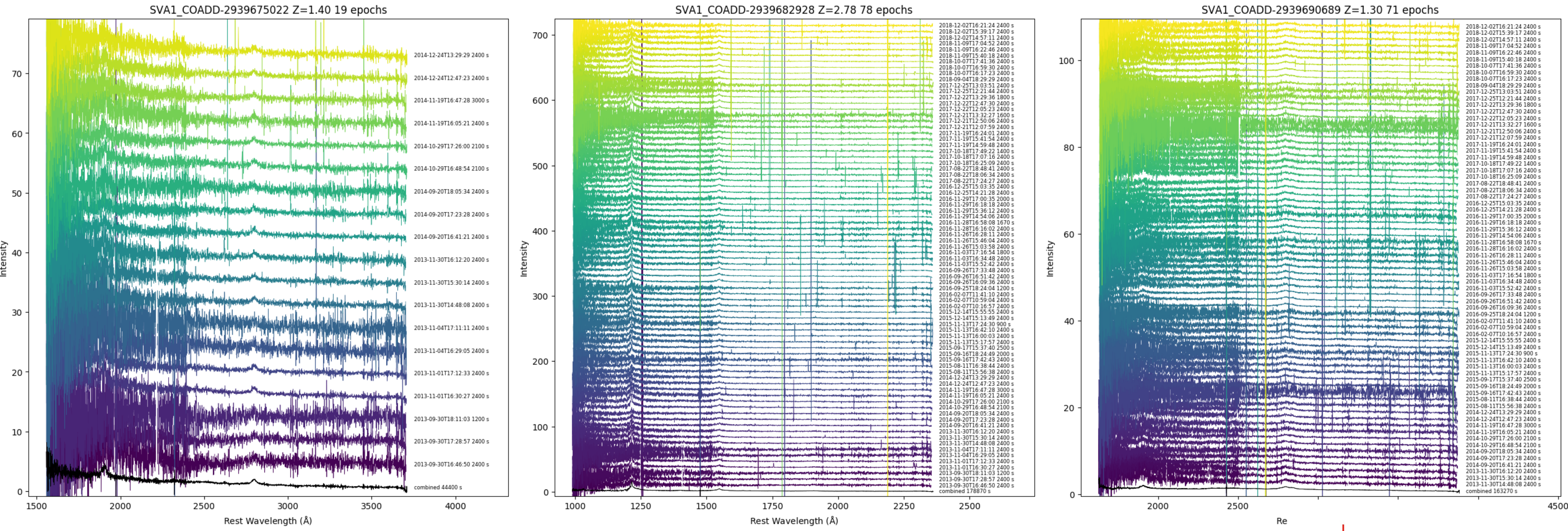
Data Central’s Simple Spectral Access Service



Dr Brent Miszalski
brent.miszalski@mq.edu.au

James Tocknell

Dr Simon O’Toole

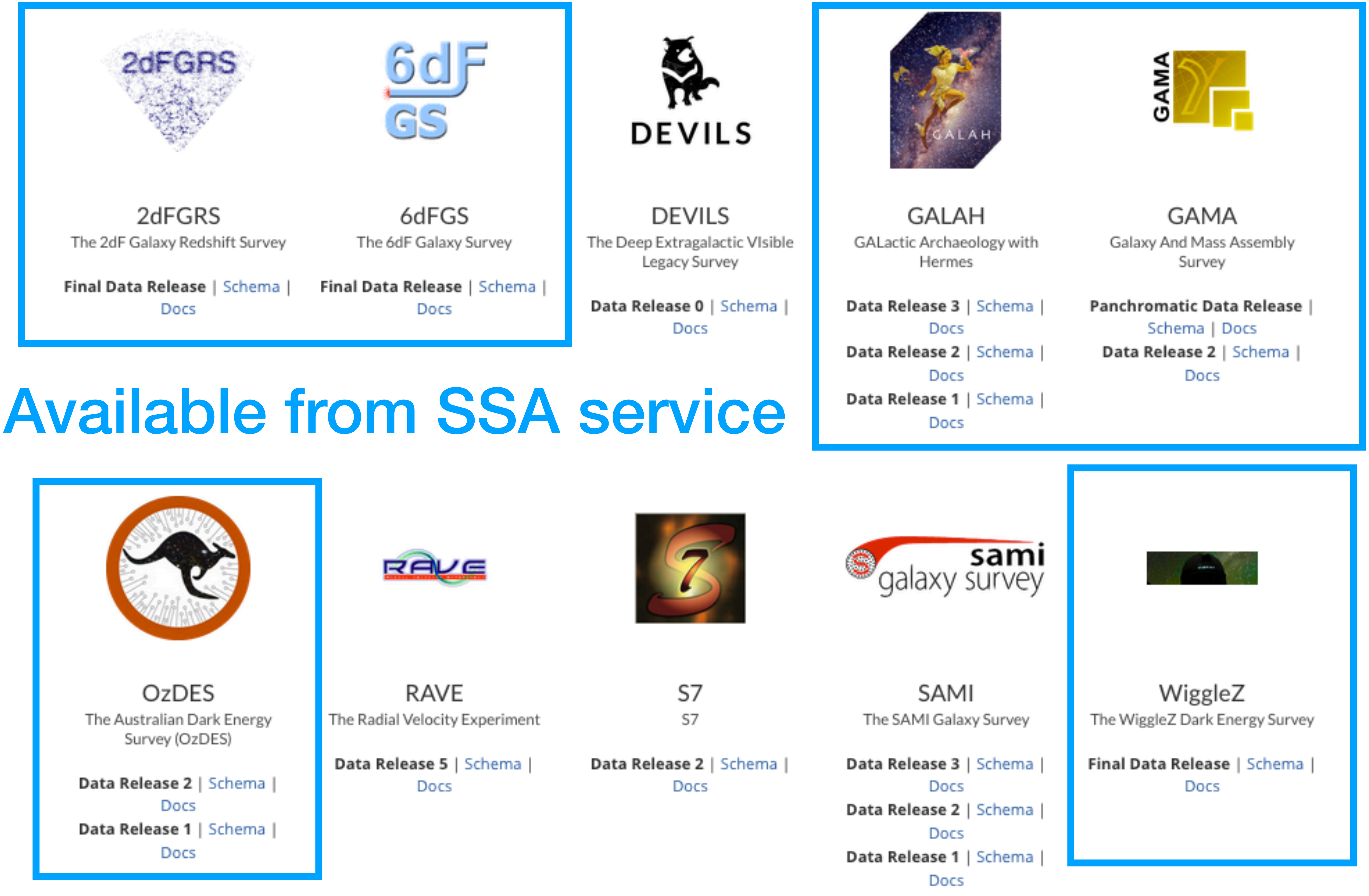
- Data Central (datacentral.org.au) is an e-research platform and data archive developed at Australian Astronomical Optics (AAO), Macquarie University, that facilitates cutting-edge science.

- It provides web-based tools and archive functionality for scientists from a range of disciplines to explore, collaborate and make new discoveries.

- New **SIA** (July 2020) and **SSA** (February 2021) services. Lots of Python examples.

- **Python 3 Django** implementations: Focus on curated metadata and accessibility.

Millions of spectra: AAT and UKST surveys



2dFGRS
The 2dF Galaxy Redshift Survey
Final Data Release | Schema | Docs

6dFGS
The 6dF Galaxy Survey
Final Data Release | Schema | Docs

DEVILS
The Deep Extragalactic VIsible Legacy Survey
Data Release 0 | Schema | Docs

GALAH
GALactic Archaeology with Hermes
Data Release 3 | Schema | Docs
Data Release 2 | Schema | Docs
Data Release 1 | Schema | Docs

GAMA
Galaxy And Mass Assembly Survey
Panchromatic Data Release | Schema | Docs
Data Release 2 | Schema | Docs

Available from SSA service

OzDES
The Australian Dark Energy Survey (OzDES)
Data Release 2 | Schema | Docs
Data Release 1 | Schema | Docs

RAVE
The Radial Velocity Experiment
Data Release 5 | Schema | Docs

S7
S7
Data Release 2 | Schema | Docs

SAMI
The SAMI Galaxy Survey
Data Release 3 | Schema | Docs
Data Release 2 | Schema | Docs
Data Release 1 | Schema | Docs

WiggleZ
The WiggleZ Dark Energy Survey
Final Data Release | Schema | Docs

Links

- Endpoint: https://datacentral.org.au/vo/ssa/query
- Main doc page: https://docs.datacentral.org.au/reference/services/simple-spectral-access-ssa-service/
- Several Python examples: https://docs.datacentral.org.au/help-center/examples/simple-spectral-access-ssa-examples/
- Specutils loaders: https://github.com/astropy/specutils/tree/main/specutils/io/default_loaders

Data handling

- **Data Central ecosystem:** All spectra ingested using Django model for spectra tied to survey catalogues.

- **Obscore Django model** developed by **James Tocknell** and myself: Attached to each spectrum to complement the existing Data Central Django model.

- **Heterogeneous spectra:** Each survey has a different approach…

    - Multiple spectra: could be one per extension or multiple in one extension (multispec). Images in some extensions!

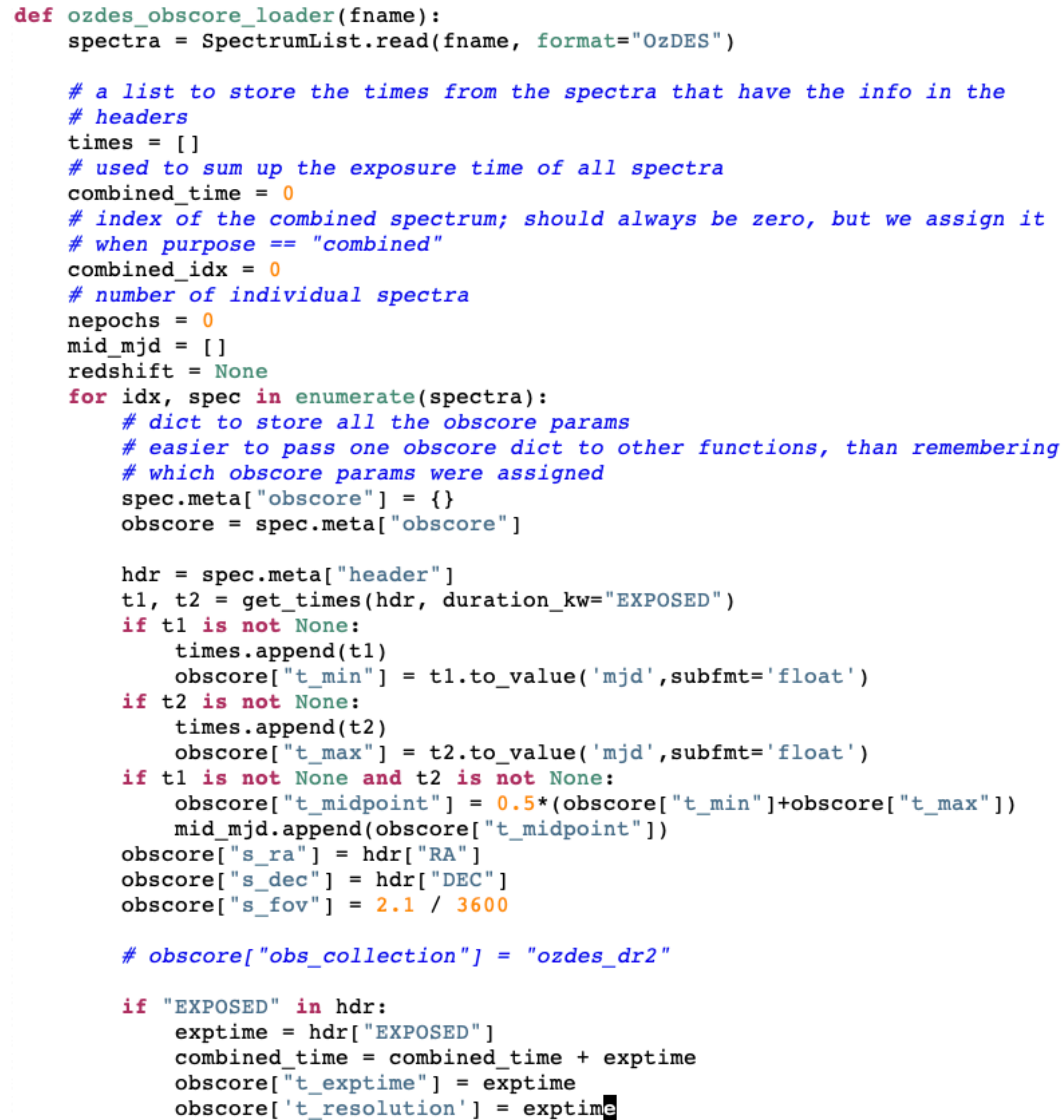
    - Need way to load each spectrum and produce 1D spectra readable by all clients

- **James Tocknell** developed **astropy specutils loaders** for each format => Spectrum1D object for each spectrum.

- Internal code to populate obscore parameters of each Spectrum1D

    - Calculate missing header keywords, total exposure times for multiple epochs or combined spectra, etc.

Example obscore loader: OzDES spectra

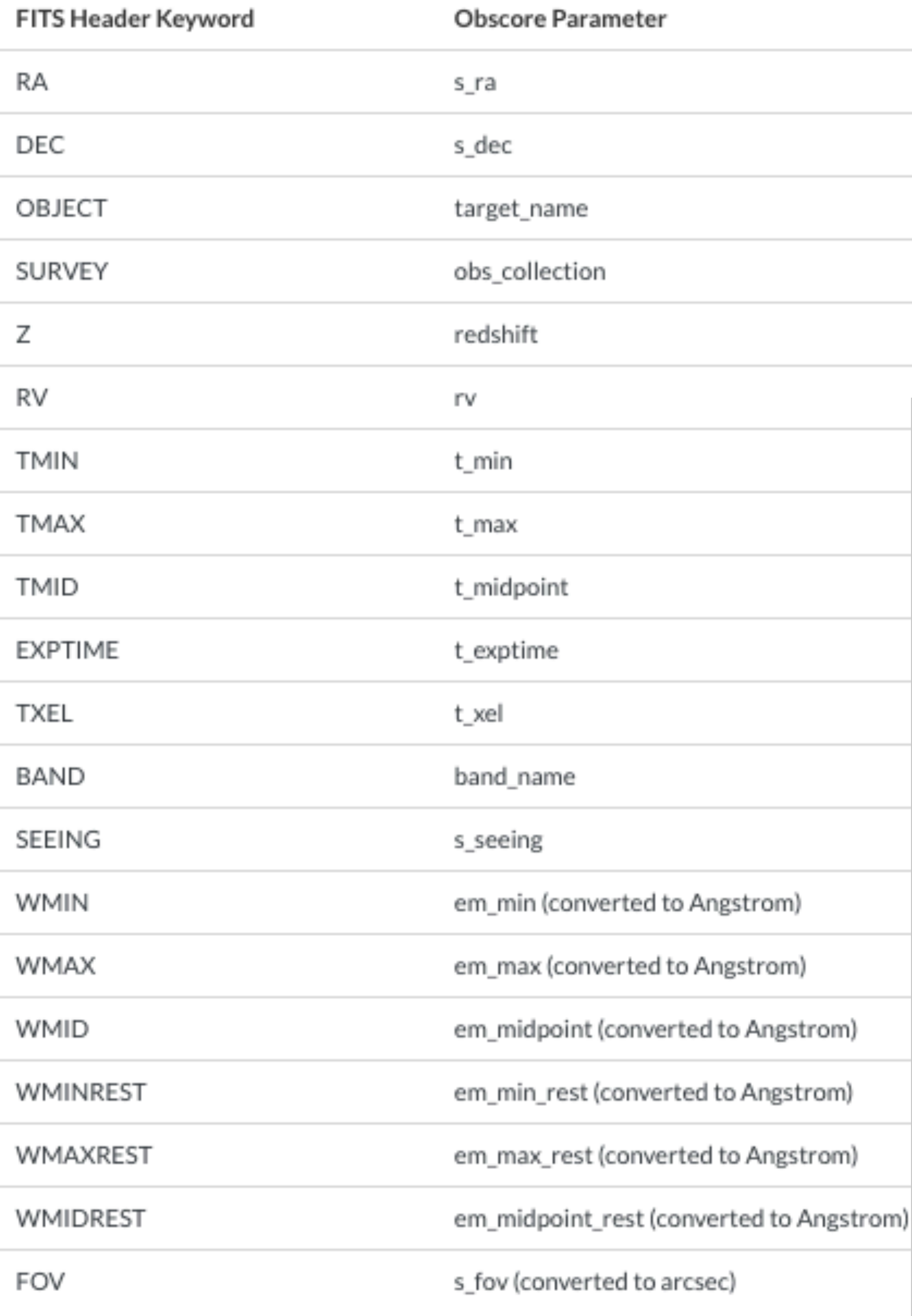


```
def ozdes_obscore_loader(fname):
    spectra = SpectrumList.read(fname, format="OzDES")

    # a list to store the times from the spectra that have the info in the
    # headers
    times = []
    # used to sum up the exposure time of all spectra
    combined_time = 0
    # index of the combined spectrum; should always be zero, but we assign it
    # when purpose == "combined"
    combined_idx = 0
    # number of individual spectra
    nepochs = 0
    mid_mjd = []
    redshift = None
    for idx, spec in enumerate(spectra):
        # dict to store all the obscore params
        # easier to pass one obscore dict to other functions, than remembering
        # which obscore params were assigned
        spec.meta["obscore"] = {}
        obscore = spec.meta["obscore"]

        hdr = spec.meta["header"]
        t1, t2 = get_times(hdr, duration_kw="EXPOSED")
        if t1 is not None:
            times.append(t1)
            obscore["t_min"] = t1.to_value('mjd',subfmt='float')
        if t2 is not None:
            times.append(t2)
            obscore["t_max"] = t2.to_value('mjd',subfmt='float')
        if t1 is not None and t2 is not None:
            obscore["t_midpoint"] = 0.5*(obscore["t_min"]+obscore["t_max"])
            mid_mjd.append(obscore["t_midpoint"])
        obscore["s_ra"] = hdr["RA"]
        obscore["s_dec"] = hdr["DEC"]
        obscore["s_fov"] = 2.1 / 3600

        # obscore["obs_collection"] = "ozdes_dr2"

        if "EXPOSED" in hdr:
            exptime = hdr["EXPOSED"]
            combined_time = combined_time + exptime
            obscore["t_exptime"] = exptime
            obscore['t_resolution'] = exptime
```

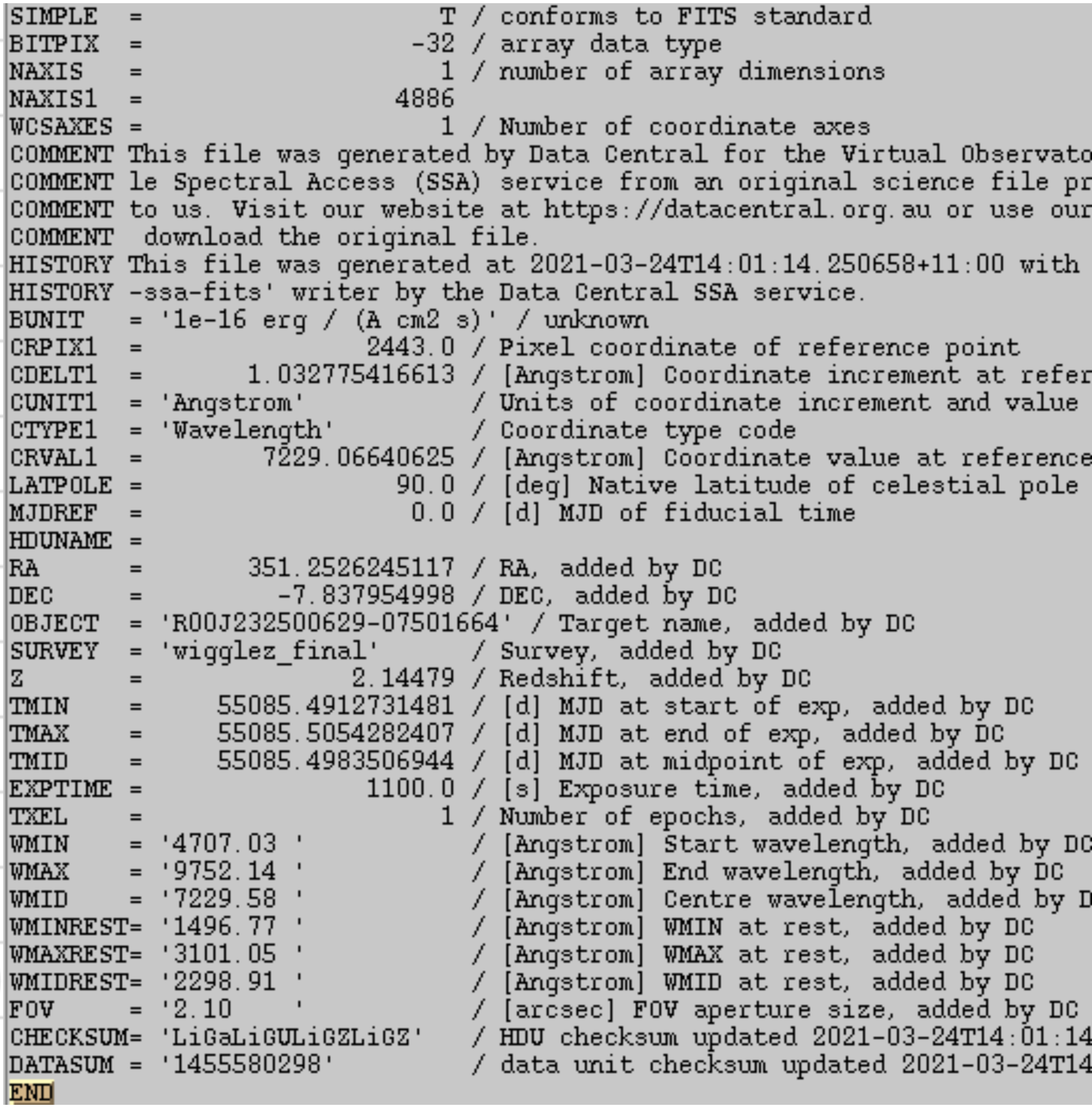
Output data formats

- Datalink service **slink** that extracts spectra of interest
    - VOTable output by default, to allow for **TOPCAT/SPLAT** preview of spectra
    - Can add **&RESPONSEFORMAT=fits** to urls to return FITS format
- **Simplified 1D spectra:** Available from **access_url**
    - Accessible spectra, readable from majority of clients
    - A few essential header keywords added by Data Central
- **Original spectra:** Available from **full_data_url**
    - Survey team provided file that contains spectrum of interest
    - Often complex format, may require loaders to open **(specutils github repo)**
    - Full original header information + other spectra (sky background, variance, etc.)

FITS header keywords added by Data Central



FITS Header Keyword	Obscore Parameter
RA	s_ra
DEC	s_dec
OBJECT	target_name
SURVEY	obs_collection
Z	redshift
RV	rv
TMIN	t_min
TMAX	t_max
TMID	t_midpoint
EXPTIME	t_exptime
TXEL	t_xel
BAND	band_name
SEEING	s_seeing
WMIN	em_min (converted to Angstrom)
WMAX	em_max (converted to Angstrom)
WMID	em_midpoint (converted to Angstrom)
WMINREST	em_min_rest (converted to Angstrom)
WMAXREST	em_max_rest (converted to Angstrom)
WMIDREST	em_midpoint_rest (converted to Angstrom)
FOV	s_fov (converted to arcsec)

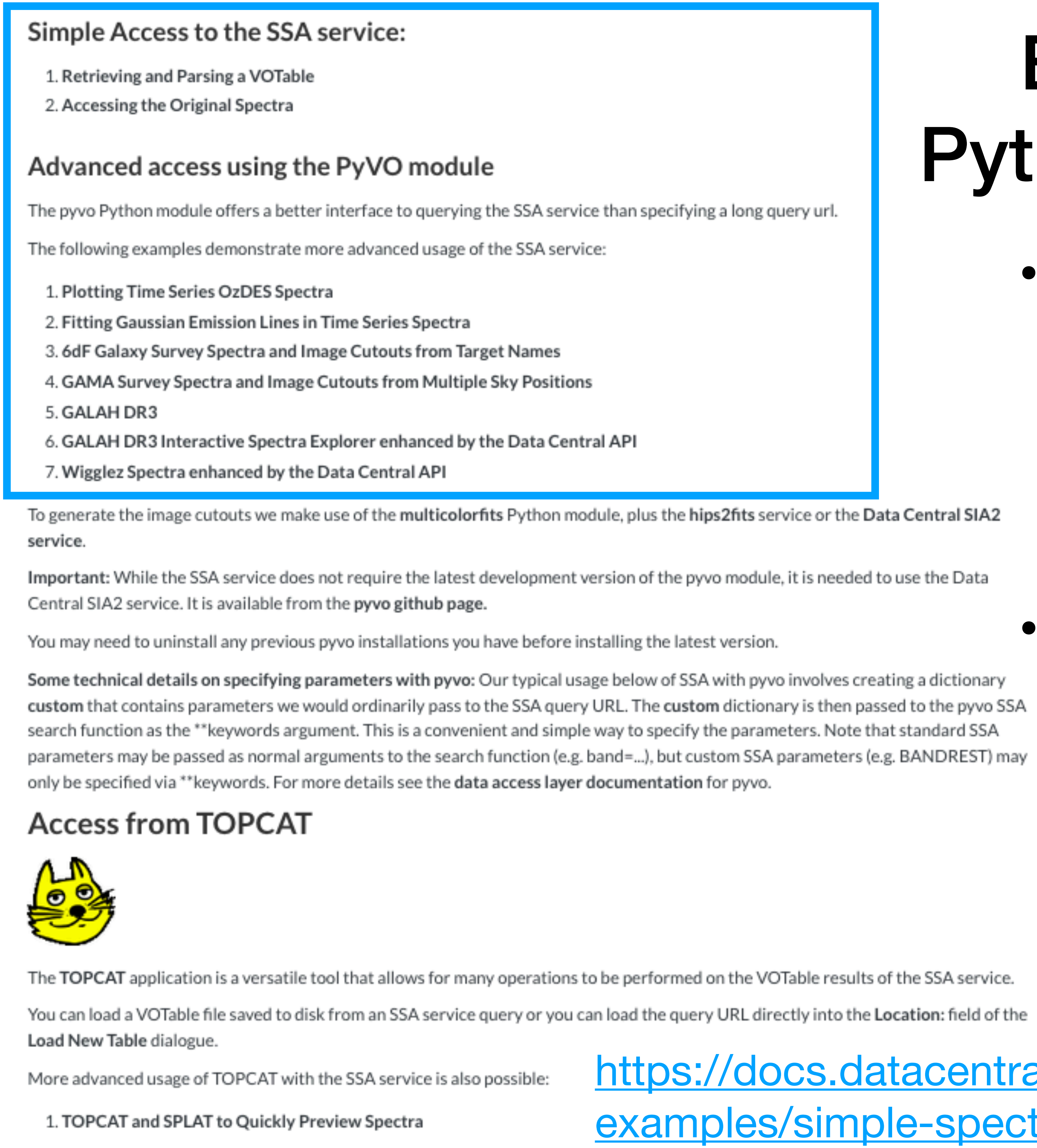


```
SIMPLE  =                    T / conforms to FITS standard
BITPIX  =                  -32 / array data type
NAXIS   =                    1 / number of array dimensions
NAXIS1  =                 4886
WCSAXES =                    1 / Number of coordinate axes
COMMENT This file was generated by Data Central for the Virtual Observato
COMMENT le Spectral Access (SSA) service from an original science file pr
COMMENT to us. Visit our website at https://datacentral.org.au or use our
COMMENT  download the original file.
HISTORY This file was generated at 2021-03-24T14:01:14.250658+11:00 with
HISTORY -ssa-fits' writer by the Data Central SSA service.
BUNIT   = '1e-16 erg / (A cm2 s)' / unknown
CRPIX1  =               2443.0 / Pixel coordinate of reference point
CDELT1  =       1.032775416613 / [Angstrom] Coordinate increment at refer
CUNIT1  = 'Angstrom'           / Units of coordinate increment and value
CTYPE1  = 'Wavelength'         / Coordinate type code
CRVAL1  =        7229.06640625 / [Angstrom] Coordinate value at reference
LATPOLE =                 90.0 / [deg] Native latitude of celestial pole
MJDREF  =                  0.0 / [d] MJD of fiducial time
HDUNAME =
RA      =       351.2526245117 / RA, added by DC
DEC     =         -7.837954998 / DEC, added by DC
OBJECT  = 'R00J232500629-07501664' / Target name, added by DC
SURVEY  = 'wigglez_final'      / Survey, added by DC
Z       =              2.14479 / Redshift, added by DC
TMIN    =     55085.4912731481 / [d] MJD at start of exp, added by DC
TMAX    =     55085.5054282407 / [d] MJD at end of exp, added by DC
TMID    =     55085.4983506944 / [d] MJD at midpoint of exp, added by DC
EXPTIME =               1100.0 / [s] Exposure time, added by DC
TXEL    =                    1 / Number of epochs, added by DC
WMIN    = '4707.03 '           / [Angstrom] Start wavelength, added by DC
WMAX    = '9752.14 '           / [Angstrom] End wavelength, added by DC
WMID    = '7229.58 '           / [Angstrom] Centre wavelength, added by D
WMINREST= '1496.77 '           / [Angstrom] WMIN at rest, added by DC
WMAXREST= '3101.05 '           / [Angstrom] WMAX at rest, added by DC
WMIDREST= '2298.91 '           / [Angstrom] WMID at rest, added by DC
FOV     = '2.10    '           / [arcsec] FOV aperture size, added by DC
CHECKSUM= 'LiGaLiGULiGZLiGZ'   / HDU checksum updated 2021-03-24T14:01:14
DATASUM = '1455580298'         / data unit checksum updated 2021-03-24T14
END
```

Example Python scripts

- Extensive use of **astropy**, **pyvo** and **matplotlib**. Convert SSA query results from VOTable to pandas dataframes with **`to_pandas()`**

- Mainly handle spectra internally without saving to disk. Easy to modify examples to write out spectra



Simple Access to the SSA service:

1. Retrieving and Parsing a VOTable
2. Accessing the Original Spectra

Advanced access using the PyVO module

The pyvo Python module offers a better interface to querying the SSA service than specifying a long query url.

The following examples demonstrate more advanced usage of the SSA service:

1. Plotting Time Series OzDES Spectra
2. Fitting Gaussian Emission Lines in Time Series Spectra
3. 6dF Galaxy Survey Spectra and Image Cutouts from Target Names
4. GAMA Survey Spectra and Image Cutouts from Multiple Sky Positions
5. GALAH DR3
6. GALAH DR3 Interactive Spectra Explorer enhanced by the Data Central API
7. Wigglez Spectra enhanced by the Data Central API

To generate the image cutouts we make use of the **multicolorfits** Python module, plus the **hips2fits** service or the **Data Central SIA2 service**.

Important: While the SSA service does not require the latest development version of the pyvo module, it is needed to use the Data Central SIA2 service. It is available from the **pyvo github page.**

You may need to uninstall any previous pyvo installations you have before installing the latest version.

Some technical details on specifying parameters with pyvo: Our typical usage below of SSA with pyvo involves creating a dictionary **custom** that contains parameters we would ordinarily pass to the SSA query URL. The **custom** dictionary is then passed to the pyvo SSA search function as the **keywords argument. This is a convenient and simple way to specify the parameters. Note that standard SSA parameters may be passed as normal arguments to the search function (e.g. band=...), but custom SSA parameters (e.g. BANDREST) may only be specified via **keywords. For more details see the **data access layer documentation** for pyvo.

Access from TOPCAT

The **TOPCAT** application is a versatile tool that allows for many operations to be performed on the VOTable results of the SSA service.

You can load a VOTable file saved to disk from an SSA service query or you can load the query URL directly into the **Location:** field of the **Load New Table** dialogue.

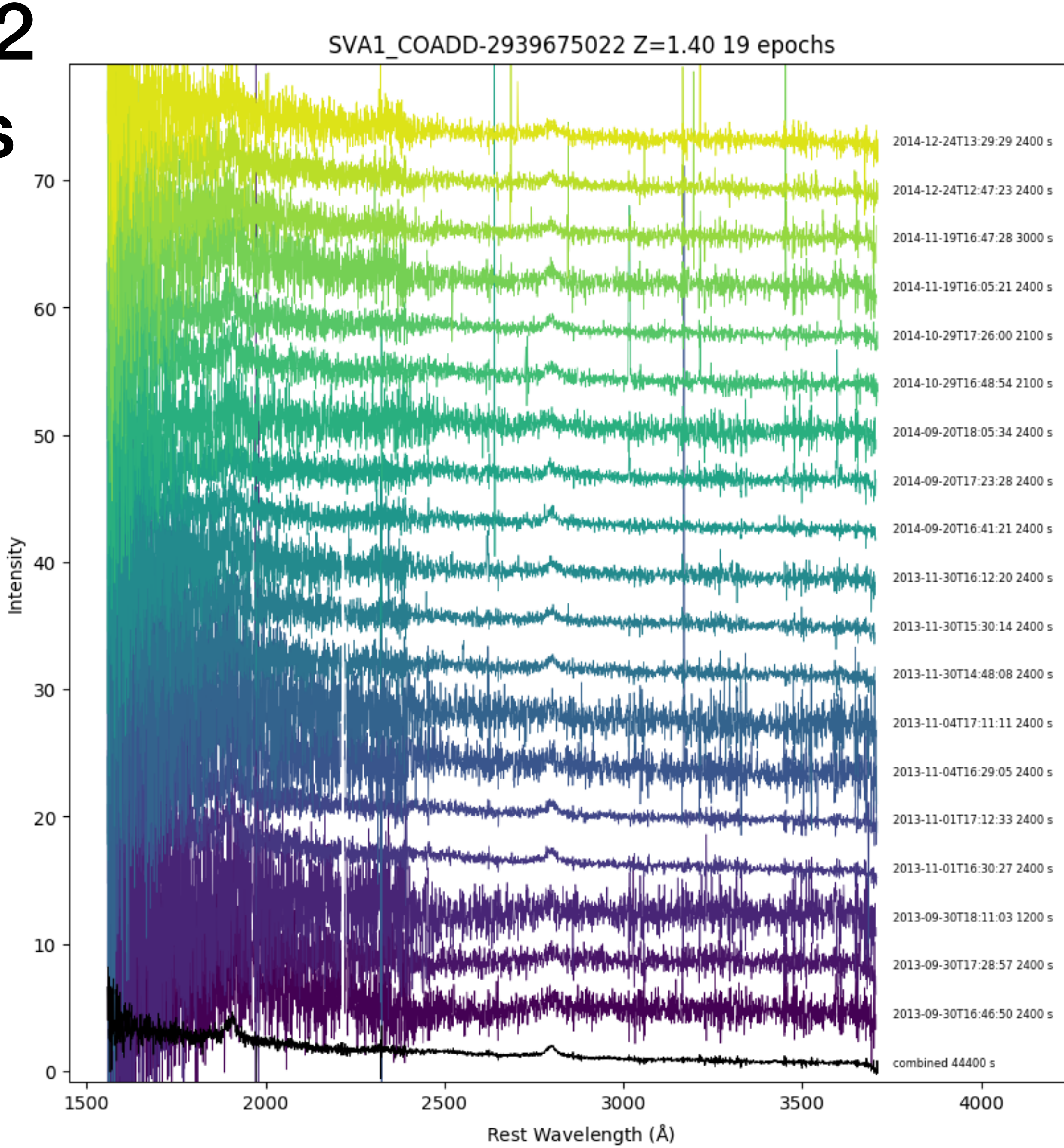
More advanced usage of TOPCAT with the SSA service is also possible:

1. TOPCAT and SPLAT to Quickly Preview Spectra

https://docs.datacentral.org.au/help-center/examples/simple-spectral-access-ssa-examples/

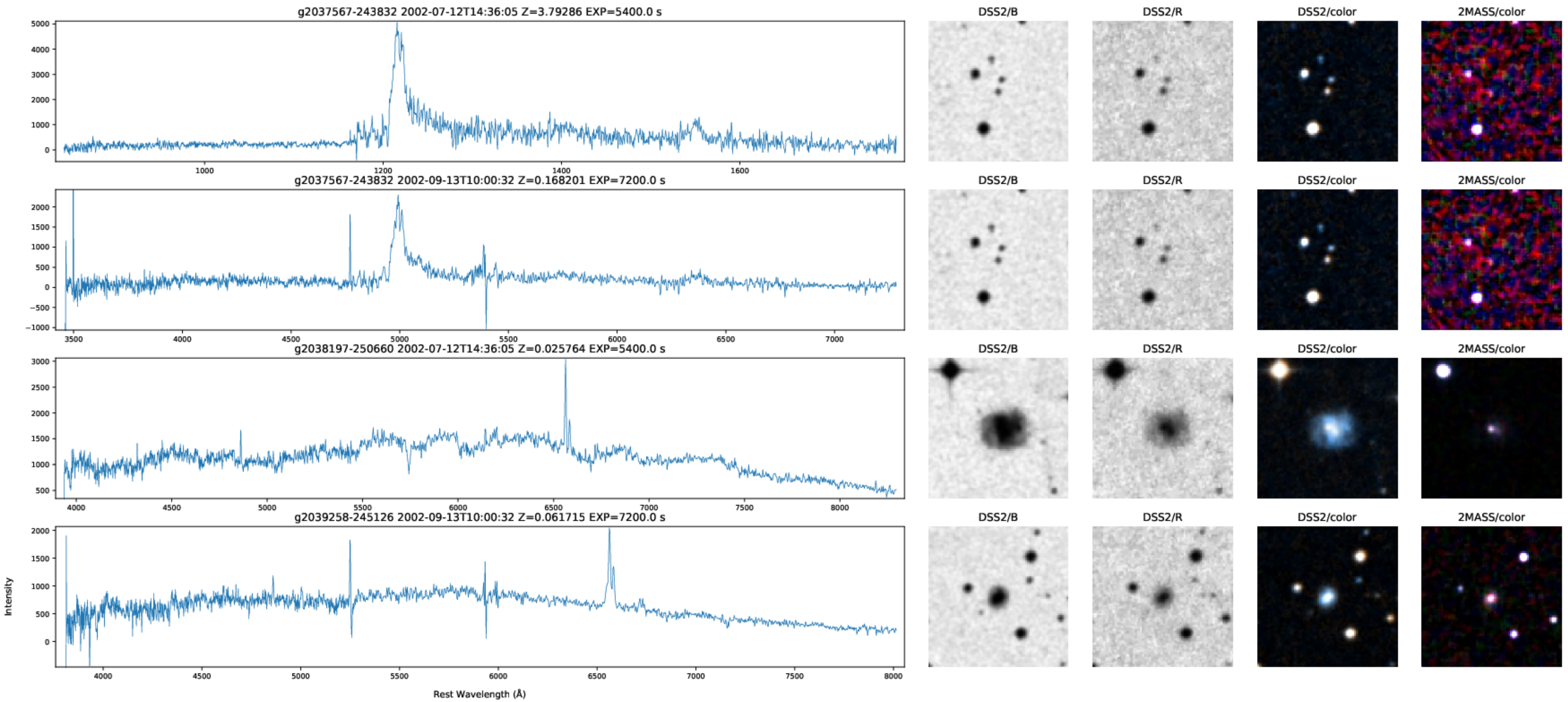
OzDES DR2 time series

- IRAF **specplot** like display of time series spectra from SSA query
- Easy to access dozens of spectra via **pyvo** and create plots of spectra with **matplotlib**
- Can specify individual target with **TARGETNAME** or select only spectra that overlap a specific rest wavelength using **BANDREST**



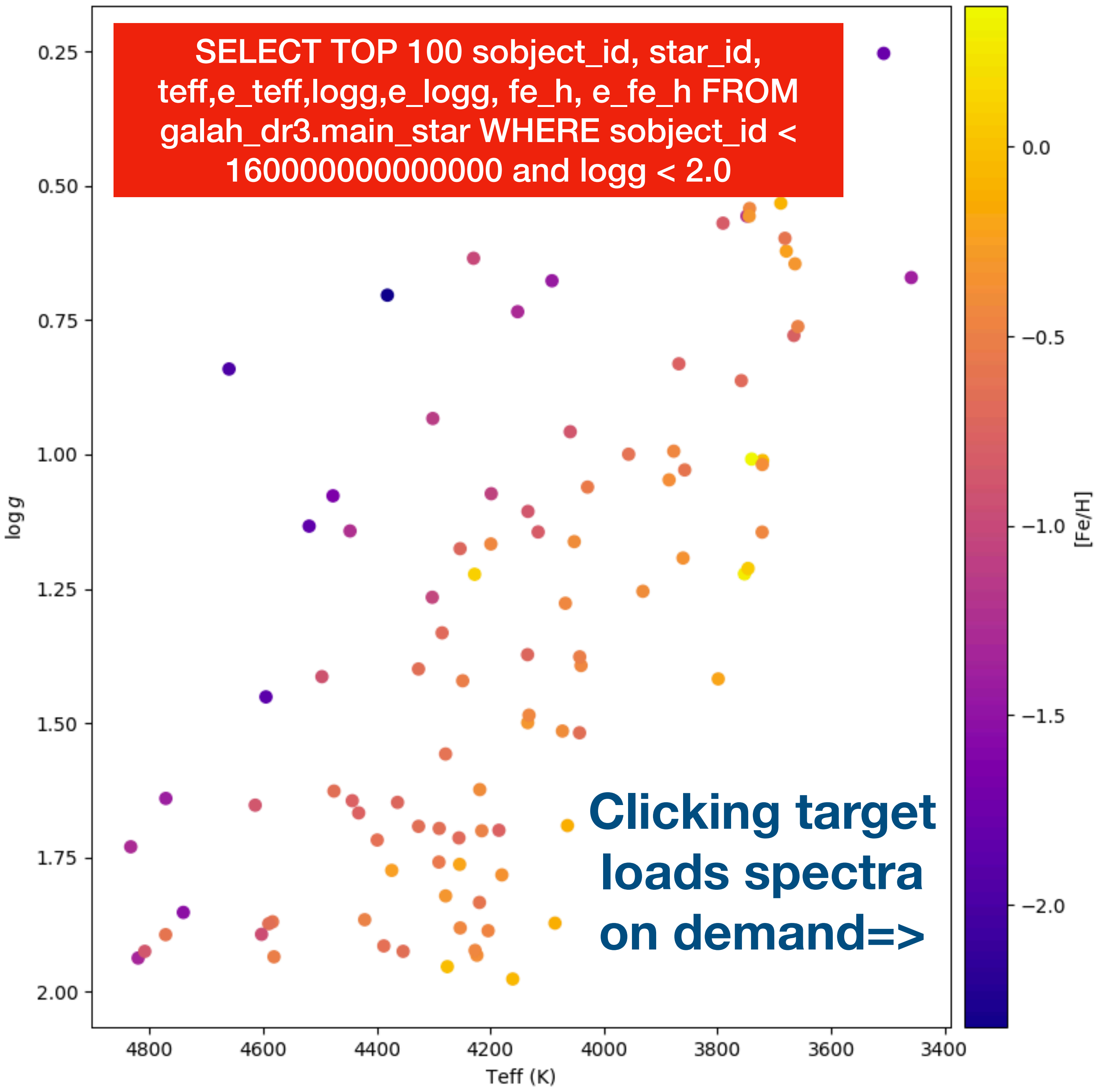
6dFGS: Generate PDFs of spectra and image cutouts

- **6dFGS** final data release spectra from SSA.
- 2MASS and DSS **HiPS** image cutouts from **hips2fits** service (CDS).
- Several page PDFs of plots: *supersedes* functionality of web archive (WFAU/ROE).



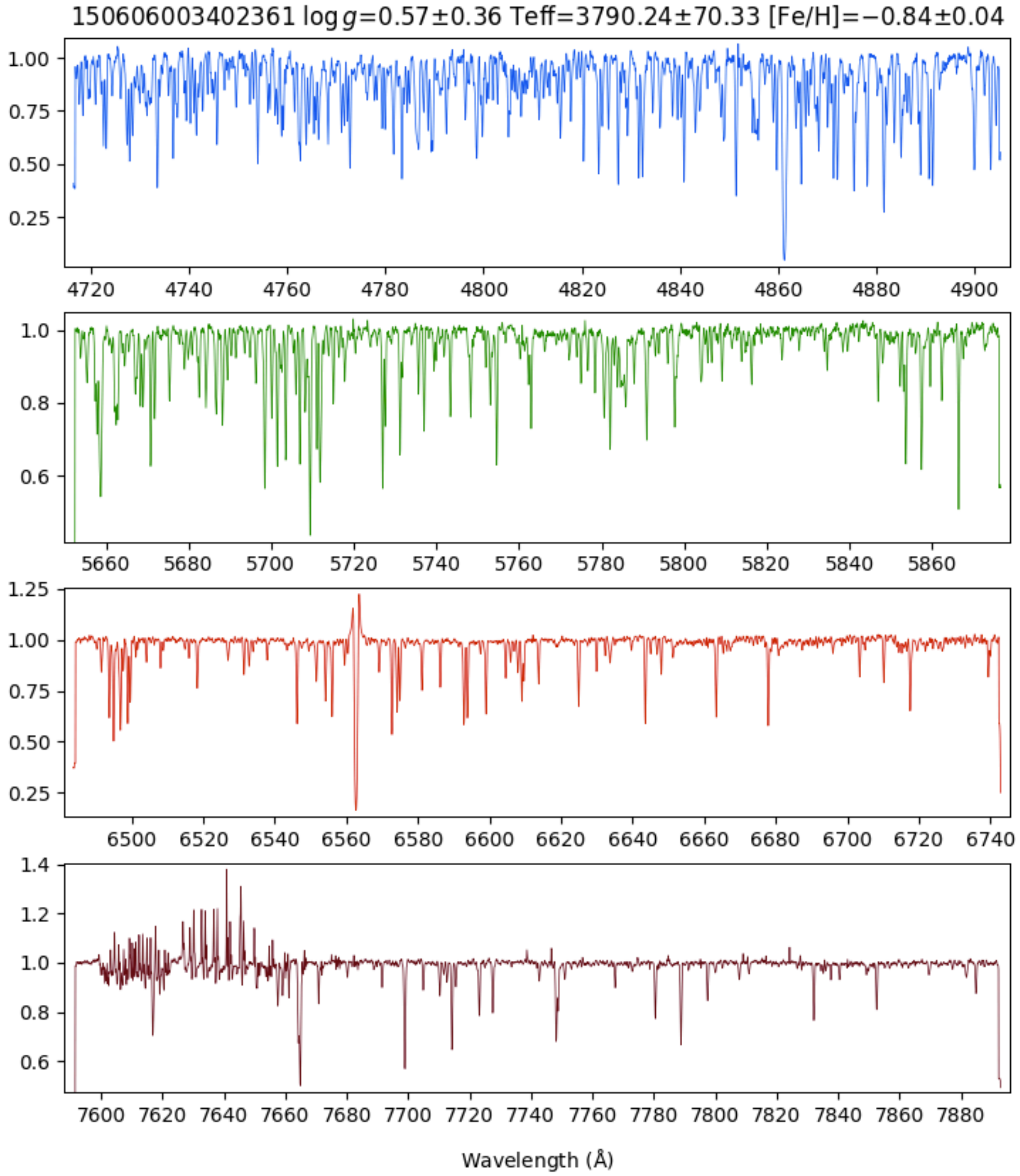
GALAH DR3 SSA + API

- **GALAH DR3** catalogue query via Data Central API => interactive spectrum viewer
- Use object id to easily get spectra from SSA service + params not in SSA (Teff, log g, [Fe/H], etc)



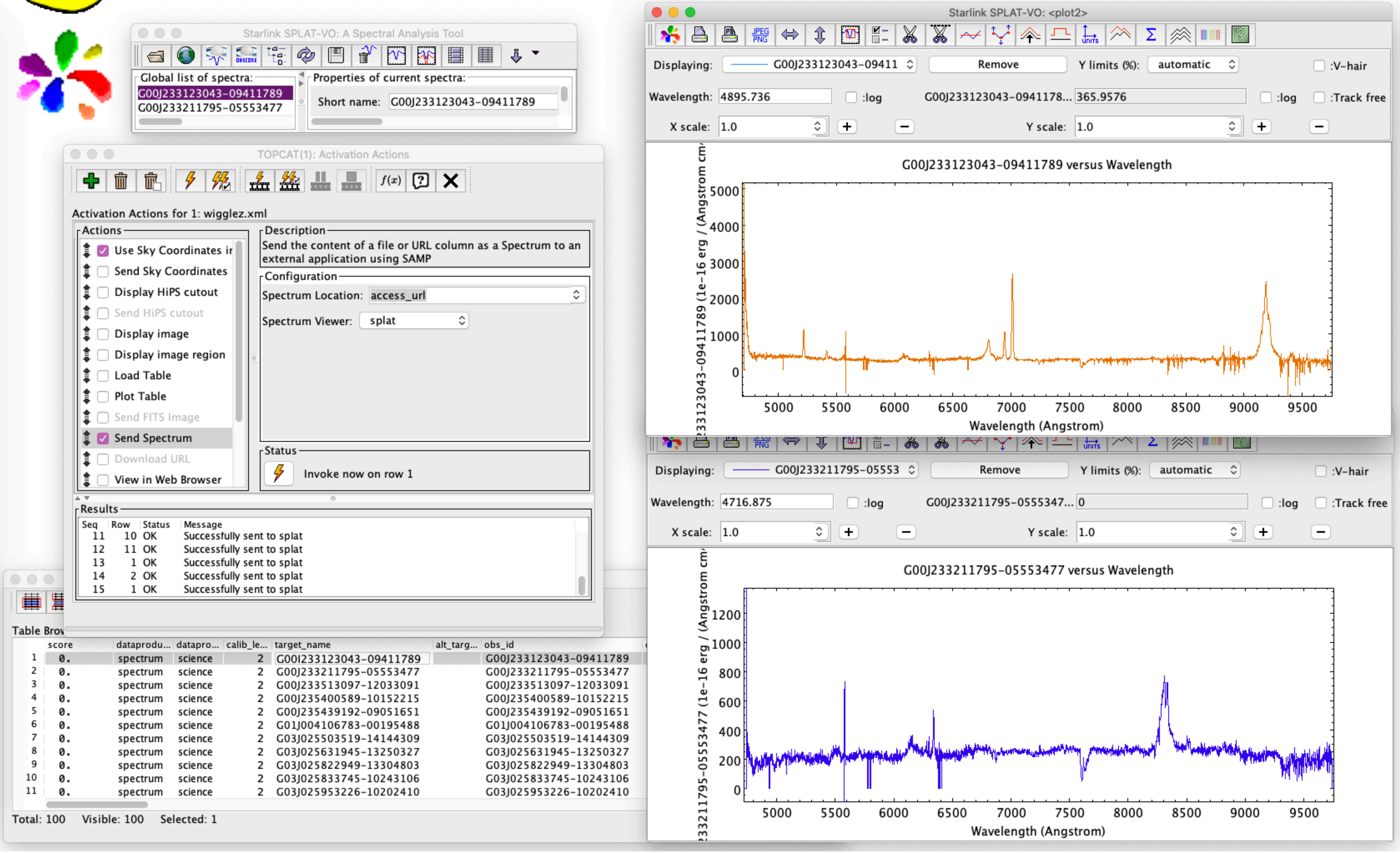
```
SELECT TOP 100 sobject_id, star_id,
teff,e_teff,logg,e_logg, fe_h, e_fe_h FROM
galah_dr3.main_star WHERE sobject_id <
160000000000000 and logg < 2.0
```

Clicking target loads spectra on demand=>



150606003402361 $\log g=0.57\pm0.36$ Teff=3790.24±70.33 [Fe/H]=−0.84±0.04

TOPCAT and SPLAT: quickly view SSA results



Starlink SPLAT-VO: A Spectral Analysis Tool

Global list of spectra:
G00J233123043-09411789
G00J233211795-05553477

Properties of current spectra:
Short name: G00J233123043-09411789

TOPCAT(1): Activation Actions

Activation Actions for 1: wigglez.xml

Actions
- ☑ Use Sky Coordinates ir
- □ Send Sky Coordinates
- □ Display HiPS cutout
- □ Send HiPS cutout
- □ Display image
- □ Display image region
- □ Load Table
- □ Plot Table
- □ Send FITS Image
- ☑ Send Spectrum
- □ Download URL
- □ View in Web Browser

Description
Send the content of a file or URL column as a Spectrum to an external application using SAMP

Configuration
Spectrum Location: access_url
Spectrum Viewer: splat

Status
Invoke now on row 1

Results

Seq	Row	Status	Message
11	10	OK	Successfully sent to splat
12	11	OK	Successfully sent to splat
13	1	OK	Successfully sent to splat
14	2	OK	Successfully sent to splat
15	1	OK	Successfully sent to splat

Table Brow

	score	dataprodu...	datapro...	calib_le...	target_name	alt_targ...	obs_id
1	0.	spectrum	science	2	G00J233123043-09411789		G00J233123043-09411789
2	0.	spectrum	science	2	G00J233211795-05553477		G00J233211795-05553477
3	0.	spectrum	science	2	G00J233513097-12033091		G00J233513097-12033091
4	0.	spectrum	science	2	G00J235400589-10152215		G00J235400589-10152215
5	0.	spectrum	science	2	G00J235439192-09051651		G00J235439192-09051651
6	0.	spectrum	science	2	G01J004106783-00195488		G01J004106783-00195488
7	0.	spectrum	science	2	G03J025503519-14144309		G03J025503519-14144309
8	0.	spectrum	science	2	G03J025631945-13250327		G03J025631945-13250327
9	0.	spectrum	science	2	G03J025822949-13304803		G03J025822949-13304803
10	0.	spectrum	science	2	G03J025833745-10243106		G03J025833745-10243106
11	0.	spectrum	science	2	G03J025953226-10202410		G03J025953226-10202410

Total: 100 Visible: 100 Selected: 1

Starlink SPLAT-VO: <plot2>

Displaying: G00J233123043-09411 Y limits (%): automatic
Wavelength: 4895.736 □ :log G00J233123043-0941178... 365.9576 □ :log □ :Track free □ :V-hair
X scale: 1.0 Y scale: 1.0

Displaying: G00J233211795-05553 Y limits (%): automatic
Wavelength: 4716.875 □ :log G00J233211795-0555347... 0 □ :log □ :Track free □ :V-hair
X scale: 1.0 Y scale: 1.0

TOPCAT: fundamentally helpful during SSA development!

SSA + AAT 2dF archive

- **2dFdr:** Data reduction of AAT 2dF spectra
- **Pipeline as A Web Service (PAWS):** On demand automated reduction (coming soon to AAT archive). **Fast:** reduce 960 science exposures in 48 minutes.
- Plan to retrospectively reduce archived 2dF observations and make available via SSA service
- **PAWS+SSA: Transient follow-up** enabled by quick-turnaround staging of spectra

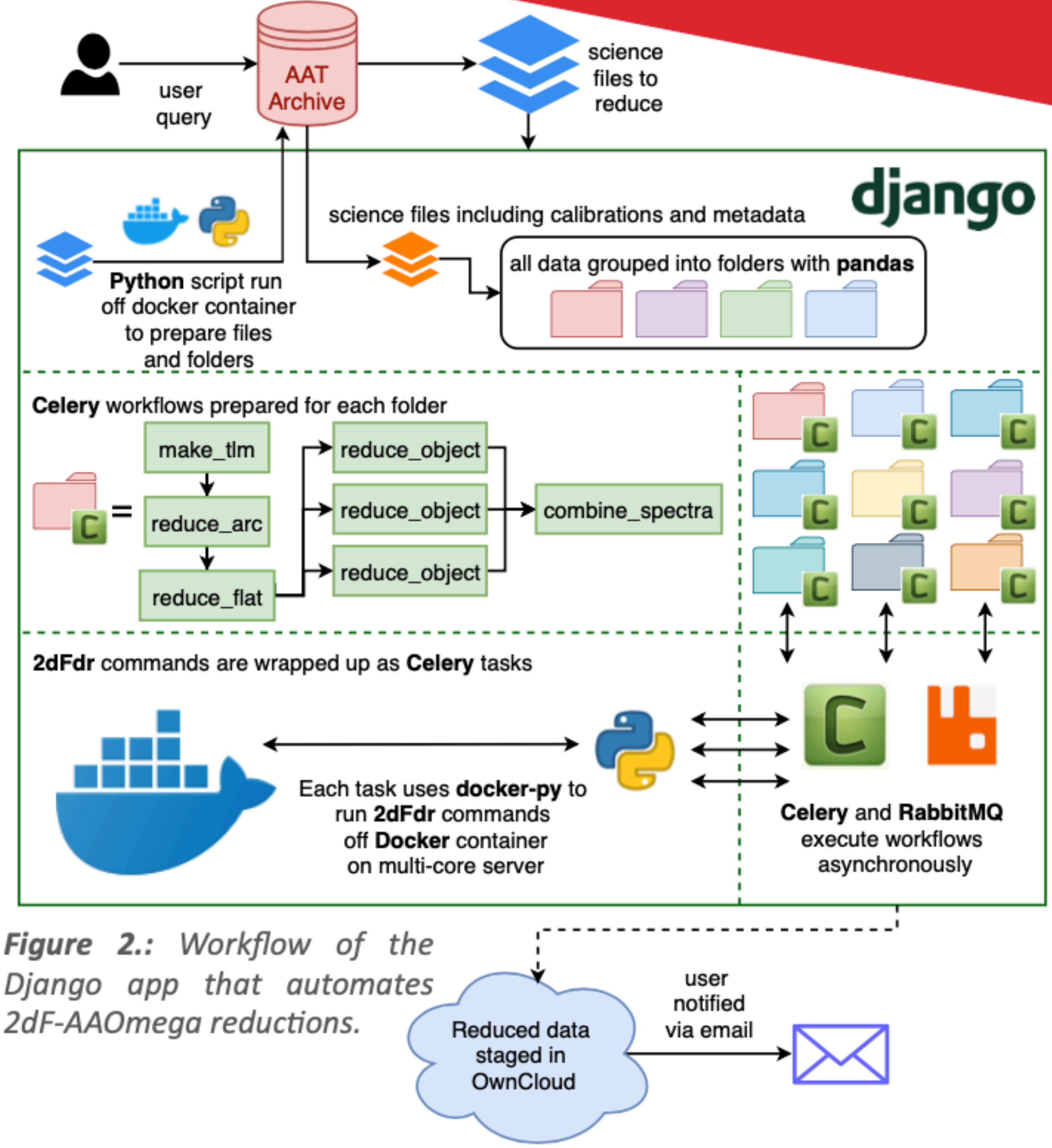
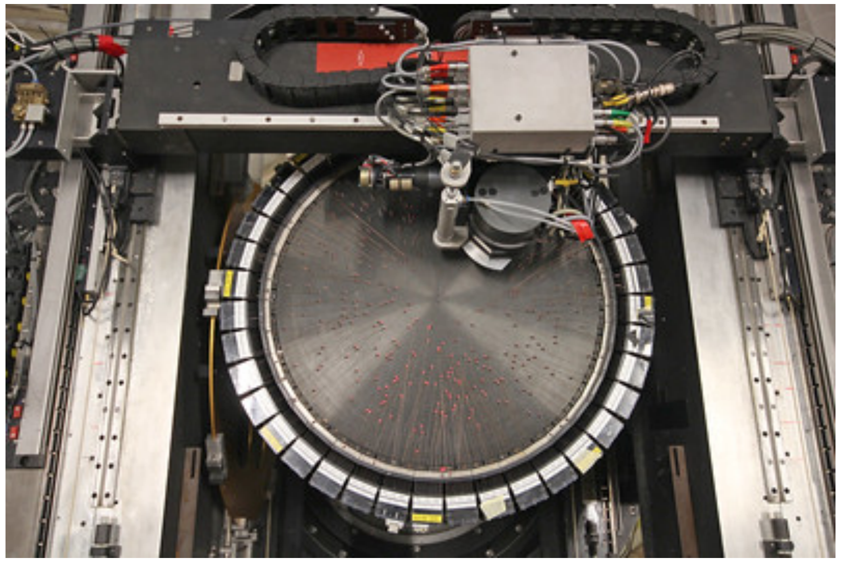

user query
AAT Archive
science files to reduce
django
Python script run off docker container to prepare files and folders
science files including calibrations and metadata
all data grouped into folders with pandas
Celery workflows prepared for each folder
C =
make_tlm
reduce_arc
reduce_flat
reduce_object
reduce_object
reduce_object
combine_spectra
C
2dFdr commands are wrapped up as Celery tasks
Each task uses docker-py to run 2dFdr commands off Docker container on multi-core server
Celery and RabbitMQ execute workflows asynchronously
Reduced data staged in OwnCloud
user notified via email

Figure 2.: Workflow of the Django app that automates 2dF-AAOmega reductions.



2dF field plate with 400 optical fibres
Credit: Ángel R. López Sánchez

Miszalski et al. 2020:
ADASS 2020 poster

Future plans

- Reduced and add archival 2dF spectra
- Python implementation: straightforward to incorporate new technologies
    - Could use STMOC as input, then filter results using **MOCPy**
- **Incorporate IFU survey data:** Currently do not include (e.g. SAMI DR3)
    - SODA/Datalink services could create maps, extract and coadd spectra, etc
- **TAP Wrapper:** Would allow for queries on all obscore information of SSA service
- Add PNG previews of spectra
- Register SSA and SIA services with **IVOA registries**
- **Generalise SIA/SSA:** one data discovery service for all real-world datasets including time series and data cubes. Use datalink for heavy lifting/customisation