
ALeRCE API Documentation

Release 0.1

ALeRCE Team

Sep 14, 2020

CONTENTS:

1	ALeRCE ZTF DB	3
1.1	Query the DB	3
1.2	Query an object	5
2	ALeRCE AVRO/Stamps	7
2.1	Request AVRO information	7
3	catsHTM XMatch	9
3.1	Arguments and units in requests	9
3.2	Cone search	9
3.3	Cross-match	9
3.4	Available catalogs:	10
4	Indices and tables	13

[ALeRCE](#) is a Chilean-led alert broker for time-domain astronomy.

ALeRCE implements different services that can be accessed through APIs. These APIs are for public use and most of them do not require authentication.

The APIs are currently being used in [ZTF-Explorer](#) and [SN Hunter](#).

ALERCE ZTF DB

This API gives access to ALerCE Annotated ZTF API, with corrected magnitudes, objects statistics and the object data.

The access point to the API is [ztf.alerce.online], querying the API is done with **http POST requests** with parameters in **JSON** format.

1.1 Query the DB

1.1.1 POST ztf.alerce.online/query

This will query the DB objects and get the stats for the matching objects.

The current fields to query the db are the following:

```
{
  total : number (if not set the total is counted and the query is slower) ,
  records_per_pages : number (default 20) ,
  page : number (default 1) ,
  sortBy : :py:class:`str` columnName (default nobs) ,
  query_parameters : {
    filters : {
      //ZTF object id
      oid : ZTFXXXXXX ,
      //Number of detections
      nobs : {
        min : int ,
        max : int
      },
      //Late Classifier (Random Forest)
      classrf : string or int ,
      pclassrf : float [0-1] ,
      //Early Classifier (Stamp Classifier)
      classeealy : list, string or int ,
      pclasseealy : float [0-1] ,
    }
    //Coordinate based search (RA,DEC) and Search Radius.
    coordinates : {
      ra : float degrees ,
      dec : float degrees ,
      sr : float degrese
    },
    dates : {
```

(continues on next page)

(continued from previous page)

```
//First detection (Discovery date)
firstmjd : {
  min : float mjd ,
  max : float mjd
}
}
```

The response contains the following:

```
{
  "total": int,
  "num_pages": int,
  "page": int,
  "result": {
    <ObjectId>: <ObjectStats>
  }
}
```

Where ObjectId is the ZTF id and ObjectStats is a JSON document with the stats.

Example

Get the recent objects

```
curl -X POST ztf.alerce.online/query -d @- << EOF
{
  "query_parameters":{
    "dates":{
      "firstmjd":{
        "min": 58682
      }
    }
  }
}
EOF
```

Get the last 100 objects classified as SNe

```
curl -X POST ztf.alerce.online/query -d @- << EOF
{
  "records_per_pages":100,
  "query_parameters": {
    "filters": {"classearly": 2},
    "dates":{
      "firstmjd":{
        "min":58680
      }
    }
  },
  "sortBy": "pclassearly",
  "total":100
}
EOF
```


1.1.2 POST ztf.alerce.online/get_sql

Get the SQL Query done to the ZTF DB using the same parameters as `/query`

1.1.3 Example

```
curl -X POST ztf.alerce.online/get_sql -d @- << EOF
{
  "query_parameters":{
    "dates":{
      "firstmjd":{
        "min": 58682
      }
    }
  }
}
EOF
```

1.2 Query an object

To get an specific ZTF Object information.

1.2.1 POST ztf.alerce.online/get_detection

Get all the detections for an object.

```
curl -X POST ztf.alerce.online/get_detections -d @- << EOF
{
  "oid":"ZTF18abbvavt"
}
EOF
```

1.2.2 POST ztf.alerce.online/get_non_detection

Get non detections of an object.

```
curl -X POST ztf.alerce.online/get_non_detections -d @- << EOF
{
  "oid":"ZTF18abbvavt"
}
EOF
```

1.2.3 POST ztf.alerce.online/get_stats

Get stats for an object.

```
curl -X POST ztf.alerce.online/get_stats -d @- << EOF
{
  "oid":"ZTF18abbvavt"
}
EOF
```

1.2.4 POST ztf.alerce.online/get_probabilities

Get probabilities of the models for an object.

```
curl -X POST ztf.alerce.online/get_probabilities -d @- << EOF
{
  "oid":"ZTF18abbvavt"
}
EOF
```

1.2.5 POST ztf.alerce.online/get_features

Get features computed for an object (list of features pending).

```
curl -X POST ztf.alerce.online/get_features -d @- << EOF
{
  "oid":"ZTF18abbvavt"
}
EOF
```

ALERCE AVRO/STAMPS

API to get an specified avro/stamp from alert files of an ZTF Object. This API uses **GET requests** parameters.

2.1 Request AVRO information

2.1.1 GET avro.alerce.online/get_avro

Get an specified avro

```
curl http://avro.alerce.online/get_stamp?oid=ZTF19abguqsi&candid=928474644915015004
```

Parameters:

- oid: ZTF Object ID.
- candid: ZTF Alert ID.

2.1.2 GET avro.alerce.online/get_avro_info

Get an specified avro information as a JSON file (This comes without the stamps)

```
curl http://avro.alerce.online/get_avro_info?oid=ZTF19abguqsi&  
↪candid=928474644915015004
```

Parameters:

- oid: ZTF Object ID.
- candid: ZTF Alert ID.

2.1.3 GET avro.alerce.online/get_stamp

Get an specific stamp

```
curl http://avro.alerce.online/get_stamp?oid=ZTF19abguqsi&candid=928474644915015004&  
↪format=png&type=science
```

Parameters:

- oid: ZTF Object ID.
- candid: ZTF Alert ID.

- format: ["png","fits"]
- type: ["science","template","difference"]

CATSHTM XMATCH

Cats Service provides cone search and cross-match on different *catalogs*. This service is based on [catsHTM](#).

3.1 Arguments and units in requests

The parameters and units in the catsHTM API are the following:

Parameter	unit
catalog	string
ra	degrees
dec	degrees
radius (optional)	arcsec

3.2 Cone search

Cone search on a specific catalog or on all of them.

```
GET catshtm.alerce.online/conesearch[_all]
```

3.2.1 Cone search on a catalog

```
curl "catshtm.alerce.online/conesearch?catalog=GAIADR1&ra=357.73373&dec=14.20514&  
↪radius=100"
```

3.2.2 Cone search on all catalogs

```
curl "catshtm.alerce.online/conesearch_all?ra=357.73373&dec=14.20514&radius=10"
```

3.3 Cross-match

Get the closest object, given RA and Dec, for a catalog or all of them.

```
GET catshtm.alerce.online/crossmatch[_all]
```

3.3.1 Cross-match on a catalog

Since the radius argument is optional, there are two ways to perform cross-match on a catalog.

1. Providing a value for the radius:

```
curl "catshtm.alerce.online/crossmatch?catalog=SDSSDR10&ra=357.73373&dec=14.20514&radius=10"
```

1. Not providing one:

```
curl "catshtm.alerce.online/crossmatch?catalog=SDSSDR10&ra=357.73373&dec=14.20514"
```

If a radius is provided, then that value is used. If not, the default value for that catalog is used. See default values in *Available catalogs*

3.3.2 Cross-match on all catalogs

For cross-matching on all catalogs, the same rule of providing a radius or not applies. Therefore, there are two ways to send the request.

1. With radius:

```
curl "catshtm.alerce.online/crossmatch_all?ra=357.73373&dec=14.20514&radius=100"
```

1. Without radius:

```
curl "catshtm.alerce.online/crossmatch_all?ra=357.73373&dec=14.20514"
```

3.4 Available catalogs:

For catalogs that have a value listed, this value is the radius used for cone search and cross-match. Otherwise, the default radius of 50 arcsec is used.

- AAVSO_VSX
- AKARI
- APASS (2 arcsec)
- CRTS_per_var
- Cosmos
- DECaLS (0.1 arcsec)
- FIRST
- GAIADR1 (0.1 arcsec)
- GAIADR2 (0.1 arcsec)
- GALEX
- HSCv2

- IPHAS
- IRACgc
- NEDz
- NVSS (10.8 arcsec)
- PTFpc
- ROSATfsc
- SAGE
- SDSSDR10
- SDSSoffset (0.1 arcsec)
- SWIREz
- SkyMapper (0.4 arcsec)
- SpecSDSS
- TMASS (2MASS)
- TMASSxsc (2MASSxsc)
- UCAC4
- UKIDSS
- VISTAviking
- VSTatlas
- VSTkids
- WISE
- XMM (8 arcsec)
- unWISE

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`