

Data Acquisition

Why is Machine Learning Difficult?

- The answer is that the data that required to train computers is most of the times **not available**.
- And in other cases, data is in a **very raw format that requires a lot of cleaning and feature engineering**.

What are some common ways to collect data?

1. Collection from an publicly available data repository (files). ✅
2. Web APIs (JSON)
3. Scraping a website (check for legality).
4. Databases (SQL / NoSQL)

Problem Statement: How to get all tweets with a tag **omicron**?

Get all the tweet for from twitter talking about #omicron

Transition question : **Does Twitter offer some sort of service that allows to query data in this manner?**

WWW

The World Wide Web is about communication between web clients and web servers.

Clients are often browsers (Chrome, Edge, Safari), but they can be any type of program or device.

Servers are most often computers in the cloud.



HTTP Request / Response

Communication between clients and servers is done by requests and responses:

A client (a browser) sends an HTTP request to the web A web server receives the request The server runs an application to process the request The server returns an HTTP response (output) to the browser The client (the browser) receives the response

The HTTP Request Circle

A typical HTTP request / response circle:

The browser requests an HTML page. The server returns an HTML file. The browser requests a style sheet. The server returns a CSS file. The browser requests an JPG image. The server returns a JPG file. The browser requests JavaScript code. The server returns a JS file. The browser requests data. The server returns data (in XML or JSON).

In []:

The GET Method

GET is used to request data from a specified resource.

Note that the query string (name/value pairs) is sent in the URL of a GET request:

```
/test/demo_form.php?name1=value1&name2=value2
```

Some notes on GET requests:

- GET requests can be cached
- GET requests remain in the browser history
- GET requests can be bookmarked
- GET requests should never be used when dealing with sensitive data
- GET requests have length restrictions
- GET requests are only used to request data (not modify)

The POST Method

POST is used to send data to a server to create/update a resource.

The data sent to the server with POST is stored in the request body of the HTTP request:

```
POST /test/demo_form.php HTTP/1.1
Host: w3schools.com

name1=value1&name2=value2
```

Some notes on POST requests:

- POST requests are never cached
- POST requests do not remain in the browser history
- POST requests cannot be bookmarked
- POST requests have no restrictions on data length

In []:

What is an API?

Application Programming Interface (API), is a software that allows two applications to talk to each other (exchanging the data). Each time you check the weather on your phone, or using a Google Service, you're using an API.

APIs are just like the function calls, but those functions are sitting on the web server and API is the way to invoke those functions in your program.

1. We can send a request to the web server (to its API) to get the data.
2. In return if the call is successfully made the API returns us the data mostly in the **json format**.

There are some websites or APIs those offers are open to all and provide free data. Whereas mostly APIs are paid and require some sort of Authentication with the API Keys.

Use cases of APIs:

- APIs can be used to call a function on web server to perform a task and return the required response.
- APIs can also be used to get/send the data over the internet.

Let's first look at few free web APIs and then we will explore paid web api.

How to make these API calls?

Number of things that you should know while making a request on the web:

1. Protocol - HTTP
2. Authentication credentials for the API being called.
3. Functional requirements of the API (URL for the endpoint) - parameters, syntax, and sample response structure (is it JSON or something else) - using the language of your choice.
4. [Optional]: Are there 3rd party solutions available to make these requests easy? - e.g.: `yahoofinance`, `tweepy`

1. How to send an HTTP request using Python?

requests package

In [33]:

```
# !pip install requests
```

In [34]:

```
import requests
import json
```

Get Public IP address - Know your IP address

Another open Web API is to know your public IP Address. Try calling <https://api.ipify.org> (<https://api.ipify.org>) this URL from the address bar of your browser.

In [35]:

```
url = "https://api.ipify.org"
```

In [36]:

```
response = requests.get(url)
```

In [37]:

```
response.status_code
```

Out[37]:

200

In [38]:

```
response.text
```

Out[38]:

'223.190.80.1'

In []:

Coming back to our task: Getting Omicron -tagged tweets using Twitter Search API

Getting tweets tagged Omicron from the Twitter.

1. Set up a project on Twitter Developer API.
2. Create a new app.

In [39]:

```
!pip install tweepy
```

Requirement already satisfied: tweepy in /Users/harshit/miniconda3/envs/dsml_env/lib/python3.9/site-packages (4.5.0)

Requirement already satisfied: requests<3,>=2.27.0 in /Users/harshit/miniconda3/envs/dsml_env/lib/python3.9/site-packages (from tweepy) (2.27.1)

Requirement already satisfied: requests-oauthlib<2,>=1.0.0 in /Users/harshit/miniconda3/envs/dsml_env/lib/python3.9/site-packages (from tweepy) (1.3.1)

Requirement already satisfied: idna<4,>=2.5 in /Users/harshit/miniconda3/envs/dsml_env/lib/python3.9/site-packages (from requests<3,>=2.27.0->tweepy) (3.3)

Requirement already satisfied: charset-normalizer~=2.0.0 in /Users/harshit/miniconda3/envs/dsml_env/lib/python3.9/site-packages (from requests<3,>=2.27.0->tweepy) (2.0.11)

Requirement already satisfied: urllib3<1.27,>=1.21.1 in /Users/harshit/miniconda3/envs/dsml_env/lib/python3.9/site-packages (from requests<3,>=2.27.0->tweepy) (1.26.8)

Requirement already satisfied: certifi>=2017.4.17 in /Users/harshit/miniconda3/envs/dsml_env/lib/python3.9/site-packages (from requests<3,>=2.27.0->tweepy) (2021.10.8)

Requirement already satisfied: oauthlib>=3.0.0 in /Users/harshit/miniconda3/envs/dsml_env/lib/python3.9/site-packages (from requests-oauthlib<2,>=1.0.0->tweepy) (3.2.0)

API v2

- API link: <https://developer.twitter.com/en/docs/twitter-api/tweets/search/introduction>
(<https://developer.twitter.com/en/docs/twitter-api/tweets/search/introduction>)
- Fields to extract: <https://developer.twitter.com/en/docs/twitter-api/data-dictionary/object-model/tweet>
(<https://developer.twitter.com/en/docs/twitter-api/data-dictionary/object-model/tweet>)

Authenticate yourself using the bearer token

In [1]:

```
import tweepy

##add your bearer token
client = tweepy.Client(bearer_token='AAAAAAAAAAAAAAAAAAAAAPPSYwEAAAAARlorVWosZZfuZs5')
```

Extract tweets from an account

In [2]:

```
query = 'from:scaler_official -is:retweet'

tweets = client.search_recent_tweets(query=query, tweet_fields = ['created_at', 'author_id', 'text', 'public_metrics'],
                                     max_results=100)

print(len(tweets.data))
```

22

In [3]:

```
for tweet in tweets.data:
    print(tweet.text)
    print(tweet.author_id)
    print('-----')
```

Meme template here 📌 <https://t.co/ufRlutRacP> (<https://t.co/ufRlutRacP>)
1194875574331699200

Create your own version of this meme and we'll share the best ones. Me me on!

@iAmitBhadana <https://t.co/RJ17dkLGEU> (<https://t.co/RJ17dkLGEU>)
1194875574331699200

@ksridhar02 Thanks! Keep an eye out for more such videos! 😊
1194875574331699200

2022 - Became the first Indian-origin Prime Minister of the UK from 25th October.

1194875574331699200

2022 - Resigned from Boris Johnson's cabinet and launched a bid to succeed him as the UK PM.

1194875574331699200

2020 - During Lockdown, supported the businesses with £330 bn, and the hospitality industry with Eat Out to Help Out scheme.

1194875574331699200

2019 - Appointed as the Chief Secretary to the Treasury and later got promoted to Chancellor of the Exchequer.

1194875574331699200

2015 - Elected as an MP from Richmond, for the environment, food, and rural affairs and got re-elected again after 2 years.

1194875574331699200

2014 - Selected as the Conservative candidate for Richmond.

1194875574331699200

2009 - Got married to Akshata Murthy, daughter of Narayana Murthy.

1194875574331699200

2006 - Gained MBA from Stanford on a Fulbright scholarship and started working for TCI Fund Management.

1194875574331699200

2001 - Graduated in Philosophy, Politics and Economics from Oxford and started working as an Analyst at Goldman Sachs.

1194875574331699200

1980 - Born in Southampton, England to parents of Punjabi descent.

1194875574331699200

Here's the timeline of the youngest PM of the UK.

A thread 📌 <https://t.co/kRRoCUHFhZ> (<https://t.co/kRRoCUHFhZ>)
1194875574331699200

Are you a #DataEngineering enthusiast with a lot of doubts?
Join us for an #AMA session with Shashank Mishra, Data Engineer III at Expedia, live tomorrow at 12 PM on our Instagram stories.

Drop in your questions below and get them answered! <https://t.co/P2YpxbI7aV> (<https://t.co/P2YpxbI7aV>)
1194875574331699200

Check out her full journey:
<https://t.co/LbIKhBLDXs> (<https://t.co/LbIKhBLDXs>)
1194875574331699200

From Kanpur to SDE 2 at Zeta Suite, Nidhi's journey is that of learning and practicing day in and day out. Eventually landing herself a job that drives her vision to creating an impact.

#SCALER <https://t.co/OWNP3DAwo5> (<https://t.co/OWNP3DAwo5>)
1194875574331699200

Did you know? 🤔
#DataScientist and #MLEngineer are two different roles that require different skill sets.

Watch this video to get a quick view of what skills are vital for both roles.

#CreateImpact <https://t.co/bCUr6t2iI9> (<https://t.co/bCUr6t2iI9>)
1194875574331699200

@yashgau94053618 Hi Yash, we've gone ahead and taken the necessary steps to resolve your concern from our end. And we've shared the relevant information regarding the same with you over DM. Please be assured that this issue has been taken care of.
1194875574331699200

@swapnilkhurana Hi Swapnil, this is not what we stand for & we'd like to highlight this internally. We appreciate you taking the time to bring this forward & our apologies once again for the experience you've had to endure. Please DM us your contact details, we'll get in touch with you to assist <https://t.co/5hQ0ZeV3xn> (<https://t.co/5hQ0ZeV3xn>)
1194875574331699200

That's where it all started 😍

#SCALER #Childrensday <https://t.co/weOjV2zXX5> (<https://t.co/weOjV2zXX5>)
1194875574331699200

No cache, no problem ✨

#SCALERComics #ProgrammingHumor <https://t.co/BiQxjrMrdR> (<https://t.co/BiQxjrMrdR>)
1194875574331699200

In [4]:

```
##. getting twitter user id from username
client.get_user(username="elonmusk")
```

Out[4]:

```
Response(data=<User id=44196397 name=Elon Musk username=elonmusk>, includes={}, errors=[], meta={})
```

In [5]:

```
## get the following list of Elon Musk

fol_ing_list = client.get_users_following(id="44196397")
len(fol_ing_list.data)
```

Out[5]:

100

In [6]:

```
fol_ing_list.data[0]
```

Out[6]:

```
<User id=15485441 name=Jimmy Fallon username=jimmyfallon>
```

In [7]:

```
query = "#omicron -is:retweet"

data = {
    'text': [],
    'created_at': [],
    'author_id': []
}

for tweet in tweepy.Paginator(client.search_recent_tweets, query=query,
                               tweet_fields=['created_at', 'author_id']).flatten(1000):
    data['text'].append(tweet.text)
    data['created_at'].append(tweet.created_at)
    data['author_id'].append(tweet.author_id)
```

In [8]:

```
len(data['text'])
```

Out[8]:

1000

In [9]:

```
import pandas as pd

df = pd.DataFrame(data)
df.head()
```

Out[9]:

| | text | created_at | author_id |
|---|--|------------------------------|---------------------|
| 0 | More #Omicron Bullsh\$t... https://t.co/J9Hza32V7y | 2022-11-18 15:08:56+00:00 | 1445213314561323013 |
| 1 | #新冠病毒 #OMICRON 子变体 #BA5 现在美国只占30% #新冠 病例, 不再占据优... | 2022-11-18 15:05:21+00:00 | 826274564741017600 |
| 2 | ✗ #Falso que la cepa #Ómicron XBB sea más grave... | 2022-11-18 15:00:01+00:00 | 1277632595040272385 |
| 3 | ১৮ নভেম্বরের কোভিড বুলেটিন। দেখে নিন এক নজরে।n... | 2022-11-18 15:00:00+00:00 | 1192758232890003457 |
| 4 | #Healthcare News: More than two million people... | 2022-11-18 14:55:11+00:00 | 1419346168585326595 |

In [10]:

```
df.to_csv("tweets.csv", index=False)
```

Extract tweets from an account

In [50]:

```
# Replace with your own search query
query = 'from:scaler_official -is:retweet'

tweets = client.search_recent_tweets(query=query, tweet_fields=['created_at', 'author_id'])

print(len(tweets.data))
print("=====")

for tweet in tweets.data:
    print(tweet.text)
    print("-----")
    if len(tweet.context_annotations) > 0:
        print(tweet.author_id)
```

38

=====

"Sorry sir, we've canceled your flight. Please accept a gift card worth \$\$\$\$." But why? 😞
The reason usually is OVERBOOKING.

Here's a peep at EP 4 of #DataBro, where Guru shows how data plays a big role in overbooked flights.

#SCALEROriginals @thejordindian <https://t.co/hz7hJaBGKQ> (<https://t.co/hz7hJaBGKQ>)

Try these tips at your next #hackathon!

Comment below if you have more useful tips 📌

#CreateImpact

Takeaways from Saumya's hackathon preparation ¹⁰⁰ <https://t.co/wsV0sG9Xe>

In [51]:

```
## get ID from the username
client.get_user(username="elonmusk")
```

Out[51]:

```
Response(data=<User id=44196397 name=Elon Musk username=elonmusk>, includes={}, errors=[], meta={})
```

In [52]:

```
## get users that I am following
client.get_users_following(id="709107001926615040")
```

Out[52]:

```
Response(data=[<User id=38899543 name=Chrisman username=chrismanfrank
>, <User id=1345707144943906816 name=Vaibhav Domkundwar username=vaibh
avbetter>, <User id=1450424698853072897 name=Alex Mathers username=Ale
xmathers84>, <User id=965699949512900608 name=Clint Murphy username=IA
mClintMurphy>, <User id=360254234 name=JK MOLINA username=OneJKMolina
>, <User id=1435768999187558400 name=Joe Portsmouth 🇺🇸 username=joe_po
rtsmouth>, <User id=1511333035 name=Andrew Gazdecki username=agazdecki
>, <User id=343990983 name=Arvid Kahl username=arvidkahl>, <User id=92
0321515077414912 name=Readwise username=readwise>, <User id=2722375254
name=Charlotte Crowther 🌸 username=CharlotCrowther>, <User id=1245260
977626587136 name=Jay Alammar username=JayAlammar>, <User id=544816363
name=Naman Bhalla username=Naman_Bhalla>, <User id=1291505170581319680
name=Synthesis username=synthesischool>, <User id=14826955 name=Abhima
nyu Saxena username=asxna>, <User id=1191207924 name=Erik Bernhardsson
username=bernhardsson>, <User id=441465751 name=Tanishq Mathew Abraham
username=iScienceLuvr>, <User id=2286218053 name=Shreya Shankar userna
me=sh_reya>, <User id=2621989106 name=Yang Song username=YSongStanford
>, <User id=75773685 name=madhavan malolan username=madhavanmalolan>,
<User id=3448284313 name=elvis username=omarsar0>, <User id=97707247 n
ame=Gautam Kamath username=thegautamkamath>, <User id=1202267633049100
291 name=merve 🧡 username=mervenoyann>, <User id=897534330079072256 n
ame=Aishwarya Kamath username=ashkamath20>, <User id=13383937385115893
77 name=labml.ai username=labmlai>, <User id=907007346546810881 name=A
leksa Gordić username=gordic_aleksa>, <User id=895328122924564480 name
=OpenMined username=openminedorg>, <User id=1018787154695213056 name=K
ritika Prakash username=kritipraks>, <User id=1605 name=Sam Altman use
rname=sama>, <User id=35473191 name=Mark Saroufim username=marksaroufi
m>, <User id=481647429 name=Code2040 username=Code2040>, <User id=1310
983355215740928 name=Billy Broas username=BillyBroas>, <User id=168805
139 name=Yannic Kilcher, Tech Sister username=ykilcher>, <User id=1965
8826 name=New Scientist username=newscientist>, <User id=4416456732 na
me=Andrew D. Huberman, Ph.D. username=hubermanlab>, <User id=796584325
name=Alfredo Canziani username=alfcnz>, <User id=1282121312 name=Tim U
rban username=waitbutwhy>, <User id=312681953 name=Sahil Bloom usernam
e=SahilBloom>, <User id=1180699114612215809 name=Commudle username=com
mudle>, <User id=1381544381509947393 name=Papers with Datasets usernam
e=paperswithdata>, <User id=331827860 name=Josh Tobin username=josh_to
bin_>, <User id=1237312746 name=Rosanne Liu username=savvyRL>, <User i
d=1254910635906789378 name=ML Collective username=ml_collective>, <Use
r id=1085303366253633536 name=Full Stack Deep Learning username=full_s
tack_dl>, <User id=3259586191 name=Goku Mohandas username=GokuMohandas
>, <User id=1127786941422792704 name=Made With ML username=MadeWithML
>, <User id=221938964 name=Jerome Pesenti username=an_open_mind>, <Use
r id=593407733 name=Sayak Paul username=RisingSayak>, <User id=1464233
1 name=GREG ISENBERG username=gregisenberg>, <User id=400636259 name=D
mitry Petrov username=FullStackML>, <User id=14417215 name=Ryan Hoover
username=rrhoover>, <User id=14931736 name=Rob Hope 🇧🇷 username=robhop
e>, <User id=14372143 name=Jason Fried username=jasonfried>, <User id=
1118579966 name=Nikhil Thorat username=nsthorat>, <User id=15278016 na
me=Chip Huyen username=chipro>, <User id=629844555 name=Nik Sharma use
rname=mrsharma>, <User id=205199260 name=Alok Jain ⚡ username=Weekend
Investng>, <User id=3070985713 name=NFX username=NFX>, <User id=117218
5038 name=Chad Scherrer username=ChadScherrer>, <User id=8500962 name=
```

```
Venkatesh Rao username=vgr>, <User id=101354595 name=Shaan Puri username=ShaanVP>, <User id=861961251961749504 name=Jatin Mittal username=iamjaykm>, <User id=4407451109 name=Kyle Prinsloo username=kylepdotco>, <User id=293815131 name=Julian Shapiro username=Julian>, <User id=64844802 name=al6z username=al6z>, <User id=381813372 name=Joe Norman username=normonics>, <User id=636513296 name=Nikita Bier username=nikitabier>, <User id=13418472 name=Willy Woo username=woonomic>, <User id=5252591 name=Bill Gurley username=bgurley>, <User id=15370019 name=Josh Wolfe username=wolfejosh>, <User id=284278132 name=Morgan Housel username=morganhousel>, <User id=968341850145918981 name=Xavier Helgesen 🌍📖🇺🇦 username=XavierHelgesen>, <User id=20263710 name=Keith Rabois username=rabois>, <User id=133818287 name=Wes Kao 🏠 username=wes_kao>, <User id=2239670346 name=Jonathan Frankle username=jefrankle>, <User id=28131948 name=Joscha Bach username=Plinz>, <User id=1189487223642697728 name=Your Daily Epsilon of Math username=Daily_Epsilon>, <User id=491821358 name=Emma Bostian username=EmmaBostian>, <User id=337119125 name=Jane Manchun Wong username=wongmjane>, <User id=373008565 name=Rong Lu username=davorabbit>, <User id=381289719 name=Nassim Nicholas Taleb username=nntaleb>, <User id=17797942 name=A Simple Life username=position_trader>, <User id=3511430425 name=Fermat's Library username=fermatlibrary>, <User id=229339862 name=Terence Parr username=the_antlr_guy>, <User id=866469276978364416 name=Matt Gardner username=nlpmattg>, <User id=16141659 name=Julien Chaumond username=julien_c>, <User id=186420551 name=clem 😊 username=ClementDelangue>, <User id=778764142412984320 name=Hugging Face username=huggingface>, <User id=326295039 name=Vincent D. Warmerdam username=fishnets88>, <User id=851773048973807617 name=The Stoic Emperor username=TheStoicEmperor>, <User id=32379544 name=Ankur Warikoo username=warikoo>, <User id=14832563 name=Paras Chopra username=paraschopra>, <User id=33104659 name=Shane Parrish username=ShaneAParrish>, <User id=745273 name=Naval username=naval>, <User id=168987151 name=Derek Muller username=veritasium>, <User id=29873662 name=Marques Brownlee username=MKBHD>, <User id=2178012643 name=Balaji Srinivasan username=balajis>, <User id=865622395 name=Sebastian Raschka username=rasbt>, <User id=5011881 name=Will Kurt username=willkurt>, <User id=254791849 name=Scientific Python username=SciPyTip>, <User id=2877269376 name=Grant Sanderson username=3blue1brown>], includes={}, errors=[], meta={'result_count': 100, 'next_token': 'KECRKGVPRP21EZZZ'})
```

Query tweets with tag - Omicron

In [53]:

```
# Replace with your own search query
query = '#omicron'

data = {'text': [],
        'created_at': [],
        'author_id': []}

# Replace the limit=1000 with the maximum number of Tweets you want
for tweet in tweepy.Paginator(client.search_recent_tweets, query=query,
                              tweet_fields=['context_annotations', 'created_at', 'au
data['text'].append(tweet.text)
data['created_at'].append(tweet.created_at)
data['author_id'].append(tweet.author_id)

print(data)
```

```
{'text': ['#COVID19 Daily global cases fell after a spike in the sprin
g but are now rising again, with the emergence of the BA.4 and BA.5 su
bvariants of the #Omicron variant. https://t.co/EDIMWykytZ', (https://t.co/EDIMWykytZ), 'Reassuring news about #SARSCoV2 #omicron BA.4 and
BA.5 infections in England: \n\n"There was no evidence of reduced vac
cine effectiveness against hospitalisation for BA.4 or BA.5 as compare
d to BA.2"\n\nPreprint from our team @UKHSA 🙌 https://t.co/ePmhr71lul
\n\nMore (https://t.co/ePmhr71lul\n\nMore) details here 🙌 https://t.c
o/aAkbItKcgv', (https://t.co/aAkbItKcgv), 'The official cited the sit
uation in #India as an example, where the #Omicron subvariant BA2.75 i
s spreading faster than other variants, according to a media report. h
https://t.co/NSk2ayVlrj', (https://t.co/NSk2ayVlrj), '"According to th
e data, #Europe is still experiencing the circulation of the #Omicron
BA.5 variant, although the peak of the summer wave was a few weeks ag
o in many of the member states," said Marco Cavaleri, Head of the EMA
\'s health threats and vaccines strategy. https://t.co/30vCPF07RG', (h
https://t.co/30vCPF07RG), 'The latest The Creativeworld Dave Daily! ht
https://t.co/pVu5KTXkks (https://t.co/pVu5KTXkks) Thanks to @Deanmadalba
#omicron #breaking', '#LongCovid in #England 🇬🇧:\n>2 % der gesamten
...
```

In [54]:

```
import pandas as pd
df = pd.DataFrame(data)
df.head()
```

Out[54]:

| | text | created_at | author_id |
|---|--|------------------------------|---------------------|
| 0 | #COVID19 Daily global cases fell after a spike... | 2022-09-03 08:41:42+00:00 | 1202820866595123200 |
| 1 | Reassuring news about #SARSCoV2 #omicron BA.4 ... | 2022-09-03 08:35:28+00:00 | 1184232077933383682 |
| 2 | The official cited the situation in #India as ... | 2022-09-03 08:32:02+00:00 | 834013638323224576 |
| 3 | "According to the data, #Europe is still exper... | 2022-09-03 08:31:59+00:00 | 834013638323224576 |
| 4 | The latest The Creativeworld Dave Daily! https... | 2022-09-03 08:28:16+00:00 | 168617320 |

Writing to a file

In [55]:

```
df.to_csv('./sample_tweets.csv', index=False)
```

Getting the Universities Data

This is the first type of API where just by calling the API (<http://universities.hipolabs.com>) as it is will not give the data. We need to provide the further more information to the API. Something like what we do while calling functions i.e. - passing the parameters.

Here we will pass query parameters while calling the API.

URL = <http://universities.hipolabs.com/search?name=Amity&country=India>
(<http://universities.hipolabs.com/search?name=Amity&country=India>). We can pass the name and the country

In [56]:

```
url = "http://universities.hipolabs.com/search?name=Amity&country=India"
```

In [57]:

```
response = requests.get(url)
```

In [58]:

```
response.status_code
```

Out[58]:

200

In [59]:

```
amity = json.loads(response.text)
```

In [60]:

```
amity
```

Out[60]:

```
[{'domains': ['amity.edu'],  
  'alpha_two_code': 'IN',  
  'country': 'India',  
  'web_pages': ['http://www.amity.edu/'],  
  'name': 'Amity University',  
  'state-province': None}]
```

If we want to get all the universities of India, we can skip the name parameter and just provide the country=India

<http://universities.hipolabs.com/search?country=India> (<http://universities.hipolabs.com/search?country=India>)

In [61]:

```
url = "http://universities.hipolabs.com/search?country=India"
```

In [62]:

```
response = requests.get(url)
print(response.status_code)
countries = json.loads(response.text)
```

200

In [63]:

```
# Got a list of Universities
type(countries)
```

Out[63]:

list

In [64]:

```
# There are total of 800 university data.
len(countries)
```

Out[64]:

806

In [65]:

countries[:5]

Out[65]:

```
[{'domains': ['upes.ac.in'],
  'alpha_two_code': 'IN',
  'country': 'India',
  'web_pages': ['https://www.upes.ac.in/'],
  'name': 'University of Petroleum and Energy Studies',
  'state-province': 'Dehradun'},
 {'domains': ['davietjal.org'],
  'alpha_two_code': 'IN',
  'country': 'India',
  'web_pages': ['http://www.davietjal.org/'],
  'name': 'DAV Institute of Engineering & Technology',
  'state-province': 'Punjab'},
 {'domains': ['lpu.in'],
  'alpha_two_code': 'IN',
  'country': 'India',
  'web_pages': ['http://www.lpu.in/'],
  'name': 'Lovely Professional University',
  'state-province': 'Punjab'},
 {'domains': ['somaiya.edu'],
  'alpha_two_code': 'IN',
  'country': 'India',
  'web_pages': ['https://somaiya.edu/'],
  'name': 'Somaiya Vidyavihar',
  'state-province': None},
 {'domains': ['ncuindia.edu'],
  'alpha_two_code': 'IN',
  'country': 'India',
  'web_pages': ['http://www.ncuindia.edu/'],
  'name': 'NorthCap University',
  'state-province': 'Haryana'}]
```

OpenWeatherMap

The open weather map API provides us the data about the weather provided the region or lat,long or other parameters. Since the API is not open for all and it requires some kind of authentication using the API Key.

Steps to get a Key:

- Create an account on <https://openweathermap.org/> (<https://openweathermap.org/>).
- Visit https://home.openweathermap.org/api_keys (https://home.openweathermap.org/api_keys).
- Create a New Key.

This API key will be sent for each API calling to authenticate a real user.

I can get the weather information about Delhi about passing `q=delhi` which means query.

In [66]:

```
url = "https://api.openweathermap.org/data/2.5/weather?q=delhi&appid=9b199c2b6cd2fbc"
```


In [67]:

```
response = requests.get(url)
```

In [68]:

```
weather = json.loads(response.text)
```

In [69]:

```
weather
```

Out[69]:

```
{'coord': {'lon': 77.2167, 'lat': 28.6667},  
 'weather': [{'id': 721,  
               'main': 'Haze',  
               'description': 'haze',  
               'icon': '50d'}],  
 'base': 'stations',  
 'main': {'temp': 309.2,  
           'feels_like': 311.71,  
           'temp_min': 309.2,  
           'temp_max': 309.2,  
           'pressure': 1005,  
           'humidity': 38},  
 'visibility': 5000,  
 'wind': {'speed': 4.63, 'deg': 290},  
 'clouds': {'all': 40},  
 'dt': 1662194117,  
 'sys': {'type': 1,  
         'id': 9165,  
         'country': 'IN',  
         'sunrise': 1662165010,  
         'sunset': 1662210691},  
 'timezone': 19800,  
 'id': 1273294,  
 'name': 'Delhi',  
 'cod': 200}
```

The API gives you a lot of information like the coordinates, weather information etc.

Since we're using a free tier of the API, we will not get very accurate data every hour and also there is a limit on the number of API calls you can make. If you intend to make a productionized Machine Learning model, then it's preferred to go for the paid versions of the API.

In []:

In []:

Scraping Books data from a website using BeautifulSoup

- Website: <http://books.toscrape.com/> (<http://books.toscrape.com/>)
- Go to Travel category listed in the left panel.
- We want to collect data about all of these pages.
- Inspect and walk them through HTML structure.

BeautifulSoup is a parsing library which also does a pretty good job of fetching contents from URL and allows you to parse certain parts of them without any hassle.

In [70]:

```
##link that we'll scrape
baseurl = "http://books.toscrape.com/"

r = requests.get(baseurl)

r.content
```

Out[70]:

```
b'<!DOCTYPE html>\n<!--[if lt IE 7]>      <html lang="en-us" class="no
-js lt-ie9 lt-ie8 lt-ie7"> <![endif]-->\n<!--[if IE 7]>      <html
lang="en-us" class="no-js lt-ie9 lt-ie8"> <![endif]-->\n<!--[if IE 8]>
<html lang="en-us" class="no-js lt-ie9"> <![endif]-->\n<!--[if gt IE
8]><!--> <html lang="en-us" class="no-js"> <!--<![endif]-->\n      <head
>\n          <title>\n      All products | Books to Scrape - Sandbox\n</ti
tle>\n\n          <meta http-equiv="content-type" content="text/html; ch
arset=UTF-8" />\n          <meta name="created" content="24th Jun 2016 0
9:29" />\n          <meta name="description" content="" />\n          <met
a name="viewport" content="width=device-width" />\n          <meta name
="robots" content="NOARCHIVE,NOCACHE" />\n\n          <!-- Le HTML5 shi
m, for IE6-8 support of HTML elements -->\n          <!--[if lt IE 9]>\n
<script src="//html5shim.googlecode.com/svn/trunk/html5.js"></script>
\n          <![endif]-->\n\n          \n          <link rel="shortcut ic
on" href="static/oscar/favicon.ico" />\n          \n\n          \n
\n          \n          <link rel="stylesheet" type="text/css" href="sta
tic/oscar/css/styles.css" />\n          \n          <link rel="stylesheet" href="s
tatic/oscar/is/bootstrap-datetimepicker/bootstrap-datetimepicker.css"
```

In [71]:

```
!pip install bs4
from bs4 import BeautifulSoup
```

```
Requirement already satisfied: bs4 in /Users/harshit/miniconda3/envs/d
sml_env/lib/python3.9/site-packages (0.0.1)
Requirement already satisfied: beautifulsoup4 in /Users/harshit/minico
nda3/envs/dsml_env/lib/python3.9/site-packages (from bs4) (4.10.0)
Requirement already satisfied: soupsieve>1.2 in /Users/harshit/minicon
da3/envs/dsml_env/lib/python3.9/site-packages (from beautifulsoup4->bs
4) (2.3.1)
```

In [72]:

```
##creating a soup object of all the HTML content
soup = BeautifulSoup(r.content)

##parsing the HTML to find the ul tag and get all the li tags inside.
ul_list = soup.find('ul', class_="nav-list").ul.find_all("li")

##creating the link using the href attribute from the anchor tag
baseurl + ul_list[0].a['href']
```

Out[72]:

```
'http://books.toscrape.com/catalogue/category/books/travel_2/index.htm
l'
```

In [73]:

```
def extract_categories(baseurl):
    """
    Extract all the categories and store their links
    corresponding to their category names.
    """
    categories = {}
    r = requests.get(baseurl)
    soup = BeautifulSoup(r.content, features="html.parser")
    categories_lists = soup.find("ul", class_="nav-list").ul.find_all("li")
    for li in categories_lists:
        categories.update({li.text.strip(): baseurl + li.a['href']})
    return categories
```

In [74]:

```
extract_categories(baseurl)
```

Out[74]:

```
{'Travel': 'http://books.toscrape.com/catalogue/category/books/travel_2/index.html',  
 'Mystery': 'http://books.toscrape.com/catalogue/category/books/mystery_3/index.html',  
 'Historical Fiction': 'http://books.toscrape.com/catalogue/category/books/historical-fiction_4/index.html',  
 'Sequential Art': 'http://books.toscrape.com/catalogue/category/books/sequential-art_5/index.html',  
 'Classics': 'http://books.toscrape.com/catalogue/category/books/classics_6/index.html',  
 'Philosophy': 'http://books.toscrape.com/catalogue/category/books/philosophy_7/index.html',  
 'Romance': 'http://books.toscrape.com/catalogue/category/books/romance_8/index.html',  
 'Womens Fiction': 'http://books.toscrape.com/catalogue/category/books/womens-fiction_9/index.html',  
 'Fiction': 'http://books.toscrape.com/catalogue/category/books/fiction_10/index.html',  
 'Childrens': 'http://books.toscrape.com/catalogue/category/books/childrens_11/index.html',  
 'Religion': 'http://books.toscrape.com/catalogue/category/books/religion_12/index.html',  
 'Nonfiction': 'http://books.toscrape.com/catalogue/category/books/nonfiction_13/index.html',  
 'Music': 'http://books.toscrape.com/catalogue/category/books/music_14/index.html',  
 'Default': 'http://books.toscrape.com/catalogue/category/books/default_15/index.html',  
 'Science Fiction': 'http://books.toscrape.com/catalogue/category/books/science-fiction_16/index.html',  
 'Sports and Games': 'http://books.toscrape.com/catalogue/category/books/sports-and-games_17/index.html',  
 'Add a comment': 'http://books.toscrape.com/catalogue/category/books/add-a-comment_18/index.html',  
 'Fantasy': 'http://books.toscrape.com/catalogue/category/books/fantasy_19/index.html',  
 'New Adult': 'http://books.toscrape.com/catalogue/category/books/new-adult_20/index.html',  
 'Young Adult': 'http://books.toscrape.com/catalogue/category/books/young-adult_21/index.html',  
 'Science': 'http://books.toscrape.com/catalogue/category/books/science_22/index.html',  
 'Poetry': 'http://books.toscrape.com/catalogue/category/books/poetry_23/index.html',  
 'Paranormal': 'http://books.toscrape.com/catalogue/category/books/paranormal_24/index.html',  
 'Art': 'http://books.toscrape.com/catalogue/category/books/art_25/index.html',  
 'Psychology': 'http://books.toscrape.com/catalogue/category/books/psychology_26/index.html',  
 'Autobiography': 'http://books.toscrape.com/catalogue/category/books/autobiography_27/index.html',  
 'Parenting': 'http://books.toscrape.com/catalogue/category/books/parenting_28/index.html',  
 'Adult Fiction': 'http://books.toscrape.com/catalogue/category/books/
```

```
adult-fiction_29/index.html',
'Humor': 'http://books.toscrape.com/catalogue/category/books/humor_30/index.html',
'Horror': 'http://books.toscrape.com/catalogue/category/books/horror_31/index.html',
'History': 'http://books.toscrape.com/catalogue/category/books/history_32/index.html',
'Food and Drink': 'http://books.toscrape.com/catalogue/category/books/food-and-drink_33/index.html',
'Christian Fiction': 'http://books.toscrape.com/catalogue/category/books/christian-fiction_34/index.html',
'Business': 'http://books.toscrape.com/catalogue/category/books/business_35/index.html',
'Biography': 'http://books.toscrape.com/catalogue/category/books/biography_36/index.html',
'Thriller': 'http://books.toscrape.com/catalogue/category/books/thriller_37/index.html',
'Contemporary': 'http://books.toscrape.com/catalogue/category/books/contemporary_38/index.html',
'Spirituality': 'http://books.toscrape.com/catalogue/category/books/spirituality_39/index.html',
'Academic': 'http://books.toscrape.com/catalogue/category/books/academic_40/index.html',
'Self Help': 'http://books.toscrape.com/catalogue/category/books/self-help_41/index.html',
'Historical': 'http://books.toscrape.com/catalogue/category/books/historical_42/index.html',
'Christian': 'http://books.toscrape.com/catalogue/category/books/christian_43/index.html',
'Suspense': 'http://books.toscrape.com/catalogue/category/books/suspense_44/index.html',
'Short Stories': 'http://books.toscrape.com/catalogue/category/books/short-stories_45/index.html',
'Novels': 'http://books.toscrape.com/catalogue/category/books/novels_46/index.html',
'Health': 'http://books.toscrape.com/catalogue/category/books/health_47/index.html',
'Politics': 'http://books.toscrape.com/catalogue/category/books/politics_48/index.html',
'Cultural': 'http://books.toscrape.com/catalogue/category/books/cultural_49/index.html',
'Erotica': 'http://books.toscrape.com/catalogue/category/books/erotica_50/index.html',
'Crime': 'http://books.toscrape.com/catalogue/category/books/crime_51/index.html'}
```

In []:

Extracting all book info in each of these categories.

In [75]:

```
url = "http://books.toscrape.com/catalogue/category/books/mystery_3/index.html"
res = requests.get(url)

soup = BeautifulSoup(res.content)
```

In [76]:

soup

Out[76]:

```
<!DOCTYPE html>
<!--[if lt IE 7]>      <html lang="en-us" class="no-js lt-ie9 lt-ie8 lt-ie7"> <![endif]--><!--[if IE 7]>      <html lang="en-us" class="no-js lt-ie9 lt-ie8"> <![endif]--><!--[if IE 8]>      <html lang="en-us" class="no-js lt-ie9"> <![endif]--><!--[if gt IE 8]><!--><html class="no-js" lang="en-us"> <!--<![endif]-->
<head>
<title>
    Mystery |
    Books to Scrape - Sandbox

</title>
<meta content="text/html; charset=utf-8" http-equiv="content-type"/>
<meta content="24th Jun 2016 09:29" name="created"/>
<meta content="
" name="description"/>
<meta content="width=device-width" name="viewport" />
```

In [77]:

```
data = {
    'product_page_url': [],
    'title': [],
    'price_including_tax': [],
    'number_available': []
}
```

In [78]:

```
book_page_list = soup.find('ol', class_="row").find_all("li")
book_page_list
```

Out[78]:

```
[<li class="col-xs-6 col-sm-4 col-md-3 col-lg-3">
  <article class="product_pod">
    <div class="image_container">
      <a href="../../sharpp-objects_997/index.html"></a>
    </div>
    <p class="star-rating Four">
      <i class="icon-star"></i>
      <i class="icon-star"></i>
      <i class="icon-star"></i>
      <i class="icon-star"></i>
      <i class="icon-star"></i>
    </p>
    <h3><a href="../../sharpp-objects_997/index.html" title="Sharp Objects">Sharp Objects</a></h3>
    <div class="product_price">
      <p class="price color">£47.82</p>
```

In [79]:

```
book_page_list[0].a['href'].split("/")[3:]
```

Out[79]:

```
['sharp-objects_997', 'index.html']
```

In [80]:

```

for li in book_page_list:
    href_split = li.a['href'].split("/")[3:]
    book_page_link = "http://books.toscrape.com/catalogue/" + "/".join(href_split)

    print(book_page_link)

    req = requests.get(book_page_link)
    soup = BeautifulSoup(req.content, features="html.parser")

    data['product_page_url'].append(book_page_link)

    rows = soup.find("table").find_all("tr")

    title = soup.find('title').text.strip()
    data['title'].append(title)

    data['price_including_tax'].append(rows[3].find('td').text.strip())
    data['number_available'].append(rows[5].find('td').text.strip())

```

```

http://books.toscrape.com/catalogue/sharp-objects_997/index.html (http://books.toscrape.com/catalogue/sharp-objects_997/index.html)
http://books.toscrape.com/catalogue/in-a-dark-dark-wood_963/index.html (http://books.toscrape.com/catalogue/in-a-dark-dark-wood_963/index.html)
http://books.toscrape.com/catalogue/the-past-never-ends_942/index.html (http://books.toscrape.com/catalogue/the-past-never-ends_942/index.html)
http://books.toscrape.com/catalogue/a-murder-in-time_877/index.html (http://books.toscrape.com/catalogue/a-murder-in-time_877/index.html)
http://books.toscrape.com/catalogue/the-murder-of-roger-ackroyd-hercule-poirot-4_852/index.html (http://books.toscrape.com/catalogue/the-murder-of-roger-ackroyd-hercule-poirot-4_852/index.html)
http://books.toscrape.com/catalogue/the-last-mile-amos-decker-2_754/index.html (http://books.toscrape.com/catalogue/the-last-mile-amos-decker-2_754/index.html)
http://books.toscrape.com/catalogue/that-darkness-gardiner-and-renner-1_743/index.html (http://books.toscrape.com/catalogue/that-darkness-gardiner-and-renner-1_743/index.html)
http://books.toscrape.com/catalogue/tastes-like-fear-di-marnie-rome-3_742/index.html (http://books.toscrape.com/catalogue/tastes-like-fear-di-marnie-rome-3_742/index.html)
http://books.toscrape.com/catalogue/a-time-of-torment-charlie-parker-14_657/index.html (http://books.toscrape.com/catalogue/a-time-of-torment-charlie-parker-14_657/index.html)
http://books.toscrape.com/catalogue/a-study-in-scarlet-sherlock-holmes-1_656/index.html (http://books.toscrape.com/catalogue/a-study-in-scarlet-sherlock-holmes-1_656/index.html)
http://books.toscrape.com/catalogue/poisonous-max-revere-novels-3_627/index.html (http://books.toscrape.com/catalogue/poisonous-max-revere-novels-3_627/index.html)
http://books.toscrape.com/catalogue/murder-at-the-42nd-street-library-raymond-ambler-1_624/index.html (http://books.toscrape.com/catalogue/murder-at-the-42nd-street-library-raymond-ambler-1_624/index.html)
http://books.toscrape.com/catalogue/most-wanted_623/index.html (http://books.toscrape.com/catalogue/most-wanted_623/index.html)
http://books.toscrape.com/catalogue/hide-away-eve-duncan-20_620/index.html (http://books.toscrape.com/catalogue/hide-away-eve-duncan-20_620/index.html)

```



```
index.html)
http://books.toscrape.com/catalogue/boar-island-anna-pigeon-19_613/index.html (http://books.toscrape.com/catalogue/boar-island-anna-pigeon-19_613/index.html)
http://books.toscrape.com/catalogue/the-widow_609/index.html (http://books.toscrape.com/catalogue/the-widow_609/index.html)
http://books.toscrape.com/catalogue/playing-with-fire_602/index.html (http://books.toscrape.com/catalogue/playing-with-fire_602/index.html)
http://books.toscrape.com/catalogue/what-happened-on-beale-street-secrets-of-the-south-mysteries-2_506/index.html (http://books.toscrape.com/catalogue/what-happened-on-beale-street-secrets-of-the-south-mysteries-2_506/index.html)
http://books.toscrape.com/catalogue/the-bachelor-girls-guide-to-murder-herringford-and-watts-mysteries-1_491/index.html (http://books.toscrape.com/catalogue/the-bachelor-girls-guide-to-murder-herringford-and-watts-mysteries-1_491/index.html)
http://books.toscrape.com/catalogue/delivering-the-truth-quaker-midwife-mystery-1_464/index.html (http://books.toscrape.com/catalogue/delivering-the-truth-quaker-midwife-mystery-1_464/index.html)
```

In []:

In [81]:

data

Out[81]:

```
{'product_page_url': ['http://books.toscrape.com/catalogue/sharp-objects_997/index.html',
  'http://books.toscrape.com/catalogue/in-a-dark-dark-wood_963/index.html',
  'http://books.toscrape.com/catalogue/the-past-never-ends_942/index.html',
  'http://books.toscrape.com/catalogue/a-murder-in-time_877/index.html',
  'http://books.toscrape.com/catalogue/the-murder-of-roger-ackroyd-hercule-poirot-4_852/index.html',
  'http://books.toscrape.com/catalogue/the-last-mile-amos-decker-2_754/index.html',
  'http://books.toscrape.com/catalogue/that-darkness-gardiner-and-renner-1_743/index.html',
  'http://books.toscrape.com/catalogue/tastes-like-fear-di-marnie-rome-3_742/index.html',
  'http://books.toscrape.com/catalogue/a-time-of-torment-charlie-parke-14_657/index.html',
  'http://books.toscrape.com/catalogue/a-study-in-scarlet-sherlock-holmes-1_656/index.html',
  'http://books.toscrape.com/catalogue/poisonous-max-revere-novels-3_627/index.html',
  'http://books.toscrape.com/catalogue/murder-at-the-42nd-street-library-raymond-ambler-1_624/index.html',
  'http://books.toscrape.com/catalogue/most-wanted_623/index.html',
  'http://books.toscrape.com/catalogue/hide-away-eve-duncan-20_620/index.html',
  'http://books.toscrape.com/catalogue/boar-island-anna-pigeon-19_613/index.html',
  'http://books.toscrape.com/catalogue/the-widow_609/index.html',
  'http://books.toscrape.com/catalogue/playing-with-fire_602/index.html',
  'http://books.toscrape.com/catalogue/what-happened-on-beale-street-secrets-of-the-south-mysteries-2_506/index.html',
  'http://books.toscrape.com/catalogue/the-bachelor-girls-guide-to-murder-herringford-and-watts-mysteries-1_491/index.html',
  'http://books.toscrape.com/catalogue/delivering-the-truth-quaker-midwife-mystery-1_464/index.html'],
'title': ['Sharp Objects | Books to Scrape - Sandbox',
  'In a Dark, Dark Wood | Books to Scrape - Sandbox',
  'The Past Never Ends | Books to Scrape - Sandbox',
  'A Murder in Time | Books to Scrape - Sandbox',
  'The Murder of Roger Ackroyd (Hercule Poirot #4) | Books to Scrape - Sandbox',
  'The Last Mile (Amos Decker #2) | Books to Scrape - Sandbox',
  'That Darkness (Gardiner and Renner #1) | Books to Scrape - Sandbox',
  'Tastes Like Fear (DI Marnie Rome #3) | Books to Scrape - Sandbox',
  'A Time of Torment (Charlie Parker #14) | Books to Scrape - Sandbox',
  'A Study in Scarlet (Sherlock Holmes #1) | Books to Scrape - Sandbox',
  'Poisonous (Max Revere Novels #3) | Books to Scrape - Sandbox',
  'Murder at the 42nd Street Library (Raymond Ambler #1) | Books to Scrape - Sandbox']
```

```

'Most Wanted | Books to Scrape - Sandbox',
'Hide Away (Eve Duncan #20) | Books to Scrape - Sandbox',
'Boar Island (Anna Pigeon #19) | Books to Scrape - Sandbox',
'The Widow | Books to Scrape - Sandbox',
'Playing with Fire | Books to Scrape - Sandbox',
'What Happened on Beale Street (Secrets of the South Mysteries #2) |
Books to Scrape - Sandbox',
"The Bachelor Girl's Guide to Murder (Herringford and Watts Mysterie
s #1) | Books to Scrape - Sandbox",
'Delivering the Truth (Quaker Midwife Mystery #1) | Books to Scrape
- Sandbox'],
'price_including_tax': ['£47.82',
'£19.63',
'£56.50',
'£16.64',
'£44.10',
'£54.21',
'£13.92',
'£10.69',
'£48.35',
'£16.73',
'£26.80',
'£54.36',
'£35.28',
'£11.84',
'£59.48',
'£27.26',
'£13.71',
'£25.37',
'£52.30',
'£20.89'],
'number_available': ['In stock (20 available)',
'In stock (18 available)',
'In stock (16 available)',
'In stock (16 available)',
'In stock (15 available)',
'In stock (14 available)',
'In stock (14 available)',
'In stock (14 available)',
'In stock (14 available)',
'In stock (14 available)',
'In stock (12 available)',
'In stock (12 available)',
'In stock (12 available)',
'In stock (12 available)',
'In stock (12 available)',
'In stock (12 available)',
'In stock (11 available)',
'In stock (11 available)',
'In stock (7 available)',
'In stock (7 available)',
'In stock (7 available)']}]

```

In [82]:

```
df = pd.DataFrame(data)
df
```

Out[82]:

| | product_page_url | title | price_including_tax | number_available |
|---|---|--|---------------------|---------------------|
| 0 | http://books.toscrape.com/catalogue/sharp-obje... | Sharp Objects Books to Scrape - Sandbox | £47.82 | In stock availat |
| 1 | http://books.toscrape.com/catalogue/in-a-dark-... | In a Dark, Dark Wood Books to Scrape - Sandbox | £19.63 | In stock availat |
| 2 | http://books.toscrape.com/catalogue/the-past-n... | The Past Never Ends Books to Scrape - Sandbox | £56.50 | In stock availat |
| 3 | http://books.toscrape.com/catalogue/a-murder-i... | A Murder in Time Books to Scrape - Sandbox | £16.64 | In stock availat |
| 4 | http://books.toscrape.com/catalogue/the-murder... | The Murder of Roger Ackroyd (Hercule Poirot #4... | £44.10 | In stock availat |
| 5 | http://books.toscrape.com/catalogue/the-last-m... | The Last Mile (Amos Decker #2) Books to Scra... | £54.21 | In stock availat |
| 6 | http://books.toscrape.com/catalogue/that-darkn... | That Darkness (Gardiner and Renner #1) Books... | £13.92 | In stock availat |
| 7 | http://books.toscrape.com/catalogue/tastes-lik... | Tastes Like Fear (DI Marnie Rome #3) Books t... | £10.69 | In stock availat |
| 8 | http://books.toscrape.com/catalogue/a-time-of-... | A Time of Torment (Charlie Parker #14) Books... | £48.35 | In stock availat |
| 9 | http://books.toscrape.com/catalogue/a-study-in... | A Study in Scarlet (Sherlock Holmes #1) Book... | £16.73 | In stock availat |

| | product_page_url | title | price_including_tax | number_available |
|----|---|--|---------------------|---------------------|
| 10 | http://books.toscrape.com/catalogue/poisonous-... | Poisonous (Max Revere Novels #3) Books to Sc... | £26.80 | In stock availat |
| 11 | http://books.toscrape.com/catalogue/murder-at-... | Murder at the 42nd Street Library (Raymond Amb... | £54.36 | In stock availat |
| 12 | http://books.toscrape.com/catalogue/most-wante... | Most Wanted Books to Scrape - Sandbox | £35.28 | In stock availat |
| 13 | http://books.toscrape.com/catalogue/hide-away-... | Hide Away (Eve Duncan #20) Books to Scrape -... | £11.84 | In stock availat |
| 14 | http://books.toscrape.com/catalogue/boar-islan... | Boar Island (Anna Pigeon #19) Books to Scrap... | £59.48 | In stock availat |
| 15 | http://books.toscrape.com/catalogue/the-widow_... | The Widow Books to Scrape - Sandbox | £27.26 | In stock availat |
| 16 | http://books.toscrape.com/catalogue/playing-wi... | Playing with Fire Books to Scrape - Sandbox | £13.71 | In stock availat |
| 17 | http://books.toscrape.com/catalogue/what-happe... | What Happened on Beale Street (Secrets of the ... | £25.37 | In stock availat |
| 18 | http://books.toscrape.com/catalogue/the-bachel... | The Bachelor Girl's Guide to Murder (Herringfo... | £52.30 | In stock availat |
| 19 | http://books.toscrape.com/catalogue/delivering... | Delivering the Truth (Quaker Midwife Mystery #... | £20.89 | In stock availat |

In []:

In []: