**BÀI THU HOẠCH BUỔI 1**

**NHÓM THỰC HIỆN: LÊ VĂN ĐÔNG, NGUYỄN VĂN ĐẠI**

**PHẦN 1. REST API LÀ GÌ ?**

**API** (**A**pplication **P**rogramming **I**nterface – giao diện lập trình ứng dụng) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API không phải giao diên UI hiển thị cho người dùng cuối mà là giao diện giữa các ứng dụng làm việc với nhau API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như [JSON](https://topdev.vn/blog/json-la-gi/) hay XML.

VD: Client( website, mobile,…) khi muốn lấy dữ liệu về các bài báo từ một server, nó sẽ gửi yêu cầu tới server bằng một đường dẫn URL có dạng <https://baomoi.com/api/news> khi này server sẽ đọc URL và xử lí công việc tương ứng với URL cụ thể là trả về danh sách sản phẩm, client nhận lại danh sách các bài báo và hiển thị cho người dùng.

Việc sử dụng API giúp cho client không cần quan tâm server lập trình bằng ngôn ngữ nào( PHP, java, .NET … ), xử lí như thế nào, chỉ cần trả về dữ liệu theo yêu cầu, dễ dàng phát triển ứng dụng( mobile, website, windows…).

**REST**( **Re**presentational **S**tate **T**ransfer) hiểu đơn giản nó là một bộ các ràng buộc và quy ước, khi áp dụng đầy đủ vào hệ thống của bạn thì ta có 1 hệ thống REST. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

**REST API** là việc áp dụng các quy chuẩn của REST vào trong việc thiết kế API một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các tài nguyên. REST là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng khác nhau giao tiếp với nhau.Việc viết theo quy chuẩn có rất nhiều lợi ích: nhìn vào API có thể đoán được nó dùng để làm gì, thống nhất được cấu trúc cách đặt tên thuận tiện cho phát triển.

**PHẦN 2. TỔ CHỨC API ?**

Là cách viết các API theo một chuẩn chung, giúp dễ đọc dễ sử dụng.

Sử dụng các phương thức :

* GET (SELECT): Trả về một Resource hoặc một danh sách Resource.
* POST (CREATE): Tạo mới một Resource.
* PUT (UPDATE): Cập nhật thông tin cho Resource.
* PATCH (UPDATE): Cập nhật một thành phần, thuộc tính của Resouce.
* DELETE (DELETE): Xoá một Resource.
* HEAD – Trả về thông tin chung của một hoặc danh sách Resource.
* OPTIONS – Trả về thông tin mà người dùng được phép với Resource.

Chuẩn dữ liệu trả về: Nên sử dụng JSON làm chuẩn dữ liệu giao tiếp bởi nhẹ dễ dàng làm việc do được tổ chức dưới dạng key và value.

Đặt tên cho API : để dễ dàng sử dụng ta nên đặt tên API lên quan tới mục đích của URL kết hợp việc sử dụng các phương thức:

VD:

GET <https://baomoi.com/api/baibao> : nhận về một danh sách các bài báo

GET <https://baomoi.com/api/baibao/1> : nhận về một bài báo có mã là 1

POST <https://baomoi.com/api/baibao> : đăng một bài báo mới lên website

DELETE <https://baomoi.com/api/baibao/1> : xóa về một bài báo có mã là 1

PUT <https://baomoi.com/api/baibao/1> : cập nhật lại một bài báo có mã là 1

Status code

Tùy thuộc vào trạng thái của server mà mã trạng thái sẽ gửi về cho client:

* 200 OK – Trả về thành công cho những phương thức GET, PUT, PATCH hoặc DELETE.
* 201 Created – Trả về khi một Resouce vừa được tạo thành công.
* 204 No Content – Trả về khi Resource xoá thành công.
* 304 Not Modified – Client có thể sử dụng dữ liệu cache.
* 400 Bad Request – Request không hợp lệ.
* 401 Unauthorized – Request cần có sự authentication.
* 403 Forbidden – Server hiểu request nhưng bị từ chối không cho phép.
* 404 Not Found – Không tìm thấy route từ URI
* 405 Method Not Allowed – Phương thức không cho phép với user hiện tại.
* 410 Gone – Resource không còn tồn tại, Version cũ đã không còn hỗ trợ.
* 415 Unsupported Media Type
* 422 Unprocessable Entity – Dữ liệu không được kiểm chứng
* 429 Too Many Requests – Request bị từ chối do bị giới hạn

Việc tổ chức API như vậy ta thấy rất tường minh và dễ sử dụng.

**PHẦN 3. CALL API ?**

Khi đã có API làm thế nào để web Client có thể sử dụng, dưới đây là các cách call API ngôn ngữ js:

1. Sử dụng xhr

const xhr = new XMLHttpRequest();

const url = ‘https://baomoi.com/api/baibao/1’

xhr.open( ‘GET’, url);

xhr.responseType = ‘json’;

xhr. send();

xhr.onload = function () {

if (xhr.status != 200) {

alert( `error ${xhr.status} : ${xhr.statusText}` );

}

else{

Console.log(xhr.response);

}

1. Sử dụng fetch:

try {

const urL = 'https://baomoi.com/api/baibao /1 ' ;

const response = await fetch(url);

const responseJS0N = await response. json() ;

console.log (responseJS0N) ;

}   
catch (error) {

conso1e. log('Fail to fetch products: ', error);

}

**PHẦN 4. CHỨC NĂNG SEARCH:**

Đầu tiên là một form html gồm một input để người dùng nhập và một nút tìm kiếm để gửi nội dung.

Khi người dùng gõ vào khung tìm kiếm, js sẽ bắt sự kiện onchange của thẻ input cứ mỗi lần nhập thay đổi nội dung thẻ nó sẽ call api kèm theo nội dung thẻ có dạng: https://baomoi.com/api/news/search?q=nội+dung+tìm+kiếm, khi này server sẽ lấy chuỗi truy vấn gửi lên và tìm kiếm các bài viết bằng cách đọc từ nguồn dữ liệu( cụ thể là database) những nội dung có chứa chuỗi tìm kiếm, chuyển về dạng chuỗi JSON trả về cho client.

Khi này client sẽ nhận được chuỗi JSON và chuyển thành mảng, sẽ có danh sách các bài viết liên quan đến chuỗi tìm kiếm và hiển thị dưới dạng gợi ý ở phía dưới thẻ input lưu ý chỉ gợi ý giới hạn ví dụ như 10 bài viết.

Người dùng có thể chọn phần gợi ý hoặc nhập hết chuỗi mình cần tìm nhất nút gửi, khi này js sẽ call api để nhận về hết các kết quả tìm kiếm và hiển thị hết trên một trang html.