**REST là gì?**

REST không phải là một **giao thức** (protocol) như HTTP, mà là một **kiến trúc phần mềm** (architectural style) – được mô tả lần đầu bởi **Roy Fielding trong luận văn tiến sĩ năm 2000**.

Nó đưa ra một **tập hợp 6 nguyên tắc (constraints)** mà nếu tuân thủ, một API mới được xem là **RESTful**.

**REST khác HTTP thế nào?**

| **Thuộc tính** | **REST** | **HTTP** |
| --- | --- | --- |
| Loại | Kiến trúc | Giao thức truyền thông |
| Vai trò | Định hướng cách thiết kế API | Cách truyền dữ liệu qua mạng |
| Có thể dùng với | HTTP, FTP, WebSocket,... | Chỉ là 1 giao thức |
| Phổ biến nhất | REST + HTTP | (HTTP là nền tảng cho hầu hết API) |

REST hay dùng HTTP đơn giản vì: **WWW (Web) trước đó đã dùng HTTP**, nên khi rest ra đời, việc dùng HTTP để triển khai RESTful API là **hợp lý và tiện lợi**.

**6 nguyên tắc cơ bản của REST**

| **Nguyên tắc** | **Ý nghĩa** |
| --- | --- |
| 1. **Client-Server** | Client và Server tách biệt, giao tiếp qua API |
| 2. **Stateless** | Mỗi request **độc lập**, không lưu trạng thái ở server |
| 3. **Cacheable** | Phản hồi có thể được cache (HTTP caching) |
| 4. **Uniform Interface** | Interface phải thống nhất: URL, method (GET, POST,...), response,... |
| 5. **Layered System** | Có thể có nhiều tầng (middleware, gateway...) |
| 6. **Code on Demand (tuỳ chọn)** | Gửi mã (JavaScript) từ server cho client thực thi |