OGC® DOCUMENT: 25-014

External identifier of this OGC® document: http://www.opengis.net/doc/spec/ogcapi-edr-3/1.0

OGC API - ENVIRONMENTAL DATA RETRIEVAL STANDARD - PART 3: PROFILES

STANDARD
Implementation

DRAFT

Version: 1.0
Submission Date: 2025-06-02
Approval Date: 2029-03-30
Publication Date: 2029-03-30
Editor: Mark Burgoyne, Charles Heazel

Notice for Drafts: This document is not an OGC Standard. This document is distributed for review and comment. This document is subject to change without notice and may not be referred to as an OGC Standard.
Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

License Agreement

Suggested additions, changes and comments on this document are welcome and encouraged. Such suggestions may be submitted using the online change request form on OGC web site: http://ogc.standardstracker.org/

Copyright notice

Note

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

CONTENTS

LIST OF TABLES

LIST OF RECOMMENDATIONS

I ABSTRACT

<Insert Abstract Text here>

II KEYWORDS

The following are keywords to be used by search engines and document catalogues.

ogcdoc, OGC document, API, openapi, html

III PREFACE

The aim of an OGC API EDR profile is to ensure interoperability between API implementations. To achieve this, it is essential that providers use a consistent approach when defining collections and instances of collections. An OGC EDR profile will specify a set of requirements that an API implementation must support to be a compliant implementation.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

IV SECURITY CONSIDERATIONS

No security considerations have been made for this Standard.

V SUBMITTING ORGANIZATIONS

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

- UK Met Office
- Heazeltech

VI SUBMITTERS

All questions regarding this submission should be directed to the editor or the submitters:

Name	Affiliation

VII CONTRIBUTORS

Additional contributors to this Standard include the following:

Individual name(s), Organization

PREFACE

PREFACE

NOTE: Insert Preface Text here. Give OGC specific commentary: describe the technical content, reason for document, history of the document and precursors, and plans for future work. > Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

1

SCOPE

1 SCOPE

NOTE: Insert Scope text here. Give the subject of the document and the aspects of that scope covered by the document.

2

CONFORMANCE

2 CONFORMANCE

Conformance to the ModSpec by a profile of the OGC API — Environmental Data Retrieval Standard modular standard can be tested by inspection. The test suite is provided in Annex A.

Part 3: Core of the ModSpec (this document) defines one requirements class and one related conformance class:

- The Core: Common requirements for specifying standards documents. See Clause 7 and [annex-A-1]

The ModSpec contains normative language and thus places requirements on conformance, or mechanism for adoption, of candidate standards to which the ModSpec applies. In particular:

- OGC API-EDR Requirements Class: Core specifies the core requirements which shall be met by all standards claiming conformance to this Standard.
- Mapping the ModSpec to types of models gives information on how the ModSpec is to be applied to extensions to the core model for requirements and conformance clauses.

Such extensions are defined in additional Parts (volumes) to the ModSpec Standard.

3

NORMATIVE REFERENCES

3 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.

ISO: ISO 19106, *Geographic information — Profiles*. International Organization for Standardization, Geneva https://www.iso.org/standard/26011.html.

OGC API — Environmental Data Retrieval Standard

The ModSpec Model — Part 1: Core — A Standard for Designing and Writing Modular Standards

OpenAPI Initiative (OAI). **OpenAPI Specification 3.0** [online]. 2020 [viewed 2025-01-03]. The latest patch version at the time of publication of this standard was 3.0.4, available at https://spec.openapis.org/oas/v3.0.4

OpenAPI Initiative (OAI). **OpenAPI Specification 3.1** [online]. 2021 [viewed 2025-01-03]. The latest patch version at the time of publication of this standard was 3.1.1, available at https://spec.openapis.org/oas/v3.1.1

4

TERMS AND DEFINITIONS

4 TERMS AND DEFINITIONS

This document uses the terms defined in OGC Policy Directive 49, which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this document and OGC documents do not use the equivalent phrases in the ISO/IEC Directives, Part 2.

This document also uses terms defined in the OGC Standard for Modular specifications (OGC 08-131r3), also known as the ‘ModSpec’. The definitions of terms such as standard, specification, requirement, and conformance test are provided in the ModSpec.

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

4.1. Collection

Body of resources that belong or are used together. An aggregate, set, or group of related resources.

4.2. Conformance Module; Conformance Test Module

A set of related conformance classes and their associated components.

Note 1 to entry: When no ambiguity is possible, the word `test` may be omitted. i.e. conformance test module is the same as conformance module. Conformance modules may be nested in a hierarchical way.

[**SOURCE:** OGC 08-131r5]

4.3. Conformance Class; Conformance Test Class

A set of conformance tests that must be passed to receive a single certificate of conformance.

Note 1 to entry: When no ambiguity is possible, the word *test* may be left out, so conformance test class maybe called a conformance class.

[**SOURCE:** OGC 08-131r5]

4.4. Conformance Test

A test, abstract or real, of one or more requirements contained within a standard, or set of standards.

[**SOURCE:** OGC 08-131r5]

4.5. Requirement

Expression in the content of a standard conveying criteria to be fulfilled if compliance with the standard is to be claimed and from which no deviation is permitted.

[**SOURCE:** OGC 08-131r5]

4.6. Requirements Class

An aggregate of requirements with a single standardization target type that must all be satisfied to pass a conformance test Class.

[**SOURCE:** OGC 08-131r5]

4.7. Requirements Module

A set of related requirement classes and their associated components.

[**SOURCE:** OGC 08-131r5]

4.8. Standardization Goal

A concise statement of the problem that the standard helps address and the strategy envisioned for achieving a solution. This strategy typically identifies real-world entities that need to be modified or constrained. At the abstract level, those entities are the Standardization Target Types.

[**SOURCE:** OGC 08-131r5]

4.9. Standardization Target

Entity to which some requirements of a standard apply.

Note 1 to entry: The standardization target is the entity which may receive a certificate of conformance for a requirements class.

[**SOURCE:** OGC 08-131r5]

4.10. Standardization Target Type

Type of entity or set of entities to which the requirement of a standard apply

Note 1 to entry: For example, the standardization target type for The OGC API – Features Standard are Web APIs. The standardization target type for the CDB Standard is “datastore”. It is important to understand that a standard’s root standardization target type can have sub-types, and that there can be a hierarchy of target types. For example, a Web API can have sub types of client, server, security, and so forth. As such, each requirements class can have a standardization target type that is a sub-type of the root.

[**SOURCE:** OGC 08-131r5]

5

CONVENTIONS

5 CONVENTIONS

This sections provides details and examples for any conventions used in the document. Examples of conventions are symbols, abbreviations, use of XML schema, or special notes regarding how to read the document.

5.1. Identifiers

The normative provisions in this standard are denoted by the URI

`http://www.opengis.net/doc/spec/ogcapi-edr-3/1.0`

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

5.1.1. Shortcuts

In the interest of readability, the following terms will be used as shorthand for more complex text:

- Profile: A Profile is a standard or specification which restricts and/or extends an existing standard. This standard defines the rules for creating a profile of the OGC API — Environmental Data Retrieval Standard. The term “Profile” will be used in this document as shorthand for “profile of the OGC API — Environmental Data Retrieval Standard”.
- OGC API-EDR: The term OGC API-EDR will be used in this document as shorthand for the term “OGC API — Environmental Data Retrieval Standard”

6

CONTEXT

6 CONTEXT

6.1. Standardization Goal

The goal of this Standard is to ensure interoperabilty between implementations of the OGC API — Environmental Data Retrieval Standard (EDR API).

The OGC EDR Standard does not try to address every possible application domain. Rather, it provides a foundation which can be tailored for a specific domain. The result of this tailoring is a domain specific “profile” of the EDR API Standard.

A significant risk to this approach is that, in the act of profiling, interoperabilty will be compromized. This risk can be mitigated by establishing rules for how the EDR API Standard can be profiled. The goal of this Standard is to define a set of rules sufficient to ensure interoperability while retaining the adaptability needed to support domain-specific requirements.

6.2. Standardization Target Type

The Standardization Target Type for this Standard is the set of standards and specifications which profile the OGC API — Environmental Data Retrieval Standard.

It is important to understand that:

- This Standard is a standard for writing standards. It does not address the EDR API implementation.
- This Standard is a profile of the OGC ModSpec Model — Part 1: Core — A Standard for Designing and Writing Modular Standards (ModSpec).
- Implementations of this Standard are Profiles of the OGC API — Environmental Data Retrieval Standard
- The profiling model used is defined in ISO 19106:2004 Geographic information — Profiles

6.3. Profiles

ISO 19106:2004 Geographic information — Profiles is the ISO TC211 Standard for developing profiles of ISO TC211 Standards. This standard defined two conformance classes. These conformance classes can be thought of as two classes of profile.

- A Class 1 profile is a pure subset of the ISO geographic information standards.
- A Class 2 profile has the same basis as Class 1 but include extensions within the contexts permitted in the base standard. Additionally, a Class 2 profile permits the profiling of non-ISO geographic information standards as part of the profile.

In other words, a Class 1 profile restricts the base standard while a Class 2 profile both restricts and extends the base standard.

Both Class 1 and Class 2 Profiles of the OGC-EDR Standard are allowed.

7

REQUIREMENTS CLASS CORE

7 REQUIREMENTS CLASS CORE

REQUIREMENTS CLASS 1: REQUIREMENTS CLASS ‘CORE’	
IDENTIFIER	`http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
CONFORMANCE CLASS	Conformance class A.1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/conf/conf-class-core`
TARGET-TYPE	OGC API-EDR Profile Standard
NORMATIVE STATEMENTS	Requirement 18: `/req/core/modspec` Requirement 2: `/req/core/publishing` Requirement 3: `/req/core/api` Requirement 19: `/req/core/edr-conformant` Requirement 1: `/req/core/root` Requirement 4: `/req/core/requirements-set` Requirement 20: `/req/core/parameter-names` Requirement 7: `/req/core/collectionid` Requirement 6: `/req/core/extent` Requirement 14: `/req/core/instanceid` Requirement 8: `/req/core/output-format` Requirement 5: `/req/core/paging-support` Requirement 21: `/req/core/status-codes` Requirement 22: `/req/core/links` Requirement 9: `/req/core/data-query` Requirement 10: `/req/core/data-query-area` Requirement 11: `/req/core/data-query-corridor` Requirement 12: `/req/core/data-query-cube` Requirement 13: `/req/core/data-query-instances` Requirement 15: `/req/core/data-query-position` Requirement 16: `/req/core/data-query-radius` Requirement 17: `/req/core/data-query-trajectory` Requirement 23: `/req/core/asynchronous` Requirement 24: `/req/core/pubsub`

7.1. Resources and Paths

Web APIs provide access to resources through URI paths.

Table 1 — Environmental Data Retrieval API Paths

PATH TEMPLATE	METHOD	RESOURCE
{root}/	GET	Landing page
{root}/api	GET	API Description (optional)
{root}/conformance	GET	Conformance Classes
{root}/collections	GET	Metadata describing the collections of data available from this API.
{root}/collections/{collectionId}	GET	Metadata describing the collection of data which has the unique identifier{collectionId}
{root}/collections/{collectionId}/items	GET	Retrieve metadata about available items
{root}/collections/{collectionId}/{queryType}	GET, POST (Optional)	Retrieve data according to the query pattern from a collection with the unique identifier {collectionId}
{root}/collections/{collectionId}/instances	GET	Retrieve metadata about instances of acollection
{root}/collections/{collectionId}/instances/{instanceId}	GET	Retrieve metadata from a specific instanceof a collection with the unique identifiers{collectionId} and {instanceId}
{root}/collections/{collectionId}/instances/{instanceId}/{query Type}	GET, POST (Optional)	Retrieve data according to the query pattern from a specific instance of a collection with the unique identifiers{collection Id} and {instanceId}

Where:

- {root} = Base URI for the API server
- {collectionId} = an identifier for a specific collection of data
- {instanceId} = an identifier for a specific version or instance of a collection of data
- {queryType} = an identifier for a specific query pattern to retrieve data from a specific collection of data

7.1.1. API Landinding Page

Path = {root}/

The purpose of the landing page is to provide clients with a starting point for using the API. Any resource exposed through an API can be accessed by following paths or links starting from the landing page.

The landing page includes three metadata elements: title, description, and attribution. These three elements describe the API as a whole. Clients can expect to encounter metadata that is more resource-specific as they follow links and paths from the landing page.

Section 9.1 of API Common Part 1 /req/landing-page/root-op /req/landing-page/root-success

text

The landing page provides links that support exploration of the resources offered via the API. The most important component of a landing page is a list of links. OGC API — Common already requires some common links. The OGC API — Environmental Data Retrieval Standard does not impose any additional requirements on this resource.

The normative JSON Schema for the Landing Page is defined in the LandingPage.yaml document. While this schema provides a rich body of information about the API, only the Links property is required.

Profiles of the OGC API — Environmental Data Retrieval Standard are expected to provide a richer description of the API. The additional content which Profiles will mandate is defined in the following requirment.

REQUIREMENT 1	
IDENTIFIER	`/req/core/root`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
STATEMENT	The EDR Landing Page schema only requires the `links` property. A Profile of the OGC API — Environmental Data Retrieval Standard *SHALL* require the following additional properties and content:
A	The `Title` property *SHALL* be required and populated
B	The `Description` property *SHALL* be required and populated
C	The `Keywords` property *SHALL* be required and at least one keyword entry *SHALL* be populated
D	The `Provider` property *SHALL* be required and populated with the name and url properties
E	The `Contact` property *SHALL* be required and populated with the ——
F	The `Links` property *SHALL* define the links that SHALL be included in the Root response.

NOTE: Contacts and Links properties are not finished.

7.1.2. API Definition

Path = {root}/api

Every API implementation is required to provide an API Definition resource that describes the capabilities provided by that API. This resource can be used by client developers to understand the supported services, by software clients to connect to the server, and by development tools to support the implementation of servers and clients.

While the OGC API — Environmental Data Retrieval Standard leaves the selection of format for the API Definition document to the API implementor, at this time OpenAPI 3.0 and 3.1 are the only options supported.

Profiles of the OGC API — Environmental Data Retrieval Standard are to be more restricted. The following requirements mandate that there be an API Definition document to provide a common base for all implementatins, that this document be an OpenAPI document, that the specified minimum content is included, and that the version(s) of OpenAPI used are advertised.

REQUIREMENT 2	
IDENTIFIER	`/req/core/publishing`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
STATEMENT	An EDR profile *SHALL* be published as an OpenAPI JSON document.
A	The rules described in the requirements *SHALL* be encoded using the OpenAPI 3.1 specification.
B	The requirement rules *SHALL* be encoded in either the OpenAPI Path Item or in the Response object schema sections of the document.
C	The profile OpenAPI document *SHALL* describe the profile EDR API as follows: • The servers attributes of the OpenAPI root object *SHALL* be blank (the profile is not linked to specific implementations) • The Extent requirement rules *SHALL* be encoded in the JSON schema defined in the 200 responses for the /collections and /collections/{collection} id Paths object of the profile Open API document • The data_query type requirement rules *SHALL* be encoded in the JSON schema defined in the 200 responses for the /collections and /collections/{collection} id Paths object of the profile OpenAPI document • The data_query types *SHALL* be encoded as Paths objects in the OpenAPI document, where appropriate the output_format, default_output_format, crs, within_units, width-units, height-units and limit (paging) requirements SHALL be encoded in the child Parameter objects of the Paths object. • The output_format requirement rules *SHALL* be encoded in the 200 responses of the data_query type Paths objects

REQUIREMENT 2	
	- The Parameter_names requirements *SHALL* be encoded in the JSON schema defined in the 200 responses for the /collections and /collections/{collection} id Paths object of the profile OpenAPI document.
D	An EDR API *SHALL* advertise the location of the profile OpenAPI document it complies with
E	An EDR API *SHALL* advertise the location of the profile OpenAPI document it complies with in the links section of the API root with a link relation value of 'profile'

NOTE 1: Consider moving part 3 to a separate requirement.

REQUIREMENT 3	
IDENTIFIER	`/req/core/api`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
STATEMENT	A Profile of the OGC API — Environmental Data Retrieval Standard *SHALL* require that an implementation specify the versions of OpenAPI supported.

NOTE 2: OpenAPI 3.0 and OpenAPI 3.1 are two distinct Conformance Classes in the OGC API-EDR Standard. This requirement can be addressed in a Profile by including the appropriate conformance classes at {root}/conformance.

7.1.3. Declaration of Conformance Classes

Path = {root}/conformance

To support "generic" clients that want to access implementations of multiple OGC API Standards and extensions — and not "just" a specific API server, the OGC API-EDR Standard requires that an implementation declares the conformance classes it claims to have implemented.

NOTE 1: Add requirement that the Profile requires implementations to include both the core conformance class of the Profile and at least one of the OpenAPI conformance classes.

NOTE 2: Get guidence from the OGC Naming Authority on valid URIs for Profiles.

7.1.4. Collections

Path = {root}/collections

An API may support multiple collections. Additional requirements address how the Profile should document requirements at the per-collection level as well as on the landing page (where appropriate)

NOTE 1: A service may consist of multiple collections. While there may be common rules for all collections, a profile should be able to support different rules depending on the collection.

REQUIREMENT 4	
IDENTIFIER	`/req/core/requirements-set`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
A	The profile SHALL consists of a set of requirements for a collection and (if the collection supports instances) the instances of the collection. For each of the attributes listed, if it is in the collection (or instance), there SHALL be a requirement to define it.
B	A profile MAY include requirements for the landing page.
C	A profile MAY include requirements for multiple collections.

NOTE 2: Question — what is the purpose of this requirement?

Paging restrictions (limit parameter provided in the request, multi-page response). EDR only uses this parameter in requests to the query resources.

REQUIREMENT 5	
IDENTIFIER	`/req/core/paging-support`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
STATEMENT	If a Profile of the OGC API — Environmental Data Retrieval Standard is *extended* to support paging, then:
A	A requirement _SHALL _ be created for each combination of query pattern and output format that must support paging.
B	Each paging requirement *SHALL* specify the default number of items to return per page request.

Extent restrictions (one extent per collection provided in the response). Extents are also included in each {root}/collections/{collectionId} response. In short, an Extent appears in the metadata for a collection. Any request which returns a collection also returns an extent.

REQUIREMENT 6	
IDENTIFIER	`/req/core/extent`

REQUIREMENT 6	
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
A	A requirement MAY be defined specifying the rules for defining the Collection extent.
B	An extent requirement MAY specify the minimum spatial bounds that SHALL be supported
C	An extent requirement MAY specify the minimum temporal bounds that SHALL be supported
D	An extent requirement MAY specify the minimum vertical bounds that SHALL be supported
E	An extent requirement MAY specfiy the minimum bounds of custom extents that SHALL be supported
F	An extent requirements definition SHALL specify the rules for the spatial CRS. The attributes are constrained by one of: • Enumerated list of valid crs values • Regular expression defining valid crs string patterns.
G	If the collection has a temporal dimension an extent requirements definition SHALL specify the rules fo temporal TRS. The attributes are constrained by one of: • Enumerated list of valid trs values • Regular expression defining valid trs string patterns.
H	If the collection has a vertical dimension an extent requirements definition SHALL specify the rules for the vertical VRS. The attributes are constrained by one of: • Enumerated list of valid vrs values • Regular expression defining valid vrs string patterns.
I	For any custom dimensions an extent requirements definition SHALL specify the rules for the custom reference systems. The attributes are constrained by one of: • custom dimension name • custom dimension reference value • Where applicable enumerated list of valid custom dimension values
STATEMENT	(Regular expressions could be used to restrict reference system definitions to WKT2 or EPSG values)

NOTE 3: Needs to be re-visited

7.1.5. Collection

Path = {root}/collections/{collectionId}

Collection ID parameter restictions

REQUIREMENT 7	
IDENTIFIER	`/req/core/collectionid`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
STATEMENT	If a Profile of the OGC API — Environmental Data Retrieval Standard *restricts* the valid values of the Collection ID parameter, then:
A	The Profile *SHALL* specify the rules that the Collection ID values must follow.
B	Those rules *SHALL* be specified using either: • identifier string or • Regular expression defining valid string patterns.

Output Format restrictions

The -f parameter. Data format for the output data (available options are listed in the collectionsresponse).

REQUIREMENT 8	
IDENTIFIER	`/req/core/output-format`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
STATEMENT	For every output_format specified in any of the data_query enumerated lists, there *SHALL* be a requirement which defines the schema or structure of the data (depending on the format).

RECOMMENDATION 1	
IDENTIFIER	`/rec/core/output-format`
STATEMENT	The recommended definition approaches are as follows: • JSON – Link to a JSON Schema definition • XML – Link to a XML Schema definition • CSV, TSV, PSV, SSV – Link to a definition based on the CSV on the web recommendations available from the CSV on the Web Working Group. • Other types (e.g. binary file types) – Link to a description of the format

NOTE: Question: Where should the CSV citation point. There are multiple CSV on the Web Recommendataions.

7.1.6. Query

Path = {root}/collections/{collectionId}/{queryType}

A profile should specify the supported queries (how?)

7.1.7. Data Query (not in EDR)

REQUIREMENT 9	
IDENTIFIER	`/req/core/data-query`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
STATEMENT	A Profile of the OGC API — Environmental Data Retrieval Standard Standard *SHALL* require definition of the supported data queries.
A	The data_queries definitions *SHALL* specify which data queries a service supports. This can be defined as follows: • Enumerated list of query types
B	Each data_query type listed *SHALL* have a requirement definition.

7.1.8. Area Query

Path = {root}/collections/{collectionId}/area

REQUIREMENT 10	
IDENTIFIER	`/req/core/data-query-area`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
STATEMENT	If a Profile of the OGC API — Environmental Data Retrieval Standard *restricts* data queries by making the Area query mandatory, then:
A	The Profile *SHALL* include a requirement mandating the Area query.
B	The Area query requirement *SHALL* specify the following: • Enumerated list of output_format types • The default_output_format

REQUIREMENT 10	
	• Enumerated list of crs_details values • Enumerated list of the operations that the query supports (i.e. GET, POST)

7.1.9. Corridor Query

Path = {root}/collections/{collectionId}/corridor

REQUIREMENT 11	
IDENTIFIER	`/req/core/data-query-corridor`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
STATEMENT	If a Profile of the OGC API — Environmental Data Retrieval Standard *restricts* data queries by making the Corridor query mandatory, then:
A	The Profile *SHALL* include a requirement mandating the Corridor query.
B	The Corridor requirement *SHALL* specify the following: • Enumerated list of output_format types • The default_output_format • Enumerated list of crs_details values • Enumerated list of width-units values • Enumerated list of height-units values • Enumerated list of the operations that the query supports (i.e. GET, POST)

7.1.10. Cube Query

Path = {root}/collections/{collectionId}/cube

REQUIREMENT 12	
IDENTIFIER	`/req/core/data-query-cube`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
STATEMENT	If a Profile of the OGC API — Environmental Data Retrieval Standard *restricts* data queries by making the Cube query mandatory, then:

REQUIREMENT 12	
A	The Profile *SHALL* include a requirement mandating the Cube query.
B	The Cube query requirement *SHALL* specify the following: • Enumerated list of output_format types • The default_output_format • Enumerated list of crs_details values • Enumerated list of the operations that the query supports (i.e. GET, POST)

7.1.11. Locations Query

Path = {root}/collections/{collectionId}/locations

TBD

7.1.12. Instances Query

Path = {root}/collections/{collectionId}/instances

Having multiple versions or instances of the same collection, where the same information is reprocessed or regenerated is not unusal. Although these versions could be described as newcollections the instance query type allows this data to be described as different views of the same collection.

REQUIREMENT 13	
IDENTIFIER	`/req/core/data-query-instances`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
STATEMENT	If a Profile of the OGC API — Environmental Data Retrieval Standard *extends* data queries by making the Instances within a Collection queryable, then:
A	Instances *SHALL* be defined in the data_queries enumerated list.
B	A *NULL* value *SHALL* be used to indicate that no child instances can be queried.

7.1.12.1. Parameter instanceId

Path — Instance {root}/collections/{collectionId}/instances/{instanceId}

Instance ID restrictions

REQUIREMENT 14	
IDENTIFIER	`/req/core/instanceid`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
STATEMENT	If a Profile of the OGC API — Environmental Data Retrieval Standard *restricts* the valid values of the Instance ID parameter, then:
A	The Profile *SHALL* specify the rules that the Instance ID values must follow.
B	Those rules *SHALL* be specified using either: • identifier string • Regular expression defining valid string patterns.

7.1.12.2. Parameter queryType

Path — Instance Query {root}/collections/{collectionId}/instances/{instanceId}/{queryType}

7.1.13. Items Query

Path = {root}/collections/{collectionId}/items

EDR does not make any explicit changes to this path.

7.1.14. Position Query

Path = {root}/collections/{collectionId}/positions

REQUIREMENT 15	
IDENTIFIER	`/req/core/data-query-position`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
STATEMENT	If a Profile of the OGC API — Environmental Data Retrieval Standard *restricts* data queries by making the Position query mandatory, then:
A	The Profile *SHALL* include a requirement mandating the Position query.

REQUIREMENT 15	
B	The Position query requirement *SHALL* specify the following: • Enumerated list of output_format types • The default_output_format • Enumerated list of crs_details values • Enumerated list of the operations that the query supports (i.e. GET, POST)
C	The position query requirement *SHALL* also specify the logic used in selecting the data returned by the response, i.e. exact, nearest neighbour, most representative or interpolated.

7.1.15. Radius Query

Path = {root}/collections/{collectionId}/radius

REQUIREMENT 16	
IDENTIFIER	`/req/core/data-query-radius`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
STATEMENT	If a Profile of the OGC API — Environmental Data Retrieval Standard *restricts* data queries by making the Radius query mandatory, then:
A	The Profile *SHALL* include a requirement mandating the Radius query.
B	The Radius query requirement *SHALL* specify the following: • Enumerated list of output_format types • The default_output_format • Enumerated list of crs_details values • Enumerated list of within_units values • Enumerated list of the operations that the query supports (i.e. GET, POST)

7.1.16. Trajectory Query

Path = {root}/collections/{collectionId}/trajectory

REQUIREMENT 17	
IDENTIFIER	`/req/core/data-query-trajectory`

REQUIREMENT 17	
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
STATEMENT	If a Profile of the OGC API — Environmental Data Retrieval Standard *restricts* data queries by making the Trajectory query mandatory, then:
A	The Profile *SHALL* include a requirement mandating the Trajectory query.
B	The Trajectory query requirement *SHALL* specify the following: • Enumerated list of output_format types • The default_output_format • Enumerated list of crs_details values • Enumerated list of the operations that the query supports (i.e. GET, POST)

7.2. Parameters

7.3. General Requirements

Profile is conformant with the ModSpec

REQUIREMENT 18	
IDENTIFIER	`/req/core/modspec`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
STATEMENT	A profile of the OGC API — Environmental Data Retrieval Standard *SHALL* be conformant to the OGC Modular Specification.

Implementations of the Profile are conformant with EDR Part 1

REQUIREMENT 19	
IDENTIFIER	`/req/core/edr-conformant`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`

REQUIREMENT 19	
STATEMENT	A profile of the OGC API — Environmental Data Retrieval Standard *SHALL* require that a conformant implementation (standardization target) of that profile also demonstrate conformance to the OGC API — Environmental Data Retrieval Standard.

A common focus of Profiles is to restrict the values of Path parameters. The Profile should fully define requirements for these restrictions.

REQUIREMENT 20	
IDENTIFIER	`/req/core/parameter-names`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
STATEMENT	If a Profile of the OGC API — Environmental Data Retrieval Standard *restricts* the valid values and definitions of parameter_names, then,
A	Requirements *SHALL* be defined which specify the parameter_names and their definitions.
B	The parameter_names requirement definitions *SHALL* specify the required parameter_names objects in full including: • name, • units, • data type and • measurement duration
STATEMENT	for example:

```
"parameter_names": {
    "prmsl": {
        "type": "Parameter",
        "description": "Air pressure at sea level",
        "unit": {
            "label": "Pascals",
            "symbol": {
                "value": "Pa",
                "type": "https://qudt.org/vocab/unit/PA"
            }
        },
        "observedProperty": {
            "id": "http://codes.wmo.int/grib2/codeflag/4.2/_0-3-1",
            "label": "MSL Pressure"
        }
    },
    "t2m": {
        "type": "Parameter",
        "description": "Air temperature at 2m",
        "unit": {
        "label": "Kelvin",
        "symbol": {
            "value": "K",
            "type": "https://qudt.org/vocab/unit/K"
        }
        },
        "observedProperty": {
```

REQUIREMENT 20

```
                    "id": "http://codes.wmo.int/grib2/codeflag/4.2/0-0-0",
                    "label": "Air temperature at 2m"
                    }
                },
                "dd": {
                    "type": "Parameter",
                    "description": "Wind Direction",
                    "unit": {
                        "label": "degree true",
                        "symbol": {
                            "value": "°",
                            "type": "https://qudt.org/vocab/unit/DEG"
                        }
                    },
                    "observedProperty": {
                        "id": "http://codes.wmo.int/grib2/codeflag/4.2/0-2-0",
                        "label": "Wind Direction"
                    },
                    "measurementType": {
                        "method": "mean",
                        "duration": "-PT10M"
                    }
                },
                "ff": {
                    "type": "Parameter",
                    "description": "10m Wind Speed",
                    "unit": {
                        "label": "m/s",
                        "symbol": {
                          "value": "ms-1",
                          "type": "https://qudt.org/vocab/unit/M/s"
                        }
                    },
                    "observedProperty": {
                        "id": "http://codes.wmo.int/grib2/codeflag/4.2/0-2-1",
                        "label": "10m Wind Speed"
                    },
                    "measurementType": {
                        "method": "mean",
                        "duration": "-PT10M"
                    }
                }
            }
```

7.3.1. Http Status Codes

HTTP response

- Response status codes

REQUIREMENT 21

IDENTIFIER	`/req/core/status-codes`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`

REQUIREMENT 21	
STATEMENT	A Profile of the OGC API — Environmental Data Retrieval Standard *SHALL* require that the definitions of all http status codes SHALL be provided.
A	These definitions *SHALL* provide the following: • A description of the cause of the error. • A JSON schema for the message body structure

7.3.2. Links

- Response links

REQUIREMENT 22	
IDENTIFIER	`/req/core/links`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
STATEMENT	If a Profile of the OGC API — Environmental Data Retrieval Standard *restricts* valid responses to only those which include links, then:
A	The Profile *SHALL* require that link objects are included in a response.
B	The link objects *SHALL* defined the required link objects in full.
C	The link objects *SHALL* require that the `href`, `rel` and `type` attributes are populated.

STATEMENT	for example:
	`{`
	`"href": "https://creativecommons.org/licenses/by-nc/4.0/",`
	`"rel": "licence",`
	`"type": "text/html"`
	`},`
	`{`
	`"href": "https://docs.ogc.org/cs/21-069r2/21-069r2.html",`
	`"rel": "service-doc",`
	`"title": "CoverageJSON Community Standard v1.0"`
	`"type": "text/html"`
	`}`

7.3.3. Asynchronous Queries

While Web protocols typically use request-response operations, there is also support for asychonous operations.

HTTP Asynchrouous — This requirement address the use of HTTP asynchronous operations such as Webhooks and Callbacks.

REQUIREMENT 23	
IDENTIFIER	`/req/core/asynchronous`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
STATEMENT	If a Profile *extends* the OGC API — Environmental Data Retrieval Standard with support for asynchronous opperations, then:
A	Requirements *SHALL* be defined for each query type that is asynchronous
B	Each asynchronous query type requirement *SHALL* define the HTTP Status Code and provide a message schema and text used to inform the user that the response is asynchronous.
C	Each asynchronous query type requirement *SHALL* document the mechanism for delivering the result of the asynchronous query.

PERMISSION 1	
IDENTIFIER	`/per/core/asynchronous`
STATEMENT	The documentation of the mechanism for delivering the result of the asynchronous query *MAY* be provided through a link to an external document.

Publish-Subscribe — This requirement addresses the use of Publish-Subscribe protocols. These are protocols supported in addition to HTTP.

REQUIREMENT 24	
IDENTIFIER	`/req/core/pubsub`
INCLUDED IN	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
STATEMENT	If a Profile of the OGC API — Environmental Data Retrieval Standard *extends* the supported operations to include Publish-Subscribe operations, then:
A	Support for the OGC API — Environmental Data Retrieval — Part 2: Publish-Subscribe workflow Standard *SHALL* be required.
B	The pubsub requirement *SHALL* specify the channels that *SHALL* be supported
C	The pubsub requirement *SHALL* specify the payloads that a pubsub channel *SHALL* support

8

MEDIA TYPES FOR ANY DATA ENCODING(S)

8 MEDIA TYPES FOR ANY DATA ENCODING(S)

A section describing the MIME-types to be used is mandatory for any standard involving data encodings. If no suitable MIME type exists in http://www.iana.org/assignments/media-types/index.html then this section may be used to define a new MIME type for registration with IANA.

A

ANNEX A (INFORMATIVE) CONFORMANCE CLASS ABSTRACT TEST SUITE (NORMATIVE)

ANNEX A (INFORMATIVE) CONFORMANCE CLASS ABSTRACT TEST SUITE (NORMATIVE)

A.1. Conformance Class Core

CONFORMANCE CLASS A.1: CONFORMANCE CLASS ‘CORE’	
IDENTIFIER	`http://www.opengis.net/spec/ogcapi-edr-3/1.0/conf/conf-class-core`
REQUIREMENTS CLASS	Requirements class 1: `http://www.opengis.net/spec/ogcapi-edr-3/1.0/req/req-class-core`
CONFORMANCE TESTS	Abstract test A.1: `/conf/core/modspec` Abstract test A.2: `/conf/core/publishing` Abstract test A.3: `/conf/core/api` Abstract test A.4: `/conf/core/edr-conformant` Abstract test A.5: `/conf/core/root` Abstract test A.6: `/conf/core/requirements-set` Abstract test A.7: `/conf/core/parameter-names` Abstract test A.8: `/conf/core/collectionid` Abstract test A.9: `/conf/core/extent` Abstract test A.10: `/conf/core/instanceid` Abstract test A.11: `/conf/core/output-format` Abstract test A.12: `/conf/core/paging-support` Abstract test A.13: `/conf/core/status-codes` Abstract test A.14: `/conf/core/links` Abstract test A.15: `/conf/core/data-query` Abstract test A.16: `/conf/core/data-query-area` Abstract test A.17: `/conf/core/data-query-corridor` Abstract test A.18: `/conf/core/data-query-cube` Abstract test A.19: `/conf/core/data-query-instances` Abstract test A.20: `/conf/core/data-query-position` Abstract test A.21: `/conf/core/data-query-radius` Abstract test A.22: `/conf/core/data-query-trajectory` Abstract test A.23: `/conf/core/asynchronous`

CONFORMANCE CLASS A.1: CONFORMANCE CLASS ‘CORE’
Abstract test A.24: `/conf/core/pubsub`

ABSTRACT TEST A.1	
IDENTIFIER	`/conf/core/modspec`
REQUIREMENT	Requirement 18: `/req/core/modspec`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.2	
IDENTIFIER	`/conf/core/publishing`
REQUIREMENT	Requirement 2: `/req/core/publishing`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.3	
IDENTIFIER	`/conf/core/api`
REQUIREMENT	Requirement 3: `/req/core/api`
TEST PURPOSE	Paraphrase the requirement — Validate that all the parts of a requirement are testable and that Failure to pass any part of a requirement is also a failure to pass the associated conformance test.
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.4	
IDENTIFIER	`/conf/core/edr-conformant`

ABSTRACT TEST A.4	
REQUIREMENT	Requirement 19: `/req/core/edr-conformant`
TEST PURPOSE	Validate that the Profile Standard requires that all implementations demonstrate conformance with the OGC API-EDR Standard.
TEST METHOD	Inspect the document to verify the above.

NOTE: this “purpose” requires more specificity.

ABSTRACT TEST A.5	
IDENTIFIER	`/conf/core/root`
REQUIREMENT	Requirement 1: `/req/core/root`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.6	
IDENTIFIER	`/conf/core/requirements-set`
REQUIREMENT	Requirement 4: `/req/core/requirements-set`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.7	
IDENTIFIER	`/conf/core/parameter-names`
REQUIREMENT	Requirement 20: `/req/core/parameter-names`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.8	
IDENTIFIER	`/conf/core/collectionid`
REQUIREMENT	Requirement 7: `/req/core/collectionid`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.9	
IDENTIFIER	`/conf/core/extent`
REQUIREMENT	Requirement 6: `/req/core/extent`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.10	
IDENTIFIER	`/conf/core/instanceid`
REQUIREMENT	Requirement 14: `/req/core/instanceid`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.11	
IDENTIFIER	`/conf/core/output-format`
REQUIREMENT	Requirement 8: `/req/core/output-format`
TEST PURPOSE	TBD

ABSTRACT TEST A.11	
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.12	
IDENTIFIER	`/conf/core/paging-support`
REQUIREMENT	Requirement 5: `/req/core/paging-support`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.13	
IDENTIFIER	`/conf/core/status-codes`
REQUIREMENT	Requirement 21: `/req/core/status-codes`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.14	
IDENTIFIER	`/conf/core/links`
REQUIREMENT	Requirement 22: `/req/core/links`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.15	
IDENTIFIER	`/conf/core/data-query`

ABSTRACT TEST A.15	
REQUIREMENT	Requirement 9: `/req/core/data-query`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.16	
IDENTIFIER	`/conf/core/data-query-area`
REQUIREMENT	Requirement 10: `/req/core/data-query-area`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.17	
IDENTIFIER	`/conf/core/data-query-corridor`
REQUIREMENT	Requirement 11: `/req/core/data-query-corridor`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.18	
IDENTIFIER	`/conf/core/data-query-cube`
REQUIREMENT	Requirement 12: `/req/core/data-query-cube`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.19	
IDENTIFIER	`/conf/core/data-query-instances`
REQUIREMENT	Requirement 13: `/req/core/data-query-instances`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.20	
IDENTIFIER	`/conf/core/data-query-position`
REQUIREMENT	Requirement 15: `/req/core/data-query-position`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.21	
IDENTIFIER	`/conf/core/data-query-radius`
REQUIREMENT	Requirement 16: `/req/core/data-query-radius`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.22	
IDENTIFIER	`/conf/core/data-query-trajectory`
REQUIREMENT	Requirement 17: `/req/core/data-query-trajectory`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.23	
IDENTIFIER	`/conf/core/asynchronous`
REQUIREMENT	Requirement 23: `/req/core/asynchronous`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

ABSTRACT TEST A.24	
IDENTIFIER	`/conf/core/pubsub`
REQUIREMENT	Requirement 24: `/req/core/pubsub`
TEST PURPOSE	TBD
TEST METHOD	Inspect the document to verify the above.

B

ANNEX B (INFORMATIVE) TITLE

B

ANNEX B (INFORMATIVE) TITLE

NOTE: Place other Annex material in sequential annexes beginning with “B” and leave final two annexes for the Revision History and Bibliography

C

ANNEX C (INFORMATIVE) REVISION HISTORY

C

ANNEX C (INFORMATIVE) REVISION HISTORY

Table C.1

DATE	RELEASE	EDITOR	PRIMARY CLAUSES MODIFIED	DESCRIPTION
2016-04-28	0.1	G. Editor	all	initial version

BIBLIOGRAPHY

BIBLIOGRAPHY

NOTE: The TC has approved Springer LNCS as the official document citation type.

Springer LNCS is widely used in technical and computer science journals and other publications

– Actual References:

[n] Journal: Author Surname, A.: Title. Publication Title. Volume number, Issue number, Pages Used (Year Published)

[1] *OGC: OGC Testbed 12 Annex B: Architecture* (2015).