

# OGC API Environmental Data Retrieval (OGC API-EDR) Profile definitions

## Purpose:

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.

To achieve interoperability a profile must provide rules for the following Collection options:

## id

A Requirement definition would specify the rules that the Collection id string must follow. This can be one of the following:

- identifier string
- Regular expression defining valid string patterns.

## Extent

An extent [requirements](#) definition SHALL specify the rules for the spatial CRS, and where supported temporal TRS, vertical VRS and any custom dimension attributes. The attributes are constrained by one of:

- Enumerated list of valid crs values
- Regular expression defining valid crs string patterns.

If the collection has a temporal dimension one of:

- Enumerated list of valid trs values
- Regular expression defining valid trs string patterns.

If the collection has a vertical dimension one of:

- Enumerated list of valid vrs values
- Regular expression defining valid vrs string patterns.

If the collection requires a custom dimension as requirements definition SHALL specify:

- custom dimension name
- custom dimension reference value
- Where applicable enumerated list of valid custom dimension values

(Regular expressions could be used to restrict reference system definitions to WKT2 or EPSG values)

## Data\_queries

A data\_queries [requirement](#) definition SHALL specify which data queries a service supports. This can be defined as follows:

- Enumerated list of query types

Each data\_query type listed SHALL also have a requirement definition.

### Position

A 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)

A 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.

### Radius

A 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)

### Area

An area 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)

### Cube

A cube [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)

## Trajectory

A trajectory [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)

## Corridor

A 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)

## Instances

Instances SHALL be defined in the data\_queries enumerated list if the collection has child instances that can be queried.

## Parameter\_names

The parameter\_names [requirement](#) definition would specify the required parameter\_names objects in full. The parameter\_names object defines the name, units, data type and measurement duration of the data value so it is essential that all implementations are consistent for interoperability.

## Output\_format

For every output\_format specified in any of the data\_query enumerated lists there SHOULD be a [requirement](#) which defines the schema or structure of the data (depending on the format).

Suggested 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
- Other types (e.g. binary file types) – Link to a description of the format

## Paging support

Paging support depends on query pattern and output format. A [requirement](#) SHALL be created for each combination of query pattern and output format that must support paging.

Each paging [requirement](#) SHOULD also include a recommendation for the default number of items to return per page request.

## Asynchronous queries

A [requirement](#) SHALL be defined for each query type that is asynchronous.

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.

The [requirement](#) will also document (this can be a link to another document) the mechanism for delivering the result of the asynchronous query.

## HTTP Status Codes

For every http status code supported by the profile. A [requirement](#) SHALL be created. The [requirement](#) SHALL provide the following:

- A description of the cause of the error.
- A JSON schema for the message body structure

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.

For each of the [requirements](#) defined in the profile there SHALL be a conformance test which defines how to validate compliance with the requirement.

## Publishing an EDR profile

An EDR profile SHALL be published as an OpenAPI JSON document. The rules described in the [requirements](#) SHALL be encoded using the [OpenAPI 3.1 specification](#). The [requirement](#) rules are encoded in either the OpenAPI Path Item or in the Response object schema sections of the document.

The profile OpenAPI document should 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

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

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

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.

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'

## Glossary

### Collection

Body of resources that belong or are used together. An aggregate, set, or group of related resources. ([OGC20-024](#))

### Conformance Module; Conformance Test Module

Set of related tests, all within a single conformance test class ([OGC 08-131r3](#))

NOTE: 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.

### Conformance Class; Conformance Test Class

Set of conformance test modules that SHALL be applied to receive a single certificate of conformance ([OGC 08-131r3](#)).

NOTE: When no ambiguity is possible, the word `\_test\_` may be left out, so conformance test class maybe called a conformance class.

### Requirement

Expression in the content of a document conveying criteria to be fulfilled if compliance with the document is to be claimed and from which no deviation is permitted ([OGC 08-131r3](#))

### Requirements Class

Aggregate of all requirement modules that SHALL all be satisfied to satisfy a conformance test class ([OGC 08-131r3](#))

### Requirements Module

Aggregate of requirements and recommendations of a specification against a single standardization target type ([OGC 08-131r3](#))

## Standardization Target

Entity to which some requirements of a standard apply ([OGC 08-131r3](#))

NOTE: The standardization target is the entity which may receive a certificate of conformance for a requirements class.



## Bibliography

\* Open Geospatial Consortium: The Specification Model -- A Standard for Modular specifications [[OGC 08-131](#)]

\* [OpenAPI Specification v3.1.1](#)