

Data Crawl with Python

Python + Selenium

IMDB Demo

July 28th, 2019

Outline

- ① **Introduction**
- ② **Python Installation**
- ③ **Selenium Installation**
- ④ **Browser Driver Installation**
- ⑤ **Find Data in Website**
- ⑥ **Appendix**

Introduction

- 通过 **Selenium** 与浏览器进行互动，**Python** 可以模仿浏览器操作，执行鼠标点击、键盘操作、网页数据获取等操作。在实际编程中，可以调出浏览器界面，进行可视化调整，十分方便快捷。结合简单的数据获取命令，**Python + Selenium** 即可完成数据爬取操作，同时 **Python** 可以继续用作后续的数据文件生成和数据处理等操作。
- 首先需要安装 **Python**，之后需要安装 **Selenium** 包。由于在调用浏览器操作的时候，需要使用浏览器驱动，还需下载浏览器驱动，并将下载好的浏览器驱动 (*chromedriver.exe*) 与编写的程序 (*.py*) 放在相同路径下。

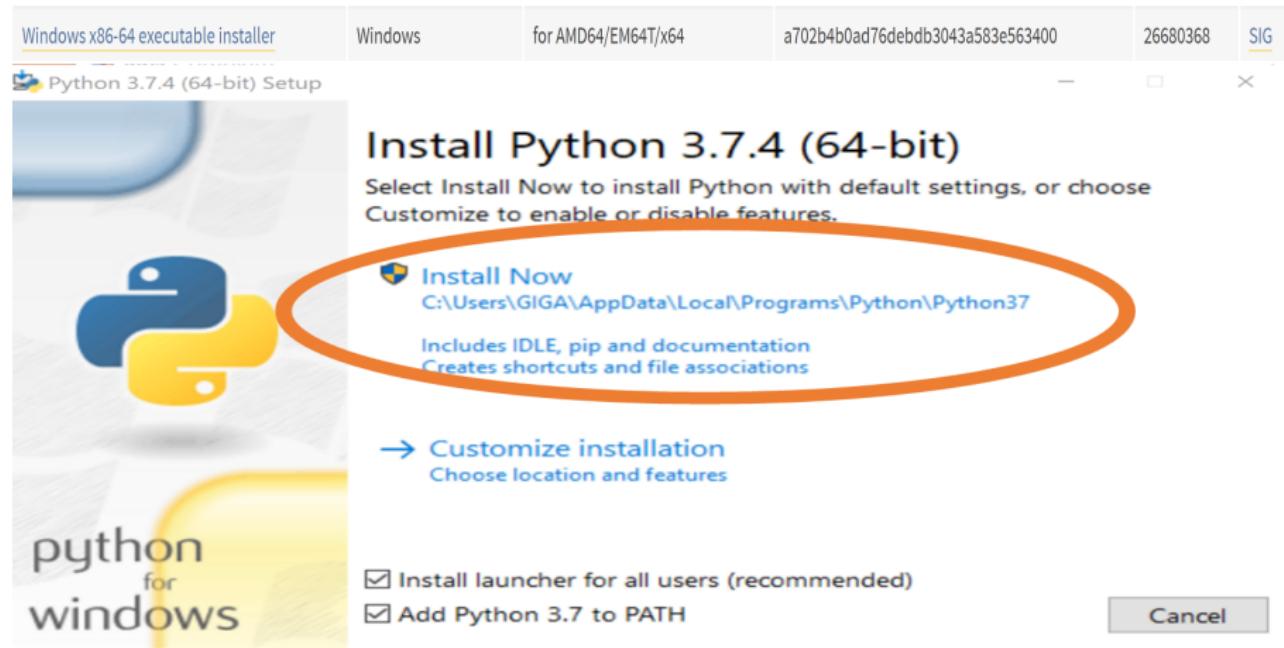
Examples

An example program for IMDB data crawling is given in *CrawlerDemo.py* and default chrome driver is given as *chromedriver.exe*

Python Installation

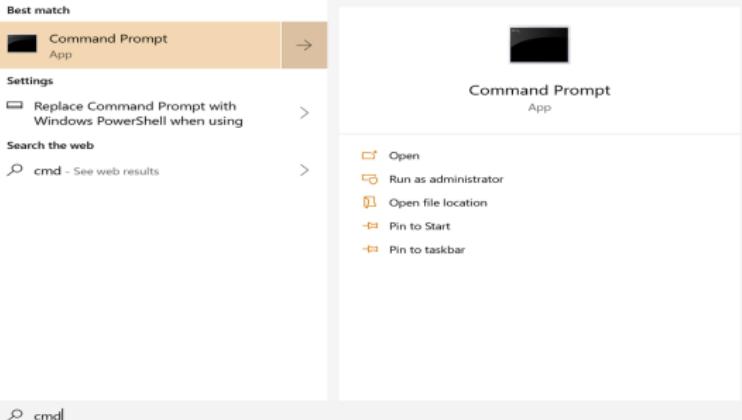
Python is available in following site...

<https://www.python.org/downloads/release/python-374>



Python Installation

After installing **Python**, use **cmd** to see whether the installation is successful. If console shows as following, installation is completed...



The screenshot shows the Windows Start Menu search interface. A search bar at the bottom contains the text "cmd". Above it, the "Best match" section highlights "Command Prompt App". Below this are sections for "Settings" (with an option to "Replace Command Prompt with Windows PowerShell when using"), "Search the web" (with a result for "cmd - See web results"), and a separator line.

On the right side, a detailed view of the "Command Prompt" app is shown, listing its context menu options: Open, Run as administrator, Open file location, Pin to Start, and Pin to taskbar.

```
C:\Windows\system32>python
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>

C:\Windows\system32>python
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()

C:\Windows\system32>
```

Selenium Installation

Selenium is required to perform visualized data crawl, thus this package should be downloaded and installed for our **Python**. **Pip** is a tool for **Python** to collect and install package from internet for local **Python**. "pip Install" command is used for this purpose...

```
C:\Windows\system32>pip install selenium
Collecting selenium
  Downloading https://files.pythonhosted.org/packages/80/d6/4294f0b4bce4de0abf13e17190289f9d0613b0
    /selenium-3.141.0-py2.py3-none-any.whl (904kB)
      73% |███████████| 665kB 534kB/s eta 0:00:01
```

Again, we use **cmd** to run "pip install" command. With above window showing "complete" information, the installation for **Selenium** is completed

Browser Driver Installation

Since **Selenium** runs by employing browser, browser driver is required. We use **Google Chrome** as our choice for browser and download its driver from website. A default version of *chromedriver.exe* is appended in our demo example...

The screenshot shows two windows side-by-side. On the left is a web browser displaying the official ChromeDriver page at chromedriver.chromium.org. The page has sections for 'Downloads', 'Getting started', 'Capabilities & ChromeOptions', and 'Current Releases'. It also includes links for 'chromium.org站内的其它相关信息' and 'If you are using Chrome from Dev or Canary channel, please download ...'. On the right is a screenshot of the Google Chrome application window. The menu bar at the top includes '文件(F)', '编辑(E)', '视图(V)', '设置(S)', '帮助(H)', and '退出(X)'. Below the menu bar, a context menu is open over a blank area of the browser window, showing options like '关于 Google Chrome(G)', '帮助中心(H)', and '报告问题(R)...'.

ChromeDriver - WebDriver for Chrome
chromedriver.chromium.org/ - 翻译此页
Getting started with ChromeDriver on Desktop (Windows, Mac, Linux) ... Security Considerations, with recommendations on keeping ChromeDriver safe

Downloads
If you are using Chrome version 76, please download ...
please download ...

Getting started
You can also read Getting Started with Android or Getting Started ...

Capabilities & ChromeOptions
Use the ChromeOptions class. This is supported by Java ...
chromium.org站内的其它相关信息 »

Current Releases

- If you are using Chrome version 76, please download [ChromeDriver 76.0.3809.68](#)
- If you are using Chrome version 75, please download [ChromeDriver 75.0.3770.142](#)
- If you are using Chrome version 74, please download [ChromeDriver 74.0.3729.6](#)
- If you are using Chrome version 73, please download [ChromeDriver 73.0.3683.68](#)
- For older version of Chrome, please see below for the version of ChromeDriver that supports it.

If you are using Chrome from Dev or Canary channel, please download [ChromeDriver 76.0.3809.68](#). This is not officially supported, but in most cases it should work without major issues.

For more information on selecting the right version of ChromeDriver, please

关于 Google Chrome(G)
帮助中心(H)
报告问题(R)... Alt+Shift+I

Google Chrome

Google Chrome 已是最新版本
版本 75.0.3770.142 (正式版本) (64 位)

获取有关 Chrome 的帮助

报告问题

Find Data in Website

We use our demo program to illustrate the procedures for locating elements in website and retrieving desirable data. Reference file operation for data processing and storage is also available in code...

In our demo, we want to collect information, including directors and rating for some movies, from **IMDB**. The search conditions include *Movie Name* and *Release Year*...

- Following code starts the search in **IMDB**

```
movieDirectorCrawled = list()
movieRatingCrawled = list()
for movie in movieInfoHolder:
    # visit the website
    chromeDriver.get(websiteCrawling)
    # locate and find search bar by xpath
    searchBar = chromeDriver.find_element_by_xpath("./input[@type='text'][@name='q'][@id='navbar-query']")
    # input movie name, movie[0] + Keys.ENTER perform the same as following button click
    searchBar.send_keys(movie[0])
    # locate and find search button by xpath
    searchButton = chromeDriver.find_element_by_xpath("./button[@id='navbar-submit-button'][@type='submit']")
    # click the search button
    searchButton.click()
    # sleep one seconds for loading the search results
    time.sleep(1)
```

Find Data in Website

- We need first find way to locate the desired data in website...



- Since we want to type movie name and then search in **IMDB**, the search bar and search button are wanted.
- We first locate the search bar for entering movie name...



Find Data in Website

- We want to find its corresponding position in the web source file...

The screenshot shows the IMDB homepage with a search bar at the top. The search bar has the placeholder text "Find Movies, TV shows, Celebrities and more...". Below the search bar, there's a section titled "Opening This Week" featuring movie posters and titles like "Once Upon a Time ... in Hollywood", "Skin", and "Mike Wallace Is Here". To the right of the search bar, the browser's developer tools are open, specifically the "Elements" tab. It highlights the search bar element with a yellow border and shows its HTML structure:

```
<input id="navbar-query" type="text" value="Find Movies, TV shows, celebrities and more..."/>
```

- We then locate the search button for submitting our search...

The screenshot shows the search results for "Testament of Youth" on the IMDB website. The search bar at the top contains the query "Testament of Youth". Below the search bar, the results are listed under the heading "Titles". The results include various entries such as "Testament of Youth (2014)", "Testament of Youth (1979) (TV Mini-Series)", and "1918 (1979) (TV Episode)". To the right of the search bar, the browser's developer tools are open, highlighting the search button with a yellow border. The button has the ID "navbar-submit-button.prim" and the class "primary btn". The surrounding HTML code includes the search input field and the button itself.

- We want to find its corresponding position in the web source file...

The screenshot shows the IMDB homepage again, with the search bar at the top containing the placeholder text "Find Movies, TV shows, Celebrities and more...". Below the search bar, there's a section titled "Opening This Week" with movie posters and titles. The browser's developer tools are open, highlighting the search button with a yellow border. The button has the ID "navbar-submit-button.prim" and the class "primary btn". The surrounding HTML code includes the search input field and the button itself.

Find Data in Website

- After entering movie name and click search button to search in IMDB...

The screenshot shows the IMDB search results for "Testament of Youth". The first result is "Testament of Youth (2014)". To the right, the browser's developer tools (F12) are open, displaying the HTML code for the search results. The code includes elements like `<tr>`, `<td>`, and `` pointing to the movie's page.

- Following code retrieves all valid options for our search, along with their link (use "Release Year" to identify the wanted movie)

```
# locate the section for movies, there are multiple "table" element with class as "findList", we want the first one
movieItemSection = chromeDriver.find_element_by_xpath("//[@id='main']/div/div[2]/table")
# find all movie items related with the searching movie name
movieItems = movieItemSection.find_elements_by_tag_name("tr")
```

- Corresponding movie link information in the web source file...

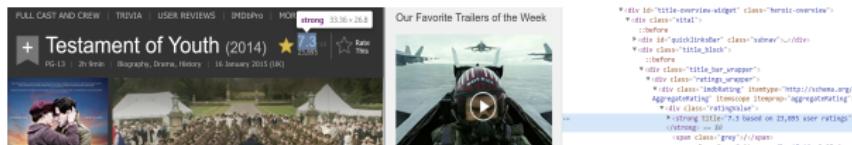
The screenshot shows the IMDB search results for "Testament of Youth". The first result is "Testament of Youth (2014)". To the right, the browser's developer tools (F12) are open, displaying the HTML code for the search results. The code includes elements like `<tr>`, `<td>`, and `` pointing to the movie's page.

Find Data in Website

- We go to detailed movie page to collect the information of movies...



- Corresponding rating information in the web source file...



- Corresponding director information in the web source file...



Find Data in Website

- Following code collect the director and rating information in each detailed movie page...

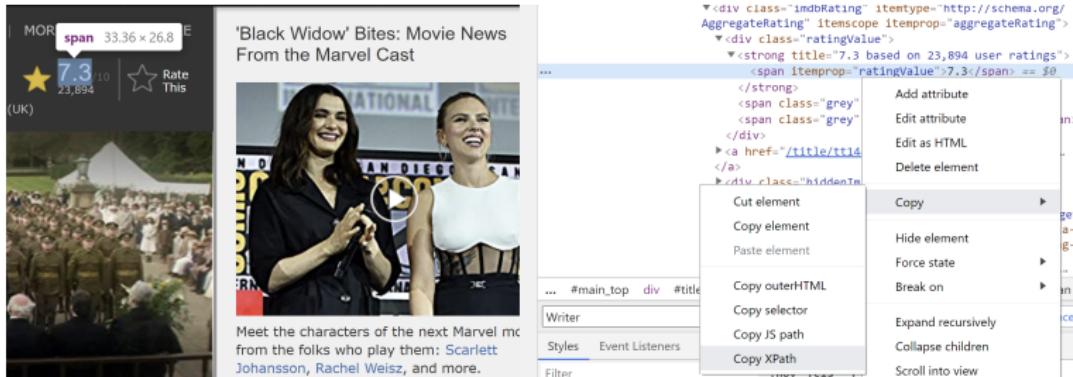
```
chromedriver.get(movieLink)
time.sleep(1)
rating = chromedriver.find_element_by_xpath(
    ".//*[@id='title-overview-widget']/div[1]/div[2]/div/div[1]/div[1]/div[1]/strong/span")
directorInfoElements = chromedriver.find_element_by_xpath(
    ".//*[@id='title-overview-widget']/div[2]/div[1]").find_element_by_xpath(
    ".//h4[contains(text(), 'Director')]/..").find_elements_by_tag_name("a")
directorList = [info.text for info in directorInfoElements]
movieDirectorCrawled.append(directorList)
movieRatingCrawled.append(rating.text)
break
```

Appendix

- How to run Python program in WIN cmd console?

```
C:\Users\GIGA>cd C:\Users\GIGA\PycharmProjects\DataCrawler  
C:\Users\GIGA\PycharmProjects\DataCrawler>python CrawlerDemo.py
```

- How to find and write the XPath of website element in easy way?



Reference

- About **XPath**

Reading materials and extended guide for **XPath**

- About **Python**

Reading materials and extended guide for **Python**