DEOS Data API

Version 1.0

Released 06/01/2026

TABLE OF CONTENTS

TABLE OF CONTENTS 1
Introduction 1
Accessing API Endpoints 1
API Endpoints 3
Station Metadata 3
Data Types 5
Latest Observation - [API Key Required] 6
Summary Data - [API Key Required] 7
Historic Data Retrieval - [API Key Required] 9
Observing Platform 11
Observing Platform Search 12
DEOS API Libraries 13
Python 14
- Installation 14
- Usage 14
- Function List 15

R 20
- Installation 20
- Usage 20
- Function List 21

Introduction

The services below provide options for accessing DEOS Network data programmatically. Most services support JSON and CSV formats for returning results, and where it makes sense, geojson format is available. Types of data available through this API include station metadata, recent conditions, and historical 5-minute and daily data.

Accessing API Endpoints

Some services require an API key in order to retrieve data. You can sign up for an API key by filling out the access request form:

https://services.cema.udel.edu/deos_services/register

Once you have submitted your request, you will receive a confirmation email and our staff will review your information. Upon approval of your request, an API key will be sent to your email. This API key will be tied to your email address and should only be used by you. Sharing or making your API key available may lead to your access being revoked.

When accessing API controlled endpoints, you will need to send your API key to the server as a request header entry, X-API-KEY, with your key set as the value. API endpoints that require a key for access are noted in each endpoint's description within this guide. Using curl, you can supply your API key using the -H flag (curl -H 'X-API-KEY: YOUR_API_KEY' API_ENDPOINT).

Some services are limited by the number of requests made per minute and the amount of data that can be retrieved in a single request. These limitations are noted in each endpoint's description. This is done to ensure adequate resources are available for all users and block malicious/bot traffic as best as possible while still allowing for users to retrieve necessary data.

Please note: some API endpoints have request rate restrictions in order to minimize the impact on our servers. If you experience rate limit errors when using the DEOS Data API, consider adding delays between your API requests to avoid this issue. Further, users who abuse their access to the system (making frequent large data requests, spamming the API, etc) are at risk of having their access revoked. Please use your API access responsibly and consider other users and the stability of our system when making requests. If you need to access a large amount of data, please contact us for assistance.

Regarding the X-API-KEY: The **X-API-KEY** header is a custom HTTP header used to authenticate requests to an API. Clients must include this header in every request so the server can validate the request before granting access and returning requested data. Each key uniquely identifies a user and must be kept secret. For our purposes, this key allows for

restricting large data requests or repeated small requests in a short time frame that could impact system availability and processing. Users are encouraged to consider their request and its impact on others before submitting large data pulls. If you have need of a large timeframe of data, please reach out to us for assistance.
The X-API-KEY is straightforward to use in many languages, select examples follow:

R Statistical Software:

```
library(httr2)
request("https://api.example.com/v1/data") |>
  req_headers("X-API-KEY" = "YOUR_API_KEY") |>
  req_perform()
```

Or

```
library(httr)

GET(
  "https://api.example.com/v1/data",
  add_headers("X-API-KEY" = "YOUR_API_KEY")
)
```

Javascript:

```
fetch("https://api.example.com/v1/data", {
  method: "GET",
  headers: {
    "X-API-KEY": "YOUR_API_KEY"
  }
})
  .then(res => res.json())
  .then(data => console.log(data));
```

Python:

```
import requests
response = requests.get(
    "https://api.example.com/v1/data",
    headers={"X-API-KEY": "YOUR_API_KEY"}
)
print(response.json())
```

cURL:

```
curl -H ‘X-API-KEY: YOUR_API_KEY’ “https://api.example.com/v1/data”
```

API Endpoints

Station Metadata

Purpose: Retrieves high level metadata about all stations within the DEOS network or a specific station within the network. The latitude and longitude for each station are rounded to 4 decimal places and the elevation is rounded to the nearest decimal place. The station names returned from this service are used in data retrieval end points covered later in this document. Several optional parameters are provided as a means to restrict stations to a useful subset based on common criteria (ie. subsetting by county, station type, or watershed).

Base URL:
https://services.cema.udel.edu/deos_services/api/station_metadata/
https://services.cema.udel.edu/deos_services/api/station_metadata/{STATION_NAME}

Note: {STATION_NAME} refers to the 'station_name' values returned by this service. Try the first Base URL above to get a list of 'station_name' values (e.g. DAGF).

Optional Parameters:

- weather_station(boolean): When set to a True value, only weather stations will be returned. When set to a False value, only entries that are not identified as a weather station will be returned.
- streamflow_station(boolean): When set to a True value, only streamflow stations will be returned. When set to a False value, only entries that are not identified as a streamflow station will be returned.
- tidal_station(boolean): When set to a True value, only tidal stations will be returned. When set to a False value, only entries that are not identified as a tidal station will be returned.
- waterbouy_station(boolean): When set to a True value, only water buoy stations will be returned. When set to a False value, only entries that are not identified as a weather water buoy will be returned.
- water_quality(boolean): When set to a True value, only water quality stations will be returned. When set to a False value, only entries that are not identified as water quality stations will be returned.
- ground_water(boolean): When set to a True value, only ground water stations will be returned. When set to a False value, only entries that are not identified as ground water stations will be returned.
- in_use(boolean): Limits returned entries to active (currently installed and taking observations) stations. Setting this to a False value will return decommissioned stations.
    - **Default**: TRUE (only return active stations)
- county(string): Limit returned entries to a given county name.
- watershed(string): Limit returned entries to a given huc12 watershed name
- huc12_id(number): Limit returned entries to a given huc12 watershed id

- format(string): sets the desired output format for the service. Currently supported options are csv, json(**Default**), or geojson.

Example Usage:

Retrieve all metadata for all DEOS stations:
https://services.cema.udel.edu/deos_services/api/station_metadata

Retrieve all metadata for all DEOS stations in Kent County:
https://services.cema.udel.edu/deos_services/api/station_metadata?county=kent

Retrieve metadata for the Newark, DE Ag Farm Location:
https://services.cema.udel.edu/deos_services/api/station_metadata/DAGF

Retrieve all DEOS weather stations in CSV format:
https://services.cema.udel.edu/deos_services/api/station_metadata?weather_station=true&format=CSV

Retrieve all DEOS groundwater stations in geojson format:
https://services.cema.udel.edu/deos_services/api/station_metadata/?groundwater_station=true&format=geojson

Data Types

Purpose: Retrieves high level metadata for all data types or a single data type (based on id) currently defined within DEOS. Data types refers to an observation recorded by a sensor(ie. Air Temperature), derived from a sensor measurement(Wind Chill/Heat Index), or an aggregated measurement (ie. Daily Mean Air Temperature).

Base URL:
https://services.cema.udel.edu/deos_services/api/data_types/
https://services.cema.udel.edu/deos_services/api/data_types/{DATA_TYPE_ID}

Note: {DATA_TYPE_ID} values refer to a 'data_type_id' value returned by this service. Try the first Base URL above to get a list of 'data_type_id' values.

Optional Parameters:

- format(string): sets the desired output format for the service. Currently supported options are:
  csv, json (**Default**)

Example Usage:

Retrieve all defined data types:
https://services.cema.udel.edu/deos_services/api/data_types

Retrieve all defined data types in CSV format:
https://services.cema.udel.edu/deos_services/api/data_types?format=CSV

Retrieve metadata for Air Temperature data type:
https://services.cema.udel.edu/deos_services/api/data_types/2

Latest Observation - [API Key Required]

Purpose: Retrieves the last observation for all stations within the DEOS network or a given station within the DEOS network. The observation returned will update as new data is ingested in the system making this an easy way to dynamically monitor conditions at a specific site or across the network. A valid API key is required in order to retrieve data from this service. Include your API key in the request by setting the X-API-KEY header to your key.

Base URL:
https://services.cema.udel.edu/deos_services/api/latest_ob/
https://services.cema.udel.edu/deos_services/api/latest_ob/{STATION_NAME}

Note: {STATION_NAME} refers to 'station_name' values returned by the station_metadata service.

Optional Parameters:

- format(string): sets the desired output format for the service. Currently supported options are:
  csv, json (**Default**), geojson
- data_types: csv list of data_type_ids to restrict output to.
  - **Default**: No restrictions/all available data returned

Example Usage:

Retrieve latest observation at all DEOS stations:
curl -H 'X-API-KEY: YOUR_API_KEY'
'https://services.cema.udel.edu/deos_services/api/latest_ob'

Retrieve latest Air Temperature and Precipitation data at Newark, DE Ag Farm location:
curl -H 'X-API-KEY: YOUR_API_KEY'
'https://services.cema.udel.edu/deos_services/api/latest_ob/DAGF?data_types=2,10'

Retrieve latest observations at Newark, DE Ag Farm Location in CSV format
curl -H 'X-API-KEY: YOUR_API_KEY'
'https://services.cema.udel.edu/deos_services/api/latest_ob/DAGF?format=CSV'

Summary Data - [API Key Required]

Purpose: Retrieves summary data for a given station at the daily timescale. Retrieving a single day or range of days based on a start and end date is permitted. A valid API key is required in order to retrieve data from this service. Include your API key in the request by setting the X-API-KEY header to your key.

> **Usage Restrictions:** Each API key is limited to 24 requests/minute and 31 days/request. If you receive a usage limit error message from the API request, consider adding time (i.e., delays) between your requests, or reducing the amount of data (i.e., fewer days) in each request.

Base URL:
https://services.cema.udel.edu/deos_services/api/summary/{STATION_NAME}
https://services.cema.udel.edu/deos_services/api/summary/{STATION_NAME}/{START_TIME}
https://services.cema.udel.edu/deos_services/api/summary/{STATION_NAME}/{START_TIME}/{END_TIME}

Note: {STATION_NAME} refers to 'station_name' values returned by the station_metadata service. Time formats should be in the form of YYYY-mm-dd format (Example: 2025-01-01).

Optional Parameters:

- data_types(csv integer list): a comma separated list of data_type_ids to restrict output to.
  - Default: **None** (No restrictions/all available data returned)
- data_quality(boolean): When True, returned values have passed necessary quality control criteria ensuring best quality data is available. When False, values that have failed quality control criteria will also be included.
  - **Default**: True (return quality controlled data)
- format(string): sets the desired output format for the service. Currently supported options are:
  csv, json (**Default**)

Example Usage:

Retrieve a summary of yesterday's observations at the Newark, DE Ag Farm:
curl -H 'X-API-KEY: YOUR_API_KEY'
'https://services.cema.udel.edu/deos_services/api/summary/DAGF'

Retrieve summary of yesterday's temperature range at the Newark, DE Ag Farm:
curl -H 'X-API-KEY: YOUR_API_KEY'
'https://services.cema.udel.edu/deos_services/api/summary/DAGF?data_types=43,44,45'

Retrieve mean daily temperature at the Newark, DE Ag Farm location for the January 2025 in CSV format:
curl -H 'X-API-KEY: YOUR_API_KEY'
'https://services.cema.udel.edu/internal_services/api/summary/DAGF/2025-01-1/2025-01-31?data_types=43&format=CSV'

Historic Data Retrieval - [API Key Required]

Purpose: Retrieves raw/base 5-minute observations for a given station during the requested start and end timestamp. A valid API key is required in order to retrieve data from this service. Include your API key in the request by setting the X-API-KEY header to your key.

> **Usage Restrictions:** Each API key is limited to 10 requests/minute and one day's worth of data per request. If you receive a usage limit error message from the API request, consider adding time (i.e., delays) between your requests, or reducing the amount of data in each request.

Base URL:
https://services.cema.udel.edu/deos_services/api/data/{STATION_NAME}/{START_TIME}
https://services.cema.udel.edu/deos_services/api/data/{STATION_NAME}/{START_TIME}/{END_TIME}

Note: {STATION_NAME} refers to 'station_name' values returned by the station_metadata service. Time formats should be in the form of 'YYYY-mm-dd HH:MM' format (Example: 2025-01-01 00:00).

Optional Parameters:

- fill_missing(boolean): attempts to create missing entries based on common 5-minute timestamp
- data_types: a CSV list of data_type_ids to restrict output to
    - **Default**: None (No restrictions/all available data returned)
- data_quality(boolean): When True, returned values have passed necessary quality control criteria ensuring best quality data is available. When False, values that have failed quality control criteria will also be included.
    - **Default**: True(return quality controlled data)
- format(string): sets the desired output format for the service. Currently supported options are:
  csv, json (**Default**)

Example Usage:

Retrieve an hour of data at the Newark, DE Ag Farm Location:
curl -H 'X-API-KEY: YOUR_API_KEY'
'https://services.cema.udel.edu/internal_services/api/data/DAGF/2025-01-01 00:00/2025-01-01 01:00'

Retrieve air temperature measurements at the Newark, DE Ag Farm Location:
curl -H 'X-API-KEY: YOUR_API_KEY'
'https://services.cema.udel.edu/internal_services/api/data/DAGF/2025-01-01 00:00/2025-01-01 01:00?data_types=2'

Retrieve air temperature measurements at the Newark, DE Ag Farm Location in CSV format:
curl -H 'X-API-KEY: YOUR_API_KEY'
'https://services.cema.udel.edu/internal_services/api/data/DAGF/2025-01-01 00:00/2025-01-01 01:00?data_types=2&format=CSV'

Observing Platform

Purpose: retrieves the data types observed at a given station along with a period of record of observations. Please note: We do not guarantee that observations will exist during the period of record, just that a sensor capable of measuring the observation was installed on the starting date and on the ending date.

Base URL:
https://services.cema.udel.edu/deos_services/api/station_platform/
https://services.cema.udel.edu/deos_services/api/station_platform/{STATION_NAME}

Note: {STATION_NAME} refers to 'station_name' values returned by the station_metadata service.

Optional Parameters:

- format(string): sets the desired output format for the service. Currently supported options are:
  csv, json (**Default)**

Example Usage:

Retrieve data types observed at all stations:
https://services.cema.udel.edu/deos_services/api/station_platform

Retrieve data types observed at the Newark, DE Ag Farm location:
https://services.cema.udel.edu/deos_services/api/station_platform/DAGF

Observing Platform Search

Purpose: Allows for searching for stations observing a specified data_type_id (see Data Types). A timestamp may be supplied to further restrict output to stations with a sensor installed measuring the data_type_id at a specific time. It is important to note that the list of stations returned indicates the station had a sensor installed capable of measuring the selected data type. It does not mean a valid measurement was recorded that day. Observations could be missing due to sensor issues, QC problems, technical work, or other issues.

Base URL:
https://services.cema.udel.edu/deos_services/api/station_platform/search/{DATA_TYPE_ID}
https://services.cema.udel.edu/deos_services/api/station_platform/search/{DATA_TYPE_ID}/{TIMESTAMP}

Note: {DATA_TYPE_ID} values refer to 'data_type_id' values returned by the data_types service. Time formats should be in the form of YYYY-mm-dd format (Example: 2025-01-01).

Optional Parameters:
None

Example Usage:

Retrieve stations observing Air Temperature
https://services.cema.udel.edu/deos_services/api/station_platform/search/2

Retrieve stations observing Air Temperature on 2025-01-01
https://services.cema.udel.edu/deos_services/api/station_platform/search/2/2025-01-01

DEOS API Libraries

API libraries are provided in R and Python to facilitate accessing API end-points defined above. These libraries are wrappers around the API endpoints listed above and are intended to remove technical hurdles with data retrieval in order to speed up data analysis and research.

If you run into issues accessing data, please contact us as cema-info@udel.edu. We will try to support you as best as possible with retrieving data but cannot provide support or assistance with debugging your code, installing/setting up environments, or other issues not related to data access/retrieval.

The libraries are available on GitHub using
git clone git@github.com:udel-cema/deos_api_scripts.git
We recommend staying up to date with the library versions as endpoints may change or improvements to code/bug fixes may solve an important issue.

Python

Installation

Download or clone the GitHub repository to the directory you wish to work from. You will need to have the requests and json library installed in your python environment. These are both lightweight with minimal requirements. If you would like to use Pandas for data analysis, install that in your environment as well.

Usage

Import the deos_api scripts as you would any other library. Help functions are available from within Python via help(deos_api) or help(deos_api.function_name).

Function List

Station_metadata

station_metadata(station_name = None, pandas = False)

Required Parameters:
None

Optional Parameters:

- station_name: a specific station name with the DEOS network to restrict retrieval to
    - **Default**: None(return all stations)
- pandas: If the pandas library is available, a DataFrame is returned containing the requested data
    - **Default**: False(return Python dictionary)

Usage:
Example 1:
Retrieve all stations as a Python dictionary
import deos_api
all_stations = deos_api.station_metadata()

Example 2:
Retrieve all stations as a Pandas DataFrame
import deos_api
import pandas as pd
all_stations = deos_api.station_metadata(pandas=True)

Example 3:
Retrieve the Newark, DE Ag Farm Location only
import deos_api
all_stations = deos_api.station_metadata('DAGF')
OR
import deos_api
all_stations = deos_api.station_metadata(station_name = 'DAGF')

Data Types

data_types(data_type = None, pandas = False)

Required Parameters
None

Optional Parameters:

- data_type: a specific data_type defined for the DEOS network to restrict retrieval to
    - **Default**: None(return all available data_types)
- pandas: If the pandas library is available, a DataFrame is returned containing the requested data
    - **Default**: False(return Python dictionary)

Usage:
Example 1:
Retrieve all available data types as a Python dictionary
import deos_api
all_data_types = deos_api.data_types()

Example 2:
Retrieve all available data types as a Pandas DataFrame
import deos_api
import pandas as pd
all_data_types = deos_api.data_types(pandas=True)

Example 3:
Retrieve Air Temperature data type only
import deos_api
all_data_types = deos_api.data_types(2)
OR
import deos_api
all_stations = deos_api.station_metadata(data_type = 2)

Latest Observations

latest_ob(api_key, station_name = None, data_types = None, pandas=False)

Required Parameters:

- api_key: a valid API key(sent upon registration). **NOTE**: *number of access requests is restricted based on time interval. You may be denied when making multiple requests in quick succession. See the Latest Observations API endpoint section for more information.*

Optional Parameters:

- station_name: A specific station within the DEOS network to restrict data retrieval to
  - **Default**: None(return all available station data)
- data_types: Restrict data retrieval to a specific data type id or a CSV list of data type ids.
  - **Default**: None(return all available data)
- pandas: If the pandas library is available, a DataFrame is returned containing the requested data
  - **Default**: False(return Python dictionary)

Usage:
Example 1:
Retrieve last data available at the Newark, DE Ag Farm Location
import deos_api
deos_api.latest_ob(YOUR_API_KEY_HERE, 'DAGF')

Example 2:
Retrieve last Air Temperature observed at the Newark, DE Ag Farm Location
import deos_api
deos_api.latest_ob(YOUR_API_KEY_HERE, 'DAGF', 2)

Example 3:
Retrieve last Air Temperature and Precipitation observed at the Newark, DE Ag Farm Location
import deos_api
deos_api.latest_ob(YOUR_API_KEY_HERE, 'DAGF', '2,10')

Summary Data

summary_data(api_key, station_name, start_time = None, end_time = None, data_types = None, pandas=False)

Required Parameters:

- api_key: a valid API key(sent upon registration). **NOTE**: *number of access requests is restricted based on time interval. You may be denied when making multiple requests in quick succession. See the Summary Data API endpoint section for more information.*
- station_name: A specific station within the DEOS network (Example: DAGF)

Optional Parameters:

- start_time: The beginning timestamp to retrieve data
    - **Default**: None(Yesterday's date)
- end_time: The last timestamp to retrieve data
    - **Default**: None(Yesterday's date)
- Data_types: a specific data type id or a CSV list of data type ids to restrict data retrieval to.
    - **Default**: None(retrieve all available data)
- pandas: If the pandas library is available, a DataFrame is returned containing the requested data
    - **Default**: False(return Python dictionary)

Usage:
Example 1:
Retrieve summarized data for yesterday at the Newark, DE Ag Farm Location
import deos_api
deos_api.summary_data(YOUR_API_KEY_HERE, 'DAGF')

Example 2:
Retrieve min, mean, and max Air Temperature measurements for yesterday at the Newark, DE Ag Farm location
import deos_api
deos_api.summary_data(YOUR_API_KEY_HERE, 'DAGF', data_types='43,44,45')

Example 3:
Retrieve summarized data for January 2025 at the Newark, DE Ag Farm Location
import deos_api
deos_api.summary_data(YOUR_API_KEY_HERE, 'DAGF', '2025-01-01', '2025-01-31')

Historical Data

historic_data(api_key, station_name, start_time, end_time = None, data_types = None, pandas = False)

Required Parameters:

- api_key: a valid API key(sent upon registration). **NOTE**: *number of access requests is restricted based on time interval. You may be denied when making multiple requests in quick succession. See the Historical Data API endpoint section for more information.*
- station_name: A specific station within the DEOS network (Example: DAGF)
- start_time: The beginning timestamp to retrieve data

Optional Parameters:

- end_time: The last timestamp to retrieve data
    - **Default**: None(start time(a single 5-minute observation))
- Data_types(str): a specific data type id or a CSV list of data type ids to restrict data retrieval to.
    - **Default**: None(retrieve all available data)
- pandas: If the pandas library is available, a DataFrame is returned containing the requested data
    - **Default**: False(return Python dictionary)

Usage:
Example 1:
Retrieve data for 2025-01-01 00:00 at the Newark, DE Ag Farm location:
import deos_api
deos_api.historic_data(YOUR_API_KEY_HERE, 'DAGF', '2025-01-01 00:00')

Example 2:
Retrieve data for 2025-01-01 00:00 to 2025-01-01 01:00 at the Newark, DE Ag Farm location:
import deos_api
deos_api.historic_data(YOUR_API_KEY_HERE, 'DAGF', '2025-01-01 00:00', '2025-01-01 01:00')

Example 3:
Retrieve Air Temperature for 2025-01-01 01:00 to 2026-01-01 02:00 at the Newark, DE Ag Farm location:
import deos_api
deos_api.historic_data(YOUR_API_KEY_HERE, 'DAGF', '2026-01-01 01:00', '2026-01-01 02:00', '2')

R

Installation

Download or clone the GitHub repository to the directory you wish to work from. You will need to have the httr and jsonlite library installed in your R environment. These are both lightweight with minimal requirements.

From within R, install the library as you would normally. For example:
install.packages('R/deos_api/', repos=NULL)

Or from the included source package file
install.packages('R/deosapi_1.0.0.tar.gz', repos=NULL)

Usage

Once installed, import the library as you would normally. For example:

library(deosapi)

Help functions are available from within R via ?(deos_api) or help(deos_api.function_name).

Function List

Station Metadata

station_metadata(station_name = NA)

Required Parameters:
None

Optional Parameters:

- station_name: a specific station name with the DEOS network (Example: DAGF)
    - **Default**: NA (return all stations)

Usage:
Example 1:
Retrieve all stations
library(deosapi)
all_stations = station_metadata()

Example 2:
Retrieve the Newark, DE Ag Farm Location only
library(deosapi)
all_stations = station_metadata('DAGF')

Data Types

data_types(data_type = None)

Required Parameters:
None

Optional Parameters:

- data_type: a specific data type id defined for the DEOS network
    - **Default**: None(return all available data_types)
- pandas: If the pandas library is available, a DataFrame is returned containing the requested data
    - **Default**: False(return Python dictionary)

Usage:
Example 1:
Retrieve all data types defined in the system
library(deosapi)
data_types()

Example 2:
Retrieve metadata for the Air Temperature data type
library(deosapi)
data_types(2)

Latest Observations

latest_ob(api_key, station_name = None, data_types = None)

Required Parameters:

- api_key: a valid API key(sent upon registration). **NOTE**: *number of access requests is restricted based on time interval. You may be denied when making multiple requests in quick succession. See the Latest Observations API endpoint section for more information.*

Optional Parameters:

- station_name: A specific station within the DEOS network
  - **Default**: None(return all available station data)
- data_types: Restrict data retrieval to a specific data type id or a CSV list of data type ids.
  - **Default**: None(return all available data)
- pandas: If the pandas library is available, a DataFrame is returned containing the requested data
  - **Default**: False(return Python dictionary)

Usage:
Example 1:
Retrieve latest conditions across the entire DEOS network
library(deosapi)
latest_ob(YOUR_API_KEY)

Example 2:
Retrieve latest conditions at the Newark, DE Ag Farm
library(deosapi)
latest_ob(YOUR_API_KEY, 'DAGF')

Example 3:
Retrieve latest Air Temperature data at the Newark, DE Ag Farm Location
library(deosapi)
latest_ob(YOUR_API_KEY, 'DAGF', 2)

Summary Data

summary_data(api_key, station_name, start_time = None, end_time = None, data_types = None)

Required Parameters:

- api_key: a valid API key(sent upon registration). **NOTE**: *number of access requests is restricted based on time interval. You may be denied when making multiple requests in quick succession. See the Summary Data API endpoint section for more information.*
- station_name: A specific station within the DEOS network (Example: DAGF)

Optional Parameters:

- start_time: The beginning timestamp for data retrieval
    - **Default**: None(Yesterday's date)
- end_time: The last timestamp for data retrieval
    - **Default**: None(Yesterday's date)
- data_types: a specific data type id or a CSV list of data type ids
    - **Default**: None(retrieve all available data)
- pandas: If the pandas library is available, a DataFrame is returned containing the requested data
    - **Default**: False(return Python dictionary)

Usage:
Example 1:
Retrieve summarized observations for yesterday at the Newark, DE Ag Farm location
library(deosapi)
summary_data(YOUR_API_KEY, 'DAGF')

Example 2:
Retrieve summarized observations for Jan 1 through Jan 5 2025 at the Newark, DE Ag Farm location
library(deosapi)
summary_data(YOUR_API_KEY, 'DAGF', '2025-01-01', '2025-01-05')

Example 3:
Retrieve mean daily Air Temperature observations for Jan 1 through Jan 5 2025 at the Newark, DE Ag Farm location
library(deosapi)
summary_data(YOUR_API_KEY, 'DAGF', '2025-01-01', '2025-01-05', 43)

Historical Data

historic_data(api_key, station_name, start_time, end_time = None, data_types = None)

Required Parameters:

- api_key: a valid API key(sent upon registration). **NOTE**: *number of access requests is restricted based on time interval. You may be denied when making multiple requests in quick succession. See the Historical Data API endpoint section for more information.*
- station_name: A specific station within the DEOS network to restrict data retrieval to
- start_time: The beginning timestamp for data retrieval

Optional Parameters:

- end_time: The last timestamp for data retrieval
    - **Default**: None(start time(a single 5-minute observation))
- data_types(str): a specific data type id or a CSV list of data type ids
    - **Default**: None(retrieve all available data)
- pandas: If the pandas library is available, a DataFrame is returned containing the requested data
    - **Default**: False(return Python dictionary)

Usage:
Example 1:
library(deosapi)
historic_data(YOUR_API_KEY, 'DAGF', '2025-01-01 00:00')

Example 2:
library(deosapi)
historic_data(YOUR_API_KEY, 'DAGF', '2025-01-01 00:00', '2025-01-01 00:10')

Example 3:
library(deosapi)
historic_data(YOUR_API_KEY, 'DAGF', '2025-01-01 00:00', '2025-01-01 00:10', 2)