



POLITECNICO
MILANO 1863

Exercise Session - API

Andrea Tocchetti
andrea.tocchetti@polimi.it

API without Libraries

APIs can be invoked through HTTP methods (e.g., GET, POST, etc.) depending on the kind of operations to be performed.

As an end-user, the most used requests for APIs are GET requests.

There are many different tools available online to develop and test APIs.

Postman is one of the most renowned ones.

Tools - Postman

API Collection

Query

Request Parameters

Result

The screenshot displays the Postman web interface. On the left sidebar, the 'API Collections' section is expanded, showing a collection named 'API 101' with several endpoints. A red box highlights this section, with an arrow pointing to the 'API Collection' label. The main workspace shows a 'GET' request to 'postman-echo.com/get?firstName=foo1&lastName=foo2'. A red box highlights the request URL, with an arrow pointing to the 'Query' label. Below the URL, the 'Query Params' section is visible, showing a table with two parameters: 'firstName' and 'lastName'. A red box highlights this table, with an arrow pointing to the 'Request Parameters' label. The bottom section shows the 'Body' tab with a JSON response. A red box highlights the JSON response, with an arrow pointing to the 'Result' label.

Home Workspaces Reports Explore

Search Postman

Demo Workspace New Import

Watching APIs / Postman Echo GET

Save

GET postman-echo.com/get?firstName=foo1&lastName=foo2

Send

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
<input checked="" type="checkbox"/> firstName	foo1	filter by firstName	X
<input checked="" type="checkbox"/> lastName	foo2	filter by lastName	X
Key	Value	Description	

Body Cookies (1) Headers (7) Test Results (1/1)

200 OK 795 ms 770 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "args": {
3     "firstName": "foo1",
4     "lastName": "foo2"
5   },
6   "headers": {
7     "x-forwarded-proto": "http",
8     "x-forwarded-port": "80",
9     "host": "postman-echo.com",
10    "x-amzn-trace-id": "Root=1-5fc5d991-72ae448f234cb4dd2c177b84",
11    "user-agent": "PostmanRuntime/7.26.8",
```

Example - iTunes Store API

Retrieve the first 10 artists whose name contains “maroon”

<https://itunes.apple.com/search?term=maroon&entity=allArtist&attribute=allArtistTerm&limit=10>

```
1  [
2    "resultCount": 10,
3    "results": [
4      {
5        "wrapperType": "artist",
6        "artistType": "Artist",
7        "artistName": "Maroon 5",
8        "artistLinkUrl": "https://music.apple.com/us/artist/maroon-5/1798556?uo=4",
9        "artistId": 1798556,
10       "amgArtistId": 529962,
11       "primaryGenreName": "Rock",
12       "primaryGenreId": 21
13     },
14     {
15       "wrapperType": "artist",
16       "artistType": "Artist",
17       "artistName": "Maroon",
18       "artistLinkUrl": "https://music.apple.com/us/artist/maroon/5183483?uo=4",
19       "artistId": 5183483,
20       "amgArtistId": 692779,
21       "primaryGenreName": "Rock",
22       "primaryGenreId": 21
23     },
24     {
25       "wrapperType": "artist",
26       "artistType": "Artist",
```

Example - iTunes Store API

Retrieve all the artist whose surname is “Stallone”

<https://itunes.apple.com/search?term=Stallone&entity=movieArtist>

```
1  [
2    "resultCount": 6,
3    "results": [
4      {
5        "wrapperType": "artist",
6        "artistType": "Artist",
7        "artistName": "Frank Stallone",
8        "artistLinkUrl": "https://music.apple.com/us/artist/frank-stallone/91734783?uo=4",
9        "artistId": 91734783,
10       "amgArtistId": 5513,
11       "primaryGenreName": "Pop",
12       "primaryGenreId": 14
13     },
14     {
15       "wrapperType": "artist",
16       "artistType": "Movie Artist",
17       "artistName": "Sylvester Stallone",
18       "artistLinkUrl": "https://itunes.apple.com/us/artist/sylvester-stallone/186852826?uo=4",
19       "artistId": 186852826,
20       "primaryGenreName": "Drama",
21       "primaryGenreId": 4406
22     },
23     {
24       "wrapperType": "artist",
25       "artistType": "Movie Artist",
26       "artistName": "Sistine Stallone",
```

Example - iTunes Store API

Retrieve all the “Star Wars” movies

<https://itunes.apple.com/search?term=StarWars&entity=movie&attribute=movieTerm>

```
1  {
2    "resultCount": 7,
3    "results": [
4      {
5        "wrapperType": "track",
6        "kind": "feature-movie",
7        "collectionId": "1587064325",
8        "trackId": "1063466898",
9        "artistName": "J.J. Abrams",
10       "collectionName": "Star Wars: The Skywalker Saga 9-Movie Collection",
11       "trackName": "Star Wars: The Force Awakens",
12       "collectionCensoredName": "Star Wars: The Skywalker Saga 9-Movie Collection",
13       "trackCensoredName": "Star Wars: The Force Awakens",
14       "collectionArtistId": "410641764",
15       "collectionArtistViewUrl": "https://itunes.apple.com/us/artist/buena-vista-home-entertainment-inc/410641764?uo=4",
16       "collectionViewUrl": "https://itunes.apple.com/us/movie/star-wars-the-force-awakens/id1063466898?uo=4",
17       "trackViewUrl": "https://itunes.apple.com/us/movie/star-wars-the-force-awakens/id1063466898?uo=4",
18       "previewUrl": "https://video-ssl.itunes.apple.com/itunes-assets/Video82/v4/a3/ef/25/a3ef253a-208e-3cbc-cbf0-bc444dae2f8d/mzvf_6313981593442783545_640x354_h264lc.U.p.m4v",
19       "artworkUrl130": "https://is4-ssl.mzstatic.com/image/thumb/Video123/v4/1f/2b/ae/1f2bae7f-62a1-1055-8471-401291b6dcd0/pr_source.lsr/30x30bb.jpg",
20       "artworkUrl160": "https://is4-ssl.mzstatic.com/image/thumb/Video123/v4/1f/2b/ae/1f2bae7f-62a1-1055-8471-401291b6dcd0/pr_source.lsr/60x60bb.jpg",
21       "artworkUrl100": "https://is4-ssl.mzstatic.com/image/thumb/Video123/v4/1f/2b/ae/1f2bae7f-62a1-1055-8471-401291b6dcd0/pr_source.lsr/100x100bb.jpg",
22       "collectionPrice": 9.99,
23       "trackPrice": 9.99,
24       "trackRentalPrice": 3.99,
25       "collectionHdPrice": 9.99,
26       "trackHdPrice": 9.99,
```

Sometimes APIs are available through libraries.

Most of them are available in **Python** which is widely used to develop scripts for data collection and data analysis.

Python environments are useful to manage Python libraries.

Anaconda is a tool through which you can manage Python environments. It also provides tools to code and perform data analysis.

Tools - Anaconda

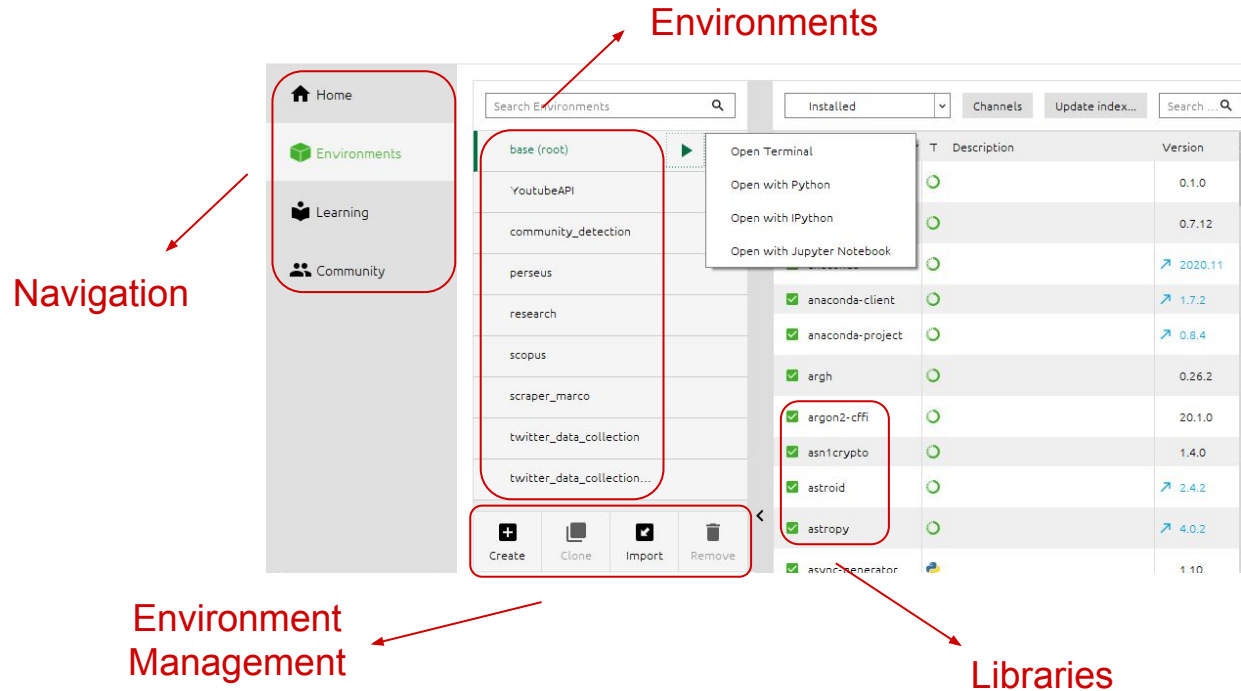
The image shows the Anaconda Navigator desktop application. On the left is a sidebar with navigation links: Home, Environments, Learning, and Community. A red box highlights this sidebar, with a red arrow pointing to it from the label "Navigation". At the top of the main panel, there is a dropdown menu labeled "Applications on" with "base (root)" selected, and a "Channels" button. A red box highlights this area, with a red arrow pointing to it from the label "Active Environment". The main panel displays a grid of application tiles. A red box highlights the "JupyterLab" tile, which is the active environment. A red arrow points from the label "JupyterLab Tool" to this tile. Other tiles include CMD.exe Prompt, DataLore, IBM Watson Studio Cloud, Notebook, Powershell Prompt, Qt Console, Spyder, Glueviz, Orange3, PyCharm Professional, and RStudio. Each tile shows its icon, name, version, a brief description, and a button to either "Launch" or "Install" the application.

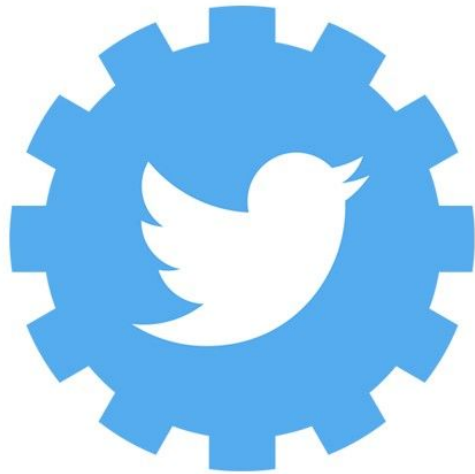
Navigation

Active Environment

JupyterLab Tool

Tools - Anaconda





“The Twitter API enables programmatic access to Twitter in unique and advanced ways. Use it to analyze, learn from, and interact with Tweets, Direct Messages, users, and other key Twitter resources.” - [Twitter API Documentation](#)

There are two different versions of [Twitter API](#), namely v1 and v2. While the first is free and open to everyone, the latter is novel and requires the developer to submit an authorization for each application they develop.

The second version also includes an access to past tweets, while the first one only provides the last week of tweets.



OpenStreetMap (OSM) is a collaborative project to create a free editable geographic database of the world.

Basically, it is a crowdsourced map of the world.

Users may collect data using manual survey, GPS devices, aerial photography, and other free sources, or use their own local knowledge of the area.



OpenWeather (OW) provides a set of weather-related products, including short-term and long-term forecasting, history and observations for any location.

Most of their services are provided through APIs.

ANY
Questions?



POLITECNICO
MILANO 1863

Exercise Session - Scraping

Document Object Model (DOM)

“The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document.”

In other words it is a standard for how to get, change, add, or delete HTML elements.

Document Object Model (DOM)

When a web page is loaded, the browser creates a Document Object Model of the page.

In particular, it defines

- The HTML elements as objects
- The properties of all HTML elements
- The methods to access all HTML elements
- The events for all HTML elements

There are several sources that can be scraped

- **Social networks:** Facebook, Instagram, etc.
- **Retailers:** Amazon, Tripadvisor, Google Maps, etc.
- Any website with valuable contents and no official API (e.g. events, train schedules, etc.)

Scraping is good, but...

Through scraping you are able to collect the data you want, **but...**

Some websites have some protections against scraping. They are able to detect weird behaviours (e.g., repeated scrolling) and prevent you from scraping.

The problem to solve is to correctly and efficiently process HTML pages to extract valuable information. Since HTML code is **tagged text**, any tool that allows to manipulate and process text could be potentially used to parse a page.

Most websites have **dynamic-generated content**: downloading and parsing the HTML page could not be sufficient.

A preliminary step is retrieving a static page from the Web, using one of many tools that support this action:

- [curl](#), using command line
- [requests](#), using Python scripts

Another basic step is parsing the page, extracting data that respects specific patterns:

- [XPath](#)
- [Regular Expressions](#)

What happens if we want to obtain pages in sequence (e.g.: pages of reviews from Amazon or TripAdvisor)?

What we need to do if the page loads the content we need only after clicking a specific field or while scrolling the page itself (e.g.: Facebook and Instagram posts)?

Python is one of the best languages to build scrapers: it is easy to learn, fast to code and it has plenty of libraries to get anything done efficiently.

Two modules are needed to set up a scraper:

- [Selenium](#)
- [BeautifulSoup](#)

Selenium is a module developed for automating websites testing. It simulates clicks, drags, scrolls and any other possible interaction with a website.

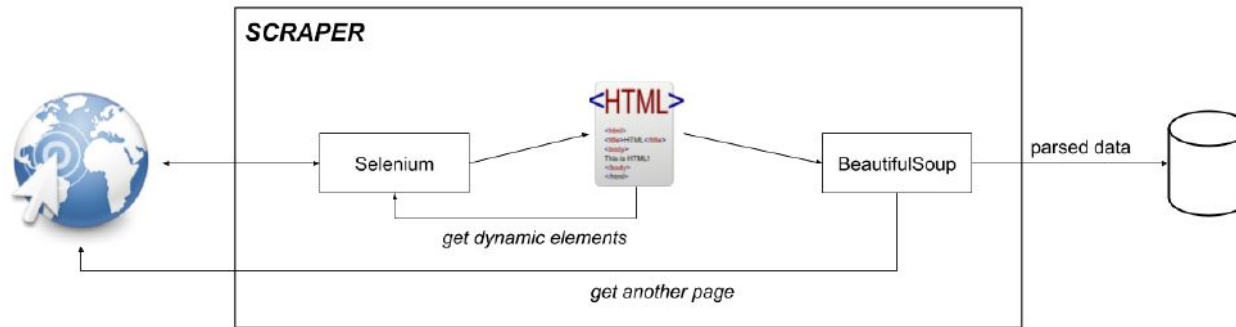
For web scraping, it is used to retrieve data that is not loaded with the source page or to navigate several pages of the site.

BeautifulSoup is an HTML (and XML) parser: it presents a set of API to navigate through an HTML document, accessing each tag and hierarchy easily.

In a scraper architecture, it represents the main source to extract data after the page is fully loaded.

High-level Pipeline

The image below represents the interactions between the two presented modules: their simple combination is the basis for a web scraper.



Most of the times, the HTML changes over time, especially the classes and sometimes the structure of the HTML.

Therefore, it is really important to keep the scraper **updated**.

An example of “Google Maps” scraping is available in the “Scraping” Folder.

N.B. within the folder, there is a “**chromedriver**” executable the you need to substitute with the one suited for you

- Check out your Google Chrome version
- Visit this [Link](#)
- Download and put the correct **chromedriver** version within the folder

ANY
Questions?

Postman Download - [Link](#)

Studio Ghibli API - [Link](#)

iTunes Store API - [Link](#)

Anaconda Download - [Link](#)

Anaconda Environment - [Link](#)

API & Scraping Exercises Scripts - [Link](#)