

1. RESTful API là gì ?

RESTful API (Representational State Transfer Application Programming Interface) là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế web services) tiện cho các việc quản lý các resource. Chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, dữ liệu động, ...), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP

- API là gì?

API 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 có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.

API giống như một thực đơn ở nhà hàng: bạn chọn món (gửi yêu cầu), nhà bếp nấu (xử lý), và bạn nhận lại món ăn (kết quả). API là cách các phần mềm "nói chuyện" với nhau.

- RESTful là gì?

REST (Representational State Transfer) 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 là một nguyên tắc thiết kế API, giúp việc giao tiếp giữa các hệ thống trở nên đơn giản, hiệu quả và dễ mở rộng. RESTful API tuân thủ các quy tắc của REST.

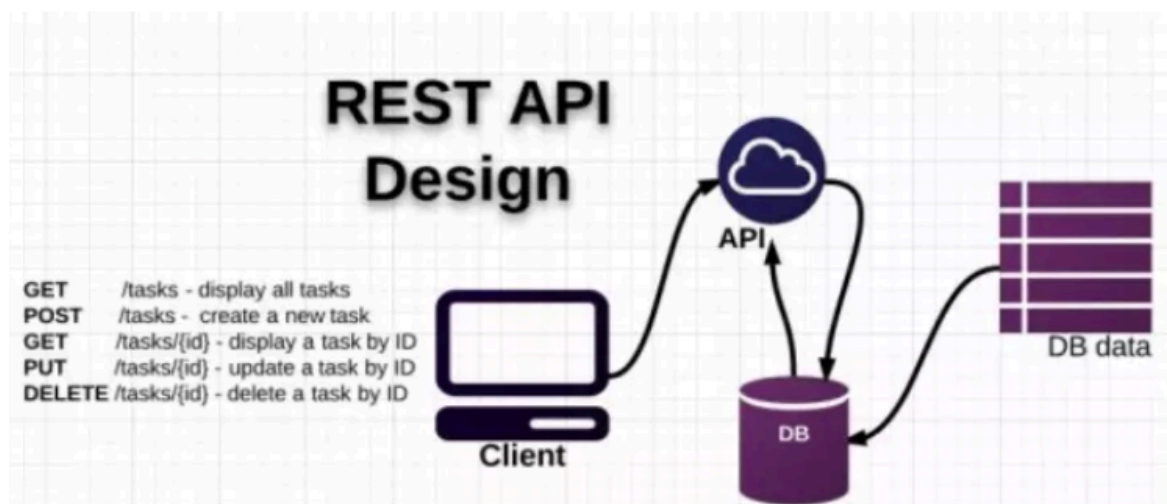
RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile...) khác nhau giao tiếp với nhau.

Thuật ngữ	Định nghĩa đơn giản	Ví dụ
API	Giao thức, quy tắc để phần mềm giao tiếp với nhau	API Google Maps cung cấp dữ liệu bản đồ
REST	Kiến trúc thiết kế API dùng HTTP và các phương thức	Gửi yêu cầu GET, POST đến URL cụ thể để lấy dữ liệu
RESTful API	API thiết kế theo REST, có URL và phương thức rõ ràng	Các endpoint của Google Maps như /map/location, /maps/route

Đặc điểm chính của RESTful API

Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE...) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful 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.

- Sử dụng HTTP: Giao tiếp qua các phương thức HTTP như GET (lấy dữ liệu), POST (tạo mới), PUT (cập nhật), DELETE (xóa).
- Stateless: Mỗi yêu cầu từ client tới server là độc lập, server không lưu trạng thái của client giữa các lần gọi.
- Dữ liệu trả về thường là JSON hoặc XML: Phổ biến nhất là JSON vì dễ đọc và nhẹ.
- Địa chỉ URL rõ ràng: Mỗi tài nguyên (ví dụ: người dùng, sản phẩm) sẽ có một địa chỉ URL riêng biệt.

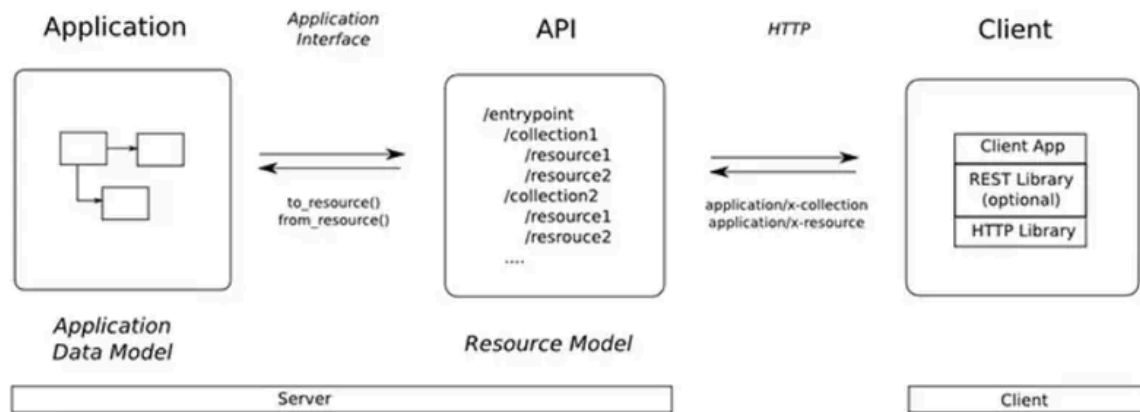


Ví dụ minh họa

Giả sử bạn có một website quản lý sách:

- GET /books: Lấy danh sách tất cả sách.
- GET /books/1: Lấy thông tin sách có ID là 1.
- POST /books: Thêm sách mới.
- PUT /books/1: Cập nhật thông tin sách có ID là 1.
- DELETE /books/1: Xóa sách có ID là 1.

2. RESTful API hoạt động như thế nào ?



REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.

- 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): Xóa 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.

Authentication request và cấu trúc dữ liệu trả về

RESTful API không sử dụng session và cookie, nó sử dụng một *access_token* với mỗi request (jwt). Dữ liệu trả về thường có cấu trúc như sau:

```
{
  "status_code": 200,
  "data": [
    {
      "name": "ManhLD",
      "email": "manhld@example.com",
      "ny": "not found"
    },
    {
      "name": "Ahri",
      "email": "ahriKDA@lmht.com",
      "ny": "Ezreal"
    }
  ]
}
```

```
    }  
  ],  
  error_messages: ""  
}
```

Status code

Khi chúng ta request một API nào đó thường thì sẽ có vài status code để nhận biết sau:

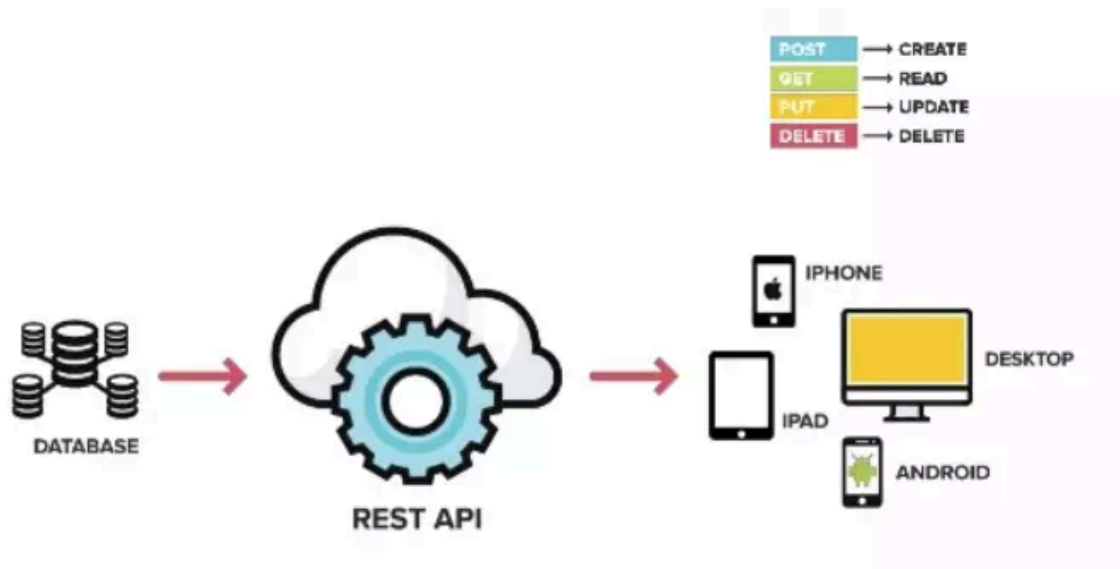
- 200 OK – Trả về thành công cho những phương thức GET, PUT, PATCH hoặc DELETE.
- 201 Created – Trả về khi một Resource vừa được tạo thành công.
- 204 No Content – Trả về khi Resource xoá thành công.
- 304 Not Modified – Client có thể sử dụng dữ liệu cache.
- 400 Bad Request – Request không hợp lệ
- 401 Unauthorized – Request cần có auth.
- 403 Forbidden – bị từ chối không cho phép.
- 404 Not Found – Không tìm thấy resource 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 – Không hỗ trợ kiểu Resource này.
- 422 Unprocessable Entity – Dữ liệu không được xác thực
- 429 Too Many Requests – Request bị từ chối do bị giới hạn

Quản lí version của api

Khi thiết api cho app ios hay client side, chúng ta nên đặt version cho các api. Ví dụ như endpoint sau: api/v1/users

Điều này sẽ giúp hệ thống sau khi nâng cấp lên version mới vẫn hỗ trợ các api của version cũ, cũng như giúp việc bảo trì, sửa chữa dễ dàng hơn.

Ưu điểm của RESTFUL API là gì ?



- Đơn giản, dễ hiểu, dễ sử dụng
- Tăng khả năng mở rộng hệ thống
- Dễ tích hợp với các nền tảng khác nhau (web, mobile, IoT,...)