



Table of Contents

1. General	2
2. Change proprietary period end date	3
3. Search objects	4
4. Get object (details/entries)	6
5. Get file	8
6. Feedback messages	9

1. General

This manual serves for explanation of the additional APIs maintained by the TNS, for changing proprietary period end date and retrieving object lists and object details by various criteria. The *Bulk report* API is described in the corresponding *Bulk report manual*.

As opposed to the *Asynchronous Bulk report* mechanism, the APIs described here are *synchronous*; i.e. a reply JSON is obtained immediately, without having to check for a reply for a given report-id.

We have set up on our *sandbox* site an API TEST FORM webpage where it is possible to experiment with the various APIs. The URL of the API TEST FORM page is:

<https://sandbox.wis-tns.org/api>

where three forms are available for experimenting with the following APIs:

- *Bulk report* (as mentioned, refer to the *Bulk report manual*)
- *Change prop period*
- *Search*, which includes the *Search*, *Get object*, and *Get file* APIs.

An `api_key` and `tns_marker` should always be provided (defining the sending Bot). The input JSON-formatted data is placed in the relevant section's JSON editor, and upon submission, the reply JSON is displayed on the same editor.

Further explanations about the sandbox can be found in *Sec. 2* of the *Bulk report manual*.

Explanations about the Bots and the API-Keys can be found in *Sec. 3* of the *Bulk report manual*.

A PHP sample code is provided for assisting with the development of the *Change prop. period end date* and the *Get object* APIs, as examples, and this is explained in *Sec. 7*.

Do not hesitate to [contact us](#) for any assistance and/or clarifications.

2. Change proprietary period end date

2.1. General

This API serves for updating the prop. period end date of reported entries, for switching an AT report and/or classification report and/or spectra to public, either on the same day or some future date (back-dating is not possible). Only a bot that is associated with the group that submitted the report (AT/Classification/spectra) can request to perform such a change to the prop. period end date. The sample code as described in Sec. 8 provides an example for this API.

2.2. URL: <https://www.wis-tns.org/api/set/prop-period>

Sandbox URL: <https://sandbox.wis-tns.org/api/set/prop-period>

2.3. Data description

- Type: POST
- Mandatory parameters:
 - `api_key` – the bot's API Key
 - `User-Agent` – the bot's `tns_marker`
 - `data` – holds the JSON report
- See the `tns_2_api_change_prop_period.py` sample code.

2.4. JSON key-value list

- Specified in **red** are mandatory keys; Not part of the JSON: in **green** previous keyword names (will still hold for some time but better to replace with the new names), and clarifications in **blue**.
- It is possible to change the end prop. period date for any of the following reported entries (by toggling the '0' / '1' values): AT report (together with the associated photometry points), Classification report, and the uploaded spectra.

```
{
  "objname": "",
  "reporting_groupid": "",
  "end_prop_period_date": "",
  "at": "1", // "1" or "0" for updating or not the relevant
  "classification": "0", // ATs/classifications/spectra.
  "spectra": "0",
  "frb": "0"
}
```

2.5. If no errors, the reply feedback contains counters of the updated entries, e.g.:

```
{
  "id_code": 200,
  "id_message": "OK",
  "data": {
    "affected_rows": {
      "at": {
        "at_reps_count": "1",
```

```

    "obj_count": "1",
    "phot_count": "2"
  },
  "classification": 0,
  "spectra": 0
}
}
}

```

It is recommended to check the relevant counter/s for verifying that the operation was successful. Relevant possible errors are e.g. {0,6,7}, as specified in the feedback messages table in Sec. 6.

3. Search objects

3.1. General

This API serves for listing the object names (objname, prefix and id) satisfying given search criteria. The search criteria are matching the query fields available on the Search page (applying “AND” operator for whichever combination of parameters), in particular:

- Cone search for given RA/DEC, radius & units.
If RA/DEC are specified without values for radius/units, the default values used are 3 arcsec.
- Object name (like)
- Internal name (like)
- Etc...

Access permissions are taken into account. If an object is proprietary (not public), only a Bot that is associated with its reporting (source) group will have access to it. Likewise, for the following *Get object* API.

3.2. URL: <https://www.wis-tns.org/api/get/search>

Sandbox URL: <https://sandbox.wis-tns.org/api/get/search>

3.3. Data description

- Type: POST
- Mandatory parameters:
 - api_key – the bot’s API Key
 - User-Agent – the bot’s tns_marker
 - data – holds the JSON report
- See the tns_2_api_search.py sample code, Search obj.

3.4. JSON key-value list

e.g. A cone search, including a given internal name:

```

{
  "ra": "05:24:18.0",
  "dec": "+09:10:37.0",
  "radius": "5",
  "units": "arcsec",
  "objid": "",

```

```

"objname": "",
"objname_exact_match": 0,
"internal_name": "Gaia16azh",
"internal_name_exact_match": 0,
"public_timestamp": "",
"reported_period_value": "",
"reported_period_units": "months",
"unclassified_at": "0",
"classified_sne": "0",
"include_frb": "0",
"name": "",
"name_like": "0",
"isTNS_AT": "all",
"public": "all",
"coords_unit": "arcsec",
"reporting_groupid": [],
"data_source_groupid": [],
"classifier_groupid": [],
"objtype": [],
"at_type": [],
"discovery_date_start": "",
"discovery_date_end": "",
"discovery_mag_min": "",
"discovery_mag_max": "",
"discoverer": "",
"classifier": "",
"spectra_count": "",
"redshift_min": "",
"redshift_max": "",
"hostname": "",
"ext_catid": "",
"ra_range_min": "",
"ra_range_max": "",
"decl_range_min": "",
"decl_range_max": "",
"discovery_instrument": [],
"classification_instrument": [],
"associated_groups": [],
"official_discovery": "",
"official_classification": "",
"at_rep_remarks": "",
"class_rep_remarks": "",
"frb_repeat": "",
"frb_repeater_of_objid": "",
"frb_measured_redshift": "",
"frb_dm_range_min": "",
"frb_dm_range_max": "",
"frb_rm_range_min": "",
"frb_rm_range_max": "",
"frb_snr_range_min": "",
"frb_snr_range_max": "",
"frb_flux_range_min": "",
"frb_flux_range_max": ""
}

```

3.5. The JSON reply array contains a list of object entries; e.g.:

```
{
  "id_code": 200,
  "id_message": "OK",
  "data": [
    {
      "objname": "2016evp",
      "prefix": "AT",
      "objid": "13984"
    }
  ]
}
```

4. Get object (details/entries)

4.1. General

This API retrieves object details per given object name, including its associated photometry and/or spectra if requested.

The object name should be exact (e.g. the 'objname' value as obtained by the previous *Search objects* API), and without prefix; e.g. '2016abc'.

If the object's spectra are requested, the ascii/FITS files of the spectrum/a will be specified in the JSON key-value reply list. To download the actual file/s use the following *Get file* API.

4.2. URL: <https://www.wis-tns.org/api/get/object>

Sandbox URL: <https://sandbox.wis-tns.org/api/get/object>

4.3. Data description

- Type: POST
- Mandatory parameters:
 - api_key – the bot's API Key
 - User-Agent – the bot's tns_marker
 - data – holds the JSON report
- See the tns_2_api_search.py sample code, Get obj.

4.4. JSON key-value list

```
{
  "objname": "2023wtq",
  "objid": "",
  "photometry": "0",    // "1" or "0" for retrieving or not the associated photometry/spectra
  "spectra": "1"
}
```

4.5. The reply JSON contains the object details. Spectra section within the reply may look like, e.g.:

```
  "spectra": [  
  {  
    "obsdate": "2023-11-27 05:13:02",  
    "jd": 2460275.717384259,  
    "public": "1",  
    "end_prop_period": null,  
    "exptime": "2400",  
    "source_groupid": "63",  
    "source_group_name": "ePESSTO+",  
    "observer": "M. Kopsacheili, M. Gonzalez-Banuelos",  
    "reducer": "C. Gutierrez, M. Kopsacheili, M. Gonzalez-Banuelos",  
    "asciifile": "https://tns2.wis-tns.org/system/files/uploaded/ePESSTO%2B/tns_2023wtq_2023-11-  
27_05-13-02_ESO-NTT_EFOSC2-NTT_ePESSTO%2B.ascii",  
    "fitsfile": "https://tns2.wis-tns.org/system/files/uploaded/ePESSTO%2B/tns_2023wtq_2023-11-  
27_05-13-02_ESO-NTT_EFOSC2-NTT_ePESSTO%2B.fits",  
    "remarks": "",  
    "source_group": {  
      "id": "63",  
      "name": "ePESSTO+"  
    },  
    "instrument": {  
      "id": "31",  
      "name": "EFOSC2-NTT"  
    },  
    "telescope": {  
      "id": "1",  
      "name": "Object"  
    }  
  },  
  {  
    "obsdate": "2023-11-13 06:17:07",  
    "jd": 2460261.761886574,  
    "public": "1",  
    "end_prop_period": null,  
    "exptime": "900",  
    "source_groupid": "63",  
    "source_group_name": "ePESSTO+",  
    "observer": "T. Moore, A. Amer, P. Ramsden",  
    "reducer": "S. Srivastav, C. Angus",  
    "asciifile": "https://tns2.wis-tns.org/system/files/uploaded/ePESSTO%2B/tns_2023wtq_2023-11-  
13_06-17-07_ESO-NTT_EFOSC2-NTT_ePESSTO%2B.ascii",  
    "fitsfile": "https://tns2.wis-tns.org/system/files/uploaded/ePESSTO%2B/tns_2023wtq_2023-11-  
13_06-17-07_ESO-NTT_EFOSC2-NTT_ePESSTO%2B.fits",  
    "remarks": "",  
    "source_group": {  
      "id": "63",  
      "name": "ePESSTO+"  
    },  
    "instrument": {  
      "id": "31",  
      "name": "EFOSC2-NTT"  
    },  
    "telescope": {  
      "id": "1",  
      "name": "Object"  
    }  
  }  
]
```

}

5. Get file

5.1. General

This API serves for downloading a given file via providing the appropriate URL, as e.g. obtained from the *Get object* API.

Clearly, a check is performed that the requesting Bot (via the supplied `api_key`) has access permission to the required URL.

Files on the public domain (not proprietary) can be accessed without providing an `api_key`.

5.2. URL: The file's URL; e.g. from the 'asciifile' key above:

```
https://www.wis-tns.org/system/files/uploaded/None/None_2016G_2016-01-10_14:42:31_LJT_YFOOSC.dat
```

5.3. Data description

- Type: POST
- Mandatory parameters:
 - `api_key` – the bot's API Key
 - User-Agent – the bot's `tns_marker`

6. Feedback messages

6.1. General

The feedback messages can be easily identified by their message id's, allowing for efficient parsing of the resulting feedbacks. We list here the mapping between the message id's and their corresponding texts. (The majority of the listed messages are relevant for the Bulk report API; messages 7,10,110 are relevant for the APIs of this manual.)

6.2. Mapping of the messages

Message id	Message	Provided values
General		
200	OK	
201	Created	
400	Bad request	
401	Unauthorized	
403	Forbidden	
404	Not Found	
Blocking errors		
0	Invalid	
1	Last non-detection should precede the Discovery Datetime	
2	At least one Photometry point - that of the discovery - should be filled	
3	Required field	
4	Proprietary period cannot extend more than 100 years	
5	An identical AT report (sender, RA/DEC, discovery date) already exists	
6	Last non-detection or archival info must be filled	
7	Proprietary period cannot be backdated	
10	Unauthorized	
20	ASCII file or FITS file must be uploaded	
21	At least one Spectra should be filled	
22	An ASCII file must be uploaded	
Feedbacks		
100	Transient object was inserted	objname (eg '2016abc')
101	Transient object exists	objname (eg '2016abc'), prefix (eg 'AT'), type, RA, DEC, separation
102	Related file uploaded	
103	Submitted	
110	No results found	
120	New object type set	new_object_type
121	Object name prefix has changed	new_object_name
122	Object redshift was set	new_redshift
123	Object redshift changed	new_redshift