RESTFUL API

Các thành phần chính:

a, API ( Application Programming Interface)

1, Khái niệm:

-là tập hợp các quy tắc cơ chế, như cầu nối cho phép các chương trình khác nhau kết nối với nhau.

- API trả dữ liệu cho Client yêu cầu sử dụng thông qua kiểu dữ liệu JSON hay XML.

2, Ứng dụng:

-Web API : là hệ thống API được sử dụng trong các hệ thống website. Giúp kết nối thông tin, tổng hợp dữ liệu Client yêu cầu.

-API trên hệ điều hành: Windows hay Linux đều có nhiều API, cung cấp tài liệu API là đặc tả các hàm, phương thức cũng như các giao thức kết nối, giúp cho lập tình viên có thể tạo ra các phần mềm ứng dụng có thể tương tác trực tiếp với hệ điều hành của mình.

-API của thư viện phần mềm hay framework: API mô tả và quy định các hành động mong muốn mà các thư viện cung cấp.

3,Ưu và nhược điểm:

\*Ưu điểm:

-Khả năng tích hợp linh hoạt, dễ dàng lấy nội dung từ bất kì website hoặc ứng dụng để tăng trải nghiệm người dùng.

-Tự động hóa sản phẩm trong việc sử dụng web API, quá trình quản lý công việc sẽ diễn ra một cách tự động hóa, cập nhập công việc, giúp tăng năng suất, tạo hiệu quả cao và dễ dàng quản lý.

-Cập nhập thông tin thời gian thực: API có khả năng thay đổi cà cập nhập thời gian thực, công nghệ được ứng dụng để truyền dữ liệu tốt hơn, chính xác và cung cấp dịch vụ linh hoạt hơn.

-Có tiêu chuẩn chung dễ sử dụng nên dễ dàng điều chỉnh,hỗ trợ cho các MVP.

\* Nhược điểm:

+Web API chưa hoàn toàn là RESTful service, mới chỉ hỗ trợ mặc định GET, POST

+ Đòi hỏi kiến thức chuyên sâu và kinh nghiệm backend tốt.

+ Tốn thời gian và chi phí cho việc nâng cấp và vận hành

+Tính bảo mật không cao nếu không giới hạn điều kiện kỹ.

b, RESTFUL (Representational State Transfer)

1, Khái niệm

-là một phương thức chuyển đổi cấu trúc dữ liệu, thiết kế các API cho ứng dụng web.

- sử dụng phương thức HTTP đơn giản để tạo sự liên kết cho các máy, sử dụng động từ HTTP tiêu chuẩn để làm việc với tài nguyên như: get, post, put, patch, delete.

2,Tác dụng:

-làm API dễ dàng sử dụng và thuận tiện khi scale hệ thống.

-code đơn giản và ngắn gọn.

-Giúp cho Front-end và Back-end dễ dàng tìm thấy tiếng nói chung.

3. Gồm 4 thành phần chính của 1 request

\* Endpoint (url): là đường dẫn à Client yêu cầu tới.

\*Method (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.

-DELETE (DELETE): Xoá một Resource.

\*Headers: là tiêu đề của yêu cầu cung cấp thông tin cho client và server

\*Body: chứa tất cả thông tin mà client muốn gửi hco server

HTTP status code

-HTTP status code là yếu tố quan trọng trong HTTP message response được server trả về cho client

* 2xx Success (Thành công):
  + 200 OK - Request đã được tiếp nhận và xử lý thành công.
  + 201 Created - Trả về khi một resouce vừa được tạo thành công.
  + 202 Accepted - Request được chấp nhận cho xử lý, nhưng việc xử lý chưa hoàn thành.
  + 204 No Content - Server đã xử lý thành công request nhưng không trả về bất cứ content nào.
* 4xx Client Error (Lỗi Client):
  + 400 Bad Request - Request không hợp lệ.
  + 401 Unauthorized - Request cần có auth.
  + 403 Forbidden - bị từ chối không cho phép.
  + 404 Not Found - Không tìm thấy resource từ URI.
* 5xx Server Error (Lỗi Server):
  + 500 Internal Server Error - Một thông báo chung chung, được đưa ra khi Server gặp phải một trường hợp bất ngờ.
  + 501 Not Implemented - Server không công nhận các Request method hoặc không có khả năng xử lý nó.

Note: Một số nguyên tắc thiết kế chính của RESTful API sử dụng HTTP:

1. Một resource có một URL để định danh và nó là duy nhất nên tránh thiết kế phức tạp. Collection/item/collection

2. Sử dụng danh từ để biểu thị 1 collection

3.Dùng dấu (-) để ngăn cách giữa các cụm từ

4.Sử dụng chữ thường cho toàn bộ endpoint