DRAFT INTERNATIONAL STANDARD
ISO/DIS 19168-1

ISO/TC **211** Secretariat: **SIS**

Voting begins on:
2019-10-07

Voting terminates on:
2019-12-30

Geographic information — Geospatial API for features —

Part 1:
Core

ICS: 35.240.70

THIS DOCUMENT IS A DRAFT CIRCULATED FOR COMMENT AND APPROVAL. IT IS THEREFORE SUBJECT TO CHANGE AND MAY NOT BE REFERRED TO AS AN INTERNATIONAL STANDARD UNTIL PUBLISHED AS SUCH.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

RECIPIENTS OF THIS DRAFT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

This document is circulated as received from the committee secretariat.

Reference number
ISO/DIS 19168-1:2019(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

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 documents 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).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

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.

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.

Introduction

OGC API standards 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.

The OGC API family of standards is organized by resource type. This standard 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.

If you are unfamiliar with the term 'feature', the explanations on Spatial Things, Features and Geometry in the W3C/OGC Spatial Data on the Web Best Practice document provide more detail.

OGC API Features provides API building blocks to create, modify and query features on the Web. OGC API Features is comprised of multiple parts, each of them is a separate standard. This part, the "Core", specifies the core capabilities and is restricted to fetching features where geometries are represented in the coordinate reference system 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 coordinate reference systems, multiple datasets and collection hierarchies.

By default, every API implementing this standard 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 API building blocks specified in this standard 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, the W3C/OGC Spatial Data on the Web Best Practices 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 and used in schema.org.

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

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

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

A subset of the OGC API family of standards is expected to be published by ISO. For example, this document is in the process to be published by ISO as ISO 19168-1. 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". That is, 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 that in ISO is published as ISO 19168 / "Geographic Information - Geospatial API for Features";
- "OGC API - Features - Part 1: Core" to refer to this document that in ISO is published as ISO 19168-1 / "Geographic Information - Geospatial API for Features - Part 1: Core".

This standard defines the resources listed in Table 1. For an overview of the resources, see section 7.1 Overview.

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

Resource	Path	HTTP method	Document reference
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

Implementations of OGC API Features are intended to support two different approaches how clients can use the API.

In the first approach, clients are implemented with knowledge about this standard and its resource types. The clients navigate the resources based on this knowledge and based on the responses provided by the API. The API definition may be used to determine details, e.g. on filter parameters, but this may not be necessary depending on the needs of the client. These are clients that are in general able to use multiple APIs as long as they implement OGC API Features.

The other approach targets developers that are not familiar with the OGC API standards, but want to interact with spatial data provided by an API that happens to implement OGC API Features. In this case the developer will study and use the API definition - typically an OpenAPI document - to understand the API and implement the code to interact with the API. This assumes familiarity with the API definition language and the related tooling, but it should not be necessary to study the OGC API standards.

Geographic Information – Geospatial API for Features – Part 1: Core

1 Scope

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

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

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.

Open API Initiative: **OpenAPI Specification 3.0.2**, https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.2.md

Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T.: IETF RFC 2616, **HTTP/1.1**, http://tools.ietf.org/rfc/rfc2616.txt

Rescorla, E.: IETF RFC 2818, **HTTP Over TLS**, http://tools.ietf.org/rfc/rfc2818.txt

Klyne, G., Newman, C.: IETF RFC 3339, **Date and Time on the Internet: Timestamps**, http://tools.ietf.org/rfc/rfc3339.txt

Nottingham, M.: IETF RFC 8288, **Web Linking**, http://tools.ietf.org/rfc/rfc8288.txt

van den Brink, L., Portele, C., Vretanos, P.: OGC 10-100r3, **Geography Markup Language (GML) Simple Features Profile**, http://portal.opengeospatial.org/files/?artifact_id=42729

Butler, H., Daly, M., Doyle, A., Gillies, S., Hagen, S., Schaub, T.: IETF RFC 7946, **The GeoJSON Format**, https://tools.ietf.org/rfc/rfc7946.txt

W3C: **HTML5**, W3C Recommendation, http://www.w3.org/TR/html5/

Schema.org: http://schema.org/docs/schemas.html

3 Terms and definitions

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

3.1
dataset
collection of data, published or curated by a single agent, and available for access or download in one or more formats

Note 1 to entry: The use of 'collection' in the definition from [DCAT] is broader than the use of the term collection in this specification. See the definition of 'feature collection.'

[SOURCE: DCAT, 5.3, modified — Note 1 to entry has been added]

3.2
distribution
represents an accessible form of a **dataset**

Note 1 to entry: EXAMPLE: a downloadable file, an RSS feed or an API.

[SOURCE: DCAT, 5.4, modified — sentence split into definition and Note 1 to entry]

3.3
feature
abstraction of real world phenomena

Note 1 to entry: If you are unfamiliar with the term 'feature', the explanations on Spatial Things, Features and Geometry in the W3C/OGC Spatial Data on the Web Best Practice document provide more detail.

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

3.4
feature collection
collection
a set of **features** from a **dataset**

Note 1 to entry: In this Standard, 'collection' is used as a synonym for 'feature collection'. This is done to make, for example, URI path expressions shorter and easier to understand for those that are not geo-experts.

3.5
Web API
API using an architectural style that is founded on the technologies of the Web

Note 1 to entry: Best Practice 24: Use Web Standards as the foundation of APIs in the W3C Data on the Web Best Practices provides more detail.

[SOURCE: DWBP, 8.10.1, modified – definition extracted from sentence]

4 Conformance

This standard defines six requirements / conformance classes.

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

The main requirements class is:

- Core.

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

The *Core* does not mandate a specific encoding or format for representing features or feature collections. Four requirements classes depend on the *Core* 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* may 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 (Overview) includes a discussion about the recommended encodings.

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

- OpenAPI specification 3.0.

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

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

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

Conformance with this standard shall be checked using all the relevant tests specified in Annex A (normative) 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.

5 Conventions

5.1 Identifiers

The normative provisions in this standard 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 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.
- **license**: Refers to a license 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 that the link's context conforms to.
- **data**: Indicates that the link's context is a distribution of a dataset that is an API and refers to the root resource of the dataset in the 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 simply is 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 Clause 2 of RFC 3986 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. In the best case, documentation will be available in HTML and in a format that can be processed by software to connect to the API.

This standard 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 standard - or other parts of the OGC API family of standards - 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) 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 must be "based upon" a schema or parameter component in the OGC schema repository.

In this 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.