GMN Python API

Ricky Bassom

Jan 05, 2024

CONTENTS

CHAPTER

ONE

USAGE

Simple meteor analysis example:

```
from gmn_python_api import data_directory as dd
from gmn_python_api import meteor_trajectory_reader

# Analyse recorded meteor data for the 24th of July 2019
traj_file_content = dd.get_daily_file_content_by_date("2019-07-24")

# Read data as a Pandas DataFrame
traj_df = meteor_trajectory_reader.read_data(traj_file_content)

print(f"{traj_df['Vgeo (km/s)'].max()} km/s was the fastest geostationary velocity")
# Output: 65.38499 km/s was the fastest geostationary velocity

print(f"{traj_df.loc[traj_df['IAU (code)'] == 'PER'].shape[0]} Perseid meteors")
# Output: 3 Perseid meteors

print(f"Station #{traj_df['Num (stat)'].mode().values[0]} recorded the most meteors")
# Output: Station #2 recorded the most meteors
```

The meteor trajectory data model can be loaded offline:

```
from gmn_python_api.meteor_trajectory_schema import get_model_meteor_trajectory_dataframe

traj_df = get_model_meteor_trajectory_dataframe()
```

See the *Data Directory* section for details about how to access meteor trajectory data using the GMN Data Directory.

See the *REST API* section for details about how to access meteor trajectory data using the GMN REST API.

See the *Data Schemas* section for meteor trajectory DataFrame features.

See the *API Reference* section for function and variable definitions.

CHAPTER

TWO

GMN_PYTHON_API

GMN Python API.

2.1 Submodules

2.1.1 gmn_python_api.data_directory

This module contains functions to read meteor trajectory files from the GMN Data Directory.

Module Contents

Functions

get_all_daily_file_urls()	Get all daily meteor trajectory file urls from the GMN Data Directory.
get_all_monthly_file_urls()	Get all monthly meteor trajectory file urls from the GMN Data Directory.
get_all_file_url()	Get the URL of the meteor trajectory file containing all data.
get_daily_file_url_by_date(date_str[, current_date])	Get the URL of the daily meteor trajectory file for a given date.
get_monthly_file_url_by_month(date_str)	Get the URL of the monthly meteor trajectory file for a given month.
get_file_content_from_url(file_url)	Get the content of a meteor trajectory file from a given URL.
get_daily_file_content_by_date(date_str[, current_date])	Get the content of the daily meteo trajectory file for a given date.
get_monthly_file_content_by_date(date_str)	Get the content of the monthly meteor trajectory file for a given date.
get_all_file_content()	Get the content of the meteor trajectory file containing all data.
_get_url_paths(url[, ext])	Get all paths from a directory listing URL.

Attributes

BASE_URL	The base URL for meteor trajectory files in the GMN Data Directory.
DATA_START_DATE	The date of the earliest meteor trajectory file in the GMN Data Directory.
DAILY_DIRECTORY	The name of the directory containing daily meteor trajectory files in the base
MONTHLY_DIRECTORY	The name of the directory containing monthly meteor trajectory files in the base
SUMMARY_FILE_EXTENSION	The extension of the meteor trajectory files in the GMN Data Directory
SUMMARY_TODAY_FILENAME	The filename of the most recent meteor trajectory file.
SUMMARY_YESTERDAY_FILENAME	The filename of the meteo trajectory file from yesterday.
SUMMARY_ALL_FILENAME	The filename of the meteor trajectory file containing all data.
DAILY_DATE_INPUT_FORMAT	The daily string date format that should be passed in as a parameter to the
MONTHLY_DATE_INPUT_FORMAT	The monthly string date format that should be passed in as a parameter to the

gmn_python_api.data_directory.**BASE_URL: str = 'https://globalmeteornetwork.org/data/traj_summary_data/'**

The base URL for meteor trajectory files in the GMN Data Directory.

gmn_python_api.data_directory.**DATA_START_DATE: datetime.date**

The date of the earliest meteor trajectory file in the GMN Data Directory.

gmn_python_api.data_directory.**DAILY_DIRECTORY: str = 'daily/'**

The name of the directory containing daily meteor trajectory files in the base URL.

gmn_python_api.data_directory.**MONTHLY_DIRECTORY: str = 'monthly/'**

The name of the directory containing monthly meteor trajectory files in the base URL.

gmn_python_api.data_directory.**SUMMARY_FILE_EXTENSION: str = 'txt'**

The extension of the meteor trajectory files in the GMN Data Directory

gmn_python_api.data_directory.**SUMMARY_TODAY_FILENAME: str = 'traj_summary_latest_daily.txt'**

The filename of the most recent meteor trajectory file.

gmn_python_api.data_directory.**SUMMARY_YESTERDAY_FILENAME: str = 'traj_summary_yesterday.txt'**

The filename of the meteo trajectory file from yesterday.

gmn_python_api.data_directory.**SUMMARY_ALL_FILENAME: str = 'traj_summary_all.txt'**

The filename of the meteor trajectory file containing all data.

gmn_python_api.data_directory.**DAILY_DATE_INPUT_FORMAT: str = '%Y-%m-%d'**

The daily string date format that should be passed in as a parameter to the functions in this module.

gmn_python_api.data_directory.**MONTHLY_DATE_INPUT_FORMAT: str = '%Y-%m'**

The monthly string date format that should be passed in as a parameter to the functions in this module.

gmn_python_api.data_directory.**get_all_daily_file_urls**()

Get all daily meteor trajectory file urls from the GMN Data Directory.

Returns
A list of all daily file urls.

Raises
requests.HTTPError if the data directory url doesn't return a 200 response.

Return type
List[str]

gmn_python_api.data_directory.**get_all_monthly_file_urls**()

Get all monthly meteor trajectory file urls from the GMN Data Directory.

Returns
A list of all monthly file urls.

Raises
requests.HTTPError if the data directory url doesn't return a 200 response.

Return type
List[str]

gmn_python_api.data_directory.**get_all_file_url**()

Get the URL of the meteor trajectory file containing all data.

Returns
The URL of the file containing all data.

Return type
str

gmn_python_api.data_directory.**get_daily_file_url_by_date**(*date_str*, *current_date=None*)

Get the URL of the daily meteor trajectory file for a given date.

Parameters

- **date_str** (*str*) – The date of the daily file to get in the format YYYY-MM-DD.
- **current_date** (*Optional[get_daily_file_url_by_date.date]*) – The current date. Defaults to datetime.now().

Returns
The URL of the daily file.

Raises
FileNotFoundError if the daily file cannot be found. Or requests.HTTPError is raised if the file url doesn't return a 200 response.

Return type
str

gmn_python_api.data_directory.**get_monthly_file_url_by_month**(*date_str*)

Get the URL of the monthly meteor trajectory file for a given month.

Parameters
date_str (*str*) – The date of the monthly file to get in the format YYYY-MM.

Returns
The URL of the monthly file.

Raises
FileNotFoundError if the monthly file cannot be found. Or requests.HTTPError is raised if the file url doesn't return a 200 response.

Return type
str

gmn_python_api.data_directory.**get_file_content_from_url**(*file_url*)

Get the content of a meteor trajectory file from a given URL.

Parameters

- **url** – The URL of the meteor trajectory file.
- **file_url** (*str*) –

Returns
The content of the file.

Raises
requests.HTTPError If the file url doesn't return a 200 response.

Return type
str

gmn_python_api.data_directory.**get_daily_file_content_by_date**(*date_str*, *current_date=None*)

Get the content of the daily meteo trajectory file for a given date.

Parameters

- **date_str** (*str*) – The date of the daily file to get in the format YYYY-MM-DD.
- **current_date** (*Optional[datetime.date]*) – The current date. Defaults to datetime.now().

Returns
The content of the daily file.

Raises
requests.HTTPError if the data directory url doesn't return a 200 response.

Return type
str

gmn_python_api.data_directory.**get_monthly_file_content_by_date**(*date_str*)

Get the content of the monthly meteor trajectory file for a given date.

Parameters
date_str (*str*) – The date to get the monthly file for in the format YYYY-MM.

Returns
The content of the monthly file.

Raises
requests.HTTPError if the data directory url doesn't return a 200 response.

Return type
str

gmn_python_api.data_directory.**get_all_file_content**()

Get the content of the meteor trajectory file containing all data.

Returns
The content of the file containing all data.

Raises
requests.HTTPError if the data directory url doesn't return a 200 response.

Return type
str

gmn_python_api.data_directory.**_get_url_paths**(*url*, *ext=''*)

Get all paths from a directory listing URL.

Parameters

- **url** (*str*) – The URL to get the paths from.
- **ext** (*str*) – The extension to filter by.

Returns
A list of all paths.

Raises
requests.HTTPError if the URL doesn't return a 200 response.

Return type
List[str]

2.1.2 gmn_python_api.gmn_rest_api

This module contains functions to read data from the GMN REST API. The REST API uses the Datasette API endpoint. More info: https://gmn-python-api.readthedocs.io/en/latest/rest_api.html

Module Contents

Functions

get_meteor_summary_data_all([where, having, order_by, ...])	Get all meteor summary data from the Meteor Summary GMN REST API endpoint.
get_meteor_summary_data_iter([where, having, order_by])	An iterator for fetching meteor summary data from the Meteor Summary GMN REST API
get_meteor_summary_data([where, having, order_by])	Get meteor summary data from the Meteor Summary GMN REST API endpoint starting from
get_data(sql)	Get data from the General GMN REST API endpoint using a custom SQL query.
get_data_from_url(query_url)	Get data from a specified GMN REST API endpoint URL.
_http_get_response(url)	Perform an HTTP GET request and return the response.

Attributes

GMN_REST_API_DOMAIN
QUERY_URL
METEOR_SUMMARY_QUERY_URL

gmn_python_api.gmn_rest_api.**GMN_REST_API_DOMAIN** = **'https://explore.globalmeteornetwork.org'**

gmn_python_api.gmn_rest_api.**QUERY_URL**

gmn_python_api.gmn_rest_api.**METEOR_SUMMARY_QUERY_URL**

exception gmn_python_api.gmn_rest_api.**LastModifiedError**

Bases: Exception

Raised when the data has modified since the last request.

gmn_python_api.gmn_rest_api.**get_meteor_summary_data_all**(*where=None*, *having=None*, *order_by=None*, *last_modified_error_retries=3*)

Get all meteor summary data from the Meteor Summary GMN REST API endpoint.

Parameters

- **where** (*Optional[str]*) – Optional parameter to filter data via a SQL WHERE clause e.g. meteor.unique_trajectory_identifier = '20190103131723_6dnE3'.
- **having** (*Optional[str]*) – Optional parameter to filter data via a SQL HAVING clause e.g. participating_stations LIKE '%US0003%'.
- **order_by** (*Optional[str]*) – Optional parameter to specify the order of results via a SQL ORDER BY clause e.g. meteor.unique_trajectory_identifier DESC.
- **last_modified_error_retries** (*int*) – Number of times to retry if the data has modified since the last request.

Raises

LastModifiedError: If the data has modified since the last request too many times.

Raises

requests.exceptions.HTTPError: If the HTTP response status code is not 200 OK.

Returns

A list of json data.

Return type

List[Dict[str, Any]]

gmn_python_api.gmn_rest_api.**get_meteor_summary_data_iter**(*where=None*, *having=None*, *order_by=None*)

An iterator for fetching meteor summary data from the Meteor Summary GMN REST API endpoint in pages. This is useful for processing large amounts of data. The data is returned in pages of 1000 rows.

Parameters

- **where** (*Optional[str]*) – Optional parameter to filter data via a SQL WHERE clause e.g. meteor.unique_trajectory_identifier = '20190103131723_6dnE3'.
- **having** (*Optional[str]*) – Optional parameter to filter data via a SQL HAVING clause e.g. participating_stations LIKE '%US0003%'.
- **order_by** (*Optional[str]*) – Optional parameter to specify the order of results via a SQL ORDER BY clause e.g. meteor.unique_trajectory_identifier DESC.

Raises
requests.exceptions.HTTPError: If the HTTP response status code is not 200 OK.

Raises
LastModifiedError: If the data has modified since the last request.

Raises
requests.exceptions.HTTPError: If the HTTP response status code is not 200 OK.

Returns
An iterable of json data.

Return type
Iterable[List[Dict[str, Any]]]

gmn_python_api.gmn_rest_api.**get_meteor_summary_data**(*where=None*, *having=None*, *order_by=None*)

Get meteor summary data from the Meteor Summary GMN REST API endpoint starting from the first page.

Parameters

- **where** (*Optional[str]*) – Optional parameter to filter data via a SQL WHERE clause e.g. meteor.unique_trajectory_identifier = '20190103131723_6dnE3'.
- **having** (*Optional[str]*) – Optional parameter to filter data via a SQL HAVING clause e.g. participating_stations LIKE '%US0003%'.
- **order_by** (*Optional[str]*) – Optional parameter to specify the order of results via a SQL ORDER BY clause e.g. meteor.unique_trajectory_identifier DESC.

Raises
requests.exceptions.HTTPError: If the HTTP response status code is not 200 OK.

Returns
Tuple of json data, next URL for pagination, and the last modified date of the GMN data store. If iterating through pages, last_modified should be checked against the last_modified of the previous page. If they are different, then the data has modified since the last request, and the pagination is invalid.

Return type
Tuple[List[Dict[str, Any]], Optional[str], Optional[str]]

gmn_python_api.gmn_rest_api.**get_data**(*sql*)

Get data from the General GMN REST API endpoint using a custom SQL query.

Parameters
sql (*str*) – SQL query to execute (read-only).

Raises
requests.exceptions.HTTPError: If the HTTP response status code is not 200 OK.

Returns
Tuple containing a list of dictionaries containing meteor trajectory data and the last modified date of the GMN data store. If iterating through pages, last_modified should be checked against the last_modified of the previous page. If they are different, then the data has modified since the last request, and the pagination is invalid.

Return type
Tuple[List[Dict[str, Any]], Optional[str]]

gmn_python_api.gmn_rest_api.**get_data_from_url**(*query_url*)

Get data from a specified GMN REST API endpoint URL.

Parameters
query_url (*str*) – URL for querying data from the GMN REST API.

Raises
requests.exceptions.HTTPError: If the HTTP response status code is not 200 OK.

Returns
Tuple of json data, next URL for pagination, and the last modified date of the GMN data store. If iterating through pages, last_modified should be checked against the last_modified of the previous page. If they are different, then the data has modified since the last request, and the pagination is invalid.

Return type
Tuple[List[Dict[str, Any]], Optional[str], Optional[str]]

gmn_python_api.gmn_rest_api.**_http_get_response**(*url*)

Perform an HTTP GET request and return the response.

Parameters
url (*str*) – URL for the HTTP GET request.

Raises
requests.exceptions.HTTPError: If the HTTP response status code is not 200 OK.

Returns
Tuple containing the response text, the next URL for pagination, and the last modified date of the GMN data store.

Return type
Tuple[str, Optional[str], Optional[str]]

2.1.3 gmn_python_api.iau_showers

The module contains functions for retrieving IAU meteor shower information.

Module Contents

Functions

get_iau_showers()	Gets the official list of IAU shower numbers, codes and names.

Attributes

IAU_SHOWERS_LIST_URL	The url that contains the list of IAU shower information.

gmn_python_api.iau_showers.**IAU_SHOWERS_LIST_URL = 'https://www.ta3.sk/IAUC22DB/MDC2007/Etc/streamfulldata.txt'**

The url that contains the list of IAU shower information.

gmn_python_api.iau_showers.**get_iau_showers**()

Gets the official list of IAU shower numbers, codes and names.

Returns
A dictionary, where the key is the shower number, of dictionaries containing the IAU shower information.

Raises
requests.HTTPError if the source server doesn't return a 200 response.

Return type
Dict[str, Dict[str, str]]

2.1.4 gmn_python_api.meteor_trajectory_reader

This module contains functions to load meteor trajectory data into Pandas DataFrames.

Module Contents

Functions

read_data(data[, input_camel_case, output_camel_case])	Reads meteor trajectory data either as a CSV string or a list of dicts into a Pandas
_convert_camel_case_to_verbose_column_names(	Converts the column names in a DataFrame containing meteor trajectory data to verbose
_set_camel_case_column_names(dataframe)	Sets the column names in a DataFrame containing meteor trajectory data to camel case
_set_data_types(dataframe)	Sets the data types and index column in a DataFrame containing meteor trajectory

Attributes

DATETIME_FORMAT

gmn_python_api.meteor_trajectory_reader.**DATETIME_FORMAT = '%Y-%m-%d %H:%M:%S.%f'**

gmn_python_api.meteor_trajectory_reader.**read_data**(*data*, *input_camel_case=False*, *output_camel_case=False*)

Reads meteor trajectory data either as a CSV string or a list of dicts into a Pandas
DataFrame. Columns available in the DataFrame can be found here: https://gmn-python-api.readthedocs.io/en/latest/data_schemas.html

Parameters

- **data** (*Union[str, List[Dict[str, Any]]]*) – The meteor trajectory data. Either a CSV string from the GMN data directory or a JSON from the GMN REST API.
- **input_camel_case** (*Optional[bool]*) – If True, the input data is assumed to have camel case column names e.g. m_deg
- **output_camel_case** (*Optional[bool]*) – If True, DataFrame column names will be camel cased e.g. m_deg

Returns
Pandas DataFrame of the meteor trajectory data.

Return type
pandas.DataFrame

gmn_python_api.meteor_trajectory_reader.**_convert_camel_case_to_verbose_column_names**(*dataframe*)

Converts the column names in a DataFrame containing meteor trajectory data to verbose
e.g. beginning_utc_time to Beginning (UTC Time).

Parameters
dataframe (*pandas.DataFrame*) – The meteor trajectory dataframe to convert the column names for.

Returns
The meteor trajectory dataframe with verbose column names.

Return type
pandas.DataFrame

gmn_python_api.meteor_trajectory_reader.**_set_camel_case_column_names**(*dataframe*)

Sets the column names in a DataFrame containing meteor trajectory data to camel case
e.g. m_deg.

Parameters
dataframe (*pandas.DataFrame*) – The meteor trajectory dataframe to set the column names for.

Returns
None.

Return type
None

gmn_python_api.meteor_trajectory_reader.**_set_data_types**(*dataframe*)

Sets the data types and index column in a DataFrame containing meteor trajectory
data. The input dataframe must be in verbose column name format e.g. "Beginning (UTC Time)".

Parameters
dataframe (*pandas.DataFrame*) – The meteor trajectory dataframe to set the data types for.

Returns
None.

Return type
None

2.1.5 gmn_python_api.meteor_trajectory_schema

This module contains functions for handling the current meteor trajectory data schema.

Module Contents

Functions

get_column_names([output_camel_case])	Get the column names of the current supported meteor trajectory schema.
get_model_meteor_trajectory_dataframe([output_	Get the current supported model meteor trajectory file as a DataFrame.
get_verbose_camel_case_column_name_bidict()	Get a bidirectional dictionary that maps the verbose and camel case column names.

Attributes

SCHEMA_VERSION	The supported meteor trajectory data format version.
_MODEL_METEOR_TRAJECTORY_FILE_PATH	Model meteor trajectory file, full size.
_MODEL_METEOR_TRAJECTORY_FILE_ONE_ROW_PATH	Model meteor trajectory file, just one data row.

gmn_python_api.meteor_trajectory_schema.**SCHEMA_VERSION = '1.0'**
The supported meteor trajectory data format version.

gmn_python_api.meteor_trajectory_schema.**_MODEL_METEOR_TRAJECTORY_FILE_PATH**
Model meteor trajectory file, full size.

gmn_python_api.meteor_trajectory_schema.**_MODEL_METEOR_TRAJECTORY_FILE_ONE_ROW_PATH**
Model meteor trajectory file, just one data row.

gmn_python_api.meteor_trajectory_schema.**get_column_names**(*output_camel_case=False*)
Get the column names of the current supported meteor trajectory schema.

Parameters
output_camel_case (*bool*) – Whether to return the column names in camel case or verbose

Returns
The column names of the current supported meteor trajectory model.

Return type
List[str]

gmn_python_api.meteor_trajectory_schema.**get_model_meteor_trajectory_dataframe**(*output_camel_case=False*)

Get the current supported model meteor trajectory file as a DataFrame.

Parameters
output_camel_case (*bool*) – Whether to return the column names in camel case or verbose

Returns
The model meteor trajectory file as a DataFrame.

Return type
pandas.DataFrame

gmn_python_api.meteor_trajectory_schema.**get_verbose_camel_case_column_name_bidict**()

Get a bidirectional dictionary that maps the verbose and camel case column names.

Returns
A bidirectional dictionary that maps the verbose and camel case column names.

Return type
Dict[str, str]

CHAPTER

THREE

DATA DIRECTORY

The GMN provides a Data Directory of meteor trajectory CSV data. The `gmn-python-api` library allows you to read from the directory (see *data_directory API Reference section* for function and variable details).

3.1 Example 1

```
from gmn_python_api import data_directory as dd
from gmn_python_api import meteor_trajectory_reader

# Get meteor data from the 2019-07-24
traj_file_content = dd.get_daily_file_content_by_date("2019-07-24")
traj_df = meteor_trajectory_reader.read_data(traj_file_content)
```

3.2 Example 2

```
from gmn_python_api import data_directory as dd
from gmn_python_api import meteor_trajectory_reader

import pandas as pd

# Get meteor data from the 2019-07-24 and 2019-07-25, and combine into a single dataframe
traj_file_content_1 = meteor_trajectory_reader.read_data(
    dd.get_daily_file_content_by_date("2019-07-24"))
traj_file_content_2 = meteor_trajectory_reader.read_data(
    dd.get_daily_file_content_by_date("2019-07-25"))
traj_df = pd.concat([traj_file_content_1, traj_file_content_2])
```

3.3 Example 3

```
from gmn_python_api import data_directory as dd
from gmn_python_api import meteor_trajectory_reader

# Get meteor data from July 2019
traj_file_content = dd.get_monthly_file_content_by_date("2019-07")
traj_sum_df = meteor_trajectory_reader.read_data(traj_file_content)
```

Fields available in the Pandas Dataframes can be found in the *Data Schemas* section.

More info can be found in the *data_directory API Reference section*.

CHAPTER

FOUR

GMN REST API

The GMN REST API provides an interface to query and retrieve meteor trajectory data by constructing read-only SQL queries on the GMN Data Store database.

We use Datasette to provide the REST API.

4.1 HTTP GET Requests

4.1.1 The General REST API Endpoint

The General REST API Endpoint allows you to make custom read-only SQL queries on the GMN Data Store. The database structure can be found here.

The endpoint is available at: `https://explore.globalmeteornetwork.org/gmn_rest_api?<query_parameters>`

The endpoint supports the following query parameters:

- `sql`: An SQL SELECT query to execute. This is required.
- `data_shape`: The shape of the data to return. Default is `objects`.
- `data_format`: The format of the data to return. Default is `json`. `csv` is also supported.

The structure of the response body is described here.

The endpoint has a maximum limit of 1000 rows. I suggest using `LIMIT` and `OFFSET` in your SQL query to paginate results. The truncated attribute in the response body will be set to true if the results have been truncated.

The response will include the header `last-modified` with the last modified time of the database in nanoseconds. You can use this when making subsequent requests to the endpoint to ensure you are using the same version of the database. E.g. when paginating results.

Queries are cached for 1 hour with a maximum size of 1GB on the server. The cache is invalidated if the GMN Data Store database has been modified by our data ingestion processes which usually run twice a day.

Queries are blocked if they take longer than 3 seconds to execute. I recommend using EXPLAIN QUERY PLAN to check the query execution plan before running a query. If you need to run a long-running query, please contact us.

4.1.2 The Meteor Summary REST API Endpoint

The Meteor Summary REST API Endpoint allows you to retrieve meteor properties from the GMN Data Store in a combined format. The properties available are described here.

It does this by substituting parts of this SQL SELECT query.

The Meteor Summary REST API endpoint is available at: `https://explore.globalmeteornetwork.org/gmn_rest_api/meteor_summary?<query_parameters>`

The API supports the following query parameters:

- `where`: A SQL SELECT WHERE clause to filter the results. Default is no filter. E.g. `iau_code = 'PER'`.
- `having`: A SQL HAVING clause to filter the results. Default is no filter. E.g. `participating_stations LIKE '%US0001%'`.
- `order_by`: A SQL ORDER BY clause to order the results. Default is no order. E.g. `meteor.unique_trajectory_identifier DESC`.
- `data_shape`: The shape of the data to return. Default is `objects`.
- `data_format`: The format of the data to return. Default is `json`. `csv` is also supported.
- `page`: The page number of the results to return. Default is 1. A maximum of 1000 results are returned per page. 0 rows are returned if the page number is greater than the number of pages of results.

The structure of the response body is described here. Ignore the truncated attribute in the response body which should always be false. The truncated attribute comes from the underlying Datasette library.

The response will include the header `last-modified` with the last modified time of the database in nanoseconds. You can use this when making subsequent requests to the endpoint to ensure you are using the same version of the database. E.g. when paginating results.

The response will also include the header `Link` with the `rel="next"` attribute to indicate the next page of results. E.g.

```
(</gmn_rest_api/meteor_summary?page=4&order_by=&data_format=&data_shape=&where=iau_code+
↪%3D+%27DSA%27>; rel="next")
```

The final page of results will include no data.

Queries are cached for 1 hour with a maximum size of 1GB on the server. The cache is invalidated if the GMN Data Store database has been modified by our data ingestion processes.

Queries are blocked if they take longer than 3 seconds to execute. I recommend using EXPLAIN QUERY PLAN to check the query execution plan before running a query. If you need to run a long-running query, please contact us.

4.1.3 Examples

Get the number of stations in the network

```
GET https://explore.globalmeteornetwork.org/gmn_rest_api?sql=SELECT+COUNT(*)+FROM+station
{
    "ok": true,
    "rows": [
        {
            "COUNT(*)": 348
        }
    ],
```

(continues on next page)

(continued from previous page)

```
    "truncated": false
}
```

Get meteor properties using unique trajectory identifier

```
GET https://explore.globalmeteornetwork.org/gmn_rest_api/meteor_summary?where=meteor.
↪unique_trajectory_identifier='20181225032412_2Sciw'
{
    "ok": true,
    "rows": [
        {
            "unique_trajectory_identifier": "20181225032412_2Sciw",
            "beginning_julian_date": 2458477.6418141434,
            "beginning_utc_time": "2018-12-25 03:24:12.742673",
            "iau_no": null,
            "iau_code": null,
            "sol_lon_deg": 273.012865,
            "app_lst_deg": 38.707634,
            "rageo_deg": 100.94855,
            "sigma": 0.1528,
            "decgeo_deg": 23.51387,
            "sigma_1": 0.2154,
            ...
            "participating_stations": "US0002,US0008"
        }
    ],
    "truncated": false
}
```

Get all recorded meteors on the 26th of December 2018 ordered by geostationary velocity

```
GET https://explore.globalmeteornetwork.org/gmn_rest_api/meteor_summary?
↪where=date(beginning_utc_time)='2018-12-26'&order_by=vgeo_km_s DESC
{
    "ok": true,
    "rows": [
        {
            "unique_trajectory_identifier": "20181226073247_wguje",
            ...
        },
        {
            "unique_trajectory_identifier": "20181226090057_xsxZ2",
            ...
        },
        {
            "unique_trajectory_identifier": "20181226071615_2uV0b",
            ...
        },
        ...
```

(continues on next page)

(continued from previous page)

```
    ],
    "truncated": false
}
```

Get all recorded meteors on the 2nd of January 2019 recorded by station US0001

```
GET https://explore.globalmeteornetwork.org/gmn_rest_api/meteor_summary?
↪where=date(beginning_utc_time)='2019-01-02'&having=participating_stations LIKE '%US0001
↪%'
{
    "ok": true,
    "rows": [
        {
            "unique_trajectory_identifier": "20190102091919_1KtJa",
            ...
        },
        {
            "unique_trajectory_identifier": "20190102092322_UkbbD",
            ...
        },
        {
            "unique_trajectory_identifier": "20190102101547_tcvE4",
            ...
        },
        ...
    ],
    "truncated": false
}
```

4.2 Python API

The `gmn_rest_api` Python module provides a Python interface to query and retrieve meteor trajectory data from the General REST API Endpoint and Meteor Summary REST API Endpoint.

Data returned from the Meteor Summary endpoint can be loaded into a Pandas DataFrame using the `meteor_trajectory_reader.read_data` function:

```
from gmn_python_api import gmn_rest_api
from gmn_python_api import meteor_trajectory_reader

data = gmn_rest_api.get_meteor_summary_data_all(where="iau_code = 'SCC' and beginning_
↪utc_time > '2019-01-01' and beginning_utc_time < '2019-04-05'",
                                                having="participating_stations LIKE '
↪%US0003%'",
                                                order_by="sol_lon_deg DESC")
df = meteor_trajectory_reader.read_data(data, input_camel_case=True)  # input_camel_
↪case=True is required for the Meteor Summary endpoint
#                                  Beginning (Julian date)  ... Participating (stations)
# Unique trajectory (identifier)                            ...
```

(continues on next page)

(continued from previous page)

```
# 20190128121133_bmAQL                    2.458512e+06  ...          [US0002, US0003]
# 20190105074710_89UEE                    2.458489e+06  ...          [US0003, US0009]
# [2 rows x 85 columns]
```

See the *gmn_rest_api API Reference section* for more information.

CHAPTER

FIVE

DATA SCHEMAS

GMN data fields are accessible through Pandas DataFrames produced by the `gmn-python-api` library. See the *meteor_trajectory_schema API Reference section* for function and variable details.

5.1 Accessing meteor trajectory fields code example

```
from gmn_python_api import data_directory as dd
from gmn_python_api import meteor_trajectory_reader

# Access column names (verbose)
traj_file_content = dd.get_daily_file_content_by_date("2019-07-24")
traj_df = meteor_trajectory_reader.read_data(
    traj_file_content,
    output_camel_case=False,
)

traj_df.iloc[0]['Vgeo (km/s)']
# 63.95235

# Access column names (camel case)
traj_file_content = dd.get_daily_file_content_by_date("2019-07-24")
traj_df = meteor_trajectory_reader.read_data(
    traj_file_content,
    output_camel_case=True,
)

traj_df.iloc[0]['vgeo_km_s']
# 63.952355
```

The model data file is `meteor_trajectory_schema._MODEL_METEOR_TRAJECTORY_FILE_PATH`. The one line version of the file is `meteor_summary_schema._MODEL_METEOR_TRAJECTORY_FILE_ONE_ROW_PATH`.

Verbose and camel case column names can be found below.

5.2 Meteor Trajectory Features

Listing of current data schema (version 1.0).

Verbose Name	Camel Case Name	Description
Unique trajectory (identifier)	unique_trajectory_identifier	(Index) A 20-character string containing the beginning time (rounded to
Beginning Julian date	beginning_julian_date	Julian date of the beginning of the meteor.
Beginning (UTC Time)	beginning_utc_time	UTC time of the beginning of the meteor.
IAU (No)	iau_no	IAU shower number, see https://www.ta3.sk/IAUC22DB/MDC2007/Ro
IAU (code)	iau_code	Three-letter IAU shower code. Sporadic meteors have a code "...".
Sol lon (deg)	sol_lon_deg	Solar longitude of the beginning of the meteor.
App LST (deg)	app_lst_deg	Apparent local sidereal time of the beginning of the meteor.
RAgeo (deg)	rageo_deg	Geocentric right ascension in the J2000 epoch.
+/- (sigma) or +/- (sigma.x)	sigma or sigma_x	One sigma error (repeated for every previous value). sigma (without nu
DECgeo (deg)	decgeo_deg	Geocentric declination in the J2000 epoch.
LAMgeo (deg)	lamgeo_deg	Geocentric ecliptic longitude in the J2000 epoch.
BETgeo (deg)	betgeo_deg	Geocentric ecliptic latitude in the J2000 epoch.
Vgeo (km/s)	vgeo_km_s	Geocentric velocity.
LAMhel (deg)	lamhel_deg	Heliocentric ecliptic longitude in the J2000 epoch.
BEThel (deg)	bethel_deg	Heliocentric ecliptic latitude in the J2000 epoch.
Vhel (deg)	vhel_km_s	Heliocentric velocity.
a (AU)	a_au	Semi-major axis.
e	e	Eccentricity.
i (deg)	i_deg	Inclination.
peri (deg)	peri_deg	Argument of perihelion.
node (deg)	node_deg	Ascending node.
Pi (deg)	pi_deg	Longitude of perihelion.
b (deg)	b_deg	Latitude of perihelion.
q (AU)	q_au	Perihelion distance.
f (deg)	f_deg	True anomaly at the beginning of the meteor.
M (deg)	m_deg	Mean anomaly.
Q (AU)	q_au_	Aphelion distance.
n (deg/day)	n_deg_day	Mean motion in the orbit.
T	t_years	Orbital period.
TisserandJ	tisserandj	Tisserand's parameter with respect to Jupiter.
RAapp (deg)	raapp_deg	Apparent ground-fixed radiant right ascension in the epoch of date.
DECapp (deg)	decapp_deg	Apparent ground-fixed radiant declination in the epoch of date.
Azim +E (of N deg)	azim_e_of_n_deg	Apparent ground-fixed radiant azimuth (+east of due north convention).
Elev (deg)	elev_deg	Apparent ground-fixed radiant elevation (i.e. entry angle).
Vinit (km/s)	vinit_km_s	Apparent ground-fixed initial velocity.
Vavg (km/s)	vavg_km_s	Apparent ground-fixed average velocity.
LatBeg (+N deg)	latbeg_n_deg	Latitude of the beginning of the meteor.
LonBeg (+E deg)	lonbeg_e_deg	Longitude of the beginning of the meteor.
HtBeg (km)	htbeg_km	Begin height of the meteor (above the WGS84 ellipsoid).
LatEnd (+N deg)	latend_n_deg	Latitude of the meteor end.
LonEnd (+E deg)	lonend_e_deg	Longitude of the meteor end.
HtEnd (km)	htend_km	End height of the meteor (above the WGS84 ellipsoid).
Duration (sec)	duration_sec	Observed meteor duration.
Peak (AbsMag)	peak_absmag	Peak magnitude normalized to the range of 100 km.
Peak Ht (km)	peak_ht_km	Height at which with peak magnitude occured.
F (param)	f_param	The F parameter defined as (HtBeg - PeakHt)/(HtBeg - HtEnd)

Table 1 – continued from previous page

Verbose Name	Camel Case Name	Description
Mass kg (tau=0.7%)	mass_kg_tau_0_7	Mass in kilograms computed with a dimensionless luminous efficiency
Qc (deg)	qc_deg	Maximum convergence angle between all stations that observed the met
MedianFitErr (arcsec)	medianfiterr_arcsec	Median angular trajectory fit errors in arc seconds.
Beg in (FOV)	beg_in_fov	Beginning of the meteor observed by at least one camera.
End in (FOV)	end_in_fov	Ending of the meteor observed by at least one camera.
Num (stat)	num_stat	Number of stations which observed the meteor.
Participating (stations)	participating_stations	Station codes of stations which observed the meteor.

Source: https://globalmeteornetwork.org/data/media/GMN_orbit_data_columns.pdf

CHAPTER

SIX

TROUBLESHOOTING

6.1 Numpy typing error

```
AttributeError: module 'numpy.typing' has no attribute 'NDArray'
```

Try installing a newer version of Numpy using:

```
pip install numpy>1.20.3
```

6.2 Pandas typing error

```
ImportError: cannot import name 'x' from 'pandas._typing'
```

This is a known issue with some newer versions of Pandas. Try installing an older version using:

```
pip uninstall pandas
pip install pandas==1.1.5
```

https://stackoverflow.com/questions/65684415/exporting-csv-shows-importerror-cannot-import-name-compressionoptions-from-p

6.3 Scipy No BLAS/LAPACK libraries found

Install `openblas` and `lapack`, see link below.

https://github.com/scipy/scipy/issues/9005

CHAPTER

SEVEN

CONTRIBUTOR GUIDE

Thank you for your interest in improving this project. This project is open-source under the MIT license and welcomes contributions in the form of bug reports, feature requests, and pull requests.

Here is a list of important resources for contributors:

- Source Code
- Documentation
- Issue Tracker

7.1 How to report a bug

Report bugs on the Issue Tracker.

When filing an issue, make sure to answer these questions:

- Which operating system and Python version are you using?
- Which version of this project are you using?
- What did you do?
- What did you expect to see?
- What did you see instead?

The best way to get your bug fixed is to provide a test case, and/or steps to reproduce the issue.

7.2 How to request a feature

Request features on the Issue Tracker.

7.3 How to set up your development environment

You need Python 3.8, 3.9, and 3.10. Follow this guide to install pyenv and the required Python versions.

And the following tools:

- Poetry
- Nox
- nox-poetry

Install the package with development requirements:

```
poetry install
```

You can now run an interactive Python session, or the command-line interface:

```
poetry run python
```

```
poetry run gmn-python-api
```

7.4 How to test the project

Run the full test suite:

```
nox
```

List the available Nox sessions:

```
nox --list-sessions
```

You can also run a specific Nox session. For example, invoke the unit test suite like this:

```
nox --session=unit-tests
```

Unit tests are located in the `tests` directory, and are written using the pytest testing framework.

7.5 How to build the documentation

```
nox --session=docs
```

Built documentation is located in the `docs/_build` directory.

7.6 How to submit changes

Open a pull request to submit changes to this project.

Your pull request needs to meet the following guidelines for acceptance:

- The Nox test suite must pass without errors and warnings.
- Include unit tests. This project maintains 100% code coverage.
- If your changes add functionality, update the documentation accordingly.

Feel free to submit early, though—we can always iterate on this.

It is recommended to open an issue before starting work on anything. This will allow a chance to talk it over with the owners and validate your approach.

7.7 Useful links

- https://cookiecutter-hypermodern-python.readthedocs.io/en/2021.11.26/guide.html

CHAPTER

EIGHT

LICENSE

```
MIT License

Copyright © 2022 Ricky Bassom

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

**The software is provided "as is", without warranty of any kind, express or
implied, including but not limited to the warranties of merchantability,
fitness for a particular purpose and noninfringement. In no event shall the
authors or copyright holders be liable for any claim, damages or other
liability, whether in an action of contract, tort or otherwise, arising from,
out of or in connection with the software or the use or other dealings in the
software.**
```

This library provides a Python API for accessing open Global Meteor Network (GMN) meteor trajectory data. Global meteor data is generated using a network of low-light cameras pointed towards the night sky. Meteor properties (radiants, orbits, magnitudes and masses) are produced by the GMN and are available through this library.


Leeds
Preston
Hull
Sheffield
Dublin
Liverpool
Lincoln
Nottingham
England
Birmingham
Peterborough
Norwich
Cymru / Wales
Worcester
Cambridge
St George's Channel
Gloucester
Oxford
Swansea
Cardiff
London
Southend-on-Sea
Bath
Dunkerque
Exeter
Southampton
Brighton
Strait of Dover / Pas de Calais
Portsmouth
Plymouth
Guernsey
Le Havre
Rouen
Groningen
Oldenburg
Assen
Zwolle
Haarlem
Nederland
Den Haag
Arnhem
Münster
Middelburg
Eindhoven
Nordrhein-Westfalen
Brugge
Vlaanderen
Düsseldorf
Aachen
Bonn
Siegen
Lille
België / Belgique / Belgien
Koblenz
Hauts-de-France
Letzebuerg

CHAPTER

NINE

FEATURES

- Listing available daily and monthly meteor trajectory files from the GMN Data Directory.
- Downloading specific meteor trajectory data from the GMN Data Directory or GMN REST API.
- Functions for loading meteor trajectory data into Pandas DataFrames.
- Functions for retrieving available IAU registered meteor showers.

CHAPTER

TEN

REQUIREMENTS

- Python 3.8, 3.9, or 3.10

CHAPTER

ELEVEN

INSTALLATION

You can install `gmn-python-api` via pip from PyPI:

```
pip install gmn-python-api
```

Or install the latest development code, through TestPyPI or directly from GitHub via pip:

```
pip install -i https://test.pypi.org/simple/ --extra-index-url https://pypi.org/simple␣
↪gmn-python-api==<version>
```

Or

```
pip install git+https://github.com/rickybassom/gmn-python-api
```

Refer to the *Troubleshooting* guide if you encounter any issues.

CHAPTER

TWELVE

USAGE

Simple meteor analysis example:

```
from gmn_python_api import data_directory as dd
from gmn_python_api import meteor_trajectory_reader

# Analyse recorded meteor data for the 24th of July 2019
traj_file_content = dd.get_daily_file_content_by_date("2019-07-24")

# Read data as a Pandas DataFrame
traj_df = meteor_trajectory_reader.read_data(traj_file_content)

print(f"{traj_df['Vgeo (km/s)'].max()} km/s was the fastest geostationary velocity")
# Output: 65.38499 km/s was the fastest geostationary velocity

print(f"{traj_df.loc[traj_df['IAU (code)'] == 'PER'].shape[0]} Perseid meteors")
# Output: 3 Perseid meteors

print(f"Station #{traj_df['Num (stat)'].mode().values[0]} recorded the most meteors")
# Output: Station #2 recorded the most meteors
```

Please see the Usage and API Reference sections for more details.

CHAPTER

THIRTEEN

CONTRIBUTING

Contributions are very welcome. To learn more, see the *Contributing guide*.

CHAPTER

FOURTEEN

LICENSE

Distributed under the terms of the MIT license, `gmn-python-api` is free and open source software.

PYTHON MODULE INDEX

g

INDEX

Symbols

B

D

G

I

L

M

Q

R

S