ISI Emerging Markets Group

EMIS CEIC

QUICK START GUIDE
PyCEIC PACKAGE

1. PRODUCT OVERVIEW

PyCEIC package is a Python interface to CEIC API with convenience functions for retrieving CEIC macroeconomics data specifically. It is designed for easy and direct access to the time series information. The package can be used together other Python libraries for data manipulation, visualization and analysis. PyCEIC package requires valid credentials for CEIC.

2. PyCEIC FEATURES

PyCEIC key features include:

- Login with valid CEIC username and password
- Detailed search filters, including keywords, regions, topics, indicators, latest information only and etc.
- Easy to use functions for time series retrieving and manipulation
- Dictionaries
- Footnotes
- Details series metadata
- Proxy support
- Session reuse – no need to login and logout the package every time when start the environment

3. PREREQUISITES FOR USING PyCEIC

In order to start using the PyCEIC package, users will need:

- An active CEIC database subscription
- Valid CEIC login credentials – registered email and password
- Access to PyCEIC package
- Python environment
- Access to our hosts:
    - https://api.ceicdata.com/v2
    - https://api.ceicdata.com.cn/v2
- Open port 443

4. INSTALLING AND UPDATING PyCEIC PACKAGE

When using PyCEIC package for the first time, users will first need to install the package.

Installation:

```
pip install --extra-index-url https://downloads.ceicdata.com/python
ceic_api_client
```

Users which are already using PyCEIC package can update to the latest version by updating the package. If user does not have the latest package version, a notification will be receive on login with instructions for update.

Update:

```
pip install --extra-index-url https://downloads.ceicdata.com/python
ceic_api_client --upgrade
```

5. USING CEIC DATA WITH PyCEIC

Accessing CEIC data from Python console:

Python is a general-purpose programming language that is becoming more and more popular for doing data science. PyCEIC package is designed for direct interaction with CEIC Databases in the Python console, but it can be used as good in a file. In order to obtain CEIC data, users need to import the PyCEIC package and sign in with their CEIC subscription credentials:

1. Import PyCEIC package:
   ```
   from ceic_api_client.pyceic import Ceic
   ```
2. Log in to start your session and access your subscribed data and all PyCEIC package functions:
   ```
   Ceic.login("<username>",”<password>”)
   ```
3. If you are using proxy environment, you need to run the *Ceic.set_proxy()* method before you log in.

Once the session starts, quick actions include:

- Data search, incl. advanced search options (see *Ceic.series_search()*)
- Series, data and metadata retrieval (see *Ceic.series()*, *Ceic.series_data()*, *Ceic.series_metadata()*)

```
>>> from ceic_api_client.pyceic import Ceic
>>> Ceic.login("                                    ")
CEIC Password:
WARNING: The PyCEIC package is designed as direct
interaction interface to CEIC macroeconomic data and
any data usage abuse attempt will be recorded.
>>> Ceic.series_metadata(310917301)
{'data': [{'deleted_time_points': None,
           'entity_id': '310917301',
           'layout': [],
           'metadata': {'classification': None,
                        'country': {'id': 'YY', 'name': 'Test'},
                        'end_date': datetime.date(2010, 12, 1),
                        'frequency': {'id': 'M', 'name': 'Monthly'},
                        'id': 310917301,
                        'indicator': None,
                        'key_series': False,
                        'last_change': {'is_infinity': False,
                                        'is_opposite': False,
                                        'value': 9.56057051},
                        'last_update_time': datetime.datetime(2019, 1, 22, 5, 55, 12, tzinfo=tzutc()),
                        'last_value': 2960.48,
                        'multiplier_code': 'MN',
                        'name': '',
                        'new_series': False,
                        'number_of_observations': 543,
                        'period_end': 31,
                        'province': None,
                        'source': {'country_id': None,
                                   'id': 'YY-TEST',
                                   'name': 'NRT SOURCE TEST'},
                        'start_date': datetime.date(1962, 10, 1),
                        'status': {'id': 'B', 'name': 'Rebased'},
                        'unit': {'id': '20039', 'name': 'RMB mn'}},
           'replacements': None,
           'subscribed': False,
           'time_points': None,
           'ui': None}],
 'errors': None}
>>>
```

Accessing CEIC data from Spyder software:

Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. PyCEIC package is designed to work easy in both scripting and console mode of Spyder. In order to obtain CEIC data, users need to import the PyCEIC package and sign in with their CEIC subscription credentials:

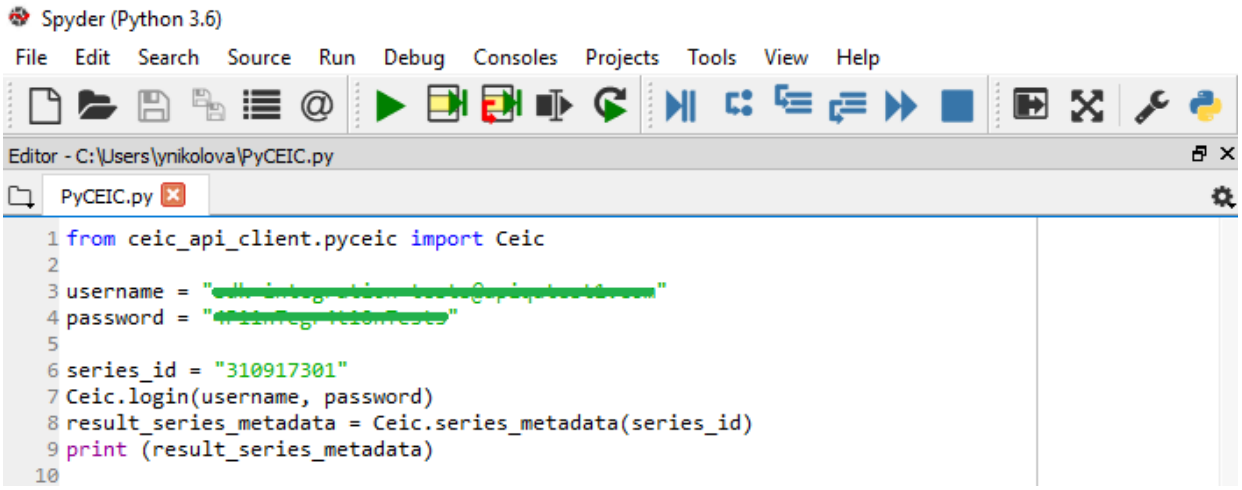
1. Import PyCEIC package:
   `from ceic_api_client.pyceic import Ceic`

2. Log in to start your session and access your subscribed data and all PyCEIC package functions:
   `Ceic.login("<username>","<password>")`

3. If you are using proxy environment, you need to run the *Ceic.set_proxy()* method before you log in.

Once the session starts, quick actions include:

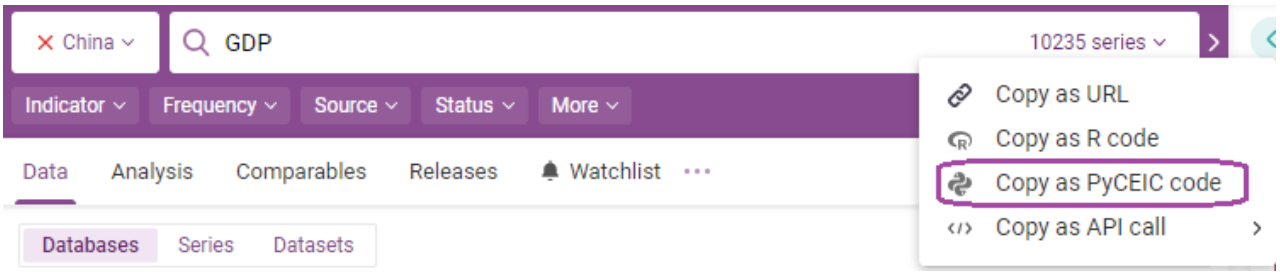
- Data search, incl. advanced search options (see *Ceic.series_search()*)
- Series, data and metadata retrieval (see *Ceic.series()*, *Ceic.series_data()*, *Ceic.series_metadata()*)



Using CDMNext UI for generating PyCEIC code:

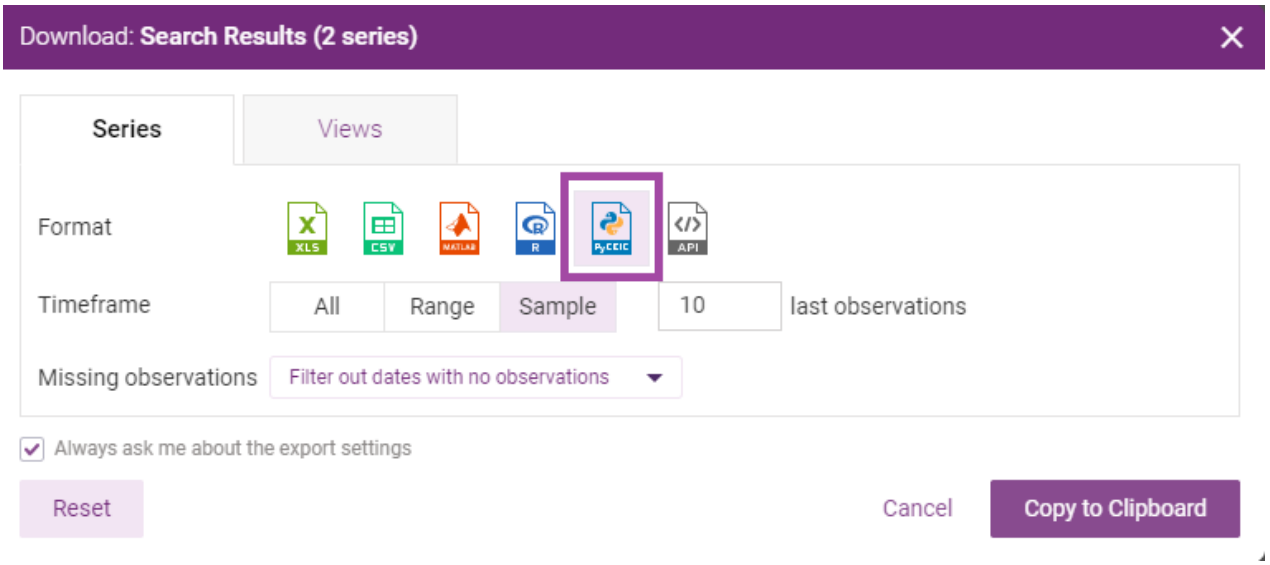
Alternatively, users can follow their usual workflow in CDMNext and use the copy-to-clipboard (CTC) options for PyCEIC available in CDMNext:

CTC for PyCEIC option in CDMNext Search section:



Copy-to-clipboard option for PyCEIC here generated Python code for PyCEIC package, which can be pasted in Python environment and will produce the same search, including all filters used.

CTC for PyCEIC option in CDMNext Download section:



Copy-to-clipboard option for PyCEIC here generated Python code for PyCEIC package, which can be pasted in Python environment and will retrieve the selected series data with defined options. User can select the time period for retrieving series time points and how to handle periods with missing values.

6. EXAMPLES

Example 1: Search for Inflation headline series from Eurostat for 3 European counties – France, Germany and Italy. Get the basic series information.

```
from ceic_api_client.pyceic import Ceic
Ceic.login("username","password")

ceic_countries=[]
ceic_sources=[]
keywords='Harmonized Index of Consumer Price'

#Find the country codes for France, Germany and Italy:
countries_result = Ceic.geo()
for country in countries_result.data:
    if country.title in ('France','Germany','Italy'):
        ceic_countries.append(country.id)
#Find the Eurostat source:
sources_result = Ceic.sources()
for source in sources_result.data:
    if source.name in ('Eurostat'):
        ceic_sources.append(source.id)

# Search for active series with "Harmonized Index of Consumer Price" in their
name, from Eurostat for France, Germany and Italy
search_series_result = Ceic.search(keyword=keywords, name_only=True, status='T',
geo=ceic_countries, source=ceic_sources)

# For every such series, display series id, name and latest data (last timepoint)
for series_list in search_series_result:
    for series in series_list.data.items:
        series_id=series.metadata.id
        series_name=series.metadata.name
        series_tp=series.metadata.last_update_time
        series_tp_value=series.metadata.last_value
        print(series_id, series_name, series_tp, series_tp_value)
```

Example 2: Plot Federal's Fund information since the beginning of the year. The affected series are: "Federal Funds Rate", "Federal Funds Target Rates Upper" and "Federal Funds Target Rates Lower".

```
from ceic_api_client.pyceic import Ceic
import pandas
import matplotlib.pyplot as plot

Ceic.login("username ","password")

keywords='Federal Funds'
series_ids=[]

# Search for daily series for US Federal Funds.
search_series_result = Ceic.search(keyword=keywords, name_only=True, status='T',
geo=["3100"], database=["DAILY"])
```

```
# For every such series, get it's ID
for series result list in search series result:
    for series result in series result list.data.items:
        series_ids.append(series_result.metadata.id)

# For the finded series Ids, get the data since the begging of the year
series_result = Ceic.series(series_ids,start_date='2019-01-01')

# Display the timepoint values for the 3 Federal Funds series in one graphic
for series in series result.data:
    time points dict = dict((tp.date, tp.value) for tp in series.time_points)
    series = pandas.Series(time_points_dict)
    plot.plot(series)
plot.show()
```

Example 3: Check the coverage of CEIC Financial Market sector in Thailand.

```
from ceic api client.pyceic import Ceic
Ceic.login("username","password")

table_list=[]

#Find the topic under which CEIC store information about Thailand:
topics = Ceic.layout database_topics('GLOBAL')
for topic in topics.data:
    if topic.metadata.name in ('Thailand'):
        ceic_topic=topic.metadata.id

#Find the topic under which CEIC store information about Financial Market of
Thailand:
sections = Ceic.layout topic sections(ceic_topic)
for section in sections.data:
    if 'Financial Market' in section.metadata.name:
        ceic_section=section.metadata.id

# Get the coverage list of tables for Financial Market of Thailand:
tables = Ceic.layout section_tables(ceic_section)
for table in tables.data:
    table list.append(table.metadata.name)
print(table_list)
```

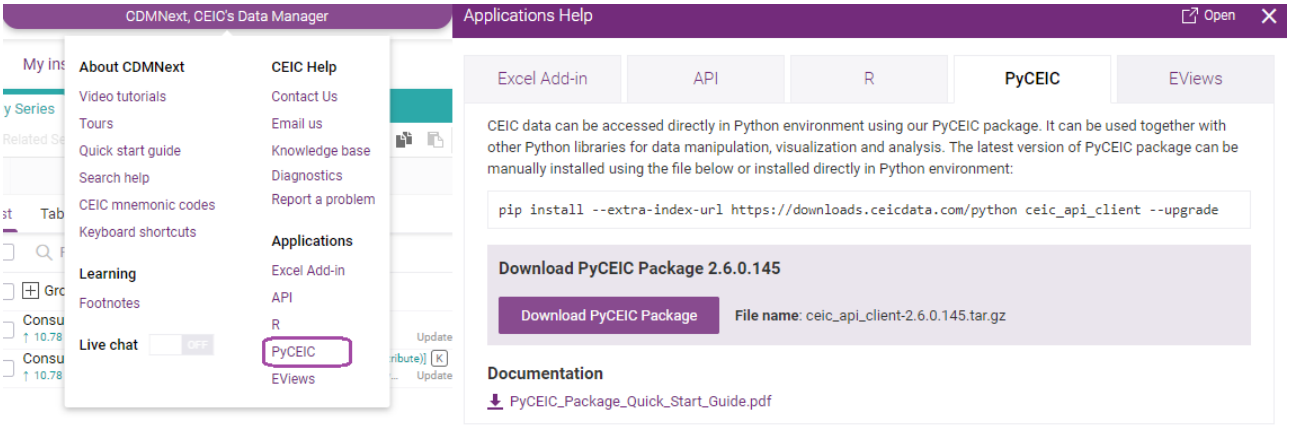
Example 4: Get related breakdowns pertaining to series by your choice.

```
from ceic api client.pyceic import Ceic
Ceic.login("username","password")

table_list=[]

# We are interested by series 378259897 "CPI: Food: Fish & Seafood: Fish Paste"
series_id='378259897'

# Get series parent table
series layouts list = Ceic.series layouts(series_id)
for series layout in series layouts list.data:
    for layout in series layout.layout:
        table_list.append(layout.table.id)

# Get series under the same table
for table in table list:
    seriess list = Ceic.layout_table_series(table)
    print(seriess_list)
```

7. SUPPORT

PyCEIC package was developed by CEIC data. For any questions or inquiries, please contact your CEIC Account Manager or contact our Support team at helpdesk@ceicdata.com

For technical documentation details, please refer our CDMNext Help section.



APPENDIX: PyCEIC PACKAGE TECHNICAL INFORMATION

PyCEIC

Title: PyCEIC package
Package: ceic_api_client
Module: pyceic
Class: Ceic
Author: ***CEIC – An ISI Emerging Markets Group Company***
Maintainer: ***CEIC*** <helpdesk@ceicdata.com>
Description: The PyCEIC package allows to securely connect and retrieve CEIC data in Python console and Python-based softwares for data analysis, to search time series, import series and metadata and update data points in real time.
Encoding: UTF-8

PyCEIC functions:

__init__

Description:
Constructor for the CEIC SDK Interface.

Usage:
```
  init  (self, username=None, password=None, proxy_url=None, proxy_username=None,
proxy_password=None)
```

Arguments:
- username: Login username
- password: Login password
- proxy_url: Proxy URL
- proxy_username: Proxy username
- proxy_password: Proxy password

Examples:
```
ceic = Ceic()
ceic = Ceic("my_username", "my_password")
```

Ceic.set_server()

Description:
Changes the API server address.

Usage:
```
set_server(server)
```

Arguments:
- server: Server URL

Examples:
```
Ceic.set_server("http://some-other-server.com/my/api/path")
```

Ceic.get_server()

Description:
Gets the address of the currently set API server.

Usage:
```
get_server()
```

Arguments:
None

Examples:
```
current_server = Ceic.get_server()
```

Ceic.set_proxy()

Description:
Sets a proxy connection.

Usage:
```
set_proxy(proxy_url=None, proxy_username=None, proxy_password=None)
```

Arguments:
- proxy_url: Proxy URL
- proxy_username: Proxy username
- proxy_password: Proxy password

Examples:
```
Ceic.set proxy("https://my proxy username:my proxy password@my-proxy-name.com")
Ceic.set proxy("http://my-proxy-server.com", "my_proxy_username",
"my_proxy_password")
```

Ceic.login()

Description:
Attempts to login and create a login session.

Usage:
```
login(username=None, password=None)
```

Arguments:
- username: Login username
- password: Login password

Examples:
```
Ceic.login("my_username", "my_password")
```

Ceic.logout()

Description:
Attempts to logout and delete the saved session.

Usage:
```
logout()
```

Arguments:
None

Examples:
```
Ceic.logout()
```

Ceic.series()

Description:
Gets full series data. Result contains both metadata and time-points data. Returns an iterable object which contains result data. Each iteration will return an object with up to 20 result objects.

Usage:
```
series(series_id, **kwargs)
```

Arguments:

- series_id: A single series id can be passed as a string, an integer, or a list. Multiple series ids can be passed as a list only. You can pass series IDs, series SR codes or insight series IDs.
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en – English
  - zh – Chinese
  - ru – Russian
  - id – Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.
- count: Limit the amount of latest time-points returned, by the number specified.
- start_date: Limits the start date after which the time-points will be returned. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- end_date: Limits the end date before which the time-points will be returned. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- updated_after: Returns only the updated time-points after the date specified. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- blank_observations: If it's set to true, empty time-points will be returned
- with_replacements_metadata: If it is `true` result will contain replacements metadata not only list of id`s.
- forecast_only: If it is `true` result will only contain series with forecast
- with_release_only: If it is `true` result will only contain series with released schedule
- with_replacements_only: If it is `true` result will only contain series with suggestions

Examples:
```
Ceic.series('210438802')
Ceic.series(['210438802', '370007807'])
Ceic.series(['SR4478574', '370007807'])
Ceic.series('4d8320c9-060f-484b-ba4b-2adff3eb6f91')
Ceic.series('370007807', start date=datetime.date(2017, 8, 29))
for series result in Ceic.series(['list_with_more_than_20_series']):
    for series in series result.data:
            # Do something with series
```

Ceic.series_metadata()

Description:
Gets series metadata only. Returns an iterable object which contains result data. Each iteration will return an object with up to 20 result objects.

Usage:
```
series_metadata(series_id, **kwargs)
```

Arguments:

- series_id: A single series id can be passed as a string, an integer, or a list. Multiple series ids can be passed as a list only. You can pass series IDs, series SR codes or insight series IDs.
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en – English
  - zh – Chinese
  - ru – Russian
  - id – Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.
- with_replacements_metadata: If it is `true` result will contain replacements metadata not only list of id`s.
- forecast_only: If it is `true` result will only contain series with forecast
- with_release_only: If it is `true` result will only contain series with released schedule
- with_replacements_only: If it is `true` result will only contain series with suggestions

Examples:
```
Ceic.series metadata('210438802')
Ceic.series metadata(['210438802', '370007807'])
Ceic.series metadata (['SR4478574', '370007807'])
Ceic.series metadata ('4d8320c9-060f-484b-ba4b-2adff3eb6f91')
for series result in Ceic.series metadata(['list_with_more_than_20_series']):
    for series in series result.data:
            # Do something with series
```

Ceic.series_data()

Description:
Gets series time-points only. Returns an iterable object which contains result data. Each iteration will return an object with up to 20 result objects.

Usage:
```
series_data(series_id, **kwargs)
```

Arguments:

- series_id: A single series id can be passed as a string, an integer, or a list. Multiple series ids can be passed as a list only. You can pass series IDs, series SR codes or insight series IDs.
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en – English
  - zh – Chinese
  - ru – Russian
  - id – Indonesian
  - jp – Japanese
- with_model_information: If set to `true` returns the model names as part of the response.
- count: Limit the amount of latest time-points returned, by the number specified.
- start_date: Limits the start date after which the time-points will be returned. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- end_date: Limits the end date before which the time-points will be returned. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- updated_after: Returns only the updated time-points after the date specified. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- blank_observations: If it's set to true, empty time-points will be returned

Examples:

```
Ceic.series data('210438802')
Ceic.series data(['210438802', '370007807'])
Ceic.series data (['SR4478574', '370007807'])
Ceic.series data ('4d8320c9-060f-484b-ba4b-2adff3eb6f91')
Ceic.series data('370007807', start date=datetime.date(2017, 8, 29))
for series result in Ceic.series data(['list_with_more_than_20_series']):
    for series in series result.data:
            # Do something with series
```

Ceic.series_layouts()

Description:
Gets series layout information only. Returns an iterable object which contains result data. Each iteration will return an object with up to 20 result objects.

Usage:

```
series_layouts(series_id, **kwargs)
```

Arguments:

- series_id: A single series id can be passed as a string, an integer, or a list. Multiple series ids can be passed as a list only.
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en – English
  - zh – Chinese
  - ru – Russian
  - id – Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.
- count: Limit the amount of latest time-points returned, by the number specified.
- start_date: Limits the start date after which the time-points will be returned. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- end_date: Limits the end date before which the time-points will be returned. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- updated_after: Returns only the updated time-points after the date specified. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- blank_observations: If it's set to true, empty time-points will be returned
- with_replacements_metadata: If it is `true` result will contain replacements metadata not only list of id`s.

Examples:

```
Ceic.series layouts('210438802')
Ceic.series layouts(['210438802', '370007807'])
for series result in Ceic.series layouts(['list_with_more_than_20_series']):
    for series in series result.data:
            # Do something with series
```

Ceic.series_statistics()

Description:
Returns list of series statistics. Returns series containing statistics only. Accepts one or more Series IDs or Series Codes.

Usage:

```
series_statistics(id, **kwargs)
```

Arguments:

- id: Series ID or Series Code to return. Accepts only one series ID or series Code. (Required)
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en – English
  - zh – Chinese
  - ru – Russian
  - id – Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.

Examples:
```
Ceic.series statistics('210438802')
Ceic.series_ statistics (['210438802', '370007807'])
```

Ceic.series_releases()

Description:
Returns list of releases for a single series id. Lists the release schedule for a single series id.

Usage:
```
series_releases(id, **kwargs)
```

Arguments:
- id: A single CEIC series ID.
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en - English
  - zh - Chinese
  - ru - Russian
  - id - Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.
- release_date_from: Will return releases with first observation before release_date_from.
- release_date_to: Will return releases with last observation after release_date_to.
- release_status: List of comma separated release statuses to return. Default is "Released". Other options are "Pending" and "Delayed".
- limit: Number of records to return in the range 1 - 100. Default is 100.
- offset: The offset from which the records will be returned.

Examples:
```
Ceic.series releases(262498601)
Ceic.series_releases(262498601, release_status = ['Released','Pending'])
```

Ceic.search()

Description:
Allows searching for series by a keyword and additional filtering criteria. Each filtering criteria accepts one or more, comma separated code values. See Dictionary functions for details on how to retrieve a specific filter code. The multi-dimensional filters include the economic classification and indicators (defined by CEIC database structure), region/country, frequency, unit, source, status and observation date. Returns an iterable object which contains result data. Each iteration will return an object with up to 100 result objects.

Usage:
```
search(keyword=None, **kwargs)
```

Arguments:
- keyword: Search term. One or more keywords. May contain special words further controlling the search results. Keyword search tips:

  - Retail Sales - Show series with both keywords while the sequence of keywords is irrelevant. Equivalent to search Sales Retail
  - Retail AND Sales - Show results: series with terms of Retail AND Sales, regardless of the sequence. E. g. Retail Sales, Automobile Sales Retail.
  - Retail;Sales - Show series with either keyword and series with both keywords while the sequence of keywords is irrelevant, equivalent to search: Sales;Retail
  - Retail OR Sales - Show results: series with terms of Retail OR Sales, regardless of the sequence. E. g. Retail Sales, Retail Trade, Sales Price, Motor Vehicle Sales
  - Retail NOT Sales - Narrow a search by excluding specific terms while the sequence of keywords is relevant. Show results: series with terms that include Retail, but NOT Sales. E. g. Retail Trade, Retail Price, Retail Bank
  - Retail Sales NOT (Hong Kong) - Narrow a search by excluding a set of words in parentheses while the sequence of keywords in parentheses is irrelevant, equivalent to search: Retail Sales NOT (Hong Kong). Show results: series with terms that include Retail Sales, but NOT Hong Kong, such as Retail Sales YoY: China, Retail Sales YoY: United States
- name_only: This filter related with the `keyword` filter. If it's `true` keyword search will be searched only in series name instead of all series attributes.
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en – English
  - zh – Chinese
  - ru – Russian
  - id – Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.
- limit: Number of records to return in the range 1 - 100. Default is 100.
- offset: The offset from which the records will be returned. Used to get the next set of records when the limit is reached.
- database: Database filter. One or more comma separated database code values. Use `/dictionary/databases` to get an up to date list of available databases.
  - WORLD - *World Trend Plus*
  - GLOBAL - *Global Database*
  - CEICGLBKS - *Global Key Series Database*
  - PMI - *Markit Purchasing Managers' Index*
  - DAILY - *Daily Database*
  - BRAZIL - *Brazil Premium Database*
  - RUSSIA - *Russia Premium Database*
  - INDIA - *India Premium Database*
  - INDONESIA - *Indonesia Premium Database*
  - CN - *China Premium Database*
  - OECD-MEI - *OECD - Main Economic Indicators*
  - OECD-EO - *OECD - Economic Outlook*
  - OECD-PROD - *OECD - Productivity*
- frequency: Frequency filter. One or more comma separated frequency code values.
  - D - Daily
  - W - Weekly
  - M - Monthly
  - Q - Quarterly
  - S - Semi-annual
  - Y – Annual
  - Z – Quinquennial
  - T - Decadal
- geo: Geo filter. One or more comma separated geo id values. See related Dictionary function to get the full list of accepted geo ids.
- source: Source filter. One or more comma separated source code values. See related Dictionary function to get the full list of accepted sources.

- unit: Unit filter. One or more comma separated unit code values. See related Dictionary function to get the full list of accepted units.
- indicator: Indicator filter. One or more comma separated indicator code values. See related Dictionary function to get full list of accepted indicators.
- region: Region filter. One or more comma separated region code values. See related Dictionary function to get the full list of accepted regions.
- subscribed_only: Show only results for subscribed series when set to `true`. By default show results for all the series found.
- key_only: Show only 'key' series when set to `true`.
- new_only: Show only series created less than 1 month ago when set to `true`.
- start_date_before: Will return series with first observation before `start_date_before`. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- end_date_after: Will return series with last observation after `end_date_after`. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- created_after: Will return series created after `created_after` date. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- updated_after: Will return series last time updated after `updated_after` date. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- status: Series status filter. One or more comma separated status code values. When not explicitly set, defaults to T.
  - T - Active
  - C - Discontinued
  - B - Rebased
- topic: Topic filter. One or more comma separated topic code values.
- section: Section filter. One or more comma separated section code values.
- table: Table filter. One or more comma separated table code values.
- order: Sort order. Default is `relevance`.
- direction: Sort order direction. Default is `asc`. Accepted values: `asc` - ascending `desc` - descending
- filter_id: Filter ID used to define a subset of data over which the search will be executed. When combined with additional search criterion, the result will be an intersection of both.
- with_replacements_metadata: If it is `true` result will contain replacements metadata not only list of id`s
- forecast_only: If it is `true` result will only contain series with forecast
- with_release_only: If it is `true` result will only contain series with released schedule
- with_replacements_only: If it is `true` result will only contain series with suggestions
- facet: List of facets to return

Examples:

```
Ceic.search('China', frequency=['Y','M'], order='popular')
Ceic.search('Retail AND Sales', name only=True, geo='0')
for search_result in Ceic.search('GDP'):
      for series in search result.data:
            # Do something with series
```

Ceic.dictionaries()

Description:
Full dictionary list. Returns all the available dictionaries.

Usage:

```
dictionaries(**kwargs)
```

Arguments:

- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en – English

  - zh – Chinese
  - ru – Russian
  - id – Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.

Examples:
```
Ceic.dictionaries()
```

Ceic.indicators()

Description:
Returns full list of supported indicators, their codes and the related top level classifications.

Usage:
```
indicators(**kwargs)
```

Arguments:
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en – English
  - zh – Chinese
  - ru – Russian
  - id – Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.

Examples:
```
Ceic.indicators()
```

Ceic.classifications()

Description:
Returns full list of supported top level classifications and their codes.

Usage:
```
classifications(**kwargs)
```

Arguments:
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en – English
  - zh – Chinese
  - ru – Russian
  - id – Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.

Examples:
```
Ceic.classifications()
```

Ceic.classification_indicators()

Description:
Returns full list of indicators for specific classification.

Usage:
```
classification_indicators(classification_id, **kwargs)
```

Arguments:

- classification_id: The ID of the specific classification
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en – English
  - zh – Chinese
  - ru – Russian
  - id – Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.

Examples:

```
Ceic.classification_indicators('200019')
```

Ceic.country_sources()

Description:
Returns full list of sources for a specific country.

Usage:

```
country_sources(country_id, **kwargs)
```

Arguments:

- country_id: The ID of the specific country
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en - English
  - zh - Chinese
  - ru - Russian
  - id - Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.

Examples:

```
Ceic.country_sources('DE')
```

Ceic.geo()

Description:
Returns full list of geo entities and their codes.

Usage:

```
geo(**kwargs)
```

Arguments:

- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en - English
  - zh - Chinese
  - ru - Russian
  - id - Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.

Examples:

```
Ceic.geo()
```

Ceic.regions()

Description:
Returns full list of supported regions and their codes.

Usage:
```
regions(**kwargs)
```

Arguments:
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en - English
  - zh - Chinese
  - ru - Russian
  - id - Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.

Examples:
```
Ceic.regions()
```

Ceic.sources()

Description:
Returns full list of supported sources and their codes.

Usage:
```
sources(**kwargs)
```

Arguments:
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en - English
  - zh - Chinese
  - ru - Russian
  - id - Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.

Examples:
```
Ceic.sources()
```

Ceic.units()

Description:
Returns full list of supported units and their codes.

Usage:
```
units(**kwargs)
```

Arguments:
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en - English
  - zh - Chinese
  - ru - Russian
  - id - Indonesian
  - jp – Japanese
- with_model_information: If set to `true` returns the model names as part of the response.

Examples:
```
Ceic.units()
```

Ceic.frequencies()

Description:
Returns full list of supported frequencies and their codes.

Usage:
```
frequencies(**kwargs)
```

Arguments:
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en - English
  - zh - Chinese
  - ru - Russian
  - id - Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.

Examples:
```
Ceic.frequencies()
```

Ceic.statuses()

Description:
Returns full list of supported statuses and their codes.

Usage:
```
statuses(**kwargs)
```

Arguments:
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en - English
  - zh - Chinese
  - ru - Russian
  - id - Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.

Examples:
```
Ceic.statuses()
```

Ceic.layout_databases()

Description:
Returns list of layout databases. This is the top level from the layout hierarchy.

Usage:
```
layout_databases(**kwargs)
```

Arguments:
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en – English
  - zh – Chinese

  - ru – Russian
  - id – Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.
- frequency: Frequency filter. One or more comma separated frequency code values.
  - D - Daily
  - W - Weekly
  - M - Monthly
  - Q - Quarterly
  - S - Semi-annual
  - Y – Annual
  - Z – Quinquennial
  - T - Decadal
- geo: Geo filter. One or more comma separated geo id values. See related Dictionary function to get the full list of accepted geo ids.
- source: Source filter. One or more comma separated source code values. See related Dictionary function to get the full list of accepted sources.
- unit: Unit filter. One or more comma separated unit code values. See related Dictionary function to get the full list of accepted units.
- indicator: Indicator filter. One or more comma separated indicator code values. See related Dictionary function to get full list of accepted indicators.
- region: Region filter. One or more comma separated region code values. See related Dictionary function to get the full list of accepted regions.
- subscribed_only: Show only results for subscribed series when set to `true`. By default show results for all the series found.
- key_only: Show only 'key' series when set to `true`.
- new_only: Show only series created less than 1 month ago when set to `true`.
- start_date_before: Will return series with first observation before `start_date_before`. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- end_date_after: Will return series with last observation after `end_date_after`. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- created_after: Will return series created after `created_after` date. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- updated_after: Will return series last time updated after `updated_after` date. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- status: Series status filter. One or more comma separated status code values. When not explicitly set, defaults to T.
  - T - Active
  - C - Discontinued
  - B - Rebased
- filter_id: Filter ID used to define a subset of data over which the search will be executed. When combined with additional search criterion, the result will be an intersection of both.

Examples:

```
Ceic.layout_databases()
Ceic.layout_databases(frequency=['W','M'])
```

Ceic.layout_database_topics()

Description:
Returns list of topics for a specific database.

Usage:
layout_database_topics(database_id, **kwargs)

Arguments:

- database_id: The database ID
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en - English
  - zh - Chinese
  - ru - Russian
  - id - Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.
- frequency: Frequency filter. One or more comma separated frequency code values.
  - D - Daily
  - W - Weekly
  - M - Monthly
  - Q - Quarterly
  - S - Semi-annual
  - Y – Annual
  - Z – Quinquennial
  - T - Decadal
- geo: Geo filter. One or more comma separated geo id values. See related Dictionary function to get the full list of accepted geo ids.
- source: Source filter. One or more comma separated source code values. See related Dictionary function to get the full list of accepted sources.
- unit: Unit filter. One or more comma separated unit code values. See related Dictionary function to get the full list of accepted units.
- indicator: Indicator filter. One or more comma separated indicator code values. See related Dictionary function to get full list of accepted indicators.
- region: Region filter. One or more comma separated region code values. See related Dictionary function to get the full list of accepted regions.
- subscribed_only: Show only results for subscribed series when set to `true`. By default show results for all the series found.
- key_only: Show only 'key' series when set to `true`.
- new_only: Show only series created less than 1 month ago when set to `true`.
- start_date_before: Will return series with first observation before `start_date_before`. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- end_date_after: Will return series with last observation after `end_date_after`. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- created_after: Will return series created after `created_after` date. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- updated_after: Will return series last time updated after `updated_after` date. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- status: Series status filter. One or more comma separated status code values. When not explicitly set, defaults to T.
  - T - Active
  - C - Discontinued
  - B - Rebased
- filter_id: Filter ID used to define a subset of data over which the search will be executed. When combined with additional search criterion, the result will be an intersection of both.

Examples:

```
Ceic.layout_database_topics('GLOBAL')
Ceic.layout_database_topics('GLOBAL', with_model_information=True)
```

Ceic.layout_topic_sections()

Description:
Returns list of sections for a specific topic.

Usage:
```
layout_topic_sections(topic_id, **kwargs)
```

Arguments:

- topic_id: The topic ID
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en - English
  - zh - Chinese
  - ru - Russian
  - id - Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.
- frequency: Frequency filter. One or more comma separated frequency code values.
  - D - Daily
  - W - Weekly
  - M - Monthly
  - Q - Quarterly
  - S - Semi-annual
  - Y – Annual
  - Z – Quinquennial
  - T - Decadal
- geo: Geo filter. One or more comma separated geo id values. See related Dictionary function to get the full list of accepted geo ids.
- source: Source filter. One or more comma separated source code values. See related Dictionary function to get the full list of accepted sources.
- unit: Unit filter. One or more comma separated unit code values. See related Dictionary function to get the full list of accepted units.
- indicator: Indicator filter. One or more comma separated indicator code values. See related Dictionary function to get full list of accepted indicators.
- region: Region filter. One or more comma separated region code values. See related Dictionary function to get the full list of accepted regions.
- subscribed_only: Show only results for subscribed series when set to `true`. By default show results for all the series found.
- key_only: Show only 'key' series when set to `true`.
- new_only: Show only series created less than 1 month ago when set to `true`.
- start_date_before: Will return series with first observation before `start_date_before`. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- end_date_after: Will return series with last observation after `end_date_after`. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- created_after: Will return series created after `created_after` date. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- updated_after: Will return series last time updated after `updated_after` date. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- status: Series status filter. One or more comma separated status code values. When not explicitly set, defaults to T.
  - T - Active
  - C - Discontinued
  - B - Rebased
- filter_id: Filter ID used to define a subset of data over which the search will be executed. When combined with additional search criterion, the result will be an intersection of both.

Examples:
```
Ceic.layout_topic_sections('TP23149')
Ceic.layout_topic_sections('TP23149', frequency=['W','M'])
```

Ceic.layout_section_tables()

Description:
Returns list of tables for a specific section.

Usage:
layout_section_tables(section_id, **kwargs)

Arguments:

- section_id: The section ID
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en - English
  - zh - Chinese
  - ru - Russian
  - id - Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.
- frequency: Frequency filter. One or more comma separated frequency code values.
  - D - Daily
  - W - Weekly
  - M - Monthly
  - Q - Quarterly
  - S - Semi-annual
  - Y – Annual
  - Z – Quinquennial
  - T - Decadal
- geo: Geo filter. One or more comma separated geo id values. See related Dictionary function to get the full list of accepted geo ids.
- source: Source filter. One or more comma separated source code values. See related Dictionary function to get the full list of accepted sources.
- unit: Unit filter. One or more comma separated unit code values. See related Dictionary function to get the full list of accepted units.
- indicator: Indicator filter. One or more comma separated indicator code values. See related Dictionary function to get full list of accepted indicators.
- region: Region filter. One or more comma separated region code values. See related Dictionary function to get the full list of accepted regions.
- subscribed_only: Show only results for subscribed series when set to `true`. By default show results for all the series found.
- key_only: Show only 'key' series when set to `true`.
- new_only: Show only series created less than 1 month ago when set to `true`.
- start_date_before: Will return series with first observation before `start_date_before`. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- end_date_after: Will return series with last observation after `end_date_after`. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- created_after: Will return series created after `created_after` date. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- updated_after: Will return series last time updated after `updated_after` date. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- status: Series status filter. One or more comma separated status code values. When not explicitly set, defaults to T.
  - T - Active
  - C - Discontinued
  - B - Rebased
- filter_id: Filter ID used to define a subset of data over which the search will be executed. When combined with additional search criterion, the result will be an intersection of both.

Examples:
```
Ceic.layout_section_tables('SC23158')
Ceic.layout_section_tables('SC23158', frequency=['W','M'])
```

Ceic.layout_table_series()

Description:
Returns list of series inside of a specific table

Usage:
```
layout_table_series(table_id, **kwargs)
```

Arguments:

- table_id: The section ID
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en - English
  - zh - Chinese
  - ru - Russian
  - id - Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.
- frequency: Frequency filter. One or more comma separated frequency code values.
  - D - Daily
  - W - Weekly
  - M - Monthly
  - Q - Quarterly
  - S - Semi-annual
  - Y – Annual
  - Z – Quinquennial
  - T - Decadal
- geo: Geo filter. One or more comma separated geo id values. See related Dictionary function to get the full list of accepted geo ids.
- source: Source filter. One or more comma separated source code values. See related Dictionary function to get the full list of accepted sources.
- unit: Unit filter. One or more comma separated unit code values. See related Dictionary function to get the full list of accepted units.
- indicator: Indicator filter. One or more comma separated indicator code values. See related Dictionary function to get full list of accepted indicators.
- region: Region filter. One or more comma separated region code values. See related Dictionary function to get the full list of accepted regions.
- subscribed_only: Show only results for subscribed series when set to `true`. By default show results for all the series found.
- key_only: Show only 'key' series when set to `true`.
- new_only: Show only series created less than 1 month ago when set to `true`.
- start_date_before: Will return series with first observation before `start_date_before`. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- end_date_after: Will return series with last observation after `end_date_after`. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- created_after: Will return series created after `created_after` date. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- updated_after: Will return series last time updated after `updated_after` date. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- status: Series status filter. One or more comma separated status code values. When not explicitly set, defaults to T.
  - T - Active

  - C - Discontinued
  - B - Rebased
- filter_id: Filter ID used to define a subset of data over which the search will be executed. When combined with additional search criterion, the result will be an intersection of both.

Examples:
```
Ceic.layout table series('TB2863517')
Ceic.layout_table_series('TB2863517', lang='zh')
```

Ceic.footnotes()

Description:
Download footnotes for a database, topic, section or table. The function is also going to download additional resource files in a folder named "resources".

Usage:
```
footnotes(**kwargs)
```

Arguments:
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en – English
  - zh – Chinese
  - ru – Russian
  - id – Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.
- node_code: Node Code - Database ID, Topic ID, Section ID or Table ID
- download_path: Directory path where you want to download the files.

Examples:
```
Ceic.footnotes("TP43532")
Ceic.footnotes(TP43532", download_path = "C:\\Users\\User123\\footnotes)
```

Ceic.insights()

Description:
Returns full list of CDMNext user created insights.

Usage:
```
insights(**kwargs)
```

Arguments:
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en – English
  - zh – Chinese
  - ru – Russian
  - id – Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.

Examples:
```
Ceic.insights()
Ceic.insights(with_model_information=True)
```

Ceic.search_insights()

Description:

Search for insights. Those could be user created, shared, or CEIC created ones. Returns an iterable object which contains result data. Each object can contain up to 20 result objects.

Usage:
```
search_insights(**kwargs)
```

Arguments:
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en - English
  - zh - Chinese
  - ru - Russian
  - id - Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.
- group: Insights group. Default is `my`. Possible values:
  - favorite
  - my
  - analytics
  - shared
  - recent
  - all
  - gallery
  - data_talk
  - wpic_platinum
- limit: Number of records to return
- offset: The offset from which the records will be returned
- order: Sort order. Possible values:
  - name
  - edit_date
  - open_date
- direction: Sort order direction. Possible values:
  - asc
  - desc
- tags: List of insight tags to search by tag
- categories: List of insights categories to search by category

Examples:
```
Ceic.search insights()
Ceic.search insights(order='name')
for search result in Ceic.search insights(order='name'):
      for insight in search result.data:
            # Do something with insight
```

Ceic.insights_categories()

Description:
Returns list of insight categories. To be used wtih group filters \"favorite\", \"my\", \"shared\", \"recent\", all\".

Usage:
```
insights_categories(**kwargs)
```

Arguments:
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en – English
  - zh – Chinese
  - ru – Russian

  - id – Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.

Examples:
```
Ceic.insights categories()
Ceic.insights_categories(lang='jp')
```

Ceic.gallery_insights_categories()

Description:
Returns list of gallery categories. To be used with group filters \"analytics\" and \"gallery\".

Usage:
```
gallery_insights_categories(**kwargs)
```

Arguments:
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en – English
  - zh – Chinese
  - ru – Russian
  - id – Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.

Examples:
```
Ceic.gallery insights categories()
Ceic.gallery_insights_categories(lang='jp')
```

Ceic.insight()

Description:
Returns information about a specified insight.

Usage:
```
insight(insight_id, **kwargs)
```

Arguments:
- insight_id: The insight ID
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en – English
  - zh – Chinese
  - ru – Russian
  - id – Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.

Examples:
```
Ceic.insight('MY INSIGHT ID')
Ceic.insight('MY_INSIGHT_ID', lang='jp')
```

Ceic.download_insight()

Description:
Returns one or more links to the insight report. When the report generation takes too much time to complete in a timely manner, returns HTTP 408. In this case the request have to be repeated after a minute. Once the report is generated, consecutive requests are returned immediately. Each successful response returns one

or more download links that expires in 5 minutes. The client application consuming the API shall download the file within this period or send additional request to the API.

Usage:
```
download_insight(insight_id, file_format, **kwargs)
```

Arguments:
- insight_id: The insight ID
- file_format: Insight report file format. Possible values:
  - xlsx
  - pdf
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en – English
  - zh – Chinese
  - ru – Russian
  - id – Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.

Examples:
```
Ceic.download_insight('MY_INSIGHT_ID', 'pdf')
```

Ceic.insight_series()

Description:
Returns all series from the specified insight(s), including all time-points and metadata, as well as their layout in the insight context in terms of grouping and separators. Returns an iterable object containing insight series result data.

Usage:
```
insight_series(insight_id, **kwargs)
```

Arguments:
- insight_id: The insight ID
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en – English
  - zh – Chinese
  - ru – Russian
  - id – Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.
- count: Limit the amount of latest time-points returned, by the number specified.
- start_date: Limits the start date after which the time-points will be returned. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- end_date: Limits the end date before which the time-points will be returned. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- updated_after: Returns only the updated time-points after the date specified. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- blank_observations: If it's set to true, empty time-points will be returned.
- with_replacements_metadata: If it is `true` result will contain replacements metadata not only list of id`s.
- forecast_only: If it is `true` result will only contain series with forecast
- with_release_only: If it is `true` result will only contain series with released schedule
- with_replacements_only: If it is `true` result will only contain series with suggestions
- limit: Number of records to return in the range 1 - 100. Default is 100.

- offset: The offset from which the records will be returned.

Examples:
```
Ceic.insight series('MY INSIGHT ID')
Ceic.insight series('MY INSIGHT ID', blank observations=True)
for series result in Ceic.insight series('MY INSIGHT_ID'):
      for insight series in series result.data:
            # Do something with series
```

Ceic.insight_series_data()

Description:
Returns all series time-points from the specified insight series. Returns an iterable object containing insight series time-points result data. Each object can contain up to 20 result objects.

Usage:
```
insight_series_data(insight_id, **kwargs)
```

Arguments:
- insight_id: The insight ID
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en - English
  - zh - Chinese
  - ru - Russian
  - id - Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.
- count: Limit the amount of latest time-points returned, by the number specified.
- start_date: Limits the start date after which the time-points will be returned. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- end_date: Limits the end date before which the time-points will be returned. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- updated_after: Returns only the updated time-points after the date specified. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- blank_observations: If it's set to true, empty time-points will be returned.
- limit: Number of records to return in the range 1 - 100. Default is 100.
- offset: The offset from which the records will be returned.

Examples:
```
Ceic.insight series data('MY INSIGHT ID')
Ceic.insight series data('MY INSIGHT ID', with replacements metadata=True)
for series result in Ceic.insight series data('MY_INSIGHT_ID'):
      for insight series in series result.data:
            # Do something with series
```

Ceic.insight_series_metadata()

Description:
Returns all series metadata from the specified insight(s), as well as their layout in the insight context in terms of grouping and separators. Returns an iterable object containing insight series time-points result data. Each object can contain up to 20 result objects.

Usage:
```
insight_series_metadata(insight_id, **kwargs)
```

Arguments:
- insight_id: The insight ID

- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en - English
  - zh - Chinese
  - ru - Russian
  - id - Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.
- with_replacements_metadata: If it is `true` result will contain replacements metadata not only list of id`s.
- forecast_only: If it is `true` result will only contain series with forecast
- with_release_only: If it is `true` result will only contain series with released schedule
- with_replacements_only: If it is `true` result will only contain series with suggestions
- limit: Number of records to return in the range 1 - 100. Default is 100.
- offset: The offset from which the records will be returned.

Examples:

```
Ceic.insight series metadata('MY INSIGHT ID')
Ceic.insight series metadata('MY INSIGHT ID', with replacements metadata=True)
for series result in Ceic.insight series metadata('MY_INSIGHT_ID'):
      for insight series in series result.data:
            # Do something with series
```

Ceic.releases()

Description:
Released schedule search. Allows searching for series' release schedule by a keyword and additional filtering criteria.

Usage:

```
releases(keyword=None, **kwargs)
```

Arguments:

- keyword: Search term. One or more keywords. May contain special words further controlling the search results. Keyword search tips:
  - Retail Sales - Show series with both keywords while the sequence of keywords is irrelevant. Equivalent to search Sales Retail
  - Retail AND Sales - Show results: series with terms of Retail AND Sales, regardless of the sequence. E. g. Retail Sales, Automobile Sales Retail.
  - Retail;Sales - Show series with either keyword and series with both keywords while the sequence of keywords is irrelevant, equivalent to search: Sales;Retail
  - Retail OR Sales - Show results: series with terms of Retail OR Sales, regardless of the sequence. E. g. Retail Sales, Retail Trade, Sales Price, Motor Vehicle Sales
  - Retail NOT Sales - Narrow a search by excluding specific terms while the sequence of keywords is relevant. Show results: series with terms that include Retail, but NOT Sales. E. g. Retail Trade, Retail Price, Retail Bank
  - Retail Sales NOT (Hong Kong) - Narrow a search by excluding a set of words in parentheses while the sequence of keywords in parentheses is irrelevant, equivalent to search: Retail Sales NOT (Hong Kong). Show results: series with terms that include Retail Sales, but NOT Hong Kong, such as Retail Sales YoY: China, Retail Sales YoY: United States
- name_only: This filter related with the `keyword` filter. If it's `true` keyword search will be searched only in series name instead of all series attributes.
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en – English
  - zh – Chinese
  - ru – Russian

  - id – Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.
- limit: Number of records to return in the range 1 - 100. Default is 100.
- offset: The offset from which the records will be returned. Used to get the next set of records when the limit is reached.
- database: Database filter. One or more comma separated database code values. Use `/dictionary/databases` to get an up to date list of available databases.
  - WORLD - *World Trend Plus*
  - GLOBAL - *Global Database*
  - CEICGLBKS - *Global Key Series Database*
  - PMI - *Markit Purchasing Managers' Index*
  - DAILY - *Daily Database*
  - BRAZIL - *Brazil Premium Database*
  - RUSSIA - *Russia Premium Database*
  - INDIA - *India Premium Database*
  - INDONESIA - *Indonesia Premium Database*
  - CN - *China Premium Database*
  - OECD-MEI - *OECD - Main Economic Indicators*
  - OECD-EO - *OECD - Economic Outlook*
  - OECD-PROD - *OECD - Productivity*
- frequency: Frequency filter. One or more comma separated frequency code values.
  - D - Daily
  - W - Weekly
  - M - Monthly
  - Q - Quarterly
  - S - Semi-annual
  - Y – Annual
  - Z – Quinquennial
  - T - Decadal
- geo: Geo filter. One or more comma separated geo id values. See related Dictionary function to get the full list of accepted geo ids.
- source: Source filter. One or more comma separated source code values. See related Dictionary function to get the full list of accepted sources.
- unit: Unit filter. One or more comma separated unit code values. See related Dictionary function to get the full list of accepted units.
- indicator: Indicator filter. One or more comma separated indicator code values. See related Dictionary function to get full list of accepted indicators.
- region: Region filter. One or more comma separated region code values. See related Dictionary function to get the full list of accepted regions.
- subscribed_only: Show only results for subscribed series when set to `true`. By default show results for all the series found.
- key_only: Show only 'key' series when set to `true`.
- new_only: Show only series created less than 1 month ago when set to `true`.
- start_date_before: Will return series with first observation before `start_date_before`. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- end_date_after: Will return series with last observation after `end_date_after`. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- created_after: Will return series created after `created_after` date. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- updated_after: Will return series last time updated after `updated_after` date. The date is in ISO 8601 format (YYYYMMDD or YYYY-MM-DD) or a datetime type format.
- status: Series status filter. One or more comma separated status code values. When not explicitly set, defaults to T.
  - T - Active
  - C - Discontinued

  - B - Rebased
- topic: Topic filter. One or more comma separated topic code values.
- section: Section filter. One or more comma separated section code values.
- table: Table filter. One or more comma separated table code values.
- order: Sort order. Default is `relevance`.
- direction: Sort order direction. Default is `asc`. Accepted values: `asc` - ascending `desc` - descending
- filter_id: Filter ID used to define a subset of data over which the search will be executed. When combined with additional search criterion, the result will be an intersection of both.
- with_replacements_metadata: If it is `true` result will contain replacements metadata not only list of id`s
- forecast_only: If it is `true` result will only contain series with forecast
- with_release_only: If it is `true` result will only contain series with released schedule
- with_replacements_only: If it is `true` result will only contain series with suggestions
- facet: List of facets to return
- release_date_from: Will return releases with first observation before `release_date_from`
- release_date_to: Will return releases with last observation after `release_date_to`
- release_status: List of release statuses to return

Examples:
```
Ceic.releases('China', frequency=['Y','M'],release_date_from='2020-01-01')
for search result in Ceic.releases('GDP'):
      for series in search result.data:
            # Do something with series
```

Ceic.release_series()

Description:
Returns list with release code series. Lists the series for a specific release identifier code.

Usage:
```
release_series(code, **kwargs)
```

Arguments:
- code: Release identifier code for a group of series with the same release schedule. (required)
- lang: Preferred language code in which data will be returned. Defaults to `English` if no translation in the language specified is available. Possible Values:
  - en - English
  - zh - Chinese
  - ru - Russian
  - id - Indonesian
  - jp - Japanese
- with_model_information: If set to `true` returns the model names as part of the response.
- limit: Number of records to return in the range 1 - 100. Default is 100.
- offset: The offset from which the records will be returned.

Examples:
```
Ceic.release series('10000416')
for series result in Ceic.release series('10000416'):
      for insight series in series result.data:
            # Do something with series
```