

How to API i

General

Preamble

- I cant answer everything, most of it is learning by doing
- Most of the APIs are not really user-friendly documented
- Hopefully it will not take 2 hours

What is an API?

API = Application Programming Interface

- programmatic way to retrieve data from online servers
- can be anything (uploading, downloading, instructing)
- different API types: REST, SDMX, JSON, ... (*think dialects*)

Goal: Format Request in specific way to get specific data from

What is SDMX?



The official site for the SDMX community

A global initiative to improve Statistical Data and Metadata

an API for statistical agencies

- BIS, ECB, EUROSTAT, IMF, OECD, World Bank, UN (?), ILO, ma
- used in many different contexts (e.g NSO pushes to IO etc)
- lengthy 145 page User Guide [Link](#)
- only very small / relevant subset covered here (GET DATA)

How does the SDMX API work?

each call must have identifiers for:

- Statistical Agency (*Provider*)
- Database (*Dataflow*)
- Columns / Groupings (*Dimensions*), e.g.
 - selected Countries
 - Indicator you want
 - Time Period
 - **Dimensions differ for each dataset!**

A Basic call: chain all of these together to get the wanted data.

very abstract, so lets see some

Examples

IMF

On the Website

Goal: CPI data since 1960 for USA, United Kingdom, and Germany

1. data.imf.org
2. [CPI Dataset](#)
3. [Data Explorer](#)



[ABOUT](#) [RESEARCH](#) [COUNTRIES](#) [CAPACITY DEVELOPMENT](#) [NEWS](#) [VIDEOS](#) [DATA](#)

Search IMF Data 

[Data Home](#) > [Data Explorer](#)

Data Explorer

▼ Dataset: **Consumer Price Index (CPI)**

▼ Display All/ By Table: **Full List** ...

▼ Country

▼ Index type

▼ Expenditure Category

▼ Type of Transformation

▼ Frequency  

▼ Time Period: 01/01/2017 - 31/12/2026

everything with an icon = Dimension

Series Count: 30,757

APPLY

Consumer Price Index (CPI) ⓘ

	Country	Index type	Expenditure Ca
①	Aruba, Kingdom of the Nethe...	Consumer price index (CPI)	Food and non-a
①	Aruba, Kingdom of the Nethe...	Consumer price index (CPI)	Food and non-a
①	Aruba, Kingdom of the Nethe...	Consumer price index (CPI)	Food and non-a
①	Aruba, Kingdom of the Nethe...	Consumer price index (CPI)	Food and non-a
①	Aruba, Kingdom of the Nethe...	Consumer price index (CPI)	Food and non-a
①	Aruba, Kingdom of the Nethe...	Consumer price index (CPI)	Food and non-a
①	Aruba, Kingdom of the Nethe...	Consumer price index (CPI)	Food and non-a
①	Aruba, Kingdom of the Nethe...	Consumer price index (CPI)	Food and non-a
①	Aruba, Kingdom of the Nethe...	Consumer price index (CPI)	Food and non-a
①	Aruba, Kingdom of the Nethe...	Consumer price index (CPI)	Food and non-a
①	Aruba, Kingdom of the Nethe...	Consumer price index (CPI)	Food and non-a
①	Aruba, Kingdom of the Nethe...	Consumer price index (CPI)	Alcoholic bever
①	Aruba, Kingdom of the Nethe...	Consumer price index (CPI)	Alcoholic bever

- In the Filter Tab, select categories for all dimensions (one or more)
- then click on Listing Icon and switch to Id
- take note of the IDs you want
- THE ORDER IS IMPORTANT!

Dimension	Name
Country	United States, United Kingdom, Germany
Index Type	Consumer Price Index
Expenditure	All Items
Type of Transformation	Period average, Y-O-Y percentage change
Frequency	Annual

In the Code

First, lets import the relevant libraries:

```
1 library(tidyverse)
2 library(rsdmx)
```

Then we define our key by pasting together the selected IDs for

```
1 COUNTRIES <- "USA+GBR+DEU" # All countries
2 INDICATOR <- "CPI" # selected Indicator
3 EXPENDITURE <- "_T" # All items / Total
4 TRANSFORMATION <- "YOY_PCH_PA_PT" # Percentage change
5 FREQUENCY <- "A" # Annual
6
7 key <- paste0(COUNTRIES, ".", INDICATOR, ".", EXPENDITURE, ".", TRANSFORMATION, ".", FREQUENCY)
8
9 key
```

```
[1] "USA+GBR+DEU.CPI._T.YOY_PCH_PA_PT.A"
```

Now, we call the API, specifying our provider, database, key, and

```
1 raw_data <- readSDMX(  
2   providerId = "IMF_DATA", # IMF as Provider  
3   resource = "data",      # we want data  
4   flowRef = "CPI",        # from the CPI database  
5   key = key,               # our carefully created key  
6   start = 1960,           # lets limit it to start in 1960  
7 )
```

```
[rsdmx][INFO] Fetching 'https://api.imf.org/external/sdmx/2.1/data/CPI/USA+GB  
startPeriod=1960'
```

The response is in SDMX format, so we have to convert it to a data frame.

The response is in SDMX format, so we have to convert it to a data frame.

```

1 df <- as.data.frame(raw_data)
2
3 head(df)

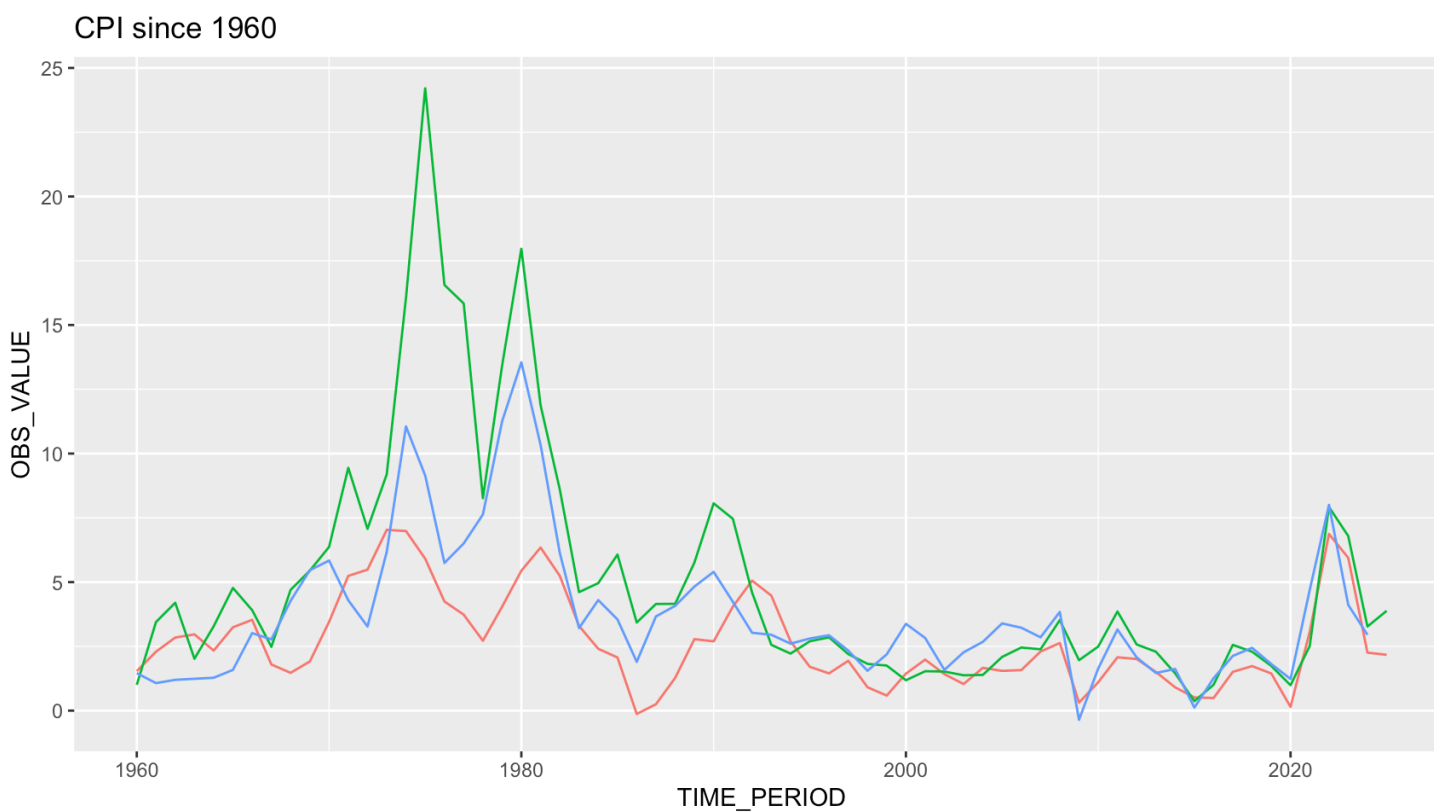
```

	COUNTRY <chr>	INDEX_TYPE <chr>	COICOP_1999 <chr>	TYPE_OF_TRANSFORMA <chr>
1	DEU	CPI	_T	YOY_PCH_PA_PT
2	DEU	CPI	_T	YOY_PCH_PA_PT
3	DEU	CPI	_T	YOY_PCH_PA_PT
4	DEU	CPI	_T	YOY_PCH_PA_PT
5	DEU	CPI	_T	YOY_PCH_PA_PT
6	DEU	CPI	_T	YOY_PCH_PA_PT

6 rows | 1-7 of 14 columns

Lets make a quick plot

```
1 df %>%
2   select(COUNTRY, TIME_PERIOD, OBS_VALUE) %>% # select relevant columns
3   mutate(                                     # change types of columns
4     OBS_VALUE = as.numeric(OBS_VALUE),
5     TIME_PERIOD = as.numeric(TIME_PERIOD)
6   ) %>%
7   ggplot(aes(x=TIME_PERIOD, y=OBS_VALUE, color=COUNTRY)) + # basic gg
8     geom_line() + # present as line plot
9     labs(title="CPI since 1960") # add title
```



OECD

Goal: Official Development Assistance (ODA)

1. data-explorer.oecd.org/
2. [DAC Table 1](#)

On the Website

Refine your data selection:

Time period	10
Donor	1
Measure	1
Flow type	1
Unit of measure	1
Price base	1

1) build a selection

7 data points selected in this dataset with:

Donor:	...> DAC countries	Measure:	...> Official Development Assistance
Flow type:	Grant equivalents	Unit of measure:	US dollar
Time period:	Last 10 periods (s)		

Overview	Table	Chart	Labels	Layout
----------	-------	-------	--------	--------

DAC1: Flows by provider (ODA+OOF+Private) i

Donor: DAC countries • **Measure:** Official Development Assistance

Price base: Constant prices

Combined unit of measure: US dollar, Millions, 2023

Time period	
2018	
2019	
2020	
2021	
2022	
2023	
2024	

Refine your data selection:

Time period	10
Donor	1
Measure	1
Flow type	1
Unit of measure	1
Price base	1

3) copy the API code

7 data points selected in this dataset with:

Donor: ...>DAC countries | Measure: ...>Official De

Flow type: Grant equivalents | Unit of measure: US

Time period: Last 10 years (4)

Overview

Table

Chart

Labels

Developer API query builder

The application programming interface (API) based on the SDMX standard allows you to access the data using simple RESTful URL and HTTP header of JSON.

To get started, check how to access [OECD data via API](#). For more information, see the [API documentation](#).

Data query

SDMX flavour: Flat Time series

```
https://sdmx.oecd.org/public/rest/data/OECD.DCD.FSD,DSD_DAC1@DF_DAC1,1.6/DAC._Z.1010..1160.USD.Q?startPeriod=2015&dimensionAtObservation=AllDimensions
```

Copy code

In my Code

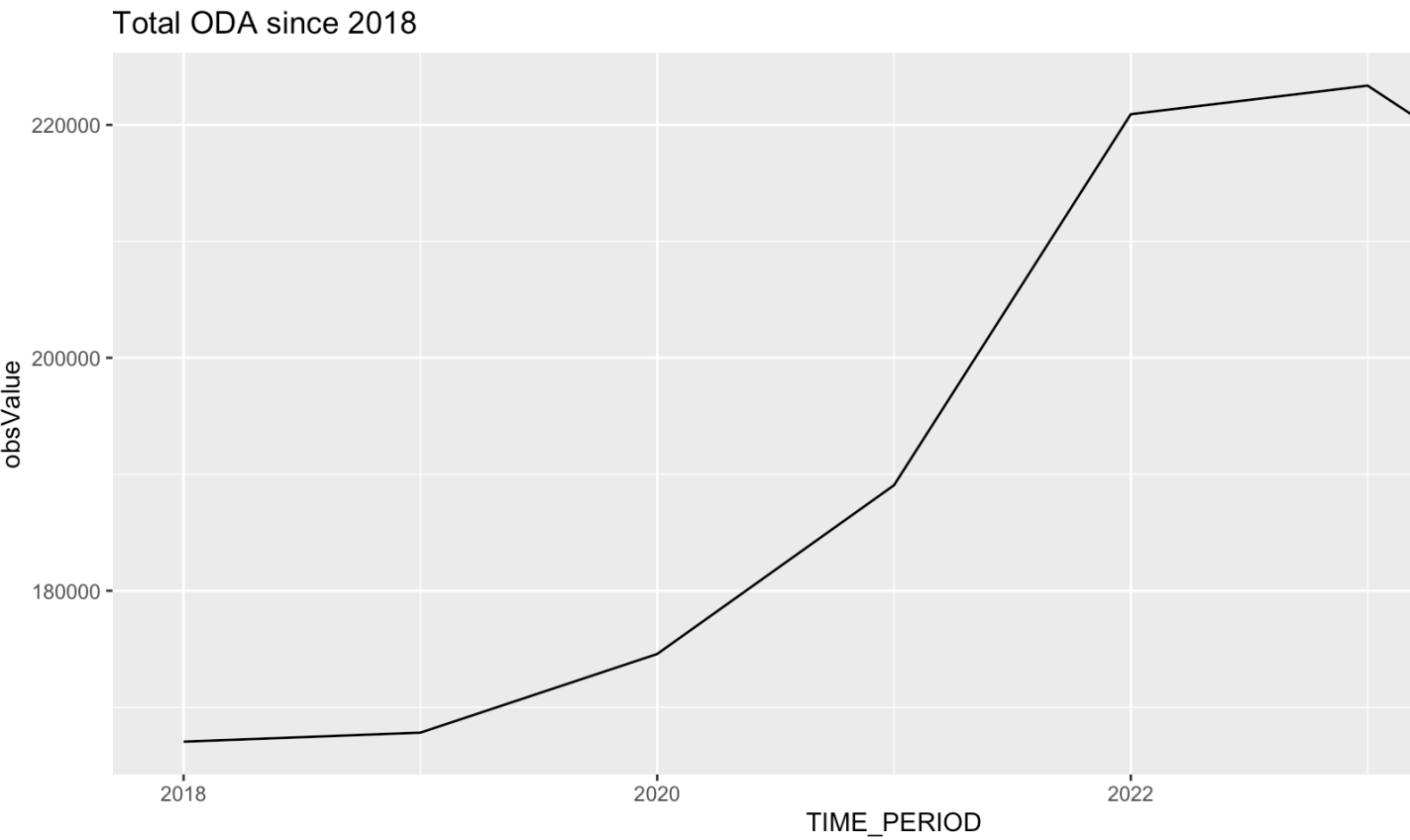
```
1 url <- "https://sdmx.oecd.org/public/rest/data/OECD.DCD.FSD,DSD_DAC1@DF_L"
```

Much easier...

```
1 raw_data_oecd <- readSDMX(url)
2 df_oecd <- as.data.frame(raw_data_oecd)
```

Again, a quick plot

```
1 df_oecd %>%
2   mutate(
3     obsValue = as.numeric(obsValue),
4     TIME_PERIOD = as.numeric(TIME_PERIOD)
5   ) %>%
6   ggplot(aes(x=TIME_PERIOD, y=obsValue)) +
7   geom_line() +
8   labs(title="Total ODA since 2018")
```



Exercise

Prerequisites

The RSDMX Package

```
1 install.packages("rstmx")
```

Your Task

Unemployment Rate for Spain and Italy since 2012

- both ways possible
- different tradeoffs:
 - IMF = much easier to search, but more difficult to access
 - OECD = vice versa

Links:

- data.imf.org
- data-explorer.oecd.org

The Solution: IMF

IMF DATA

Search IMF Data

🔍

🏠 Data Home > Data Explorer

Data Explorer

⏪

▼ Dataset: World Economic Outlook (WEO) ⚙️ ⋮

▼ Display All/ By Table: Full List ⋮

▼ Country: Spain, Italy (2) × 📄 ⚙️ ⋮

▼ Indicator: Unemployment rate (1) × 📄 ⋮

▼ Frequency 📄 ⋮

▼ Time Period: 01/01/2012 - 31/12/2026

World Economic Outlook (WEO) ⓘ

	Country	Indicator	Freq
ⓘ	Spain	Unemployment rate	Ann
ⓘ	Italy	Unemployment rate	Ann

```
1 COUNTRIES <- "ESP+ITA" # Spain and Italy
2 INDICATOR <- "LUR" # Unemployment Rate
3 FREQUENCY <- "A" # Annual
4
5 key <- paste0(COUNTRIES, ".", INDICATOR, ".", FREQUENCY)
```

```
1 raw_data_unemp <- readSDMX(
2   providerId = "IMF_DATA",
3   resource = "data",
4   flowRef = "WE0", # World Economic Outlook
5   key = key,
6   start = 2012,
7   )
```

```
[rsdmx][INFO] Fetching 'https://api.imf.org/external/sdmx/2.1/data/WE0/ESP+IT'
```

```
1 df_unemp <- as.data.frame(raw_data_unemp)
2 df_unemp %>%
3   select(COUNTRY, TIME_PERIOD, OBS_VALUE) %>%
4   pivot_wider(names_from=COUNTRY, values_from=OBS_VALUE)
```

TIME_PERIOD	ESP
<chr>	<chr>
2012	24.788
2013	26.095
2014	24.443
2015	22.058
2016	19.635
2017	17.225
2018	15.255
2019	14.105
2020	15.533
2021	14.92

1-10 of 19 rows

The Solution: Eurostat

Search for “Unemployment Rate” on data-explorer.oecd.org

Misleading: namend “monthly Unemployment Rate”, but has a

This Table has everything we need: [Link](#)

Refine your data selection:

Frequency of observation & Time period	13
Reference area	2
Adjustment	1
Sex	1
Age	1

26 data points selected in this dataset with:

Reference area: Italy Spain | Transformation: No

Adjustment: Calendar and seasonally adjusted | Sex: T

Frequency of observation: Annual | Time period: Last 13 period

Clear all

Overview

Table

Chart

Labels

Monthly unemployment rates

Age: 15 years or over • Frequency of observation: Annual

Measure: Monthly unemployment rate •

Unit of measure: Percentage of labour force in the same subgroup,

Time period		2012	2013	2014	2015	2016	2017
Reference area							
Italy		10.9	12.4	12.8	12.0	11.8	11.5
Spain		24.8	26.1	24.5	22.1	19.7	17.8

```
1 url <- "https://sdmx.oecd.org/public/rest/data/OECD.SDD.TPS,DSD_LFS@DF_IA"
```

```
1 raw_data_eurostat <- readSDMX(url)
2 df_eurostat <- as.data.frame(raw_data_eurostat)
```

```
1 df_eurostat %>%
2   select(REF_AREA, TIME_PERIOD, obsValue) %>%
3   pivot_wider(names_from=REF_AREA, values_from=obsValue)
```

TIME_PERIOD	REF_AREA	obsValue
<chr>	<dbl>	<dbl>
2012	EU	10.8833
2013	EU	12.3666
2014	EU	12.8250
2015	EU	12.0000
2016	EU	11.7500
2017	EU	11.2583
2018	EU	10.6000
2019	EU	9.9333
2020	EU	9.3000
2021	EU	9.5583

1-10 of 13 rows

Helpful Tips

Github Copilot

Coding Assistant

- with [free tier](#)
- directly integrated in RStudio, completes your code on TAB

Note:

- not well trained on (SDMX) API calls, so you probably have to
- tidyverse = works really good (although sometimes quite in

Quarto Markdown

New format in [RStudio](#)

- combines Prose and Code
- very good for documenting code and pasting links etc
- allows to produce PDFs, HTML, slides, ...

Tidyverse Cheat Sheets

All packages in the tidyverse in R have very good cheat sheets.

- **dplyr** (for transforming data): [PDF](#)
- **ggplot** (data visualisation): [PDF](#)
- **tidyr** (for cleaning data): [PDF](#)

Links

- [SDMX.org](https://sdmx.org)
- [imfapi](#) package for R (quite new, but should work)