

# API là gì?

-Application Programming interface-Giao diện lập trình ứng dụng

+ Bạn có 1 màn hình và 1 máy tính, muốn 2 đối tượng này kết nối/ giao tiếp được với nhau, bạn cần dây kết nối để màn hình và máy tính kết nối được với nhau thì phải dùng đúng dây đúng cổng, ta hiểu dây nối đó là ngôn ngữ chung của cả 2, và đó chính là API.

+Hệ thống đặt xe Grab, gồm 2 ứng dụng là: app tài xế và app khách hàng. KH chọn thông tin điểm đến, hệ thống tính giá và nhờ API để hiển thị thông tin điểm đến điểm đón đó trên app tài xế, giúp tài xế nắm thông tin quãng đường.

## RESTful API?

-Định nghĩa: Là 1 tiêu chuẩn được sử dụng trong việc thiết kế API cho các phần mềm, ứng dụng, web service để tạo sự thuận tiện cho quản lý resources

-Cấu trúc:

+ReQuest:

- \*URL: địa chỉ là định vị 1 tài nguyên duy nhất(webpage, img, video) –gọi là resources.

- \*GET: Y/c server trả ra thông tin 1 resource (mở Menu-> hiển thị list món ăn)

- \*POST:Y/c server tạo ra 1 resource mới (Tạo đơn hàng trên Shopee)

- \*PUT:Y/c server cho them/sửa vào 1 resource vào hệ thống đang có (Sửa stt, cmt trên Fb)

- \*DELETE: Y/c server xóa 1 resource (Xóa 1 bức ảnh trên fb)

+Headers: Nơi chứa thông tin cần thiết của req, các thông số kèm theo khi gửi yêu cầu lên server, cái mà end-users ko biết đến (độ dài req body, time gửi req, thiết bị?...)

+Body: Nơi chứa thông tin mà client nhập vào( Đặt grab->thông tin trong body sẽ là: sdt khách hàng, địa chỉ?...)

+Response:

- \*Status code: Là những con số có 3 chữ số và duy nhất 1 nghĩa

- +1xx:Thể hiện thông tin

- +2xx: Thể hiện sự thành công

- +3xx: Thể hiện sự điều hướng (305: use proxy)

- +4xx:Thể hiện lỗi client(404: not found)

- +5xx: Thể hiện lỗi server

# Dữ liệu dạng Json và XML

+Json

```
{  
  "key1": "value1"  
  "key2": ""value2"
```

.....

Ví dụ Json:

```
{  
  "id": 123  
  "name": "dung"  
}
```

+XML

```
<key><value></key>
```

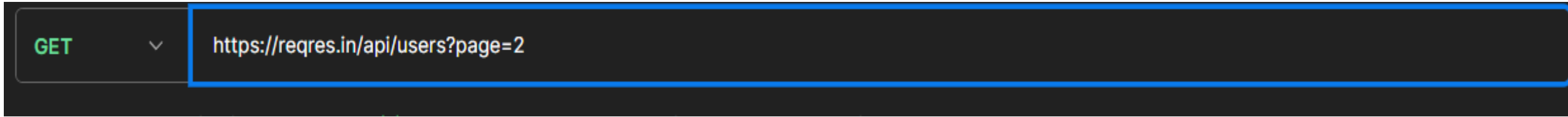
```
<id><123></id>
```

```
<name><dung></name>
```

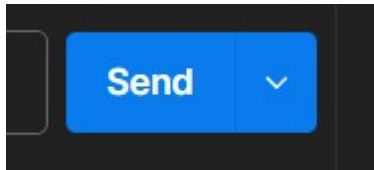
# Test API bằng Postman

+) Tạo trang mới (+) rồi nhập URL

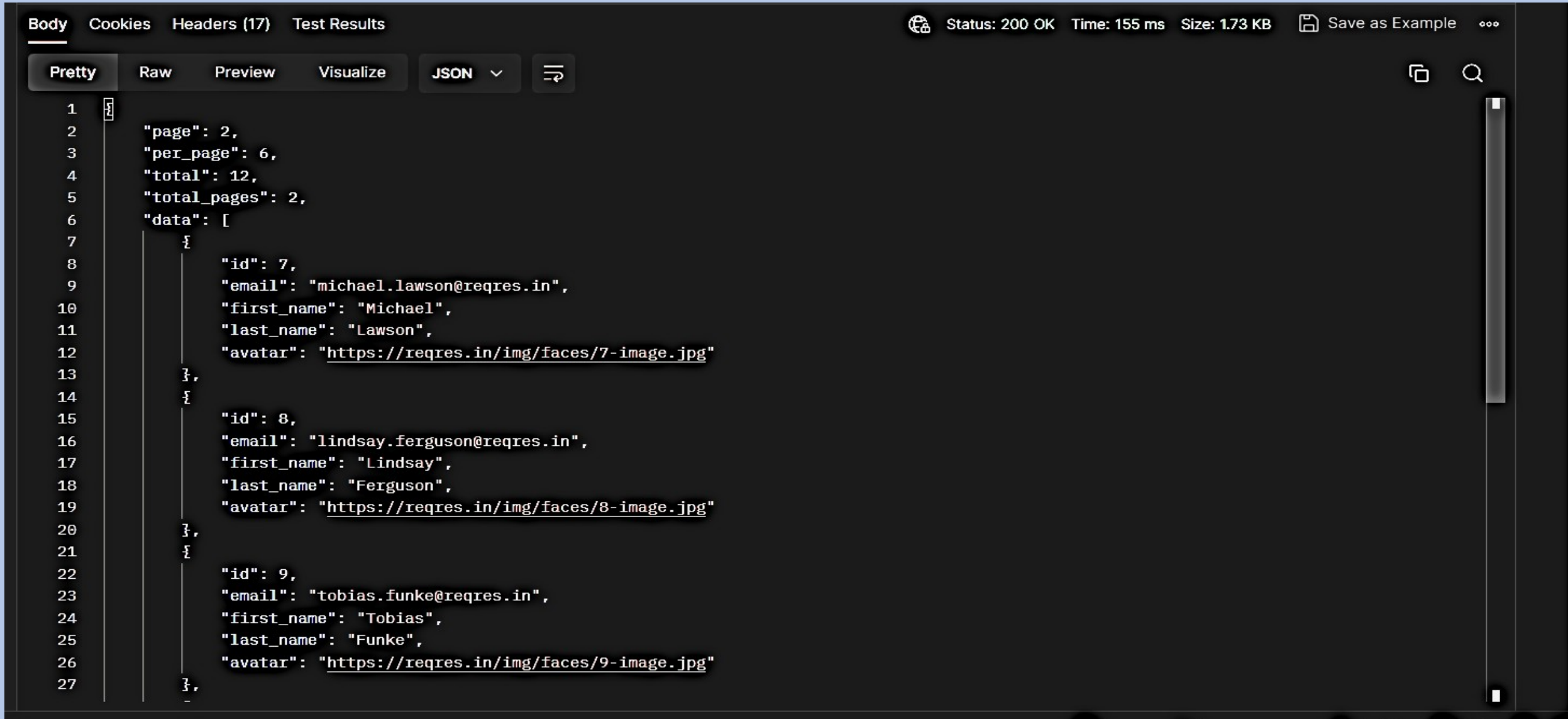
+) Bằng phương thức GET để yêu cầu server gửi list users



+) Bấm “Send”



+ ) Server đã gửi list users thành công ở “Budy” trong client người dung dưới dạng Json



The screenshot shows the 'Body' tab of a web browser's developer tools. The status bar at the top indicates 'Status: 200 OK', 'Time: 155 ms', and 'Size: 1.73 KB'. The JSON is displayed in a 'Pretty' format. The response structure includes pagination metadata and a list of user objects.

```
1 {
2   "page": 2,
3   "per_page": 6,
4   "total": 12,
5   "total_pages": 2,
6   "data": [
7     {
8       "id": 7,
9       "email": "michael.lawson@reqres.in",
10      "first_name": "Michael",
11      "last_name": "Lawson",
12      "avatar": "https://reqres.in/img/faces/7-image.jpg"
13    },
14    {
15      "id": 8,
16      "email": "lindsay.ferguson@reqres.in",
17      "first_name": "Lindsay",
18      "last_name": "Ferguson",
19      "avatar": "https://reqres.in/img/faces/8-image.jpg"
20    },
21    {
22      "id": 9,
23      "email": "tobias.funke@reqres.in",
24      "first_name": "Tobias",
25      "last_name": "Funke",
26      "avatar": "https://reqres.in/img/faces/9-image.jpg"
27    }
28  ]
29 }
```

Ngoài ra có thể đặt tên biến  
cú pháp sử dụng {{tên\_biến}}

url						Fork   0		Save	Share	...	
<input type="text" value="Filter variables"/>											
	Variable	Type	Initial value	Current value							
<input checked="" type="checkbox"/>	url	default	https://reqres.in	https://reqres.in							
<input checked="" type="checkbox"/>	name	default		Name-80725							
	Add new variable										

Với phương thức POST  
-Tạo mới users với “name” và “job”

The screenshot displays a REST client interface with a dark theme. At the top, the URL bar shows `{{url}}/api/users` with a dropdown menu set to **POST**. To the right of the URL bar are buttons for **Save**, **Send**, and a lightbulb icon. Below the URL bar, a tabbed interface includes **Params**, **Authorization**, **Headers (9)**, **Body** (selected), **Pre-request Script**, **Tests**, and **Settings**. The **Body** tab shows a JSON payload in raw format: 

```
{  "name": "morpheus",  "job": "leader"}
```

. The bottom section of the interface shows the response under the **Body** tab, with a status of **201 Created**, a time of **404 ms**, and a size of **746 B**. The response body is displayed in a pretty-printed JSON format: 

```
{  "name": "morpheus",  "job": "leader",  "id": "527",  "createdAt": "2023-06-19T01:32:55.486Z"}
```

-Ngoài ra Tests có thể giúp người dùng tìm kiếm, duyệt nhanh,thêm các thông tin bằng Snippets có sẵn.

The screenshot displays the Postman application interface. At the top, a POST request is configured to the endpoint `{{url}}/api/users`. The 'Tests' tab is active, showing a series of JavaScript test scripts:

```
1 pm.test("Status code is 201", function () {
2   pm.response.to.have.status(201);
3 pm.test("Body matches string", function () {
4   pm.expect(pm.response.text()).to.include("leader");
5 pm.test("check job is leader", function () {
6   var jsonData = pm.response.json();
7   pm.expect(jsonData.job).to.eql("leader");
8 });
9 };
10 };
```

Below the test scripts, the 'Body' tab is selected, showing the response in 'Pretty' JSON format:

```
{
  "name": "Name-23116",
  "job": "leader",
  "id": "360",
  "createdAt": "2023-06-19T01:38:33.959Z"
}
```

The status bar at the bottom indicates a successful response with status 201 Created, a time of 375 ms, and a size of 754 B. The interface also includes tabs for Params, Authorization, Headers (9), Body, Pre-request Script, Tests, Settings, Cookies, and Test Results (3/3).



END