BEApy

Release 0.6.0-dev7

Sep 26, 2019

Contents

This package provides high(er) level methods for interacting with the Bureau of Economic Analysis (BEA) API. If you want to learn how to use BEApy check out the *Examples*, and if you want to see the documentation, go to *Module reference*.

CHAPTER 1

Contents

1.1 Examples

1.2 Module reference

1.2.1 Constructor

BEA(api_key)	The primary class in the BEApy package, handles sending requests and manipulating the results.

1.2.2 Raw API call

BEAAPI.send_request(params)	Base method for sending a GET request to the API.

1.2.3 Metadata

BEA.dataset_list()	Get a list of all availible datasets (API method getDatasetList)
BEA.parameter_list(dataset)	Get a list of availible parameters for the dataset (API method getParameterList)
BEA.parameter_values(dataset, parameter, ...)	Get a list of possible values for the parameter in a dataset (API method getParameterValues).
BEA.find_table(dataset, search)	A helper function to search for a table based on its description (or some words from it).

1.2.4 Get Data

BEA.get_data(dataset, **kwargs)	Get the data (API method getData).

1.2.5 Results

BEAResults(results, dataset)	This class holds the results from the *BEA.get_data* call.

1.2.6 References

Module for sending requests to BEA API and handling errors

class beapy.api.**BEAAPI**(*api_key*)
Base class for communicating with the API, outputs raw JSON results.

This class communicates with the BEA API and handles possible errors, returning unfiltered JSON results in case of success. It stores your API key on initiation, and uses it for future requests. It serves as a base class to enable the *BEA* class to abstract from requests and error handling.

Parameters **api_key** (str) – Your BEA API key

send_request(*params*)
Base method for sending a GET request to the API.

This method sends a GET request to the base url with the params + the API key + JSON format and handles error messages. If successful, returns the Results part of the response.

Parameters **params** (dict) – Additional params to add to the GET request

Return type dict

exception beapy.api.**BEAError**(*status_code*, *message*)
Error class to handle BEA API specific errors

class beapy.**BEA**(*api_key*)
The primary class in the BEApy package, handles sending requests and manipulating the results. Needs to be initiated with your BEA API key once, it is stored for all later requests.

Parameters **api_key** (str) – Your BEA API key

dataset_list()
Get a list of all availible datasets (API method getDatasetList)

Returns A DataFrame with a description and name of availible datasets.

Return type pd.DataFrame

find_table(*dataset*, *search*)
A helper function to search for a table based on its description (or some words from it).

This function will return all tables from a dataset, whose description (as would be returned by *parameter_values*) contains the search string. This can be useful to find a table based on its name on the BEA website.

The supported datasets (ones with tables) are: NIPA, NIUnderlyingDetail, FixedAssets, GDPbyIndustry, UnderlyingGDPbyIndustry, InputOutput and Regional.

Additionaly, the function also performs the search on some table-like parameters of other datasets (Indicator for ITA, TypeOfInvestment for IIP, TypeOfService for IntlServTrade and SeriesID for MNE).

Parameters

- **dataset** (str) – The name of the dataset
- **parameter** – A string which will try to be matched to a substring of tables' description.

Returns A DataFrame with a name and description of matching tables

Return type pd.DataFrame

get_data(*dataset*, ***kwargs*)

Get the data (API method getData).

Parameters

- **dataset** (str) – The name of the dataset
- **kwargs** – Additional parameters (years, table name, etc.) passed to the API. The values of the arguments have to be strings or lists of strings. Additionaly, for datasets that admit the Frequency parameter, its value must be set and can not contain multiple values (or ALL).

Returns A *BEAResults* object containing the results.

Return type *BEAResults*

parameter_list(*dataset*)

Get a list of availible parameters for the dataset (API method getParameterList)

Usually the parameter list will also include the value of the ALL option, as well as data type, whether the parameteris is required, and whether it accepts multiple values.

Parameters **dataset** (str) – The name of the dataset

Returns A DataFrame with a description and name of availible parameters, as well as some other options.

Return type pd.DataFrame

parameter_values(*dataset*, *parameter*, ***kwargs*)

Get a list of possible values for the parameter in a dataset (API method getParameterValues).

Optionally, filter the list based on values of other parameters (API method getParameterValuesFiltered). Be careful in this case, as the filtered method is not implemented for all datasets.

Parameters

- **dataset** (str) – The name of the dataset
- **parameter** (str) – The name of the (target) parameter whose values you are retrieving
- **kwargs** – Additional arguments (parameters), based on which to filter the target parameter. All arguments should be strings or lists of strings. If kwags is empty, getParameterValues API method is used. Otherwise getParameterValuesFiltered is used.

Returns A DataFrame with a description and name (key) of availible parameters values

Return type pd.DataFrame

class beapy.beapy.**BEAResults**(*results*, *dataset*)

This class holds the results from the *BEA.get_data* call. It implements a special processing for each dataset, see notes bellow.

Parameters

- **results** (dict) – The dictionary of results for get_data() call obtained from the API

- **dataset** (str) – dataset from which the results were obtained

data(*with_details=False*)

This method returns the DataFrame containing the data. It has an optional argument `with_details`, which determines whether non-data columns are returned as well.

Parameters **with_details** (*False*) – Whether or not to return non-data columns

Returns A DataFrame with a description and name of availible parameters, as well as some other options.

Return type pd.DataFrame

1.3 Changelog

1.3.1 0.5.0

- First release

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

b

Index