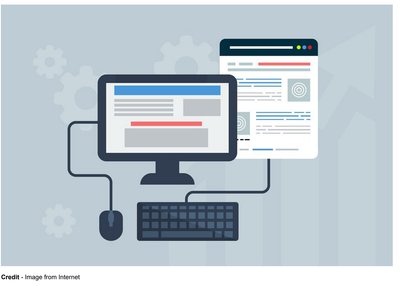
Introduction to Web-Scraping

- Web Harvesting
- Web Data Extraction
- Screen Scraping
- Growth Hacking

A method of collecting data from websites is called Web Scraping. Usually the software or the script that does this process is termed as `Bot` or `Web Crawler`.

Usual way of doing it

- Collecting data from online and storing it in your local file or database.
- Collecting data from online and deploying it as an API or URL for further usages.



Credit - Image from Internet

- API - (Application Programing Interface) acts as a mediator between server and the client machine.
  - Imagine API to be a URL (link) in which the data is obtained by slightly changing the behaviour.
  - Client ((User) requests for the data to the server through API.
  - Server responds the user if the request is valid (success - status code → 200).

Web Scraping relationship with Hacking

In a lot of ways web scraping has been termed as a growth hacking technique to build up sales pipeline and determine how the competitors are setting their prices.

Well that comes under marketing field. How is data science and coding related to web scraping.

More information - https://www.entrepreneur.com/article/296906

Web scraping is used to collect the data which is publicly open. It helps so many businesses in so many ways

- To understad the customer behaviour.
- To estimate or understand what the customer is craving for.
- To make machine learning model **from the public data** and predict the customer interest.
- so on.

Is web scraping legal?

There are two dimensions here as well

- Good Bots
- Bad Bots

Good Bots - They value the owner's standards and abide with the rules of scraping. They value the customers point in knowing more with less effort like `price comparison`, `social sentiment guaging`, `helping market researchers` and other so many aspects.

Bad Bots - Very much opposite to `Good Bots`. Data Breach, User account hacking, Online Fraud, Unauthorized vulnerability scans, Spam and digital ad fraud.

Bad Bot **vs** Good Bot

Credits - Image from Internet

Web scraping is not illegal after all. Startups love web scraping to understand customers and they are getting the data without much effort in partnering with other data providers.

More information - https://www.imperva.com/blog/is-web-scraping-illegal/

Web Scraping in Python

Web scraping in python can be done using the following packages.

- requests
- bs4
- selenium - requires chromium or firefox driver
- scrapy

Credits - Image from Internet

Installation of the packages

For Windows - open command prompt

- **bs4** - `py -m pip install bs4 --user`
- **requests** - `py -m pip install requests --user`

For Linux - open terminal

- **bs4** - `pip install bs4 --user`
- **requests** - `pip install requests --user`

But before doing this, make sure your `pip` is recognized in Windows

Live coding

JSON - JavaScript Object Notation

- lightweigt data interchange format
- easy for humans to read
- extraction is done by parsing method
- it can be taken as a dictionary in python

Struncture of JSON

```
{
    "key" : "value",
    "key" : {
            "sub_key" : "value",
            "sub_key" : "value"
        },
    "key" : [
        {
            "sub_key" : "value",
            "sub_key" : "value"
        },
        {
            "sub_key" : "value",
            "sub_key" : "value"
        }
    ],
    "key" : "value",
    "key" : ["value", "value", "value"]
}
```

Let's scrape the device location

In [1]:
```
# ip_url = 'http://ip-api.com/json'

import requests

class DeviceTracker():
    def __init__(self, ip_url):
        self.ip_url = ip_url

    def get_device_data(self):
        ip_req = requests.get(url=self.ip_url)
        if ip_req.status_code == 200:
            ip_data = ip_req.json()
        else:
            ip_data = {}
        return ip_data

    def get_user_loc(self):
        ip_data = self.get_device_data()
        city_name = ip_data['city']
        return city_name
```

In [2]:
```
ip_url = 'http://ip-api.com/json'
ip_dev = DeviceTracker(ip_url=ip_url)
city_name = ip_dev.get_user_loc()
print(city_name)
```

```
Hyderabad
```

Let's scrape the location of any place and get the weather data

In [3]:
```
# 'http://api.openweathermap.org/data/2.5/weather?q={}&appid=9d41bd4e5bffd04e03a6cb683206559'
# name - anything
# celsius - temp - 273
# farenheit - celsius * 9/5 + 32

import requests

class WeatherApp(DeviceTracker):
    def __init__(self, ip_url):
        self.ip_url = ip_url
        self.weather_url = 'http://api.openweathermap.org/data/2.5/weather?q={}&appid=9d41bd4e5bffd04e03a6cb683206559'
        self.place_name = None

    def get_weather_data(self):
        self.place_name = input("Please enter valid city name: ")
        wea_url = self.weather_url.format(self.place_name)
        wea_req = requests.get(url=wea_url)
        if wea_req.status_code == 200:
            wea_data = wea_req.json()
        else:
            print("-----------------")
            print("The entered place name is not valid")
            print("Getting the user location ...")
            self.place_name = self.get_user_loc()
            wea_url = self.weather_url.format(self.place_name)
            wea_req = requests.get(url=wea_url)
            wea_data = wea_req.json()
        return wea_data

    def get_parsed_details(self):
        weather_data = self.get_weather_data()
        desc = weather_data['weather'][0]['description']
        temp = weather_data['main']['temp']

        celsius = temp - 273
        farenheit = celsius * (9/5) + 32

        humidity = weather_data['main']['humidity']
        wind_speed = weather_data['wind']['speed']
        all_clouds = weather_data['clouds']['all']

        print("-----------------")
        print("The weather details of the place - {}".format(self.place_name))
        print("Weather description = ", desc)
        print("The temp in celsius = ", round(celsius, 2))
        print("The temp in farenheit = ", round(farenheit, 2))
        print("The wind speed - {} mpg".format(wind_speed))
        print("Humidity = ", humidity)
        print("Total clouds - ", all_clouds)

        return True
```

In [4]:
```
ip_url = 'http://ip-api.com/json'
w_app = WeatherApp(ip_url=ip_url)
w_app.get_parsed_details()
```

```
Please enter valid city name: chennai
-----------------
The weather details of the place - chennai
Weather description =  haze
The temp in celsius =  25.15
The temp in farenheit =  77.27
The wind speed - 4.89 mpg
Humidity =  78
Total clouds =  40
```

Out[4]: True

What did we learn?

- Web scraping definition
- Bot and crawlers
- Web scraping and Growth hacking
- Web scraping legal/illegal
- Live coding