ioos/**erddapy**

Python interface for ERDDAP

18 Contributors | 70 Used by | 70 Stars | 29 Forks

docs: https://ioos.github.io/erddapy

source: https://github.com/ioos/erddapy

Scope

- What is erddapy:
  - ERDDAP URL builder
  - Tool for other tools
  - Use python to programmatically return anything that ERDDAP RESTful API can do
  - Some basic data export: netCDF, CSV, xarray, etc.
- What it isn’t:
  - A particular server/service/region data browser
  - A (user friendly) data exploration tool
  - A (complete) multiple ERDDAP server search

Dependency graph

Dependencies | Dependents | Dependabot

Export SBOM

Repositories that depend on **erddapy**

70 Repositories | **12 Packages** | Owner

- pyo-oracle
- ceotr-erddap-proxy
- euroargodev / **VirtualFleet** virtualfleet — ☆ 6 — 2
- axiom-data-science / **intake-erddap** — ☆ 0 — 1
- app-common
- oceanmodeling / **searvey** — ☆ 16 — 6
- modaat-console
- lifewatch / **bpnsdata** — ☆ 2 — 0
- ioos / **gliderpy** — ☆ 5 — 10
- rbardaji / **mooda** — ☆ 16 — 3
- axiom-data-science / **ocean_data_gateway** ocean-data-gateway — ☆ 5 — 3
- euroargodev / **argopy** — ☆ 159 — 37

```python
from erddapy import ERDDAP

server = "https://gliders.ioos.us/erddap"

e = ERDDAP(server=server)
e.dataset_id = "whoi_406-20160902T1700"
e.protocol = "tabledap"
e.variables = [
    "depth",
    "latitude",
    "longitude",
    "salinity",
    "temperature",
    "time",
]

e.constraints = {
    "time>=": "2016-09-03T00:00:00Z",
    "time<=": "2017-02-10T00:00:00Z",
    "latitude>=": 38.0,
    "latitude<=": 41.0,
    "longitude>=": -72.0,
    "longitude<=": -69.0,
}

df = e.to_pandas(
    index_col="time (UTC)",
    parse_dates=True,
).dropna()
```

```python
import matplotlib.dates as mdates

fig, ax = plt.subplots(figsize=(17, 2))
cs = ax.scatter(
    df.index,
    df["depth (m)"],
    s=15,
    c=df["temperature (Celsius)"],
    marker="o",
    edgecolor="none",
)

ax.invert_yaxis()
ax.set_xlim(df.index[0], df.index[-1])
xfmt = mdates.DateFormatter("%H:%Mh\n%d-%b")
ax.xaxis.set_major_formatter(xfmt)

cbar = fig.colorbar(cs, orientation="vertical",
extend="both")
cbar.ax.set_ylabel("Temperature ($^\circ$C)")
 ax.set_ylabel("Depth (m)")
```