ODS Database API Operation Description

2024.05.15

2024.01.05

Added "corr_integ_time_sec" and "slew_sec" ods_data attribute descriptions to the public documentation page.

> On 11/1/23 13:54, Daniel Faes wrote:
>
> Hi Bang and all,
>
> I inform you that the VLA data sender is already generating two new JSON fields:
>
> - "corr_integ_time_sec": correlator integration time (float type). Currently, the field is only generated if "src_is_pulsar_bool=false"
>
> - "slew_sec": the time taken for the array to reach the source (counted from "src_start_utc"; float type)

2023.10.17

This is to summarize the current state of the 'helloworld' version of this API.

The ODS REST/API server now listens on two ports:

8191 - is the HTTPS port that SpaceX and other external entities will use. It uses the URL:
https://obs.vla.nrao.edu:8191/ and is limited to the following paths only:
- /ods_data -- for only the upcoming observations, no date-range fetches allowed.
- /docs -- sends the document page (that only documents external functions)
- /openapi.yaml -- semds the raw openapi yaml file

8192 - is the HTTP port that will be used internally and has wider access to the ODS database. It uses the URL: http://turing.aoc.nrao.edu:8192/ and can do all things that the server supports which is currently the following paths:
- /ods_data -- as above but also includes time-range fetches.
- /ods_put -- facilitates adding data to the database
- /docs -- sends the document page (that only documents external functions)
- /openapi.yaml -- semds the raw openapi yaml file
- /openapi.json -- semds the raw openapi json file

See below for more detailed descriptions of the 'ods_data' and 'ods_put' operations.

2023.10.17

Made it so the server also supports an internal URL:

http://turing.aoc.nrao.edu:8192/

That gives broader access to the database (adding data to it and fetching data by time range).

2023.10.16

Made the server available to the outside world using HTTPS on the following URL:

https://obs.vla.nrao.edu:8191/

2023.10.11

We added the ability to fetch ods_data from the database by time range. This is done by using 'from' and 'to' attributes in the query portion of the fetch URL. It is described in greater detail below.

2023.10.02

Changed everything from Observatory Status Sharing (OSS) to Operational Data Sharing (ODS) (it was discovered that there already is an existing OSS at NRAO.

(Original Description)

We have the initial version of the ODS Database REST API up and running and Daniel and Cat are populating the database with it from their VLA ODS data senders.

The API will provide information about all upcoming observation sources (scans?) It does this by fetching records from the database whose 'src_end_utc' > now().

The API also provides a mechanism for adding information to the database without having to know about the database itself.

~~The API has not been exposed outside of NRAO networks yet.~~

The REST URLs for getting and putting data to the ODS are explained here:

Getting ODS Data from the database:

The data is sent from the database as an array of JSON objects where each object contains information about a single observation source. Information from all sites is sent in the same response and the information is sorted by by 'src_start_utc'

To get all database entries with a src_end_utc greater than now:

http://turing.aoc.nrao.edu:8192/ods_data

or

https://obs.vla.nrao.edu:8191/ods_data

Example Response:

{"ods_data":[

{"notes":"inAdv:True","src_id":"3C147","src_ra":85.65057465,"site_id":"vla","src_dec":49.85200932222222,"freq_lower":4000000000,"freq_upper":8000000000,"src_end_utc":"2023-06-28T14:00:06.000103","trk_rate_ra":0,"trk_rate_dec":0,"src_start_utc":"2023-06-28T13:51:09.000048","site_lat":34.07861,"site_lon":-107.61806,"site_elev":2115},

{"notes":"inAdv:True","src_id":"3C147","src_ra":85.65057465,"site_id":"vla","src_dec":49.85200932222222,"freq_lower":1000000000,"freq_upper":2000000000,"src_end_utc":"2023-06-28T14:01:05.000776","trk_rate_ra":0,"trk_rate_dec":0,"src_start_utc":"2023-06-28T14:00:06.000103","site_lat":34.07861,"site_lon":-107.61806,"site_elev":2115},

{"notes":"inAdv:True","src_id":"3C147","src_ra":85.65057465,"site_id":"vla","src_dec":49.85200932222222,"freq_lower":1000000000,"freq_upper":2000000000,"src_end_utc":"2023-06-28T14:06:04.000140","trk_rate_ra":0,"trk_rate_dec":0,"src_start_utc":"2023-06-28T14:01:05.000776","site_lat":34.07861,"site_lon":-107.61806,"site_elev":2115},

{"notes":"inAdv:True","src_id":"J0555+3948","src_ra":88.8783567,"site_id":"vla","src_dec":39.813656944444446,"freq_lower":1000000000,"freq_upper":2000000000,"src_end_utc":"2023-06-28T14:11:02.000504","trk_rate_ra":0,"trk_rate_dec":0,"src_start_utc":"2023-06-28T14:06:04.000140","site_lat":34.07861,"site_lon":-107.61806,"site_elev":2115},

{"notes":"inAdv:True","src_id":"FRB121102A","src_ra":82.99458333333334,"site_id":"vla","src_dec":33.14791666666667,"freq_lower":1000000000,"freq_upper":2000000000,"src_end_utc":"2023-06-28T14:22:48.000632","trk_rate_ra":0,"trk_rate_dec":0,"src_start_utc":"2023-06-28T14:11:02.000504","site_lat":34.07861,"site_lon":-107.61806,"site_elev":2115}

]}

If there is no data to fetch the response will be an empty array:

{"ods_data":[]}

Getting ODS Data from the database by time range (Internal Use Only):

2023.10.11 We added the ability to fetch ods_data from the database by time range. This is done by using 'from' and 'to' attributes in the query portion of the fetch URL:

http://turing.aoc.nrao.edu:8192/ods_data?from='2023-10-04 21:00:00'&to='2023-10-04 23:00:00'

That will return all ods_data entries that were added to the database between those times.

The format for the timestamp is: '2023-10-04 21:00:00'. It must be quoted and contain a space between the date and time portions. A database query error message will be returned if the timestamp is malformed.

The following rules apply:

if 'from' & 'to' are absent - the ods_data for 'src_end_utc' >= now() are fetched (the same as described above)

if 'from' & 'to' are both present - fetch ods_data that was entered into the database between those times

if 'from' only is present - fetch ods_data that was entered into database starting at that time and ending with the last entry in the database

if 'to' only is present - fetch ods_data from the beginning of the database up to that time.

Adding ODS data to the database (Internal Use Only):

Data is added one row at a time; each row provides the antenna information for a single observation source.

There are two ways to add a row to the database.

1) HTTP GET where the JSON is in the URL:

http://turing.aoc.nrao.edu:8192/ods_put?{"notes":"inAdv:True","src_id":"FRB121102A","src_ra":82.99458333333334,"site_id":"vla","src_dec":33.14791666666667,"freq_lower":1000000000,"freq_upper":2000000000,"src_end_utc":"2023-06-28T14:22:48.000632","trk_rate_ra":0,"trk_rate_dec":0,"src_start_utc":"2023-06-28T14:11:02.000504","site_lat":34.07861,"site_lon":-107.61806,"site_elev":2115}]}

You can enter that URL into a Web Browser and it will add the row to the database. Note, if you send the URL from your own program, you'll likely have to URL Encode it.

2) HTTP POST where the JSON is in the HTTP Request body:

http://turing.aoc.nrao.edu:8192/ods_put

If the PUTs worked properly, you should get the following JSON response:

{"ods_response":"INSERTed 1 row(s)"}

If not, you should get some sort of error message (in a JSON object).