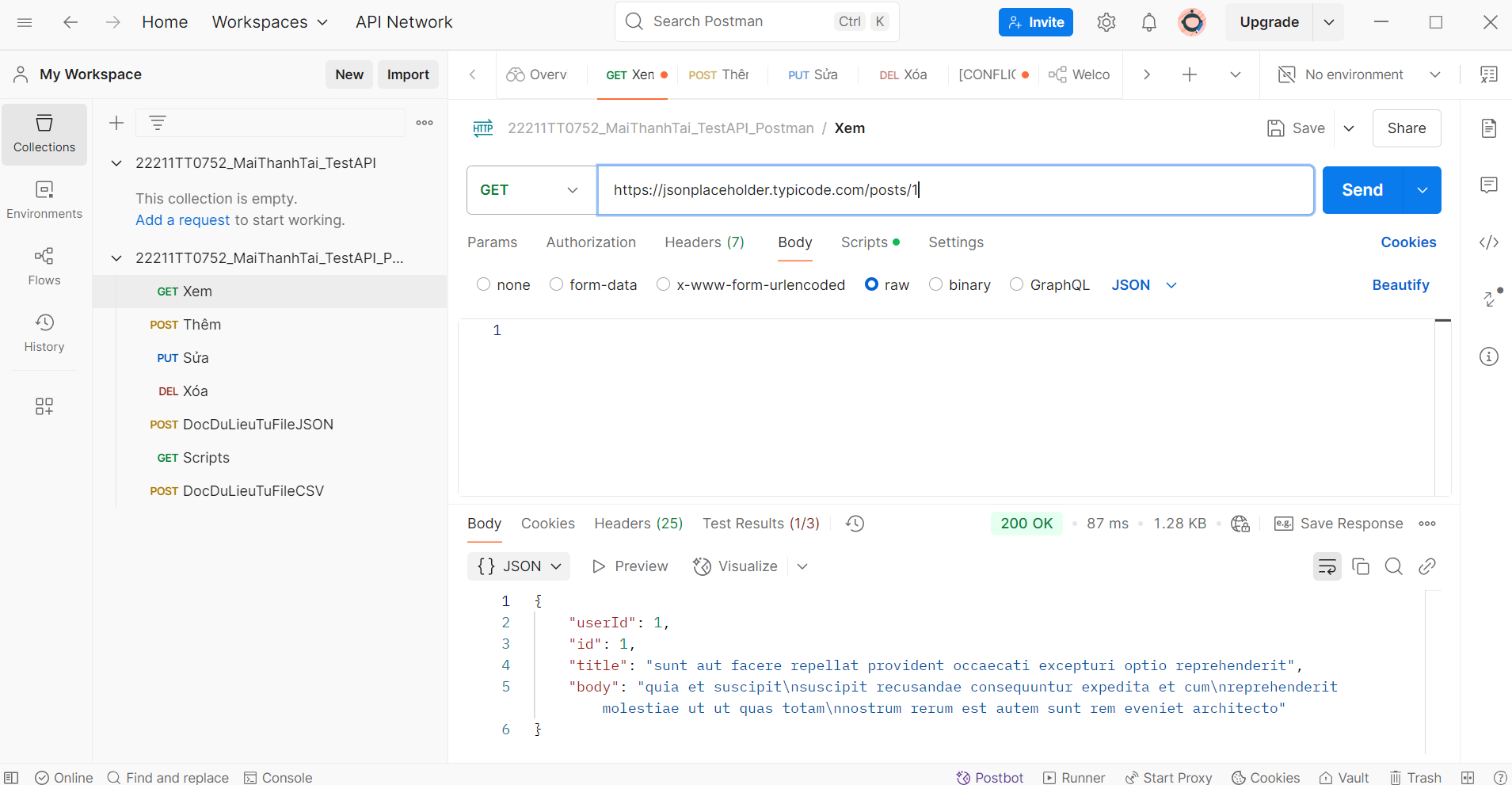
**1.Xem danh sách**

1.1Xem tại phần tử số 1

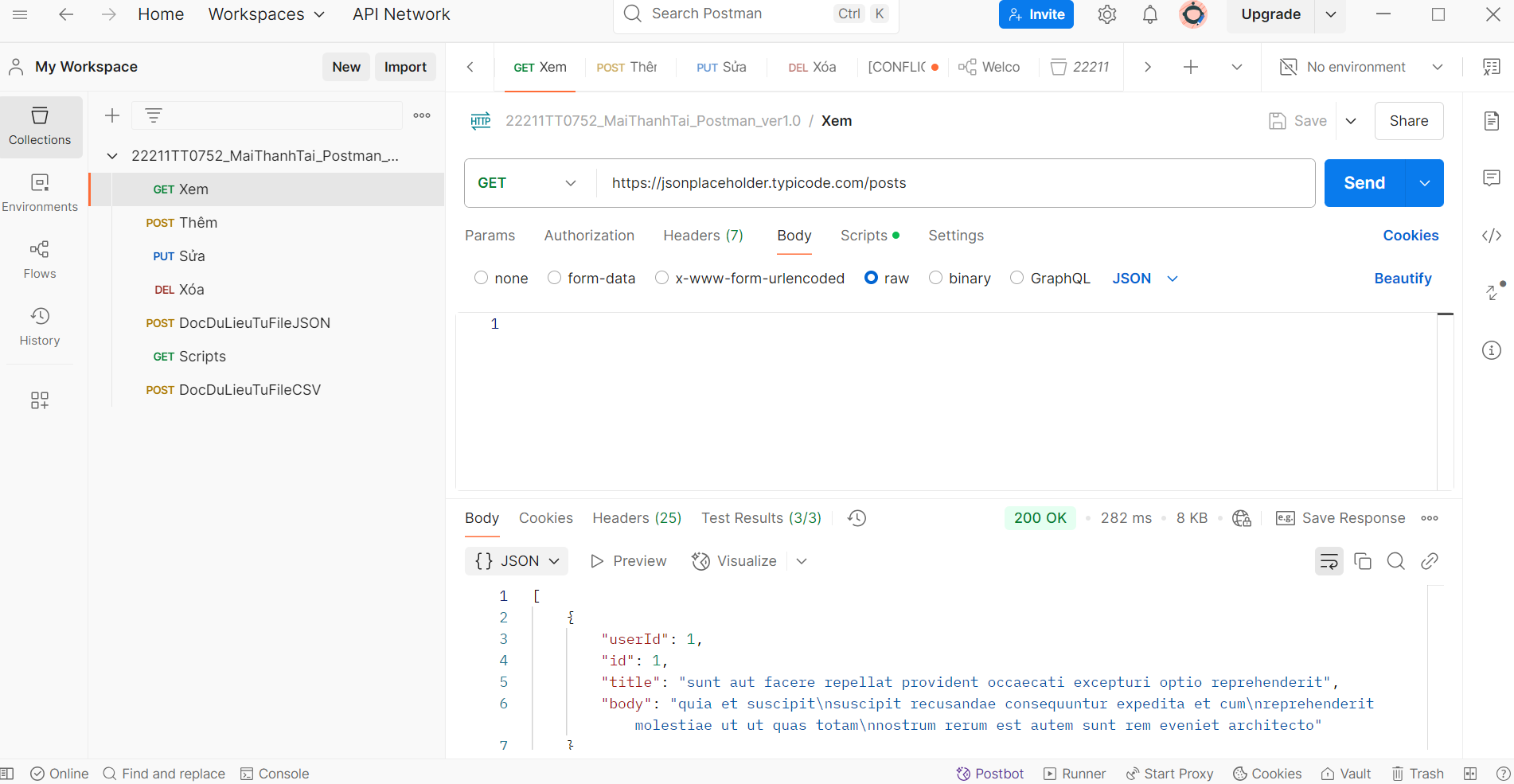


 **Phương thức:** GET (dùng để lấy dữ liệu từ server).

 **URL:** https://jsonplaceholder.typicode.com/posts/1 (API giả lập).

 **Mục đích:** Lấy thông tin của bài viết có id = 1.

1.2.Xem tất cả

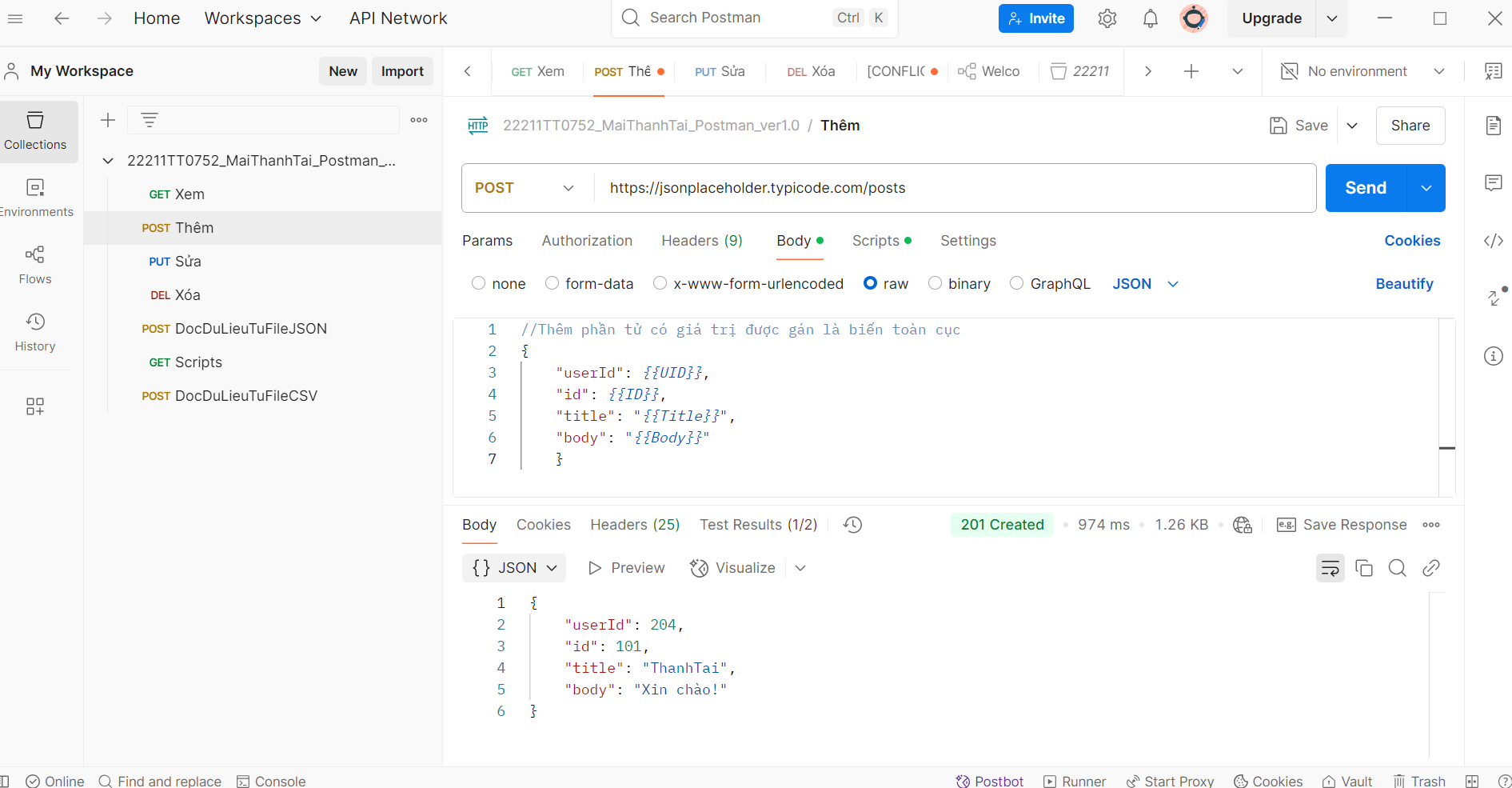


 **Phương thức:** GET (Lấy dữ liệu từ server).

 **URL:** https://jsonplaceholder.typicode.com/posts (API giả lập).

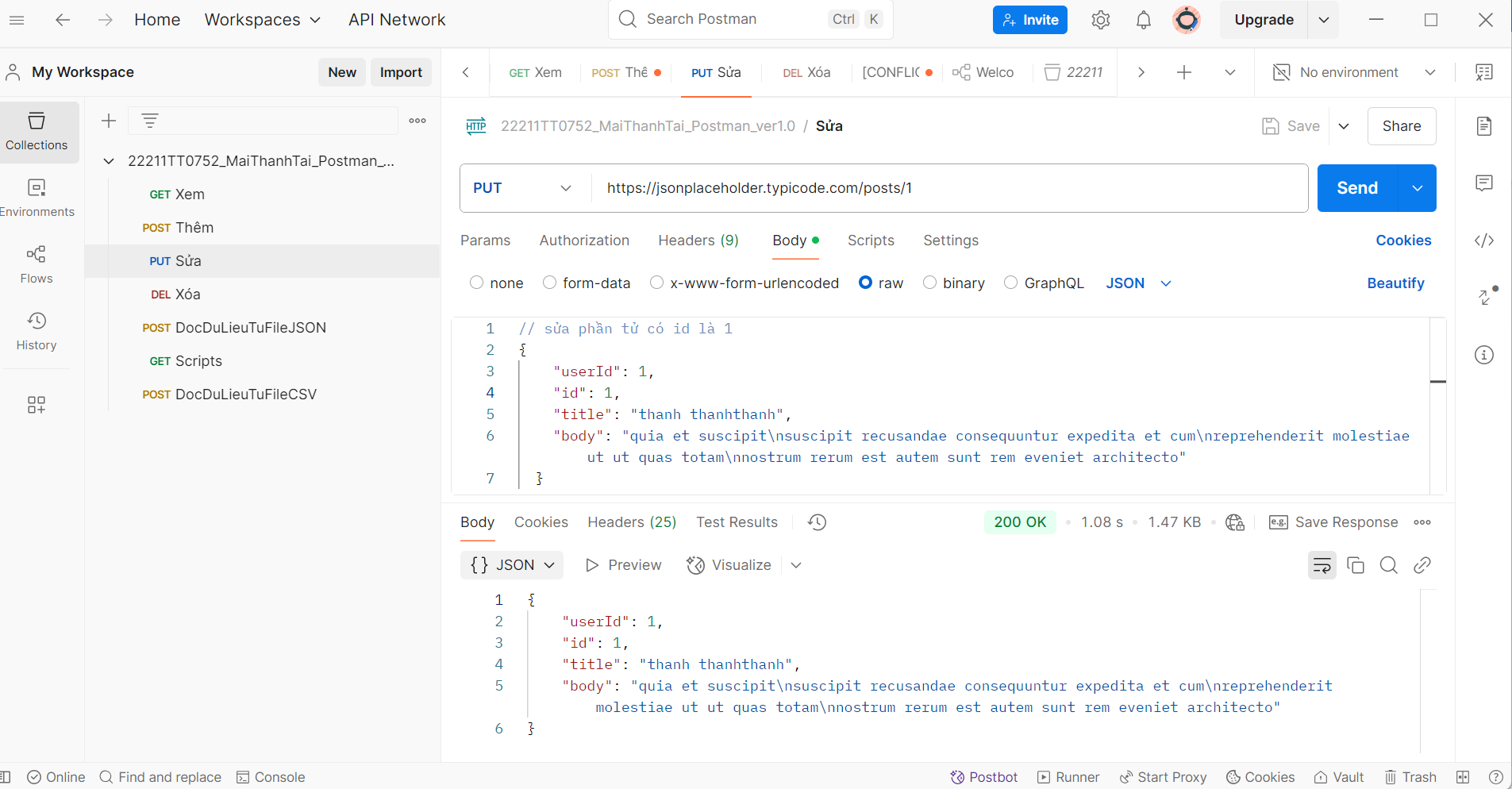
 **Mục đích:** Lấy toàn bộ danh sách các bài viết.

**2. Thêm**



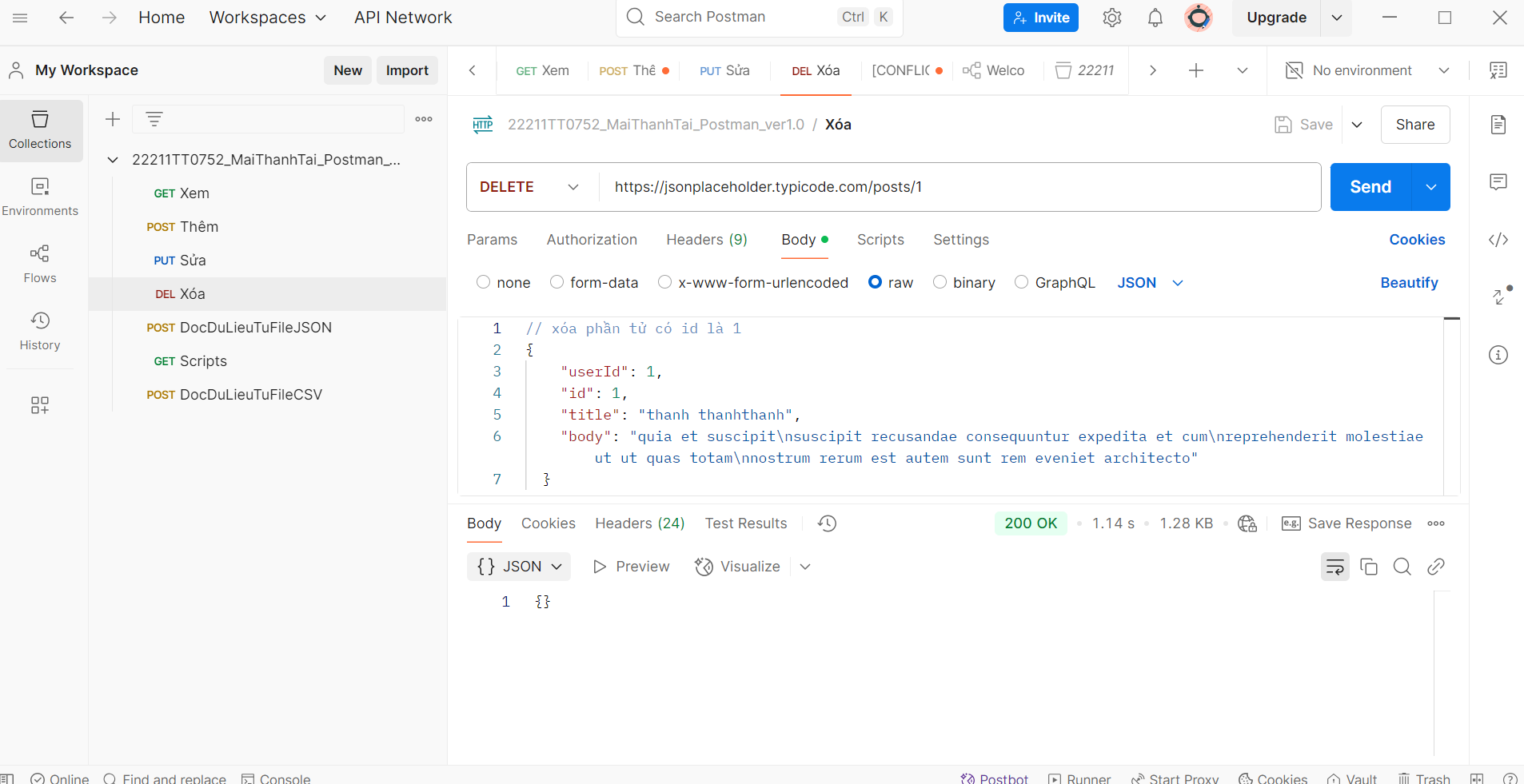
Ảnh chụp màn hình hiển thị ứng dụng Postman, một công cụ phổ biến được sử dụng để phát triển và kiểm tra API. Ảnh chụp màn hình cho thấy yêu cầu POST đang được thực hiện đến điểm cuối https://jsonplaceholder.typicode.com/posts. Yêu cầu bao gồm các tham số JSON với userId, id, title và body. Phản hồi từ máy chủ là 201 Created, cho biết yêu cầu đã thành công và tài nguyên mới đã được tạo. Phản hồi cũng bao gồm dữ liệu JSON đã tạo.

**3.Sửa phần tử có id là 1**



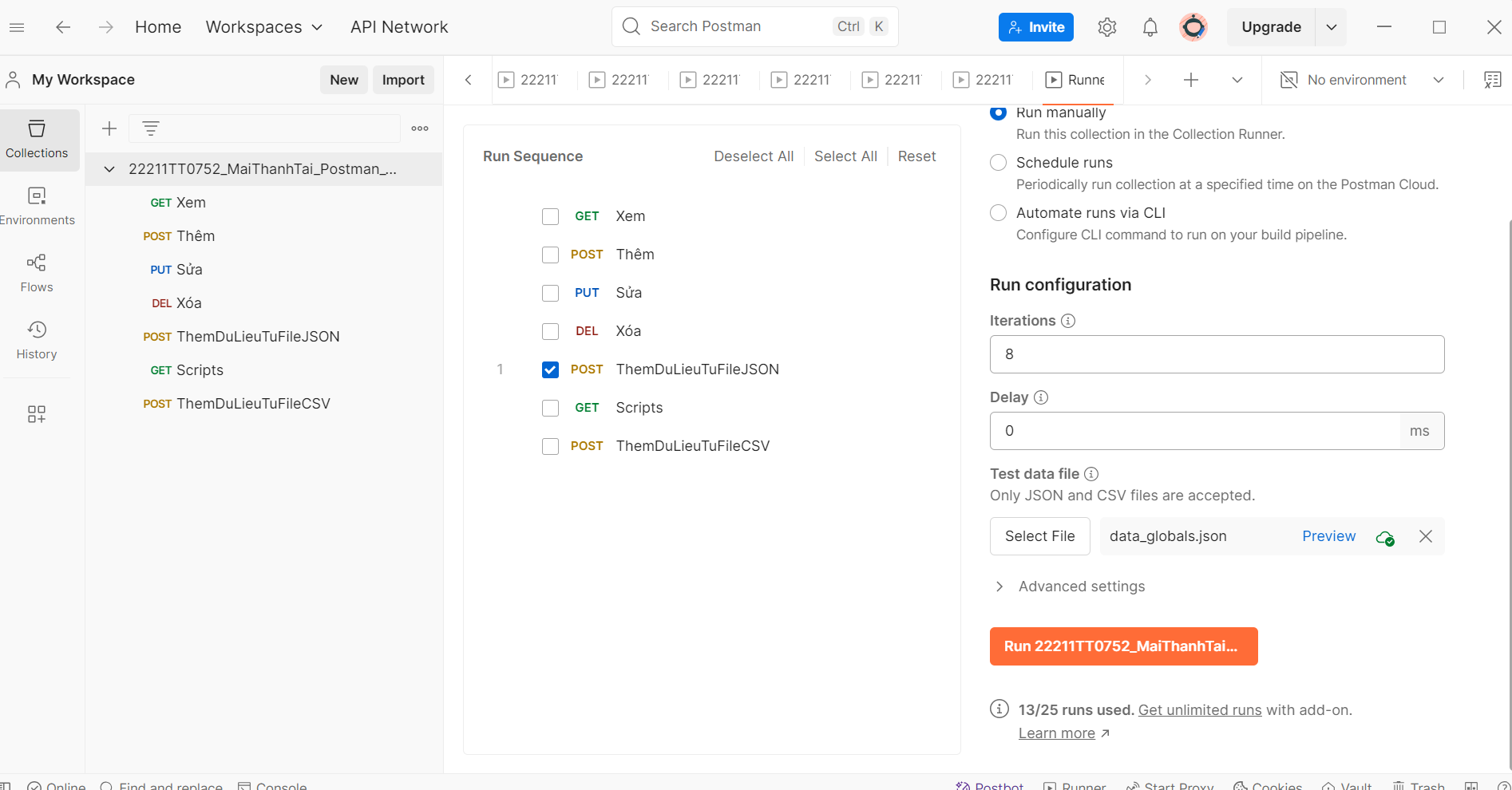
Ảnh chụp màn hình hiển thị ứng dụng Postman, một công cụ phổ biến được sử dụng để phát triển và kiểm tra API. Ảnh chụp màn hình cho thấy yêu cầu PUT đang được thực hiện đến điểm cuối https://jsonplaceholder.typicode.com/posts/1. Yêu cầu bao gồm các tham số JSON với userId, id, title và body. Phản hồi từ máy chủ là 200 OK, cho biết yêu cầu đã thành công và tài nguyên đã được cập nhật. Phản hồi cũng bao gồm dữ liệu JSON đã cập nhật.

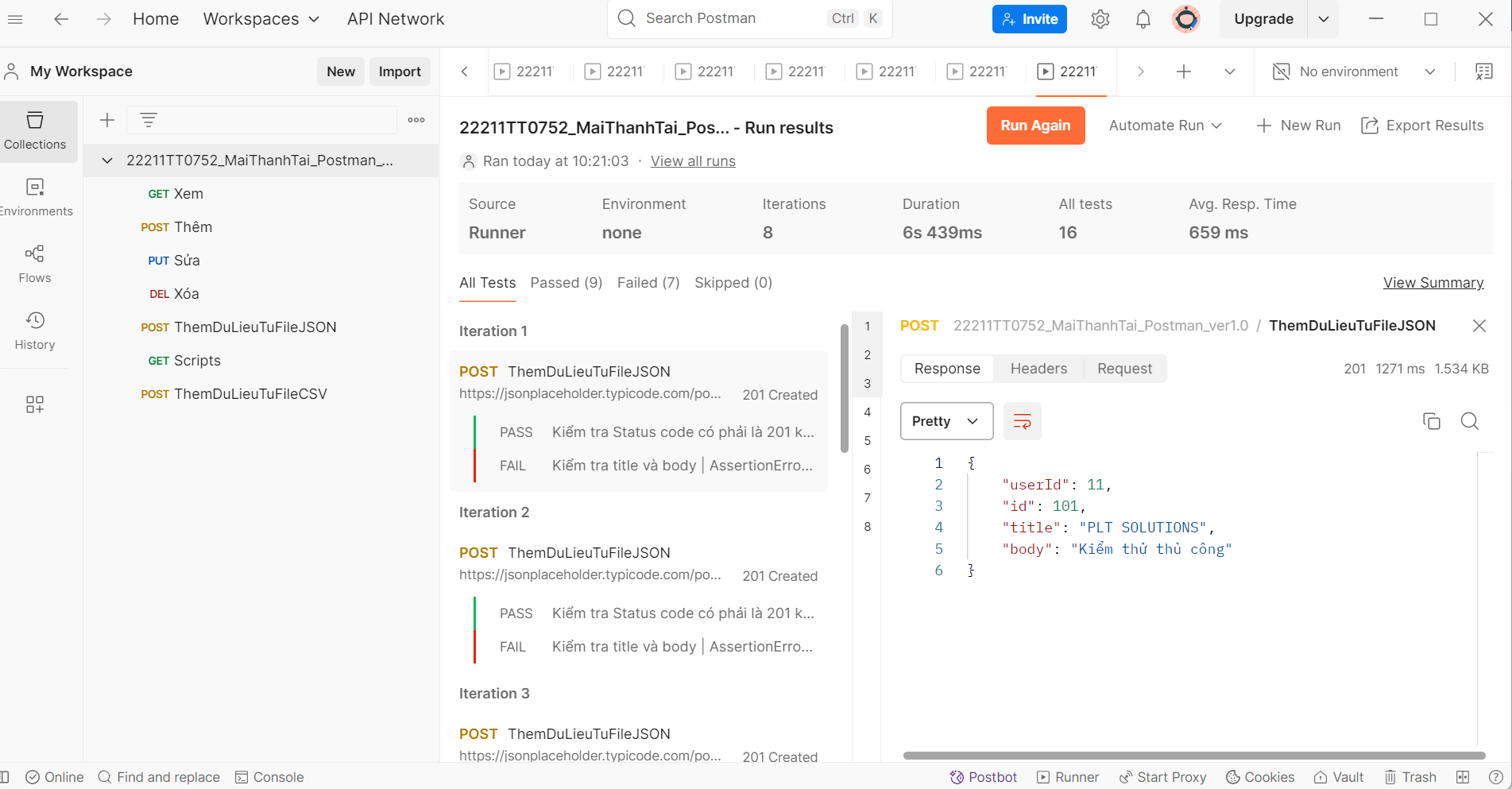
**4.Xóa phần tử có id là 1**

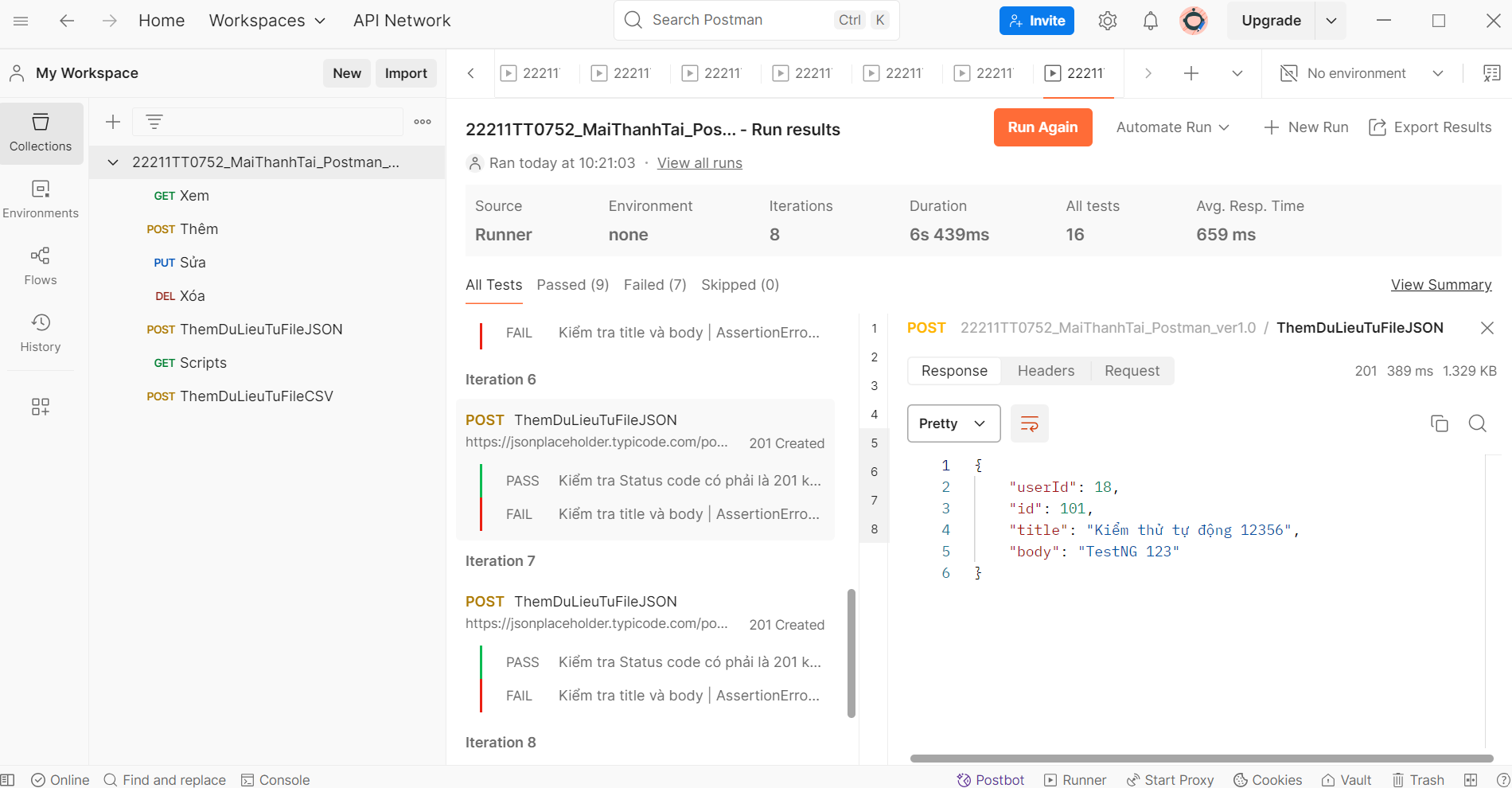


Ảnh chụp màn hình hiển thị ứng dụng Postman, một công cụ phổ biến được sử dụng để phát triển và kiểm tra API. Ảnh chụp màn hình cho thấy yêu cầu DELETE đang được thực hiện đến điểm cuối https://jsonplaceholder.typicode.com/posts/1. Yêu cầu bao gồm các tham số JSON với userId, id, title và body. Phản hồi từ máy chủ là 200 OK, cho biết yêu cầu đã thành công và tài nguyên đã bị xóa. Phản hồi cũng bao gồm dữ liệu JSON đã xóa.

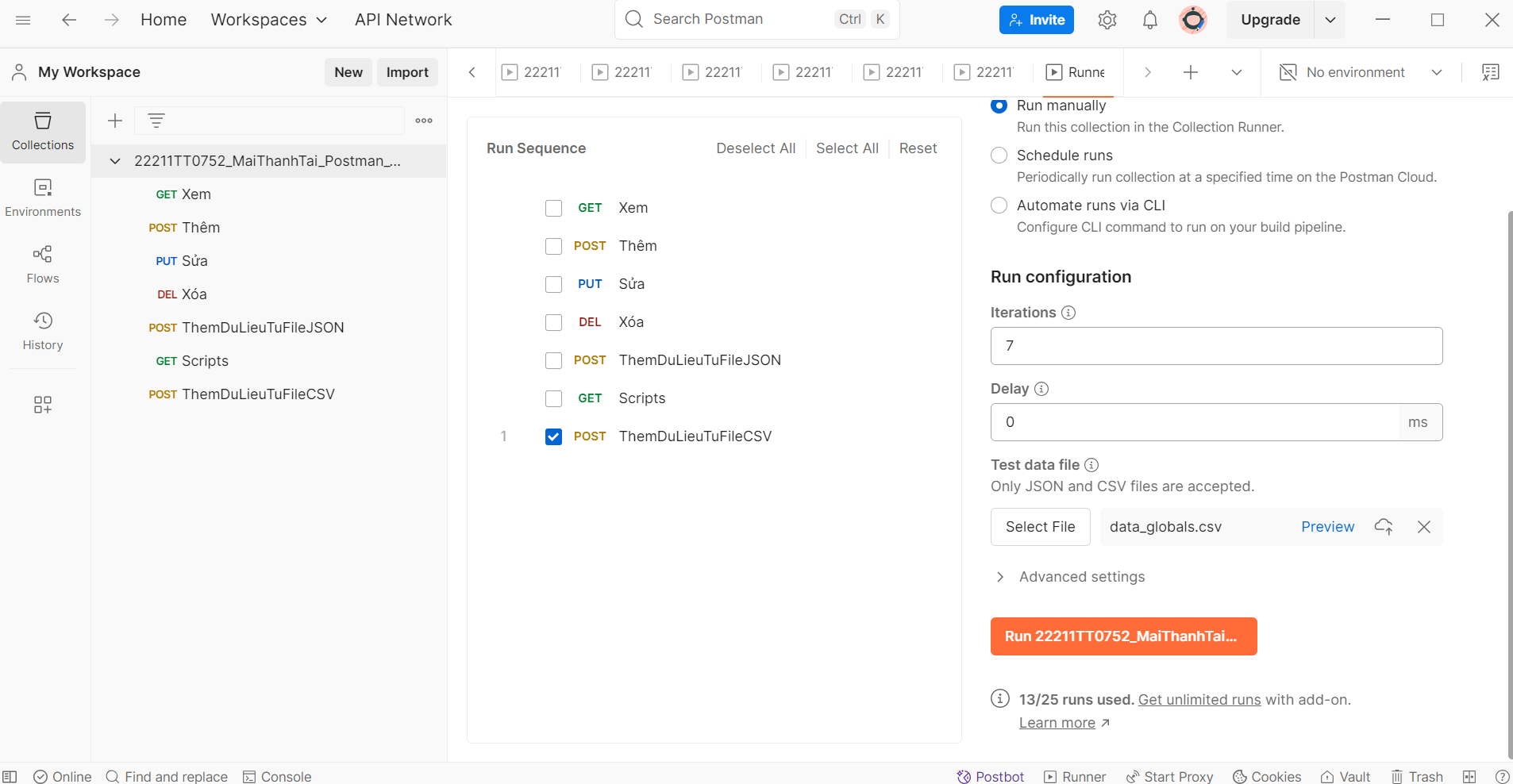
**5.Thêm dữ liệu từ file json**

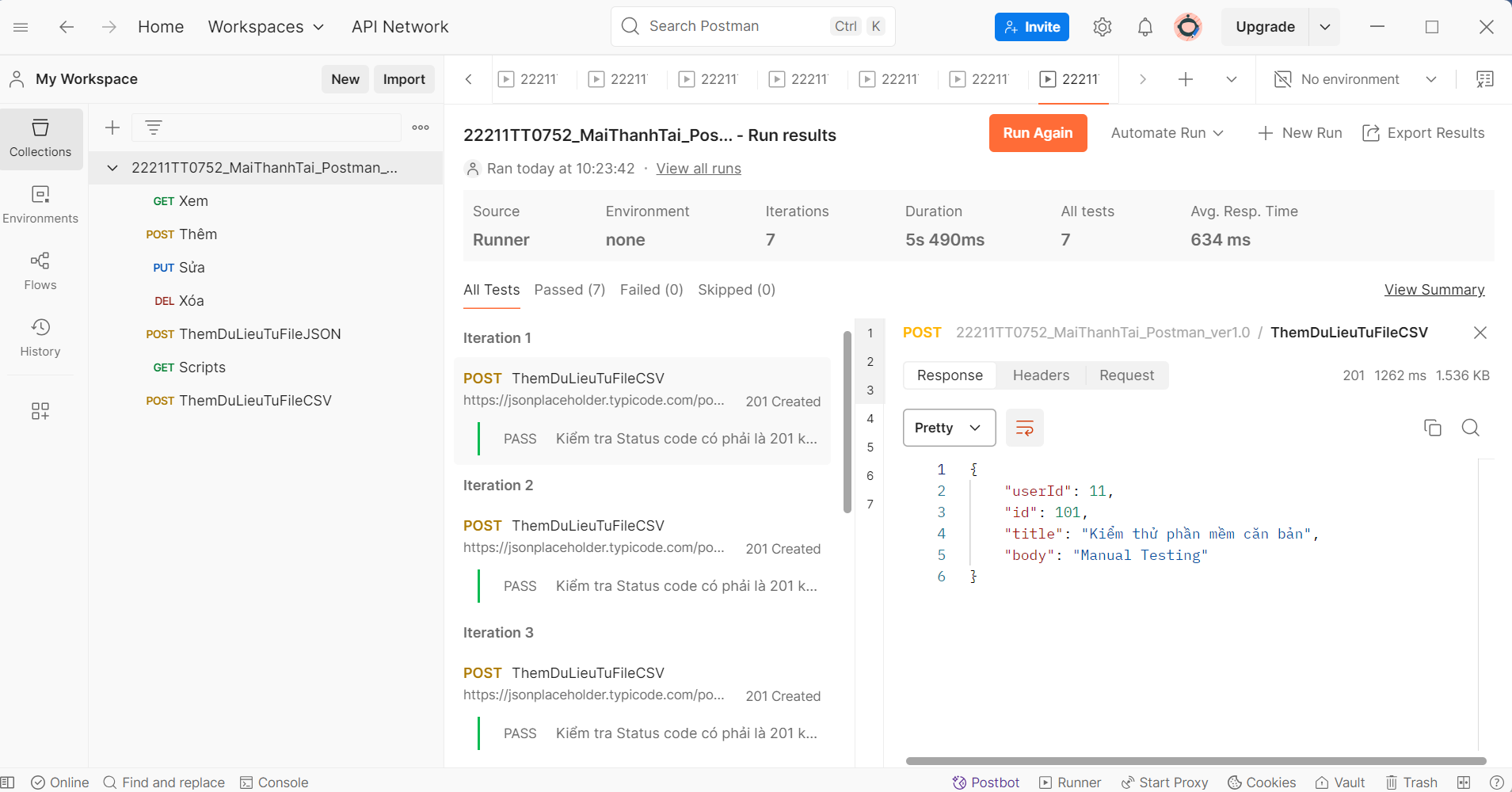




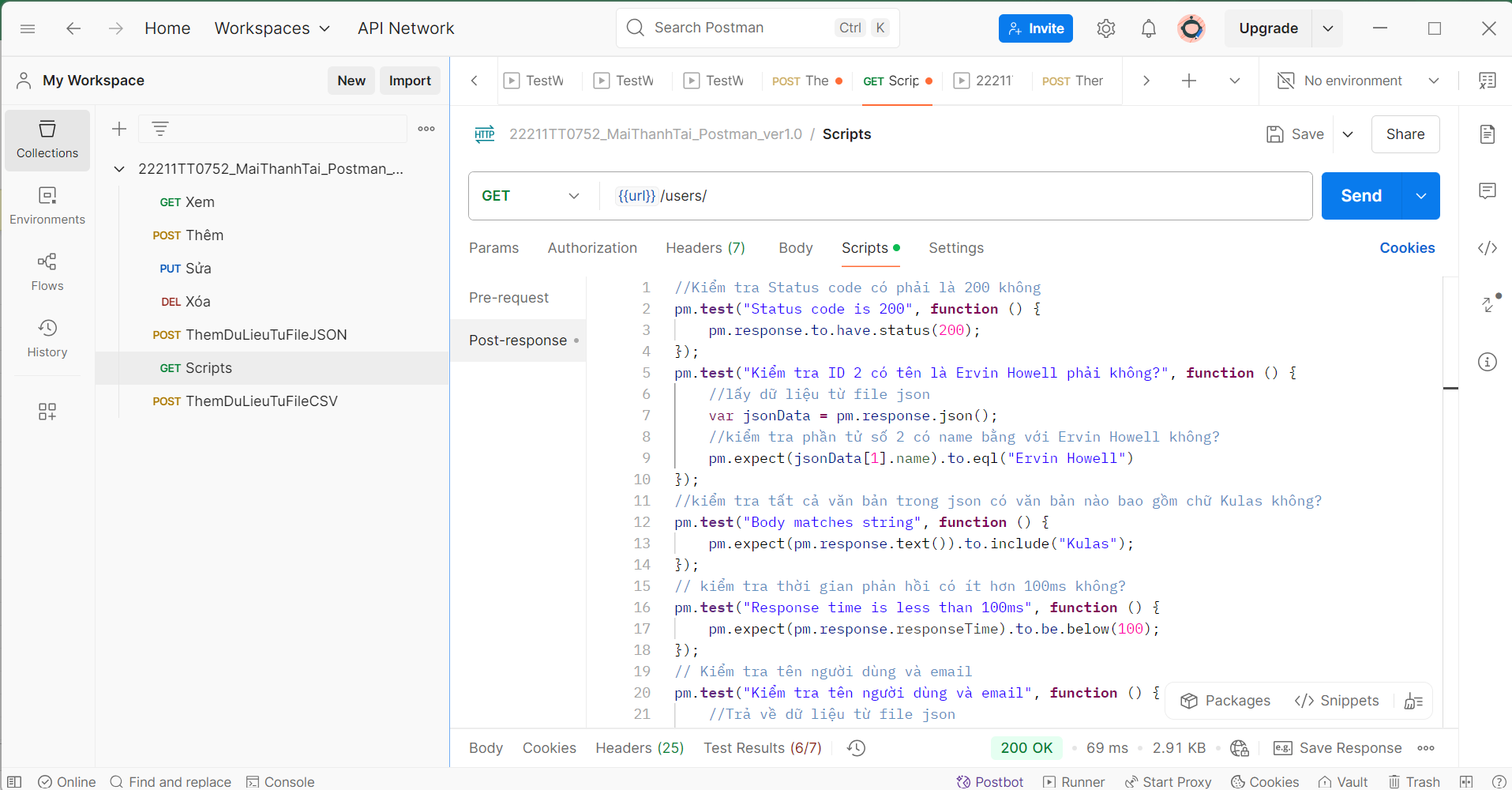


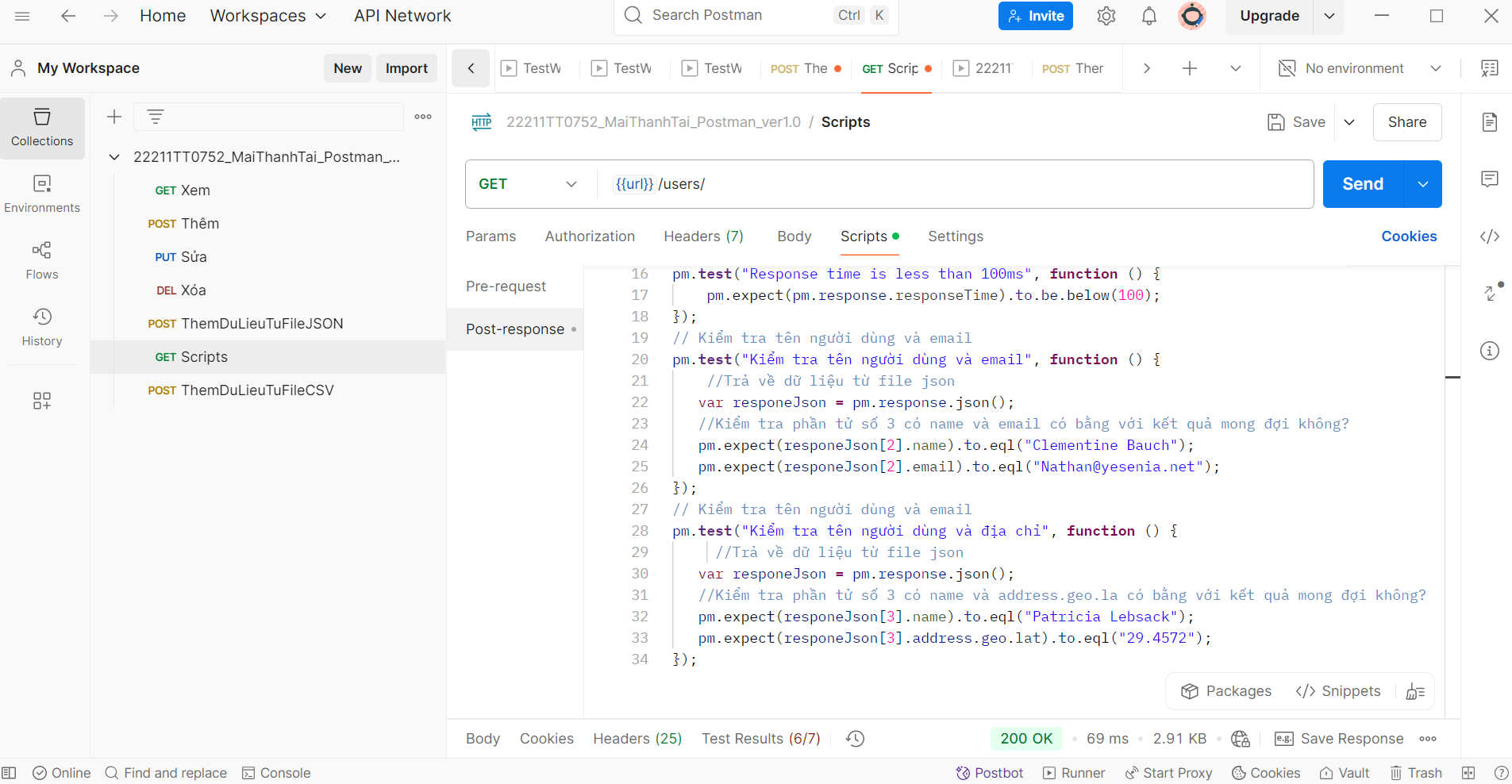
**6.Thêm dữ liệu từ file csv**

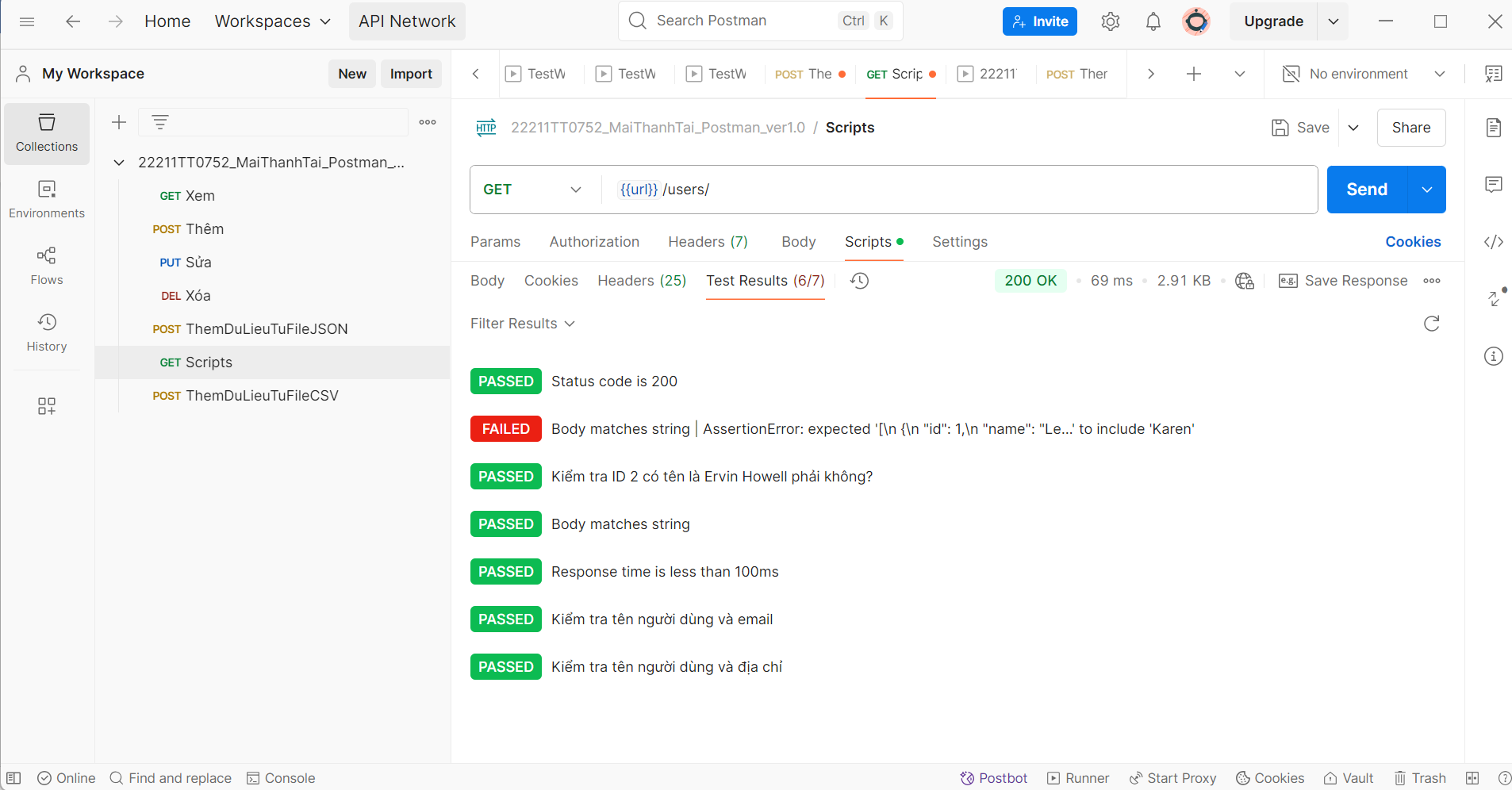




**7.Scripts**







Description:

 pm.test() là hàm của Postman dùng để viết các kiểm thử.

 "Status code is 200" là tên kiểm thử.

 pm.response.to.have.status(200); kiểm tra xem phản hồi của API có trả về mã trạng thái HTTP là 200 (OK) hay không.

* pm.response.json() chuyển đổi phản hồi từ dạng JSON thành đối tượng JavaScript (jsonData).
* jsonData[1] tương ứng với phần tử thứ hai trong mảng (vì mảng bắt đầu từ 0).
* pm.expect(...).to.eql("Ervin Howell"); kiểm tra xem **tên** của phần tử thứ hai có phải là "Ervin Howell" hay không.
* pm.response.text() chuyển đổi phản hồi thành **chuỗi văn bản** (string).
* pm.expect(...).to.include("Kulas"); kiểm tra xem chuỗi "Kulas" có tồn tại trong phản hồi hay không.
*  pm.response.responseTime trả về thời gian phản hồi của API tính bằng **ms**.
*  .to.be.below(100); kiểm tra xem thời gian phản hồi có nhỏ hơn 100ms không.
* responeJson[2] lấy ra phần tử thứ ba của mảng.
* Kiểm tra xem:
  + name có bằng "Clementine Bauch" không.
  + email có bằng "Nathan@yesenia.net" không.

 responeJson[3] lấy ra phần tử thứ tư trong mảng (index = 3).

 Kiểm tra:

* name của phần tử đó phải là "Patricia Lebsack".
* Truy cập đến **vị trí địa lý** (geo.lat) trong địa chỉ (address) và kiểm tra xem nó có bằng "29.4572" không.