***API***

1. ***Json server? Cài đặt và sử dụng***

JSON Server là một công cụ tuyệt vời cho phép bạn tạo nhanh một API REST giả lập với dữ liệu JSON. Điều này đặc biệt hữu ích cho việc phát triển front-end khi bạn cần dữ liệu mẫu để làm việc mà không cần phải chờ đợi backend hoàn thiện.

**Cài đặt**

Sử dụng npm để cài đặt json-server:

npm install -g json-server

**Chuẩn bị dữ liệu**

Tạo một file JSON (ví dụ: db.json) chứa dữ liệu cho API của bạn.

**Khởi động JSON Server**

Chạy lệnh sau trong terminal để khởi động server:

json-server --watch db.json

**Sử dụng API**

JSON Server sẽ tạo ra các endpoint API RESTful dựa trên cấu trúc dữ liệu trong file db.json. Ví dụ, với dữ liệu trên, bạn sẽ có các endpoint sau:

* GET /posts: Lấy danh sách tất cả các bài viết
* GET /posts/1: Lấy bài viết có id là 1
* POST /posts: Tạo một bài viết mới
* PUT /posts/1: Cập nhật bài viết có id là 1
* DELETE /posts/1: Xóa bài viết có id là 1

***2.        So sánh Promise và Async/Await?***



***3.        Cách triển khai Axios?***

***Kn***: -Axios là một trong những thư viện HTTP được các nhà phát triển Javascript yêu thích. Nó được sử dụng để gửi các yêu cầu HTTP, đặc biệt là các yêu cầu AJAX, từ phía máy khách cũng như phía máy chủ.

-Axios đơn giản hóa công việc này bằng cách cung cấp một giao diện trừu tượng dễ sử dụng trên giao diện XMLHttpRequest của Javascript với một số chức năng và tùy chọn cấu hình thân thiện với người dùng.

**Cài đặt**

npm install axios

hoặc

yarn add axios

**Tạo Request với Axios**

Giống như với hàm $.ajax của jQuery, Chúng ta có thể tạo bất kỳ một request HTTP nào bằng cách truyền vào các object option cho Axios

Ví dụ :

axios({

method: 'post',

url: 'https://jsonplaceholder.typicode.com/users',

data: { user }

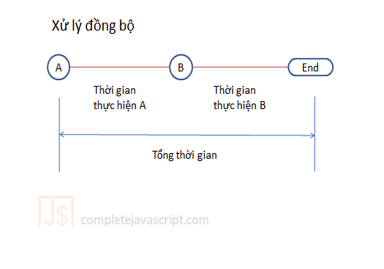
});

Một số phương thức HTTP mà chúng ta hay sử dụng(GET/POST/DELETE)

***4.        Phân biệt đồng bộ và bất đồng bộ***

**Đồng bộ** : có nghĩa là thực hiện các công việc một cách tuần tự, công việc này xong thì mới được thực hiện các công việc khác.

Ví dụ có 2 công việc A và B thì khi có nghĩa là A thực hiện xong trước rồi mới tới lượt B. Điều này nó sẽ ảnh hưởng đến hiệu suất của người dùng



**Bất đồng bộ :** Với cách xử lý bất đồng bộ, khi A bắt đầu thực hiện, chương trình tiếp tục thực hiện B mà không đợi A kết thúc.

Việc mà bạn cần làm ở đây là cung cấp một phương thức để chương trình thực hiện khi A hoặc B kết thúc.

Cơ chế giúp bạn thực hiện việc này trong JavaScript là sử dụng Callback, Promise hoặc Async/await.

