

End point: <https://fkd0kqznyh.execute-api.us-east-1.amazonaws.com>

Path /datasets

1. **GET /datasets** ใช้สำหรับดึง Project ทั้งหมดที่มีในระบบ

Query Param: ?user_id=<เลข id ของ user> ใช้สำหรับกรองเอาแค่ Project ที่มี user คน ๆ นั้นเป็นเจ้าของ

Example JSON response :

```
1  {
2    "count": 3,
3    "datasets": [
4      {
5        "dataset_id": 1,
6        "name": "Noise Measurements",
7        "description": "Crowdsourced environmental noise data",
8        "owner": "admin",
9        "source": null,
10       "created_at": "2025-10-26 05:12:11.606048"
11     },
12     {
13       "dataset_id": 2,
14       "name": "Noise 2025",
15       "description": "Crowdsourced noise measurements",
16       "owner": "admin",
17       "source": null,
18       "created_at": "2025-10-26 05:45:30.295033"
19     },
20     {
21       "dataset_id": 3,
22       "name": "Noise 2025",
23       "description": "Crowdsourced noise measurements",
24       "owner": "admin",
25       "source": null,
26       "created_at": "2025-10-26 05:52:53.966173"
27     },
28     {
29       ...
30     },
31     ...
32   ]
33 }
```

2. **POST /datasets** ใช้สำหรับสร้าง Project ใหม่ในระบบ

Example JSON request:

```
1 {
2   "name": "Test dataset name",
3   "description": "This is for test only",
4   "owner_id": 1
5 }
```

Example JSON response with status 200 OK

```
1 {
2   "dataset_id": 49,
3   "workspace": "test_workspace_name",
4   "datastore": "test_workspace_name_store",
5   "message": "Workspace and datastore created successfully for Test workspace name"
6 }
```

3. **GET /datasets/{dataset_id}** ใช้สำหรับ query เพื่อดูแค่ Project ที่ต้องการ

Example JSON response with status 200 OK:

```
1 {
2   "count": 1,
3   "datasets": [
4     {
5       "dataset_id": 49,
6       "name": "Test workspace name",
7       "description": "This is for test only",
8       "owner": "admin",
9       "source": null,
10      "created_at": "2025-11-08 04:43:44.167451"
11    }
12  ]
13 }
```

4. GET /datasets/{dataset_id}/layers ใช้สำหรับดึง Layer ทั้งหมดของ Project นั้นๆ

Example JSON response:

```
1 {
2   "count": 4,
3   "layers": [
4     {
5       "layer_id": 47,
6       "name": "noise_measurements",
7       "geom": {
8         "type": "POINT",
9         "srid": 4326
10      },
11      "description": "Noise Measurements",
12      "timestamp": "2025-11-05T11:39:58.183942",
13      "schema": [
14        {
15          "field_name": "noise_level",
16          "data_type": "numeric",
17          "unit": "String",
18          "description": "String"
19        },
20        {
21          "field_name": "duration",
22          "data_type": "numeric",
23          "unit": "String",
24          "description": "String"
25        }
26      ]
27    },
28    {...}
29  ]
30 }
```

Response ที่ได้จะบอกว่า layer ใน project นี้มี schema ยังไง data type เป็นอะไรบ้าง ใช้ Geometry เป็นอะไรและใช้หน่วยวัดในแต่ละ column เป็นอะไร

5. POST /datasets/{dataset_id}/layers ใช้สำหรับสร้าง Layer ให้ Project นั้นๆ

Example JSON request:

```
1 {
2   "layer_name": "Test layer",
3   "title": "This is for test only",
4   "srid": 4326,
5   "geom_type": "POINT",
6   "fields": [
7     {"field_name": "noise_level", "data_type": "numeric", "unit": "String", "description": "String"},
8     {"field_name": "duration", "data_type": "numeric", "unit": "String", "description": "String"}
9   ]
10 }
```

Field สำคัญที่ต้องส่งมา:

srid: คือบอกว่าข้อมูลชุดนี้จะใช้มาตรฐานการวัดหน่วยพื้นที่ยังไง เช่นในตัวอย่างจะใช้ 4326 คือ lat กับ lon

fields: รับเป็น list ของ object ซึ่งข้างในต้องบอกว่า column นี้คืออะไร มี datatype เป็นอะไร ใช้หน่วยวัดเป็นอะไรพร้อมกับคำอธิบาย

geom_type : รับเป็น string ที่บอกว่า layer นี้จะใช้ geometry ประเภทอะไร

geom type ที่รองรับ:

'POINT', 'LINESTRING', 'POLYGON', 'MULTIPOINT', 'MULTILINESTRING', 'MULTIPOLYGON'

Data type ที่รองรับ:

Integer

bigint

numeric

real

double precision

text

varchar

timestamp

boolean

Example JSON response:

```
1 {
2   "layer_id": 67,
3   "dataset_id": 50,
4   "workspace": "test_work_space_again",
5   "fields": [
6     "noise_level",
7     "duration"
8   ],
9   "message": "Layer 'Test layer' created with 2 field(s) and published to GeoServer"
10 }
```

6. **GET /datasets/{dataset_id}/layers/{layer_id}** ใช้สำหรับดึงข้อมูลที่วัดได้ทั้งหมดจาก layer นั้นซึ่งจะได้ response มาเป็นรูปแบบ OGC Features

Example JSON response:

```
1 {
2   "type": "FeatureCollection",
3   "features": [
4     {
5       "type": "Feature",
6       "id": "test_layer_view.fid--16a5a8e7_19a61b56181_-7fe4",
7       "geometry": {
8         "type": "Point",
9         "coordinates": [
10          100.5231,
11          13.7367
12        ]
13      },
14      "geometry_name": "geom",
15      "properties": {
16        "id": 51,
17        "noise_level": 72.5,
18        "duration": 15.3,
19        "timestamp": "2025-11-08T05:34:02.962Z"
20      },
21      "bbox": [
22        100.5231,
23        13.7367,
24        100.5231,
25        13.7367
26      ]
27    }
28  ],
29  "totalFeatures": 1,
30  "numberMatched": 1,
31  "numberReturned": 1,
32  "timeStamp": "2025-11-08T05:34:07.706Z",
33  "crs": {
34    "type": "name",
35    "properties": {
36      "name": "urn:ogc:def:crs:EPSG::4326"
37    }
38  },
39  "bbox": [
40    100.5231,
41    13.7367,
42    100.5231,
43    13.7367
44  ]
45 }
```

Query Param: ?CQL_FILTER=<CQL Expression goes here> ใช้สำหรับสำหรับ query แบบระบุเงื่อนไขโดยใช้ CQL Query: https://docs.geoserver.org/2.27.x/en/user/tutorials/cql/cql_tutorial.html

โดย user สามารถ query เองได้โดยระบุ CQL Expression ใน CQL_FILTER query param

เช่น อยากรู้ข้อมูลที่มี noise_level มากกว่า 80 dB สามารถระบุใน URL ได้แบบนี้ **?CQL_FILTER=noise_level > 80** ก็จะได้อข้อมูลที่มี noise level อยู่ใน range ที่ต้องการ

7. POST datasets/{dataset_id}/layers/{layer_id} ใช้สำหรับสร้างข้อมูลใหม่ใน layer นั้น

ยังใช้งานไม่ได้ขอเวลาแก้ก่อน