

VII. Cách sử dụng Environments :

Chức năng chính của Environment là 1 chỗ lưu “biến” giống như “biến” trong code để có thể tái sử dụng ở nhiều nơi.

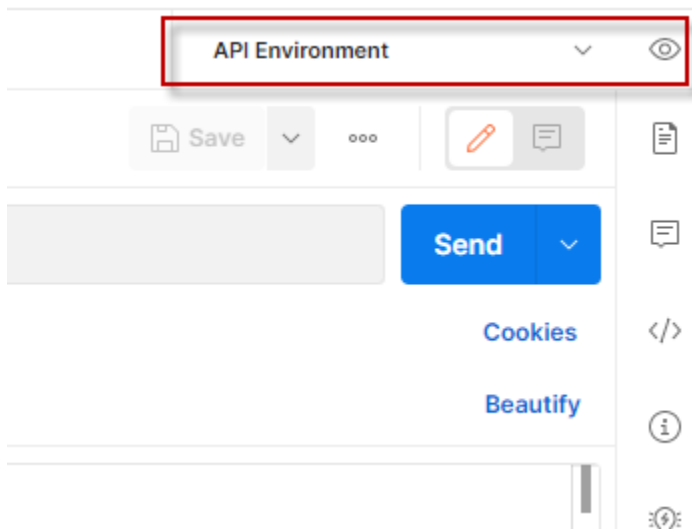
Ứng dụng:

- Nhanh chóng chuyển qua lại giữa các môi trường Dev và Product mà không cần tạo lại các request mới vì phải thay đổi lại URL.
- Giúp lưu lại giá trị của response API trước để điền vào API sau. (Phần này có kết hợp với phần Pre-request và Tests, sẽ được giới thiệu ở các phần tiếp theo).
- Không phải sửa giá trị của các tham số quá nhiều lần.

Ở Postman sẽ chia làm 2 loại Environments: Local và Global

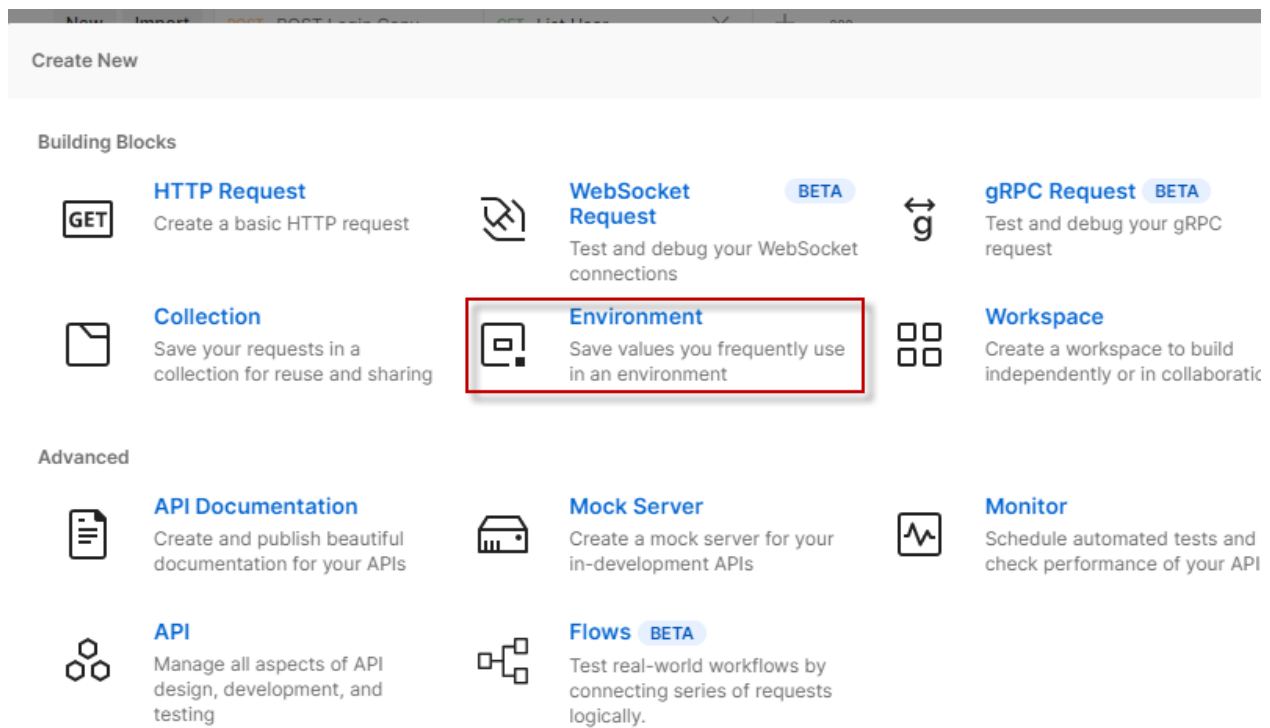
- **Local:** Phạm vi ảnh hưởng chỉ có khi chọn đúng Enviroments.
- **Global:** Phạm vi ảnh hưởng đến toàn bộ các project có trong Postman, nhưng nếu có 2 biến cùng tên ở Local và Global thì sẽ ưu tiên lấy Local.

Vị trí của Environment trong khung làm việc của postman.



Tạo 1 Enviroment:

Bước 1 : Chọn File → New → Environments



Bước 2 : Điền tên của Enviroment, tên và giá trị của biến.
 Ở đây, lấy ví dụ với biến `userProd` có giá trị là `abc123`

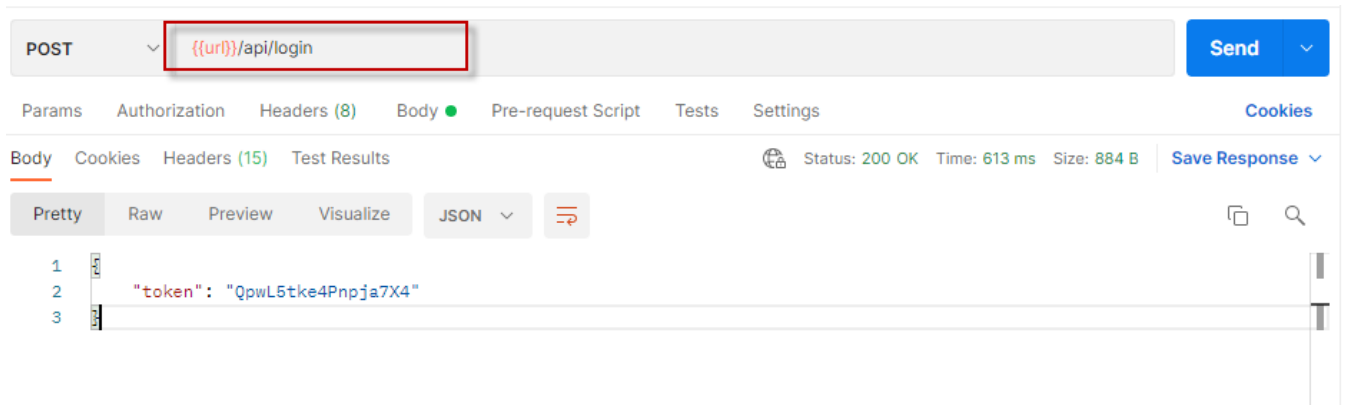
Production Environment					Fork 0			Save	Share	...	?
	VARIABLE	TYPE ?	INITIAL VALUE ?	CURRENT VALUE ?	...	Persist All	Reset All				
<input checked="" type="checkbox"/>	userProd	default	abc123								
	Add a new variable										

Dấu checkbox thể hiện rằng: có active cái biến đó hay không.

Cách sử dụng biến:

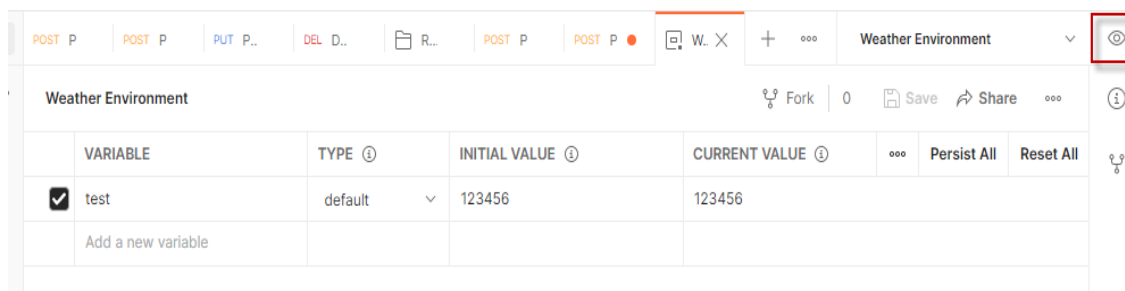
1. Lấy giá trị của các biến trên như thế nào trong request?

Trả lời: Chỉ cần viết theo cú pháp `{{tên_biến}}`: ví dụ: `{{url}}`



2. Làm thế nào để biết được giá trị của các biến trong 1 enviroment nào đó.

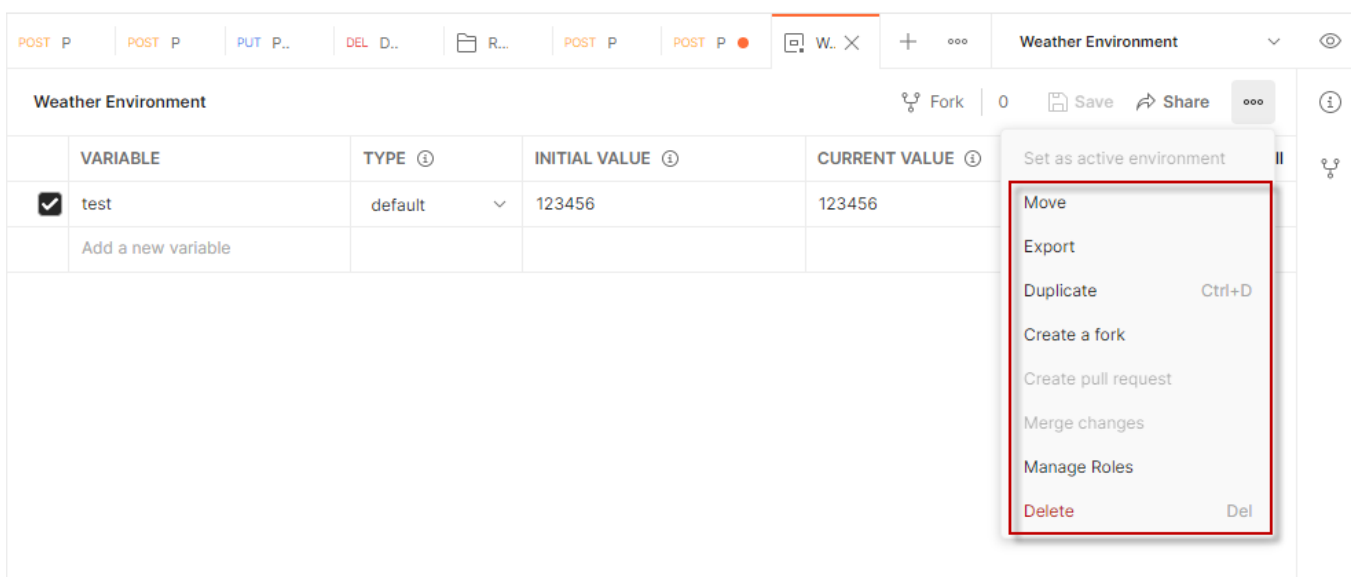
Trả lời: Click vào icon con mắt



Muốn sửa thông tin của Environment bạn có thể click vào nút *Edit* để sửa.

3. Có thể làm những gì với 1 Enviroment?

Trả lời: Postman cung cấp những chức năng đơn giản cho 1 Environment như: import – export, duplicate, add, edit, delete.

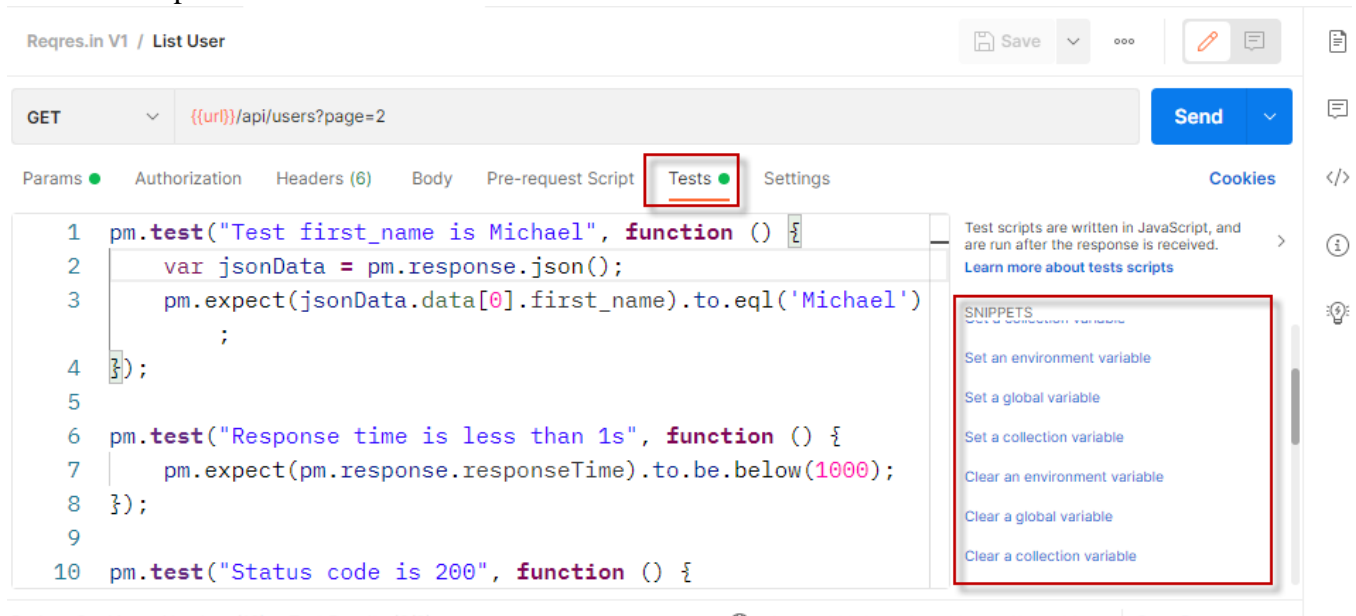


VIII. Test Response:

Test response là tính năng đặc biệt quan trọng với những người test API.

Phần này cung cấp 2 tính năng giúp người test đẩy nhanh được tốc độ test API.

1. Check tự động kết quả trả về của từng field với 1, 2 dòng code.
2. Lưu giá trị của Response thành biến trong Environment để tiếp tục truyền vào param của API tiếp theo.



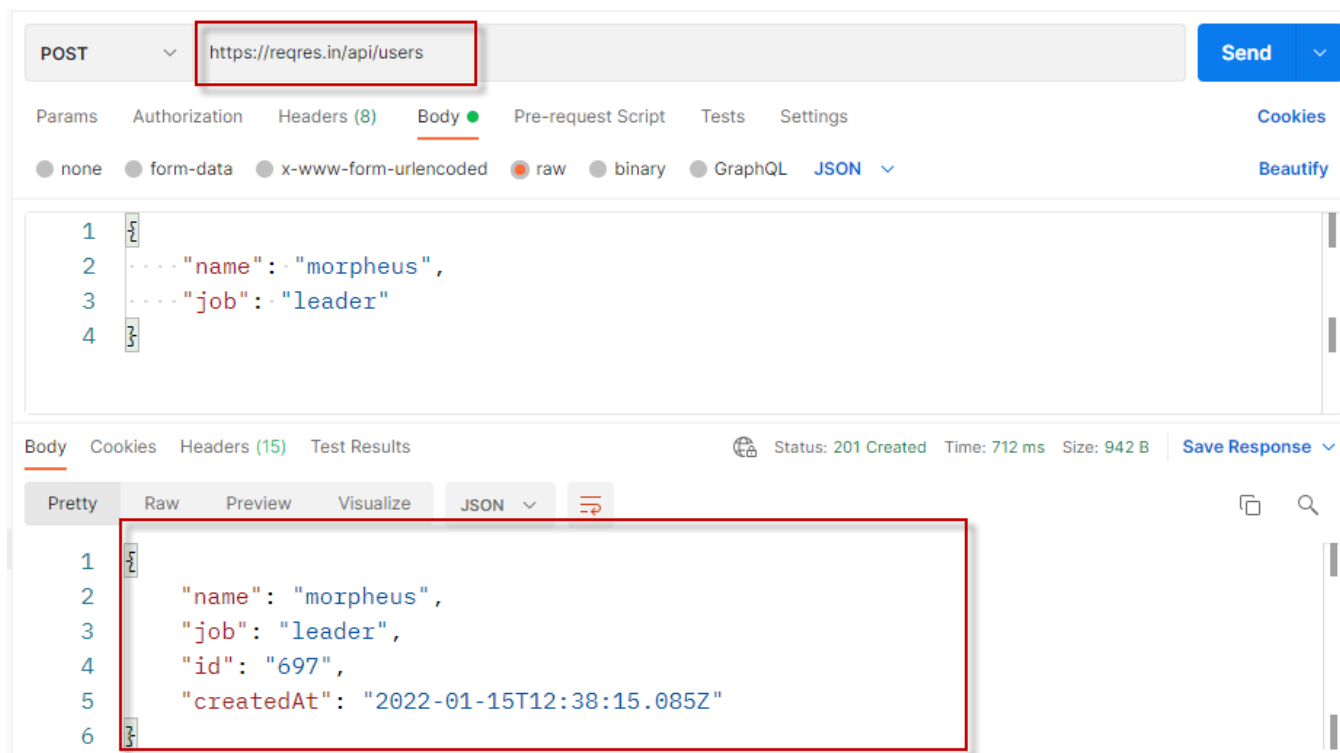
Postman cung cấp một khung làm việc để ta có thể làm việc, chỉ hỗ trợ ngôn ngữ Javascript thuần, không hỗ trợ jquery hay các thứ khác.

Phần ở bên phải là tập hợp những cú pháp Postman cung cấp sẵn cho người dùng.

Thực hành:

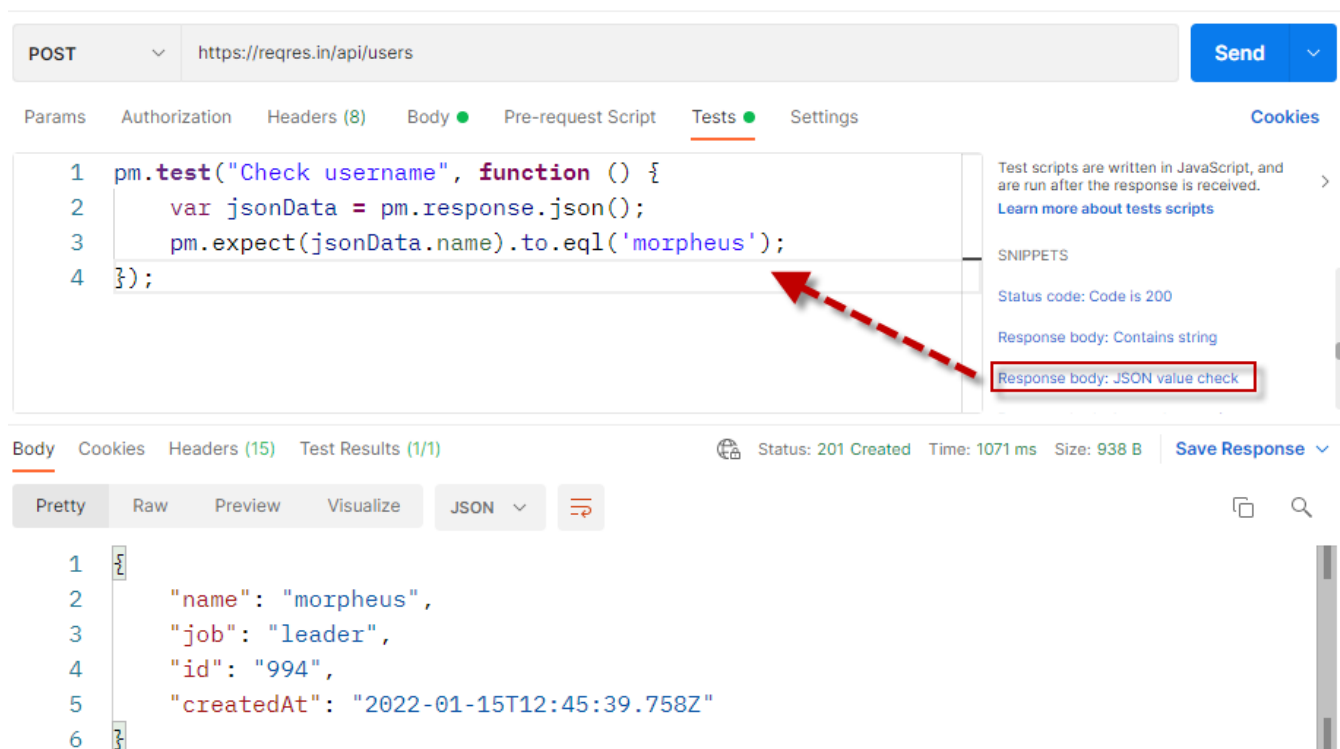
Bài tập 1: Tạo 1 request có phương thức POST-tạo mới 1 user, muốn biết là sau khi tạo xong, username trả về có đúng hay không.

Bước 1 : Chạy thử API 1 lần để lấy được cấu trúc Response của API. (Link: <https://reqres.in/api/users>)



Ta có thể thấy `user_id` nằm ở vị trí: `root > name` và trong trường hợp này id của user này là `morpheus`

Bước 2 : Viết Test

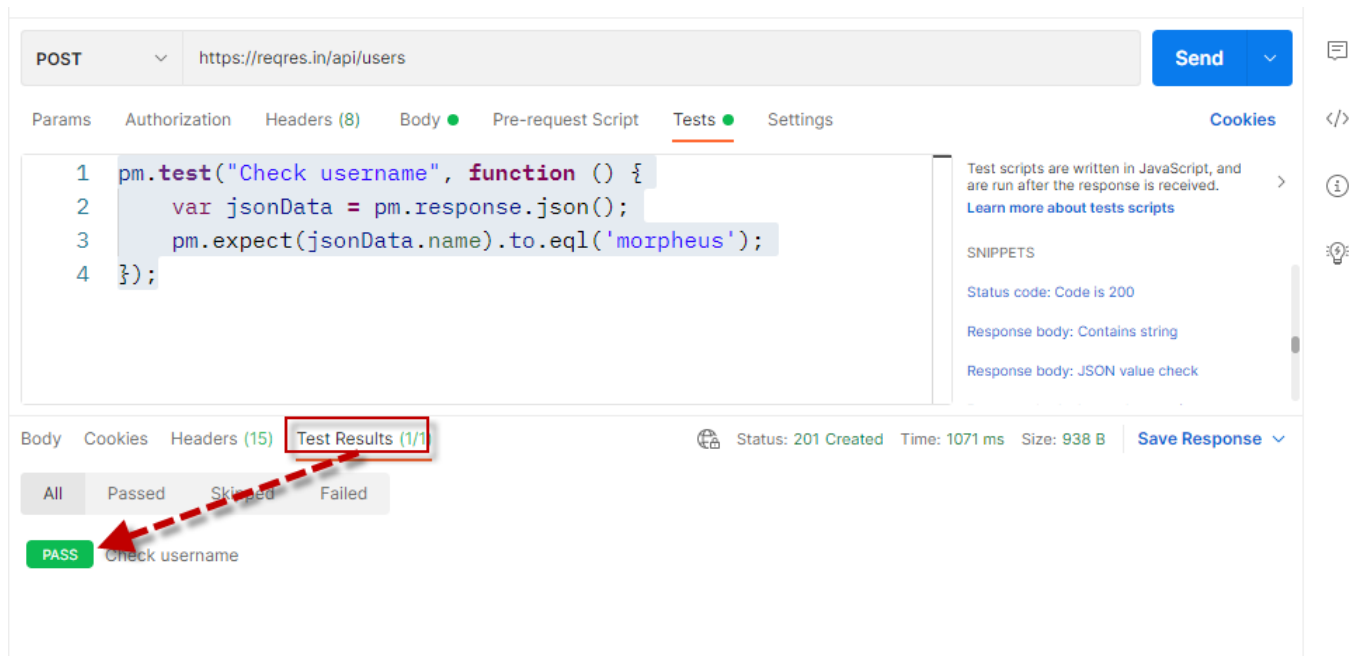


Code:

```
pm.test("Check username", function () {  
    var jsonData = pm.response.json();  
    pm.expect(jsonData.name).to.eql('morpheus');  
});
```

- 1- Parse Response trả về và lưu vào biến "jsonData".
- 2- Test xem user_name có bằng 'morpheus' không.

Bước 3: Sau khi viết xong Test thì run Request và check xem phần Test của mình có đúng không.



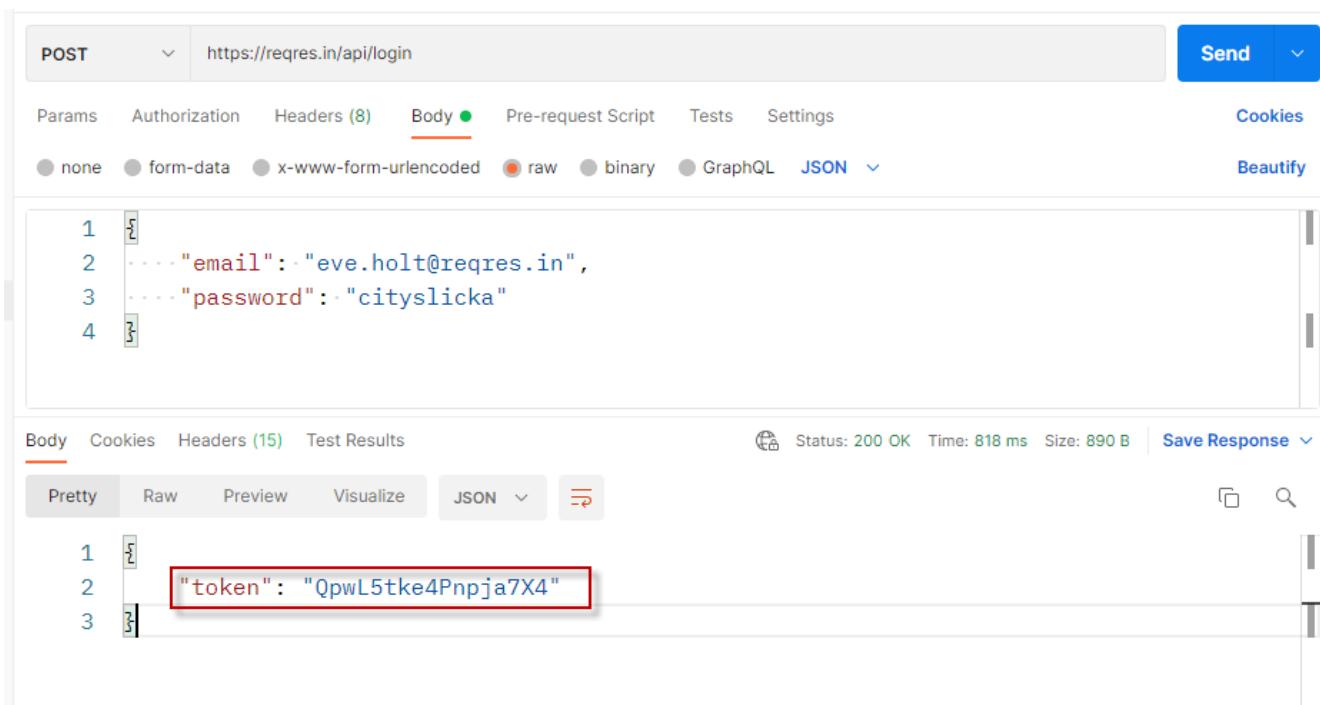
Bài tập 2:

Tạo 1 API login, sau khi login lưu lại giá trị của token để làm data cho những API tiếp theo.

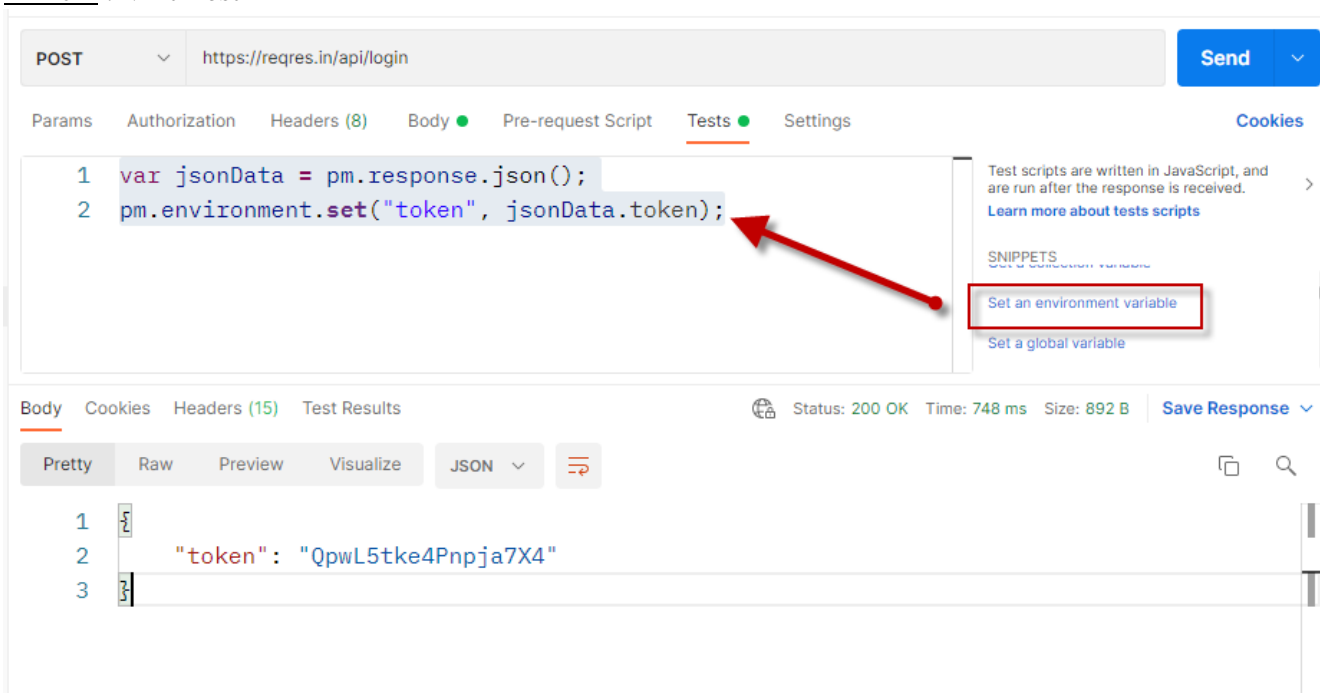
Bước 1 : Chạy thử API 1 lần để lấy được cấu trúc Response của API. (Link:

<https://reqres.in/api/login>)

Ta sẽ thấy vị trí của token là: root > token.



Bước 2 : Viết Test



Code:

```
var jsonData = pm.response.json();
pm.environment.set("token", jsonData.token);
```

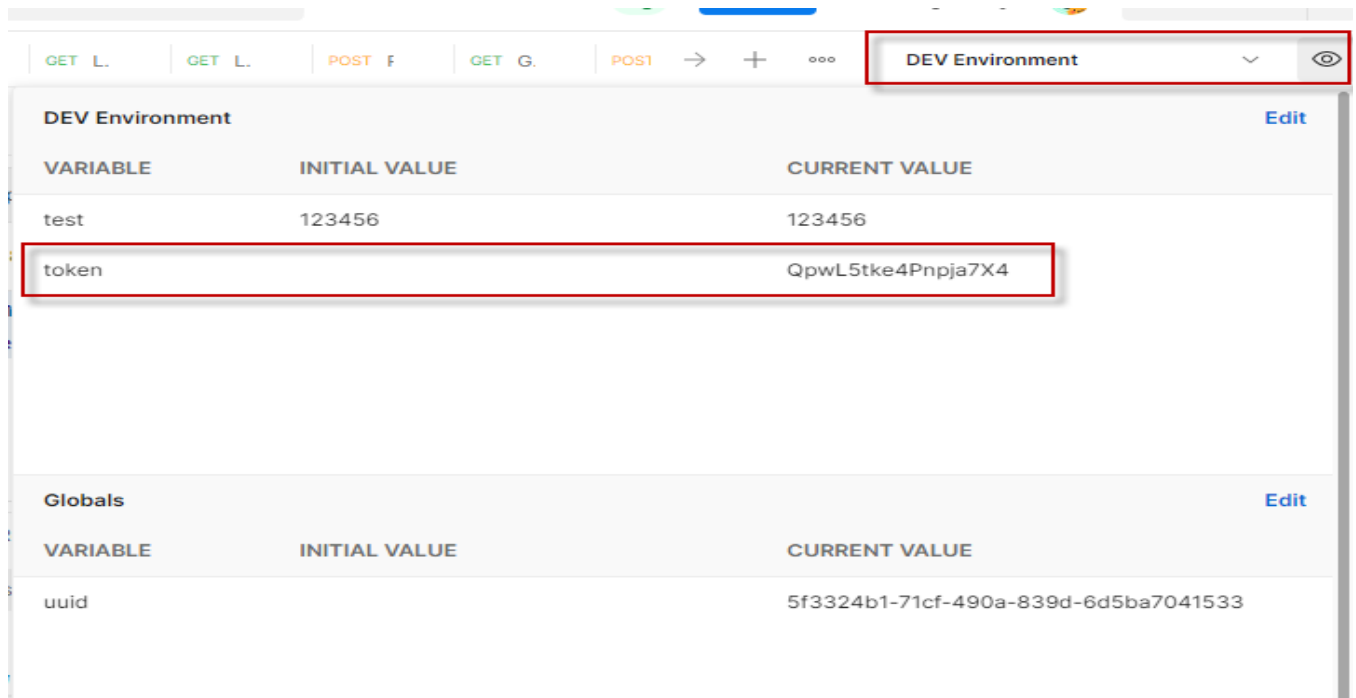
Trong đó:

variable_key: gán giá trị là "token"

variable_value: gán giá trị là jsonData.token

→ pm.environment.set("token", jsonData.token);

Sau đó, ta chỉ run lại rồi kiểm tra trong Environment thôi.



Như vậy thay vì gán biến môi trường theo kiểu thông thường, ta có thể gán biến theo cách như trên.

Bài tập 3:

Tạo 1 API hiển thị danh sách user (User List), sau đó SET tham số userId của collections giá trị Id bất kỳ từ request vừa thực thi.

Link API: <https://reqres.in/api/users?page=2>

Bước 1: Viết Test:

```
7 pm.expect(pm.response.responseTime).to.be.below(1000);
8 });
9
10 pm.test("Status code is 200", function () {
11   pm.response.to.have.status(200);
12 });
13 pm.collectionVariables.set('userId', pm.response.json().data[2].id);
```

Test scripts are written in JavaScript, and are run after the response is received.
[Learn more about tests scripts](#)

SNIPPETS

- Get a collection variable
- Set an environment variable
- Set a global variable
- Set a collection variable

Body Cookies Headers (19) Test Results (3/3) Status: 200 OK Time: 65 ms Size: 1.93 KB Save Response

Pretty Raw Preview Visualize JSON

```
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"
```

Code:

`pm.collectionVariables.set('userId', pm.response.json().data[2].id);`

Kết quả: vào collection chứa API:

	VARIABLE	INITIAL VAL...	CURRENT V	Persist All	Reset All
<input checked="" type="checkbox"/>	url	https://reqres.in	https://reqres.in		
<input checked="" type="checkbox"/>	userId		9		

[Learn more about collection variables.](#)

Bước 2: Tạo API hiển thị 1 user theo biến 'userId':

Link API: `{{url}}/api/users/{{userId}}`

Reqres.in V1 / GET Single User Save

GET ▼ `{{url}}/api/users/{{userId}}`

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (18) Test Results Status: 200 OK Time: 847 ms Size

Pretty Raw Preview Visualize JSON ▼ ↺

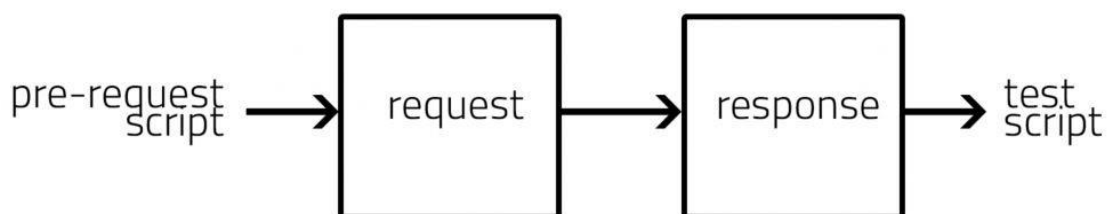
```

1  {
2    "data": {
3      "id": 9,
4      "email": "tobias.funke@reqres.in",
5      "first_name": "Tobias",
6      "last_name": "Funke",
7      "avatar": "https://reqres.in/img/faces/9-image.jpg"
8    },
9    "support": {

```

IX. Sử dụng Pre-request Script:

Đây là các bước khi gửi 1 request. Phần Pre-request sẽ là phần Postman sẽ xử lý trước khi thực hiện gửi Request, và phần test script để xử lý Response được trả về.



Vậy Pre-request có thể làm được việc gì? Nó xử lý có mỗi 1 phần thôi đó là *tạo dữ liệu (biến) để truyền vào param trong Request.*

Bài tập 4:

Viết hàm gán giá trị Id cho tham số collection **userID_new** để chạy API

`{{url}}/api/users/{{userID_new}}`

Bước 1: Viết Pre-Request Script:

Reqres.in V1 / GET Single User Copy

GET `{{url}}/api/users/{{userID_new}}` Send

Params Authorization Headers (7) Body **Pre-request Script** Tests Settings Cookies

```

1 const reqObject = {
2   url: 'https://reqres.in/api/users?page=2',
3   method: 'GET',
4   header: 'Content-Type:application/json'
5 };
6
7 pm.sendRequest(reqObject, (err, res) => {
8   pm.collectionVariables.set("userID_new", res.json().data

```

Pre-request scripts are written in JavaScript, and are run before the request is sent.
[Learn more about pre-request scripts](#)

SNIPPETS
[Get a collection variable](#)
[Set an environment variable](#)
[Set a global variable](#)
[Set a collection variable](#)

Body Cookies Headers (19) Test Results Status: 200 OK Time: 49 ms Size: 1.2 KB Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "data": {
3     "id": 9,
4     "email": "tobias.funke@reqres.in",
5     "first_name": "Tobias",
6     "last_name": "Funke"

```

Code:

```

const reqObject = {
  url: 'https://reqres.in/api/users?page=2',
  method: 'GET',
  header: 'Content-Type:application/json'
};

pm.sendRequest(reqObject, (err, res) => {
  pm.collectionVariables.set("userID_new", res.json().data[2].id);
});

```

Bước 2: kiểm tra kết quả API trả về có đúng giá trị của userID_new vừa được tạo ở bước 1:

Reqres.in V1 Share 0 0

Authorization Pre-request Script Tests **Variables**

These variables are specific to this collection and its requests.
[Learn more about collection variables.](#)

	VARIABLE	INITIAL VAL...	CURRENT V	Persist All	Reset All
<input checked="" type="checkbox"/>	url	https://reqres.in	https://reqres.in		
<input checked="" type="checkbox"/>	userId		9		
<input checked="" type="checkbox"/>	userID_new		9		
	Add a new vari...				

X. Xây dựng API Document:

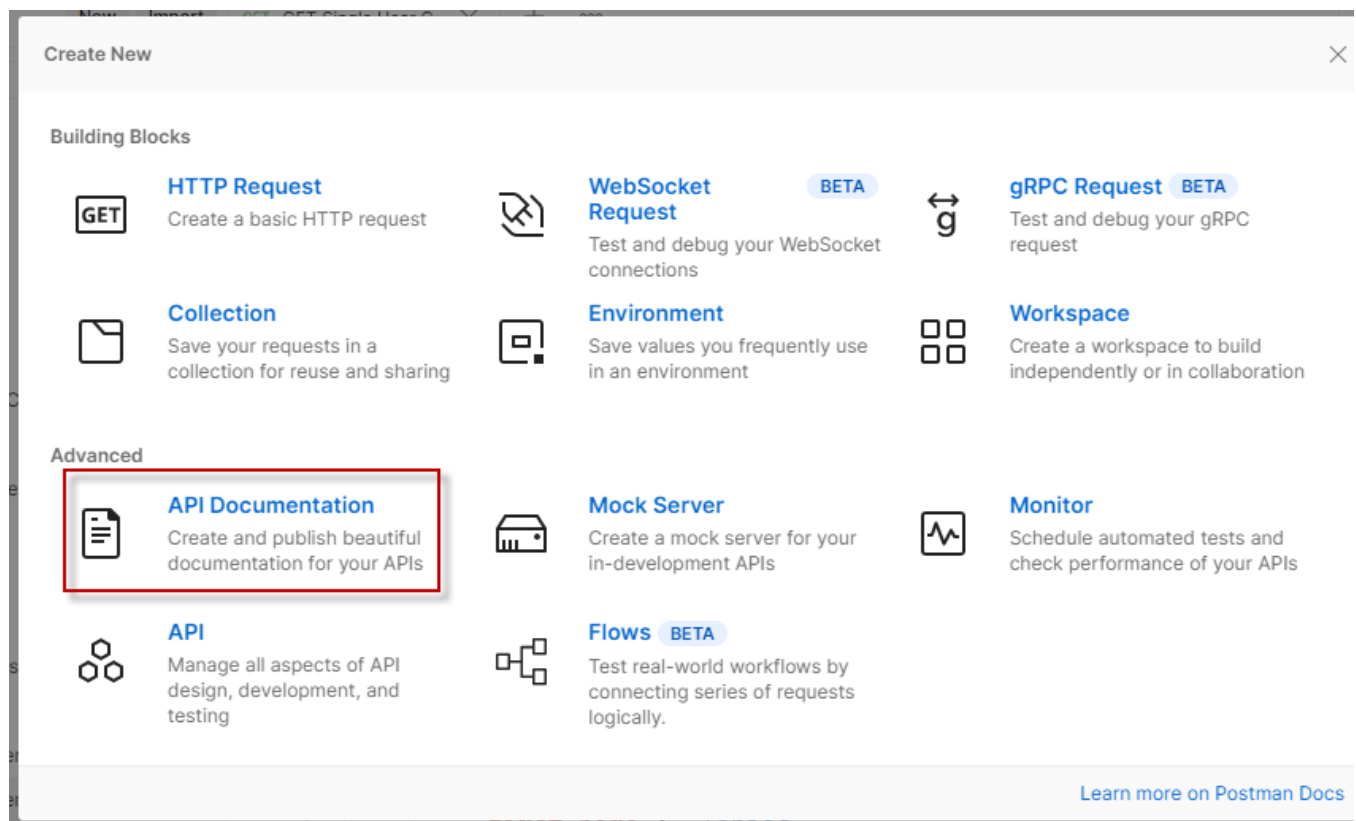
Postman ngoài việc cung cấp 1 công cụ test API còn hỗ trợ chúng ta làm API document cực kỳ chuyên nghiệp và dễ dàng. API document này có thể dùng chung cho cả team và khách hàng. Thông thường, API thường do Dev viết trên sheets nhưng đến 1 giai đoạn phát triển nào đó dev sẽ lười viết docs API hoặc API sửa nhiều, họ sẽ không nhớ để update những sửa chữa đấy.

API Document của Postman có điểm ưu việt hơn khi nó update luôn những gì mình thao tác trên khung làm việc vào API docs.

Vấn đề là 1 API Docs gồm những thành phần gì và thể hiện những phần đấy trong Postman như thế nào?

Các thành phần của 1 API Document:

- Tên của API Document
- Mô tả của API Docs
- Tên của từng API
- Mô tả của cả API: Mục đích của API, note lại các mục cần lưu ý
- Params + mô tả của params
- Sample Request
- Sample Response



Tên của từng API & Mô tả của cả API

Tên của API chính là tên chúng ta đặt cho từng Request

1. Select a collection to document 2. Write documentation 3. Next steps

Create a new collection

Select an existing collection

Create a new collection by adding individual requests. Use the "Description" column to document what each endpoint URL does. To add a request body or an example response body, click the (...) icon.

Method	Request URL	Description	...
GET	https://www.demo	GET request - user list	X
GET	URL	Description	

Tên của API Document

Chính là tên của Collection. Mô tả của API Docs

- ✓ Select a collection to document 2. Write documentation 3. Next steps

Name your collection

Demo API Document

Describe your collection

Introduction

What does your API do?

Overview

Things that the developers should know about

Authentication

What is the preferred way of using the API?

Error Codes

Give a brief description of what your collection can do and some key details such as what the collection provides and who it's for.

You can use [markdown](#) for adding headings, lists, code snippets etc. in your description.

Back

Save

✓ Select a collection to document

✓ Write documentation

3. Next steps

×

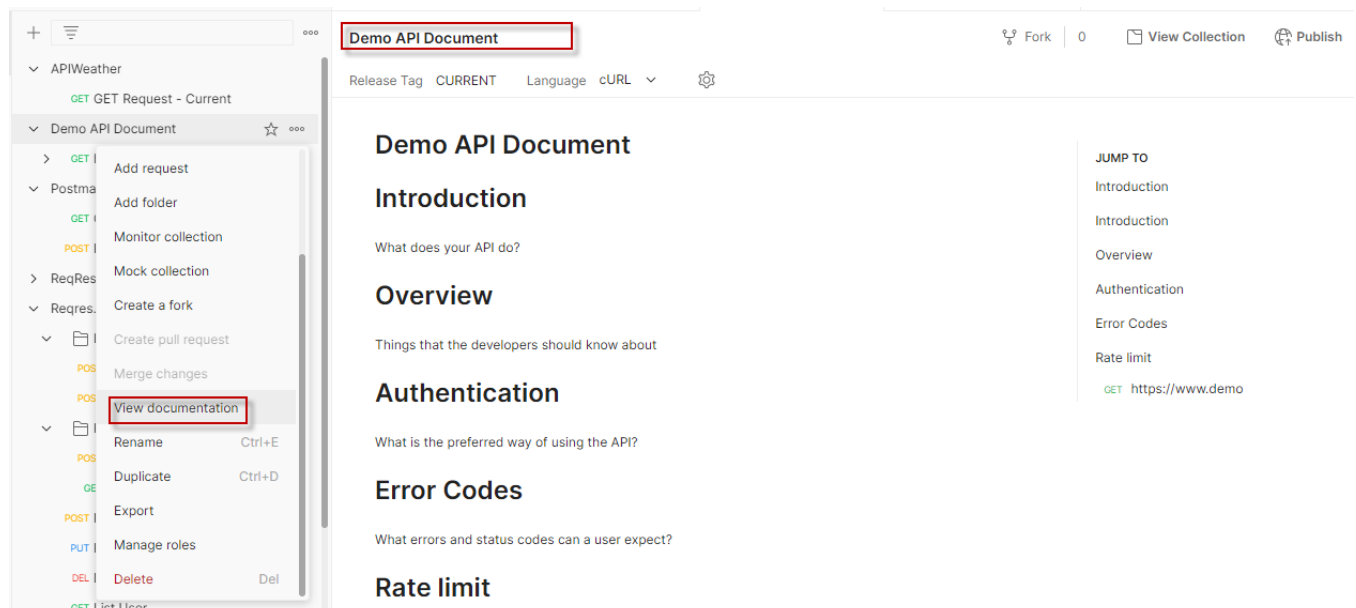
The documentation is private, so only people you share the collection with can view it.

NEXT STEPS

- Add more details to your documentation**
 For easy onboarding, you can also describe request params, body, and even collection folders. [Learn how](#)
- Share with your team**
 Share the documentation to a team workspace so your teammates can view and collaborate on it. [Learn how](#)
- Share with the world**
 Publish the documentation for people outside your team on a custom domain with your own branding. [Learn how](#)
- Add sample responses**
 Make your collection easier to consume by documenting some example responses. [Learn how to create examples](#)

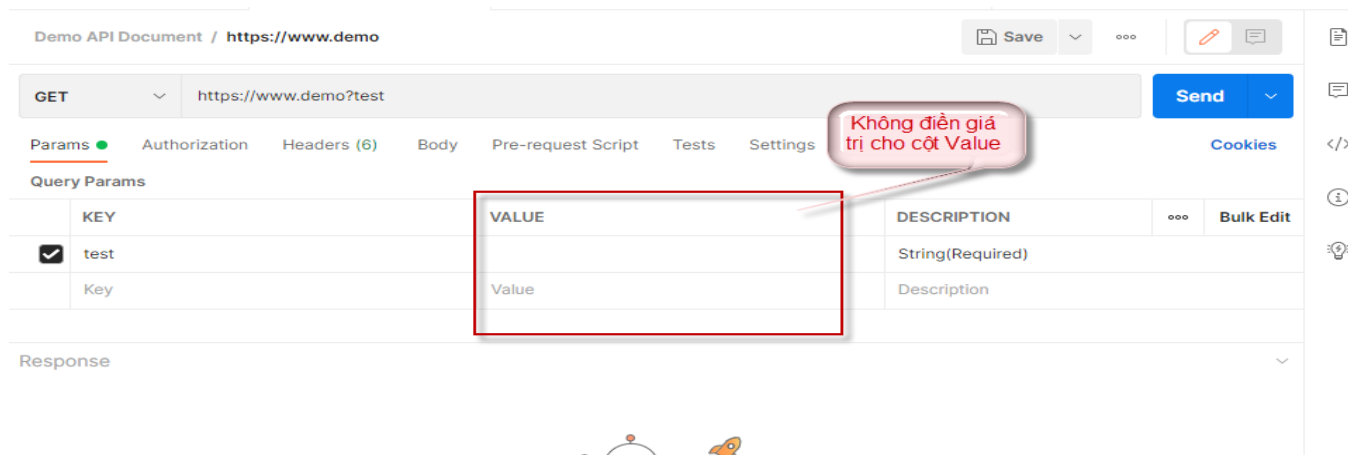
Close

View API Document:



Params + mô tả của params:

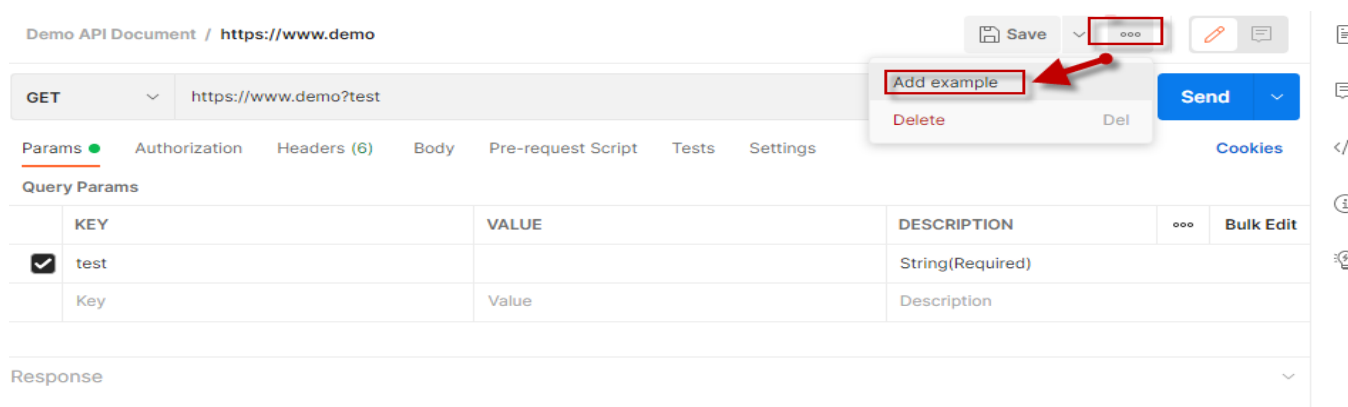
Các bạn điền vào phần Param hoặc Body giống như điền API vẫn điền trước đây nhưng khi viết API Docs, lưu ý: KHÔNG điền phần Value.



Sample Request & Sample Response

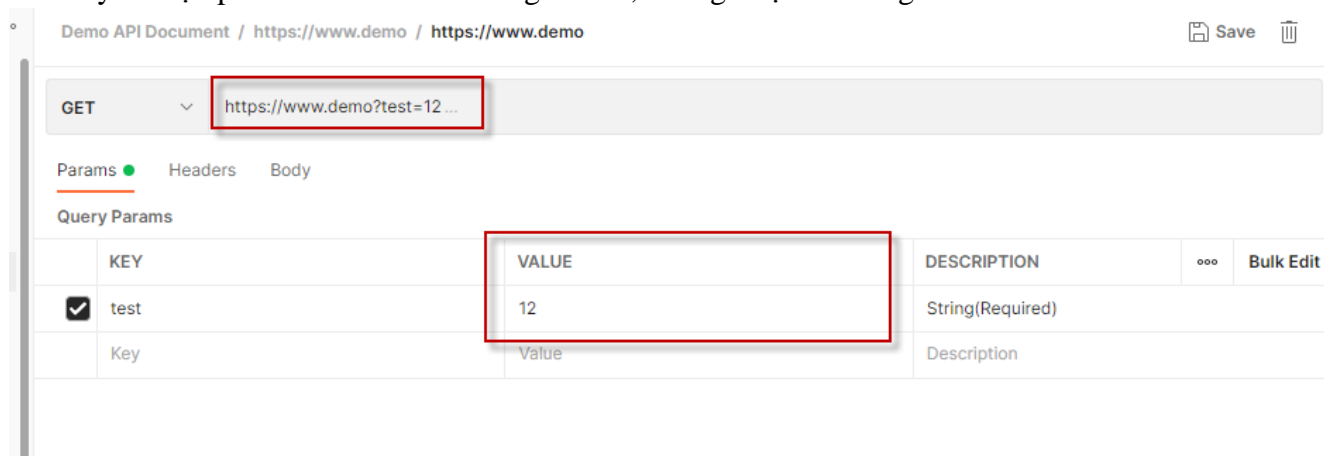
Đây là tính năng Example mà Postman cung cấp.

Add 1 Example rồi điền thông tin giống như 1 API thật sự



Sample Request

Lúc này thì bạn phải điền Value cho từng Param, không được bỏ trống.



Bài tập 5:

Tạo 1 API Document như đã hướng dẫn.