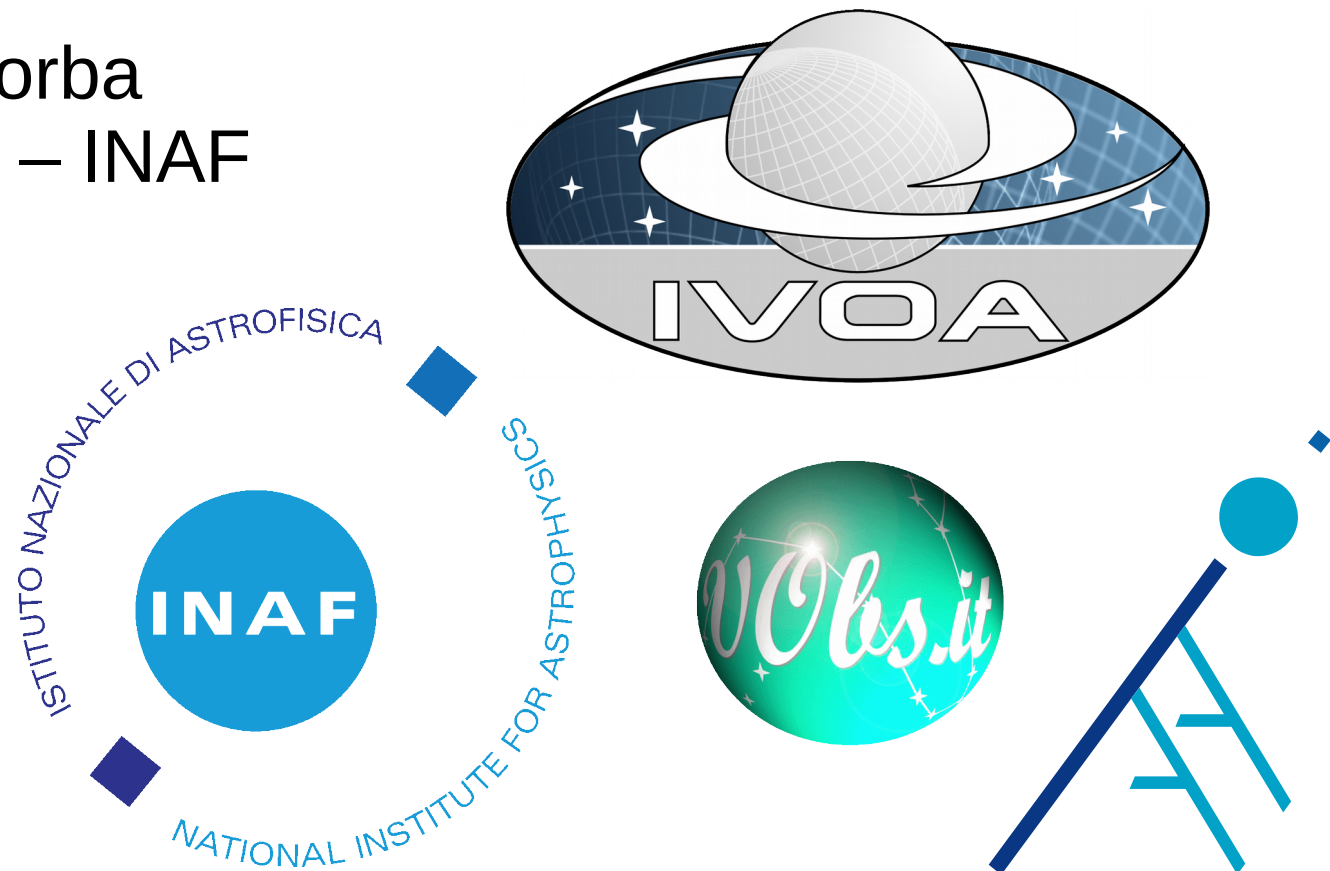


APOGEO: an automatic management system for astronomical portals

Sonia Zorba
IA2 – OATs – INAF





APOGEO – IA2



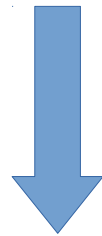
APOGEO Automatic **P**Ortal **G**enerat**O**r

IA2 – Italian Centre for Astronomical Archives

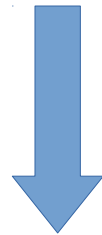
www.ia2.inaf.it

Hosts several web interfaces to astronomical archives:
TNG, LBT, Asiago, ...

Set up a TAP_SCHEMA



Configure / customize portal



Build .war and deploy

Preliminary step: set up a TAP_SCHEMA

tng_local tng_TAP_SCHEMA

Schema utype

Schema description

gia har lrs nic oig srg TNG

Table utype

Table description

Columns

- date_obs
- dec_c
- dec_rad
- detector
- exp_id
- idTNG

date_obs

Datatype: adql:DATE

Size: 0

STD

Principal

UType:

UCD:

Select portal tables and specify JOIN conditions

Main table: TNG

Secondary tables:

- nic (nic.exp_id -> TNG.exp_id)
- srg (srg.exp_id -> TNG.exp_id)
- oig (oig.exp_id -> TNG.exp_id)
- lrs (lrs.exp_id -> TNG.exp_id)

Add secondary table

Next >

Design the UI

+ Add component

+ Grid

main

</>



+ Add row

Show lattice

Name resolver

Object name...

Resolve

RA

Dec

Radius

Observ. Date

From

To

Observ. Time

From

To

Obs. Type

CALIBRATION

Instrum

NICS

Program

Object

Detector

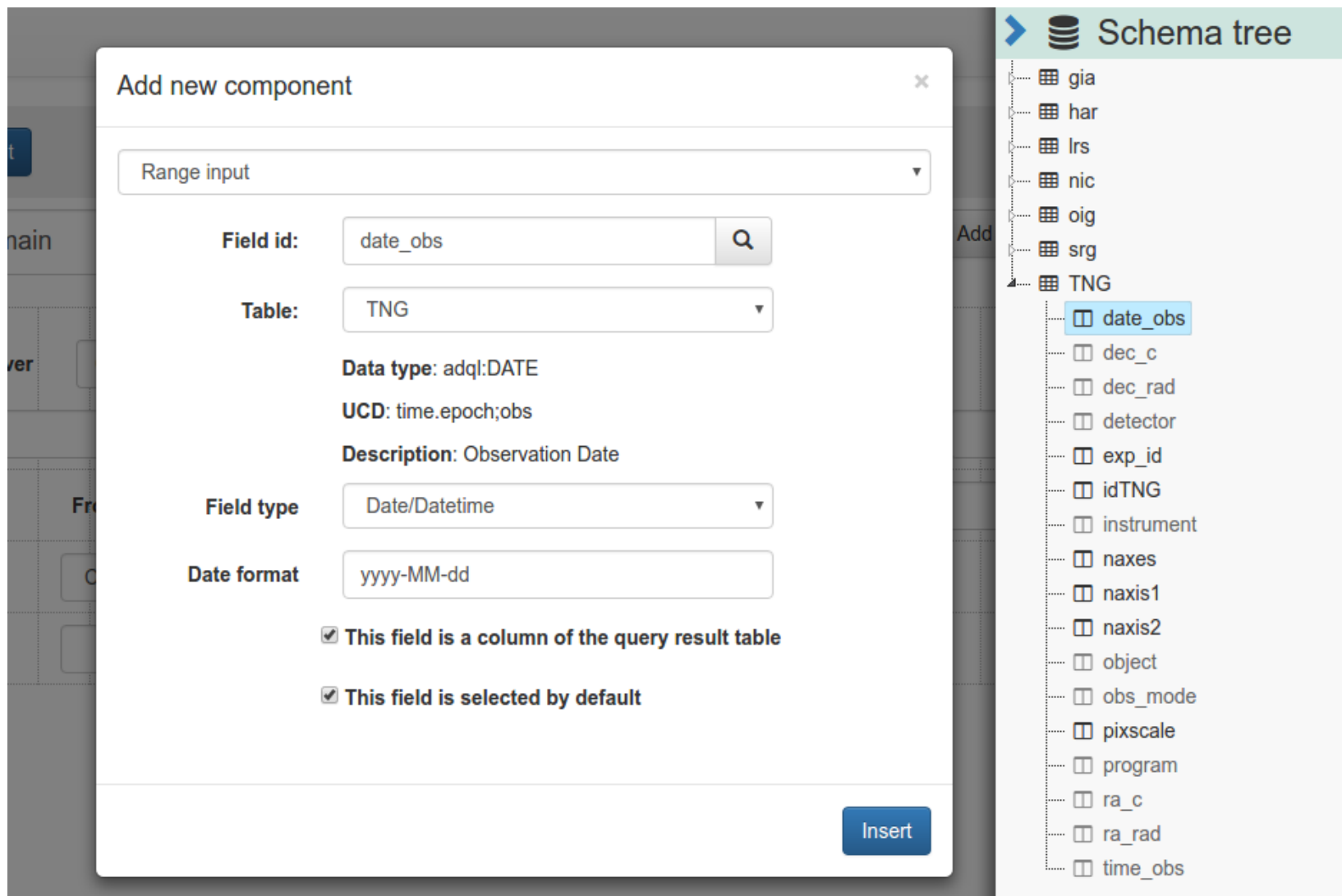
NICS

File name

< Previous

Next >

Each component is mapped to a database column



The screenshot displays the configuration interface for APOGEO. On the left, a dialog box titled "Add new component" is open, showing the configuration for a "Range input" component. The configuration includes:

- Field id:** date_obs
- Table:** TNG
- Data type:** adql:DATE
- UCD:** time.epoch;obs
- Description:** Observation Date
- Field type:** Date/Datetime
- Date format:** yyyy-MM-dd
- This field is a column of the query result table
- This field is selected by default

At the bottom right of the dialog is an "Insert" button. On the right side of the interface, a "Schema tree" panel is visible, showing a list of database tables and columns. The "TNG" table is expanded, and the "date_obs" column is highlighted in blue.

Schema tree

- gia
- har
- lrs
- nic
- oig
- srg
- TNG
 - date_obs
 - dec_c
 - dec_rad
 - detector
 - exp_id
 - idTNG
 - instrument
 - naxes
 - naxis1
 - naxis2
 - object
 - obs_mode
 - pixscale
 - program
 - ra_c
 - ra_rad
 - time_obs

.war package built via Maven Invoker API

Target application server

GlassFish Tomcat

Generate site

```
[INFO] Installing /home/sonia/generator/jq4g8d2tpn19d3ijnqoffika8r/pom.xml to
/home/sonia/.m2/repository/it/inaf/oats/ia2/DynamicPortal/1.0-SNAPSHOT/DynamicPortal-1.0-SNAPSHOT.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 5.685 s
[INFO] Finished at: 2016-10-14T10:09:44+02:00
[INFO] Final Memory: 42M/346M
[INFO] -----
```

Download .war

Currently: direct SQL query on MySQL
Planned: ADQL query on TAP

RA Dec Radius (arcmin)

Filename

Observ. Date From To

Obs. Type **Instrum** **Program**

Object

Exp. time From To








Night date From To

Airmass From To

Portal – Querying TAP



Main problem: we need pagination,
with total results count

LSQA0118.fts.gz 	2011-10-23	OBJECT	LRS	A24TAC_45
LSTA0046.fts.gz 	2011-10-26	CALIB	LRS	A24TAC_4
LSRA0067.fts.gz 	2011-10-23	OBJECT	LRS	TEST
LSDA0018.fts.gz 	2011-10-09	OBJECT	LRS	TEST
LSCZ0164.fts.gz 	2011-10-09	IMA	NICS	A24TAC_22
LSEZ0038.fts.gz 	2011-10-11	SPEHR	NICS	A24TAC_28
LSUZ0090.fts.gz 	2011-10-26	IMA	NICS	A24TAC_49

In MySQL, 2 queries:

```
SELECT SQL_CALC_FOUND_ROWS *  
FROM TNG  
LIMIT 40,20
```

```
SELECT FOUND_ROWS ()
```

In TAP, define an UDF in service capabilities, but

- not a real UDF
- hides 2 queries
- result in VOTable metadata

```
SELECT SQL_CALC_FOUND_ROWS ( ) TOP 10 *
FROM TNG
OFFSET 20
```

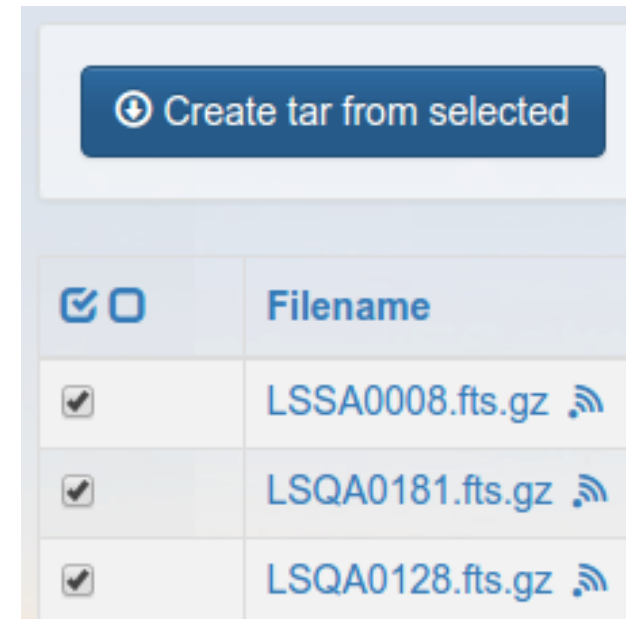
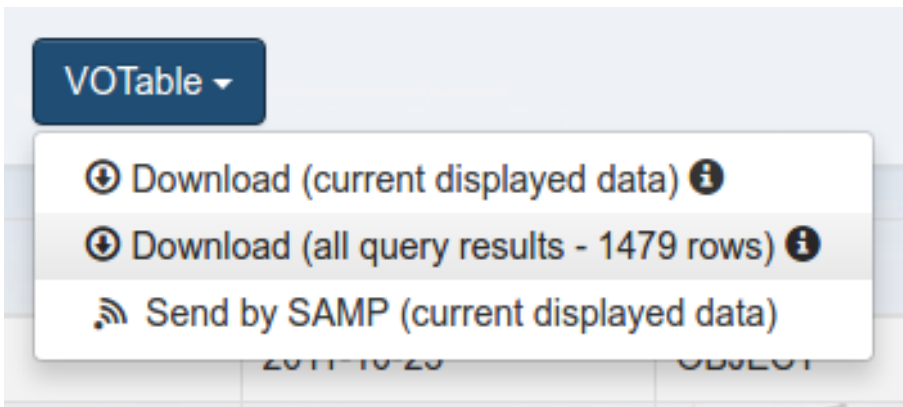
Table Parameters for 7: TAP_7_TNG

Name	Value	Description
Name	sync	Table name
Column Count	17	Number of columns
Row Count	10	Number of rows
QUERY_STATUS	OK	
TOTAL_COUNT	56	Total number of rows
QUERY	SELECT TOP 10 ...	
QUERY_STATUS	OK	

Portal – UWS

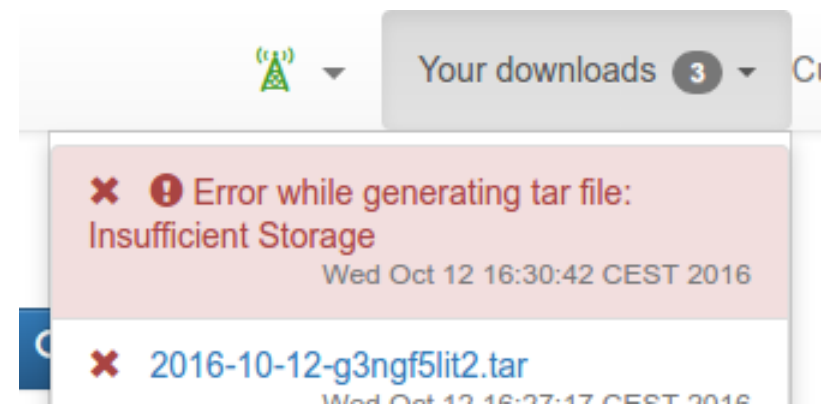
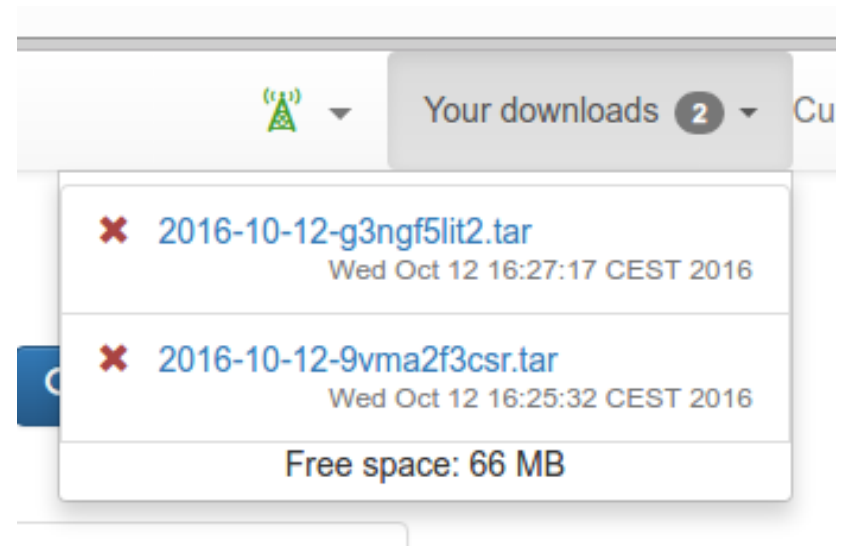
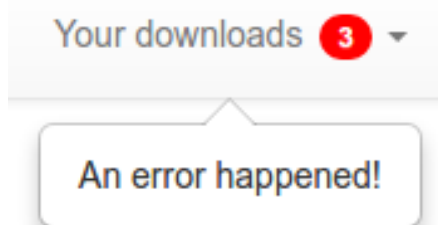
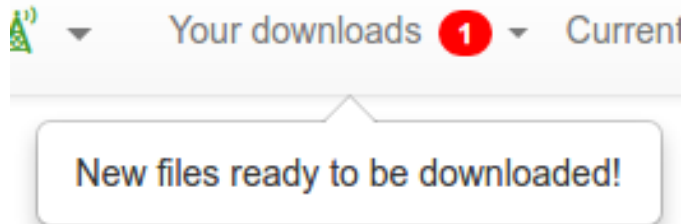
Portals use the UWS standard (OpenCADC libraries) for 2 asynchronous tasks:

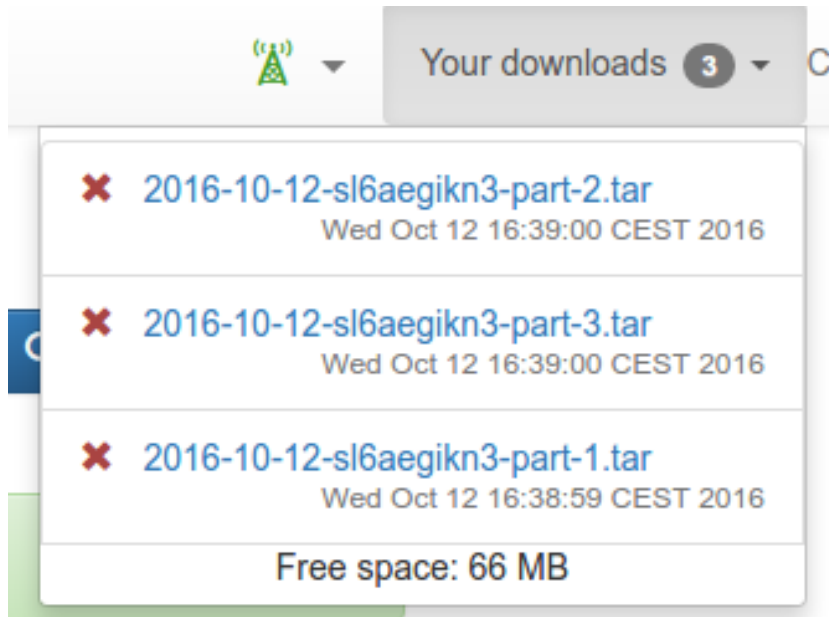
- Creation of big VOTables
- Creation of tar files from user selection



Portal – UWS

When UWS job is ended portal shows a notification





Tar and VOTables are split if they are too big, but we should allow deletion of single files.

We can't partially delete a job.

UWS specification:

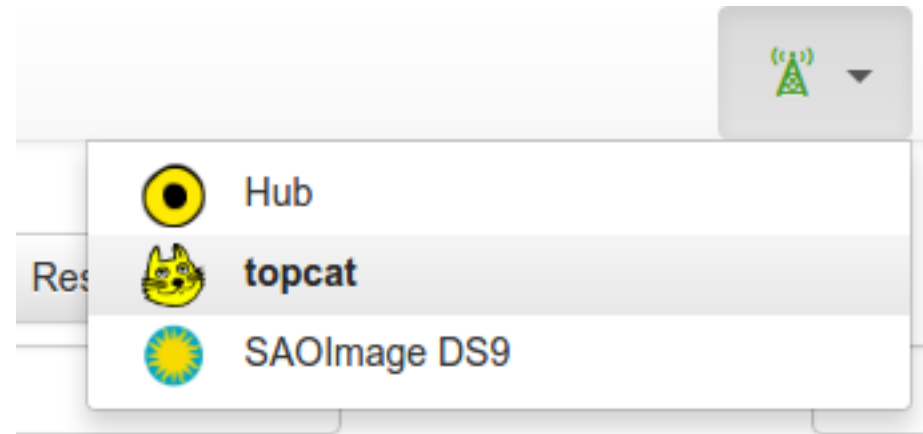
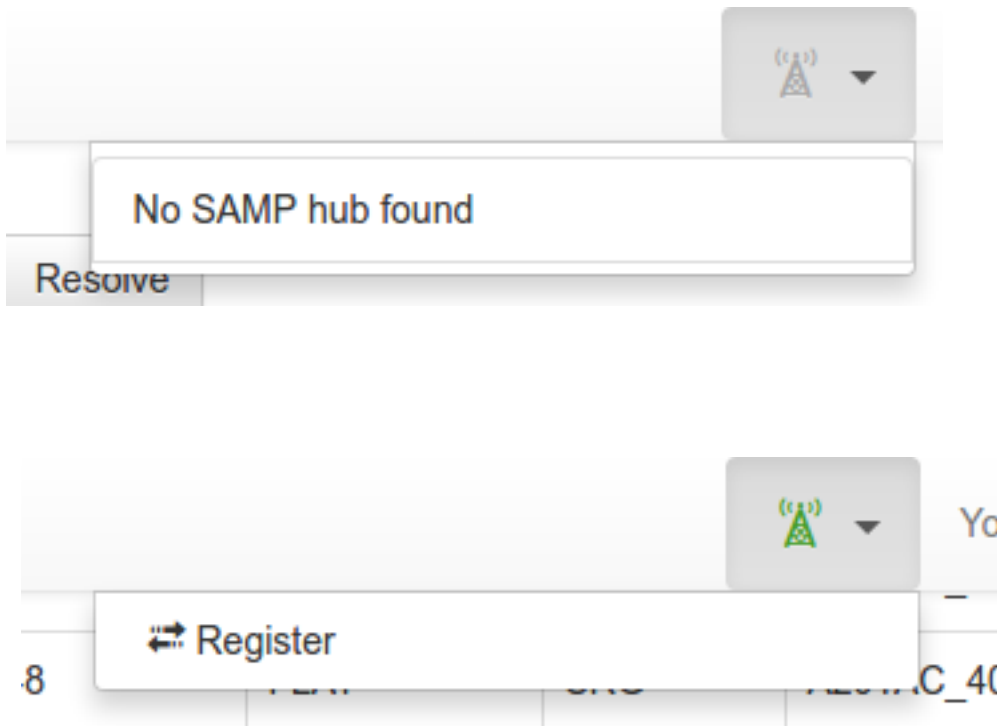
"The children of the Results List may be read but not updated or deleted. The client cannot add anything to the Results List."

Solution: user space managed by a different service

...in the future a VOspace!






SAMP

Portals can send VOTables and fit files via SAMP Web Profile, using the samp.js library.



Main problem: private data.

- Best approach: HTTPS + Authentication
- Current approach: temporary public URLs, stored in the user session

<input checked="" type="checkbox"/>	Filename	Observ. D
<input type="checkbox"/>	LRS.2012-10-13T04-29-03.117.fits.gz 	2012-10-1
<input type="checkbox"/>	LRS.2012-10-14T14-14-06.215.fits.gz 	2012-10-1
<input type="checkbox"/>	LRS.2012-10-13T19-56-13.895.fits.gz 	2012-10-1
<input type="checkbox"/>	LRS.2012-10-14T00-15-56.635.fits.gz 	2012-10-1
<input type="checkbox"/>	LRS.2012-10-14T00-49-04.672.fits.gz 	2012-10-1

SAMP – FITS example



POST /samp/file/KALA0087.FTS.gz

Response: tmsfq9nfp99



FileKey (random string)
Map the path of the file

SAMP Message:

- samp.mtype: image.load.fits
- samp.params:
 - url: http://<portal-host>/samp/file/tmsfq9nfp99



Temporary URL

SAMP – VOTables

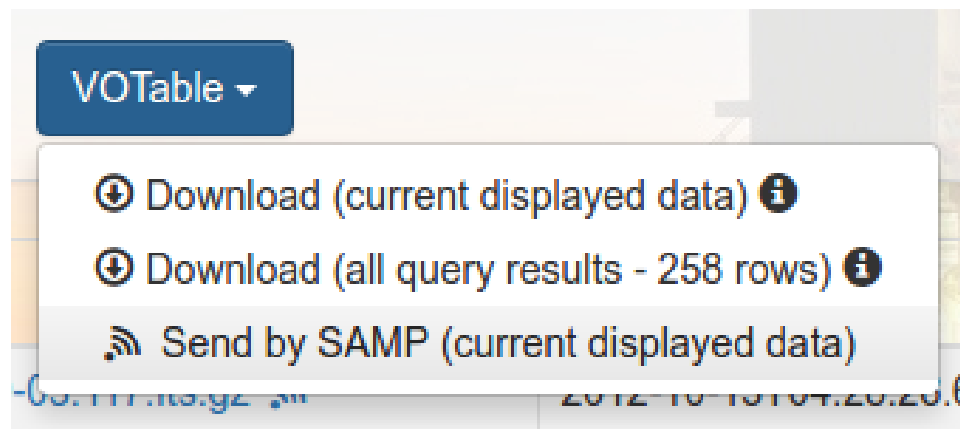
Main problem: VOTables could be big and need to be stored into files.

Current approach:

- random string is generated at login
- same url for all calls during an user session

SAMP Message:

- `samp.mtype: table.load.votable`
- `samp.params:`
 - `url: http://<portal-host>/samp/VOTable/6vk9jk16o37`



The endpoint `/samp/VOTable/<samp-session-key>` returns a VOTable built from the current displayed rows on the portal.

- Current rows are objects in memory → VOTable generation is very fast.
- No needs for creating temporary files: VOTable is written directly in the response output stream

Thanks to



- CDS Name Resolver and UCD web services
<http://cds.u-strasbg.fr/cgi-bin/Sesame>
<http://cds.u-strasbg.fr/UCD/tools.htx>
- OpenCADC
<https://github.com/opencadc>
- STIL
<http://www.star.bris.ac.uk/~mbt/stil/>
- Sampjs
<http://astrojs.github.io/sampjs/>
- UCD Validator
<https://github.com/gmantele/ucdvalidator>

Thanks for your attention!

APOGEO team:

Sonia Zorba, Andrea Bignamini, Francesco Cepparo,
Cristina Knapic, Marco Molinaro, Riccardo Smareglia