**Bắt đầu**

Để bắt đầu với việc sử dụng json-server, hãy cài đặt gói sử dụng Node Package Manager (NPM).

|  |  |
| --- | --- |
| 1 | npm install -g json-server |

Tạo một tập tin JSON giả với một số dữ liệu theo yêu cầu của bạn. Ví dụ, tôi cần một số dữ liệu JSON với thông tin người dùng như id, name, location, vân vân. Vì vậy, tôi sẽ tạo ra một tập tin gọi là info.json với các thông tin JSON sau đây:

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11 | {      "users": [{          "id": 1,          "name": "roy",          "location": "india"      }, {          "id": 2,          "name": "sam",          "location": "wales"      }]  } |

Từ Terminal, chạy json server với info.json như là nguồn dữ liệu và bạn cần phải có REST API chạy tại [http://localhost:3000](http://localhost:3000/).

|  |  |
| --- | --- |
| 1 | json-server info.json |

**Kiểm tra các Endpoint của REST API**

Vì máy chủ giả lập REST API của chúng ta đang chạy, chúng ta hãy xem làm thế nào để truy cập vào REST API sử dụng một client. Tôi đang sử dụng client Postman REST để gọi API.

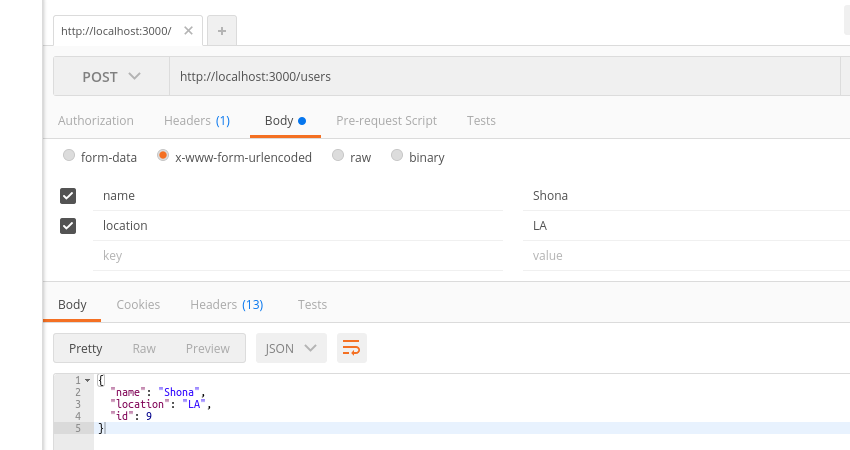
**Yêu cầu GET**

Hãy bắt đầu bằng cách tạo một yêu cầu GET đến REST URL. Bên trong tập tin JSON, chúng ta đã xác định một endpoint users, trong đó có thông tin liên quan đến những người dùng. Khi tạo yêu cầu GET đến URL <http://localhost:3000/users>, nó sẽ hiển thị dữ liệu hiện có.

|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12 | [    {      "id": 1,      "name": "roy",      "location": "india"    },    {      "id": 2,      "name": "sam",      "location": "wales"    }  ] |

**Yêu cầu POST**

Để thêm dữ liệu mới vào dữ liệu hiện có, chúng ta sẽ thực hiện một yêu cầu POST đến URL <http://localhost:3000/users>. Yêu cầu POST sẽ trông giống như dưới đây:



Hãy thử tạo một yêu cầu GET và bạn sẽ có các dữ liệu mới được thêm vào trong tập tin info.json.

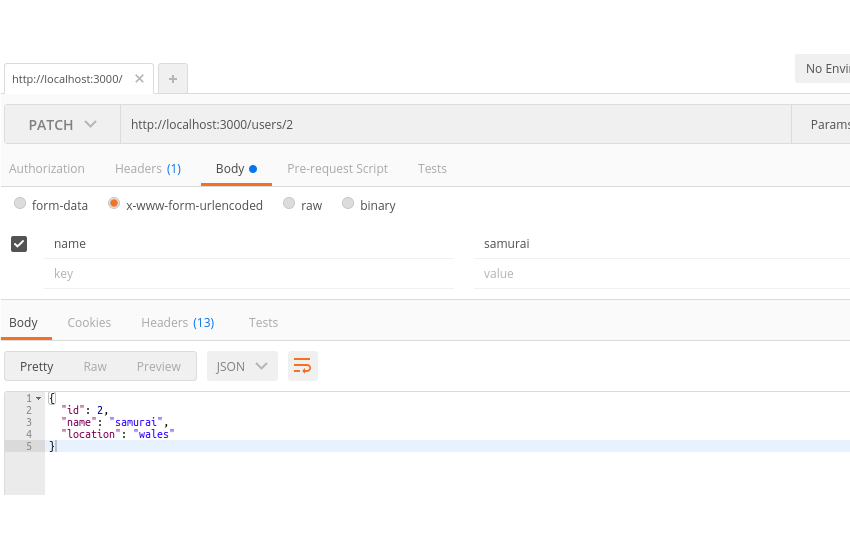
|  |  |
| --- | --- |
| 01  02  03  04  05  06  07  08  09  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | [    {      "id": 1,      "name": "roy",      "location": "india"    },    {      "id": 2,      "name": "sam",      "location": "wales"    },    {      "name": "ii",      "location": "la",      "id": 7    },    {      "name": "Shona",      "location": "LA",      "id": 8    },    {      "name": "Shona",      "location": "LA",      "id": 9    }  ] |

**Yêu cầu DELETE**

Để xóa một mục từ dữ liệu trên json-server, bạn cần phải gửi một yêu cầu DELETE đến endpoint API với Id người dùng. Ví dụ, để xóa các người dùng với Id 1, bạn cần phải gửi một yêu cầu DELETE đến endpoint [http://localhost:3000/users /1](http://localhost:3000/users/1). Một khi đã xóa, hãy thử tạo một yêu cầu GET, và người dùng với Id 1 sẽ không có ở trong JSON trả về.

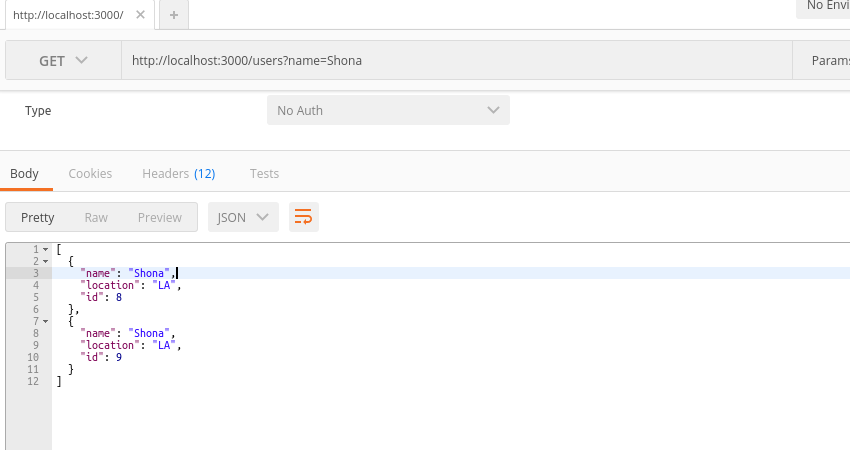
**Yêu cầu PATCH**

Để cập nhật một mục hiện có, bạn cần phải gửi một yêu cầu PATCH với các chi tiết cần để được cập nhật cho mục cụ thể đó. Ví dụ, để cập nhật các chi tiết cho người dùng với Id 2, chúng ta sẽ gửi một yêu cầu PATCH đến URL <http://localhost:3000/users/2> như hình dưới đây:



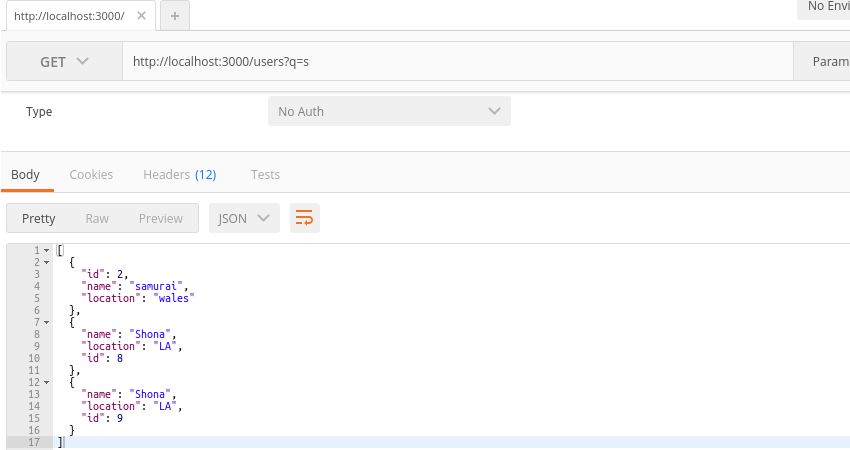
**Tìm kiếm trên REST API**

Sử dụng REST API được tạo ra bởi json-server, bạn có thể tìm kiếm trên dữ liệu để tìm dữ liệu dựa trên các tiêu chí nhất định. Ví dụ, để tìm ra người dùng với một tên cụ thể, bạn cần phải gửi một yêu cầu GET đến URL của REST API như sau:



Như đã thấy trong hình trên, gửi một yêu cầu GET đến URL <http://localhost:3000/users?name=Shona> sẽ trả về người dùng với tên Shona. Tương tự như vậy, để tìm kiếm người dùng với bất kỳ trường nào khác, bạn cần phải thêm trường đó vào chuỗi truy vấn.

Để thực hiện một tìm kiếm văn bản đầy đủ xuyên suốt endpoint của REST API, bạn cần phải thêm chuỗi tìm kiếm cùng với tham số q vào endpoint. Ví dụ, để tìm kiếm người dùng với thông tin chứa chuỗi s, yêu cầu sẽ trông giống như thế này:



**Xử lý phân trang**

Trong khi hiển thị một lưới dữ liệu phân trang, nó sẽ cần lấy một số dữ liệu dựa trên việc phân trang. Trong tình huống như vậy, json-server cung cấp các chức năng để đánh số trang dữ liệu JSON. Mặc định, số lượng dữ liệu trả về từ json-server là 10. Chúng ta có thể xác định rõ ràng giới hạn này bằng cách sử dụng tham số \_limit.

|  |  |
| --- | --- |
| 1 | http://localhost:3000/users?\_limit=5 |

Một yêu cầu GET đến URL ở trên sẽ trả về năm mẫu dữ liệu. Bây giờ, để đánh số trang dữ liệu, chúng ta cần thêm một tham số \_page vào URL. \_page xác định trang mà cần phải được lấy khi trả về dữ liệu.

|  |  |
| --- | --- |
| 1 | http://localhost:3000/users?\_limit=5&\_page=2 |

Một yêu cầu GET đến URL ở trên sẽ trả về trang thứ hai của tập dữ liệu với năm mẫu dữ liệu trên mỗi trang. Bằng cách thay đổi biến \_page, chúng ta có thể lấy các trang cần thiết.

**Xử lý phân loại**

json-server cung cấp tính năng để sắp xếp các dữ liệu trả về. Chúng ta có thể sắp xếp dữ liệu bằng cách cung cấp tên cột cần được sắp xếp và thứ tự trong đó dữ liệu cần được sắp xếp. Mặc định, dữ liệu được sắp xếp theo thứ tự tăng dần. Chúng ta có thể cung cấp tên cột trong endpoint của URL sử dụng từ khoá \_sort và xác định thứ tự bằng cách sử dụng từ khóa \_order. Dưới đây là một URL ví dụ:

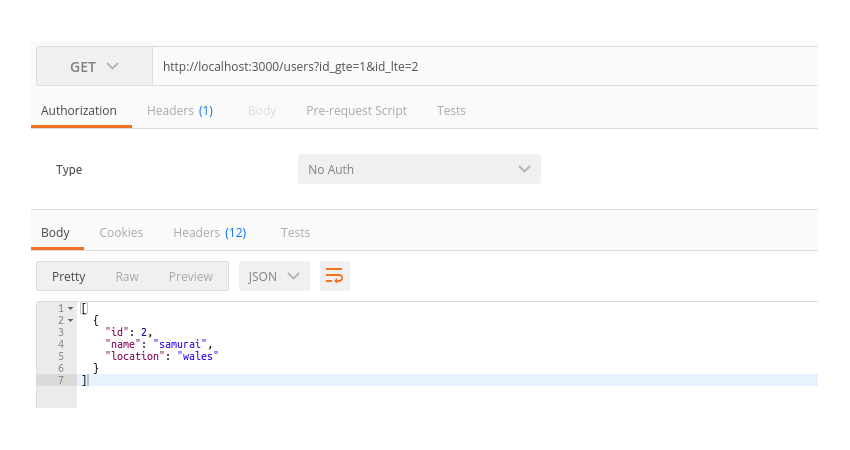
|  |  |
| --- | --- |
| 1 | http://localhost:3000/users?\_sort=id&\_order=DESC |

URL trên sẽ sắp xếp dữ liệu dựa trên cột Id, và nó sẽ được sắp xếp theo thứ tự giảm dần.

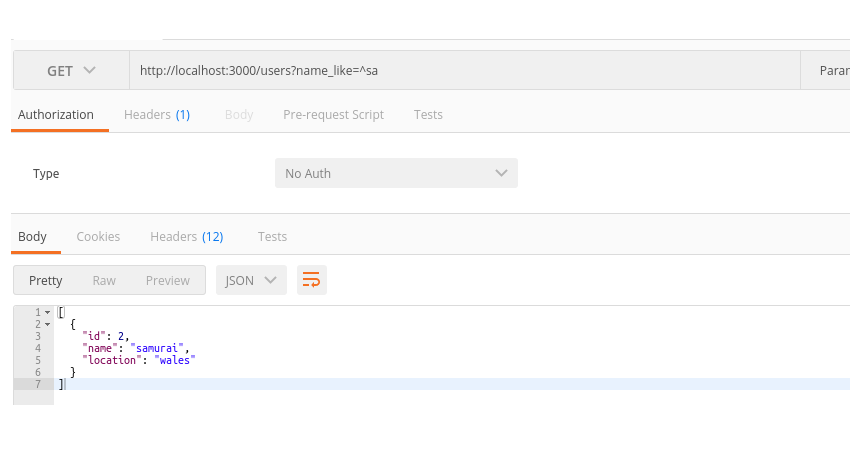
**Toán tử**

json-server cũng cung cấp chức năng để hỗ trợ các toán tử như việc tìm kiếm một mục với Id trong phạm vi giữa hai giá trị hoặc một mục phù hợp với một biểu thức chính quy cụ thể.

Để tìm một mục trong một phạm vi cụ thể, chúng ta có thể sử dụng các toán tử \_gte và \_lte. Ví dụ, để tìm người dùng với Id lớn hơn 1 và nhỏ hơn 2, tạo một yêu cầu GET đến URL <http://localhost:3000/users?id_gte=1&id_lte=2>như sau:



Nếu chúng ta muốn tìm kiếm tên bắt đầu bằng một ký tự nào đó, chúng ta có thể sử dụng biểu thức chính quy. Ví dụ, để tìm kiếm tên bắt đầu bằng chữ sa, chúng ta sẽ sử dụng toán tử \_like và tạo một yêu cầu GET đến URL [http://localhost:3000/users?name\_like=^sa](http://localhost:3000/users?name_like=%5Esa).



**Tổng kết**

Trong hướng dẫn này, bạn đã tìm hiểu cách làm thế nào để sử dụng json-server REST API để tạo ra một cơ sở dữ liệu giả lập để sử dụng nhanh chóng. Bạn cũng đã học được cách để bắt đầu sử dụng json-servervà truy vấn URL để thêm, cập nhật, chỉnh sửa và xóa dữ liệu. Bạn đã thấy cách để đánh số trang, sắp xếp và tìm kiếm dữ liệu giả. Bạn cũng đã biết cách sử dụng toán tử để tìm kiếm các mục sử dụng biểu thức chính quy.