Metadata
- ☐ View Service
- ☒ Download Service
- ☐ Discovery Service
- ☐ Data set

Select the Download Service type

- ☐ Web Feature Service (WFS)
- ☐ Pre-defined Atom
- ☐ Sensor Observation Service (SOS)
- ☐ Web Coverage Service (WCS)
- ☒ OGC API - Features

Figure 1 Validation on the OGC standards for OAPIF

§ 3.2 INSPIRE-MIF document: Setting up an INSPIRE Download service based on the OGC API-Features standard

[INSPIRE-MIF document: Setting up an INSPIRE Download service based on the OGC API-Features standard](#), [PUB-2] describes the specific INSPIRE requirements. Most of them are explained in the next chapters. This document does propose in [Note 2](#) to make it a mandatory requirement for INSPIRE to comply with [OAPIF requirements class OpenAPI 3.0](#).

§ 3.3 Multilinguality

The [multilinguality requirement class](#), [PUB-2] is mandatory for all data sets that contain information in more than one natural language. This is mostly not the case in the Netherlands, so it is of less importance.

§ 3.4 Predefined download

The [predefined download requirement class](#), [PUB-2] consists of 3 requirements for each collection to link to:

1. the metadata of the corresponding dataset
2. the corresponding entry in the [INSPIRE feature concept dictionary](#)
3. the license

§ 3.5 GeoJSON

The [GeoJSON requirement class](#) in [PUB-1] recommends to support GeoJSON for features with geometry, but as stated in https://docs.opengeospatial.org/is/17-069r4/17-069r4.html#_encodings, no encoding is mandatory. The [GeoJSON requirement class](#) in [PUB-2] also recommends to document how the GeoJSON encoding is retrieved from the INSPIRE data models.

§ 3.6 Bulk download

The [bulk download requirement class](#), [PUB-2] requires links for enclosure of the total data set and/or of each separate collection.

§ 3.7 CRS ETRS89 and WGS84

The [CRS requirement](#) in [PUB-1] requires [WGS84](#) for 2D-data and [WGS84h](#) for 3D-data as default. The [INSPIRE-CRS requirement class](#) in [PUB-2] requires also one of the INSPIRE CRS's based on ETRS89 to be supported. The [OGC API - Features - Part 2 standard](#), [PUB-5] prescribes how to support different coordinate systems with OAPIF.

For the Dutch data providers, it is recommended to also support [RD](#) for 2D data or [RD +NAP](#) for 3D data. See also: <https://docs.geostandaarden.nl/crs/crs>.

§ 3.8 GML

The use of GML as encoding for INSPIRE data can be considered in two ways. As input and as output.

When we consider the input, one would like to be able to use a source dataset of harmonized data. In most cases, this will be a GML encoded dataset. The GML encoding is at least needed to validate the data set with the [EU INSPIRE validator](#). Unfortunately, not many tooling for creating a OAPIF service is able to use GML as input. Especially when it concerns a complex GML dataset. So, a transformation to another encoding like GeoJSON is needed.

Output of GML from the OAPIF service can only be in simple features [level 0](#) and [level 2](#). So no complex features will be supported.

§ 3.9 Dutch API design rules

Dutch data providers are recommended to follow the [Dutch API design rules](#), [PUB-3]. This is not an INSPIRE requirement.

§ 3.10 Describing encoding

The standards considered in this guideline do not set a specific encoding as mandatory.

https://docs.opengeospatial.org/is/17-069r4/17-069r4.html#_encodings [PUB-1] Once another encoding than GML is used, data providers need to document how the encoding relates to the concerned INSPIRE data model. The good practice on the use of [geopackages](#) as encoding, describes how this describing could be done.

§ 3.11 Filtering

The specification for [filtering](#), [PUB-6] is still a draft version and has therefore not yet been taken into account. Some basic filtering requirements are described in [OGC API - Features - Part 1: Core](#) [PUB-1]. This only concerns filtering on a bounding box and on properties.

§ 3.12 Metadata links

The [requirement for metadata links to the data](#) in [PUB-2] have also been described in predefined download requirement class. The [requirement for metadata of the API](#) in [PUB-1] describes the metadata of the API via the API definition.

§ 3.13 Relevant documentation

Relevant documentation is shown in [appendix A](#).

§ 4. Examples of implementations after tooladjustments

Because from earlier test, Geonovum concluded tools were not compliant with the requirements, at the beginning of the year 2023, an open tender was set up to adjust the tooling. This resulted in 3 demo services that each show how you can be compliant with the requirements for OGC, INSPIRE and Dutch ADR. All demo services used the same selection of the Dutch INSPIRE Addresses in a GML file as input.

tool	main contributions	landing page
Pygeoapi	Justobject and Geocat	https://apitestbed.geonovum.nl/adr_pygeoapi/v1
Geoserver	Geosolutions	https://geonovum.geosolutionsgroup.com/geoserver/inspire/ogc/features/v1
Deegree	Wettransform	https://test.haleconnect.de/ogcapi/datasets/simplified-addresses/v1

Per tool, findings are elaborated in the next chapters when relevant.

§ 4.1 Pygeoapi versus requirements

The following findings show how Pygeoapi complies to the requirements.

RQ 1:OGC API Features Core

The OGC CITE validator gave no error at the landing page https://apitestbed.geonovum.nl/adr_pygeoapi/v1.

RQ 4: predefined download

Link to metadata of dataset: passed at /collections/AddressesNL and at /collections level:

```
{"href":  
"https://www.nationaalgeoregister.nl/geonetwork/srv/dut/catalog.search#/metad  
ata/a5f961e9-ebdd-41e2-b8e8-ab33ed340a83", "hreflang": "nl", "type":  
"text/html", "rel": "describedby", "title": "Metadata as HTML"}
```

```
{"href":  
"https://www.nationaalgeoregister.nl/geonetwork/srv/api/records/a5f961e9-  
ebdd-41e2-b8e8-ab33ed340a83/formatters/xml", "hreflang": "nl", "type":  
"application/xml", "rel": "describedby", "title": "Metadata as ISO 19139  
XML"}
```

Link to INSPIRE feature concept dictionary: passed at /collections/AddressesNL and at /collections level:

```
{"href": "https://inspire.ec.europa.eu/featureconcept/Address", "hreflang":  
"en", "type": "text/html", "rel": "tag", "title": "INSPIRE feature concept  
dictionary for addresses"}
```

Link to the license: passed at /collections/AddressesNL and at /collections level:

```
{"href": "https://creativecommons.org/publicdomain/zero/1.0/deed.en",  
"hreflang": "en", "type": "text/html", "rel": "license", "title": "CC0 1.0  
Public Domain license"}
```

See also https://apitestbed.geonovum.nl/adr_pygeoapi/v1/collections?f=json and
https://apitestbed.geonovum.nl/adr_pygeoapi/v1/collections/AddressesNL?f=json

RQ 5: GeoJSON

Items can be retrieved in Geo+Json by:

https://apitestbed.geonovum.nl/adr_pygeoapi/v1/collections/AddressesNL/items?f=json

RQ 6: bulk download

Link to bulkdownload of dataset: passed at /collections/AddressesNL and at /collections level

See also https://apitestbed.geonovum.nl/adr_pygeoapi/v1/collections?f=json and
https://apitestbed.geonovum.nl/adr_pygeoapi/v1/collections/AddressesNL?f=json {"href":
"https://service.pdok.nl/kadaster/ad/atom/downloads/addresses.gml.gz",
"hreflang": "nl", "length": 685450191, "type": "application/x-gmz", "rel":
"enclosure", "title": "Download complete dataset as GML"}

RQ 7: CRS ETRS89 and WGS84

For RD: [https://apitestbed.geonovum.nl/adr_pygeoapi/v1/collections/AddressesNL/items/1?
f=json&crs=http://www.opengis.net/def/crs/EPSG/0/28992](https://apitestbed.geonovum.nl/adr_pygeoapi/v1/collections/AddressesNL/items/1?f=json&crs=http://www.opengis.net/def/crs/EPSG/0/28992) For WGS84:

https://apitestbed.geonovum.nl/adr_pygeoapi/v1/collections/AddressesNL/items/1?f=json&crs=http://www.opengis.net/def/crs/OGC/1.3/CRS84 for ETRS89:
https://apitestbed.geonovum.nl/adr_pygeoapi/v1/collections/AddressesNL/items/1?f=json&crs=http://www.opengis.net/def/crs/EPSG/0/4258

The storagecrs can be found at the end of:

https://apitestbed.geonovum.nl/adr_pygeoapi/v1/collections/AddressesNL?f=json

With the following command line request, one can see the Content-CRS value in the header : curl -i

https://apitestbed.geonovum.nl/adr_pygeoapi/v1/collections/AddressesNL/items/1?f=json

An adjustment has been made tot the bbox filter. It now also supports the bbox-crs parameter. Only 2 addresses are available in the below defined bbox.

https://apitestbed.geonovum.nl/adr_pygeoapi/v1/collections/AddressesNL/items?f=json&bbox-crs=http://www.opengis.net/def/crs/EPSG/0/28992&bbox=252200,593000,252710,594000

RQ 8:GML

As input, a simple features GML file was used as produced by Wetransform from the complex feature GML with the transformation software Hale. As output, there is a link to the original complex feature GML-file: <https://service.pdok.nl/kadaster/ad/atom/downloads/addresses.gml.gz> Pygeoapi does not support GML-output at item-level, but this is not a requirement.

RQ 9:Dutch API design rules

It complies with all the rules, except for rule <https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-48>. This rule in the Dutch ADR prescribes that none of the API endpoints should have a trailing slash. On the other end, the OGC specification states that the landing page (i.e. "Home") should have a trailing slash. So the rules contradict. It is expected that in future, this ADR-rule will make an exception for the landingpage.

RQ 10:describing encoding

There is a link to <https://github.com/INSPIRE-MIF/2017.2/tree/master/GeoJSON/ads> at [colection/AddressesNL level](#):

```
{"href": "https://github.com/INSPIRE-MIF/2017.2/tree/master/GeoJSON/ads",  
"hreflang": "en", "type": "text/html", "rel": "about", "title": "Description  
of the encoding"}
```

RQ 11:filtering

For the use of filters, the bbox and items options were already available. Next to that, one can filter on the attributes which can be retrieved from:

https://apitestbed.geonovum.nl/adr_pygeoapi/v1/collections/AddressesNL/queryables. The specification for filtering [PUB-6] does not yet have the status "approved" and has not yet been considered.

RQ 12:metadata links

1. Metadata link of the dataset can be found at [/collections/AddressesNL level](#):

```
{"href":  
"https://www.nationaalgeoregister.nl/geonetwork/srv/dut/catalog.search#/m  
etadata/a5f961e9-ebdd-41e2-b8e8-ab33ed340a83", "hreflang": "nl" "type":  
"text/html", "rel": "describedby", "title": "Metadata as HTML"}
```

2. Metadata of the service can be obtained from:

https://apitestbed.geonovum.nl/adr_pygeoapi/v1/openapi?f=json and
https://apitestbed.geonovum.nl/adr_pygeoapi/v1/collections/AddressesNL.

Other findings

More information about the Pygeoapi adjustments to the standards can be found at
<https://pygeoapi.io/presentations/geonovum-tender-2023/>

§ 4.2 Geoserver versus requirements

The following findings show how Geoserver complies to the requirements.

RQ 1:OGC API Features Core

The OGC CITE validator gave no error at the landingpage

<https://geonovum.geosolutionsgroup.com/geoserver/inspire/ogc/features/v1>.

RQ 4:predefined download

Link to metadata of dataset: passed at [/collections/collection level](#) and at [/collections level](#).

```
{"href":"https://www.nationaalgeoregister.nl/geonetwork/srv/api/records/a5f96  
1e9-ebdd-41e2-b8e8-ab33ed340a83/formatters/xml?  
approved=true", "rel":"describedBy", "type":"application/xml", "title":"ISO  
metadata for this dataset"}
```

Link to INSPIRE feature concept dictionary: passed at [/collections/collection level](#).

```
{"href":"https://inspire.ec.europa.eu/featureconcept/Address", "rel":"tag", "ty  
pe":"text/html", "title":"INSPIRE Address feature concept."}
```

Link to the license: passed at [/collections level](#).

```
{"href":"http://creativecommons.org/publicdomain/zero/1.0/deed.nl", "rel":"lic  
ense", "type":"text/html", "title":"Dataset license."}
```

RQ 5:GeoJSON

Items can be retrieved in Geo+Json by:

<https://geonovum.geosolutionsgroup.com/geoserver/inspire/ogc/features/v1/collections/Addresses/items/1?f=application%2Fgeo%2Bjson>

RQ 6:bulk download

Link to bulkdownload of dataset: passed at [/collections/collection level](#).

```
{"href":"https://geonovum.geosolutionsgroup.com/geoserver/www/ADNL.gpkg","rel":"enclosure","type":"application/geopackage+sqlite3","title":"Addresses raw data."}
```

RQ 7:CRS ETRS89 and WGS84

For RD:

<https://geonovum.geosolutionsgroup.com/geoserver/inspire/ogc/features/v1/collections/Addresses/items/1?f=json&crs=http://www.opengis.net/def/crs/EPSG/0/28992> For WGS84:

<https://geonovum.geosolutionsgroup.com/geoserver/inspire/ogc/features/v1/collections/Addresses/items/1?f=json&crs=http://www.opengis.net/def/crs/OGC/1.3/CRS84> for ETRS89:

<https://geonovum.geosolutionsgroup.com/geoserver/inspire/ogc/features/v1/collections/Addresses/items/1?f=json&crs=http://www.opengis.net/def/crs/EPSG/0/4258>

The storagecrs can be found at the end of:

<https://geonovum.geosolutionsgroup.com/geoserver/inspire/ogc/features/v1/collections/Addresses?f=json>

With the following command line request, one can see the Content-CRS value in the header : curl -i

<https://geonovum.geosolutionsgroup.com/geoserver/inspire/ogc/features/v1/collections/Addresses/items/1?f=json>

An adjustment has been made tot the bbox filter. It now also supports the bbox-crs parameter. Only 2 addresses are available in the below defined bbox.

<https://geonovum.geosolutionsgroup.com/geoserver/inspire/ogc/features/v1/collections/Addresses/items?f=json&bbox-crs=http://www.opengis.net/def/crs/EPSG/0/28992&bbox=252200,593000,252710,594000>

RQ 8:GML

As input, a simple features GML file was used as produced by Wetransform from the complex feature GML with the transformation software Hale. As output, the following link can be found at [/collections/collection level](#). It can be used to download the first 50 records.

```
{"href":"https://geonovum.geosolutionsgroup.com/geoserver/inspire/ogc/feature s/v1/collections/Addresses/items?f=application%2Fgml%2Bxml%3Bversion%3D3.2","rel":"items","type":"application/gml+xml;version=3.2","title":"Addresses items as application/gml+xml;version=3.2"}
```

RQ 9:Dutch API design rules

It complies with all the rules, except for rule <https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-48>. This rule in the Dutch ADR prescribes that none of the API endpoints should have a trailing slash. On the other end, the OGC specification states that the landing page (i.e. "Home") should have a trailing

slash. So the rules contradict. It is expected that in future, this ADR-rule will make an exception for the landingpage.

RQ 10:describing encoding

There is a link to <https://github.com/INSPIRE-MIF/2017.2/blob/master/GeoJSON/ads/simple-addresses.md> at [/collections/collection level](#). {"href":"https://github.com/INSPIRE-MIF/2017.2/blob/master/GeoJSON/ads/simple-addresses.md","rel":"describedBy","type":"text/html","title":"GeoJSON Encoding Rule for INSPIRE Addresses"}

RQ 11:filtering

For the use of filters, the bbox and items options were already available. Next to that, one can filter on the attributes which can be retrieved from:

<https://geonovum.geosolutionsgroup.com/geoserver/inspire/ogc/features/v1/collections/Addresses?queryables>. The specification for filtering [PUB-6] does not yet have the status "approved" and has not yet been considered.

RQ 12:metadata links

1. Metadata link of the dataset can be found at [/collections/collection level](#):
{"href":"https://www.nationaalgeoregister.nl/geonetwork/srv/api/records/a5f961e9-ebdd-41e2-b8e8-ab33ed340a83/formatters/xml?approved=true","rel":"describedBy","type":"application/xml","title":"ISO metadata for this dataset"}
2. Metadata of the service can be obtained from:
<https://geonovum.geosolutionsgroup.com/geoserver/inspire/ogc/features/v1/openapi> and
<https://geonovum.geosolutionsgroup.com/geoserver/inspire/ogc/features/v1/collections/Addresses>.

Other findings

More information about the Geoserver adjustments to the standards can be found at <https://www.geonovum.nl/uploads/documents/Geosolutions.pdf>

§ 4.3 Deegree versus requirements

The following findings show how Geoserver complies to the requirements.

RQ 1:OGC API Features Core

The OGC CITE validator gave no error at the landingpage
<https://test.haleconnect.de/ogcapi/datasets/simplified-addresses/v1>.

RQ 4:predefined download

Link to metadata of dataset: passed at [/collections level](#):

```
{"href":"https://www.nationaalgeoregister.nl/geonetwork/srv/dut/catalog.search#/metadata/a5f961e9-ebdd-41e2-b8e8-ab33ed340a83","rel":"describedby","type":"text/html","title":"Metadata"}
```

Link to INSPIRE feature concept dictionary: passed at [/collections/collection level](#) and at [/collections level](#):

```
{"href":"https://inspire.ec.europa.eu/featureconcept/Address","rel":"tag","type":"text/html","title":"Feature concept Address"}
```

Link to the license: passed at [/collections level](#):

```
{"href":"http://creativecommons.org/publicdomain/zero/1.0/deed.nl","rel":"license","type":"text/html","title":"License"}
```

RQ 5:GeoJSON

Items can be retrieved in Geo+Json by:

<https://test.haleconnect.de/ogcapi/datasets/simplified-addresses/collections/SimpleAddress/items?f=json&limit=1> or <https://test.haleconnect.de/ogcapi/datasets/simplified-addresses/collections/SimpleAddress/items/nl-imbag-ad-address-0003200000133985?f=json>

RQ 6:bulk download

Link to bulkdownload of dataset: passed at [/collections/collection level](#) and at [/collections level](#).

```
{"href":"http://test.haleconnect.de/ogcapi/datasets/simplified-addresses/collections/SimpleAddress/items?bulk=true","rel":"enclosure","type":"application/xml","title":"Download all features as GML"}  
{"href":"http://test.haleconnect.de/ogcapi/datasets/simplified-addresses/collections/SimpleAddress/items?bulk=true","rel":"enclosure","type":"application/json","title":"Download all features as GeoJSON"}
```

RQ 7:CRS ETRS89 and WGS84

For RD: <https://test.haleconnect.de/ogcapi/datasets/simplified-addresses/v1/collections/SimpleAddress/items/nl-imbag-ad-address-0003200000133985?f=json&crs=http://www.opengis.net/def/crs/EPSG/0/28992>

For WGS84: <https://test.haleconnect.de/ogcapi/datasets/simplified-addresses/v1/collections/SimpleAddress/items/nl-imbag-ad-address-0003200000133985?f=json&crs=http://www.opengis.net/def/crs/OGC/1.3/CRS84>

For ETRS89: <https://test.haleconnect.de/ogcapi/datasets/simplified-addresses/v1/collections/SimpleAddress/items/nl-imbag-ad-address-0003200000133985?f=json&crs=http://www.opengis.net/def/crs/EPSG/0/4258>

For ETRS89: <https://test.haleconnect.de/ogcapi/datasets/simplified-addresses/v1/collections/SimpleAddress/items/nl-imbag-ad-address-0003200000133985?f=json&crs=http://www.opengis.net/def/crs/EPSG/0/4258>

For ETRS89: <https://test.haleconnect.de/ogcapi/datasets/simplified-addresses/v1/collections/SimpleAddress/items/nl-imbag-ad-address-0003200000133985?f=json&crs=http://www.opengis.net/def/crs/EPSG/0/4258>

The storagecrs can be found at the end of: <https://test.haleconnect.de/ogcapi/datasets/simplified-addresses/v1/collections/SimpleAddress?f=json>

With the following command line request, one can see the Content-CRS value in the header : `curl -i https://test.haleconnect.de/ogcapi/datasets/simplified-addresses/v1/collections/SimpleAddress/items?limit=1`

An adjustment has been made tot the bbox filter. It now also supports the bbox-crs parameter. Only 2 addresses are available in the below defined bbox. <https://test.haleconnect.de/ogcapi/datasets/simplified-addresses/v1/collections/SimpleAddress/items?f=json&bbox-crs=http://www.opengis.net/def/crs/EPSG/0/28992&bbox=252200,593000,252710,594000>

RQ 8:GML

As input, a simple features GML file was used as produced by Wetransform from the complex feature GML with the transformation software Hale. As output, the following links can be found at [/collections/collection level](#). They can be used to download specific records.

```
{"href":"http://test.haleconnect.de/ogcapi/datasets/simplified-addresses/collections/SimpleAddress/items?bulk=true","rel":"enclosure","type":"application/xml","title":"Download all features as GML"} or {"href":"http://test.haleconnect.de/ogcapi/datasets/simplified-addresses/collections/SimpleAddress/items","rel":"items","type":"application/gml+xml;version=3.2","title":"Features as GML"} (use parameter f=xml)
```

RQ 9:Dutch API design rules

It complies with all the rules, except for rule <https://publicatie.centrumvoorstandaarden.nl/api/adr/#api-48>. This rule in the Dutch ADR prescribes that none of the API endpoints should have a trailing slash. On the other end, the OGC specification states that the landing page (i.e. "Home") should have a trailing slash. So the rules contradict. It is expected that in future, this ADR-rule will make an exception for the landingpage.

RQ 10:describing encoding

There is a link to <https://github.com/INSPIRE-MIF/2017.2/blob/master/GeoJSON/ads> at [/collections/collection level](#). `{"href":"https://github.com/INSPIRE-MIF/2017.2/tree/master/GeoJSON/ads","rel":"describedby","type":"text/html","title":"Encoding description"}`

RQ 11:filtering

For the use of filters, the bbox and items options were already available. The specification for filtering [[PUB-6](#)] does not yet have the status "approved" and has not yet been considered.

RQ 12:metadata links

Metadata link of the dataset can be found at [/collections level](#):
`{"href":"https://www.nationaalgeoregister.nl/geonetwork/srv/dut/catalog.search`

```
h#/metadata/a5f961e9-ebdd-41e2-b8e8-ab33ed340a83", "rel": "describedby", "type": "text/html", "title": "Metadata"}
```

Metadata of the service can be obtained from: <https://test.haleconnect.de/ogcapi/datasets/simplified-addresses/v1/openapi> and <https://test.haleconnect.de/ogcapi/datasets/simplified-addresses/v1/collections/SimpleAddress>.

Other findings

More information about the Deegree adjustments to the standards can be found at <https://www.geonovum.nl/uploads/documents/deegree%20OGC%20API%20Features.pdf>

§ 4.4 General findings

1. There has been discussions whether the predefined downloadlinks (RQ4) should be at collections or collections/collection level? See also: <https://github.com/INSPIRE-MIF/gp-ogc-api-features/issues/91>. During the project we had the opinion that both should be possible, but afterwards we found out that it should be at collections level.
2. The protocol element in the metadata is based on a code list. A new protocol needs to be added to this list of [protocol values]<https://inspire.ec.europa.eu/metadata-codelist/ProtocolValue:1>). As long as it is not there, the Dutch profile for metadata can be used with the value: "OGC:API features" <https://geonovum.github.io/Metadata-ISO19119/#codelist-protocol>.
3. Another blocking issue before implementation of the OAPIF for INSPIRE is that descriptions of encodings other than GML are not yet available for most INSPIRE themes.
4. Complex GML as input and output are difficult as long as tooling (server and client) expects simple encodings.
5. One could discuss if it is useful to publish complex GML as output, because it is not in line with the aim of OGI API Features: easy to use for developers.
6. Complex GML as input needs a flattening of the data. This is needed for the software that publishes the features. It can only work with simple features, with one value per attribute and without relations to other objects. This is often not the case with the more complex INSPIRE models.

§ 4.5 Resulting documentation

Presentations can be found here: <https://www.geonovum.nl/over-geonovum/actueel/presentatie-resultaten-aanbesteding-ogc-api-features-toolaanpassing>

§ A. References

§ A.1 Informative references

[PUB-1]

OGC API Features Part1:Core. OGC. V1.0. URL: <https://docs.opengeospatial.org/is/17-069r4/17-069r4.html>

[PUB-2]

Setting up an INSPIRE Download service based on the OGC API-Features standard. INSPIRE-MIF. V1.0. URL: <https://github.com/INSPIRE-MIF/gp-ogc-api-features/blob/master/spec/oapif-inspire-download.md>

[PUB-3]

Dutch API design rules. <https://www.geonovum.nl>. 19 JULI 2020. URL: <https://www.geonovum.nl/over-geonovum/actueel/rest-api-design-rules-op-pas-toe-leg-uit-lijst>

[PUB-4]

INSPIRE UML-to-GeoJSON encoding rule. Working group on Inspire Action 2017.2. V0.1. URL: <https://github.com/INSPIRE-MIF/2017.2/blob/master/GeoJSON/geojson-encoding-rule.md#inspire-requirements-for-encoding-rules>

[PUB-5]

OGC API - Features - Part 2: Coordinate Reference Systems by Reference. OGC. V1.0. URL: <http://docs.opengeospatial.org/is/18-058r1/18-058r1.html>

[PUB-6]

OGC API - Features - Part 3: Filtering and the Common Query Language (CQL). OGC. V1.0.0, draft. URL: <https://docs.ogc.org/DRAFTS/19-079r1.html>