

Environmental Data Retrieval (EDR)

For Visualizing CoverageJSON On Maplibre-gl

Nutthapol Jansuri | Sattawat Arab
Vallaris Maps

About Speakers



Nutthapol Jansuri

Head of Application Development

Email: nutthapol.j@i-bitz.co.th

Sattawat Arab

Head of Geoinformatics

Email: sattawat.a@i-bitz.co.th

Topics

- 01 Create OGC API Environmental Data Retrieval (EDR)**

- 02 Visualizing CoverageJSON (EDR Data) on Maplibre-gl**

01

Create OGC API Environmental Data Retrieval (EDR)

Document

<https://github.com/VallarisMapsPlatforms/FOSS4G-ASIA-2023-EDR>

Contents

01 Introduction to OGC API - EDR

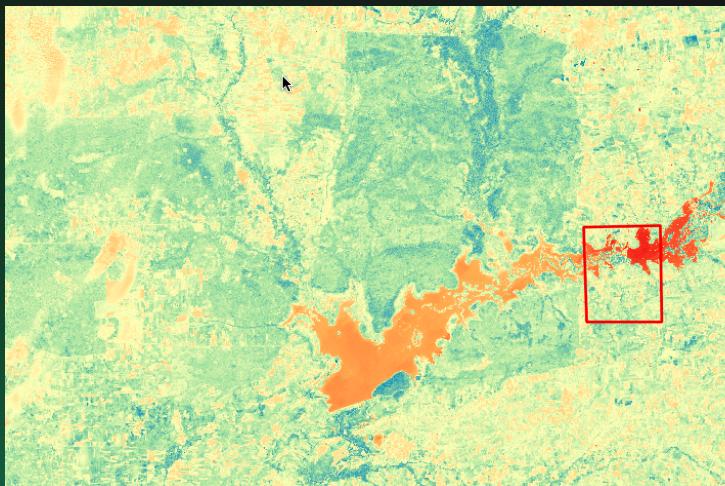
02 Example using OGC API - EDR

03 Request EDR data

04 Make your OGC API - EDR

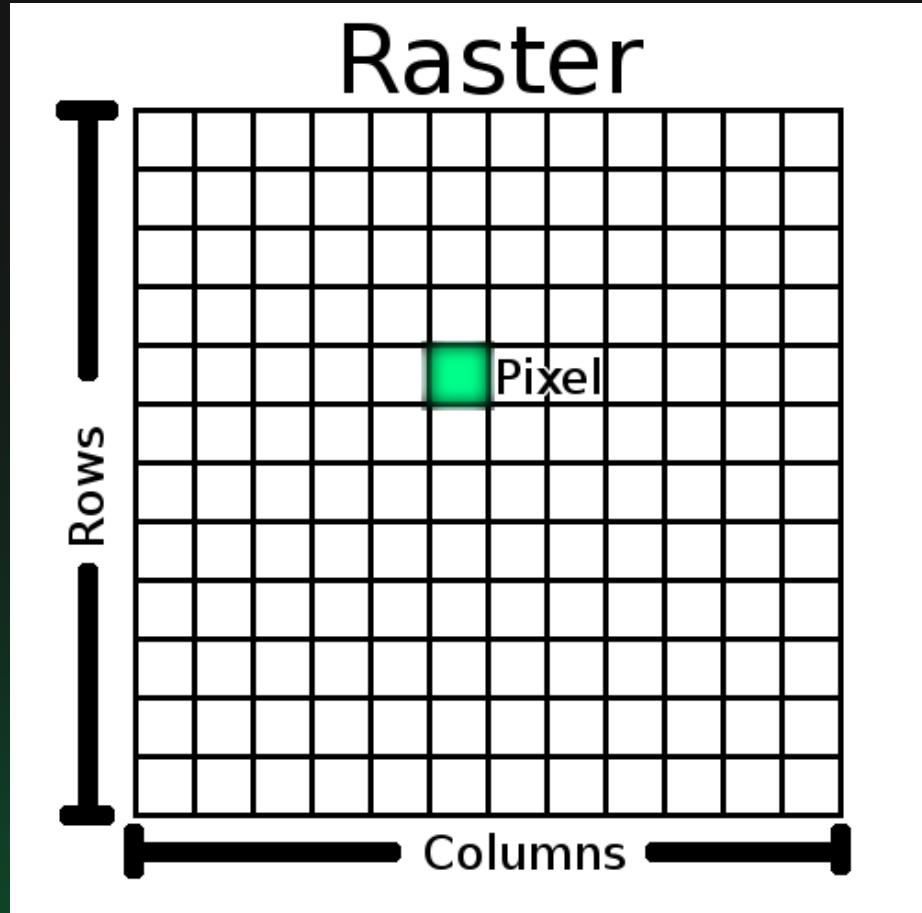
OGC API - EDR (ENVIRONMENTAL DATA RETRIEVAL)

OGC API - EDR (Environmental Data Retrieval) is a standard developed by the Open Geospatial Consortium (OGC). It modern way for access raster analysis ready data (ARD). Users have the capability to request data using services Application Programming Interface (API). EDR is patterns of structure CoverageJSON. CoverageJSON same like JSON format. Users can specific area or bounding box to retrieve raster data.



See response : <https://ogcapi.ogc.org/edr/>

Raster data & CoverageJSON



```
{  
  "type": "Coverage",  
  "domain": {  
    "type": "Domain",  
    "domainType": "Grid",  
    "axes": {  
      "x": {  
        "start": "min_longitude",  
        "stop": "max_longitude",  
        "num": "num of rows"  
      },  
      "y": {  
        "start": "min_latitude",  
        "stop": "max_latitude",  
        "num": "num of cols"  
      }  
    }  
    "ranges": {  
      "data": {  
        "type": "NdArray",  
        "dataType": "float",  
        "axisNames": [  
          "t",  
          "y",  
          "x"  
        ],  
        "shape": [  
          1,  
          "num of rows",  
          "num of cols"  
        ],  
        "values": [  
          "pixel of value"  
        ]  
      }  
    }  
  }  
}
```



Info



Tags



Servers



Search

[GET /conformance](#)[GET /collections](#)

Collection metadata ^

[GET /collections/{collectionId}](#)

Instance metadata ^

[GET /collections/{collectionId}/instances](#)

Collection data queries ^

[GET /collections/{collectionId}/position](#)[GET /collections/{collectionId}/radius](#)[GET /collections/{collectionId}/area](#)[GET /collections/{collectionId}/cube](#)[GET /collections/{collectionId}/trajectory](#)[GET /collections/{collectionId}/corridor](#)[GET /collections/{collectionId}/items](#)[GET /collections/{collectionId}/items/{itemId}](#)[GET /collections/{collectionId}/locations](#)[GET /collections/{collectionId}/locations/{locId}](#)

Instance data queries ^

[GET /collections/{collectionId}/instances/{instanceId}/position](#)[GET /collections/{collectionId}/instances/{instanceId}/radius](#)[GET /collections/{collectionId}/instances/{instanceId}/area](#)[GET /collections/{collectionId}/instances/{instanceId}/cube](#)

7185 text/html:
 7186 schema:
 7187 type: string
 7188 /collections/{collectionId}/cube:
 7189 get:
 7190 tags:
 7191 - Collection data queries
 7192 summary: Query end point for Cube queries of
 7193 collection {collectionId} defined by a cube
 7194 description: Return the data values for the data Cube
 7195 defined by the query parameters
 7196 operationId: GetDataForCube
 7197 parameters:
 7198 - name: collectionId
 7199 in: path
 7200 description: Identifier (id) of a specific
 7201 collection
 7202 required: true
 7203 schema:
 7204 type: string
 7205 - name: bbox
 7206 in: query
 7207 description: |-
 7208 Only features that have a geometry that
 7209 intersects the bounding box are selected.
 7210 The bounding box is provided as four or six
 7211 numbers, depending on whether the
 7212 coordinate reference system includes a vertical
 7213 axis (height or depth):
 7214 * Lower left corner, coordinate axis 1
 7215 * Lower left corner, coordinate axis 2
 7216 * Minimum value, coordinate axis 3 (optional)
 7217 * Upper right corner, coordinate axis 1
 7218 * Upper right corner, coordinate axis 2
 7219 * Maximum value, coordinate axis 3 (optional)
 7220 The coordinate reference system of the values is
 7221 WGS 84 longitude/latitude
 7222 (<http://www.opengis.net/def/crs/OGC/1.3/CRS84>)
 7223 unless a different coordinate
 7224 reference system is specified in the parameter
 7225 `bbox-crs`.
 7226 For WGS 84 longitude/latitude the values are in
 7227 most cases the sequence of
 7228 minimum latitude, maximum latitude, minimum longitude,
 7229 maximum longitude.

Last Saved: 6:21:17 pm - Jun 23, 2022

VALID

GET `/{collectionId}/radius`

Query end point for radius queries of collection {collectionId}

GET `/collections/{collectionId}/area`

Query end point for area queries of collection {collectionId} defined by a polygon

Parameters

Try it out

Name	Description
collectionId * required	Identifier (id) of a specific collection
string (path)	<input type="text" value="collectionId"/>
coords * required	Only data that has a geometry that intersects the area defined by the polygon are selected.
string (query)	The polygon is defined using a Well Known Text string following coords=POLYGON((x y,x1 y1,x2 y2,...,xn yn x y)) which are values in the coordinate system defined by the crs query parameter (if crs is not defined the values will be assumed to be WGS84 longitude/latitude coordinates). For instance a polygon that roughly describes an area that contains South West England in WGS84 would look like: coords=POLYGON((-6.1 50.3,-4.35 51.4,-2.6 51.6,-2.8 50.6,-5.3 49.9,-6.1,50.3)) see http://portal.opengeospatial.org/files/?artifact_id=25355 and https://en.wikipedia.org/wiki/Well-known_text_representation_of_geometry The coords parameter will only support 2D POLYGON definitions
coords	<input type="text" value="coords"/>
	Define the vertical level to return data from i.e. z-level

Example using OGC API - EDR

OGC API - EDR provides a flexible mechanism for accessing analysis ready data (ARD) that has been transformed into CoverageJSON using HTTP requests. accessible via the endpoints API.

Request EDR - Position data

```
https://url/core/api/edr/0.5/collections/{Collection}/position  
?coords=POINT(100.49723704881171 7.01349000774637)  
&crs=native  
&datetime=2023-08-04T00:00:00Z/2023-08-04T23:59:59Z
```

Response of request : <https://raw.githubusercontent.com/sattawatarab/FOSS4G-ASIA-2023-EDR/main/ogc-api-edr/response/position.json>

Request EDR - Radius data

```
https://url/core/api/edr/0.5/collections/{Collection}/radius  
?coords=POINT(100.49723704881171 7.01349000774637)  
&crs=native&within=100&within-units=M  
&datetime=2023-08-04T00:00:00Z/2023-08-04T23:59:59Z
```

Response of request : <https://raw.githubusercontent.com/sattawatarab/FOSS4G-ASIA-2023-EDR/main/ogc-api-edr/response/radius.json>

Request EDR - Radius data

```
https://url/core/api/edr/0.5/collections/{Collection}/radius  
?coords=POINT(100.49723704881171 7.01349000774637)  
&crs=native&within=100&within-units=M  
&datetime=2023-08-04T00:00:00Z/2023-08-04T23:59:59Z
```

Response of request : <https://raw.githubusercontent.com/sattawatarab/FOSS4G-ASIA-2023-EDR/main/ogc-api-edr/response/radius.json>

RRequest EDR - Cube data

```
https://url/core/api/edr/0.5/collections/NDVI/cube  
?bbox=100.4902167533075,7.001403324386018,100.5086064004459,7.014445221821404  
&crs=native  
&datetime=2023-08-01T00:00:00Z/2023-11-29T23:59:59Z
```

Response of request : <https://raw.githubusercontent.com/sattawatarab/FOSS4G-ASIA-2023-EDR/main/ogc-api-edr/response/cube.json>

RRequest EDR - Area data

```
https://url/core/api/edr/0.5/collections/NDVI/area  
?coords=POLYGON ((100.23377112820187 7.815674439517735, 100.23373919200209 7.815661560876779, 100.23377112820187  
7.815674439517735))  
&crs=native  
&datetime=2023-08-01T00:00:00Z/2023-11-29T23:59:59Z
```

Response of request : <https://raw.githubusercontent.com/sattawatarab/FOSS4G-ASIA-2023-EDR/main/ogc-api-edr/response/area.json>

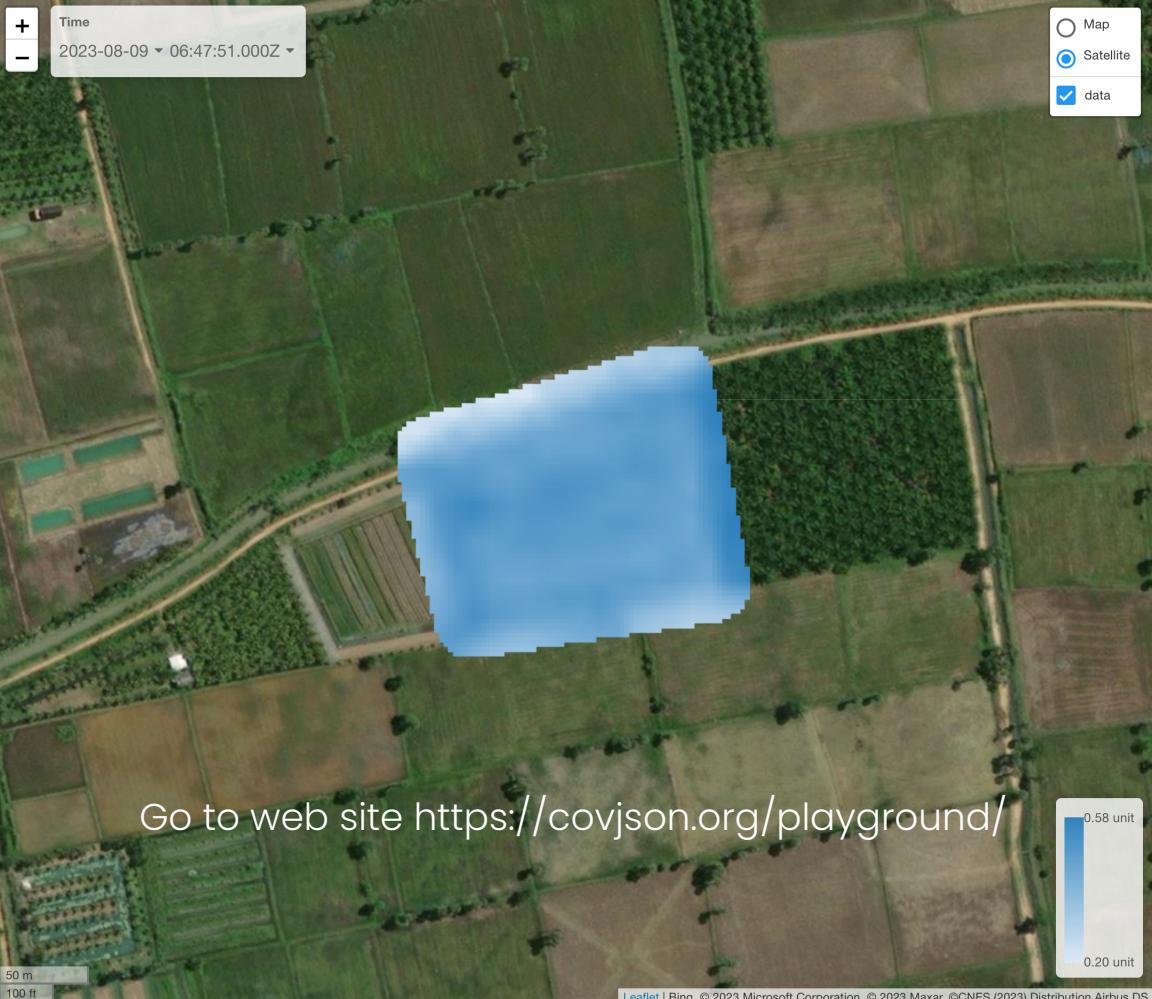
Display Data- CoverageJSON

 CoverageJSON

Playground Cookbook Tools Specification

+ Time
2023-08-09 ▾ 06:47:51.000Z ▾

Map
 Satellite
 data



Go to web site <https://covjson.org/playground/>

Leaflet | Bing, © 2023 Microsoft Corporation, © 2023 Maxar, ©CNES (2023) Distribution Airbus DS

50 m
100 ft

Examples

```
</> JSON | ? Help
```

```
1 {
2   "type": "Coverage",
3   "domain": {
4     "type": "Domain",
5     "domainType": "Grid",
6     "axes": {
7       "x": {
8         "start": 100.23365883304339,
9         "stop": 100.2354068330434,
10        "num": 76
11      },
12      "y": {
13        "start": 7.81602278618718,
14        "stop": 7.81450478618718,
15        "num": 66
16      },
17      "t": {
18        "values": [
19          "2023-08-09T06:47:51Z"
20        ]
21      }
22    },
23    "referencing": [
24      {
25        "coordinates": [
26          "x",
27          "y"
28        ],
29        "system": {
30          "type": "GeographicCRS",
31          "id": "http://www.opengis.net/def/crs/OGC/1.3/CRS:EPSG:4326"
32        }
33      },
34      {
35        "coordinates": [
36          "t"
37        ],
38        "system": {
39          "type": "TemporalRS",
40          "calendar": "Gregorian"
41        }
42      }
43    ]
44 }
```



รายละเอียดแปลง



พักอุจง
8 ไร่ 3 งาน 75.72 ตารางวา
ต.บ้านขาว อ.ระโนด จ.สงขลา
47N 636115E 864070N



ช่วงฤดูกาล

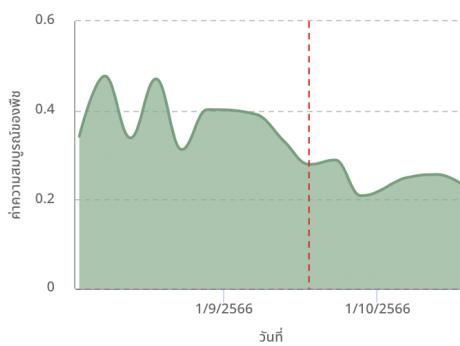
เครื่องมือวิเคราะห์ข้อมูล

ติดตามความสมบูรณ์ของพืช (1/2566)



ค่าเฉลี่ย 0.279
ค่าความสมบูรณ์ของพืชอยู่ในเกณฑ์
ดี
บันเดลงว่า พืชมีความสมบูรณ์หรือหนา...

กราฟแสดงค่าเฉลี่ยความสมบูรณ์ของพืชแต่ละวัน



ค่าอธิบายค่าความสมบูรณ์ของพืช NDVI

< 0.20	พื้นที่เปลือก หรือ พื้นที่เต็มไปด้วย...
0.20 - 0.30	พืชมีความสมบูรณ์หรือหนาแน่นต่ำ
0.30 - 0.40	พืชมีความสมบูรณ์หรือหนาแน่นปาน...
0.40 - 0.50	พืชมีความสมบูรณ์หรือหนาแน่นปาน...
0.50 - 0.60	พืชมีความสมบูรณ์หรือหนาแน่นต่อ...
> 0.60	พืชมีความสมบูรณ์หรือหนาแน่นสูง
-	พื้นที่แบบป่าคลุม

https://dragonfly.gistda.or.th



Lab : Make your OGC API - EDR

1. Install Visual Studio Code

- Open [Visual Studio Code](<https://code.visualstudio.com/download/>) in your web browser.

2. Install Postman

- Open [Postman](<https://www.postman.com/downloads/>) in your web browser.

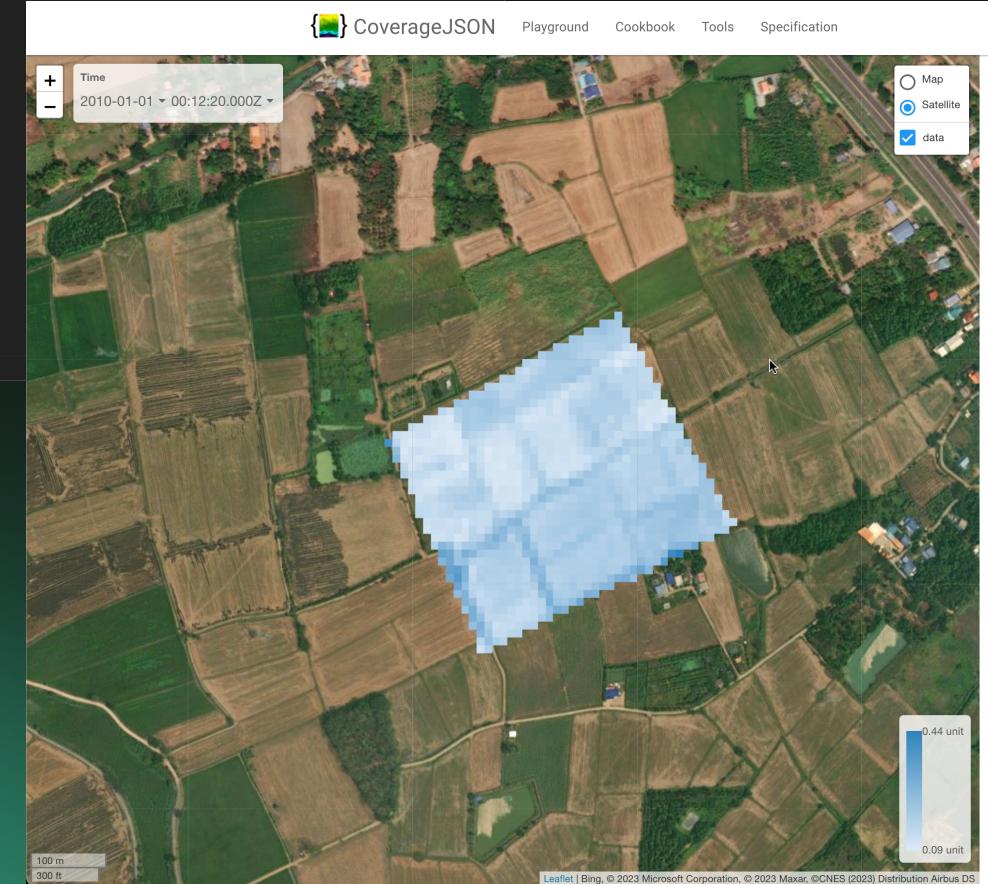
- 3 - Open [Google Colab](<https://colab.research.google.com/>) in your web browser.

2. Create Folder "foss4g":

- In the left sidebar, click on the folder icon.
- Click on the "New Folder" button.
- Name the folder "foss4g."

3. **Upload Data from Folder "lab" into Folder "Foss4g" in Colab:**

Display Data CoverageJSON



Request data using Postman

02

Visualizing CoverageJSON (EDR Data) On Maplibre-gl

Visualization CovJSON

Visualizing CovJSON data requires robust tools and libraries capable of handling its complexity and enabling users to derive meaningful interpretations.



[https://github.com/Reading-eScience-Centre/
leaflet-coverage](https://github.com/Reading-eScience-Centre/leaflet-coverage)



[https://github.com/Reading-eScience-Centre/
webworldwind-covjson](https://github.com/Reading-eScience-Centre/webworldwind-covjson)

In the present, there are only two libraries that support the visualization of CovJSON,
namely leaflet.js and web worldwind.

See reference : <https://covjson.org/tools/>

Visualization CovJSON

MapLibre GL is a powerful open-source JavaScript library for displaying interactive maps, built as a fork of Mapbox GL JS. Leveraging MapLibre GL's capabilities to visualize COVJSON data on interactive maps provides a dynamic way to explore spatial coverages.



<https://maplibre.org/>

While MapLibre GL does not have a built-in native support for covJSON, you can utilize its functionalities combined with other JavaScript libraries or custom code to achieve this visualization.

Contents

- 01 Init Next application
- 02 Example for use covjson-reader package
- 03 Convert CovJSON to Raster(PNG)
- 04 Convert CovJSON to Vector(GeoJSON)
- 05 Compare between Raster(PNG) & Vector(GeoJSON)

Source code

<https://github.com/VallarisMapsPlatforms/FOSS4G-ASIA-2023-EDR/tree/main/covjson-on-maplibregl>

Thank you