Ministry of SMEs and Startups × KOSME Korea SMEs and Startups Agency × LIKELION

RESTful API

Nguyễn Anh Tuấn

K TECH COLLEGE

Nội dung bài giảng

1 RESTful API là gì?

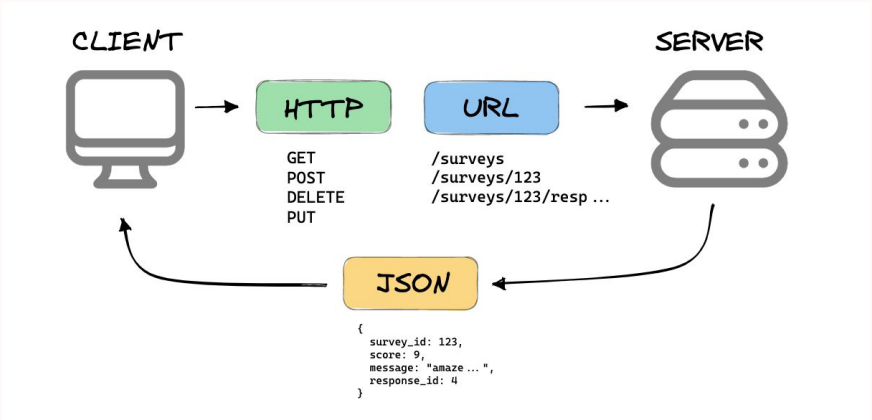
2 RESTful API hoạt động thế nào?

3 Thiết kế REST API URI

RESTful API là gì?

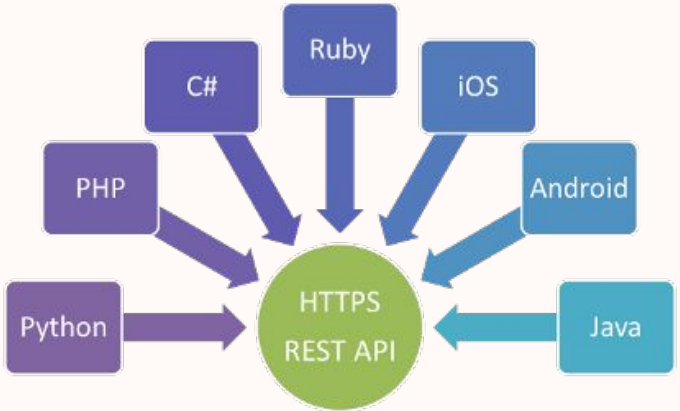
RESTful API là gì?

- **RESTful API** là một giao diện mà hai hệ thống máy tính sử dụng để trao đổi thông tin một cách an toàn qua internet.
- Bản thân **RESTful API** không phải là một loại công nghệ. Nó là phương thức tạo API với nguyên lý tổ chức nhất định.
- **RESTful API** là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.



RESTful API là gì?

- **API (Application Programming Interface)** là 1 tập các quy tắc và cơ chế mà 1 ứng dụng hay 1 thành phần sẽ tương tác với 1 ứng dụng hay thành phần khác. API có thể trả về các kiểu dữ liệu phổ biến như JSON hay XML.
- **REST (REpresentational State Transfer)** là 1 dạng chuyển đổi cấu trúc dữ liệu, 1 kiểu kiến trúc để viết API, sử dụng phương thức HTTP như GET, POST, PUT, DELETE,... để giao tiếp giữa các máy.
- **RESTful API** không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một **RESTful API**.



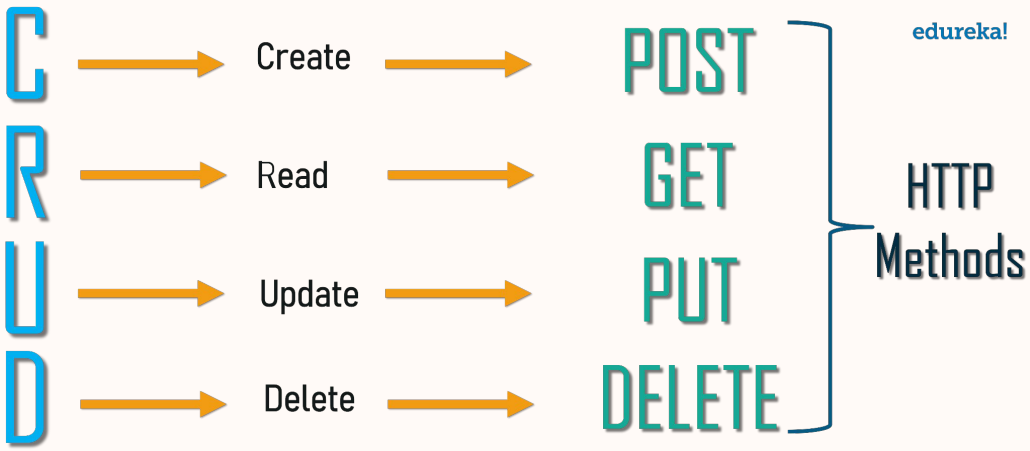
RESTful API hoạt động thế nào?

RESTful API hoạt động thế nào?

- Chức năng cơ bản của **API RESTful** cũng giống như việc duyệt Internet. Client liên hệ với máy chủ bằng cách sử dụng API khi yêu cầu tài nguyên:
  - Client gửi một yêu cầu đến máy chủ. Client làm theo tài liệu API để định dạng yêu cầu theo cách mà máy chủ hiểu được.
  - Máy chủ xác thực và xác nhận máy khách có quyền đưa ra yêu cầu đó.
  - Máy chủ nhận yêu cầu và xử lý trong nội bộ.
  - Máy chủ trả về một phản hồi đến client. Phản hồi chứa thông tin cho client biết liệu yêu cầu có thành công hay không. Phản hồi cũng bao gồm bất kỳ thông tin nào mà client yêu cầu.
  - Chi tiết về phản hồi và yêu cầu API REST sẽ khác nhau đôi chút tùy thuộc vào cách thiết kế API.

RESTful API hoạt động thế nào?

- REST hoạt động chủ yếu dựa vào giao thức HTTP:
  - **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.
- Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.



RESTful API hoạt động thế nào?

- **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. Ví dụ:
    - 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.

RESTful API hoạt động thế nào?

- Nguyên tắc REST yêu cầu phản hồi:
  - **Dòng trạng thái** (HttpStatus)
    - Dòng trạng thái chứa mã trạng thái gồm ba chữ số cho biết yêu cầu thành công hay thất bại. Ví dụ: mã 2XX cho biết đã thành công, nhưng mã 4XX và 5XX cho biết đã xuất hiện lỗi. Mã 3XX cho biết đã chuyển hướng URL.
      - 200: Phản hồi thành công thông thường
      - 201: Phản hồi thành công của phương thức POST
      - 400: Yêu cầu không chính xác khiến máy chủ không thể xử lý
      - 404: Không tìm thấy tài nguyên
  - **Nội dung thông báo**
    - Phần nội dung phản hồi chứa dạng biểu diễn tài nguyên. Client có thể yêu cầu thông tin ở định dạng XML hoặc JSON. Ví dụ 1 dạng biểu diễn JSON:
      - '{"name":"John", "age":30}'

Thiết kế REST API URI

Thiết kế REST API URI

- Ví dụ 1:
  - **POST** http://localhost:8080/create_post
  - **GET** http://localhost:8080/list_posts
  - **POST** http://localhost:8080/feature_posts
  - **POST** http://localhost:8080/edit_post/:post_id

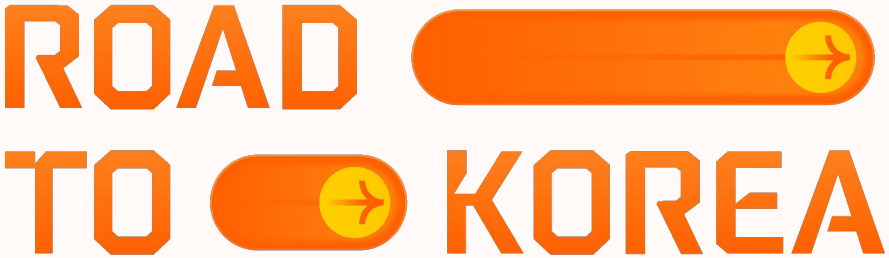
- Ví dụ 2:
  - **POST** http://localhost:8080/api/v1/posts
  - **GET** http://localhost:8080/api/v1/posts
  - **GET** http://localhost:8080/api/v1/posts/:post_id
  - **PUT** http://localhost:8080/api/v1/posts/:post_id
  - **DELETE** http://localhost:8080/api/v1/posts/:post_id

Thiết kế REST API URI

- Có 1 nguyên tắc rất rõ ràng sử dụng các method request để nói lên được nhiệm vụ của API.
- Phần URI có thể giống nhau, không cần cứ phải chứa các động từ như: create, get, update, delete nữa.
- Resource name sẽ ở dạng số nhiều (plural).
  - **GET** /v1/posts/:post_id/liked-users
  - **POST** /v1/posts/:post_id/liked-users
  - **GET** /v1/posts?page=2&limit=50 (sử dụng query string để filter hoặc phân trang)

Thiết kế REST API URI

- **Các quy ước cần chú ý**
  - Sử dụng đúng Status Code. Tránh trả về status 2xx khi trong body là error message.
  - Không dùng underscore (_), hãy dùng hyphen (-) trong URI.
  - Trong URI đều là chữ viết thường (lowercase).
  - Không nên sử dụng đuôi file (extension) trong URI (VD: .html, .xml, .json).



Nếu có bất kỳ thắc mắc nào, hãy đặt câu hỏi qua

mail@mail.com hoặc Zalo 0xxx xxx xxx