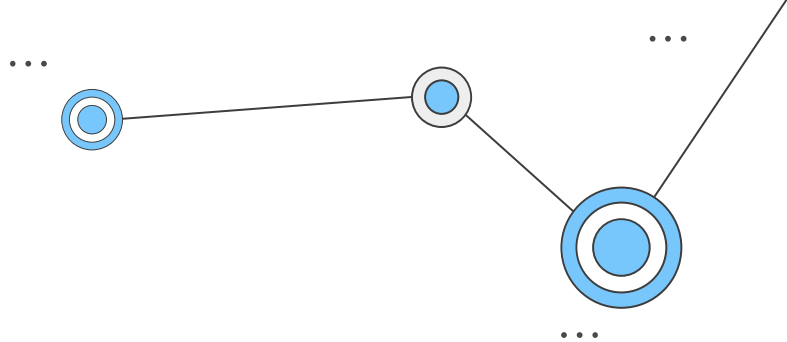




POSTMAN

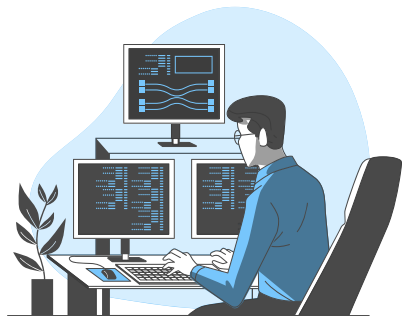


AXIOS



Khóa học Frontend

Bài 29: Học JSON Server,
Postman, Axios



Nội dung

01
...

JSON Server

02
...

Postman

03
...

FetchAPI

04
...

Axios

05
...

Ví dụ: Quản lý Sách





01

JSON Server



1.1. Khái niệm

- **JSON Server** là một thư viện giúp **tạo ra các API** từ một file JSON đơn giản.
- Chủ yếu dùng để **fake API**.

1.2. Hướng dẫn cài đặt

- **Bước 1:** Tạo 1 folder đặt tên là: **database**
- **Bước 2:** Chạy lệnh **npm init** để khởi tạo file package.json
- **Bước 3:** Chạy lệnh **npm install json-server@0.17.4**
- **Bước 4:** Tạo 1 file **db.json** với nội dung mẫu.
- **Bước 5:** Thêm vào mục script trong **package.json** dòng lệnh: **"start": "json-server --watch db.json"**
- **Bước 6:** Chạy lệnh **npm start**
- **Bước 7:** Truy cập thử các API để xem kết quả.



02

Postman



2.1. Khái niệm

- **Postman** là một **phần mềm** dùng để **kiểm thử các API** xem có trả ra kết quả đúng hay không.

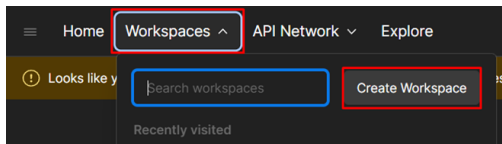
2.2. Cài đặt

- **Bước 1:** Đăng ký tài khoản online trên trang <http://www.postman.com>
- **Bước 2:** Tải phần mềm trên trang <https://www.postman.com/downloads>
- **Bước 3:** Đăng nhập vào phần mềm bằng tài khoản đã tạo ở bước 1.

2.3. Cách sử dụng

2.3.1. Khởi tạo một dự án

- **Bước 1:** Chọn Create Workspace.



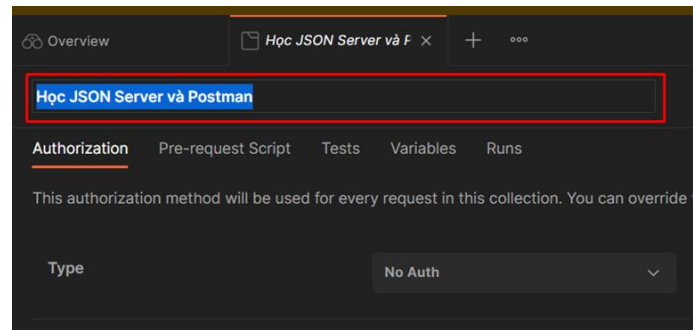
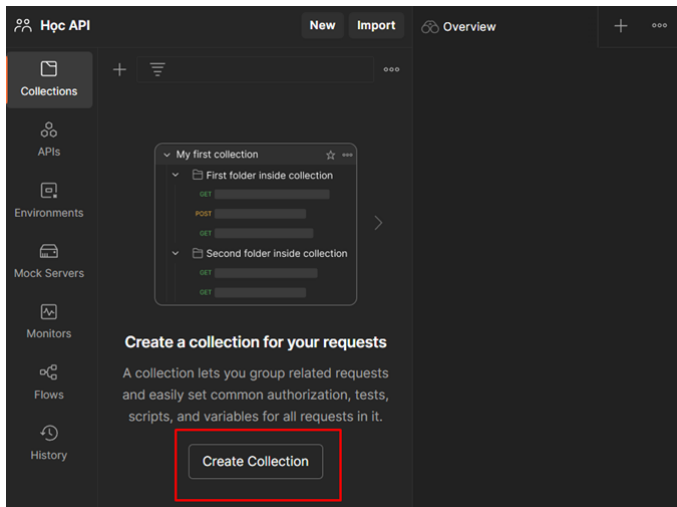
A screenshot of the 'Create workspace' form. The form has a dark background and includes the following fields and options:

- Name:** A text input field containing 'Hoc API'.
- Summary:** A text area with the placeholder text 'Add a brief summary about this workspace.'
- Visibility:** A section titled 'Determines who can access this workspace.' with the following options:
 - ☐ **Personal**: Only you can access
 - ☐ **Private**: Only invited team members can access
 - ☒ **Team**: All team members can access
 - ☐ **Partner** NEW: Only invited partners and team members can access
 - ☐ **Public**: Everyone can view
- Buttons:** At the bottom, there are two buttons: 'Create Workspace' (highlighted with a red outline) and 'Cancel'.

2.3. Cách sử dụng

2.3.1. Khởi tạo một dự án

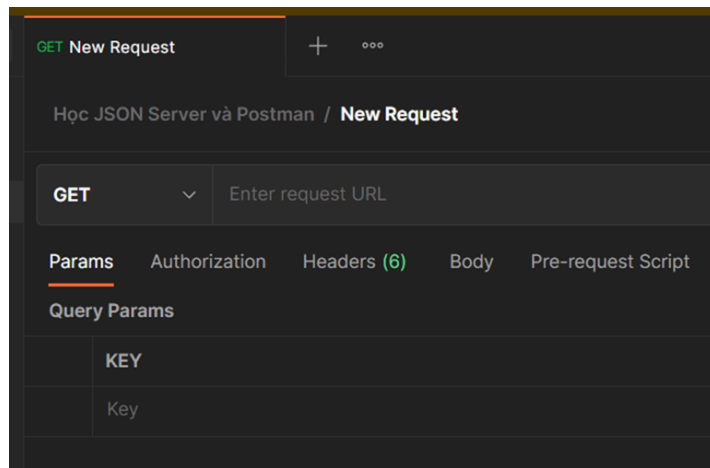
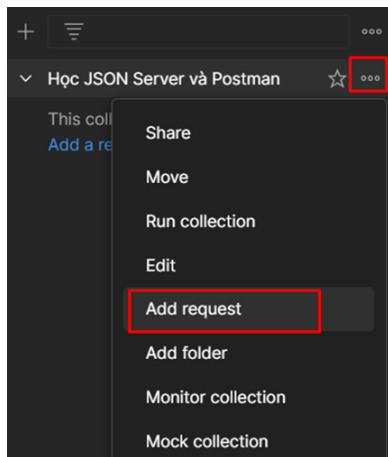
- **Bước 2:** Chọn Create Collection và đặt tên cho Collection.



2.3. Cách sử dụng

2.3.1. Khởi tạo một dự án

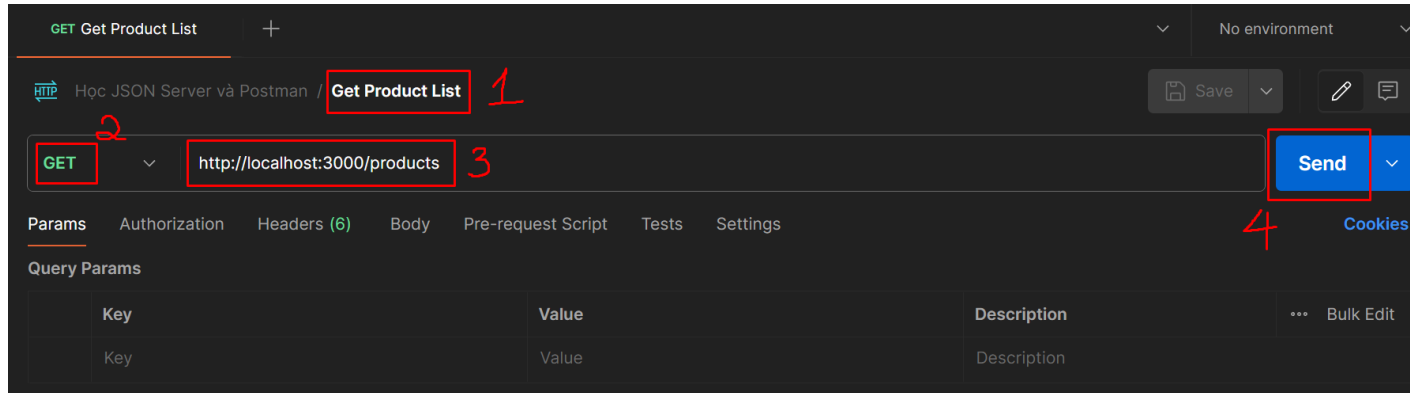
- **Bước 3:** Tạo mới một Request.



2.3. Cách sử dụng

2.3.2. Phương thức GET

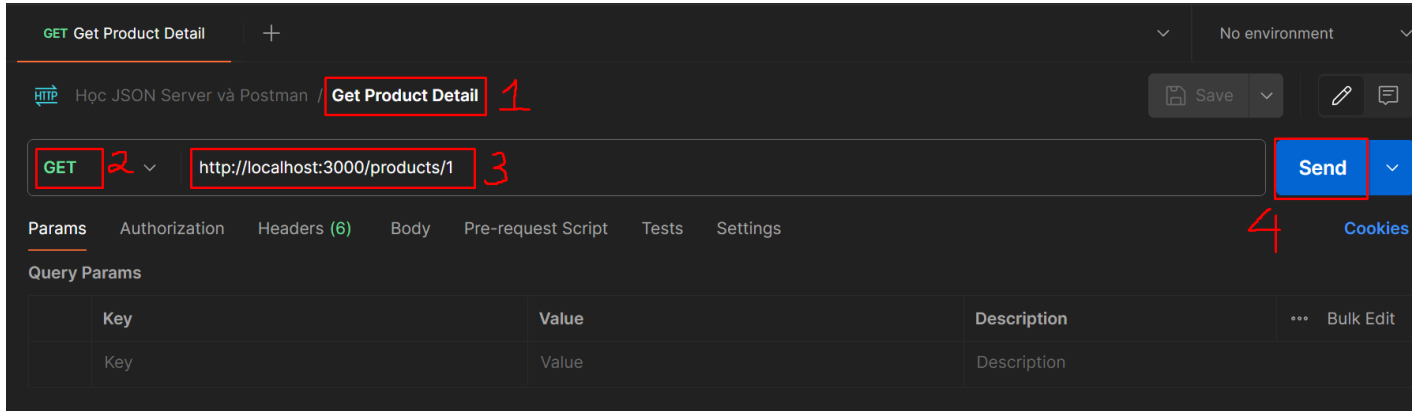
- Để **lấy** một hoặc nhiều bản ghi.
- Ví dụ 1: Lấy ra danh sách các sản phẩm.



2.3. Cách sử dụng

2.3.2. Phương thức GET

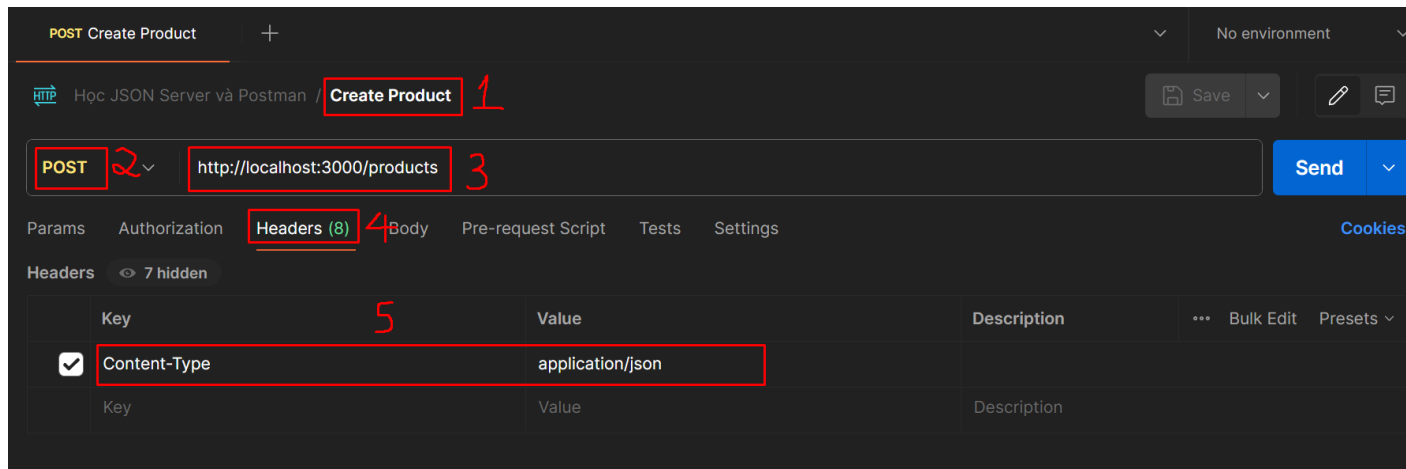
- Để **lấy** một hoặc nhiều bản ghi.
- Ví dụ 2: Lấy ra chi tiết một sản phẩm.



2.3. Cách sử dụng

2.3.3. Phương thức POST

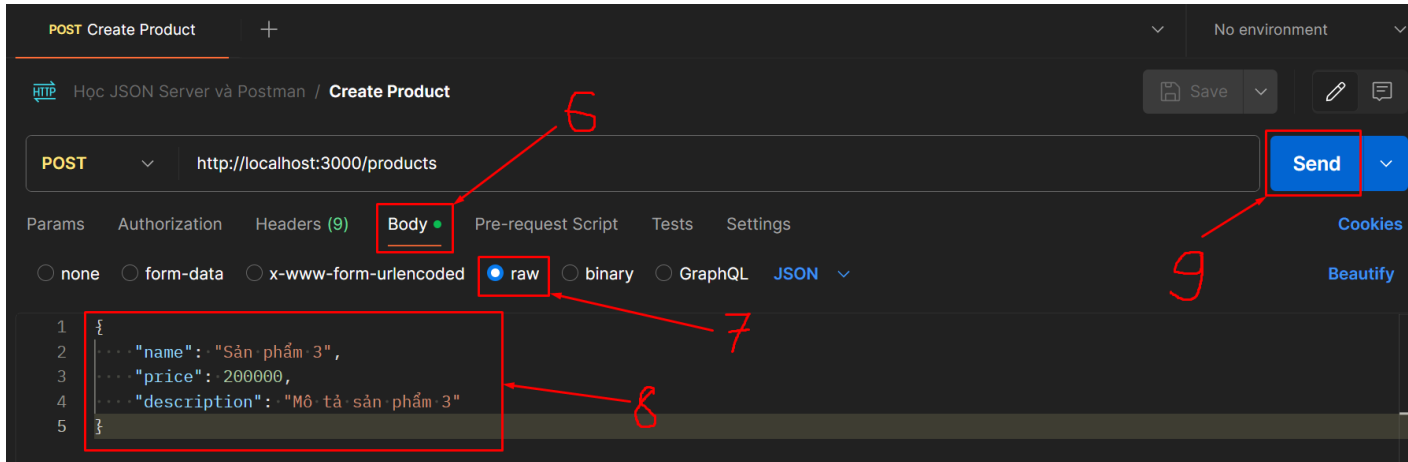
- Để **tạo mới** một bản ghi.
- Ví dụ: Tạo mới một sản phẩm. (Trường hợp data gửi lên dạng json thì headers phải thêm Content-Type là application/json)



2.3. Cách sử dụng

2.3.3. Phương thức POST

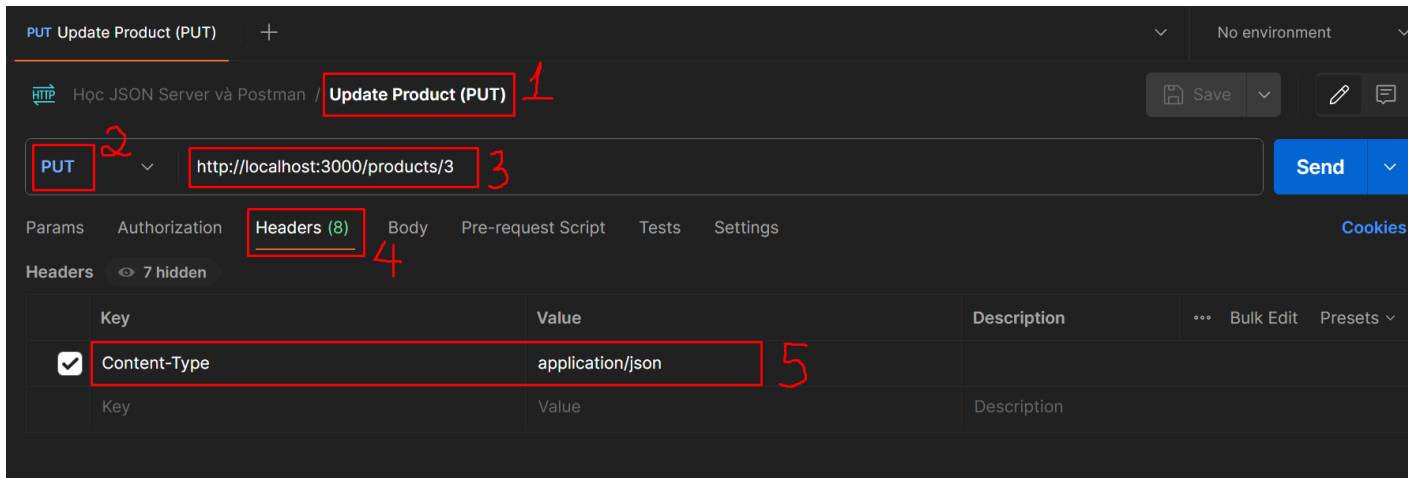
- Để **tạo mới** một bản ghi.
- Ví dụ: Tạo mới một sản phẩm. (Trường hợp data gửi lên dạng json thì headers phải thêm Content-Type là application/json)



2.3. Cách sử dụng

2.3.4. Phương thức PUT

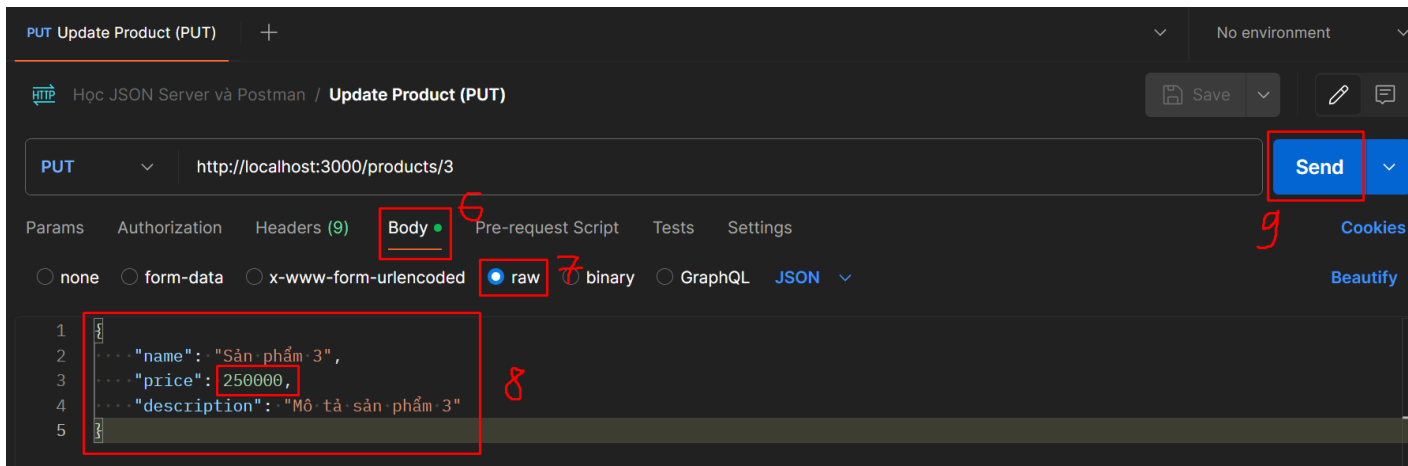
- Để **cập nhật** một bản ghi.
- Nhưng phải gửi lên **đủ các cặp key/value** (kể cả những cặp key/value không cần cập nhật).
- Ví dụ: Cập nhật giá của một sản phẩm (Nhưng phải gửi đủ các cặp key/value).



2.3. Cách sử dụng

2.3.4. Phương thức PUT

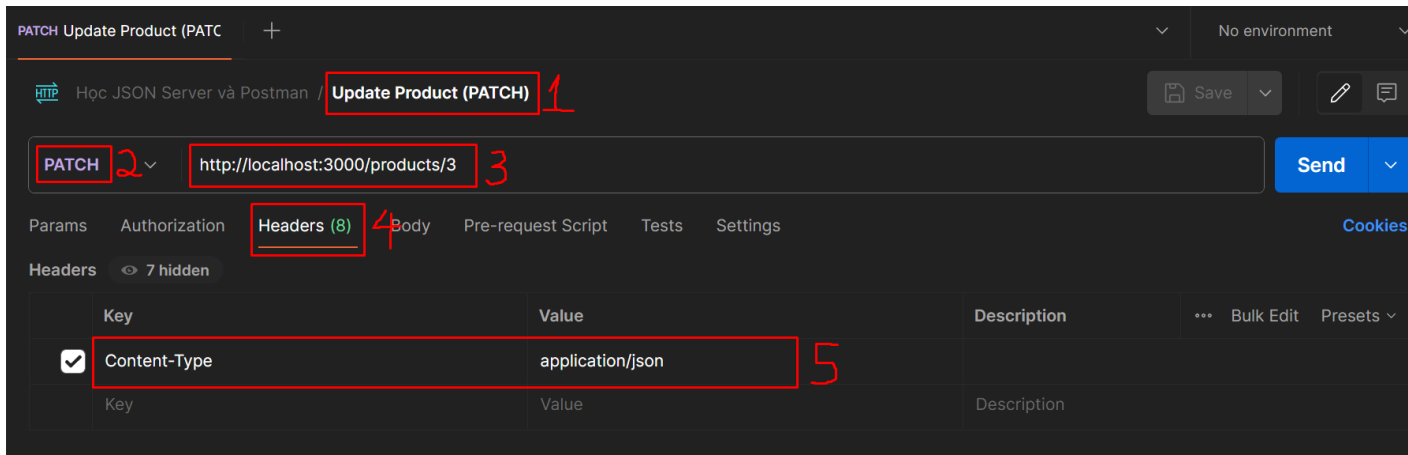
- Để **cập nhật** một bản ghi.
- Nhưng phải gửi lên **đủ các cặp key/value** (kể cả những cặp key/value không cần cập nhật).
- Ví dụ: Cập nhật giá của một sản phẩm (*Nhưng phải gửi đủ các cặp key/value*).



2.3. Cách sử dụng

2.3.5. Phương thức PATCH

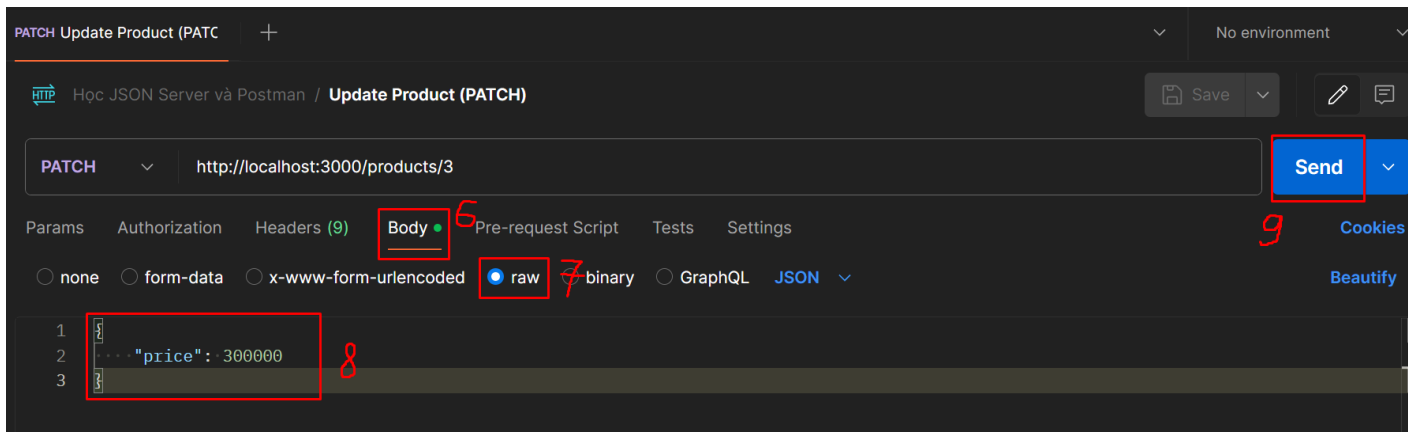
- Để **cập nhật** một bản ghi.
- **Chỉ cần** gửi lên các **cặp key/value cần cập nhật**.
- Ví dụ: Cập nhật giá của một sản phẩm (*Chỉ cần gửi lên cặp key/value cần cập nhật*).



2.3. Cách sử dụng

2.3.5. Phương thức PATCH

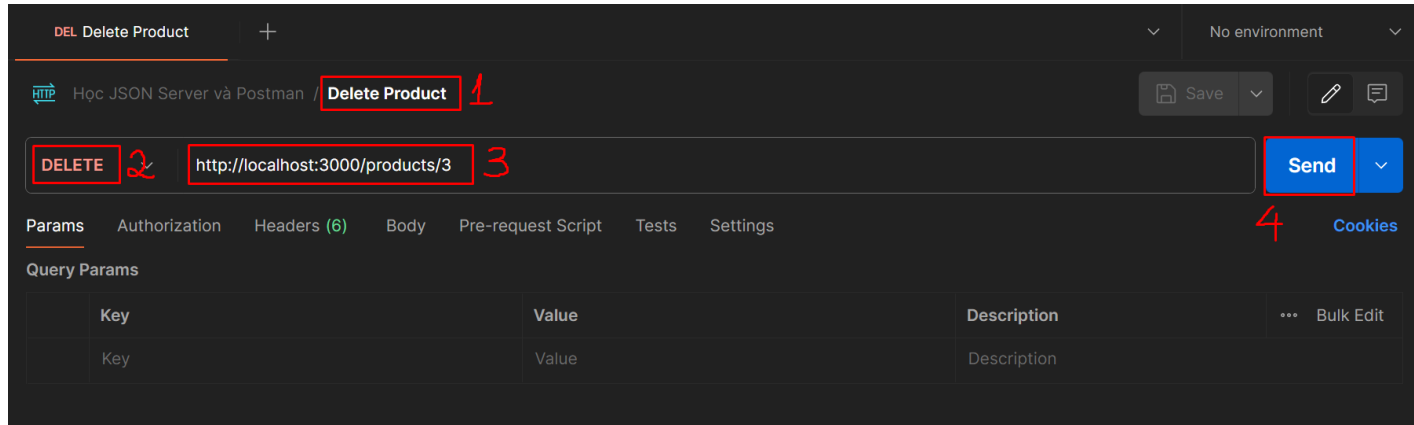
- Để **cập nhật** một bản ghi.
- **Chỉ cần** gửi lên các **cặp key/value cần cập nhật**.
- Ví dụ: Cập nhật giá của một sản phẩm (*Chỉ cần gửi lên cặp key/value cần cập nhật*).



2.3. Cách sử dụng

2.3.6. Phương thức DELETE

- Để **xóa** một bản ghi.
- Ví dụ: Xóa một sản phẩm.



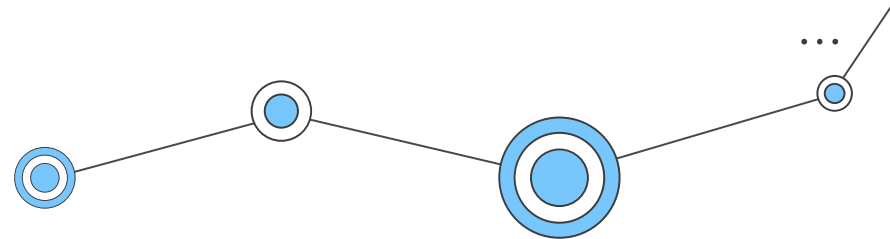
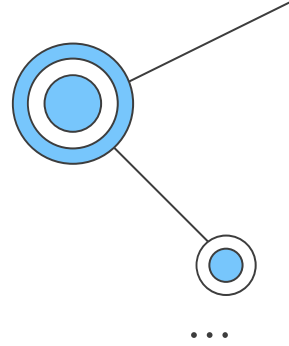
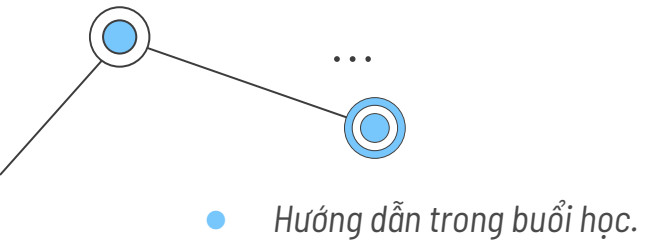


03

FetchAPI



03. FetchAPI





04

Axios

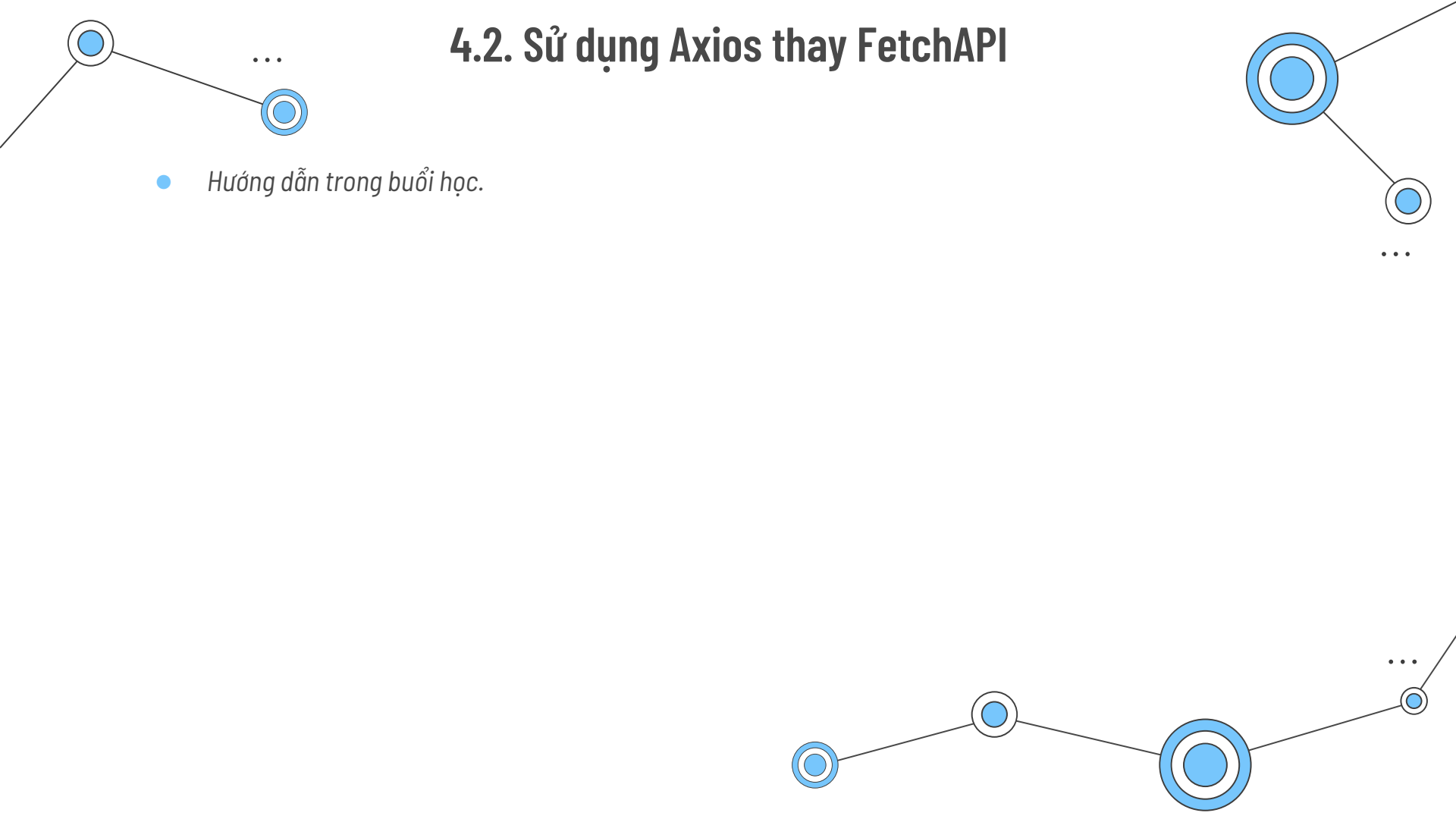


4.1. Khái niệm

- **Axios** là thư viện được sử dụng trong Javascript để thực hiện **gửi yêu cầu HTTP từ client lên server**.
- Hay hiểu đơn giản: Axios để call API tương tự hàm FetchAPI nhưng sử dụng dễ hơn.
- Trang chủ: <https://axios-http.com/vi>

4.2. Sử dụng Axios thay FetchAPI

● Hướng dẫn trong buổi học.



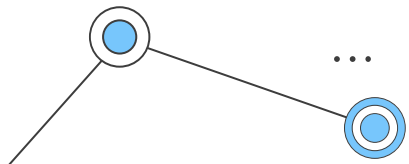
4.3. Ưu điểm của Axios so với FetchAPI

- **Dễ sử dụng:** Axios cung cấp cú pháp ngắn gọn hơn so với FetchAPI.
- **Hỗ trợ tự động biến đổi dữ liệu:** Axios tự động biến đổi dữ liệu trước khi gửi hoặc sau khi nhận phản hồi.
- **Tính tương thích cao hơn:** Axios hỗ trợ tốt hơn trên các trình duyệt cũ, còn FetchAPI không được hỗ trợ trên một số trình duyệt cũ.

05

Ví dụ: Quản lý Sách

05. Ví dụ: Quản lý Sách



• Hướng dẫn trong buổi học.

