

[Home](#)[Most tracked](#)[Just launched](#)[Satellites on orbit](#)[Alerting tools](#)[More stuff](#)[Sign in](#)

## N2YO.COM REST API v1

The purpose of the API is to provide data for software/web developers to build satellite tracking or prediction applications.

The REST API v1 is free but it is transaction limited. The current limit is 1000 transactions/hour. If you need more traffic, [contact us](#). Users are requested not to abuse the system, and not use multiple license keys to get around the hourly transaction limit. Traffic is monitored and logged. When the API is used in improper ways, user access will be limited or blocked.

For each transaction, the caller must be identified and authorized by a license key (API key). To generate a unique license key, users are requested first to [register](#) an account at n2yo.com. After login, please visit the profile page and scroll down to access the button that generates the API key. The new API key will be accessible in the profile and cannot be changed. If you need a new key, please [contact us](#) and provide an explanation. The keys issued for the SOAP Web Services API are not valid for REST API.

As a general concept, satellites are identified by their NORAD catalog number which is an integer ranging from 1 to 43235 and counting. The numbers are assigned by United States Space Command (USSPACECOM) to all Earth orbiting satellites in order of identification. If you are looking to find the id of your favorite satellite, browse [categories](#) or [search](#) the n2yo.com database until you arrive at the satellite details page, and extract the NORAD ID for your application. For instance, Space Station (ISS) is NORAD ID 25544.

For all request, method used is GET.

Base URL: <https://api.n2yo.com/rest/v1/satellite/>

API key: The API key (apiKey) must be appended at the end of the URL as &apiKey={your API key}

The following functions are currently available:

- [Get TLE](#)
- [Get satellite positions](#)
- [Get visual passes](#)
- [Get radio passes](#)
- [What's up?](#)

### Get TLE

Retrieve the Two Line Elements (TLE) for a satellite identified by NORAD id.

Request: /tle/{id}

Parameter	Type	Required	Comments
id	integer	Yes	NORAD id

Response:

Parameter	Type	Comments
satid	integer	NORAD id used in input
satname	string	Satellite name
transactionscount	integer	Count of transactions performed with this API key in last 60 minutes
tle	string	TLE on single line string. Split the line in two by \r\n to get original two lines

Example

Request: <https://api.n2yo.com/rest/v1/satellite/tle/25544&apiKey=589P8Q-SDRYX8-L842ZD-5Z9>

JSON Response:

```
{
  "info": {
    "satid": 25544,
    "satname": "SPACE STATION",
    "transactionscount": 4
  },
  "tle": "1 25544U 98067A   18077.09047010   .00001878   00000-0   35621-4 0   9999\r\n2 25544   51.6412 112.8495 0001928 208.4187 178.9720 15.54106440104358"
```

### Get satellite positions

Retrieve the future positions of any satellite as groundtrack (latitude, longitude) to display orbits on maps. Also return the satellite's azimuth and elevation with respect to the observer location. Each element in the response array is one second of calculation. First element is calculated for current UTC time.

Request: /positions/{id}/{observer\_lat}/{observer\_lng}/{observer\_alt}/{seconds}

Parameter	Type	Required	Comments
id	integer	Yes	NORAD id
observer_lat	float	Yes	Observer's latitude (decimal degrees format)
observer_lng	float	Yes	Observer's longitude (decimal degrees format)
observer_alt	float	Yes	Observer's altitude above sea level in meters
seconds	integer	Yes	Number of future positions to return. Each second is a position. Limit 300 seconds

Response:

Parameter	Type	Comments
satid	integer	NORAD id used in input
satname	string	Satellite name
transactionscount	integer	Count of transactions performed with this API key in last 60 minutes
satlatitude	float	Satellite footprint latitude (decimal degrees format)
satlongitude	float	Satellite footprint longitude (decimal degrees format)
azimuth	float	Satellite azimuth with respect to observer's location (degrees)
elevation	float	Satellite elevation with respect to observer's location (degrees)
ra	float	Satellite right ascension (degrees)
dec	float	Satellite declination (degrees)
timestamp	integer	Unix time for this position (seconds). You should convert this UTC value to observer's time zone

Example - retrieve Space Station (25544) positions for next 2 seconds. Observer is located at lat: 41.702, lng: -76.014, alt: 0

Request: <https://api.n2yo.com/rest/v1/satellite/positions/25544/41.702/-76.014/0/2&apiKey=589P8Q-SDRYX8-L842ZD-5Z9>  
JSON Response:

```
{
  "info": {
    "satname": "SPACE STATION",
    "satid": 25544,
    "transactionscount": 5
  },
  "positions": [
    {
      "satlatitude": -39.90318514,
      "satlongitude": 158.28897924,
      "sataltitude": 417.85,
      "azimuth": 254.31,
      "elevation": -69.09,
      "ra": 44.77078138,
      "dec": -43.99279118,
      "timestamp": 1521354418
    },
    {
      "satlatitude": -39.86493451,
      "satlongitude": 158.35261287,
      "sataltitude": 417.84,
      "azimuth": 254.33,
      "elevation": -69.06,
      "ra": 44.81676119,
      "dec": -43.98086419,
      "timestamp": 1521354419
    }
  ]
}
```

### Get visual passes

Get predicted visual passes for any satellite relative to a location on Earth. A "visual pass" is a pass that should be optically visible on the entire (or partial) duration of crossing the sky. For that to happen, the satellite must be above the horizon, illumintaed by Sun (not in Earth shadow), and the sky dark enough to allow visual satellite observation.

Request: [/visualpasses/{id}/{observer\\_lat}/{observer\\_lng}/{observer\\_alt}/{days}/{min\\_visibility}](#)

Parameter	Type	Required	Comments
id	integer	Yes	NORAD id
observer_lat	float	Yes	Observer's latitude (decimal degrees format)
observer_lng	float	Yes	Observer's longitude (decimal degrees format)
observer_alt	float	Yes	Observer's altitude above sea level in meters
days	integer	Yes	Number of days of prediction (max 10)
min_visibility	integer	Yes	Minimum number of seconds the satellite should be considered optically visible during the pass to be returned as result

Response:

Parameter	Type	Comments
satid	integer	NORAD id used in input
satname	string	Satellite name
transactionscount	integer	Count of transactions performed with this API key in last 60 minutes
passescount	integer	Count of passes returned
startAz	float	Satellite azimuth for the start of this pass (relative to the observer, in degrees)
startAzCompass	string	Satellite azimuth for the start of this pass (relative to the observer). Possible values: N, NE, E, SE, S, SW, W, NW
startEl	float	Satellite elevation for the start of this pass (relative to the observer, in degrees)
startUTC	integer	Unix time for the start of this pass. You should convert this UTC value to observer's time zone
maxAz	float	Satellite azimuth for the max elevation of this pass (relative to the observer, in degrees)
maxAzCompass	string	Satellite azimuth for the max elevation of this pass (relative to the observer). Possible values: N, NE, E, SE, S, SW, W, NW
maxEl	float	Satellite max elevation for this pass (relative to the observer, in degrees)
maxUTC	integer	Unix time for the max elevation of this pass. You should convert this UTC value to observer's time zone
endAz	float	Satellite azimuth for the end of this pass (relative to the observer, in degrees)
endAzCompass	string	Satellite azimuth for the end of this pass (relative to the observer). Possible values: N, NE, E, SE, S, SW, W, NW
endEl	float	Satellite elevation for the end of this pass (relative to the observer, in degrees)
endUTC	integer	Unix time for the end of this pass. You should convert this UTC value to observer's time zone
mag	float	Max visual magnitude of the pass, same scale as star brightness. If magnitude cannot be determined, the value is 100000
duration	integer	Total visible duration of this pass (in seconds)

Example - retrieve Space Station (25544) passes optically visible at least 300 seconds for next 2 days. Observer is located at lat: 41.702, lng: -76.014, alt: 0

Request: <https://api.n2yo.com/rest/v1/satellite/visualpasses/25544/41.702/-76.014/0/2/300&apiKey=589P8Q-SDRYX8-L842ZD-5Z9>

JSON Response:

```
{
  "info": {
    "satid": 25544,
    "satname": "SPACE STATION",
    "transactionscount": 4,
    "passescount": 3
  },
  "passes": [
    {
      "startAz": 307.21,
      "startAzCompass": "NW",
      "startEl": 13.08,
      "startUTC": 1521368025,
      "maxAz": 225.45,
      "maxAzCompass": "SW",
      "maxEl": 78.27,
      "maxUTC": 1521368345,
      "endAz": 132.82,
      "endAzCompass": "SE",
      "endEl": 0,
      "endUTC": 1521368660,
      "mag": -2.4,
      "duration": 485
    },
    {
      "startAz": 311.56,
      "startAzCompass": "NW",
      "startEl": 50.94,
      "startUTC": 1521451295,
      "maxAz": 37.91,
      "maxAzCompass": "NE",
      "maxEl": 52.21,
      "maxUTC": 1521451615,
      "endAz": 118.61,
      "endAzCompass": "ESE",
      "endEl": 0,
      "endUTC": 1521451925,
      "mag": -2,
      "duration": 325
    },
    {
      "startAz": 291.06,
      "startAzCompass": "WNW",
      "startEl": 3.47,
      "startUTC": 1521457105,
      "maxAz": 231.58,
      "maxAzCompass": "SW",
      "maxEl": 14.75,
      "maxUTC": 1521457380,
      "endAz": 170.63,
      "endAzCompass": "S",
      "endEl": 0,
      "endUTC": 1521457650,
      "mag": -0.1,
      "duration": 325
    }
  ]
}
```

```
    "duration": 485
  }
}
```

Get radio passes

The "radio passes" are similar to "visual passes", the only difference being the requirement for the objects to be optically visible for observers. This function is useful mainly for predicting satellite passes to be used for radio communications. The quality of the pass depends essentially on the highest elevation value during the pass, which is one of the input parameters.

Request: /radiopasses/{id}/{observer\_lat}/{observer\_lng}/{observer\_alt}/{days}/{min\_elevation}

Parameter	Type	Required	Comments
id	integer	Yes	NORAD id
observer_lat	float	Yes	Observer's latitude (decimal degrees format)
observer_lng	float	Yes	Observer's longitude (decimal degrees format)
observer_alt	float	Yes	Observer's altitude above sea level in meters
days	integer	Yes	Number of days of prediction (max 10)
min_elevation	integer	Yes	The minimum elevation acceptable for the highest altitude point of the pass (degrees)

Response:

Parameter	Type	Comments
satid	integer	NORAD id used in input
satname	string	Satellite name
transactionscount	integer	Count of transactions performed with this API key in last 60 minutes
passescount	integer	Count of passes returned
startAz	float	Satellite azimuth for the start of this pass (relative to the observer, in degrees)
startAzCompass	string	Satellite azimuth for the start of this pass (relative to the observer). Possible values: N, NE, E, SE, S, SW, W, NW
startUTC	integer	Unix time for the start of this pass. You should convert this UTC value to observer's time zone
maxAz	float	Satellite azimuth for the max elevation of this pass (relative to the observer, in degrees)
maxAzCompass	string	Satellite azimuth for the max elevation of this pass (relative to the observer). Possible values: N, NE, E, SE, S, SW, W, NW
maxEl	float	Satellite max elevation for this pass (relative to the observer, in degrees)
maxUTC	integer	Unix time for the max elevation of this pass. You should convert this UTC value to observer's time zone
endAz	float	Satellite azimuth for the end of this pass (relative to the observer, in degrees)
endAzCompass	string	Satellite azimuth for the end of this pass (relative to the observer). Possible values: N, NE, E, SE, S, SW, W, NW
endUTC	integer	Unix time for the end of this pass. You should convert this UTC value to observer's time zone

Example - retrieve Space Station (25544) radio passes above 40 degrees of elevation for next 2 days. Observer is located at lat: 41.702, lng: -76.014, alt: 0

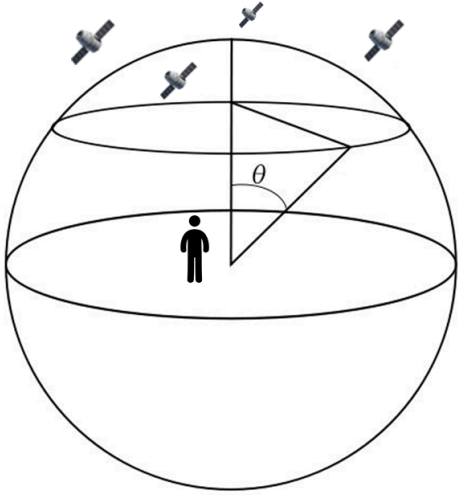
Request: <https://api.n2yo.com/rest/v1/satellite/radiopasses/25544/41.702/-76.014/0/2/40/&apiKey=589P8Q-SDRYX8-L842ZD-5Z9>  
JSON Response:

```
{
  "info": {
    "satid": 25544,
    "satname": "SPACE STATION",
    "transactionscount": 2,
    "passescount": 2
  },
  "passes": [
    {
      "startAz": 311.57,
      "startAzCompass": "NW",
      "startUTC": 1521451295,
      "maxAz": 37.98,
      "maxAzCompass": "NE",
      "maxEl": 52.19,
      "maxUTC": 1521451615,
      "endAz": 118.6,
      "endAzCompass": "ESE",
      "endUTC": 1521451925
    },
    {
      "startAz": 242.34,
      "startAzCompass": "WSW",
      "startUTC": 1521600275,
      "maxAz": 328.03,
      "maxAzCompass": "NW",
      "maxEl": 49.59,
      "maxUTC": 1521600595,
      "endAz": 47.97,
```

```
    "endAzCompass": "NE",  
    "endUTC": 1521600905  
  }  
}
```

What's up?

The "above" function will return all objects within a given search radius above observer's location. The radius ( $\theta$ ), expressed in degrees, is measured relative to the point in the sky directly above an observer (azimuth). This image may offer a better explanation:



The search radius range is 0 to 90 degrees, nearly 0 meaning to show only satellites passing exactly above the observer location, while 90 degrees to return all satellites above the horizon. Since there are many satellites and debris in the sky at any point in time, the result could be filtered by satellite category (integer). The following categories are currently available at n2yo.com:

Catgory	id	Catgory	id
Amateur radio	18	IRNSS	46
Beidou Navigation System	35	ISS	2
Brightest	1	Lemur	49
Celestis	45	Military	30
Chinese Space Station	54	Molniya	14
CubeSats	32	Navy Navigation Satellite System	24
Disaster monitoring	8	NOAA	4
Earth resources	6	O3B Networks	43
Education	29	OneWeb	53
Engineering	28	Orbcomm	16
Experimental	19	Parus	38
Flock	48	Qianfan	55
Galileo	22	QZSS	47
Geodetic	27	Radar Calibration	31
Geostationary	10	Raduga	13
Global Positioning System (GPS) Constellation	50	Russian LEO Navigation	25
Global Positioning System (GPS) Operational	20	Satellite-Based Augmentation System	23
Globalstar	17	Search & rescue	7
Glonass Constellation	51	Space & Earth Science	26
Glonass Operational	21	Starlink	52
GOES	5	Strela	39
Gonets	40	Tracking and Data Relay Satellite System	9
Gorizont	12	Tselina	44
Intelsat	11	Tsikada	42
Iridium	15	Tsiklon	41

TV	34
Weather	3
Westford Needles	37
XM and Sirius	33
Yaogan	36

Please use this function responsibly as there is a lot of CPU needed in order to calculate exact positions for all satellites in the sky. The function will return altitude, latitude and longitude of satellites footprints to be displayed on a map, and some minimal information to identify the object.

Request: /above/{observer\_lat}/{observer\_lng}/{observer\_alt}/{search\_radius}/{category\_id}

Parameter	Type	Required	Comments
observer_lat	float	Yes	Observer's latitude (decimal degrees format)
observer_lng	float	Yes	Observer's longitude (decimal degrees format)
observer_alt	float	Yes	Observer's altitude above sea level in meters
search_radius	integer	Yes	Search radius (0-90)
category_id	integer	Yes	Category id (see table). Use 0 for all categories

Response:

Parameter	Type	Comments
category	string	Category name (ANY if category id requested was 0)
transactionscount	integer	Count of transactions performed with this API key in last 60 minutes
satcount	integer	Count of satellites returned
satid	integer	Satellite NORAD id
intDesignator	string	Satellite international designator
satname	string	Satellite name
launchDate	string	Satellite launch date (YYYY-MM-DD)
satlat	float	Satellite footprint latitude (decimal degrees format)
satlng	float	Satellite footprint longitude (decimal degrees format)
satalt	float	Satellite altitude (km)

Example - retrieve Amateur Radio satellites (categ id 18) located within a search radius of 70 degrees. Observer is located at lat: 41.702, lng: -76.014, alt: 0

Request: <https://api.n2yo.com/rest/v1/satellite/above/41.702/-76.014/0/70/18&apiKey=589P8Q-SDRYX8-L842ZD-5Z9>

JSON Response:

```
{
  "info": {
    "category": "Amateur radio",
    "transactionscount": 17,
    "satcount": 3
  },
  "above": [
    {
      "satid": 20480,
      "satname": "JAS 1B (FUJI 2)",
      "intDesignator": "1990-013C",
      "launchDate": "1990-02-07",
      "satlat": 49.5744,
      "satlng": -96.7081,
      "satalt": 1227.9326
    },
    {
      "satid": 26609,
      "satname": "AMSAT OSCAR 40",
      "intDesignator": "2000-072B",
      "launchDate": "2000-11-16",
      "satlat": 5.5105,
      "satlng": -21.4478,
      "satalt": 49678.6389
    },
    {
      "satid": 40719,
      "satname": "DEORBITSAIL",
      "intDesignator": "2015-032E",
      "launchDate": "2015-07-10",
      "satlat": 43.8106,
      "satlng": -90.3944,
      "satalt": 657.5516
    }
  ]
}
```

[Links](#) [Terms of Use](#) [Privacy Policy](#) [Contact Us](#)  
Copyright © N2YO.com. All rights reserved

 [Bookmark and Share](#)