McKinsey & Company Economics Analytics Platform

Last updated: 10/12/2020

API Usage Documentation

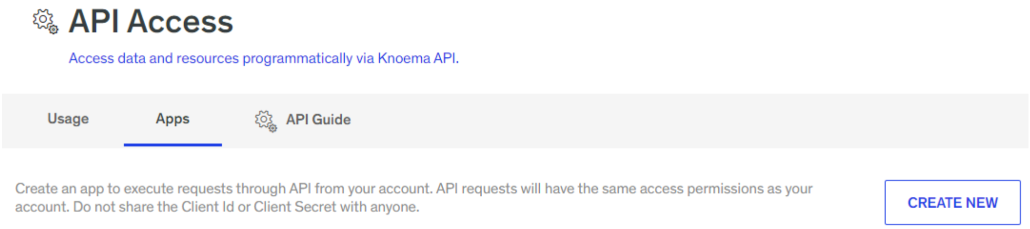
How to access data by API

You can access data by API through three different approaches, depending on your workflow: (1) create an API app; (2) request an API script in your preferred API language directly from the dataset viewer on the system; or (3) call data from DataFinder.

Option 1: Create an API App

To create an API app with a private client ID and secret, follow these steps:

1. Access your *API Access* area accessible from the top-right drop-down menu **or** directly at **https://economics.intranet.mckinsey.com/user/apps**

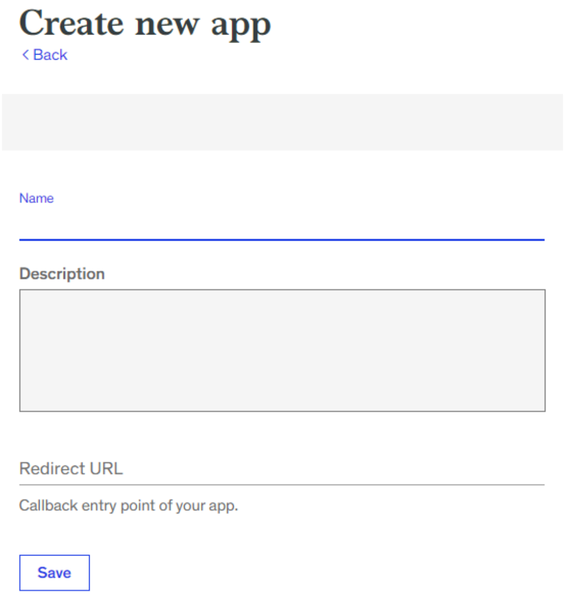


API Access

Access data and resources programmatically via Knoema API.

Create an app to execute requests through API from your account. API requests will have the same access permissions as your account. Do not share the Client Id or Client Secret with anyone.

2. Select *Create New*, give your App a title and description (optional), and click 'Save'.



Create new app

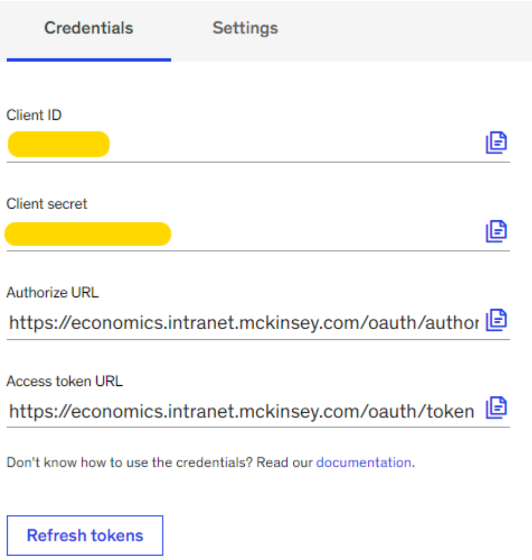
Name

Description

Redirect URL

Callback entry point of your app.

3. After saving, you will see the client ID and secret in the 'Credentials' tab of your app. This is what authorizes your direct access to the application. Do not share this with anyone.



Client ID
[redacted]

Client secret
[redacted]

Authorize URL
https://economics.intranet.mckinsey.com/oauth/author

Access token URL
https://economics.intranet.mckinsey.com/oauth/token

Don't know how to use the credentials? Read our documentation.

Note: Your system administrator may monitor the volume of API calls by user. Additional fees may also apply.

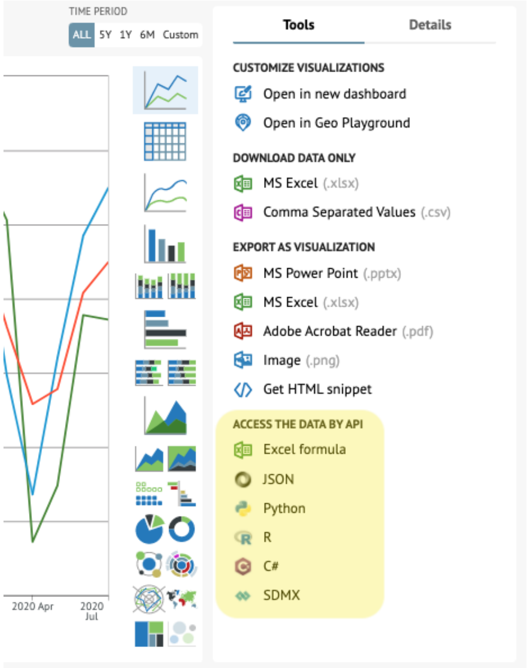
You can also check out Github libraries for assistance in your preferred language:

- Python: https://github.com/Knoema/knoema-python-driver
- R: https://github.com/Knoema/knoema-r-driver
- Java: https://github.com/Knoema/knoema-java-driver
- C#: https://github.com/Knoema/knoema-csharp-driver

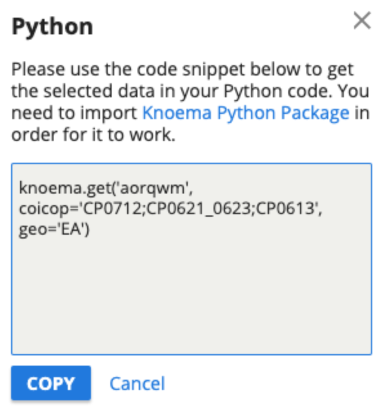
Option 2: Grab an Instant API Script

To extract a specific time series from the dataset viewer by API, follow these steps:

1. Open the dataset.
2. Select the indicator(s).
3. Choose the API format (JSON, Python, R, C#, SDMX). The script will appear in a pop up for you to copy to your application. Using your client ID and secret from Option 1, you can use this script in external applications to pull data by API.



Selecting Python, for example, could give you a script like this:



Python

Please use the code snippet below to get the selected data in your Python code. You need to import Knoema Python Package in order for it to work.

```
knoema.get('aorqwm',
coicop='CP0712;CP0621_0623;CP0613',
geo='EA')
```

Last updated: 10/12/2020

Option 3: Work with DataFinder

Our DataFinder app for Excel provides you with access to your system's data catalog from Excel so you can construct models and coordinate workbooks outside the system with your team.

McKinsey & Company

Economics Analytics Platform

Last updated: 10/12/2020

Introduction to EAP API

The following documentation describes how to use the EAP API to retrieve information on datasets, their structures (including dimensions and their elements), and of course the data itself.

All API requests have been tested in the Postman (https://www.getpostman.com/) and examples from it can be provided by request.

A separate pack of libraries can be downloaded on GitHub for Python, R, Java, and C#. These libraries have documentation available on GitHub and are not repeated here.

- Python: https://github.com/Knoema/knoema-python-driver (documentation)
- R: https://github.com/Knoema/knoema-r-driver (documentation)
- Java: https://github.com/Knoema/knoema-java-driver (documentation)
- C#: https://github.com/Knoema/knoema-csharp-driver

The application also supports SDMX (without the delta updates), more information in this can be found by the following link: https://knoema.com/dev/opendata/sdmx

There is no preferred method to retrieve data. Whether you retrieve data through Python or rest API (described in this document), the result will be the same. It all depends on the environment you wish to work in, i.e. if you're using Python, it will be easier for you to use the Python package then using rest API.

Last updated: 10/12/2020

The EAP REST API

The following provides details on using EAP's REST API to retrieve information about the datasets within your catalog as well as the data itself. As you work, append the URL extensions **highlighted in green** to your system's URL. For example, in *Authentication*, you would construct the link myportal.company.com/user/apps.

Authentication

The first step of using API is always understanding how to authenticate requests because requests are limited by count, by default.

For the EAP API, you need to know the following parameters to get a Bearer Token:

- Client ID (or App ID)
- Client secret (or App Secret)

To get these parameters, log in to EAP, open the link **/user/apps** and click *Create New*. Once your application is created you will see the Application ID and Secret, which you can use to generate Authorization header or you can pass them into the supported libraries (Python, R, Java, and C#) where the Authorization header will be generated automatically based on your personal parameters.

The example below shows how to generate it with Javascript:

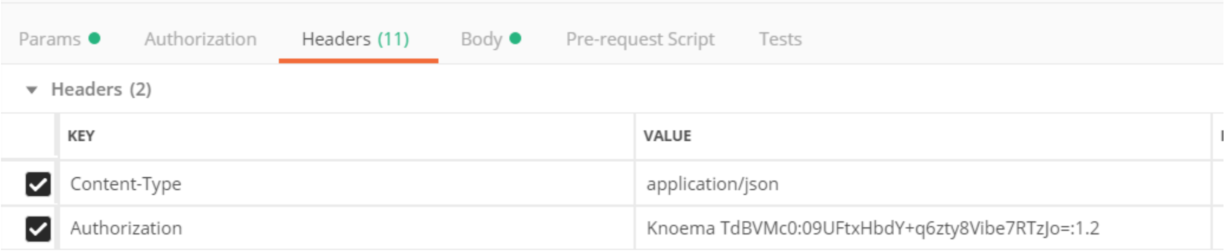
```
var appId = "";
var appSecret = "";
var kapi = {
   appid: appId,
   appsecret: appSecret,
   getAuthHeader: function() {
      var currentTime = new Date();
      var n = ("0" + currentTime.getUTCDate()).slice(-2)
         + "-" + ("0" + (currentTime.getUTCMonth()+1)).slice(-2)
         + "-" + currentTime.getUTCFullYear().toString().substr(-2)
         + "-" + ("0" + currentTime.getUTCHours()).slice(-2);
      var t = CryptoJS.HmacSHA1(appSecret, n).toString(CryptoJS.enc.Base64);
      return "Knoema " + appId + ":" + t + ":1.2"
   }
}
```

Note: The CryptoJS library is used to generate the token. You can use *jsrsassign*, which has an implementation for CryptoJS.HmacSHA1 and CryptoJS.enc.Base64 https://kjur.github.io/jsrsasign/.

- Example on JS Fiddle: https://jsfiddle.net/AlexKulikov/0bqdLhxg

Once set up, you can use it as **kapi.getAuthHeader()** to get the authorization header to use in subsequent requests. Note that the authorization header changes from time to time for security reasons so we recommend regenerating it for each request.

Following is an example of the authorization header with the value from the Postman:



	KEY	VALUE
☑	Content-Type	application/json
☑	Authorization	Knoema TdBVMc0:09UFtxHbdY+q6zty8Vibe7RTzJo=:1.2

Note that there is a 50 requests per day limit on how many requests we may send across for anonymous usage (without Authorization header). With an app, the limit is 500 requests per day. It is also possible to upgrade your application to allow unlimited use.

Authorization tag is also needed to not only increase the number of requests permitted but also to access data that is not publicly available to all users within the platform. The application created under your account shares the same access rights through API as you enjoy through registered access to the EAP.

While it is possible to share one application across several users by sharing AppID/AppSecret, we do not encourage this practice for security reasons.

Metadata

Available Datasets List

To retrieve the list datasets available to you, send a GET request with the authorization header:

Request: **/api/1.0/meta/dataset**
Method: GET
Headers: Authorization
Result: JSON (an archive of JSON objects)

Response example:

```
[
            {
        "dimensions": null,
        "ref": "https://www.imf.org/external/pubs/ft/weo/2019/01/weodata/index.aspx",
        "hasGeoDimension": false,
        "hasGeoMapColumn": false,
        "hasLatLngColumn": false,
        "updatePriority": 0,
        "replacementDataset": null,
        "type": "Regular",
        "regionDimensionId": null,
        "geoPositionDimensionId": null,
        "status": 2,
        "customMetadataFieldValues": null,
        "hasShortCut": false,
        "url": null,
        "settings":
"{\"defaultSelection_fiyjfkd\":{\"Dataset\":\"fiyjfkd\",\"Header\":[{\"FilterText\":null,\"DimensionId\":\"Time\",\"Members\"
:[],\"DatasetId\":\"fiyjfkd\",\"UiMode\":\"allData\"}],\"Stub\":[{\"FilterText\":null,\"DimensionId\":\"country\",\"Members\":[
\"1001500\",\"1001850\"],\"DatasetId\":\"fiyjfkd\"}],\"Filter\":[{\"FilterText\":null,\"DimensionId\":\"subject\",\"Members
\":[\"1000060\"],\"DatasetId\":\"fiyjfkd\"}],\"Frequencies\":[\"A\"],\"RegionIdsRequired\":false,\"RegionDimensionId\":nul
l,\"TimeseriesAttributes\":[]},\"defaultGadgetType_fiyjfkd\":\"line\"}",
        "isShortcut": true,
        "shareToCommunitiesAllowed": true,
        "publicationDate": "2019-04-09T00:00:00",
        "title": "imf-world-economic-outlook-weo-database-april-2019",
        "accessedOn": "2019-04-11T00:00:00",
        "nextReleaseDate": "2019-10-09T00:00:00",
        "lastUpdatedOn": "2019-04-12T16:23:27.103",
        "source": {
            "id": "IMF",
            "url": null,
            "name": "International Monetary Fund",
            "localizedName": "International Monetary Fund",
            "isVerified": true
        },
        "key": 4604400,
        "id": "qhblznb",
        "name": "IMF: World Economic Outlook (WEO) Database, April 2019",
        "description": "Dataset’s description with all HTML tags used.",
        "owner": "DatasetOwner",
        "isPublic": true,
        "rawDescription": "Dataset’s description as raw text."
    }
]
```

This request is usually used to retrieve the full list of available datasets. If you already know the dataset IDs (field “id” in the result above) you do not need to send this request.

Last updated: 10/12/2020

Dataset Details

To retrieve the details of the dataset by its unique ID, send a GET request with the authorization header.

Request: **/api/1.0/meta/dataset/<DatasetID>**
Request example: **/api/1.0/meta/dataset/qhblznb**

Method: GET
Headers: Authorization
Result: JSON object

Response example:

```
{
    "dimensions": [
        {
            "key": 4267050,
            "id": "country",
            "name": "Country",
            "isGeo": true,
            "geoType": "region"
        }, {
            "key": 4267060,
            "id": "subject",
            "name": "Subject",
            "isGeo": false,
            "geoType": null
        }
    ],
    "ref": "https://www.imf.org/external/pubs/ft/weo/2019/01/weodata/index.aspx",
    "hasGeoDimension": true,
    "hasGeoMapColumn": false,
    "hasLatLngColumn": false,
    "updatePriority": 0,
    "replacementDataset": null,
    "columns": [
        {
            "key": 12154510,
            "name": "Country",
            "order": 1,
            "type": "Text"
        }, {
```

```
        "key": 12154520,
        "name": "RegionId",
        "order": 2,
        "type": "Text"
      }, {
        "key": 12154530,
        "name": "ISO",
        "order": 3,
        "type": "Text"
      }, {
        "key": 12154540,
        "name": "Id",
        "order": 4,
        "type": "Text"
      }, {
        "key": 12154550,
        "name": "Subject",
        "order": 5,
        "type": "Text"
      }, {
        "key": 12154560,
        "name": "SubjectNotes",
        "order": 6,
        "type": "Text"
      }, {
        "key": 12154570,
        "name": "Subject Name as per IMF",
        "order": 7,
        "type": "Text"
      }, {
        "key": 12154580,
        "name": "Id",
        "order": 8,
        "type": "Text"
      }, {
        "key": 12154590,
        "name": "Value",
        "order": 9,
        "type": "Number"
      }, {
        "key": 12154600,
        "name": "Date",
        "order": 10,
        "type": "Date"
      }
    ],
    "type": "Regular",
    "regionDimensionId": "country",
    "geoPositionDimensionId": null,
    "status": 2,
    "customMetadataFieldValues": [],
    "hasShortCut": false,
    "url": null,
    "settings":
  "{\"defaultSelection_fiyjfkd\":{\"Dataset\":\"fiyjfkd\",\"Header\":[{\"FilterText\":null,\"DimensionId\":\"Time\",\"Members\"
  :[],\"DatasetId\":\"fiyjfkd\",\"UiMode\":\"allData\"}],\"Stub\":[{\"FilterText\":null,\"DimensionId\":\"country\",\"Members\":[
  \"1001500\",\"1001850\"],\"DatasetId\":\"fiyjfkd\"}],\"Filter\":[{\"FilterText\":null,\"DimensionId\":\"subject\",\"Members
  \":[\"1000060\"],\"DatasetId\":\"fiyjfkd\"}],\"Frequencies\":[\"A\"],\"RegionIdsRequired\":false,\"RegionDimensionId\":nul
  l,\"TimeseriesAttributes\":[]},\"defaultGadgetType_fiyjfkd\":\"line\"}",
    "isShortcut": true,
    "shareToCommunitiesAllowed": true,
    "publicationDate": "2019-04-09T00:00:00",
    "title": "imf-world-economic-outlook-weo-database-april-2019",
    "accessedOn": "2019-04-11T00:00:00",
    "nextReleaseDate": "2019-10-09T00:00:00",
    "lastUpdatedOn": "2019-04-12T16:23:27.103",
    "source": {
      "id": "IMF",
      "url": null,
      "name": "International Monetary Fund",
      "localizedName": "International Monetary Fund",
      "isVerified": true
    },
    "key": 4604400,
    "id": "qhblznb",
    "name": "IMF: World Economic Outlook (WEO) Database, April 2019",
    "description": "Dataset's description with all HTML tags used.",
    "owner": "DatasetOwner",
    "isPublic": true,
    "rawDescription": "Dataset's description as raw text."
  }
```

This request is usually used to retrieve and understand the structure of the dataset (see property “dimensions,” which tells you how many dimensions the dataset has and the IDs of each) and details such as data source, name, and description.

Once you understand the dataset structure, you're ready to retrieve its dimensions and elements.

Last updated: 10/12/2020

Dataset Dimension Details

To retrieve the details of the dimension by dataset ID and dimension ID, send a GET request with the authorization header.

Note: Instead of dimension ID, you also can use dimension name.

Request: **/api/1.0/meta/dataset/<DatasetID>/dimension/<DimensionID>**
Request example: **/api/1.0/meta/dataset/qhblznb/dimension/subject**

Method: GET
Headers: Authorization

Result: JSON object

Response example:

```
{
    "key": 4267060,
    "id": "subject",
    "name": "Subject",
    "isGeo": false,
    "datasetId": "qhblznb",
    "datasetType": 0,
    "fields": [
        {
            "key": 5856590,
            "name": "id",
            "displayName": "Id",
            "type": 6,
            "locale": null,
            "baseKey": null,
            "isSystemField": false
        }, {
            "key": 5856600,
            "name": "subjectnotes",
            "displayName": "SubjectNotes",
            "type": 6,
            "locale": null,
            "baseKey": null,
            "isSystemField": false
        }, {
            "key": 5856610,
            "name": "subject-name-as-per-imf",
            "displayName": "Subject Name as per IMF",
            "type": 6,
            "locale": null,
            "baseKey": null,
            "isSystemField": false
        }
    ],
    "items": [
        {
            "key": 1000000,
            "name": "National Accounts",
            "level": 0,
            "hasData": false,
            "fields": {
                "id": "KN.NA",
                "subjectnotes": "",
                "subject-name-as-per-imf": "National Accounts"
            }
        }, {
            "key": 1000010,
            "name": "Gross domestic product, constant prices (Percent change)",
            "level": 1,
            "hasData": true,
            "fields": {
                "id": "NGDP_RPCH",
                "subjectnotes": "Annual percentages of constant price GDP are year-on-year changes; the base year is country-specific . Expenditure-based GDP is total final expenditures at purchasers? prices (including the f.o.b. value of exports of goods and services), less the f.o.b. value of imports of goods and services. [SNA 1993]",
                "subject-name-as-per-imf": "Gross domestic product, constant prices (Percent change)"
            }
        }, {
            "key": 1000020,
            "name": "Gross domestic product, constant prices (Percent change (market exchange rates))",
            "level": 1,
            "hasData": true,
            "fields": {
                "id": "NGDP_RPCHMK",
                "subjectnotes": "",
                "subject-name-as-per-imf": "Gross domestic product, constant prices (Percent change (market exchange rates))"
            }
        }, {
            "key": 1000030,
            "name": "Gross domestic product, current prices (U.S. dollars)",
            "level": 1,
            "hasData": true,
            "fields": {
                "id": "NGDPD",
                "subjectnotes": "Values are based upon GDP in national currency converted to U.S. dollars using market exchange rates (yearly average). Exchange rate projections are provided by country economists for the group of other emerging market and developing countries. Exchanges rates for advanced economies are established in the WEO assumptions for each WEO exercise. Expenditure-based GDP is total final expenditures at purchasers? prices (including the f.o.b. value of exports of goods and services), less the f.o.b. value of imports of goods and services. [SNA 1993]",
                "subject-name-as-per-imf": "Gross domestic product, current prices (U.S. dollars)"
            }
        }],
"groups": []
}
```

This request is needed to understand the structure of each dimension (the above example is for the Subject dimension of the IMF World Economic Outlook (WEO) dataset). The *Fields* property contains a list of fields, which is useful to understand what metadata might be available for elements of the dataset. The *Items* property contains the full list of elements available within a dimension, metadata for each element is available under *fields* property of an appropriate item (see the example above). It also contains information on the Key (or ID) of each element, as well as its Name, Level (for building a hierarchy), and whether it contains data.

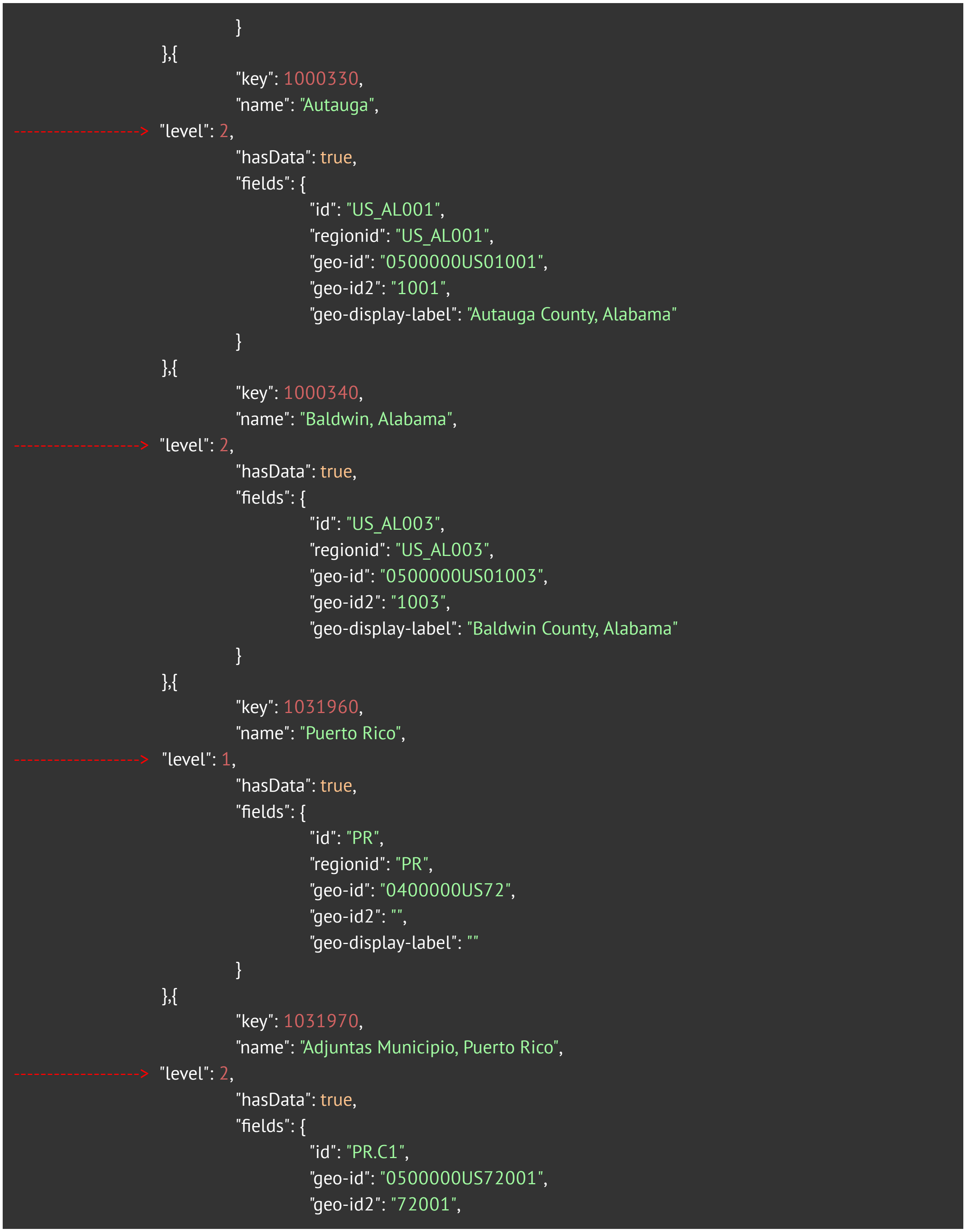
Once you know the Dataset ID and its dimensions, you can retrieve the data through Data Pivot or Data Raw requests or even a Data Simple Get Endpoint.

If your dataset consists of dimensions with a hierarchical structure, it is possible to get nesting information through your API request. To retrieve the details of the dimension by dataset ID and dimension ID, send a GET request with the authorization header.

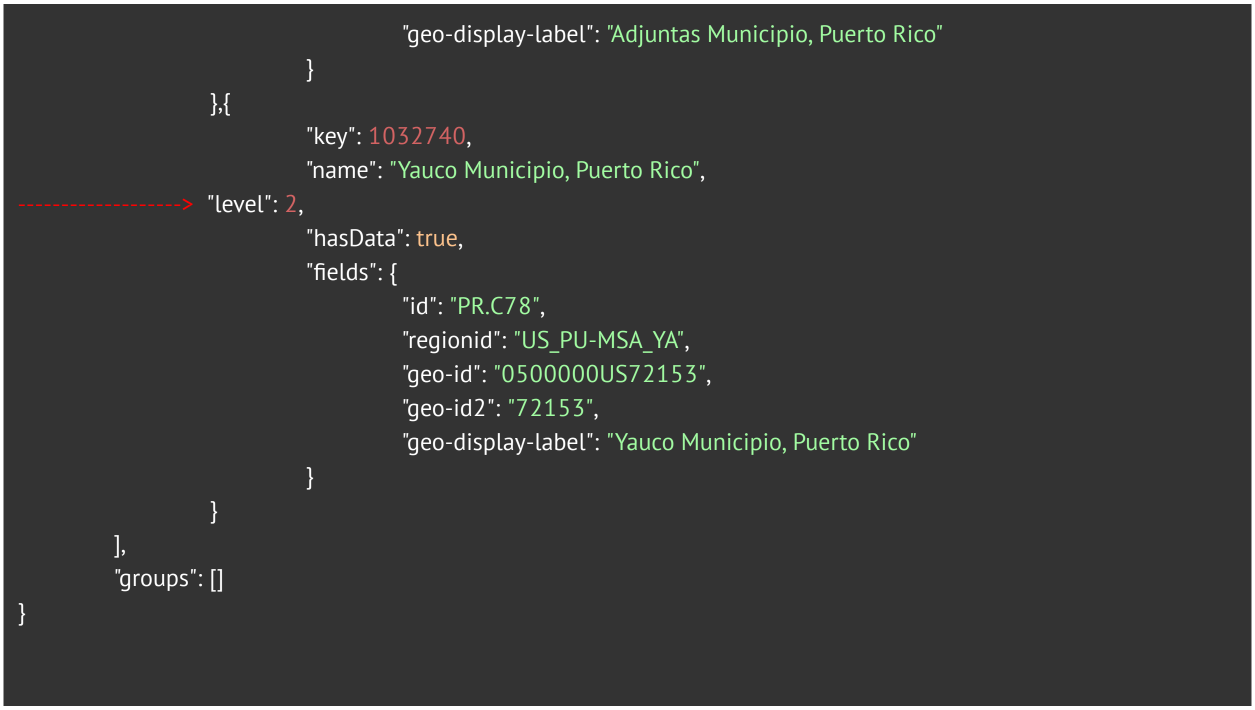
Request: **/api/1.0/meta/dataset/<DatasetID>/dimension/<DimensionID>**
Request example: **/api/1.0/meta/dataset/KPMGDHC2019/dimension/location**

Method: GET
Headers: Authorization
Result: JSON object
Response example:

```
{
            "key": 3930470,
            "id": "location",
            "name": "Location",
            "isGeo": true,
            "datasetId": "KPMGDHC2019",
            "datasetType": 0,
            "fields": [{
                            "key": 5196560,
                            "name": "id",
                            "displayName": "Id",
                            "type": 6,
                            "locale": null,
                            "baseKey": null,
                            "isSystemField": false
                    },{
                            "key": 5196570,
                            "name": "RegionId",
                            "displayName": "RegionId",
                            "type": 6,
                            "locale": null,
                            "baseKey": null,
                            "isSystemField": true
                    },
            ...
            ],
            "items": [{
                            "key": 1000000,
                            "name": "United States",
--------------> "level": 0,
                            "hasData": true,
                            "fields": {
                                    "id": "US",
                                    "regionid": "US",
                                    "geo-id": "0100000US",
                                    "geo-id2": "",
                                    "geo-display-label": ""
                            }
                    },{
                            "key": 1000320,
                            "name": "Alabama",
--------------> "level": 1,
                            "hasData": true,
                            "fields": {
                                    "id": "US_AL",
                                    "regionid": "US_AL",
                                    "geo-id": "0400000US01",
                                    "geo-id2": "1",
                                    "geo-display-label": "Alabama"
```



```
            }
        },{
            "key": 1000330,
            "name": "Autauga",
------> "level": 2,
            "hasData": true,
            "fields": {
                "id": "US_AL001",
                "regionid": "US_AL001",
                "geo-id": "0500000US01001",
                "geo-id2": "1001",
                "geo-display-label": "Autauga County, Alabama"
            }
        },{
            "key": 1000340,
            "name": "Baldwin, Alabama",
------> "level": 2,
            "hasData": true,
            "fields": {
                "id": "US_AL003",
                "regionid": "US_AL003",
                "geo-id": "0500000US01003",
                "geo-id2": "1003",
                "geo-display-label": "Baldwin County, Alabama"
            }
        },{
            "key": 1031960,
            "name": "Puerto Rico",
------> "level": 1,
            "hasData": true,
            "fields": {
                "id": "PR",
                "regionid": "PR",
                "geo-id": "0400000US72",
                "geo-id2": "",
                "geo-display-label": ""
            }
        },{
            "key": 1031970,
            "name": "Adjuntas Municipio, Puerto Rico",
------> "level": 2,
            "hasData": true,
            "fields": {
                "id": "PR.C1",
                "geo-id": "0500000US72001",
                "geo-id2": "72001",
```



```
                "geo-display-label": "Adjuntas Municipio, Puerto Rico"
            }
        },{
            "key": 1032740,
            "name": "Yauco Municipio, Puerto Rico",
------> "level": 2,
            "hasData": true,
            "fields": {
                "id": "PR.C78",
                "regionid": "US_PU-MSA_YA",
                "geo-id": "0500000US72153",
                "geo-id2": "72153",
                "geo-display-label": "Yauco Municipio, Puerto Rico"
            }
        }
    ],
    "groups": []
}
```

The request would return the structure with an [items] collection, where the elements are listed in the same order as they are shown in the dataset, and the hierarchy is based on the *level* field. In the above response example, the United States is the first item with level 0, and Alaska is the next item with level 1, indicating that it is 1 level below the United States, and so forth.

Last updated: 10/12/2020

Dataset Time-series List

To get the list of time-series from the dataset, there are two requests available: GET and POST.

GET Request (Full List)

Request: **/api/1.0/data/dataset/<DatasetID>**
Request example: **/api/1.0/data/dataset/qhblznb**

Method: GET
Headers: Authorization
Result: JSON (array of JSON objects)

Response example:

```
[
            {
      "datasetId": "qhblznb",
      "timeseriesKey": 1000070,
      "frequency": "Annual",
      "metadata": [
         {
            "dim": "country",
            "key": "1000000",
            "name": "Afghanistan"
         }, {
            "dim": "subject",
            "key": "1000150",
            "name": "Gross domestic product based on purchasing-power-parity (PPP) valuation of country GDP (Current international dollar)"
         }
      ],
      "startDate": "01/01/2002 00:00:00",
      "endDate": "01/01/2024 00:00:00"
   }, {
      "datasetId": "qhblznb",
      "timeseriesKey": 1000080,
      "frequency": "Annual",
      "metadata": [
         {
            "dim": "country",
            "key": "1000000",
                "name": "Afghanistan"
            }, {
                "dim": "subject",
                "key": "1000160",
                "name": "Gross domestic product based on purchasing-power-parity (PPP) per capita GDP (Current international dollar)"
            }
        ],
        "startDate": "01/01/2002 00:00:00",
        "endDate": "01/01/2024 00:00:00"
    }
]
```

The result is a full list of available time-series, which can be used to receive the data via the time-series key.

POST Request (Filtered)

Request: **/api/1.0/data/dataset/<DatasetID>**
Request example: **/api/1.0/data/dataset/qhblznb**

Method: POST
Headers: Authorization, Content-Type=application/json

Body example: The body should contain a list of dimensions with element keys for which you want a list of the available time-series.

```
{
  "DimensionRequest": [
    {
      "DimensionId": "Country",
      "Members": [1000000]
    }, {
      "DimensionId": "Subject",
      "Members": [1000820, 1000650, 1000490, 1001390]
    }
  ],
  "Frequency": ["A"],
  "StartDate": "1901-11-23",
  "EndDate": "2020-11-23"
}
```

Result: JSON (array of JSON objects)

Response example:

```
[
    {
        "datasetId": "qhblznb",
        "timeseriesKey": 1000310,
        "frequency": "Annual",
        "metadata": [
            {
                "dim": "country",
                "key": "1000000",
                "name": "Afghanistan"
            }, {
                "dim": "subject",
                "key": "1000650",
                "name": "General government total expenditure (Percent of GDP)"
            }
        ],
        "startDate": "01/01/2002 00:00:00",
        "endDate": "01/01/2024 00:00:00"
    }
]
```

The result provides a full list of the available time-series for your selection, which can be used to receive the data via the time-series key.

Data Requests

In this section we provide directions to guide you through retrieving data. Included are the following:

- Time series data
- Raw data
- Pivot data
- Simple dataset

Get Data by Time-series

Request: **/api/1.0/data/get**

Method: POST
Headers: Authorization, Content-Type=application/json

Body example: The body should contain information about the data you want to retrieve. You may omit the StartDate, Frequency, and CustomFields and use only Timeseries and EndDate.

```
{
  "Timeseries": [
    {
      "TimeseriesKey": 1000100,
      "DatasetId": "qhblznb"
    }, {
      "TimeseriesKey": 1000200,
      "DatasetId": "qhblznb"
    }
  ],
  "StartDate": "01/01/1991",
  "EndDate": "01/01/2018",
  "Frequency": "A",
  "CustomFields": true
}
```

Result: JSON (array of JSON objects)

Response example:

McKinsey & Company Economics Analytics Platform

Last updated: 10/12/2020

```
[
    {
        "dataset": "qhblznb",
        "id": 1000100,
        "metadata": [
            {
                "dim": "country",
                "key": 1000000,
                "name": "Afghanistan",
                "id": "512",
                "regionid": "AF",
                "iso": "AFG"
            }, {
                "dim": "subject",
                "key": 1000180,
                "name": "Gross domestic product per capita, current prices (U.S. dollars)",
                "id": "512",
                "subjectnotes": "GDP is expressed in current U.S. dollars per person. Data are derived by first converting GDP in national currency to U.S. dollars and then dividing it by total population.",
                "subject-name-as-per-imf": "Gross domestic product per capita, current prices (U.S. dollars)"
            }
        ],
        "startDate": "Tue, 01 Jan 2002 00:00:00 GMT",
        "endDate": "Mon, 01 Jan 2018 00:00:00 GMT",
        "scaleValue": 1,
        "unit": "U.S. dollars",
        "frequency": "annual",
        "data": [196.683, 196.954, 214.23, 248.049, 270.189, 321.441, 377.256, 430.867,       532.072, 602.183, 661.087, 635.635, 629.332, 594.513, 560.601, 569.531, 543.724]
    },
    {
        "dataset": "qhblznb",
        "id": 1000200,
        "metadata": [
            {
                "dim": "country",
                "key": 1000000,
                "name": "Afghanistan",
                "id": "512",
                "regionid": "AF",
                "iso": "AFG"
            }, {
                "dim": "subject",
                "key": 1000390,
                "name": "Import volume of goods and services (Percent change)",
            "id": "512",
            "subjectnotes": "Source: Various sources: Central Statistical Office;",
            "subject-name-as-per-imf": "Volume of imports of goods and services"
          }
        ],
        "startDate": "Wed, 01 Jan 2003 00:00:00 GMT",
        "endDate": "Mon, 01 Jan 2018 00:00:00 GMT",
        "scaleValue": 1,
        "unit": "Percent change",
        "frequency": "annual",
        "Data": [34.585, 0.844, 56.662, -1.924, -9.622, -8.186, 28.513, 34.258, 12.349, 30.697, 5.177, -15.896, 15.562,
 -8.126, 3.298, -1.111]
    }
]
```

To read the data, you will need to understand the Frequency and the StartDate and EndDate for each time-series. For Annual frequency, each line/step is equal to one year. So, in our example, 34.585 refers to 2003 and -1.924 refers to 2006.

Get Raw Data

Request: **/api/1.0/data/details**

Method: POST
Headers: Authorization, Content-Type=application/json

Body example: The body should specify the data you want to retrieve. You can list as many Members as you want to in the Stub & Header, and only one in the Filter (or you can omit it).

```
{
  "Header": [
    {
      "DimensionId": "Time",
      "UiMode": "range",
      "Members": [
        "2018-2019"
      ],
      "DatasetId": "qhblznb"
    }
  ],
  "Stub": [
    {
      "DimensionId": "country",
      "Members": [1000000],
      "IsGeo": true
    }, {
      "DimensionId": "subject",
      "Members": [1000010]
    }
  ],
  "Filter": [],
  "Frequencies": [
    "A"
  ],
  "Dataset": "qhblznb"
}
```

Last updated: 10/12/2020

Result: JSON

Response example:

```
{
    "columns": [
        {
            "name": "Country",
            "type": "Text",
            "status": "Dimension",
            "dimensionId": "country",
            "fieldId": null
        }, {
            "name": "RegionId",
            "type": "Text",
            "status": "Dimension",
            "dimensionId": "country",
            "fieldId": "regionid"
        }, {
            "name": "ISO",
            "type": "Text",
            "status": "Dimension",
            "dimensionId": "country",
            "fieldId": "iso"
        }, {
            "name": "Id",
            "type": "Text",
            "status": "Dimension",
            "dimensionId": "country",
            "fieldId": "id"
        }, {
            "name": "Subject",
            "type": "Text",
            "status": "Dimension",
            "dimensionId": "subject",
            "fieldId": null
        }, {
            "name": "SubjectNotes",
            "type": "Text",
            "status": "Dimension",
            "dimensionId": "subject",
            "fieldId": "subjectnotes"
        }, {
            "name": "Subject Name as per IMF",
            "type": "Text",
            "status": "Dimension",
            "dimensionId": "subject",
            "fieldId": "subject-name-as-per-imf"
        }, {
            "name": "Id",
            "type": "Text",
            "status": "Dimension",
            "dimensionId": "subject",
            "fieldId": "id"
        }, {
            "name": "Value",
            "type": "Number",
            "status": "Measure",
            "dimensionId": null,
            "fieldId": null
        }, {
            "name": "Date",
            "type": "Date",
            "status": "Date",
            "dimensionId": null,
            "fieldId": null
        }
    ],
    "data": [
        "Afghanistan",
        "AF",
        "AFG",
        "512",
        "Gross domestic product, constant prices (Percent change)",
        "Annual percentages of constant price GDP are year-on-year changes; the base year is country-specific . Expenditure-based GDP is total final expenditures at purchasers? prices (including the f.o.b. value of exports of goods and services), less the f.o.b. value of imports of goods and services. [SNA 1993]",
        "Gross domestic product, constant prices (Percent change)",
        "NGDP_RPCH",
        2.3,
        {
            "value": "01/01/2018",
            "frequency": "A"
        },
        "Afghanistan",
        "AF",
        "AFG",
        "512",
        "Gross domestic product, constant prices (Percent change)",
        "Annual percentages of constant price GDP are year-on-year changes; the base year is country-specific . Expenditure-based GDP is total final expenditures at purchasers? prices (including the f.o.b. value of exports of goods
and services), less the f.o.b. value of imports of goods and services. [SNA 1993]",
        "Gross domestic product, constant prices (Percent change)",
        "NGDP_RPCH",
        3,
        {
            "value": "01/01/2019",
            "frequency": "A"
        }
    ],
    "filter": [],
    "datasetName": "IMF: World Economic Outlook (WEO) Database, April 2019"
}
```

The columns property shows the response structure and the Data property the data itself (based on the structure).

Last updated: 10/12/2020

Get Pivot Data

Request: **/api/1.0/data/pivot**

Method: POST
Headers: Authorization, Content-Type=application/json

Body example: The body should specify the data you want to retrieve. You can list as many Members as you want to in the Stub & Header, and only one in the Filter (or you can omit it). It is similar to the body request for raw data.

```
{
  "Header": [
    {
      "DimensionId": "Time",
      "Members": [
        "2018-2019"
      ],
      "UiMode": "range"
    }
  ],
  "Stub": [
    {
      "DimensionId": "country",
      "Members": [1000000],
      "IsGeo": true
    }, {
      "DimensionId": "subject",
      "Members": [1000010]
    }
  ],
  "Filter": [],
  "Frequencies": [
    "A"
  ],
  "Dataset": "qhblznb"
}
```

Result: JSON

Response example:

```
{
    "header": [
        {
            "dimensionId": "Time",
            "filterText": null,
            "members": [
                "2018", "2019"
            ],
            "metadataFields": []
        }
    ],
    "stub": [
        {
            "dimensionId": "country",
            "filterText": null,
            "members": [
                "Afghanistan"
            ],
            "metadataFields": []
        },
        {
            "dimensionId": "subject",
            "filterText": null,
            "members": [
                "Gross domestic product, constant prices (Percent change)"
            ],
            "metadataFields": []
        }
    ],
    "filter": [],
    "datasetName": "IMF: World Economic Outlook (WEO) Database, April 2019",
    "dataset": "qhblznb",
    "keys": {
        "header": [
            {
                "dimensionId": "Time",
                "filterText": null,
                "members": [
                    "2018",
                    "2019"
                ],
                "metadataFields": []
            }
        ],
    "stub": [
      {
        "dimensionId": "country",
        "filterText": null,
        "members": [
          "1000000"
        ],
        "metadataFields": []
      },
      {
        "dimensionId": "subject",
        "filterText": null,
        "members": [
          "1000010"
        ],
        "metadataFields": []
      }
    ],
    "filter": []
  },
  "data": [
    {
      "Time": "2018-01-01T00:00:00Z",
      "subject": "Gross domestic product, constant prices (Percent change)",
      "country": "Afghanistan",
      "RegionId": "AF",
      "Frequency": "A",
      "Value": 2.3,
      "Unit": "Variation en pourcentage",
      "Scale": 1
    },
    {
      "Time": "2019-01-01T00:00:00Z",
      "subject": "Gross domestic product, constant prices (Percent change)",
      "country": "Afghanistan",
      "RegionId": "AF",
      "Frequency": "A",
      "Value": 3,
      "Unit": "Variation en pourcentage",
      "Scale": 1
    }
  ]
}
```

The Data property has the data (Value) with information on your selection (Subject, Country, Frequency, Time).

Last updated: 10/12/2020

Get Simple Dataset

Request:
/api/1.0/data/<DatasetID>?<DimensionID>=<DimensionElementKey>&<DimensionID>=<DimensionElementKey>...

Request example:
/api/1.0/data/qhblznb?country=1000000&subject=1000010&time=2018,2019

Several dimension elements keys can be provided, separated by commas.

Method: GET
Headers: Authorization

Response example:

```
{
   "header": [
      {
         "dimensionId": "Time",
         "filterText": null,
         "members": [
            "2018",
            "2019"
         ],
         "metadataFields": []
      }
   ],
   "stub": [
      {
         "dimensionId": "country",
         "filterText": null,
         "members": [
            "Afghanistan"
         ],
         "metadataFields": []
      }, {
         "dimensionId": "subject",
         "filterText": null,
         "members": [
            "Gross domestic product, constant prices (Percent change)"
         ],
```

Last updated: 10/12/2020

```
        "metadataFields": []
    }
],
"filter": [],
"datasetName": "IMF: World Economic Outlook (WEO) Database, April 2019",
"dataset": "qhblznb",
"keys": {
    "header": [
        {
            "dimensionId": "Time",
            "filterText": null,
            "members": [
                "2018",
                "2019"
            ],
            "metadataFields": []
        }
    ],
    "stub": [
        {
            "dimensionId": "country",
            "filterText": null,
            "members": [
                "1000000"
            ],
            "metadataFields": []
        }, {
            "dimensionId": "subject",
            "filterText": null,
            "members": [
                "1000010"
            ],
            "metadataFields": []
        }
    ],
    "filter": []
},
"data": [
    {
        "Time": "2018-01-01T00:00:00Z",
        "subject": "Gross domestic product, constant prices (Percent change)",
        "country": "Afghanistan",
        "RegionId": "AF",
        "Frequency": "A",
        "Value": 2.3,
        "Unit": "Variation en pourcentage",
            "Scale": 1
        }, {
            "Time": "2019-01-01T00:00:00Z",
            "subject": "Gross domestic product, constant prices (Percent change)",
            "country": "Afghanistan",
            "RegionId": "AF",
            "Frequency": "A",
            "Value": 3,
            "Unit": "Variation en pourcentage",
            "Scale": 1
        }
    ]
}
```

This approach is the easiest way to get the data and can be executed right from your browser. However, if you have a complex request, it is better to use ‘Get Pivot Data’. Also, the “Get Simple Dataset” request cannot be used with time-series.