

# Web Scrapping (& Flask API)

## Team 2:

Kazakos Christos, it22033

Katsaras Konstantinos, it22045

Linardakis Manousos, it22064

# Web Scraping

*Definition:* Python web scraping is an automated method used for collecting large amounts of data from websites and storing it in a structured form.

Χρησιμεύει έτσι ώστε να εξάγουμε – διαχειριζόμαστε δεδομένα που **δεν** είναι διαθέσιμα από κάποιο έτοιμο API!



# Popular FrameWorks For Web Scrapping



Scrapy

BeautifulSoup



# FrameWorks We Used For Web Scrapping



BeautifulSoup



# Why Selenium?

Επιλέξαμε το selenium καθώς προσφέρει ένα WebDriver Interface.

Έτσι μπορούμε με αμεσότητα να πατάμε κουμπιά με μία συνάρτηση (`click()`) ή να περιμένουμε ως ότου φορτώσει η σελίδα (`wait_until()`) σαν έναν κανονικό χρήστη.

Θα δούμε και αναλυτικότερα τις συναρτήσεις αυτές στην συνέχεια...



# Why BeautifulSoup?

Το BeautifulSoup είναι πιο “ελαφρύ” και γρήγορο από το Selenium.

Εξάγουμε **κείμενο από το html** που παίρνουμε από το selenium.

Ακολουθούν παραδείγματα (κώδικας) για καλύτερη εξήγηση των μεθόδων web scraping που χρησιμοποιήθηκαν.

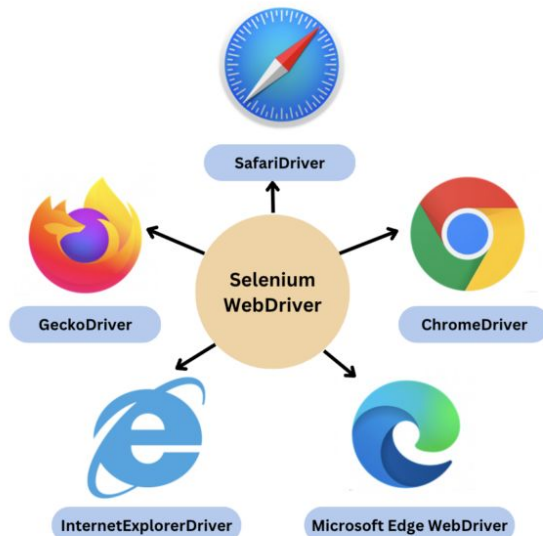
BeautifulSoup



# Web Scrapping Example Code

# Selenium Web Drivers

Το Selenium για να αλληλεπιδράσει με έναν web browser χρησιμοποιεί Web Drivers!



[image source](#)




# Selenium Available WebDrivers

```
from selenium import webdriver
```

```
webdriver.Firefox  
webdriver.FirefoxProfile  
webdriver.Chrome  
webdriver.ChromeOptions  
webdriver.Ie  
webdriver.Opera  
webdriver.PhantomJS  
webdriver.Remote  
webdriver.DesiredCapabilities  
webdriver.ActionChains  
webdriver.TouchActions  
webdriver.Proxy
```

# Web Drivers Example

```
try:
    options = Options()
    options.add_argument('--headless') # remove this if you want the browser to appear.
    # create Chrome service - installs chrome driver:
    service = Service(ChromeDriverManager().install())
    # initialise driver:
    driver = webdriver.Chrome(service=service, options=options)
except: # starts firefox
    options = webdriver.FirefoxOptions()
    options.add_argument('--headless') # remove this if you want the browser to appear.
    service = webdriver.firefox.service.Service(executable_path=GeckoDriverManager().install())
    driver = webdriver.Firefox(service=service, options=options) # go to url:
```



# Selenium: driver.get(page\_url)




Για να οδηγήσουμε τον webdriver στην επιθυμητή σελίδα, χρησιμοποιούμε την `driver.get(page_url)`.


```
discogs_url = "https://www.discogs.com"

page_url = discogs_url+"/artist/"+artist_name+"?limit=500"
print('Opening Page...')
driver.get(page_url)
```



# Αποτέλεσμα στο Web Driver

 Search artists, albums and more...  Explore ▾ Marketplace ▾ Community ▾  Log In Register

  
[More Images](#)

## Queen

Profile:


Queen is a British rock band formed in London in 1970 from the previously disbanded [Smile \(6\)](#) Rock band. Originally called Smile, later in 1970 singer [Freddie Mercury](#) came up with the new name for the band. [John Deacon](#) joined in March 1971 giving them their fourth and final bass player.

The band has released a total of 18 number-one albums, 18 number-one singles and 10 [More ▾](#)


Sites:

[queenonline.com](#), [Facebook](#), [Wikipedia](#), [Twitter](#), [MySpace](#), [YouTube](#), [Wikipedia](#), [Last.fm](#), [queenworld.com](#), [queenvault.com](#), [queenpedia.com](#), [shanemcdonald.ie](#), [ultimatequeen.co.uk](#)

### Artist

 [a81013]

[Edit Artist](#)

 [Share](#)

### Marketplace

[103,121 For Sale](#)

[Vinyl and CD](#)

## Let's manage your privacy

We and our partners store and/or access information on a device, such as unique IDs in cookies to process personal data. You may accept or manage your choices by clicking below, including your right to object where legitimate interest is used, or at any time in the privacy policy page. These choices will be signaled to our partners and will not affect browsing data. [Read our Cookie and Internet Advertising Policy](#)

### We and our partners process data to provide:

Actively scan device characteristics for identification. Use precise geolocation data. Measure content performance. Store and/or access information on a device. Select personalised content. Develop and improve products. Apply market research to generate audience insights. Create a personalised content profile. Measure ad performance. Select personalised ads. Create a personalised ads profile. Select basic ads.

[View our list of partners \(vendors\)](#)

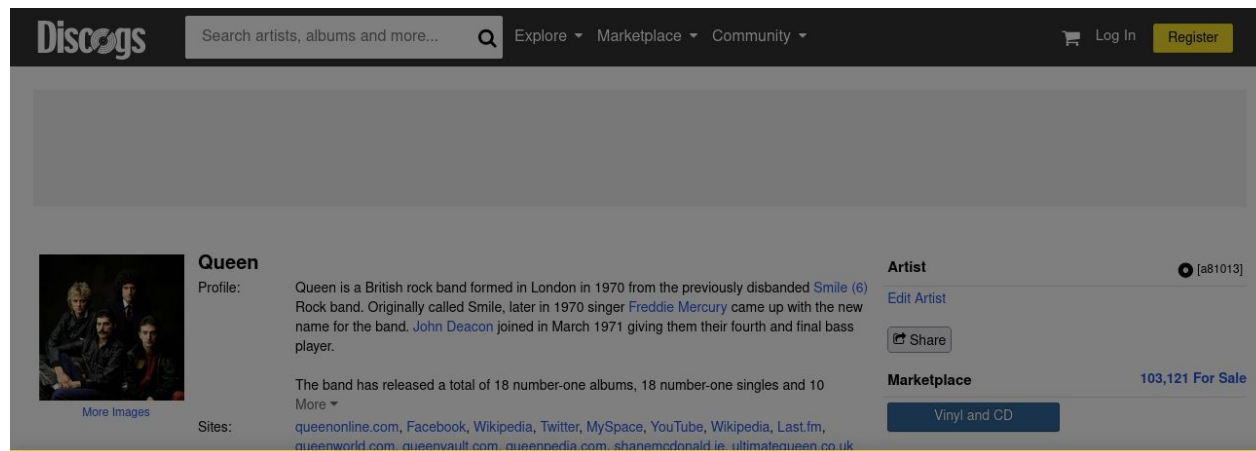
Allow all cookies

Reject all unnecessary cookies

[Show purposes](#)

# Web Scraping – Accepting Cookies

Θα χρειαστεί ύστερα να αποδεχτούμε τα cookies, έτσι ώστε να φορτωθεί όλη η σελίδα.



The screenshot shows the Discogs website interface. At the top is a navigation bar with the Discogs logo, a search bar, and links for Explore, Marketplace, and Community. On the right of the navigation bar are links for Log In and Register. The main content area displays the profile for the band Queen. It includes a profile picture, a bio stating they are a British rock band formed in 1970, and a list of social media sites. There is also a section for the band's marketplace, showing 103,121 items for sale.

## Let's manage your privacy

We and our partners store and/or access information on a device, such as unique IDs in cookies to process personal data. You may accept or manage your choices by clicking below, including your right to object where legitimate interest is used, or at any time in the privacy policy page. These choices will be signaled to our partners and will not affect browsing data. [Read our Cookie and Internet Advertising Policy](#)

## We and our partners process data to provide:

Actively scan device characteristics for identification. Use precise geolocation data. Measure content performance. Store and/or access information on a device. Select personalised content. Develop and improve products. Apply market research to generate audience insights. Create a personalised content profile. Measure ad performance. Select personalised ads. Create a personalised ads profile. Select basic ads.

[View our list of partners \(vendors\)](#)

Allow all cookies

Reject all unnecessary cookies

Show purposes

# Web Scraping – WebDriver Wait

Για να αποδεχτούμε τα cookies, όμως, πρέπει να σιγουρευτούμε ότι το κουμπί “Allow all Cookies” υπάρχει στη σελίδα. Για αυτό χρησιμοποιούμε:

```
wait = WebDriverWait(driver, 10)
button = wait.until(EC.presence_of_element_located((By.ID, "onetrust-accept-btn-handler")))
```

1. Αρχικοποιούμε ένα αντικείμενο `WebDriverWait` με μέγιστο όριο αναμονής (για να φορτώσει η σελίδα) 10 δευτερολέπτων.
2. Ύστερα, χρησιμοποιούμε την `wait.until`, η οποία περιμένει ως ότου εμφανιστεί κάποιο κουμπί με το ορισμένο id (το οποίο αντιστοιχεί στο κουμπί του “allow all cookies” για την σελίδα του discogs).

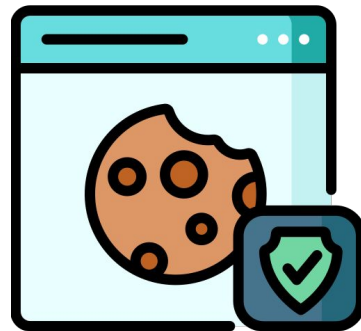


# Web Scrapping – Click (Cookies) Button

Αφού έχουμε βεβαιώσει ότι έγινε το load του κουμπιού για τα cookies, το κάνουμε click με την μέθοδο:


```
button.click()
```

Έτσι καταφέραμε να αποδεχτούμε τα cookies και να φορτώσουμε σωστά όλη τη σελίδα.



[image source](#)

# Αποτέλεσμα Accept All Cookies Click



[More Images](#)

## Queen

**Profile:** Queen is a British rock band formed in London in 1970 from the previously disbanded [Smile](#) (6) Rock band. Originally called Smile, later in 1970 singer [Freddie Mercury](#) came up with the new name for the band. [John Deacon](#) joined in March 1971 giving them their fourth and final bass player.

The band has released a total of 18 number-one albums, 18 number-one singles and 10 number-one DVDs, and have sold over 300 million albums worldwide, making them one of the world's best-selling [More](#) ▾

**Sites:** [queenonline.com](#), [Facebook](#), [Wikipedia](#), [Twitter](#), [MySpace](#), [YouTube](#), [Wikipedia](#), [Last.fm](#), [queenworld.com](#), [queenvault.com](#), [queenpedia.com](#), [shanemcdonald.ie](#), [ultimatequeen.co.uk](#)

**Members:** [Barry Mitchell](#), [Brian May](#), [Freddie Mercury](#), [John Deacon](#), [Roger Taylor](#)

**Variations:** [Viewing All](#) | [Queen](#)

Kween, Q, Qeen, Queen Productions, Quenn, The Queen, Куинн, Куин, クイーン, クイーン, 皇后乐队, 皇后合唱团, 皇后合唱团, 皇后樂隊

**Artist** [a81013]

[Edit Artist](#)

[Share](#)

**Marketplace** [103,121 For Sale](#)

[Vinyl and CD](#)




**Discography**

Search

1 – 25 of 540 [◀](#) [Prev](#) [Next](#) [▶](#)

Sort  Show  ☐ ☒ ☐

### Albums

	<a href="#">Queen</a> <input type="text" value="283 versions"/>	<a href="#">EMI</a> , <a href="#">EMI</a>	1973
	<a href="#">Sheer Heart Attack</a> <input type="text" value="267 versions"/>	<a href="#">EMI</a> , <a href="#">EMI</a>	1974
	<a href="#">Queen II</a> <input type="text" value="279 versions"/>	<a href="#">EMI</a> , <a href="#">EMI</a>	1974

**Releases** 540

[Albums](#) 83

[Singles & EPs](#) 172

[Compilations](#) 102

[Videos](#) 42

[Miscellaneous](#) 141

[Appearances](#) 257

**TURNTABLE LAB**

SHOP ONLINE 24/7  
THANKS FOR YOUR SUPPORT!



# Web Scrapping – Επιλογή Συγκεκριμένων Elements


Έχοντας φορτώσει όλη τη σελίδα, θα χρειαστεί να επιλέξουμε συγκεκριμένα στοιχεία της (όσα δηλαδή μας ενδιαφέρουν για το web scraping). Για αυτό χρησιμοποιούμε:

```
table = driver.find_element(By.ID, 'artist'):
```

**βρίσκουμε** το element με id “artist” – το αποθηκεύουμε σε μία μεταβλητή.

```
table_html = table.get_attribute('outerHTML')
```

**παίρνουμε** όλο το περιεχόμενο HTML του συγκεκριμένου table.



# Αποτέλεσμα Επιλογής “artist”

The screenshot displays the Discogs website for the artist Queen. The page features a sidebar with navigation links for Releases (540), Appearances (3587), and Unofficial (1569) items. The main content area shows a list of albums, including Queen, Sheer Heart Attack, Queen II, and Randy Newman / Queen - Stereo Pop Special-72 (LP, Transcription). A banner at the bottom of the album list reads "10/10 Albums According To The Discogs Community".

The browser's developer tool is open at the bottom, showing the HTML structure and CSS styles for the selected element. The HTML structure includes a table with the following content:

Album	Label	Year
Queen	EMI, EMI	1973
Sheer Heart Attack	EMI, EMI	1974
Queen II	EMI, EMI	1974
Randy Newman / Queen - Stereo Pop Special-72 (LP, Transcription)	BBC Transcription Services	CN 2073/S 1974

The CSS styles for the selected element are:

```
body {  
  line-height: 20px;  
}
```

The browser's console shows the following error:

```
Uncaught ReferenceError: document is not defined
```

# Web Scraping – BeautifulSoup

Στην συνέχεια θα χρειαστεί να “διαβάσουμε” όλα τα στοιχεία του table → **πολύ χρονοβόρο (με το selenium, λόγω του web driver interface)**.

Για αυτό χρησιμοποιούμε το BeautifulSoup!

BeautifulSoup



# Web Scraping – BeautifulSoup Code

```
soup = BeautifulSoup(table_html, 'html.parser')
```

το `html.parser` χρησιμοποιείται για να προσπελάσει το `html` και να φτιάξει ένα BeautifulSoup Object.

```
a_tags = soup.find_all('a', href=lambda href: href and ('/master/' in href or '/release/' in href))
```

παίρνουμε τα `href` όλων των `anchor (<a>)` του `table` που περιέχουν το string `‘/master/’` ή `‘/release/’`. Τα `href` αυτά είναι οι συνδέσμοι που οδηγούν στους δίσκους του καλλιτέχνη (της σελίδας που βρισκόμαστε).

Έτσι έχουμε αποθηκεύσει όλα τα `url` των δίσκων του καλλιτέχνη σε μία μεταβλητή `a_tags`.

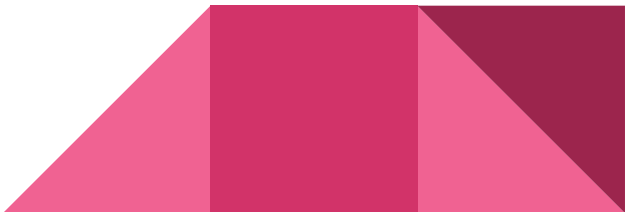


# Αποθήκευση URLs δίσκων σε dictionary

Δημιουργούμε ένα dictionary “albums” το οποίο περιέχει ως κλειδί το όνομα του δίσκου και για value το url του δίσκου αυτού:

```
for a_tag in a_tags:
    disc_txt = a_tag.text
    disc_txt = disc_txt.replace(' ', '-').replace('/', '-')
    if disc_txt not in albums: # keeps the top albums
        albums[disc_txt] = discogs_url+a_tag['href']
```

Για να πάρουμε το όνομα του δίσκου χρησιμοποιούμε το a\_tag.text → επιστρέφει το κείμενο του anchor <a>.



# Web Scraping

Έχοντας αποθηκεύσει όλα τα albums του καλλιτέχνη και τα αντίστοιχα urls τους στο dictionary, τα προσπελάζουμε πηγαίνοντας στο κάθε url με το `driver.get(page_url)` → συλλέγουμε έτσι στοιχεία για το δίσκο:

Discography

Search

Releases

540

Albums

83

Singles & EPs

172

Compilations

103

Videos

41

Miscellaneous

141

Appearances

3613

Albums

274

1 – 500 of 540

< Prev

Next >


Sort

Year, 0-9

Show

500

Albums




Queen

◀ 284 versions

EMI, EMI

1973

▼




Sheer Heart Attack

◀ 268 versions

EMI, EMI

1974

▼



Queen II

◀ 274 versions

EMI, EMI


1974

▼

# Web Scraping

Έχοντας αποθηκεύσει όλα τα albums του καλλιτέχνη και τα αντίστοιχα urls τους στο dictionary, τα προσπελάζουμε πηγαίνοντας στο κάθε url με το `driver.get(page_url)` → συλλέγουμε έτσι στοιχεία για το δίσκο:

Discography

Search 




Releases **540**

- Albums 83
- Singles & EPs 172
- Compilations 103
- Videos 41
- Miscellaneous 141




Appearances **3613**

- Albums 274

1 – 500 of 540 < Prev Next >

Sort Year, 0-9 ▾ Show 500 ▾   

### Albums

	Queen ◀ 284 versions	EMI, EMI	1973 ▾
	Sheer Heart Attack ◀ 268 versions	EMI, EMI	1974 ▾
	Queen II ◀ 274 versions	EMI, EMI	1974 ▾

# Web Scraping

Έχοντας αποθηκεύσει όλα τα albums του καλλιτέχνη και τα αντίστοιχα urls τους στο dictionary, τα προσπελάζουμε πηγαίνοντας στο κάθε url με το `driver.get(page_url)` → συλλέγουμε έτσι στοιχεία για το δίσκο:

Discography

Search

Q

Releases

540

Albums

83

Singles & EPs

172

Compilations

103

Videos

41

Miscellaneous

141

Appearances

3613

Albums

274

1 – 500 of 540

◀ Prev

Next ▶

Sort

Year, 0-9 ▼

Show


500 ▼

⌵

⌵

⌵

Albums




Queen

◀ 284 versions

EMI, EMI

1973

▼




Sheer Heart Attack

◀ 268 versions

EMI, EMI

1974

▼



Queen II

◀ 274 versions

EMI, EMI

1974

▼



# Web Scrapping

“Πατώντας” κάθε url των δίσκων του dictionary, οδηγούμαστε σε μία σελίδα στην οποία υπάρχει το κουμπί “Buy Copy”.

The screenshot shows a Firefox Web Browser window displaying the Discogs website. The address bar shows the URL: <https://www.discogs.com/master/5303-Queen-Sheer-Heart-Attack>. The page features a dark header with the Discogs logo and navigation links. A prominent banner for 'TURNTABLE LAB' is visible, along with a 'SHOP ONLINE 24/7 THANKS FOR YOUR SUPPORT!' message. The main content area displays the album 'Queen – Sheer Heart Attack' with its cover art, genre (Rock), style (Hard Rock, Classic Rock, Prog Rock), and year (1974). A tracklist is provided below the album information. On the right side, the 'Master Release' section shows the album's status and a 'Marketplace' section with a 'Buy a copy' button. The 'Statistics' section indicates that the album has 63112 'Have' counts and 67207 'Want' counts, with an average rating of 4.37/5 and 5334 ratings. A sidebar on the left contains various application icons. The bottom of the page features a 'TURNTABLE LAB' banner and a URL: [https://www.discogs.com/sell/list?master\\_id=5303&ev=mb](https://www.discogs.com/sell/list?master_id=5303&ev=mb).

# Web Scrapping

Διαλέγουμε το Buy a copy Button:

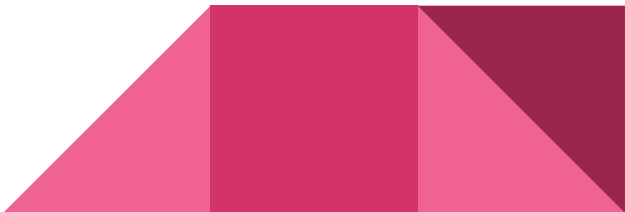
```
buy_button = soup.find('a', string='Buy a copy')
```

επιλογή anchors με string “Buy a copy” (είναι μόνο τα κουμπιά Buy a copy στη σελίδα αυτή).

και παίρνουμε το href του κουμπιού ως εξής:

```
buy_link = buy_button.get('href')
```

ύστερα το αποθηκεύουμε πάλι σε ένα dictionary.



# Web Scrapping

Με παρόμοιο τρόπο με πριν, προσπελάζουμε το καινούργιο dictionary “πατώντας” όλα τα buy a copy buttons (με το `driver.get(page_url)`) → επιλέγουμε το view release page του κάθε δίσκου της σελίδας:

The screenshot shows the Discogs Marketplace interface. At the top, there's a navigation bar with 'Marketplace' and various filters. Below this, the search results for 'Queen - Sheer Heart Attack (All Versions)' are displayed. The results are sorted by 'Listed Newest' and show 1 to 25 of 1,757 items. The first three items are visible, each with a thumbnail, title, label, catalog number, media condition, sleeve condition, seller, price, shipping, and ratings. Each item has an 'Add to Cart' button and a 'View Release Page' link. A banner at the top and bottom of the page reads '3 Trends That Prove CDs are Making a Comeback'.

Sort By:	Listed	Condition	Artist	Title	Label	Seller	Price
Sort By:	Listed	Condition	Artist	Title	Label	Seller	Price
Sort By:	Listed	Condition	Artist	Title	Label	Seller	Price
Sort By:	Listed	Condition	Artist	Title	Label	Seller	Price
Sort By:	Listed	Condition	Artist	Title	Label	Seller	Price

# Web Scraping

Επιλέγοντας το κάθε view release page παίρνουμε τιμές για τους δίσκους για διάφορες μέρες των χρόνων που είναι released:

The screenshot shows the Discogs website interface. At the top, there's a navigation bar with the Discogs logo, a search bar, and links for Explore, Marketplace, and Community. Below this, a banner for '7 Essential David Bowie Albums' is visible. The main content area features the release page for 'Queen - Sheer Heart Attack'. The album cover is shown on the left, and the release details are on the right. The details include the label (Virgin EMI Records), format (Vinyl, LP), country (Europe), release date (Sep 25, 2015), genre (Rock), and style (Hard Rock, Pop Rock, Arena Rock). A 'Tracklist' section is also present, listing six tracks with their durations and writers. On the right side, there's a 'Marketplace' section showing the current price for the vinyl (\$21.51) and buttons to 'Buy Vinyl' or 'Sell Vinyl'. Below this, a 'Statistics' section provides data on the number of copies available (3112), wanted (596), average rating (4.56/5), and ratings (325). A 'Share' button is also visible. At the bottom, there's a banner for 'Top 25 Most Expensive Items Sold on Discogs in April 2023'.

Discogs

Search artists, albums and more...

Explore Marketplace Community

Log In Register

Release [7541484]

Edit Release  
All Versions of this Release  
Recently Edited

Add to Collection Add to Wantlist

Marketplace 99 For Sale from \$21.51

Buy Vinyl Sell Vinyl

Statistics

	Have:	Want:	Avg Rating:	Ratings:	Last Sold	Lowest	Median	Highest
	3112	596	4.56 / 5	325	May 21, 2022	\$19.06	\$24.99	\$45.99

☆☆☆☆☆ Share

Tracklist

	Track	Duration	Writers
A1	Brighton Rock	5:10	Written-By - May*
A2	Killer Queen	3:00	Written-By - Mercury*
A3	Tenement Funster	2:47	Written-By - Taylor*
A4	Flick Of The Wrist	3:17	Written-By - Mercury*
A5	Lily Of The Valley	1:45	Written-By - Mercury*
A6	Now I'm Here	4:13	Written-By - May*

Top 25 Most Expensive Items Sold on Discogs in April 2023

# Web Scrapping

Έτσι, έχουμε πολλές τιμές για τον δίσκο σε διάφορες ημερομηνίες. Τα αποθηκεύουμε όλα τα αποτελέσματα σε ένα pandas για να αντιμετωπίσουμε ύστερα πιο εύκολα τα outlier values:

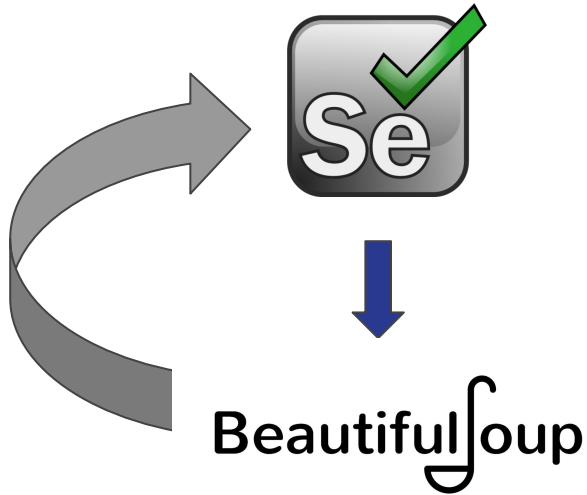
```
# Convert data list (with disc prices) into pandas DataFrame
df = pd.DataFrame(data)

# Convert the "lowest_price", "median_price", and "highest_price" columns to floats
df[["lowest_price", "median_price", "highest_price"]] = df[["lowest_price", "median_price",
"highest_price"]].applymap(lambda x: float(x.lstrip("$")))

# Convert the date column to datetime format
df['date'] = pd.to_datetime(df['date'])
```

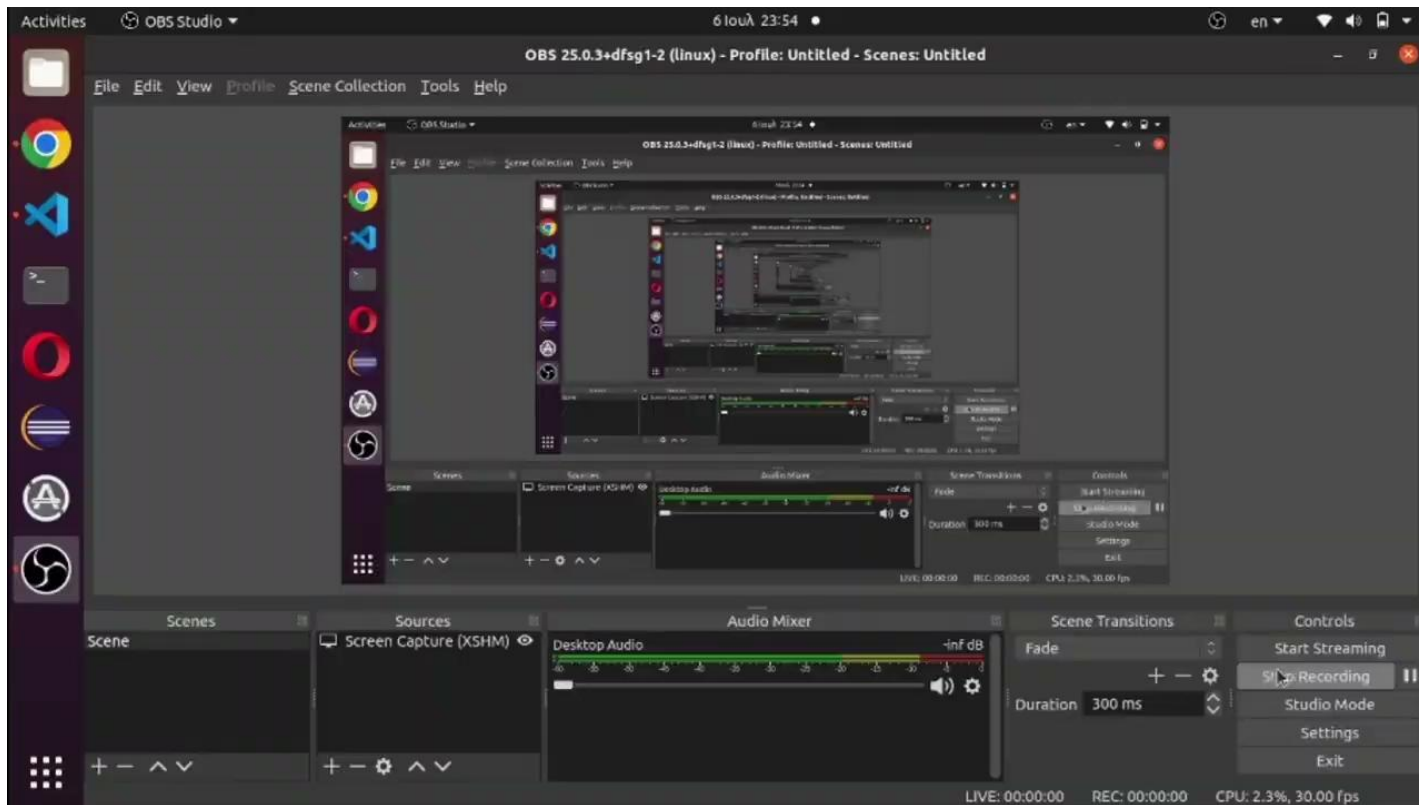


# Web Scraping Logic – Pattern



- 1) Navigate to the next page using **Selenium (easy navigation)**.
- 2) Extract data from HTML using **BeautifulSoup (fast)**.
- 3) **Repeat!**

# Web Scrapping Demo



<https://drive.google.com/file/d/1uammOiB9gioEWTtihNweGplo0oQ-PJOS/view> [Recording](#)

# Flask API



source: <https://www.educative.io/blog/python-flask-tutorial>



# Τι είναι;

Με το flask api, μπορούμε **εύκολα** και **γρήγορα** να κατασκευάσουμε ένα REST API στη Python.

Για να αρχικοποιήσουμε το REST API χρησιμοποιούμε την εντολή `app = Flask(__name__)` που δημιουργεί μια νέα παρουσία της κλάσης Flask και το `__name__` αναφέρεται στο όνομα του τρέχοντος module ή πακέτου.

Για να τρέξουμε το API, εκτελούμε το `app.run()` → ως default τρέχει στο port 5000.



## @app.route('/path')

Με τη χρήση του decorator `@app.route('/path')` πάνω από μια συνάρτηση, όπου `'/path'` είναι η διαδρομή (URL) που θα ανταποκρίνεται η εφαρμογή, καθορίζουμε ποια λειτουργία θα εκτελεστεί για κάθε αίτηση που γίνεται σε αυτή τη συγκεκριμένη διαδρομή `'/path'`. Επίσης, μπορούμε να ορίσουμε και το method του `app.route` (GET, POST, PUT ...):

```
@app.route('/discs', methods=['GET'])
def get_user_discs_api():
    """
    Retrieves the discs of the authenticated user.

    Returns:
        JSON response: A JSON response containing the user's discs.
    """
```

# Flask Authentication

Για να πάρουμε τα στοιχεία του χρήστη (από το basic authentication), χρησιμοποιούμε το request αντικείμενο του flask, στο οποίο υπάρχει το username και password του χρήστη:

```
from flask import Flask, jsonify, request
```

```
def authenticate():
    """
    Performs authentication by checking the provided username and password against the credentials in the database.
    """
    auth_fail_msg = 'Authentication failed'
    auth = request.authorization
    if not auth:
        return jsonify({'message': auth_fail_msg}), 401
    # Get the provided username and password from the request
    username = auth.username
    password = auth.password
```

# Flask jsonify

Επιστρέφουμε ένα json χρησιμοποιώντας το jsonify του flask. Έτσι, μπορούμε να στέλνουμε κάποια μηνύματα στο χρήστη μαζί με ένα status code (με το return της αντίστοιχης συνάρτησης):

```
@app.route('/discs', methods=['GET'])
def get_user_discs_api():
    """
    Retrieves the discs of the authenticated user.

    Returns:
        JSON response: A JSON response containing the user's discs.
    """
    # authentication ----
    auth = authenticate()
    if auth[1] != 200:
        return auth[0]
    username = auth[0]
    # -----
    discs = get_user_discs(username, conn)
    # Create a list of dictionaries with keys 'disc_name' and 'band_name'
    discs_list = [{'disc_name': disc[0], 'band': disc[1]} for disc in discs]

    # Return the user's discs as JSON response
    return jsonify({'user_discs': discs_list}), 200
```

# Cookies (1/2)

Τα cookies είναι μικρά κομμάτια δεδομένων που αποθηκεύονται στη συσκευή του πελάτη από έναν ιστότοπο. Στο Flask, μπορούμε να διαχειριστούμε τα cookies χρησιμοποιώντας το αντικείμενο `request.cookies` για να διαβάσουμε τα cookies από τον πελάτη και τη συνάρτηση `make_response()` για να δημιουργήσουμε μια απόκριση που περιέχει cookies.



## Cookies (2/2)

Στο παράδειγμα, η διαδρομή / ανταποκρίνεται σε αιτήσεις GET και POST. Το αντικείμενο `request.cookies` χρησιμοποιείται για να διαβάσετε το cookie με όνομα 'username' από τον πελάτη. Έπειτα, χρησιμοποιείται η συνάρτηση `make_response()` για να δημιουργηθεί μια απόκριση που περιέχει το κείμενο 'Hello, ' συν την τιμή του cookie 'username'. Τέλος, μέσω της μεθόδου `resp.set_cookie()` ορίζεται το cookie.

```
@app.route('/')
def index():
    # Διάβασμα του cookie 'username' από τον πελάτη
    username = request.cookies.get('username')

    # Επιστροφή απόκρισης που περιέχει ένα cookie
    resp = make_response('Hello, ' + username)

    # Ορισμός του cookie 'username' με την τιμή 'John Doe'
    resp.set_cookie('username', 'John Doe')

    return resp
```

# Other URLs:

Source Code:

<https://github.com/manouslinard/music-recommender>

Web Scraping demo:

<https://drive.google.com/file/d/1uammOiB9gioEWTtihNweGplo0oQ-PJOS/view?usp=sharing>



Σας ευχαριστούμε που  
παρακολουθήσατε!