Stata-Python API for bulk data download: Example with UN Comtrade

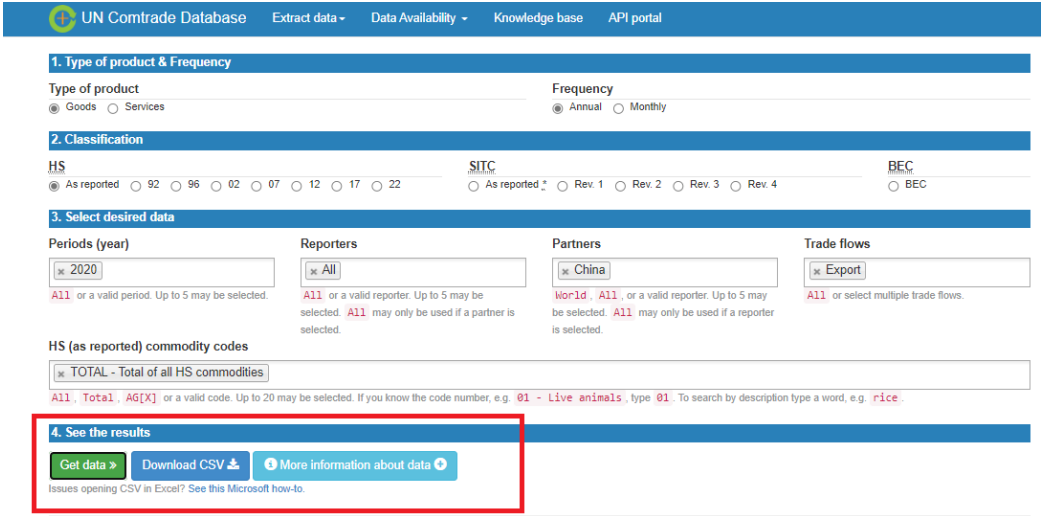
Ka Lok Wong (Steve)
Geneva Graduate Institute, Switzerland

November 18, 2022

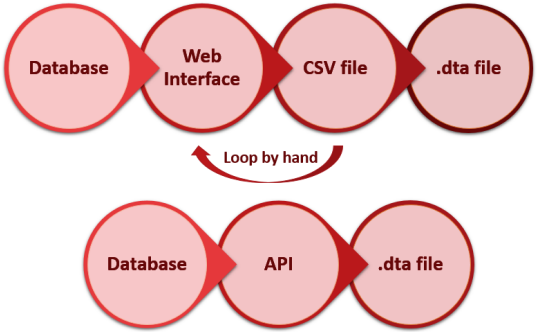
The granular data problem

- Increasing availability of trade data through public domains
  - 40+ years, 6-digit (∼5,300 products)
  - So as bulk data size
- Papers made use of Comtrade data/ BACI dataset on CEPII:
  - Gaulier and Zignago, 2012;
  - Hausman and Hidalgo, 2011;
  - Stadler et al, 2018
- Downloading and importing 40 Gb+ CSV files is tedious and **very prone** to human error

What is not an API?



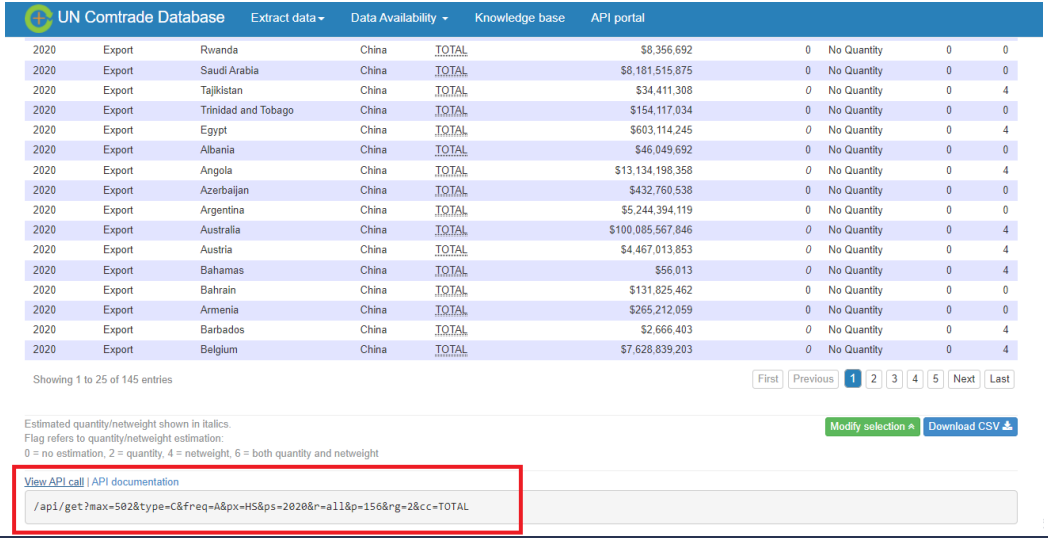
What is an API?



- **Alternative**: automatizing the tasks from the data sourcing stage, i.e. **API**
- API stands for **Application Programming Interface**
- It allows two applications to communicate with each other using requests and responses
- API documentation is therefore a **map** to find the way through the maze

Benefits:
- Time saving
- Memory saving
- Automation to avoid human error

What is an API? (cont'l)



UN Comtrade Database | Extract data | Data Availability | Knowledge base | API portal

2020	Export	Rwanda	China	TOTAL	$8,356,692	0	No Quantity	0	0
2020	Export	Saudi Arabia	China	TOTAL	$8,181,515,875	0	No Quantity	0	0
2020	Export	Tajikistan	China	TOTAL	$34,411,308	0	No Quantity	0	4
2020	Export	Trinidad and Tobago	China	TOTAL	$154,117,034	0	No Quantity	0	0
2020	Export	Egypt	China	TOTAL	$603,114,245	0	No Quantity	0	4
2020	Export	Albania	China	TOTAL	$46,049,692	0	No Quantity	0	0
2020	Export	Angola	China	TOTAL	$13,134,198,358	0	No Quantity	0	4
2020	Export	Azerbaijan	China	TOTAL	$432,760,538	0	No Quantity	0	0
2020	Export	Argentina	China	TOTAL	$5,244,394,119	0	No Quantity	0	0
2020	Export	Australia	China	TOTAL	$100,085,567,846	0	No Quantity	0	4
2020	Export	Austria	China	TOTAL	$4,467,013,853	0	No Quantity	0	4
2020	Export	Bahamas	China	TOTAL	$56,013	0	No Quantity	0	4
2020	Export	Bahrain	China	TOTAL	$131,825,462	0	No Quantity	0	0
2020	Export	Armenia	China	TOTAL	$265,212,059	0	No Quantity	0	0
2020	Export	Barbados	China	TOTAL	$2,666,403	0	No Quantity	0	4
2020	Export	Belgium	China	TOTAL	$7,628,839,203	0	No Quantity	0	4

Showing 1 to 25 of 145 entries

First | Previous | 1 | 2 | 3 | 4 | 5 | Next | Last

Estimated quantity/netweight shown in italics.
Flag refers to quantity/netweight estimation:
0 = no estimation, 2 = quantity, 4 = netweight, 6 = both quantity and netweight

Modify selection | Download CSV

View API call | API documentation

```
/api/get?max=502&type=C&freq=A&px=HS&ps=2020&r=all&p=156&rg=2&cc=TOTAL
```

Outline

Trade data as an example

Step 1: Identify the data of interest

- Country-pair: country *i* to China, for all countries in the world
- Direction: Export
- Frequency: Annual
- Time period: 2000 to 2021
- Classification: All 2-digit SITC Rev. 2

1. Input the according info on Comtrade - Get data
2. **"View API call"** in the bottom
3. The requested data is therefore accessible via:
   - http://comtrade.un.org/api/get?max=10000&type=C&freq=A&px=S2&ps=2021&r=all&p=156&rg=2&cc=AG2

Step 2: Understand the API call

Table: API explanation

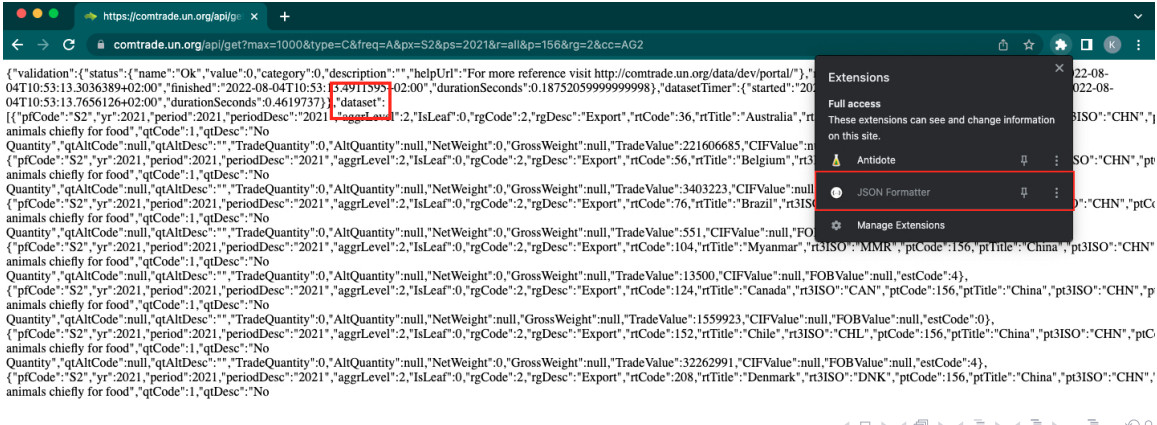
Parameter	Range	Category	In simple words
max=502	[1, 10000]	Limit	Data download limits
type=C	[C,S]	Data type	C is goods; S is Services
freq=A	[A,M]	Obs. frequency	A is annual frequency; M is monthly
px=S2	S[i], H[i], i = [1,4]	Classification system	Product code classification
ps=2021	[1962,2022]	Time period	**Year**
r=all	All, UN codes	Reporting area	All is all countries (not aggregated)
p=156	All, UN codes	Partner(s)	156 is China's UN numerical country code
rg=2	[1,2,all]	Trade flow	2 is export from reporting area to partner, 1 is import
cc=AG2	[TOTAL, AG1, ..., AG6]	Classification code	AG2 is all of the 2-digit codes

/api/get?[Parameter1]&[Parameter2]&.....

{[Parameter] = [choice] }

Step 3 Check the API link

http://comtrade.un.org/api/get?max=10000&type=C&freq=A&px=S2&ps=2021&r=all&p=156&rg=2&cc=AG2 should show a JSON file:



Step 4.1 Set up the destination folder:

- **Task_folder**
  - code
  - output

```
local filepath = "/Users/.../blogpost1_api_comtrade" // adjust to yours
*country i export to china
cd `filepath'/output/i_X_CHN

python:

[python code]

end
```

Step 4.2 Initiate Python in Stata do-file

```
local filepath = "/Users/.../blogpost1_api_comtrade" // adjust to yours
*country i export to china
cd `filepath'/output/i_X_CHN

python:

[python code]

end
```

Step 4.3 Add necessary Python packages

```
python:
import json
import numpy        as  np
import pandas       as  pd
import requests

url       =
↪  f'http://comtrade.un.org/api/get?max=10000&type=C&freq=A&px=S2&ps=2021&r=all&p=156&rg=2&cc=AG2'

result    = requests.get(url).json()
if 'dataset' in result:
        df        = pd.DataFrame(result['dataset'])
        df        = df.replace({None: np.nan})
        df.columns= [i[:32] for i in df.columns]
        df.to_stata(f'i_X_China_2021.dta')
end

use ./i_X_China_2021, clear
```

Step 4.4 API link and JSON file

```
python:
import json
import numpy         as  np
import pandas        as  pd
import requests

url       =
↪  f'http://comtrade.un.org/api/get?max=10000&type=C&freq=A&px=S2&ps=2021&r=all&p=156&rg=2&cc=AG2'

result    = requests.get(url).json()
if 'dataset' in result:
        df         = pd.DataFrame(result['dataset'])
        df         = df.replace({None: np.nan})
        df.columns= [i[:32] for i in df.columns]
        df.to_stata(f'i_X_China_2021.dta')
end

use ./i_X_China_2021, clear
```

Step 4.5 Fish the data

5) Recall that we found our data of interest from "dataset" : in Step 3

```
python:
import json
import numpy        as  np
import pandas       as  pd
import requests

url       =
↪  f'http://comtrade.un.org/api/get?max=10000&type=C&freq=A&px=S2&ps=2021&r=all&p=156&rg=2&cc=AG2'

result    = requests.get(url).json()
if 'dataset' in result:
        df        = pd.DataFrame(result['dataset'])
        df        = df.replace({None: np.nan})
        df.columns= [i[:32] for i in df.columns]
        df.to_stata(f'i_X_China_2021.dta')
end

use ./i_X_China_2021, clear
```

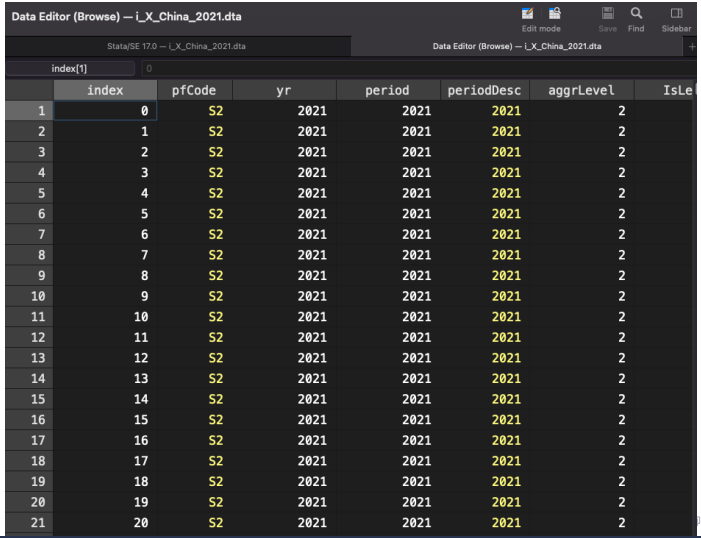
Step 4.6 Store the data in Stata format (.dta)

```
python:
import json
import numpy         as  np
import pandas        as  pd
import requests

url       =
↪  f'http://comtrade.un.org/api/get?max=10000&type=C&freq=A&px=S2&ps=2021&r=all&p=156&rg=2&cc=AG2'

result    = requests.get(url).json()
if 'dataset' in result:
        df         = pd.DataFrame(result['dataset'])
        df         = df.replace({None: np.nan})
        df.columns= [i[:32] for i in df.columns]
        df.to_stata(f'i_X_China_2021.dta')
end

use ./i_X_China_2021, clear
```

Step 4.7 Check the result

```
python:
import json
import numpy         as  np
import pandas        as  pd
import requests

url       =
↪  f'http://comtrade.un.org/api/get?max=10000&type=C&freq=A&px=S2&ps=2021&r=all&p=156&rg=2&cc=AG2'

result    = requests.get(url).json()
if 'dataset' in result:
        df         = pd.DataFrame(result['dataset'])
        df         = df.replace({None: np.nan})
        df.columns= [i[:32] for i in df.columns]
        df.to_stata(f'i_X_China_2021.dta')
end

use ./i_X_China_2021, clear
```

Step 4 Test run result



	index	pfCode	yr	period	periodDesc	aggrLevel	IsLe
1	0	S2	2021	2021	2021	2	
2	1	S2	2021	2021	2021	2	
3	2	S2	2021	2021	2021	2	
4	3	S2	2021	2021	2021	2	
5	4	S2	2021	2021	2021	2	
6	5	S2	2021	2021	2021	2	
7	6	S2	2021	2021	2021	2	
8	7	S2	2021	2021	2021	2	
9	8	S2	2021	2021	2021	2	
10	9	S2	2021	2021	2021	2	
11	10	S2	2021	2021	2021	2	
12	11	S2	2021	2021	2021	2	
13	12	S2	2021	2021	2021	2	
14	13	S2	2021	2021	2021	2	
15	14	S2	2021	2021	2021	2	
16	15	S2	2021	2021	2021	2	
17	16	S2	2021	2021	2021	2	
18	17	S2	2021	2021	2021	2	
19	18	S2	2021	2021	2021	2	
20	19	S2	2021	2021	2021	2	
21	20	S2	2021	2021	2021	2	

How about looping it over years?

python:

```
url       =
↪  f'http://comtrade.un.org/api/get?max=10000&type=C&freq=A&px=S2&ps=2021&r=all&p=156&rg=2&cc=AG2'

end
```

{[Parameter1] = [choice1] } & {[Parameter2] = [choice2] } & ...

Step 5.1 Compartmentalize the API link

```
python:
def Comtrade_Scraper   (ps: int,
                        type: str=   'C',
                        freq: str=   'A',
                        px  : str=  'S2',
                        r   : str= 'all',
                        p   : int=   156,
                        rg  : int=     2,
                        cc  : str= 'AG2'):

    base      = 'https://comtrade.un.org/api/get?max=10000'
    url       = f'{base}&type={type}&freq={freq}&px={px}&ps={ps}&r={r}&p={p}&rg={rg}&cc={cc}'
    result    = requests.get(url).json()
    if 'dataset' in result:
        df        = pd.DataFrame(result['dataset'])
        df        = df.replace({None: np.nan})
        df.columns= [i[:32] for i in df.columns]
        df.to_stata(f'i_X_China_{ps}.dta')
        return df
for i in range(2000,2022): Comtrade_Scraper(i)
end
```

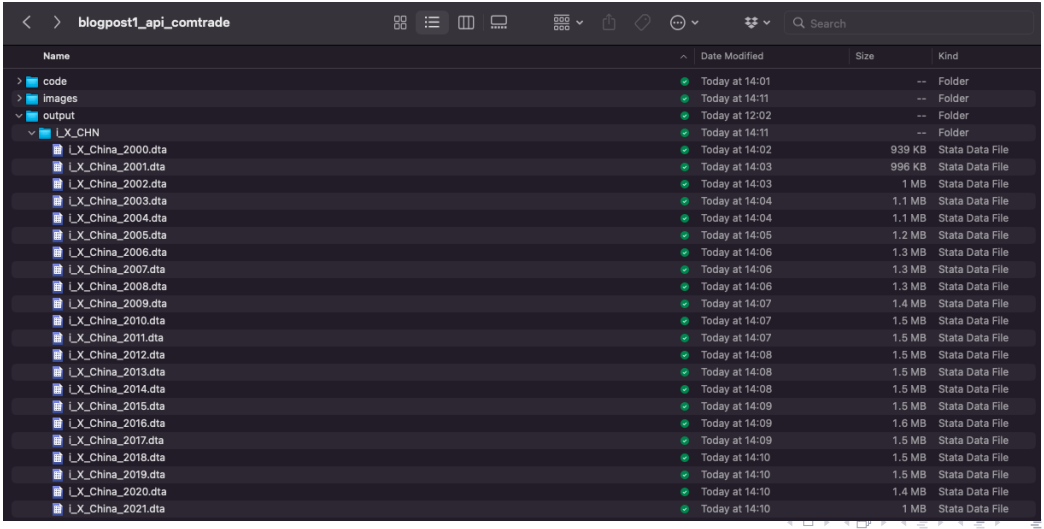
Step 5.2 Assemble the API link

```
python:
def Comtrade_Scraper   (ps: int,
                        type: str=   'C',
                        freq: str=   'A',
                        px  : str=  'S2',
                        r   : str= 'all',
                        p   : int=   156,
                        rg  : int=     2,
                        cc  : str= 'AG2'):

    base      = 'https://comtrade.un.org/api/get?max=10000'
    url       = f'{base}&type={type}&freq={freq}&px={px}&ps={ps}&r={r}&p={p}&rg={rg}&cc={cc}'
    result    = requests.get(url).json()
    if 'dataset' in result:
        df        = pd.DataFrame(result['dataset'])
        df        = df.replace({None: np.nan})
        df.columns= [i[:32] for i in df.columns]
        df.to_stata(f'i_X_China_{ps}.dta')
        return df
for i in range(2000,2022): Comtrade_Scraper(i)
end
```

Step 5.3 Introduce the loop

```
python:
def Comtrade_Scraper   (ps: int,
                        type: str=   'C',
                        freq: str=   'A',
                        px  : str=  'S2',
                        r   : str= 'all',
                        p   : int=   156,
                        rg  : int=     2,
                        cc  : str= 'AG2'):

    base      = 'https://comtrade.un.org/api/get?max=10000'
    url       = f'{base}&type={type}&freq={freq}&px={px}&ps={ps}&r={r}&p={p}&rg={rg}&cc={cc}'
    result    = requests.get(url).json()
    if 'dataset' in result:
        df        = pd.DataFrame(result['dataset'])
        df        = df.replace({None: np.nan})
        df.columns= [i[:32] for i in df.columns]
        df.to_stata(f'i_X_China_{ps}.dta')
        return df
for i in range(2000,2022): Comtrade_Scraper(i)
end
```

Step 5.4 Result



Step 5.5 Final touch

6) Final touch

```
di "`c(pwd)'" // Display path to current folder
local files : dir "`c(pwd)'" files "*.dta"
foreach x of local files {
    di "`x'" // Display file name
        append using `x'
}
save ./i_X_China_2000_2021.dta, replace
```

Some final notes

- There is an existing user-written package comtrade by Jan Ditzen.
  - Last updated on June 2020
- Comtrade recently published a new API portal
- Data limit remains a challenge, 10'000 for guest users
- There is often glitch in at least 1 file
- Possible to do two-way loop, e.g. over partner country and over time
- There is potential to rewrite into a more flexible structure in order to accommodate multiple data platforms

Credit and reference

Credits to Satyam Anand (Georgetown University) and Bernhard Bieri (World Bank) whom introduced the API magic to me.

More resources:

- Setting up Stata to use Python by Chuck Huber
- Stata-Python integration blog post by Chuck Huber
- Comtrade API documentation

Full blog post on `https://www.stevekwong.com/blog/apicomtrade`