



**International
Standard**

ISO 19168-1

**Geographic information —
Geospatial API for features —**

**Part 1:
Core**

*Information géographique — API géospatiale pour les entités —
Partie 1: Profil minimal*

**Second edition
2025-01**

iTeh Standards
(<https://standards.iteh.ai>)
Document Preview

[ISO 19168-1:2025](https://standards.iteh.ai/catalog/standards/iso/aa52c999-a781-4816-8bcb-92eef4422af4/iso-19168-1-2025)

<https://standards.iteh.ai/catalog/standards/iso/aa52c999-a781-4816-8bcb-92eef4422af4/iso-19168-1-2025>

iTeh Standards
(<https://standards.itih.ai>)
Document Preview

[ISO 19168-1:2025](https://standards.itih.ai/catalog/standards/iso/aa52c999-a781-4816-8bcb-92eef4422af4/iso-19168-1-2025)

<https://standards.itih.ai/catalog/standards/iso/aa52c999-a781-4816-8bcb-92eef4422af4/iso-19168-1-2025>



COPYRIGHT PROTECTED DOCUMENT

© ISO 2025

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

	Page
Foreword	v
Introduction	vii
1 Scope	1
2 Normative references	1
3 Terms and definitions	2
3.1 Terms and definitions.....	2
3.2 Abbreviated terms.....	3
4 Conformance	3
5 Conventions	4
5.1 Identifiers.....	4
5.2 Link relations.....	4
5.3 Use of HTTPS.....	5
5.4 HTTP URIs.....	5
5.5 API definition.....	5
5.5.1 General remarks.....	5
5.5.2 Role of OpenAPI.....	5
5.5.3 References to OpenAPI components in normative statements.....	6
5.5.4 Paths in OpenAPI definitions.....	6
5.5.5 Reusable OpenAPI components.....	6
6 Overview	7
6.1 Design considerations.....	7
6.2 Encodings.....	7
6.3 Examples.....	8
7 Requirements class "Core"	9
7.1 Overview.....	9
7.2 API landing page.....	10
7.2.1 Operation.....	10
7.2.2 Response.....	10
7.2.3 Error situations.....	11
7.3 API definition.....	11
7.3.1 Operation.....	11
7.3.2 Response.....	12
7.3.3 Error situations.....	12
7.4 Declaration of conformance classes.....	12
7.4.1 Operation.....	12
7.4.2 Response.....	13
7.4.3 Error situations.....	13
7.5 HTTP 1.1.....	13
7.5.1 HTTP status codes.....	13
7.6 Unknown or invalid query parameters.....	14
7.7 Web caching.....	15
7.8 Support for cross-origin requests.....	15
7.9 Encodings.....	15
7.10 String internationalization.....	16
7.11 Coordinate reference systems.....	16
7.12 Link headers.....	17
7.13 Feature collections.....	17
7.13.1 Operation.....	17
7.13.2 Response.....	17
7.13.3 Error situations.....	23
7.14 Feature collection.....	24
7.14.1 Operation.....	24

ISO 19168-1:2025(en)

7.14.2	Response	24
7.14.3	Error situations	24
7.15	Features	24
7.15.1	Operation	24
7.15.2	Parameter limit	25
7.15.3	Parameter bbox	26
7.15.4	Parameter datetime	27
7.15.5	Parameters for filtering on feature properties	29
7.15.6	Combinations of filter parameters	29
7.15.7	Response	30
7.15.8	Error situations	32
7.16	Feature	32
7.16.1	Operation	32
7.16.2	Response	33
7.16.3	Error situations	33
8	Requirements classes for encodings	33
8.1	Overview	33
8.2	Requirements class "HTML"	34
8.3	Requirements class "GeoJSON"	34
8.4	Requirements class "Geography Markup Language (GML), Simple Features Profile, Level 0"	36
8.5	Requirements class "Geography Markup Language (GML), Simple Features Profile, Level 2"	38
9	Requirements class "OpenAPI 3.0"	39
9.1	Basic requirements	39
9.2	Complete definition	40
9.3	Exceptions	40
9.4	Security	40
9.5	Features	40
10	Media types	41
11	Security Considerations	41
11.1	General	41
11.2	Multiple access routes	42
11.3	Multiple servers	42
11.4	Path manipulation on GET	42
11.5	Path manipulation on PUT and POST	42
Annex A (normative) Abstract test suite		43
Bibliography		59

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents. ISO shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 211, *Geographic information/geomatics*, in collaboration with the European Committee for Standardization (CEN) Technical Committee CEN/TC 287 *Geographic Information*, in accordance with the Agreement on technical cooperation between ISO and CEN (Vienna Agreement), and in collaboration with the Open Geospatial Consortium (OGC).

This second edition cancels and replaces the first edition (ISO 19168-1:2020), which has been technically revised.

<https://standards.iteh.ai/catalog/standards/iso/aa52c999-a781-4816-8bcb-92ecf4422af4/iso-19168-1-2025>
The main changes are as follows:

- the link schema has been updated to make the "rel" attribute required to align with IETF RFC 8288;
- the bounding box schemas have been updated to require 4 or 6 numbers (2D or 3D);
- the XML Schema core.xsd has been aligned with the corresponding schema for the JSON representation;
- normative references have been updated to reference newer editions (HTTP and OpenAPI);
- the definition of "dataset" has been updated;
- the definitions of the terms "landing page" and "OGC Web API" have been added;
- the IANA link relation type has been corrected to "describedby", rather than "describedBy";
- requirement /req/core/fc-limit-response-1 has been updated to clarify the behaviour if the value of the "limit" parameter is higher than the maximum value;
- recommendation /rec/core/fc-extent has been added to clarify that the bounding box of a feature collection response should be the bounding box of a matched feature, not only the features in the current page;
- recommendations /rec/core/fc-md-self-links and /rec/core/sfc-md-links have been added to clarify that "self" links should be added;
- the value of the "profile" attribute in the GML media type has been modified to be in quotation marks;

ISO 19168-1:2025(en)

- a new requirement /req/core/fc-md-extent-multi has been added to clarify that the first bounding box in a collection extent array contains all other bounding boxes in the array;
- the use of the attributes "spatial"/"temporal" in a collection extent has been clarified;
- it has been clarified that the "itemType" attribute should be included for each collection;
- the interpretation of a degenerated bounding box in the "bbox" parameter has been clarified;
- it has been clarified that a "next" link can return no additional features;
- it has been clarified that the feature identifier is mapped to the "id" attribute in GeoJSON and "@gml:id" in GML;
- missing test cases have been added;
- some specification URIs have been updated;
- various editorial corrections and updates have been applied in the document.

NOTE For more details on the changes listed, see the OGC release notes.[\[13\]](#)

A list of all parts in the ISO 19168 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

iTeh Standards (<https://standards.iteh.ai>) Document Preview

[ISO 19168-1:2025](#)

<https://standards.iteh.ai/catalog/standards/iso/aa52c999-a781-4816-8bcb-92eef4422af4/iso-19168-1-2025>

Introduction

OGC API standards^[10] define modular API building blocks to spatially enable Web APIs in a consistent way. The OpenAPI specification is used to define the API building blocks.

ISO has published a subset of the OGC API family of standards. To reflect that only a subset of the OGC API standards will be published by ISO and to avoid using organization names in the titles of ISO standards, standards from the "OGC API" series are published by ISO as "Geospatial API." For example, the title of this document in OGC is "OGC API - Features - Part 1:Core" and the title in ISO is "Geographic Information — Geospatial API for Features — Part 1: Core."

For simplicity, this document consistently uses:

- "OGC API" to refer to the family of standards for geospatial Web APIs that in ISO is published as "Geospatial API";
- "OGC API - Features" to refer to the multipart standard for features of which certain parts are published by ISO as the ISO 19168 series/"Geographic Information — Geospatial API for Features"; and
- "this document" to refer to "OGC API - Features - Part 1: Core", which is published by ISO as ISO 19168-1/"Geographic Information — Geospatial API for Features — Part 1: Core".

OGC API is organized by resource type. OGC API - Features specifies the fundamental API building blocks for interacting with features. The spatial data community uses the term "feature" for things in the real world that are of interest.

NOTE For those not familiar with the term "feature," the explanations on Spatial Things, Features and Geometry in the W3C/OGC Spatial Data on the Web Best Practice document^[7] provide more detail.

OGC API - Features provides API building blocks to create, modify and query features on the Web. The series is comprised of multiple parts, each of them a separate standard. This document (ISO 19168-1), which corresponds to one such part, the "Core", specifies the core capabilities and is restricted to fetching features where geometries are represented in the coordinate reference system (CRS) WGS 84 with axis order longitude/latitude. Additional capabilities that address more advanced needs will be specified in additional parts. Examples include support for creating and modifying features, more complex data models, richer queries, additional CRS, multiple datasets and collection hierarchies.

By default, every API implementing this document will provide access to a single dataset. Rather than sharing the data as a complete dataset, OGC API - Features offers direct, fine-grained access to the data at the feature (object) level.

The API building blocks specified in this document are consistent with the architecture of the Web. In particular, the API design is guided by the IETF HTTP/HTTPS RFCs, the W3C Data on the Web Best Practices,^[8] the W3C/OGC Spatial Data on the Web Best Practices,^[7] and the emerging OGC Web API Guidelines. A particular example is the use of the concepts of datasets and dataset distributions as defined in DCAT^[9] and used in schema.org.

This document specifies discovery and query operations that are implemented using the HTTP GET method. Support for additional methods (in particular POST, PUT, DELETE, PATCH) is specified in additional parts.

Discovery operations enable clients to interrogate the API, including the API definition and metadata about the feature collections provided by the API, to determine the capabilities of the API and retrieve information about available distributions of the dataset.

Query operations enable clients to retrieve features from the underlying data store based upon simple selection criteria, defined by the client.

This document defines the resources listed in [Table 1](#). For an overview of the resources, see [7.1](#).

Table 1 — Overview of resources, applicable HTTP methods and links to the document sections

Resource	Path	HTTP method	Subclause
Landing page	/	GET	7.2 API landing page
Conformance declaration	/conformance	GET	7.4 Declaration of conformance classes
Feature collections	/collections	GET	7.13 Feature collections
Feature collection	/collections/{collectionId}	GET	7.14 Feature collection
Features	/collections/{collectionId}/items	GET	7.15 Features
Feature	/collections/{collectionId}/items/{featureId}	GET	7.16 Feature

iTeh Standards
(<https://standards.iteh.ai>)
Document Preview

[ISO 19168-1:2025](#)

<https://standards.iteh.ai/catalog/standards/iso/aa52c999-a781-4816-8bcb-92eef4422af4/iso-19168-1-2025>

Geographic information — Geospatial API for features —

Part 1: Core

1 Scope

This document specifies the behaviour of Web APIs that provide access to features in a dataset independently of the underlying data store. This document defines discovery and query operations.

Discovery operations enable clients to interrogate the API, including the API definition and metadata about the feature collections provided by the API, to determine the capabilities of the API and retrieve information about available distributions of the dataset.

Query operations enable clients to retrieve features from the underlying data store based upon simple selection criteria, defined by the client.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

OPENAPI INITIATIVE (OAI), OpenAPI Specification 3.0 [online]. 2020 [viewed 2020-03-16]. The latest patch version at the time of publication of this standard was 3.0.3, available at <https://spec.openapis.org/oas/v3.0.3>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 2818: HTTP Over TLS [online]. Edited by E. Rescorla. 2000 [viewed 2020-03-16]. Available at <https://www.rfc-editor.org/rfc/rfc2818.html>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 3339: Date and Time on the Internet: Timestamps [online]. Edited by G. Klyne, C. Newman. 2002 [viewed 2020-03-16]. Available at <https://www.rfc-editor.org/rfc/rfc3339.html>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 7230 to RFC 7235: HTTP/1.1 [online]. Edited by R. Fielding, J. Reschke, Y. Lafon, M. Nottingham. 2014 [viewed 2020-04-28]. Available at <https://www.rfc-editor.org/rfc/rfc7230.html>, <https://www.rfc-editor.org/rfc/rfc7231.html>, <https://www.rfc-editor.org/rfc/rfc7232.html>, <https://www.rfc-editor.org/rfc/rfc7233.html>, <https://www.rfc-editor.org/rfc/rfc7234.html>, and <https://www.rfc-editor.org/rfc/rfc7235.html>

INTERNET ENGINEERING TASK FORCE (IETF), RFC 8288: Web Linking [online]. Edited by M. Nottingham. 2017 [viewed 2020-03-16]. Available at <https://www.rfc-editor.org/rfc/rfc8288.html>

OPEN GEOSPATIAL CONSORTIUM (OGC), OGC 10-100r3: Geography Markup Language (GML) Simple Features Profile [online]. Edited by L. van den Brink, C. Portele, P. Vretanos. 2012 [viewed 2020-03-16]. Available at http://portal.opengeospatial.org/files/?artifact_id=42729

INTERNET ENGINEERING TASK FORCE (IETF). RFC 7946: The GeoJSON Format [online]. Edited by H. Butler, M. Daly, A. Doyle, S. Gillies, S. Hagen, T. Schaub. 2016 [viewed 2020-03-16]. Available at <https://www.rfc-editor.org/rfc/rfc7946.html>

WHATWG. HTML, Living Standard [online, viewed 2020-03-16]. Available at <https://html.spec.whatwg.org/>

SCHEMA.ORG. Schema.org [online, viewed 2020-03-16]. Available at <https://schema.org/docs/schemas.html>

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

3.1 Terms and definitions

3.1.1

dataset

collection of data

Note 1 to entry: Published or curated by a single agent, and available for access or download in one or more serializations or formats.

Note 2 to entry: The use of "collection" in this definition is broader than the use of the term collection throughout the rest of this document. See the definition of "feature collection."

[SOURCE: DCAT,^[9] 6.6, modified — Definition has been split into definition and Note 1 to entry; Note 2 to entry has been added.]

3.1.2

distribution

specific representation of a *dataset* (3.1.1)

EXAMPLE A downloadable file, an RSS feed or an API.

[SOURCE: DCAT,^[9] 6.7, modified — Definition has been shortened.]

3.1.3

feature

abstraction of real-world phenomena

Note 1 to entry: Further details about the term "feature" can be found in Reference [7].

[SOURCE: ISO 19101-1:2014, 4.1.11, modified — Note 1 to entry has been added.]

3.1.4

feature collection

collection

set of *features* (3.1.3) from a *dataset* (3.1.1)

3.1.5

resource

entity that might be identified

Note 1 to entry: The term "resource", when used in the context of an OGC API standard, means a *web resource* (3.1.7) unless otherwise indicated.

[SOURCE: ISO 15836-2:2019, 3.1.10, modified — Notes 1 and 2 have been removed and replaced with a new Note 1 to entry.]

3.1.6

Web API

API using an architectural style that is founded on the technologies of the Web

Note 1 to entry: See Reference [8] for further detail.

Note 2 to entry: Definition adapted from Reference [8], 8.10.1. Modified by being rephrased for clarity.

3.1.7

web resource

resource (3.1.5) that is identified by a HTTP URI

3.2 Abbreviated terms

API	application programming interface
CORS	cross-origin resource sharing
CRS	coordinate reference system
HTTP	hypertext transfer protocol
HTTPS	hypertext transfer protocol secure
IANA	Internet Assigned Numbers Authority
IRI	internationalized resource identifier
OGC	Open Geospatial Consortium
RFC	request for comment
TRS	temporal coordinate reference system
URI	uniform resource identifier
WSDL	web service description language
YAML	YAML Ain't Markup Language

4 Conformance

This document defines six requirements/conformance classes.

The standardization targets of all conformance classes are "Web APIs."

The main requirements class is:

- Core.

The Core requirements class specifies requirements that all Web APIs have to implement.

The Core requirements class does not mandate a specific encoding or format for representing features or feature collections. Four requirements classes depend on the Core requirements class and specify representations for these resources in commonly used encodings for spatial data on the Web:

- HTML,
- GeoJSON,
- Geography Markup Language (GML), Simple Features Profile, Level 0, and
- Geography Markup Language (GML), Simple Features Profile, Level 2.

None of these encodings are mandatory and an implementation of the Core requirements class can also decide to implement none of them, but to implement another encoding instead.

That said, the Core requirements class includes recommendations to support, where practical, HTML and GeoJSON as encodings. [Clause 6](#) includes a discussion about the recommended encodings.

ISO 19168-1:2025(en)

The Core requirements class does not mandate any encoding or format for the formal definition of the API. One option is the OpenAPI 3.0 specification and a requirements class has been specified for OpenAPI 3.0, which depends on the Core requirements class:

- OpenAPI Specification 3.0.

As with the feature encodings, an implementation of the Core requirements class can decide to use other API definition representations in addition or instead of an OpenAPI 3.0 definition. Examples for alternative API definitions include: OpenAPI 2.0 (Swagger), future versions of the OpenAPI specification, an OWS Common 2.0 capabilities document or WSDL.

The Core requirements class is intended to be a minimal useful API for fine-grained read-access to a spatial dataset where geometries are represented in the CRS WGS 84 with axis order longitude/latitude.

Additional capabilities such as support for transactions, complex data structures, rich queries, other CRS, subscription/notification, returning aggregated results, etc. can be specified in future parts of OGC API - Features or as vendor-specific extensions.

Conformance with this document shall be checked using all the relevant tests specified in [Annex A](#) of this document. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing web site.

Table 2 — Conformance class URIs

Conformance class	URI
Core	http://www.opengis.net/spec/ogcapi-features-1/1.0/conf/core
HTML	http://www.opengis.net/spec/ogcapi-features-1/1.0/conf/html
GeoJSON	http://www.opengis.net/spec/ogcapi-features-1/1.0/conf/geojson
GML, Simple Features Profile, Level 0	http://www.opengis.net/spec/ogcapi-features-1/1.0/conf/gmlsf0
GML, Simple Features Profile, Level 2	http://www.opengis.net/spec/ogcapi-features-1/1.0/conf/gmlsf2
OpenAPI Specification 3.0	http://www.opengis.net/spec/ogcapi-features-1/1.0/conf/oas30

5 Conventions

5.1 Identifiers

The normative provisions in this document are denoted by the URI:

<http://www.opengis.net/spec/ogcapi-features-1/1.0>

All requirements and conformance tests that appear in this document are denoted by partial URIs which are relative to this base.

5.2 Link relations

To express relationships between resources, RFC 8288 (Web Linking) is used.

The following registered link relation types^[3] are used in this document.

- `alternate`: Refers to a substitute for this context.
- `collection`: The target IRI points to a resource which represents the collection resource for the context IRI.
- `describedby`: Refers to a resource providing information about the link's context.
- `item`: The target IRI points to a resource that is a member of the collection represented by the context IRI.
- `next`: Indicates that the link's context is a part of a series, and that the next in the series is the link target.

- licence: Refers to a licence associated with this context.
- prev: Indicates that the link's context is a part of a series, and that the previous in the series is the link target.
 - This relation is only used in examples.
- self: Conveys an identifier for the link's context.
- service-desc: Identifies service description for the context that is primarily intended for consumption by machines.
 - API definitions are considered service descriptions.
- service-doc: Identifies service documentation for the context that is primarily intended for human consumption.

In addition, the following link relation types are used for which no applicable registered link relation type could be identified.

- items: Refers to a resource that is comprised of members of the collection represented by the link's context.
- conformance: Refers to a resource that identifies the specifications to which the link's context conforms.
- data: Refers to the root resource of a dataset in an API.

Each resource representation includes an array of links. Implementations are free to add additional links for all resources provided by the API. For example, an "enclosure" link could reference a bulk download of a collection. Or a "related" link on a feature could reference a related feature.

5.3 Use of HTTPS

For simplicity, this document in general only refers to the HTTP protocol. This is not meant to exclude the use of HTTPS and is simply a shorthand notation for "HTTP or HTTPS." In fact, most servers are expected to use HTTPS, not HTTP.

5.4 HTTP URIs

This document does not restrict the lexical space of URIs used in the API beyond the requirements of the HTTP and URI Syntax IETF RFCs. If URIs include reserved characters that are delimiters in the URI subcomponent, these have to be percent-encoded. See RFC 3986, Clause 2^[2] for details.

5.5 API definition

5.5.1 General remarks

Good documentation is essential for every API so that developers can more easily learn how to use the API. Ideally, documentation will be available in HTML and in a format that can be processed by software to connect to the API.

This document specifies requirements and recommendations for APIs that share feature data and that want to follow a standard way of doing so. In general, APIs will go beyond the requirements and recommendations stated in this document (or other parts of OGC API) and will support additional operations, parameters, etc. that are specific to the API or the software tool used to implement the API.

5.5.2 Role of OpenAPI

This document uses OpenAPI 3.0 fragments as examples and to formally state requirements. However, using OpenAPI 3.0 is not required for implementing a server.

Therefore, the Core requirements class only requires that an API definition is provided and linked from the landing page.

A separate requirements class is specified for API definitions that follow the OpenAPI specification 3.0. This does not preclude that in the future or in parallel other versions of OpenAPI or other API descriptions are provided by a server.

NOTE This approach is used to avoid lock-in to a specific approach to defining an API, as it is expected that the API landscape will continue to evolve.

In this document, fragments of OpenAPI definitions are shown in YAML (YAML Ain't Markup Language)^[1] since YAML is easier to read than JSON and is typically used in OpenAPI editors. YAML is described by its authors as a human-friendly data serialization standard for all programming languages.

5.5.3 References to OpenAPI components in normative statements

Some normative statements (requirements, recommendations, and permissions) use a phrase that a component in the API definition of the server has to be "based upon" a schema or parameter component in the OGC schema repository.

In such a case, the following changes to the pre-defined OpenAPI component are permitted.

- If the server supports an XML encoding, XML properties may be added to the relevant OpenAPI schema components.
- The range of values of a parameter or property may be extended (additional values) or constrained (if a subset of all possible values are applicable to the server). An example for a constrained range of values is to explicitly specify the supported values of a string parameter or property using an enum.
- The default value of a parameter may be changed or added unless a requirement explicitly prohibits this.
- Additional properties may be added to the schema definition of a Response Object.
- Informative text may be changed or added, like comments or description properties.

For API definitions that do not conform to the OpenAPI Specification 3.0, the normative statement has to be interpreted in the context of the API definition language used.

5.5.4 Paths in OpenAPI definitions

All paths in an OpenAPI definition are relative to a base URL of the server.

EXAMPLE 1 URL of the OpenAPI definition

If the OpenAPI Server Object looks like this:

```
servers:  
  - url: https://dev.example.org/  
    description: Development server  
  - url: https://data.example.org/  
    description: Production server
```

The path "/mypath" in the OpenAPI definition of a Web API would be the URL <https://data.example.org/mypath> for the production server.

5.5.5 Reusable OpenAPI components

Reusable components for OpenAPI definitions for implementations of OGC API - Features are referenced from this document.