PyVO is an astropy-affiliated package providing an API for the access and retrieval of astronomical datasets from the Virtual Observatory (VO) using various VO Data Access Layer Protocols. We have recently added support for the Table Access Protocol (TAP) in pyVO. With this, pyVO now supports synchronous and asynchronous queries, including the upload of local tables. PyVO’s TAP support also allows inspection of the service metadata. Thanks to astropy integration, it is straightforward to work with the results obtained and re-use them either in further VO queries or in custom python code.

Basic TAP API

```
Module Import, Service Creation:

from pyvo.dal import tap

service = tap.TAPService(
    "http://dc.g-vo.org/tap"
)

Synchronous Query:

result = service.run_sync(
    "SELECT ROUND(rv/5) AS bin, al, sl, fe, fe_n
    FROM rave.main
    WHERE rv BETWEEN 40 AND 70 AND al IS NOT NULL"
).votable.to_table()
```

Async TAP API

```
For long-running jobs, TAP lets clients execute jobs asyn-
chronously, using the UWS job pattern.

Simple async querying:

result = service.run_async(
    "SELECT * FROM ppmxl.main"
).votable.to_table()
```

```
Adanced Async Operation:

job = service.submit_job("SELECT * FROM gaia.dr1")
job.wait()
job.raise_if_error()
result = job.fetch().votable.to_table()
```

```
Resuming Async Jobs:

print("Come back in two days and resume %s"%job.url)
```

```
The figure to the left was produced with the following pyVO program:

import matplotlib.pyplot as plt
import numpy as np
import pyvo

BIN_SIZE = 10

def clip(arr, limit):
    arr[arr<limit] = limit
    arr[arr>limit] = limit

svc = pyvo.dal.TAPService("http://dc.g-vo.org/tap")
res = svc.run_sync(
    "SELECT round(raj2000/50) AS xind, round(((dec2000+90)/100)) AS yind, count(*) as ct
    FROM tgas.main
    GROUP by xind, yind"".format(BIN_SIZE, BIN_SIZE),
    maxrec=1000000).votable.to_table()

clip(res["pmde"], 50)
clip(res["pmra"], 50)

w, b = max(res["xind"]), max(res["yind"])

for x, y, v in zip(res["pmra"], res["pmde"], res["pmra", res["pmde"]])
    u = (v[1] + v[0]) / 2
    weight = u**2
    plt.streamplot(np.arange(w+1)*BIN_SIZE, np.arange(h+1)*BIN_SIZE,
                   u, v, color=np.sqrt(u**2+v**2), linewidth=weight)
```

Acknowledgement: This work was supported by Sonderforschungsbereich SFB 881 “The Milky Way System” of the German Research Foundation (DFG).

Abstract

Module Import, Service Creation: Most TAP functionality is provided through a TAPService object constructed with the base access URL of the TAP service:

```
from pyvo.dal import tap

service = tap.TAPService(
    "http://dc.g-vo.org/tap"
)
```

Synchronous Query: A basic synchronous query is run by passing the query string to a service’s run_sync method. The query result is available in the return value’s votable.to_table() attribute (this is for compatibility with the rest of pyVO).

```
result = service.run_sync(
    "SELECT ROUND(rv/5) AS bin, al, sl, fe, fe_n
    FROM rave.main
    WHERE rv BETWEEN 40 AND 70 AND al IS NOT NULL"
).votable.to_table()
```

Job Customisation: Most TAP services enforce relatively small match limits when not giving TAP’s MAXREC argument. You can explicitly pass it:

```
result = service.run_sync("SELECT * FROM ppmxl.main",
    maxrec=1000000).votable.to_table()
```

(language works analogously).

Uploads: You can use local tables in your remote queries:

```
from astropy.table import Table

local_table = Table([ras, decs, pmras, pmdecs],
    names=('ra', 'dec', 'pmra', 'pmdec'))

response = service.run_sync(
    "***SELECT ROUND(rv/5) AS bin, al, sl, fe, fe_n
    FROM rave.main
    WHERE rv BETWEEN 40 AND 70 AND al IS NOT NULL***"
).votable.to_table()
```

```
TAP support entered pyVO in version 0.3, which will be available on pyPI soon. You can then obtain it by simply using pip install pyvo.
```

```
Meanwhile, please try our code by cloning https://github.com/pyvirtobs/pyvo.git. We also appreciate bug reports or feature requests on github.
```

Installation

TAP support entered pyVO in version 0.3, which will be available on pyPI soon. You can then obtain it by simply using pip install pyvo. Meanwhile, please try our code by cloning https://github.com/pyvirtobs/pyvo.git. We also appreciate bug reports or feature requests on github.

1 SFB 881 “The Milky Way System”
2 Universität Heidelberg, Astronomisches Rechen-Institut, Mönchhofstraße 12-14, Germany
sbecker@ari.uni-heidelberg.de