ERDDAP: Easier access to scientific data

Rich Signell - USGS
Filipe Fernandes - IOOS

OOI DDCI Kickoff Meeting, October 30, 2018
What is ERDDAP?

A data brokerage service, reading from many different types of files, databases and services, and providing access via a single standardized interface

RESTful API for access in scientific analysis packages (Matlab, Python, R), web application developers (JavaScript), and by numerical modelers (Fortran, Bash)

Advanced search built-in, and also generates ISO and json-ld metadata records to allow search via sites like data.gov, and Google Dataset Search.

Widely used for delivery of “FAIR” data in the geoscience community (more than 50 server deployments worldwide)
USGS Integrated Ocean Observing System

11 Regional Associations

17 Federal Partners

Web Standards:

Grid: OPeNDAP
Sensor: ERDDAP
Image: WMS
Metadata: CSW
ERDDAP > Advanced Search

Directions: Specify as many or as few search criteria as you want, then click Search.
Only the datasets that match all of the search criteria will appear in the results.

Full Text Search for Datasets

Search for Datasets by Category

- protocol
- cdm_data_type
- institution
- iioss_category
- keywords
- long_name
- standard_name
- variableName

Search for Datasets that have Data within Longitude, Latitude, and Time Ranges

- Maximum Latitude
- Min and Max Longitude
- Minimum Latitude
- Minimum Time
- Maximum Time

Search
<table>
<thead>
<tr>
<th>Grid DAP Data</th>
<th>Subset</th>
<th>Table DAP Data</th>
<th>Make A Graph</th>
<th>WMS Files</th>
<th>Source Data Files</th>
<th>Title</th>
<th>Summary</th>
<th>FGDC, ISO, Metadata</th>
<th>Backround Info</th>
<th>RSS</th>
<th>E-mail</th>
<th>Institution</th>
<th>Dataset ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>set data graph files</td>
<td>CE01ISSM BUOY CTDPB3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE01ISSM-BUOY-001-CTDPB-FLORT</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE01ISSM MFN CTDPB2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE01ISSM-MFN-001-CTDPB-DOSTA</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE01ISSM NSIF CTDPB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE01ISSM-NSIF-001-CTDPB-DOSTA</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE01ISSM NSIF FLORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE01ISSM-NSIF-001-FLORT</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE01ISSM NSIF SUNA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE01ISSM-NSIF-002-NUTNR</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE01ISSP CSPP CTDPF Recovered</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE01ISSP-CSPP-002-CTDPF</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE01ISSP CSPP CTDPF Telemetry</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE01ISSP-CSPP-001-CTDPF</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE01ISSP CSPP FLORT Recovered</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE01ISSP-CSPP-001-FLORT</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE01ISSP CSPP FLORT Telemetry</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE01ISSP-CSPP-001-FLORT</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE01ISSP CSPP NUTNR Recovered</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE01ISSP-CSPP-002-NUTNR</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE01ISSP CSPP OPTAA Recovered</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE01ISSP-CSPP-002-OPTAA</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE02SHSM BUOY METBK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE02SHSM-BUOY-001-METBK</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE02SHSM NSIF CTDPB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE02SHSM-NSIF-001-CTDPB</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE02SHSM NSIF FLORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE02SHSM-NSIF-001-FLORT</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE02SHSM NSIF NUTNR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE02SHSM-NSIF-001-NUTNR</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE02SHSP CSPP NUTNR Recovered</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE02SHSP-CSPP-002-NUTNR</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE04OSSM BUOY METBK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE04OSSM-BUOY-001-METBK</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE04OSSM NSIF CTDPB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE04OSSM-NSIF-001-CTDPB</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE04OSSM NSIF FLORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE04OSSM-NSIF-001-FLORT</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE04OSSM NSIF NUTNR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE04OSSM-NSIF-001-NUTNR</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE06ISSM BUOY CTDPB3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE06ISSM-BUOY-001-CTDPB-FLORT</td>
</tr>
<tr>
<td>set data graph files</td>
<td>CE06ISSM MFN CTDPB2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coastal and Global</td>
<td>CE06ISSM-MFN-001-CTDPB-DOSTA</td>
</tr>
</tbody>
</table>
ERDDAP > tabledap > Data Access Form

Dataset Title: CE06ISSP CSPP CTDPF Telemetry

Institution: Coastal and Global Scales Nodes (CGSN) (Dataset ID: CE06ISSP-CSPP-001-CTDPF)

Variable Options:
- [ ] crs
- [ ] station (station identifier)
- [ ] time (UTC)
- [ ] latitude (degrees_north)

Optional Constraints:
- Constraint #1: > <= 2018-07-01T00:00:00Z
- Constraint #2: > <= 2018-07-15T00:00:00Z

Minimum Values:
- 2147483647

Maximum Values:
- 2018-07-26T07:45:49Z
- 2018-07-20T07:58:35Z

File Formats:
- asc: View OPeNDAP-style ISO-8859-1 comma-separated text.
- csv: Download an ISO-8859-1 csv file with line 1: names; line 2: units; ISO 8601 times.
- csv0: Download an ISO-8859-1 csv file with column names or units. Times are ISO 8601 strings.
- dds: View the dataset's metadata via an ISO-8859-1 OPeNDAP Dataset Attribution Structure (DAS).
- dds: View the dataset's structure via an ISO-8859-1 OPeNDAP Dataset Descriptor Structure (DDS).
- dods: OPeNDAP clients use this to download the data in the DODS binary format.
- esriCsv: Download an ISO_8859_1.csv file for ESRI's ArcGIS 9.x and below (separate date and time columns).
- fgdc: View the dataset's UFT-8 FGDC .xml metadata.
- geoJson: Download longitude,latitude,otherColumns data as a UTF-8 GeoJSON .json file.
- graph: View a Make A Graph webpage.
- help: View a webpage with a description of tabledap.
- html: View an OPeNDAP-style HTML Data Access Form.
- htmlTable: View a UTF-8 .html webpage with the data in a table. Times are ISO 8601 strings.
- json: View a tabular-like UTF-8 .JSON file (missing value = "null", times are ISO 8601 strings).
- jsonCSV: View a UTF-8 .JSON Lines CSV file (missing value = "null", times are ISO 8601 strings).
- jsonKVP: View a UTF-8 .JSON Lines file with Key-Value pairs (missing value = "null", times are ISO 8601 strings).
- mat: Download a MATLAB binary file.
- htmlTable: View a UTF-8 .html webpage with the data in a table. Times are ISO 8601 strings.

Just generate the URL: [Documentation / Bypass this form]
IOOS Access 03: Accessing IOOS and OOI glider data from the IOOS Glider DAC via ERDDAP and erddapy

Emilio Mayorga, University of Washington & NANOOS. 2018-8-21.

IOOS Glider Data Assembly Center (DAC): "The mission of the Glider DAC is to provide glider operators with a simple process for submitting glider data sets to a centralized location, enabling the data to be visualized, analyzed, widely distributed via existing web services and the Global Telecommunications System (GTS) and archived at the National Centers for Environmental Information (NCEI)." A lot of OOI glider data are also available on the Glider DAC (at least through March 2018).

Glider DAC data are available for access online via user applications such as the Glider Map as well as via other applications and data services for machine-to-machine access. ERDDAP (see here and here) is one of those applictions. "ERDDAP is a data server that gives you a simple, consistent way to download data in the format and the spatial and temporal coverage that you want. ERDDAP is a web application with an interface for people to use. It is also a RESTful web service that allows data access directly from any computer program (e.g. Matlab, R, or webpages)."
A typical ERDDAP RESTful URL looks like:

https://data.ioos.us/gliders/erddap/tabledap/whoi_406-20160902T1700.mat?depth,latitude,longitude,salinity,temperature,time&time>=2016-07-10T00:00:00Z&time=<2017-02-10T00:00:00Z &latitude>=38.0&latitude=<41.0&longitude>=72.0&longitude=<69.0

Let's break it down to smaller parts:

- **server**: https://data.ioos.us/gliders/erddap/
- **protocol**: tabledap
- **dataset_id**: whoi_406-20160902T1700
- **response**: mat
- **variables**: depth,latitude,longitude,temperature,time
- **constraints**:
  - time>=2016-07-10T00:00:00Z
  - time=<2017-02-10T00:00:00Z
  - latitude>=38.0
  - latitude=<41.0
  - longitude>=72.0
  - longitude=<69.0

This Jupyter notebook is based on a notebook from the IOOS gallery, at http://ioos.github.io/notebooks_demos/notebooks/2018-03-01-erddapy/. A newer version of that notebook is available at https://pyoceans.github.io/erddapy/quick_intro.html, and an even newer version at this gist from https://github.com/ocelpal.

```
In [1]: import pandas as pd
    from erddapy import ERDDAP
    from erddapy.utilities import urlopen
    import hvplot.pandas
```
In [5]:
min_time = '2018-07-01T00:00:00Z'
max_time = '2018-07-15T00:00:00Z'
min_lon, max_lon = -127, -123.75
min_lat, max_lat = 43, 48
standard_name = 'sea_water_practical_salinity'

kw = {
    'standard_name': standard_name,
    'min_lon': min_lon, 'max_lon': max_lon, 'min_lat': min_lat, 'max_lat': max_lat,
    'min_time': min_time, 'max_time': max_time,
}

In [6]:
search_url = e.get_search_url(response='csv', **kw)
search_df = pd.read_csv(urlopen(search_url))
search_df = search_df[['Institution', 'Dataset ID', 'tabledap']]

Out[6]:

<table>
<thead>
<tr>
<th>Institution</th>
<th>Dataset ID</th>
<th>tabledap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coastal and Global Scales Nodes (CGSN)</td>
<td>CE01ISSM-BUOY-001-CTDBP-FLORT</td>
<td><a href="https://cgora.coas.oregonstate.edu/erddap/tabledap/ce01iissm-buoy-001-ctdbp-flort">https://cgora.coas.oregonstate.edu/erddap/tabledap/ce01iissm-buoy-001-ctdbp-flort</a></td>
</tr>
</tbody>
</table>
Construct the ERDDAP URL to get the data

```python
In [8]:
e.dataset_id = dataset_id
e.constraints = {'time': 'min_time', 'time': 'max_time'}
e.response = 'csv'
e.variables = ['time', e.get_var_by_attr(dataset_id=dataset_id, standard_name=standard_name)[0]]
print(e.get_download_url())

https://cogoms.coas.oregonstate.edu/erddap/tabledap/CE06ISSM-BUOY-001-CTDBP-FLORT.csv?time,psu&time>=1530403200.0&time<=1531612800.0
```

Read the data into Xarray

```python
In [9]:
ds = e.to_xarray(decode_times=True)
ds = ds.swap_dims(['row', 'time'])
[dvars.plot() for var in ds.data_vars];
```
In [11]: ds

Out[11]: <xarray.Dataset>
Dimensions: (time: 168)
Coordinates:
  * time        (time) datetime64[ns] 2018-07-01T00:30:14.438000128 ... 2018-07-14T22:30:16.593999872
Data variables:
  psu          (time) float64 ...
Attributes:
  acknowledgement: National Science Foundation
  cdm_data_type: TimeSeries
  cdm_timeseries_variables: latitude, longitude, feature_type_instan...
  comment: Mooring ID: CE06ISSM-00009
  Conventions: CF-1.6,ACDD-1.3, COARDS
  creator_email: cwingard@coas.oregonstate.edu
  creator_name: Christopher Wingard
  creator_url: http://oceanobservatories.org
  date_created: 2018-10-29T06:55:00Z
  date_issued: 2018-10-29T06:55:00Z
  date_metadata_modified: 2018-10-29T06:55:00Z
  date_modified: 2018-10-29T06:55:00Z
  defaultGraphQuery: time,temperature&time>=max(time)-7day&d...
  featureType: TimeSeries
  geospatial_bounds: POINT(-124.272 47.133)
  geospatial_bounds_crs: EPSG:4326
  geospatial_lat_resolution: 0.0
  geospatial_lat_units: degrees_north
  geospatial_lon_resolution: 0.0
  geospatial_lon_units: degrees_east
  geospatial_vertical_positive: down
  geospatial_vertical_resolution: 0
  geospatial_vertical_units: m
  history: 2018-10-29T06:55:00Z - pyaxiom - File cr...
my_list = []
for dataset_id in ctdbp['Dataset ID'].values:
    e.dataset_id = dataset_id
    e.variables = ['time', e.get_var_by_attr(dataset_id=dataset_id, standard_name=standard_name)[0]]
    try:
        ds = e.to_xarray(decoder_times=True)
        ds = ds.swap_dims({'row': 'time'})
        df_list.append(ds)
        print(dataset_id)
        [ds[var].plot() for var in ds.data_vars]
    except:
        pass
    if len(df_list) == 5:
        break

CE01ISSM-BUOY-001-CTDBP-FLORT
CE01ISSM-MFN-001-CTDBP-DOSTA
CE01ISSM-NSIF-001-CTDBP-DOSTA
CE02SHSM-NSIF-001-CTDBP
CE04OSSM-NSIF-001-CTDBP
Plotting Time Series Data from NetCDF

In this example we show how to programatically download and work with OOI NetCDF time series data. We will use data from the Global Argentnine Basin Flanking Subsurface Mooring A Mooring Riser CTD at 30 m. You will learn:

- how to find the data you are looking for
- how to use the machine to machine API to request data
- how to load the NetCDF data into your notebook, once the data request has completed
- how to plot data

For the instrument in this example, you will need the Reference Designator, Stream and Data Delivery Method to make the request to the M2M API. More information about the instrument can be found here: [http://ooi.visualocean.net/instruments/view/GA03FLMA-RIM01-02-CTDMOG040](http://ooi.visualocean.net/instruments/view/GA03FLMA-RIM01-02-CTDMOG040)
reproducible-notebooks

ERDDAP_timeseries_explorer
Simple interactive ERDDAP time series explorer using Jupyter Widgets

- Jupyter Notebook
- 1 star
- MIT
- Updated 2 days ago

ERDDAP_glider_search
Find and display glider data using ERDDAP

- Jupyter Notebook
- 1 star
- MIT
- Updated on May 22

OBIS_and_R
Using OBIS with R in Jupyter
Explore ERDDAP timeseries data using Jupyter Widgets

Inspired by Jason Grout's excellent ESIP Tech Dive talk on "Jupyter Widgets", this notebook uses the ipyleaflet and bqplot widgets to interactively explore the last two weeks of time series data from an ERDDAP Server. Select a standard_name from the list, then click a station to see the time series.

NOTE: To access a protected ERDDAP endpoint is protected, you can add a ~/.netrc file like this:

```
machine cgoms.coas.oregonstate.edu
login <username>
password <password>
```

In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
import pendulum
```

ipyleaflet and bqplot are both Jupyter widgets, so can interact with Python like any other widget. Since we want to click on a map in a notebook and get an interactive time series plot, they are perfect tools to use here.

In [3]:
```
import ipyleaflet as ipyl
import bqplot as bq
import ipywidgets as ipyw
```

To make working with ERDDAP simpler, we use erddapy, a high-level python interface to ERDDAP's RESTful API.

In [4]:
```
from erddapy import ERDDAP
from erddapy.utilities import urlopen
```
CGSN Dashboard

System Monitoring Plots

- Battery 1-4 Current
- GPS Lat/Long
- WT 1 + H2
- Percent Charge
CE01ISSM-MFD37-00-DCLENG000-cg_dcl_eng_dcl_gps-telemetered-deployment0007-tabledap

ERDDAP Data Server at OOI

Dataset created Nov 7, 2017
Dataset updated Nov 7, 2017

Dataset provided by
Ocean Observatories Initiative

License
The COL OOI Program is not responsible for the use of the data it provides. The reliability, quality and completeness of data obtained through OOI are intended to be used in an education or research context. It is assumed that outages and errors can occur and are dealt with by the users of the data. These data and software are not for use in operational or decision-making settings. COL makes reasonable efforts to ensure that the data provided are accurate. However, there may be no Quality Control (QC) performed on data acquired and provided through the OOI program, and there may be no Quality Assurance (QA) provided on information on those data sets. If QC/QA is performed, it is described in the metadata. The OOI program both produces and through collaborations within the geosciences community, gains access to data sets which may be redistributed either directly or indirectly at no cost. With regard to data distribution, all users must comply with any applicable U.S. export laws and regulations.

Time period covered Apr 19, 2017 - Oct 12, 2017
The Dataset's Variables and Attributes

<table>
<thead>
<tr>
<th>Row Type</th>
<th>Variable Name</th>
<th>Attribute Name</th>
<th>Data Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>attribute</td>
<td>NC_GLOBAL</td>
<td>cdm_data_type</td>
<td>String</td>
<td>Point</td>
</tr>
<tr>
<td>attribute</td>
<td>NC_GLOBAL</td>
<td>collection_method</td>
<td>String</td>
<td>telemetered</td>
</tr>
<tr>
<td>attribute</td>
<td>NC_GLOBAL</td>
<td>Conventions</td>
<td>String</td>
<td>CF-1.6, COARDS, ACDD-1.3, NCGSv1.0</td>
</tr>
<tr>
<td>attribute</td>
<td>NC_GLOBAL</td>
<td>creator_name</td>
<td>String</td>
<td>Ocean Observatories Initiative</td>
</tr>
<tr>
<td>attribute</td>
<td>NC_GLOBAL</td>
<td>creator_url</td>
<td>String</td>
<td><a href="http://oceanobservatories.org/">http://oceanobservatories.org/</a></td>
</tr>
<tr>
<td>attribute</td>
<td>NC_GLOBAL</td>
<td>date_created</td>
<td>String</td>
<td>2017-11-07T19:47:21.731626</td>
</tr>
<tr>
<td>attribute</td>
<td>NC_GLOBAL</td>
<td>date_modified</td>
<td>String</td>
<td>2017-11-07T19:47:21.731631</td>
</tr>
<tr>
<td>attribute</td>
<td>NC_GLOBAL</td>
<td>Easternmost_Easting</td>
<td>double</td>
<td>-124.0954</td>
</tr>
<tr>
<td>attribute</td>
<td>NC_GLOBAL</td>
<td>featureType</td>
<td>String</td>
<td>Point</td>
</tr>
<tr>
<td>attribute</td>
<td>NC_GLOBAL</td>
<td>geospatial_lat_max</td>
<td>double</td>
<td>44.65692</td>
</tr>
<tr>
<td>attribute</td>
<td>NC_GLOBAL</td>
<td>geospatial_lat_min</td>
<td>double</td>
<td>44.65686</td>
</tr>
<tr>
<td>attribute</td>
<td>NC_GLOBAL</td>
<td>geospatial_lat_resolution</td>
<td>double</td>
<td>0.1</td>
</tr>
<tr>
<td>attribute</td>
<td>NC_GLOBAL</td>
<td>geospatial_lat_units</td>
<td>String</td>
<td>degrees_north</td>
</tr>
<tr>
<td>attribute</td>
<td>NC_GLOBAL</td>
<td>geospatial_lon_max</td>
<td>double</td>
<td>-124.0954</td>
</tr>
</tbody>
</table>
## ERDDAP vs M2M

<table>
<thead>
<tr>
<th>Feature</th>
<th>ERDDAP</th>
<th>M2M</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESTful Interface</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Widely used by community</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Advanced Search</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ISO, JSON-LD Metadata</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Output types</td>
<td>40 (NetCDF, json, mat, csv ...)</td>
<td>2 (NetCDF, json)</td>
</tr>
</tbody>
</table>
Summary

- ERDDAP allows easier access to OOI Data
- ERDDAP makes OOI “FAIR” (Findable, Accessible, Interoperable, Reusable)
- ERDDAP works with the existing system
- Easier access to OOI Data results in more use by researchers
- Easier access to OOI Data allows more efficient data analysis, leaving more time for actual science
- Let’s make ERDDAP a first class citizen in OOI
- Let’s build OOI science and end-user applications using ERDDAP as a backend for data search and access