# Catalogue Open Data Protocol (OData API)
# User Guide

## Content

## Table of Figures



## Table Index

# 1. Introduction

## 1.1. Purpose

This document provides end users guidance on usage of the ONDA Open Data Protocol (OData) API, exposed by the Catalogue, which allows interactive data discovery and download, via computer programs/scripts.

## 1.2. Change register

| Version/Rev. | Date | Change | Reason |
|---|---|---|---|
| 1.0 | 20/06/2018 | | First Issue |
| 1.1 | 27/06/2018 | | Review |
| 1.2 | 05/07/2018 | Sections 3.1.1, 3.1.2, 3.2, 5.4 | `Metadatas` Entity has been renamed to `Metadata` |
| 1.3 | 09/07/2018 | Section 1.5 | Links update |
| 1.4 | 03/09/2018 | Section 1.5, 3.1.1 and 3.1.2<br><br>Section 4.1.6 and 0 | downloadable and footprint properties have been added.<br><br>Additional references to metadata indexes have been added.<br><br>Correction of the query concerning the sort by creation date. |
| 1.5 | 26/11/2018 | Footer | ONDA by Serco registered trademark added |
| 1.6 | 29/01/2019 | Par. 5.8 | Updated |
| 1.7 | 28/02/2019 | New Paragraphs 5.3, 5.4 and 5.7 have been added | Possibility to search a specific metadatum of a product has been implemented |

## 1.3. Structure of the document

This document is composed of the following sections:

- **Section 1** is this Introduction.
- **Section 2** contains an overview of what the Open Data Protocol (OData) is, a description of the Entity Data Model (EDM) and its Entities.

- **Section 3** provides a description of the ONDA OData Entity Data Model, its Entities and related Properties.
- **Section 4** provides the basic criteria to build an OData query by means of different filters.
- **Section 5** contains an exhaustive list of examples to use the ONDA OData API exposed by the Catalogue. Beginners to OData can also leverage this Section as a structured way to learn.

## 1.4. Acronyms

**Table 1 – Acronyms and Abbreviations**

| Acronym | Definition |
|---|---|
| AD | Applicable Document |
| API | Application Programming Interface |
| ASCII | American Standard Code for Information Interchange |
| EDM | Entity Data Model |
| ESA | European Space Agency |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hyper Text Transfer Protocol Secure |
| JSON | JavaScript Object Notation |
| RD | Reference Document |
| UUID | Universally Unique Identifier |
| URI | Uniform Unique Identifier |
| URL | Uniform Resource Locator |

## 1.5. Reference Documents

**Table 2 – Reference Documents**

| ID | Document Title | Reference |
|---|---|---|
| RD-1 | OData v4 documentation | http://www.odata.org/documentation/ |
| RD-2 | Metadata for Sentinel-1 Products | https://www.gael-systems.com/dias-metadata-sentinel-1/ |
| RD-3 | Metadata for Sentinel-2 Products | https://www.gael-systems.com/dias-metadata-sentinel-2/ |
| RD-4 | Metadata for Sentinel-3 Products | https://www.gael-systems.com/dias-metadata-sentinel-3/ |

| | | |
|---|---|---|
| RD-5 | Metadata for Envisat Products | https://www.gael-systems.com/dias-metadata-envisat/ |
| RD-6 | Metadata for Copernicus Land Products | https://www.gael-systems.com/dias-metadata-copernicus-land/ |
| RD-7 | Metadata for Copernicus Marine Products | https://www.gael-systems.com/dias-metadata-copernicus-marine/ |
| RD-8 | Metadata for Landsat-8 Products | https://www.gael-systems.com/dias-metadata-landsat-8-geotiff/ |

# 2. Open Data Protocol overview

**OData (Open Data Protocol)** is a standard that defines conventions, rules and formats for handling data on the web using Hypertext Transfer Protocol (HTTP/HTTPS) requests.

OData is based on the Representational State Transfer (REST) architecture, which allow resources – identified using Uniform Resource Identifiers (URIs) – to be published and edited by Web clients using simple HTTP messages. The OData specification defines a set of rules for constructing URIs to identify the data and metadata exposed by an OData server as well as a set of reserved URI query string operators.

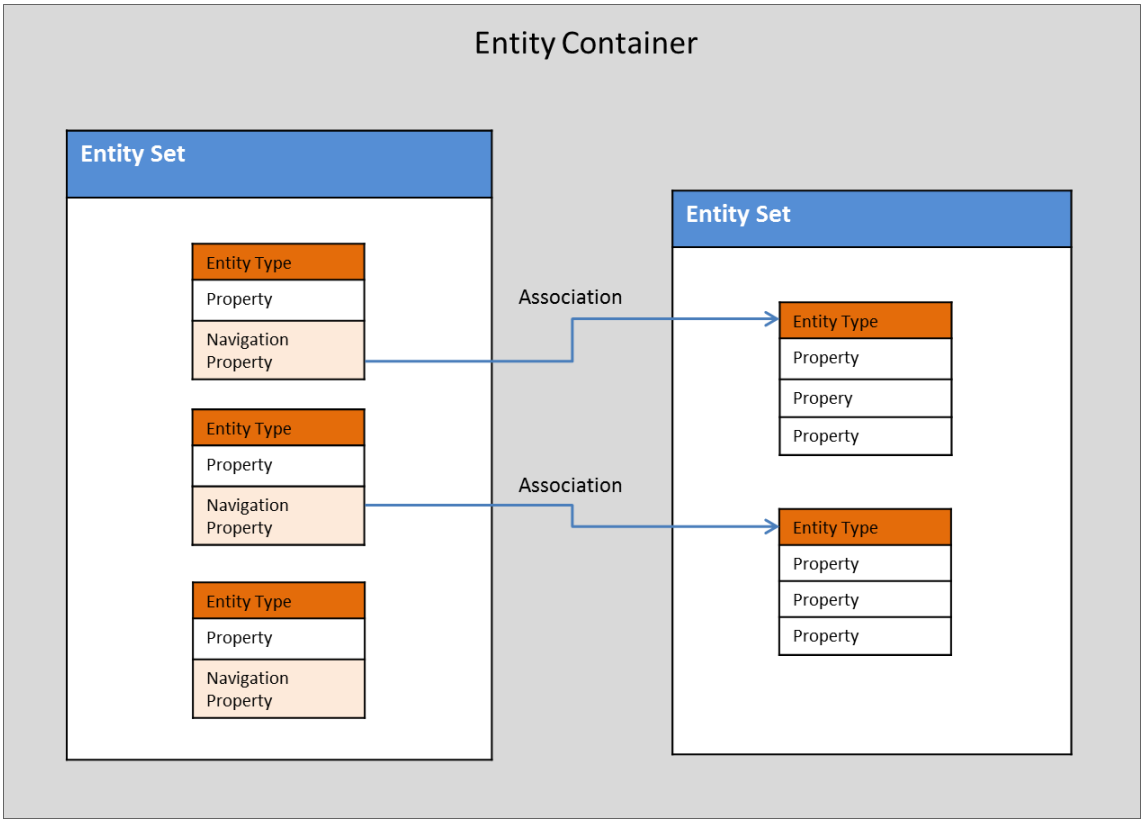
In the case of the ONDA, the OData interface exposed by the Catalogue allows browsing, selecting and downloading of EO products in the Catalogue itself.

## 2.1. Entity Data Model concept

This Section provides a high-level description of the **Entity Data Model (EDM)**: the abstract data model that must be used to describe the data exposed by an OData service.

The main concepts in describing the structure of data in EDM are depicted in Figure 1 and described in the following paragraphs.

**Figure 1 – EDM key concepts**



As we will see in detail in this Section, the EDM organizes entities into a simple hierarchy. Each entity is part of an entity set (2.1.2), and each entity set belongs to an entity container. Entities, each of which is of some entity type (2.1.1), also have a simple structure: they contain properties, each of which contains data that this entity holds. To describe the data in properties, the EDM defines a variety of data types, such as String, Boolean, Int16, Int32, Binary and DateTime. Special properties – called navigation properties – represent associations and implement connections between entities.

### 2.1.1. **Entity Type**

An **Entity Type** is the fundamental building block for describing the structure of data in the EDM. Entity types represent a specific type of data.

Each Entity Type comprises:

- a unique name
- a unique key (for uniquely identifying instances of Entity Types and allowing Entity Type instances to participate in relationships)
- data in form of Properties
- Navigation Properties (optional)

### 2.1.2. **Entity Set**

**EntitySets** are collections of instances of EntityTypes. Each instance of an entity within an entity set can be accessed by its unique key.

# 3. ONDA OData Entity Data Model

This Section specifies the ONDA OData Entity Data Model. The current version of the ONDA OData service is based on **OData v4.0** [RD-1].

## 3.1. ONDA Entity Types

Each Entity and its properties are listed in the Tables of the following paragraphs.

### 3.1.1. Product

| **EntityType name** | | |
|---|---|---|
| `Product` | | |
| **Description** | | |
| The `Product` Entity represents the fundamental data element which can be managed by the ONDA. | | |
| **Corresponding EntitySet name** | | |
| Products | | |
| **Properties** | **Type** | **Description** |
| Id (key) | guid | Unique identifier of the Product. |
| Name | String | Product Name |
| CreationDate | DateTimeOffset | Time the Product was archived |
| Offline | Boolean | Status of the product |
| Size | Int64 | Size of the Product |
| Pseudopath | String | Virtual path used to classify the product in ENS filesystem. |
| Footprint | String | Footprint (polygon or multi-poly) |
| Quicklook | Binary | Quicklook of the product (if available) |
| Downloadable | Boolean | Status of downloadability |
| **Nested Entity** | | |
| Metadata | | |

### 3.1.2. Metadata

| **EntityType name** | | |
|---|---|---|
| `Metadata` | | |
| **Description** | | |
| The `Metadata` Entity contains the full set of metadata (attributes) extracted at ingestion time for each specific mission (e.g. Sentinel-1, Sentinel-2, Sentinel-3, etc) | | |
| **Corresponding EntitySet name** | | |
| Metadata | | |
| **Properties** | **Type** | **Description** |
| ID (key) | String | Unique ID of the Metadatum (=name) |
| Name | String | Name of the Metadatum |
| Value | String | Value of the Metadatum (if present) |
| **Nested Entity** | | |
| None | | |

The following link refer to the lists of all the Metadata nested properties of the Product entity Type (see Paragraph 3.1). These properties change depending on the Product mission.

- Metadata Index S1: see [RD-2],
- Metadata Index S2: see [RD-3],
- Metadata Index S3: see [RD-4],
- Metadata for Envisat products: see [RD-5],
- Metadata for Copernicus Land products: see [RD-6],
- Metadata for Copernicus Marine products: see [RD-7],
- Metadata for Landsat-8: see [RD-8].

## 3.2. ONDA Entity Sets

The entity sets are divided in two categories:

- The ***top-level EntitySets***, contained in the highest framework. These entity sets are accessible with the following query:

  https://catalogue.onda-dias.eu/dias-catalogue/EntitySet

- The ***nested EntitySets***, accessible via an association starting from a top level Entity Set. The OData query described above does not recognize entities that are nested within other entities. This Entity Set can be explored through the "parent" Entity Set by means of the key property.

**Table 3 – ONDA entity sets**

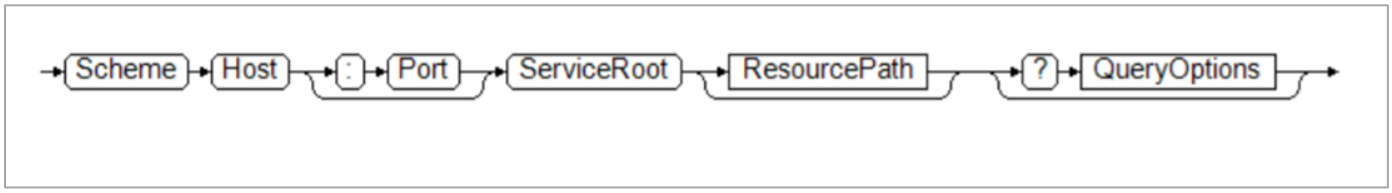
| ONDA entity set | Entity set level |
|---|---|
| `Products` | First level |
| `Metadata` | Nested |

# 4. How to create ONDA OData queries

In this Section we describe the rules for constructing URIs allowing access to data Entities and Properties listed in Section 0.

The general scheme summarizing the URI components is depicted in Figure 2.

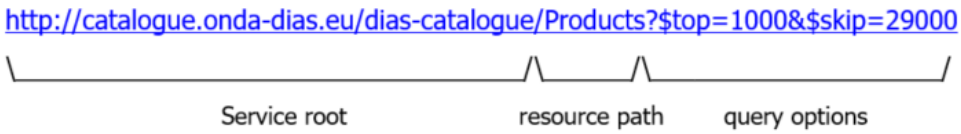
**Figure 2 – URI components scheme**




A URI used by an OData service has up to three significant parts:

- The service root (Scheme+Host:Port+ServiceRoot);
- The resource path;
- The query string options.

For Example:




The **service root URI** identifies the root of an Odata service. For the ONDA Odata interface, the **Scheme** is the protocol HTTPS, **Host** represents the ONDA server IP address or hostname catalogue.onda-dias.eu).

The **ResourcePath** identifies the resource to be interacted with and enables any aspect of the data model exposed by an Odata service to be addressed. For the ONDA interface, it is Products.

**QueryOptions** are additional parameters for the query which specify which data are returned and how they are formatted. We will analyse the query options in the Section 4.1, in order to help the user go through the most common and useful scenarios.

## 4.1. Query string options

OData supports various kinds of query options for querying data. System query options are query string parameters that control the amount and order of the data returned for the resource identified by the URL. The names of all system query options are prefixed with a dollar character, ($). A query string starts with a question mark (?), and the query options are separated by an ampersand (&). The asterisk (*) is used to specify all values. Each query option can be set on a particular value with (=).

A query string option (**QueryOption** of Figure 2) can be represented in this way:

```
$query_option_A{=value_1}&…&$query_option_N{=value_2}
```

The query options admitted by the Data Hub service are listed in the following Table:

**Table 4 – Query Options**

| Query Option | Description | Reference |
|---|---|---|
| `$top` | determines the maximum number of records to return | Section 0 |
| `$skip` | requests the number of items in the queried collection that are to be skipped and not included in the result | Section 0 |
| `$count` | allows clients to request a count of the matching resources included with the resources in the response. | Section 4.1.3 |
| `$select` | specifies a subset of properties to return /allows clients to requests a specific set of properties for each entity | Section 4.1.4 |
| `$format` | specifies the HTTPS response format e.g. XML or JSON /allows clients to request a response in a particular format | Section 4.1.5 |
| `$search` | restricts the result to include only those entities matching the specified search expression. The definition of what it means to match is dependent upon the implementation. | Section 4.1.7 |

### 4.1.1. Query option $top

A data service URI with a $top System Query Option identifies a subset of the Entities in an EntitySet identified by the Resource Path section of the URI.

This subset is formed by selecting only the first M items of the set, where M is an integer greater than or equal to zero specified by this query option. If a value less than zero is specified, the URI should be considered malformed.

*Syntax:*

https://catalogue.onda-dias.eu/dias-catalogue/Products?$top=<M>

*ONDA example:*

List the last 10 Sentinel-1 products published on the ONDA Catalogue:

https://catalogue.onda-dias.eu/dias-catalogue/Products?$search="name:S1*"&$top=10

### 4.1.2. Query option $skip

A data service URI with a $skip System Query Option identifies a subset of the Entities in an EntitySet identified by the Resource Path section of the URI.

That subset is defined by seeking N Entities into the EntitySet and selecting only the remaining Entities (starting with Entity N+1). N is an integer greater than or equal to zero specified by this query option. If a value less than zero is specified, the URI should be considered malformed.

The syntax is:

https://catalogue.onda-dias.eu/dias-catalogue/Products?$skip=<N>

A client can request a particular page of items by combining $top and $skip.

### 4.1.3. Query option $count

The $count system query option allows clients to request a count of the matching resources included with the resources in the response.

*Syntax:*

https://catalogue.onda-dias.eu/dias-catalogue/EntitySet/$count

*ONDA example:*

Total number of all products in the ONDA Catalogue:

https://catalogue.onda-dias.eu/dias-catalogue/Products/$count

Total number of Sentinel-2 products in the ONDA Catalogue:

https://catalogue.onda-dias.eu/dias-catalogue/Products/$count?$search="name:S2*"

### 4.1.4. Query option $select

The $select system query option allows the clients to requests a limited set of properties for each entity. The value of a $select System Query Option is a comma-separated list of selection clauses.

*Syntax:*

https://catalogue.onda-dias.eu/dias-catalogue/Entity?$select=Property_1[,Property_2]

*ONDA example:*

Querying the ID property of all the Products

https://catalogue.onda-dias.eu/dias-catalogue/Products?$select=id

Querying the Name and CreationDate properties of all the Products

https://catalogue.onda-dias.eu/dias-catalogue/Products?$select=name,creationDate

### 4.1.5. Query option $format

The $format system query option allows clients to request a response in a particular format. The default format is XML. Valid values for the $format query string option are:

- atom,
- xml,
- json

*ONDA example:*

Display all products in the archive in Json format:

https://catalogue.onda-dias.eu/dias-catalogue/Products?$format=json

Display all products in the archive in XML format:

https://catalogue.onda-dias.eu/dias-catalogue/Products?$format=xml

### 4.1.6. Query option $orderby

The $orderby system query option allows clients to request resources in either ascending or descending order.

*Syntax:*

https://catalogue.onda-dias.eu/dias-catalogue/Entity?$orderby=property [asc|desc]

*ONDA example:*

Querying product sorted by ascending creation date

https://catalogue.onda-dias.eu/dias-catalogue/Products?$orderby=creationDate%20asc

### 4.1.7. Query option $search

A URI with a $search System Query Option identifies a subset of the Entities from an EntitySet identified by the Resource Path section of the URI. The subset is determined by selecting only the Entities that satisfy the predicate expression specified by the query option.

*Syntax:*

https://catalogue.onda-dias.eu/dias-catalogue/Entity?$search="<keyword>:<values> [AND <keyword>:<value>]"

Depending on the keyword, the value(s) can be specified as a single value or range of values.

Search keywords can be combined with each other using Operators.

*ONDA example:*

Querying Products in a specific creation date range

https://catalogue.onda-dias.eu/dias-catalogue/Products?$search="creationDate:[2018-05-01T00:00:00.000Z%20TO%20*]"

Querying Sentinel-1 MSI Products

https://catalogue.onda-dias.eu/dias-catalogue/Products?$search="instrumentShortName:MSI"

Querying Sentinel-3 SRAL Products

https://catalogue.onda-dias.eu/dias-catalogue/Products?$search="instrumentShortName:SRAL"

Querying products with SLC product type

https://catalogue.onda-dias.eu/dias-catalogue/Products?$search=%22productType:IW_SLC__1S%22

# 5. ONDA OData queries by examples

This Section contains an exhaustive list of examples to query and filter Entities and Properties, and therefore can be used as a "beginner user manual".

## 5.1. Querying all the Entities of the ONDA OData API

*Syntax:*

https://catalogue.onda-dias.eu/dias-catalogue/

Response Payload:

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
<app:service xmlns:atom="http://www.w3.org/2005/Atom" xmlns:app="http://www.w3.org/2007/app" xmlns:metadata="http://docs.oasis-open.org/odata/ns/metadata"
 metadata:context="$metadata">
  <app:workspace>
    <atom:title>Ens.Container</atom:title>
    <app:collection href="Metadatas" metadata:name="Metadatas">
      <atom:title>Metadatas</atom:title>
    </app:collection>
    <app:collection href="Products" metadata:name="Products">
      <atom:title>Products</atom:title>
    </app:collection>
  </app:workspace>
</app:service>
```


## 5.2. Querying products in the Catalogue

The following OData URI returns all the products stored in the ONDA Catalogue. Each product record includes the Id, the product name and its properties, the link for download and the link to metadata.

*Syntax:*

https://catalogue.onda-dias.eu/dias-catalogue/Products

By default it provides a list of 100 records sorted by creation date and arranged in descending order.

## 5.3. Querying Products, showing all Metadata together with their Properties

The following OData URI returns all the products stored in the ONDA Catalogue, with their properties, and showing also all their Metadata in the response payload.

*Syntax:*

https://catalogue.onda-dias.eu/dias-catalogue/Products?$expand=Metadata

Using $skip and $top options, it is possible to choose how many products are requested.

## 5.4. Querying Products, showing all the Properties (but not the quicklook) and Metadata:

It is also possible to show, together with all the Metadata, only a subset of chosen Properties, specifying which of them.

For example:

https://catalogue.onda-dias.eu/dias-catalogue/Products?$expand=Metadata&$select=id,name,creationDate,beginPosition,offline,size,pseudopath,footprint,downloadable

## 5.5. Querying a single Product in the archive

*Syntax:*

https://catalogue.onda-dias.eu/dias-catalogue/Products(Id)

For example:

https://catalogue.onda-dias.eu/dias-catalogue/Products(1307ac04-597e-4b97-8812-81fa52d6bd1a)

The request returns an individual entity of type Product by the given Id '1307ac04-597e-4b97-8812-81fa52d6bd1a'

## 5.6. Search product metadata

*Syntax:*

https://catalogue.onda-dias.eu/dias-catalogue/Products(Id)/Metadata

For example:

https://catalogue.onda-dias.eu/dias-catalogue/Products(1307ac04-597e-4b97-8812-81fa52d6bd1a)/Metadata

The request returns all the metadata of the Product identified by the Id '1307ac04-597e-4b97-8812-81fa52d6bd1a'.

## 5.7. Search a specific metadatum

*Syntax:*

https://catalogue.onda-dias.eu/dias-catalogue/Products(Id)/Metadata('metadatum')

For example:

https://catalogue.onda-dias.eu/dias-catalogue/Products(30d01828-1b55-44a0-93a2-27c03a0e324b)/Metadata('beginPosition')

The request returns the specific 'beginPosition' metadatum of the Product identified by the Id '30d01828-1b55-44a0-93a2-27c03a0e324b'.

## 5.8. Querying the products (paging)

The URI to be used for paging the list of products in the archive shall follow the syntax below:

https://catalogue.onda-dias.eu/dias-catalogue/Products?$skip=<N>&$top=<M>

where $skip=<N> is the number of records to skip before it retrieves records in a collection and $top=<M> is the maximum number of records to return.

Example:

https://catalogue.onda-dias.eu/dias-catalogue/Products?$skip=10&$top=50

this OData URI allows to list 50 products skipping the first 10.

## 5.9.Filter the products on time-based criteria

### 5.9.1. Filtering the Products by Creation date

Example:

Select products created from 10 of May 2018 00:00:00

https://catalogue.onda-dias.eu/dias-catalogue/Products?$search="creationDate:[2018-05-10T00:00:00.000Z%20TO%20*]"

Select products created from 1 of June to 1 of July 2018

https://catalogue.onda-dias.eu/dias-catalogue/Products?$search="creationDate:[2018-06-01T00:00:00.000Z%20TO%202018-06-08T00:00:00.000Z]"

### 5.9.2. Filtering the Products by Sensing Time (start and stop)

The keywords that can be used to filter products on sensing time are:

- ***beginPosition****:* a time interval search based on the Sensing Start Time of the products.
- ***endPosition****:* a time interval search based on the Sensing Stop Time of the products.

Example:

Search every products having sensing from 1 of May to today

https://catalogue.onda-dias.eu/dias-catalogue/Products?$search="beginPosition:[2018-05-01T00:00:00.000Z%20TO%20*]"

Count every products having sensing from 1 of May to 1 of June 2018

https://catalogue.onda-dias.eu/dias-catalogue/Products/$count?$search="beginPosition:[2018-05-01T00:00:00.000Z%20TO%202018-06-01T00:00:00.000Z]"

Search every products having sensing in the last day

https://catalogue.onda-dias.eu/dias-catalogue/Products/$count?$search="beginPosition:[NOW-1DAY TO NOW]"

Search every products ingested in the last month

https://catalogue.onda-dias.eu/dias-catalogue/Products/$count?$search="beginPosition:[NOW-30DAYS TO NOW]"

Search every products ingested in the last hour

https://catalogue.onda-dias.eu/dias-catalogue/Products/$count?$search="beginPosition:[NOW-1HOUR TO NOW]"

## 5.10. Filtering the Products using the file name

The products file name can be used for filtering the products. It shall be noticed that this query criteria is not based on the Metadata indexed from the products content but the criteria is search products matching a predefined string on the file name.

Select the S1 products:

https://catalogue.onda-dias.eu/dias-catalogue/Products?$search="name:S1*"

Select the S1A products:

https://catalogue.onda-dias.eu/dias-catalogue/Products?$search="name:S1A*"

Select the last 15 S1A products on the ONDA Catalogue:

https://catalogue.onda-dias.eu/dias-catalogue/Products?$search="name:S1A*"&$top=15

Select a specific product by its name:

https://catalogue.onda-dias.eu/dias-catalogue/Products?$search="name:S1A_IW_OCN__2SDV_20170904T063918_20170904T063943_018222_01EA1A_8E10.zip"

## 5.11. Download full product from its ID

To download a product the syntax is:

https://catalogue.onda-dias.eu/dias-catalogue/Products(Id)/$value

Example:

Download product identified by Id ‘00c093ff-b140-41ac-8ebd-1227f643466c’

https://catalogue.onda-dias.eu/dias-catalogue/Products(00c093ff-b140-41ac-8ebd-1227f643466c)/$value

Note: users will be requested to sign in to start the download. Attention should be paid to the fact that Username and Password are the same used to access the ONDA User Portal.

## 5.12. Sort products by creation date

Example:

Querying product sorted by descending creation date

https://catalogue.onda-dias.eu/dias-catalogue/Products?$orderby=creationDate%20desc

## 5.13. Search products with specific metadata

Here some examples of queries to search products with specific metadata.

Search products with cycle number equal to 10

https://catalogue.onda-dias.eu/dias-catalogue/Products?$search="cycleNumber:10"

Count products with cycle number between 1 to 60

https://catalogue.onda-dias.eu/dias-catalogue/Products/$count?$search="cycleNumber:[1%20TO%2060]"

Search products with last orbit direction descending

https://catalogue.onda-dias.eu/dias-catalogue/Products?$search="lastOrbitDirection:DESCENDING"

Count products with orbit number between 0 and 8000

https://catalogue.onda-dias.eu/dias-catalogue/Products/$count?$search="orbitNumber:%20[0%20TO%208000]"

Search products with cloud cover percentage between 0 and 70

https://catalogue.onda-dias.eu/dias-catalogue/Products?$search="cloudCoverPercentage:[0%20TO%2070]"

## 5.14. Discover the products over a predefined Area Of Interest (AOI): Geographical Search

It is possible to search products on the basis of a geographical area of interest, e.g. get the list of products over a geographic area delimitated by the polygon having vertices:

https://catalogue.onda-dias.eu/dias-catalogue/Products?$search="footprint:"Intersects(<geographic type>)""

The <geographic type> value can be expressed as a polygon according to the syntax described in the following paragraph.

### 5.14.1. **POLYGON**

The syntax for the <geographic type> value expressed as a polygon is:

<geographic type> =POLYGON((P1Lon P1Lat, P2Lon P2Lat, …, PnLon PnLat, P1Lon P1Lat))

where P1Lon and P1Lat are the Longitude and Latitude coordinates of the first point of the polygon in decimal degrees (DDD) format (e.g. 2.17403, 41.40338) and so on.

The coordinates of the last point of the polygon must coincide with the coordinates of the first point of the polygon.

The polygon describing the geographical area can have a maximum of 200 points that must be within the area defined by (180/85, -180/-85).

Example:

Search every products having polarization mode VV covering the geographic area delimitated by the polygon having vertices:

-72.53804252885192%20-14.490645904156324,101.96933536265675%20-14.490645904156324,101.96933536265675%2058.04372569919812,-72.53804252885192%2058.04372569919812,-72.53804252885192%20-14.490645904156324

https://catalogue.onda-dias.eu/dias-catalogue/Products?$search="footprint:"Intersects(POLYGON((-72.53804252885192%20-14.490645904156324,101.96933536265675%20-14.490645904156324,101.96933536265675%2058.04372569919812,-72.53804252885192%2058.04372569919812,-72.53804252885192%20-14.490645904156324)))""