**LAB - Python Requests**

Viết chương trình dùng ngôn ngữ Python và thư viện requests, json để lấy dữ liệu từ API, xử lý và lưu xuống tập tin json

## **Danh sách các API mẫu**

* # https://dummyapi.io/data/v1/user?limit=10
* # https://dummyapi.io/data/v1/user/60d0fe4f5311236168a109ca
* # https://dummyapi.io/data/v1/user/60d0fe4f5311236168a109ca/post?limit=10
* # https://dummyapi.io/data/v1/post?limit=10
* # https://dummyapi.io/data/v1/post/60d21af267d0d8992e610b8d/comment?limit=10
* # https://dummyapi.io/data/v1/tag?limit=10
* # https://dummyapi.io/data/v1/tag/water/post?limit=10

## **Mô tả**

* Bài thực hành yêu cầu kết nối đến danh sách các api ở trên bằng thư viện requests.
* Trên cơ sở này, học viên thực hiện xử lý và lưu trữ những dữ liệu này xuống tập tin json.
* Máy PC/Laptop phải đáp ứng yêu cầu đã cài đặt các phần mềm/ứng dụng cơ bản cho developer.
* IDE: VSCode, IntelliJ
* Phần mềm/Ứng dụng/Công cụ: Python, Git

## **Yêu cầu kỹ thuật**

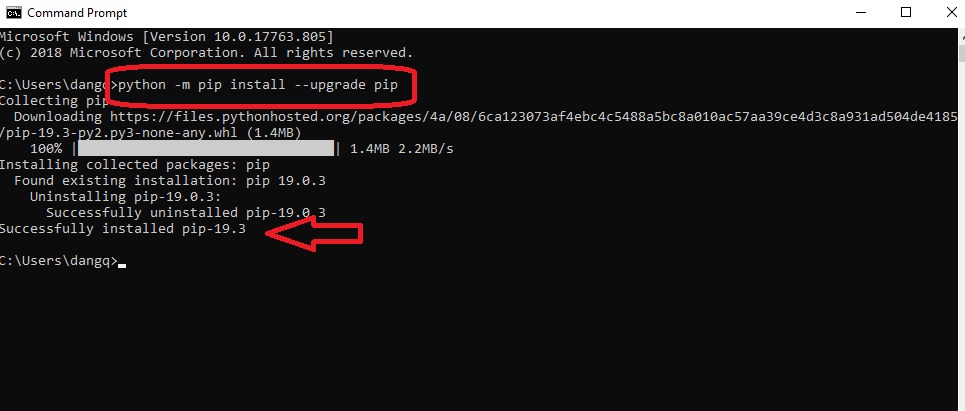
* Cài đặt thư viện requests trên máy tính.
* Viết code bằng Python thực hiện yêu cầu:
* Kết nối api lấy dữ liệu thành công.
* Thực hiện xử lý dữ liệu và lưu trữ xuống tập tin json.
* Hiển thị kết quả.

## **Các bước thực hiện**

### **Bước 1: Cài đặt thư viện Request.**

Từ PC bấm tổ hợp phím Windows + R , cmd ( yêu cầu internet )

Nâng cấp thư viện Pip :

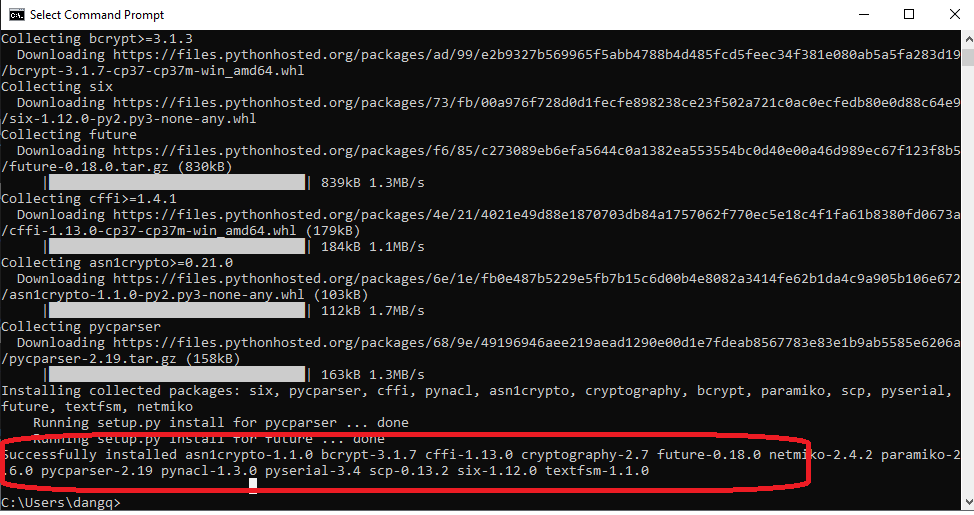


Cài đặt thư viện requests :



(**pip3 install request --user** đối với Python 3)

Kết quả :



### **Bước 2: Viết code python.**

Vào phần mềm VSCode/IntelliJ, tạo tập tin mới request\_handler.py và save C:\python

**import** thư viện **json, requests**

| import requests  import json |
| --- |

**Cấu hình danh sách API**

### **Bước 3: Thực hiện cấu hình danh sách api.**

| # base url  base\_url = "https://dummyapi.io/data/v1"  # key  headers = {"app-id": "61a1b1a6334d1b538e619c8f"}  limit = "?limit=10"  # API user  user\_list = base\_url + "/user" + limit  user\_detail = base\_url + "/user/%s" # user\_profile  user\_post = base\_url + "/user/%s/post" + limit # user\_profile post  # API post  post\_list = base\_url + "/post" + limit  comment\_list = base\_url + "/post/%s/comment" + limit  # API tag  tag\_list = base\_url + "/tag" + limit  post\_by\_tag = base\_url + "/tag/water/post" + limit |
| --- |

### **Bước 4: Thực hiện xử lý**

tạo hàm/phương thức xử lý request: get\_data\_by\_url

| def get\_data\_by\_url(url, id=None):  if id:  url = url % id  response = requests.get(url, headers=headers)  if response.ok:  content = json.loads(response.content)  if 'data' in content:  return content['data']  else:  return content  else:  # If response code is not ok (200), print the resulting http error code with description  response.raise\_for\_status() |
| --- |

**Lưu ý: các thuộc tính trên phải để chính xác như trên**

Tiếp theo, viết phương thức/hàm hiển thị dữ liệu: show\_data

| def show\_data(data):  print("The response contains {0} properties".format(len(data)))  print("\n")  if isinstance(data, list):  for item in data:  for key in item:  value = item[key]  if isinstance(value, dict) or isinstance(value, list):  value = json.dumps(value)  print(key + " : %s " % value)  else:  for key in data:  value = data[key]  if isinstance(value, dict):  value = json.dumps(value)  print(key + " : %s " % value) |
| --- |

Lấy danh sách các user, thông tin user, các bài post

| # get all user  users = get\_data\_by\_url(user\_list)  # show\_data(users)  for user in users:  user\_id = str(user['id'])  # get user profile  user\_profile = get\_data\_by\_url(user\_detail, user\_id)  # print(user\_profile)  show\_data(user\_profile)  #load post by user id  post = get\_data\_by\_url(user\_post, user\_id)  # print(user\_profile)  show\_data(post) |
| --- |

**Kiểm tra :**

Vào command line, thực hiện lệnh:

| cd C:\python  python request\_handler.py |
| --- |

**Kết quả:** trả về các dữ liệu đã được xử lý và tạo tập tin json thành công