

images.nasa.gov API Documentation

Release v1.1.1 (2017-04-11)

API Reference

The images.nasa.gov API is organized around [REST](#). Our API has predictable, resource-oriented URLs, and uses HTTP response codes to indicate API errors. We use built-in HTTP features, like HTTP authentication and HTTP verbs, which are understood by off-the-shelf HTTP clients. We support [cross-origin resource sharing](#), allowing you to interact securely with our API from a client-side web application. [JSON](#) is returned by all API responses, including errors.

Each of the endpoints described below also contains example snippets which use the [curl](#) command-line tool, [Unix pipelines](#), and the [python](#) command-line tool to output API responses in an easy-to-read format. We insert superfluous newlines to improve readability in these inline examples, but to run our examples you must remove these newlines.

API Root:

```
https://images-api.nasa.gov
```

API Endpoints:

- [/search](#)
- [/asset/{nasa_id}](#)
- [/metadata/{nasa_id}](#)
- [/captions/{nasa_id}](#)

Errors

images-api.nasa.gov uses conventional HTTP response codes to indicate the success or failure of an API request. In general, codes in the **2xx** range indicate success, codes in the **4xx** range indicate an error that failed given the information provided (e.g., a required parameter was omitted, a search failed, etc.), and codes in the **5xx** range indicate an error with our API servers (these are rare).

Most error responses contain a **reason** attribute, a human-readable message providing more details about the error.

HTTP status code summary

Code	Explanation
200 - OK	Everything worked as expected.
400 - Bad Request	The request was unacceptable, often due to missing a required parameter.
404 - Not Found	The requested resource doesn't exist.
500, 502, 503, 504 - Server Errors	Something went wrong on the API's end. (These are rare.)

Handling errors

Our API returns HTTP error responses for many reasons, such as a failed search query, invalid parameters, a query for a non-existent media asset, and network unavailability. We recommend writing code that gracefully handles all possible HTTP status codes our API returns.

Performing a search

GET /search?q={q}

Parameters:

Name	Type	Description
q (optional)	string	Free text search terms to compare to all indexed metadata.
center (optional)	string	NASA center which published the media.
description (optional)	string	Terms to search for in "Description" fields.
description_508 (optional)	string	Terms to search for in "508 Description" fields.
keywords (optional)	string	Terms to search for in "Keywords" fields. Separate multiple values with commas.
location (optional)	string	Terms to search for in "Location" fields.
media_type (optional)	string	Media types to restrict the search to. Available types: ["image", "audio"]. Separate multiple values with commas.
nasa_id (optional)	string	The media asset's NASA ID.
photographer (optional)	string	The primary photographer's name.
secondary_creator (optional)	string	A secondary photographer/videographer's name.
title (optional)	string	Terms to search for in "Title" fields.
year_start (optional)	string	The start year for results. Format: YYYY.
year_end (optional)	string	The end year for results. Format: YYYY.

Example Request:

At least one parameter is required, but all individual parameters are optional. All parameter values must be URL-encoded. Most HTTP client libraries will take care of this for you. Use `--data-urlencode` to encode values using `curl`:

```
curl -G https://images-api.nasa.gov/search
--data-urlencode "q=apollo 11"
--data-urlencode "description=moon landing"
--data-urlencode "media_type=image" |
python -m json.tool
```

The equivalent pre-encoded request looks more typical:

```
curl "https://images-api.nasa.gov/search
?q=apollo%2011
&description=moon%20landing
&media_type=image" |
python -m json.tool
```

Example Response:

Search results will come in the form of `Collection+JSON`, which contains results and information about how to retrieve more details on each result:

```
{
  "collection": {
    "href": "https://images-api.nasa.gov/search?q=apollo%2011...",
    "items": [
      {
        "data": [
          {
```

```

    "center": "JSC",
    "date_created": "1969-07-21T00:00:00Z",
    "description": "AS11-40-5874 (20 July 1969) ---
Astronaut Edwin E. Aldrin Jr., lunar module pilot
of the first lunar landing mission, poses for a
photograph beside the deployed United States flag
during Apollo 11 extravehicular activity (EVA) on
the lunar surface. The Lunar Module (LM) is on the
left, and the footprints of the astronauts are
clearly visible in the soil of the moon. Astronaut
Neil A. Armstrong, commander, took this picture
with a 70mm Hasselblad lunar surface camera. While
astronauts Armstrong and Aldrin descended in the LM
the \"Eagle\" to explore the Sea of Tranquility
region of the moon, astronaut Michael Collins,
command module pilot, remained with the Command and
Service Modules (CSM) \"Columbia\" in lunar
orbit.",
    "keywords": [
        "APOLLO 11 FLIGHT",
        "MOON",
        "LUNAR SURFACE",
        "LUNAR BASES",
        "LUNAR MODULE",
        "ASTRONAUTS",
        "EXTRAVEHICULAR ACIVITY"
    ],
    "media_type": "image",
    "nasa_id": "as11-40-5874",
    "title": "Apollo 11 Mission image - Astronaut Edwin Aldrin
poses beside th"
}
],
"href": "https://images-assets.nasa.gov/image/as11-40-5874/collection.json",
"links": [
    {
        "href": "https://images-assets.nasa.gov/image/as11-40-5874/as11-40-5874~thumb.jpg",
        "rel": "preview",
        "render": "image"
    }
]
}
...*99 more objects omitted*...
],
"links": [
    {
        "href": "https://images-api.nasa.gov/search?q=apollo+11...&page=2",
        "prompt": "Next",
        "rel": "next"
    }
],
"metadata": {
    "total_hits": 336
},
"version": "1.0"
}
}

```

Retrieving a media asset's manifest

GET /asset/{nasa_id}

Parameters:

Name	Type	Description
nasa_id	string	The media asset's NASA ID.

Example Request:

```
curl https://images-api.nasa.gov/asset/as11-40-5874 |
python -m json.tool
```

Example Response:

Asset manifest results will come in the form of Collection+JSON:

```
{
  "collection": {
    "href": "https://images-api.nasa.gov/asset/as11-40-5874",
    "items": [
      {
        "href": "https://images-assets.nasa.gov/image/as11-40-5874/as11-40-5874~orig.jpg"
      },
      {
        "href": "https://images-assets.nasa.gov/image/as11-40-5874/as11-40-5874~medium.jpg"
      },
      {
        "href": "https://images-assets.nasa.gov/image/as11-40-5874/as11-40-5874~small.jpg"
      },
      {
        "href": "https://images-assets.nasa.gov/image/as11-40-5874/as11-40-5874~thumb.jpg"
      },
      {
        "href": "https://images-assets.nasa.gov/image/as11-40-5874/metadata.json"
      }
    ],
    "version": "1.0"
  }
}
```

Retrieving a media asset's metadata location

GET /metadata/{nasa_id}

Parameters:

Name	Type	Description
nasa_id	string	The media asset's NASA ID.

Example Request:

```
curl https://images-api.nasa.gov/metadata/as11-40-5874 |
python -m json.tool
```

Example Response:

```
{
  "location": "https://images-assets.nasa.gov/image/as11-40-5874/metadata.json"
}
```

Download the JSON file at the location in the response to see the asset's metadata.

Retrieving a video asset's captions location

GET /captions/{nasa_id}

Parameters:

Name	Type	Description
nasa_id	string	The video asset's NASA ID.

Example Request:

```
curl https://images-api.nasa.gov/captions/172_ISS-Slosh |  
python -m json.tool
```

Example Response:

```
{  
  "location": "https://images-assets.nasa.gov/video/172_ISS-Slosh/172_ISS-Slosh.srt"  
}
```

Download the VTT or SRT file at the location in the response to see the video's captions.