Ocean InfoHub Release Graph

Ocean Info Hub Community

2023-07-27 19:50:42

Contents


```
%load_ext pretty_jupyter
```

```
# import packages
import pandas as pd
from SPARQLWrapper import SPARQLWrapper, JSON
import json

sparqlep = "http://graph.oceaninfohub.org/blazegraph/namespace/oih/sparql"
```

```
def get_sparql_dataframe(service, query):
    """
    Helper function to convert SPARQL results into a Pandas data frame.
    """
    sparql = SPARQLWrapper(service)
    sparql.setQuery(query)
    sparql.setReturnFormat(JSON)
    result = sparql.query()

    processed_results = json.load(result.response)
    cols = processed_results['head']['vars']

    out = []
    for row in processed_results['results']['bindings']:
```

```
        item = []
        for c in cols:
            item.append(row.get(c, {}).get('value'))
        out.append(item)

    return pd.DataFrame(out, columns=cols)
```

1 About

This is the introduction to the Ocean InfoHub Release Graph.

Besides this HTML file we would want to package

- PDF version of this
- the graphs
- the original Jupyter Notebook that builds the HTML and PDF
- any JSON-LD frames or SHACL files used in generating this document

2 First Section

This is our first section. We use so called **Jinja Markdown** here. It allows us to combine Markdown with Python variables and makes for a more dynamic report.

We can for example print pandas version such as this: 1.5.3.

```
# we create a simple dataframe for demonstration purposes
data = pd.DataFrame({"col1": [1, 2, 3, 4], "col2": ["cat1", "cat2", "cat1",␣
 ↪"cat2"]})

data.head()
```

```
   col1  col2
0     1  cat1
1     2  cat2
2     3  cat1
3     4  cat2
```

3 Tabset Root

The content of this section will be shown as tabs. This will help us avoid potential scrolling and improve the HTML UI.

3.1 First Tab

In the first tab, we can show some graphs or tables. We can output the table like this:

col1

col2

0

1

cat1

1

2

cat2

2

3

cat1

3

4

cat2

3.2 Second Tab

In the second tab, we can do the same. Btw maths also works in the tabs.

3.3

4 Not a Tabset

This section will not be tabbed because it has the same level (or higher) as the Tabset Root.

4.1 Providers

name (graph alias)	catalog	logo
IOC Africa Data Portal (africaioc)	catlog	
AquaDocs (aquadocs)	catlog	
Better Biomolecular Ocean Practices (BeBOP) as part of Ocean Biomolecular Observing Network (OBON) (bebop)	catlog	
Benguela Current Convention (BCC) GeoData Portal (benguelacc)	catlog	
Caribbean Marine Atlas catalogue (caribbeanmarineatlas)	catlog	
CIOOS (cioos)	catlog	
European Directory of Marine Environmental Research Projects (EDMERP) SeaDataNet (edmerp)	catlog	
European Directory of Marine Organisations (EDMO) SeaDataNet (edmo)	catlog	
EurOcean Organizations (euroceanorgs)	catlog	
EurOcean Projects (euroceanprojects)	catlog	
EurOcean Vessels (euroceanvessels)	catlog	
European Marine Observation and Data Network catalogue (emodnet)	catlog	
Indonesia National Oceanic Data Center (inanodc)	catlog	
CHM LAC - Documents (invemardocuments)	catlog	

name (graph alias)	catalog logo
CHM LAC - Experts (invemarexperts)	catlog
CHM LAC - Institutions (invemarinstitutions)	catlog
CHM LAC - Training (invemartraining)	catlog
CHM LAC - Vessels (invemarvessels)	catlog
Marine Training EU (marinetraining)	catlog
MASPAWIO - Marine Spatial Atlas for the Western Indian Ocean (maspawio)	catlog
Ocean Biodiversity Information System (obis)	catlog

```
rq_pcount = """SELECT ?p (COUNT(?p) as ?pCount)
WHERE
{
  ?s ?p ?o .
}
GROUP BY ?p
"""
dfc = get_sparql_dataframe(sparqlep, rq_pcount)
dfc['pCount'] = dfc["pCount"].astype(int) # convert count to int
# dfc.set_index('p', inplace=True)
dfc_sorted = dfc.sort_values('pCount', ascending=False)
```

4.2 countByLicense.rq

p

pCount

154

http://www.w3.org/1999/02/22-rdf-syntax-ns#type

7914266

75

http://www.w3.org/ns/prov#value

2554814

74

http://www.w3.org/ns/prov#used

1277407

73

http://www.w3.org/ns/prov#hadMember

1277407

72

http://www.w3.org/ns/prov#generated

1277407