

Web Scrapping Tables using Pandas

Estimated Effort: 5 mins

The Pandas library in Python contains a function `read_html()` that can be used to extract tabular information from any web page.

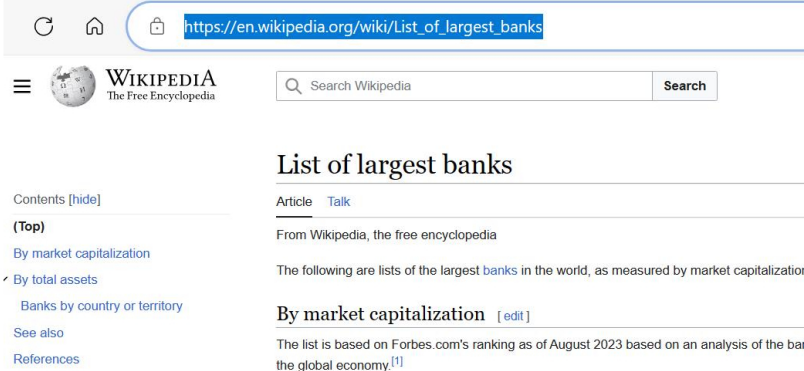
Consider the following example:

Let us assume we want to extract the list of the largest banks in the world by market capitalization, from the following link:

```
1 URL = 'https://en.wikipedia.org/wiki/List_of_largest_banks'
```

We may use `pandas.read_html()` function in python to extract all the tables in the web page directly.

A snapshot of the webpage is shown below.



We can see that the required table is the first one in the web page.

Note: This is a live web page and it may get updated over time. The image shown above has been captured in November 2023. The process of data extraction remains the same.

We may execute the following lines of code to extract the required table from the web page.

```
1 import pandas as pd
2 URL = 'https://en.wikipedia.org/wiki/List_of_largest_banks'
3 tables = pd.read_html(URL)
4 df = tables[0]
5 print(df)
```

This will extract the required table as a dataframe `df`. The output of the print statement would look as shown below.

Rank	Bank name	Market cap(US\$ billion)
0 1	JPMorgan Chase	419.25
1 2	Bank of America	231.52
2 3	Industrial and Commercial Bank of China	194.56
3 4	Agricultural Bank of China	160.68
4 5	HDFC Bank	157.91
5 6	Wells Fargo	155.87
6 7	HSBC Holdings PLC	148.90
7 8	Morgan Stanley	140.83
8 9	China Construction Bank	139.82
9 10	Bank of China	136.81

Although convenient, this method comes with its own set of limitations.

Firstly, web pages may have content saved in them as tables but they may not appear as tables on the web page.

For instance, consider the following URL showing the list of countries by GDP (nominal).

```
1 URL = 'https://en.wikipedia.org/wiki/List_of_countries_by_GDP_(nominal)'
```

The images on the web page are also saved in tabular format. A snapshot of the web page is shared below.


















Table 1

Table 2

1		United Kingdom	Europe	3,332,059	2023	3,075,688	2022	3,131,378	2021
2		France	Europe	3,049,016	2023	2,782,905	2022	2,957,880	2021
3		Italy	Europe	2,186,082	2023	2,010,432	2022	2,107,703	2021
4		Brazil	Americas	2,126,809	2023	1,920,096	2022	1,608,981	2021
5		Canada	Americas	2,117,805	2023	2,139,840	2022	1,988,336	2021
6		Russia	Europe	1,862,470	2023	2,240,422	2022	1,778,782	2021
7		Mexico	Americas	1,811,468	2023	1,414,187	2022	1,272,839	2021
8		South Korea	Asia	1,709,232	2023	1,665,246	2022	1,810,966	2021
9		Australia	Oceania	1,687,713	2023	1,675,419	2022	1,734,532	2021
10		Spain	Europe	1,582,054	2023	1,397,509	2022	1,427,381	2021

Table 3

Secondly, the contents of the tables in the web pages may contain elements such as hyperlink text and other denoters, which are also scraped directly using the pandas method. This may lead to a requirement of further cleaning of data. A closer look at table 3 in the image shown above indicates that there are many hyperlink texts which are also going to be treated as information by the pandas function.

GDP (USD million) by country								
Country/Territory	UN region	IMF ^{[1][13]}		World Bank ^[14]		United Nations ^[15]		
		Forecast	Year	Estimate	Year	Estimate	Year	
World	—	104,476,432	2023	100,562,011	2022	96,698,005	2021	
 United States	Americas	26,949,643	2023	25,462,700	2022	23,315,081	2021	
 China	Asia	17,700,899	^[n 1] 2023	17,963,171	^[n 3] 2022	17,734,131	^[n 1] 2021	
 Germany	Europe	4,429,838	2023	4,072,192	2022	4,259,935	2021	
 Japan	Asia	4,230,862	2023	4,231,141	2022	4,940,878	2021	
 India	Asia	3,732,224	2023	3,385,090	2022	3,201,471	2021	
 United Kingdom	Europe	3,332,059	2023	3,070,668	2022	3,131,378	2021	
 France	Europe	3,049,016	2023	2,782,905	2022	2,957,880	2021	
 Italy	Europe	2,186,082	2023	2,010,432	2022	2,107,703	2021	
 Brazil	Americas	2,126,809	2023	1,920,096	2022	1,608,981	2021	
 Canada	Americas	2,117,805	2023	2,139,840	2022	1,988,336	2021	
 Russia	Europe	1,862,470	2023	2,240,422	2022	1,778,782	2021	
 Mexico	Americas	1,811,468	2023	1,414,187	2022	1,272,839	2021	
 South Korea	Asia	1,709,232	2023	1,665,246	2022	1,810,966	2021	
 Australia	Oceania	1,687,713	2023	1,675,419	2022	1,734,532	2021	
 Spain	Europe	1,582,054	2023	1,397,509	2022	1,427,381	2021	

We can extract the table using the code shown below.

```
1 import pandas as pd
2 URL = 'https://en.wikipedia.org/wiki/List_of_countries_by_GDP_(nominal)'
3 tables = pd.read_html(URL)
4 df = tables(2) # the required table will have index 2
5 print(df)
```

The output of the print statement is shown below.

Country/Territory	UN region	IMF ^{[1][13]}	World Bank ^[14]	United Nations ^[15]
Country/Territory	UN region	Forecast	Estimate	Estimate
0 World	—	104476432	100562011	96698005
1 United States	Americas	26949643	25462700	23315081
2 China	Asia	17700899 ^[n 1]	17963171 ^[n 3]	17734131 ^[n 1]
3 Germany	Europe	4429838	4072192	4259935
4 Japan	Asia	4230862	4231141	4940878
...
209 Palau	Oceania	267	—	218
210 Kiribati	Oceania	246	223	227
211 Nauru	Oceania	150	151	155
212 Montserrat	Americas	—	—	72
213 Tuvalu	Oceania	63	60	60

Note that the hyperlink texts have also been retained in the code output.

It is further prudent to point out, that this method exclusively operates only on tabular data extraction. BeautifulSoup library still remains the default method of extracting any kind of information from web pages.

Author(s)

Abhishek Gagneja