Astroquery, PyVO, et Ipyaladin

Quoi de neuf pour les pythonistas?

Manon Marchand, Anais Oberto, Gilles Landais, Matthieu Baumann, Thomas Boch, and the CDS team

Centre de Données astronomiques de Strasbourg

(Pour la démo)

- python >= 3.8
- jupyterlab >= 4
- anywidget@latest

} availaible via pip

```
python -m venv ./.venv
source .venv/bin/activate
pip install jupyterlab -U
pip install anywidget -U
```

I. New functionnalities in Astroquery simbad

w. A. Oberto

I.1. Navigate Simbad's TAP_SCHEMA

```
from astroquery.simbad import Simbad
Simbad.list_columns(keyword='proper motion')
```

```
<Table length=13>
table_name   column_name   datatype               description                 unit
  object        object      object                  object                   object
---------- --------------- -------- ---------------------------------------- --------
     basic      pm_bibcode     CHAR                 Proper motion reference
     basic    pm_err_angle SMALLINT               Proper motion error angle      deg
     basic      pm_err_maj     REAL          Proper motion error major axis mas.yr-1
     basic pm_err_maj_prec SMALLINT Proper motion error major axis precision
     basic      pm_err_min     REAL          Proper motion error minor axis mas.yr-1
     basic pm_err_min_prec SMALLINT Proper motion error minor axis precision
     basic         pm_qual     CHAR                   Proper motion quality
     basic           pmdec   DOUBLE                    Proper motion in DEC mas.yr-1
     basic      pmdec_prec SMALLINT          Proper motion in DEC precision
     basic            pmra   DOUBLE                    Proper motion in RA mas.yr-1
     basic       pmra_prec SMALLINT           Proper motion in RA precision
     mesPM            pmde     REAL                     Proper motion DEC. mas.yr-1
     mesPM            pmra     REAL                     Proper motion R.A. mas.yr-1
```

```
Simbad.list_linked_tables('mesPM')
```

```
<Table length=1>
from_table from_column target_table target_column
  object      object      object        object
---------- ----------- ------------ -------------
     mesPM      oidref        basic           oid
```

keys + key_columns

I.2. Directly do a TAP query

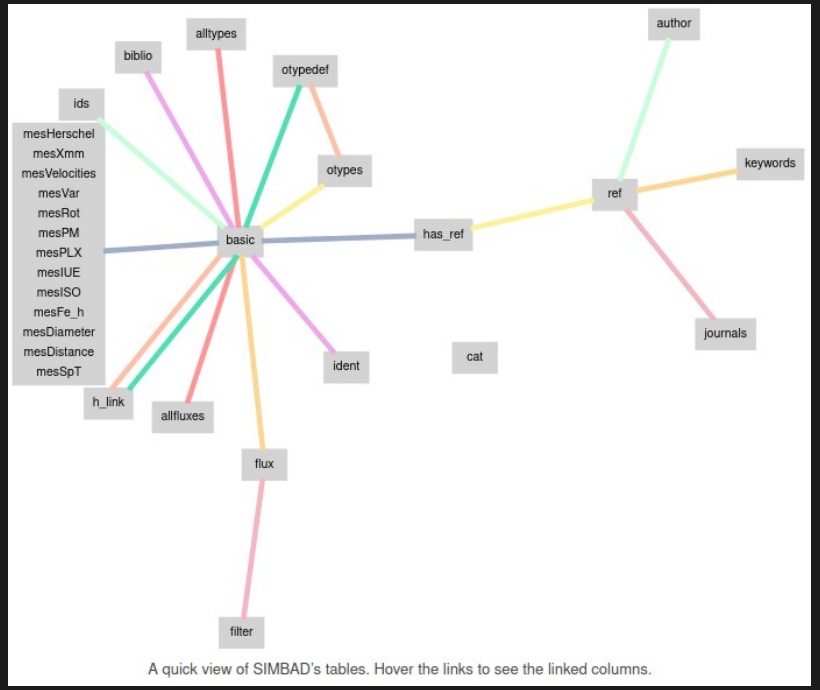
```
Simbad.query_tap('SELECT top 5 main_id FROM basic')
```

```
<Table length=5>
          main_id
           object
---------------------------
    2MASS J11025707-5846116
    2MASS J11030740-5845221
Cl* NGC 3532   FERN     531
    2MASS J11031714-5842502
               CPD-58  2970
```

I.3. Extensive(!) documentation

https://astroquery.readthedocs.io/en/latest/simbad/simbad.html#query-tap

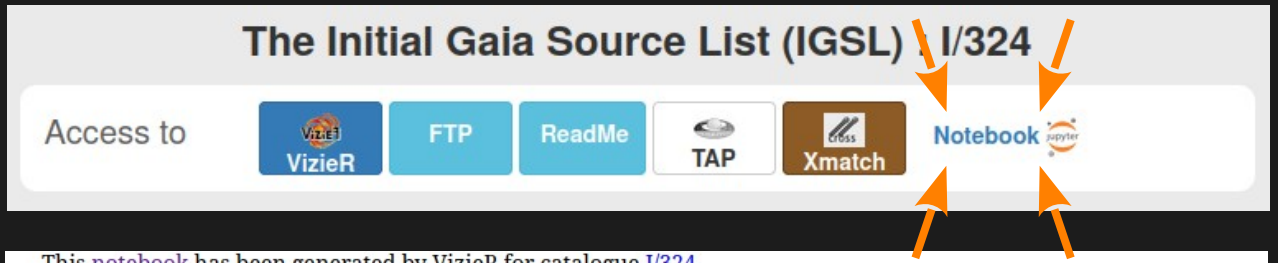
→ still need to make TAP_UPLOAD clearer



Bonus tips:
Graphviz python wrapper is shipped with Sphinx

II. A new button on VizieR pages

w. G. Landais



This notebook has been generated by VizieR for catalogue I/324

↓ Download notebook

Notebook quick view (not executable)

In [ ]:
```
# Access astronomical databases
from pyvo import registry  # version >=1.4.1

# Moc and HEALPix tools
from mocpy import MOC

# Sky visualization
from ipyaladin import Aladin

# For plots
import matplotlib.pyplot as plt
```

Welcome to VizieR example workflow

Vizier

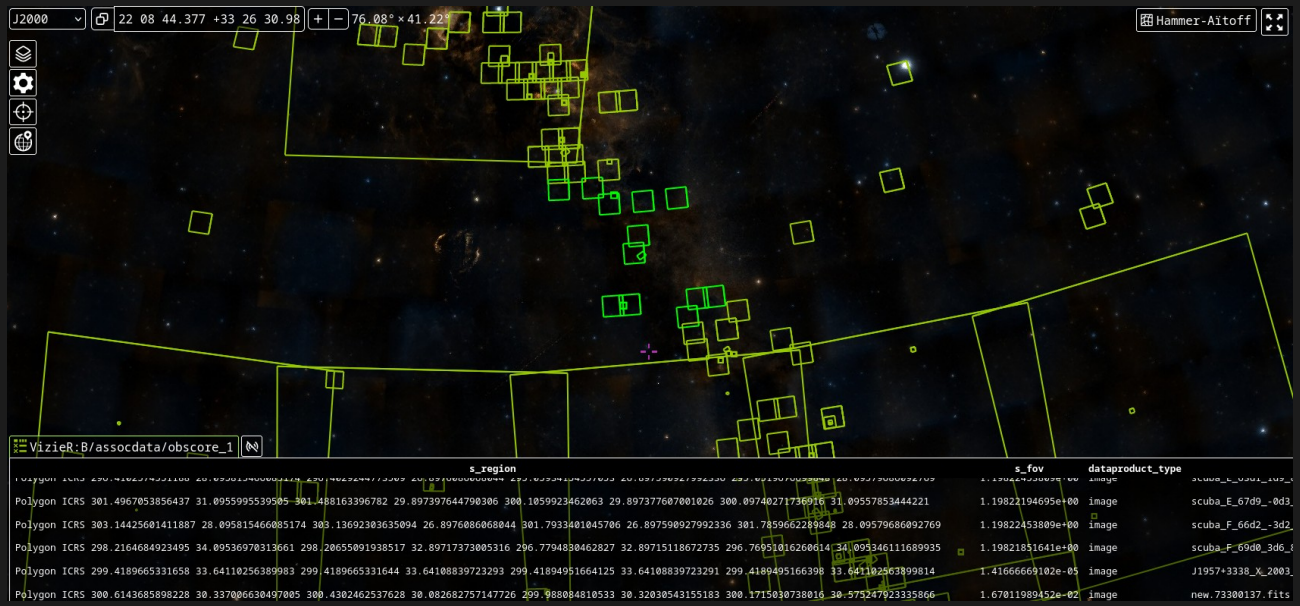
It is a generic notebook, highlighting what can be done once you chose a catalog. This workflow is suggested by CDS (Stra

II.1. VO-centered

- PyVO: Registry metadata, TAP et Obscore, SCS ⟶ 4 PRs into PyVO
- MOCpy + ipyaladin: Coverage of the catalog

II.2. TODO list

- add the new TAP Obscore view of Aladin-lite

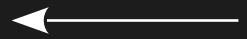


- need a standard way to link an interface to a table ⟶ discussion to come in the registry mailing list

II.3. How it's done

a notebook ≡ json

[notebooks json schema](#)



```
"cell_id": {
      "description": "A string field representing the
identifier of this particular cell.",
      "type": "string",
      "pattern": "^[a-zA-Z0-9-_]+$",
      "minLength": 1,
      "maxLength": 64
    },

"cell": {
      "type": "object",
      "oneOf": [
        { "$ref": "#/definitions/raw_cell" },
        { "$ref": "#/definitions/markdown_cell" },
        { "$ref": "#/definitions/code_cell" }
      ]
    },
```

Edits done:

- catalog name
- sections to include
- scs default parameters

Rendering:

- static view (could have been JupyterLite)

https://github.com/jupyter/nbconvert

```
jupyter nbconvert --to html notebook.ipynb
```

III. Ipyaladin

Initially working on bare ipywidgets ⟶ dropped support for JS widgets 

https://anywidget.dev/

ipyaladin is (almost) totally rewritten since version 0.3.0

⟶ demo time‼