

File grabbing interface

EPN-TAP data services may be used to perform automatic processing of datasets, in which case rapid identification of file locations would help. Although TAP queries are helpful to select files matching complicated conditions, they may be too slow for this kind of application.

This new interface is a set of web services intended to retrieve the URL of a file distributed in any EPN-TAP service easily, e. g., to avoid handling a complex directory structure. This is typically for use with a pipeline procedure or for mirroring/backup purpose. Teams wishing to use these services to handle their own data are encouraged to provide the file_name parameter in their EPN-TAP data service.

Two such services are available, which search either the access_url parameter (always present, but may link to a script rather than a file) or the file_name parameter (more accurate but not always provided in the service).

Basic use from a browser

1- Search on file_name

In a browser, look for

http://voparis-europlanet-dev.obspm.fr/planetary/data/e pn/access-url/?file_name=VV0521_05.CAL

The search is case insensitive, but exact name with extension is required

2- Search on access_url

In a browser, look for

http://voparis-europlanet-dev.obspm.fr/planetary/data/e pn/access-url/?access_url_contains=VV0521_05

Case insensitive, no extension required, also finds partial strings

3- Quick search

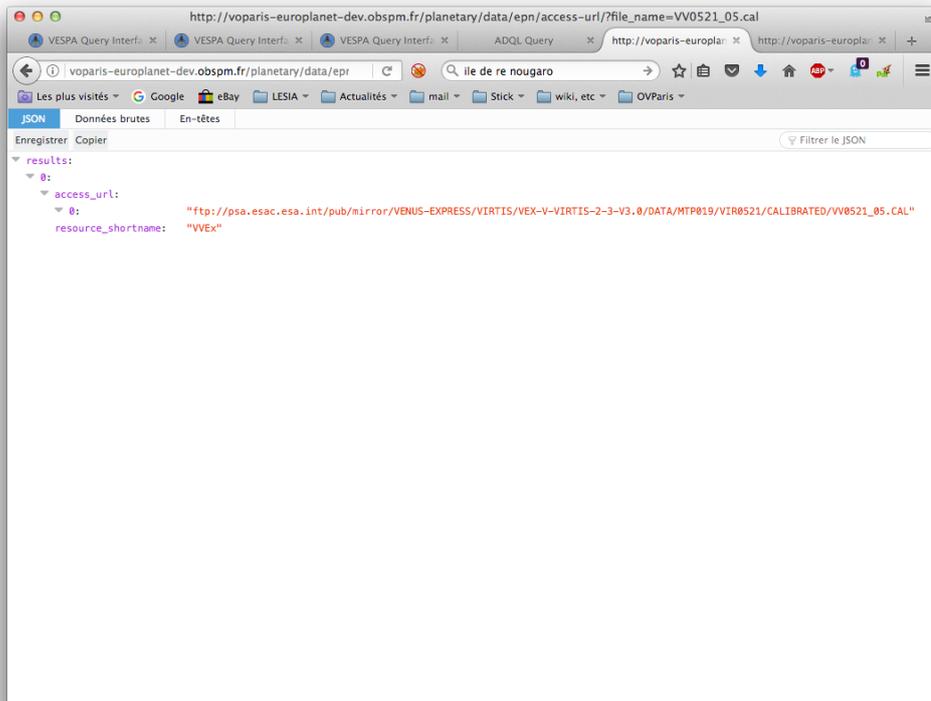
The previous searches may be restrained to a given EPN-TAP service, so as to accelerate the process.

The service short name is the schema name provided in the service_title parameter and in the registry. The request is case insensitive.

http://voparis-europlanet-dev.obspm.fr/planetary/data/e pn/access-url/?access_url_contains=VV0521_05.CAL&service_shortname=VVEx

http://voparis-europlanet-dev.obspm.fr/planetary/data/e pn/access-url/?file_name=iksfig7.xml&service_shortname=IKS

This last command is intended to be the standard access mode for pipeline processing. For this to work correctly, the exact name must be used and the file_name parameter must be provided by the service.



In all cases the browser will display a json string that contains a field called `access_url` for each match found:

```

{"results": [{"access_url": ["http://voparis-srv.obspm.fr/vo/planeto/apis/dataset/Bastet/Saturn_-_2017_14_Feb-09_Sept/od9u24m8q_x2d.fits"], "resource_shortcode": "APIS"}]}

```

It can be copied from there manually.

The field `resource_shortcode` refers to the EPN-TAP service where the match was found.

Standard use from terminal/script

1- From the terminal (under bash)

```
content=$(wget http://voparis-europlanet-dev.obspm.fr/planetary/data/epr/access-url/?file_name=VV0521_05.CAL -q -O -)
```

```
echo $content
```

=> will print a json string with a field `access_url`:

```

{"results": [{"access_url": ["ftp://psa.esac.esa.int/pub/mirror/VENUS-EXPRESS/VIRTIS/VEX-V-VIRTIS-2-3-V3.0/DATA/MTP019/VIR0521/GEOMETRY/VV0521_05.CAL"], "resource_shortcode": "VVEx"}]}

```

2- Alternatively (under bash or tcsh)

```
curl -s 'http://voparis-europlanet-dev.obspm.fr/planetary/data/epr/access-url/?access_url_contains=od9u24m8q_x2d.fits'
```

=> will print

```

{"results": [{"access_url": ["http://voparis-srv.obspm.fr/vo/planeto/apis/dataset/Bastet/Saturn_-_2017_14_Feb-09_Sept/od9u24m8q_x2d.fits"], "resource_shortcode": "APIS"}]}

```

3- Grab the url with a python routine

The URL itself can be retrieved from the terminal:

```
curl -s 'http://voparis-europlanet-dev.obspm.fr/planetary/data/epr/access-url/?access_url_contains=od9u24m8q_x2d.fits' | python readjsonaccessurl.py
```

Where `readjsonaccessurl.py` contains:

```

import sys, json;
for resource_result in json.load(sys.stdin)['results']:
    for url in resource_result['access_url']:
        print url

```

to be installed, e.g., in the current directory.

4- Use under IDL

The following example will print the json string and the url (requires IDL 8):

```
; querying the service
oUrl = OBJ_NEW('IDLnetUrl')
pipo=oUrl->Get(URL="http://voparis-europlanet-dev.obspm.fr/planetary/data/epn/access-url/?
access_url_contains=VV0521_05.CAL&service_shortcode=VVEx", /string)
print, pipo
OBJ_DESTROY, oUrl

; parsing the URL (array with one element per match)
truc= JSON_PARSE(pipo, /tostruct, /toarray)
print, truc.results.access_url(0)
; getting the file
nomf= wget(truc.results.access_url(0), dir='/tmp')
; then read it with appropriate routine, eg for VOTable: spe = read_vot(nomf)
```

Notice that IDL's older wget routine cannot query a web service.