

# Hướng Dẫn Cào Dữ Liệu Từ Trang VNEXPRESS

## I. Cài thư viện cần thiết

- ✚ Cài đặt Requests: `pip install requests`
- ✚ Cài đặt Pillow: `pip install Pillow`
- ✚ Cài đặt BeautifulSoup4 : `pip install BeautifulSoup4`

## II. Cào dữ liệu lấy tin tức về thể thao

### 1. Lấy dữ liệu

- Dữ liệu lấy sẽ là dữ liệu thể thao của trang vnexpress vì vậy chúng ta sẽ vào trang : <https://vnexpress.net/the-thao>

### Thể thao

Xem theo ngày

Video Bóng đá Tennis VM 2020 Các môn khác Hậu trường Ảnh Tường thuật F1 Lịch thi đấu



#### Ciro Immobile và Chiếc Giày Vàng kém hào nhoáng

Ghi rất nhiều bàn thắng nhưng tiền đạo của Lazio **Ciro Immobile** chưa bao giờ được xem là một ngôi sao, ngay cả khi đoạt chiếc Giày Vàng châu Âu mùa này.

44

#### Arteta: 'Đừng ví chúng tôi như Guardiola và Klopp'

Theo HLV Mikel Arteta còn quá sớm để

#### Thủ môn Bồ Đào Nha hé lộ cách cản phạt đền của Fernandes

Miguel Soares, thủ thành gần nhất cản

#### Man City bị tố gian dối trước tòa

Báo Đức Der Spiegel tiếp tục cung cấp bằng chứng cho thấy dấu hiệu gian lận tài chính của Man City.

- Tiếp theo ta sẽ lấy dữ liệu từ trang này qua BeautifulSoup. Thư viện này sẽ giúp chúng ta phân tách dữ liệu HTML hay XML thành dữ liệu cây, giúp cho dữ liệu gọn gàng hơn :

```
import requests
from bs4 import BeautifulSoup

response = requests.get("https://vnexpress.net/the-thao")
soup = BeautifulSoup(response.content, "html.parser")
print(soup)
```

- Kết quả chúng ta sẽ được trang html.

## 2. Phân tích dữ liệu

- Tiếp theo chúng ta sẽ phân tích dữ liệu cần lấy. Để xem chi tiết dữ liệu của một bài báo, chúng ta mở <https://vnexpress.net/the-thao> và ta nhấn F12.
- Mục tiêu chúng ta là tìm ra link của bài báo để trích xuất dữ liệu của từng bài báo. Sau khi phân tích ta thấy rằng link của bài báo nằm trong thẻ <a></a> và thẻ này nằm trong thẻ h3 và có class là “tilte\_news”. Chúng ta sẽ tiến hành lấy link như sau:

```
titles = soup.findAll('h3', class_='title-news')
print(titles)

links = [link.find('a').attrs["href"] for link in titles]
print(links)
```

- Kết quả chúng ta sẽ được danh sách các link của bài báo

```
['https://vnexpress.net/ciro-immobile-va-chiec-giay-vang-kem-hao-nhoang-4139575.html', 'https://vnexpress.net/lampard-chang-ai-nho-doi-a-quan-cup-fa-4139548.html', 'https://vnexpress.net/pirlo-tu-choi-ngoai-hang-anh-de-lam-hlv-u23-juventus-4139649.html']
```

### 3. Lấy dữ liệu chi tiết từng bài

- Sau khi ta đã có link, chúng ta sẽ tiến hành lấy dữ liệu.

```
for link in links:
```

```
    news = requests.get(link)
```

```
    print(news)
```

```
    soup = BeautifulSoup(news.content, "html.parser")
```

```
    print(soup)
```

```
    title = soup.find("h1", class_="title-detail")
```

```
    print("Tiêu đề: " + title.text)
```

```
    abstract = soup.find("p", class_="description")
```

```
    print("Mô tả: " + abstract.text)
```

```
    body = soup.find("p", class_="Normal")
```

```
    print("Nội dung: " + body.text)
```

- Bước đầu tiên chúng ta dùng vòng lặp for duyệt qua từng link, sau đó với mỗi link chúng ta sẽ dùng BeautifulSoup như trên và tiếp tục vào link của bài báo để phân tích. Ví dụ như : tiêu đề của bài báo nằm trong thẻ “h1” và nằm

trong class “title-detail”. Ở đây chúng ta sẽ lấy tiêu đề, mô tả, nội dung.

- Cuối cùng chúng ta sẽ lưu dữ liệu vào file json. Chúng ta sẽ tận dụng thuộc tính \_\_dict\_\_ để xử lý dữ liệu như sau (code hoàn chỉnh để cào dữ liệu):

```
import requests

from bs4 import BeautifulSoup

import json

import pandas as pd


response = requests.get("https://vnexpress.net/the-thao")
soup = BeautifulSoup(response.content, "html.parser")
#print(soup)


titles = soup.findAll('h3', class_='title-news')
#print(titles)


links = [link.find('a').attrs["href"] for link in titles]
#print(links)


class Insert (object):

    def __init__(self, tilte, abstract, body):

        self.tilte = tilte

        self.abstract = abstract

        self.body= body
```

```
last_j=[]
for link in links:
    news = requests.get(link)
    #print(news)
    soup = BeautifulSoup(news.content, "html.parser")
    #print(soup)
    tilte = soup.find("h1", class_="title-detail")
    #print("Tiêu đề: " +tilte.text)
    abstract = soup.find("p", class_="description")
    #print("Mô tả: " +abstract.text)
    body = soup.find("p", class_="Normal")
    #print("Nội dung: " +body.text)
    #dùng thuộc tính dict
    insert = Insert(tilte.text,abstract.text,body.text)
    last_j.append(insert.__dict__)

# Lưu dữ liệu vào file
with open('data.json', 'w', encoding='utf-8') as file:
    json.dump(last_j, file)
```

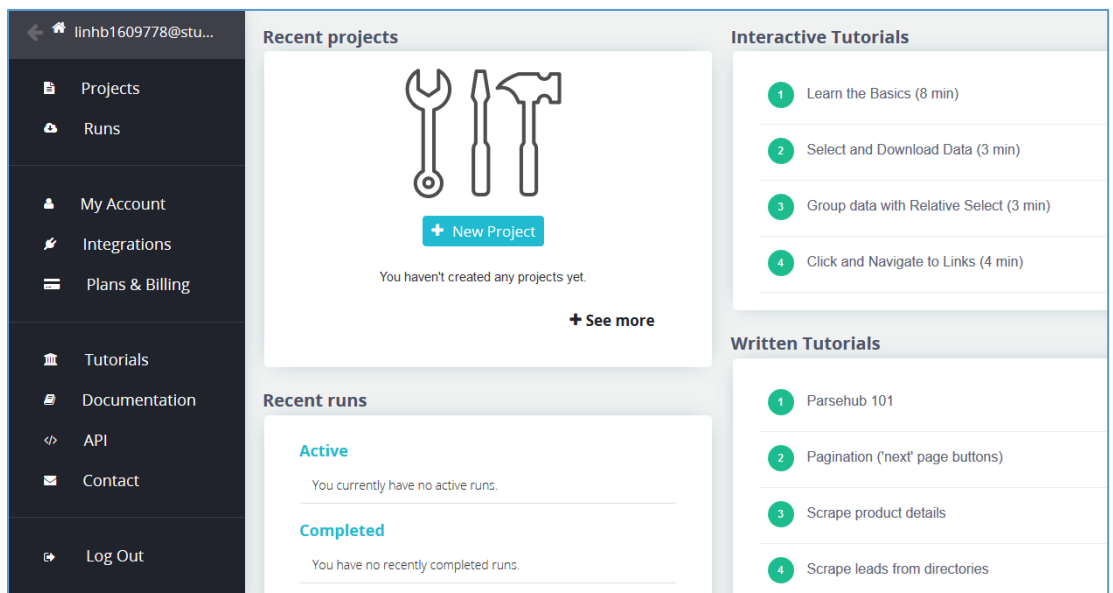
### III. Cào dữ liệu với ParseHub

#### 1. Giới thiệu ParseHub

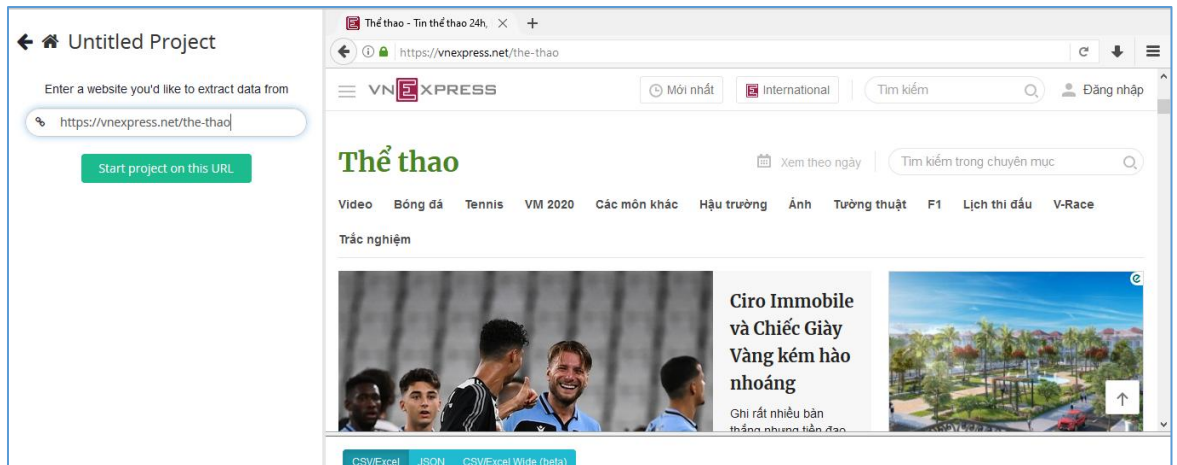
- Phần hướng dẫn bên trên là cách cào dữ liệu "truyền thống". Công cụ sau sẽ hỗ trợ chúng ta việc cào dữ liệu dễ dàng hơn.

#### 2. Chi tiết cào với ParseHub

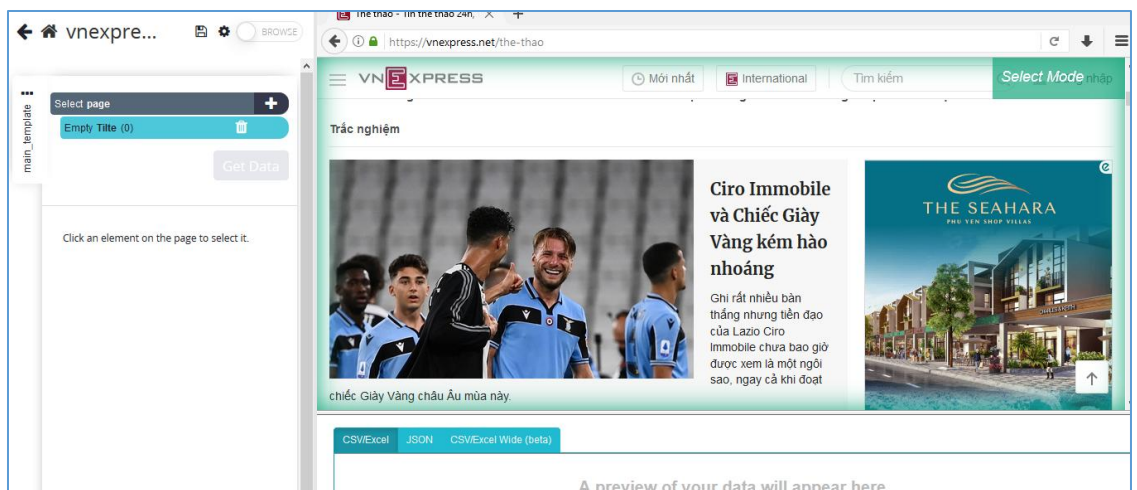
- Tải ParseHub : <https://www.parsehub.com/>
- Đầu tiên để sử dụng được công cụ này, chúng ta cần đăng kí tài khoản qua link: <https://www.parsehub.com/>
- Như vậy chúng ta sẽ có giao diện của ParseHub như sau:



- Chúng ta sẽ tiến hành tạo Projects mới để tiến hành cào dữ liệu.
- Chọn link trang web mà bạn muốn chọn, ở đây chúng ta sẽ lấy trang web: <https://vnexpress.net/the-thao>



- Chúng ta sẽ điền link trang web vào phần dưới của mục "Enter a website you'd like to extract data from" và sau đó ta sẽ nhấn "Start project on this URL".



- Ở dưới phần Select page chính là thẻ để bạn lấy thông tin ví dụ như gọi thẻ này là Title, thẻ này sẽ lấy các phần tên và link của tiêu đề bài viết. Cách lấy như sau: bạn chỉ cần chọn vào tiêu đề đó và kéo xuống dưới tiếp tục chọn tiêu đề thứ hai, công cụ này sẽ giúp bạn lấy toàn bộ tiêu đề còn lại.
- Kết quả ta được như sau:

The interface shows a template configuration on the left and a preview of a news article on the right. The template configuration includes a 'Select page' dropdown, a 'Select Title (44)' button, and fields for 'Extract name' and 'Extract url'. A 'Get Data' button is at the bottom. The 'Selection Node' section shows a tree structure: 'All element s' > 'All h3 s' > 'All element s'. A checkbox for 'Wait up to 60 seconds for elements to appear' is also present.

The preview shows a news article from VNEXPRESS titled 'Ciro Immobile và Chiếc Giày Vàng kém hào nhoáng'. The article text mentions 'Ghi rất nhiều bàn thắng nhưng tiền đạo của Lazio' and 'Ciro Immobile chưa bao giờ được xem là một ngôi sao, ngay cả khi đoạt chiếc Giày Vàng châu Âu mùa này.' Below the preview, there are tabs for 'CSV/Excel', 'JSON', and 'CSV/Excel Wide (beta)'. The JSON output is shown as follows:

```
{
  "Title": [
    {
      "name": "Ciro Immobile và Chiếc Giày Vàng kém hào nhoáng",
      "url": "https://vnexpress.net/ciro-immobile-va-chiec-giay-vang-kem-hao-nhoang-4139575.html"
    }
  ]
}
```

- Kết quả trả về sẽ là json hoặc là csv tùy theo lựa chọn của bạn. Cuối cùng bạn chỉ cần nhấn Get Data để lấy dữ liệu.
- Sau đó nhấn thêm RUN.

The 'Download Data' section shows three buttons: 'CSV/Excel', 'JSON', and 'API'. Below them, a box displays 'Template Name: main\_template' and 'Pages Scraped: 1'. A note states: 'All dates and times are in UTC +0000. Empty file with no results? [Click here](#) to fix. CSV file too big? Save the JSON file and [click here](#) to convert to CSV.'

The 'Run Details' section shows the following information:

Field	Value
Status	complete
Pages	1 collected
Initialized	2020-08-01T08:37:29
Start Time	2020-08-01T08:37:30
Finished	2020-08-01T08:37:45
API Key	tRUar1IWf0fO
Project Token	t2p9MopBxGJI
Run Token	t_nJz8ddTYVM

The 'Settings' section shows the following configuration:

Field	Value
URL	https://vnexpress.net/the-thao
Starting Template	main_template
Starting Value	{}
Load Javascript	true
Rotate IPs	false



- Và cuối cùng bạn sẽ tiến hành tải dữ liệu về bằng các định dạng như csv hay json hoặc là kết nối cấu hình API để lấy dữ liệu trực tiếp.
- Để hiểu rõ hơn về phần này các bạn có thể truy cập vào: <https://help.parsehub.com/hc/en-us>.

**HẾT**