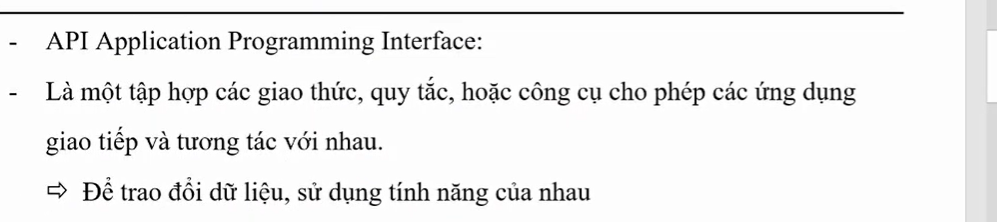
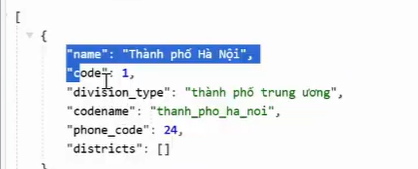
***API\_Học Lại***

***Api thực chất là một phần mềm***

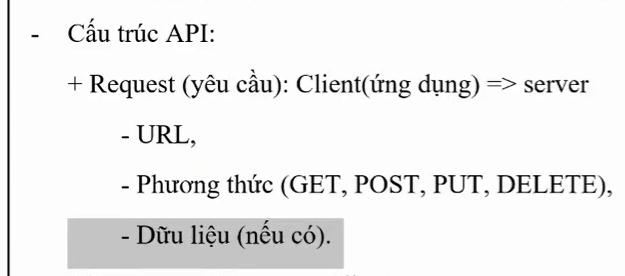


*(tương tác với database sẽ phải thông qua api -> trung gian)*



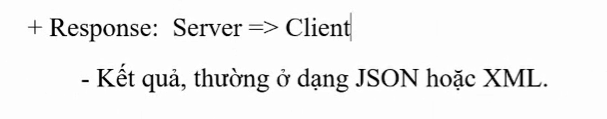
* ***Định dạng theo json***

***ở một nơi nào đó viết ra bộ api***



**Request: là yêu cầu từ ng dùng gửi lên server(lệnh gửi lên)**

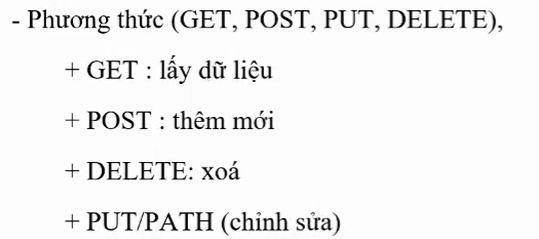
(api lấy dữ liệu từ cơ sở dữ liệu (sql lên)



**Response: là kho dữ liệu mà server gửi về cho ng dùng(kết quả hiển thị**

**Json Server là một công cụ để tạo một | giả lập API**

**Gồm các phương thức**



**Cách cài json server :**

**npm i -g** [**json-server@0.17.4**](mailto:json-server@0.17.4)

**kiểm tra phiên bản :**

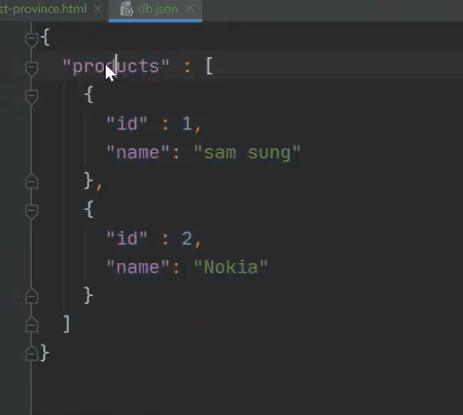
**json-server -v**

**web để học:** [**json-server - npm**](https://www.npmjs.com/package/json-server/v/0.17.4)

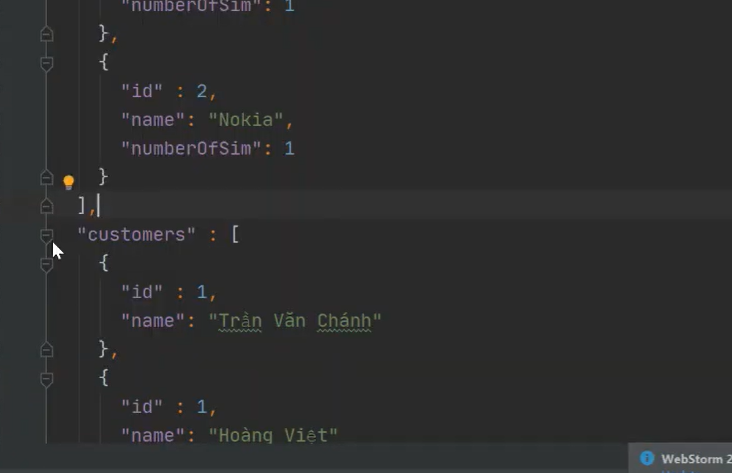
*tạo một file db.json => như một kho api giả lập*

*các thuộc tính của một đối tượng sẽ trong “ ”*

*thực chất json là một đối tượng*



Có thể có nhiều thuộc tính to trong đối tượng

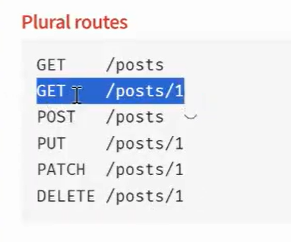


Để chạy kho api giả lập

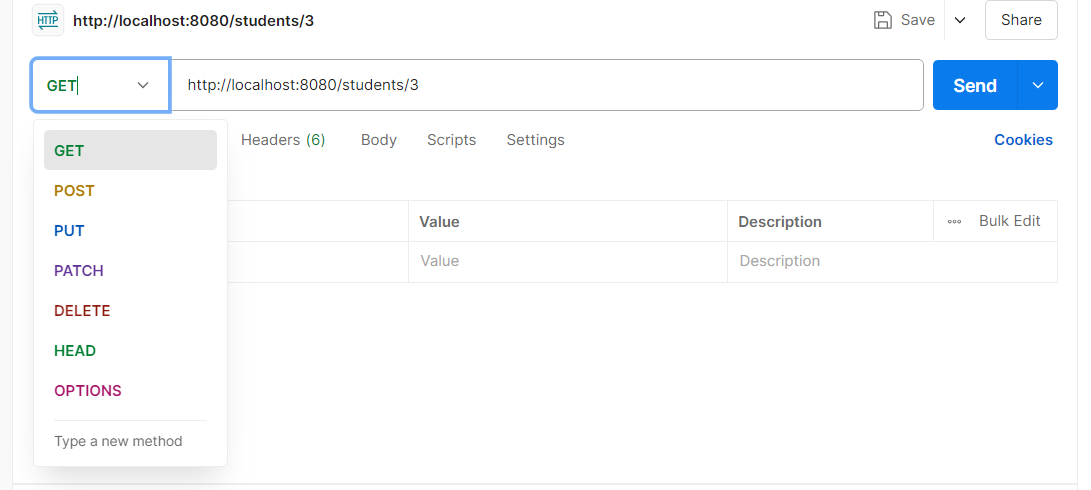
* Link đến thư mục chứ json
* Sử dụng câu lệnh: json-server --watch db.json
* ở server chạy nó là port 3000 nên là phải chỉnh lại port cho kho
* : json-server --watch db.json –-port 8080

Trong đó –port 8080 để chỉ định port hiển thị

Khi nó chạy lên sẽ mặc định tạo ra một con server (backend giả)

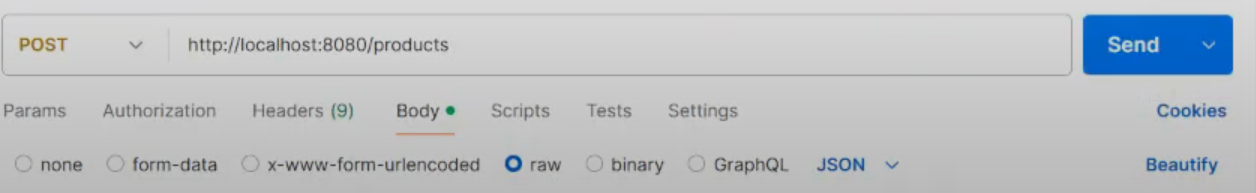


Phương thức get nhưng /id thì là detaill ( vì trả về một đối tượng)



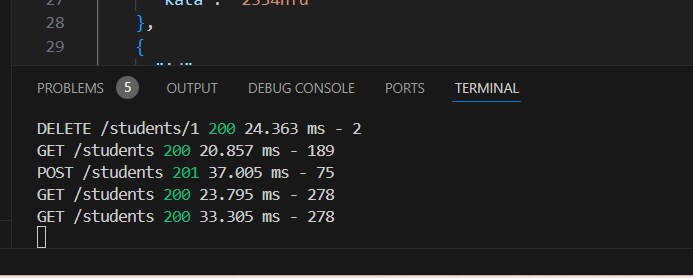
Demo phương thức( nhưng ảnh hưởng đến kho)

**Các phương thức như xóa chỉnh, sửa… đều sử dụng dc**



Sử dụng post để thêm:

Với tùy chọn body, raw => chỉnh sửa => send



Tuyệt đối k dc tắt tab này => sẽ mất ( tab này là json server giả lập backenđ)

Cách gọi API ( sử dụng axios)

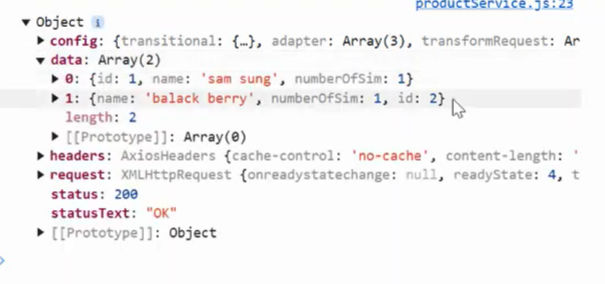
***npm i axios***

File: service

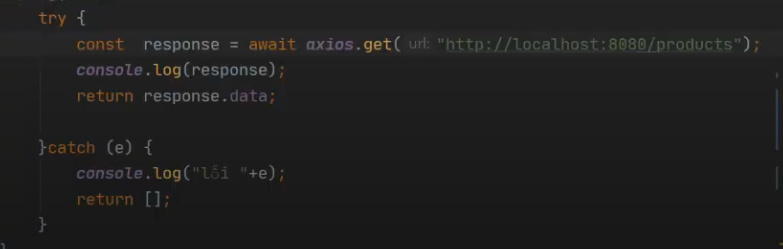
**cần import**



**Để gọi api**

* *Sử dụng cấu trúc try-catch*
* *Khai báo một biến response *

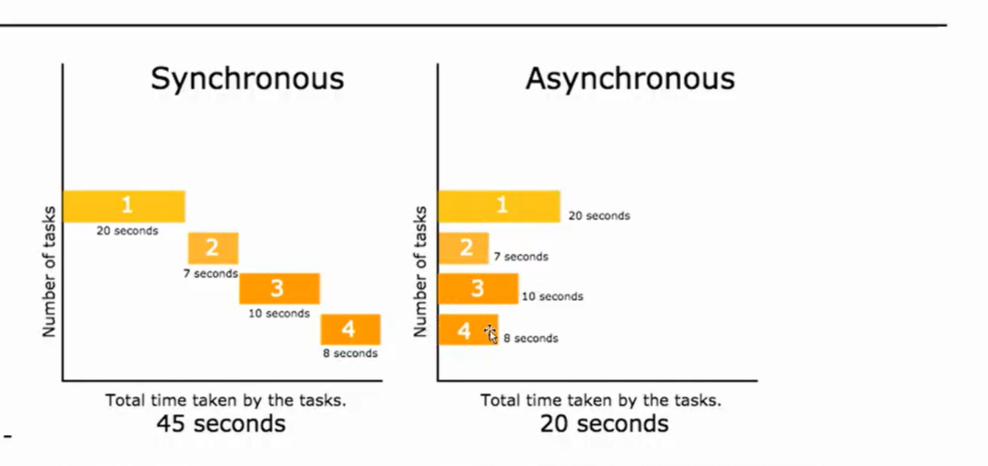
*( để lấy data từ web)=axios.get(“duonglink api);*

* *Biến response lưu kho api => khi gọi đến dùng thuộc tính data để lấy dữ liệu*
* 
* Với async ở đầu để định dạng hàm
* Và await … là để chờ thực hiện xong hết tác vụ

ở file list sử dụng biến fetchData để gọi data lên

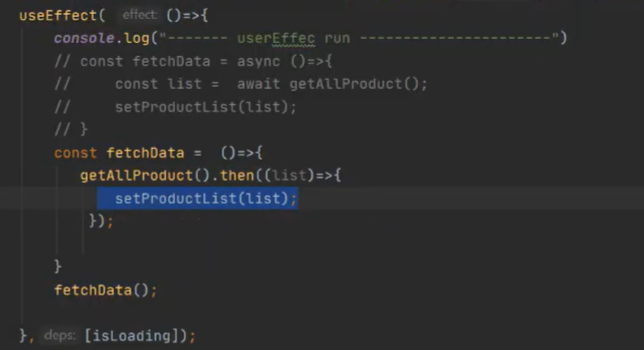
* ở useEffect sử dụng biến để gán gọi dữ liệu và gán set lại





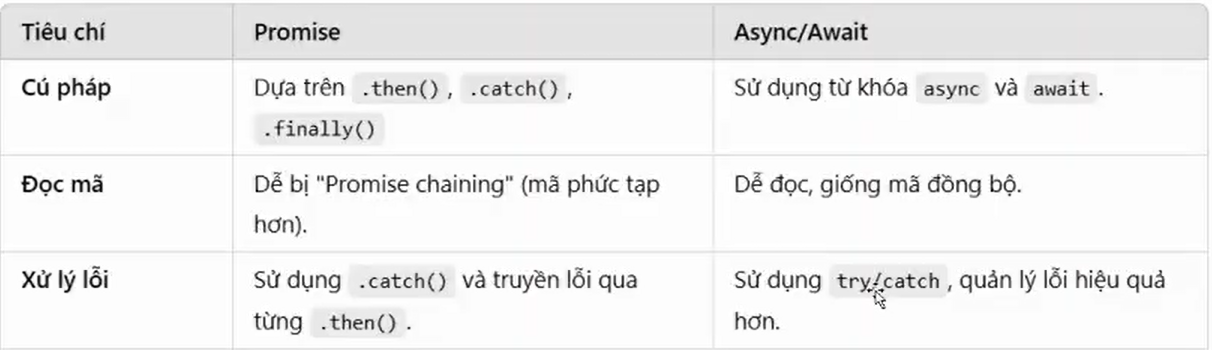
Async-await để sử lý đồng bộ còn k thì mặc định là bất đồng bộ

Hoặc ngoài sử dụng đó có thể sử dụng then



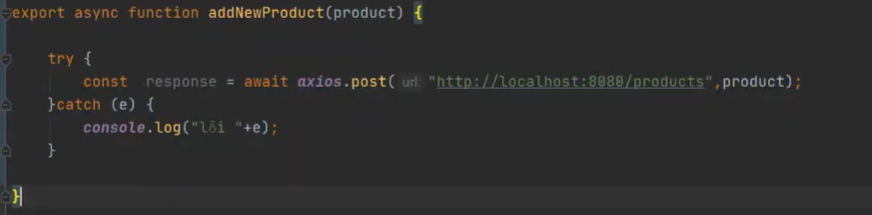
Chỉ cần .then(tên biến mới) => xử lý bất đồng bộ

Bản chất async-await dựa trên promise

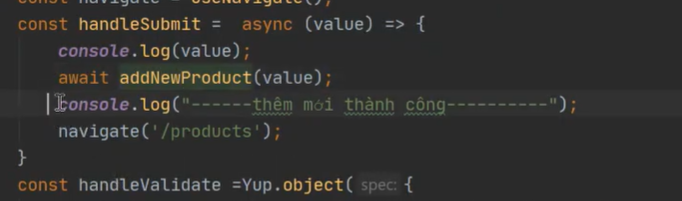


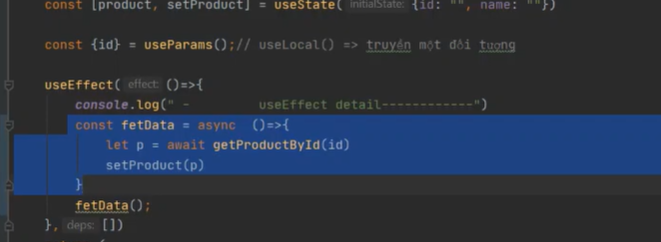
Phương thức thêm( từ kho)

Sử dụng axios.post(url,đối tượng thêm)



Thay đổi hàm submit của add (vẫn nên sử dụng async-await)



* chỉnh hàm get id bằng +id ở cuối (đường dẫn)
* 
* Chỉnh hàm detail 
* Chủ yếu sử dụng fetData để thay vào hàm set ( thay đổi thuộc tính ở trong useEffect)
* Kết hợp với async await để xử lý đồng bộ