**Báo cáo tìm hiểu về Rest api**

1. Khái niệm

* Api (Application programming Interface) là một tập 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 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.
* Rest (**RE**presentational **S**tate **T**ransfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. 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 một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. REST là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile…) khác nhau giao tiếp với nhau.

1. Yêu cầu

* Mã định danh tài nguyên duy nhất

Máy chủ xác định từng tài nguyên bằng mã định danh tài nguyên duy nhất. Đối với các dịch vụ REST, máy chủ thường thực hiện việc xác định tài nguyên bằng cách sử dụng Hệ thống định vị tài nguyên thống nhất (URL). URL chỉ định đường dẫn đến tài nguyên. URL tương tự như địa chỉ trang web mà bạn nhập vào trình duyệt của mình để truy cập vào bất kỳ trang web nào. URL còn được gọi là điểm cuối yêu cầu và chỉ định rõ cho máy chủ rằng client yêu cầu điều gì.

* Phương thức

Các nhà phát triển thường triển khai các API RESTful bằng cách sử dụng Giao thức truyền siêu văn bản (HTTP). Phương thức HTTP cho máy chủ biết máy chủ cần làm gì với tài nguyên. Sau đây là 4 phương thức HTTP phổ biến:

* Get

Client sử dụng GET để truy cập các tài nguyên nằm tại URL được chỉ định trên máy chủ. Client có thể lưu vào bộ nhớ đệm các yêu cầu GET và gửi các tham số trong yêu cầu API RESTful để chỉ dẫn máy chủ lọc dữ liệu trước khi gửi.

* Post

Client sử dụng POST để gửi dữ liệu đến máy chủ. Client gửi kèm dạng biểu diễn dữ liệu với yêu cầu. Gửi cùng một yêu cầu POST nhiều lần có tác dụng phụ là tạo ra cùng một tài nguyên nhiều lần.

* Put

Client sử dụng PUT để cập nhật các tài nguyên hiện có trên máy chủ. Không giống như POST, gửi cùng một yêu cầu PUT nhiều lần trong một dịch vụ web RESTful sẽ cho cùng một kết quả.

* Delete

Client sử dụng yêu cầu DELETE để xóa tài nguyên. Yêu cầu DELETE có thể thay đổi trạng thái máy chủ. Tuy nhiên, nếu người dùng không vượt qua quy trình xác thực thích hợp, yêu cầu sẽ không thành công.

* Tiêu đề Http

Tiêu đề yêu cầu là siêu dữ liệu được trao đổi giữa client và máy chủ. Ví dụ: tiêu đề yêu cầu cho biết định dạng của yêu cầu và phản hồi, cung cấp thông tin về trạng thái yêu cầu, v.v.

* Dữ liệu

Các yêu cầu API REST có thể bao gồm dữ liệu cho POST, PUT và các phương thức HTTP khác để hoạt động thành công.

* Tham số

Các yêu cầu API RESTful có thể bao gồm các tham số cung cấp cho máy chủ thêm chi tiết về những gì cần phải thực hiện. Sau đây là một số loại tham số khác nhau:

* Tham số đường dẫn chỉ định chi tiết của URL.
* Tham số truy vấn yêu cầu thêm thông tin về tài nguyên.
* Tham số cookie xác thực client một cách nhanh chóng.

1. Quy tắc

* Những yếu tố quyết định chất lượng Api tốt
* Self-document: nhìn vào đường dẫn API là có thể đoán được mục đích sử dụng của nó
* Flexible: Tính mở rộng của API
* Unified structure and attribute names: Thống nhất về mặt cấu trúc cho resource cũng như cách đặt tên các thuộc tính
* Clear error message: khi hệ thống xảy ra lỗi gì thì phải có hiển thị thông điệp rõ ràng để tiện cho việc fix bug
* Một số quy tắc thiết kế API
* Sử dụng HTTP method để mô tả về chức năng của resource

Chúng ta có 4 http method cơ bản bảo post, get, put, delete. Với mỗi method sẽ ứng với một chức năng tương ứng của api là CRUD(tạo, đọc, sửa và xóa)

* Sử dụng danh từ số nhiều và không sử dụng động từ
* Liên kết trong resource

1 resource chỉ được liên kết tối đa 2 đối tượng

* Versioning

Được bắt đầu bằng chữ v và kết thúc bằng 1 số nguyên dương, được đặt ở vị trí đầu tiên của resource

* Tìm kiếm

Attribute tên là ‘q’(query)

* Lựa chọn filed trả về

resource có khả năng tuỳ chỉnh những field trả về. Việc này làm tăng performance cho phía client.

* Định dạng dữ liệu trả về

Đối với những resource hỗ trợ nhiều định dạng dữ liệu trả về, HTTP-Header sẽ là nơi để xác định dịnh dạng đó.

* Content-Type: Khai báo request format
* Accept: Khai báo response format
* HTTP status code và error message
* 200 ok
* 201 created
* 202 accepted
* 400 bad request
* 401 unauthorized / 403 forbidden
* 404 not found
* 500 internal server error
* 502 bad gateway

1. Quy chuẩn