**1. REST API**

REST, hay Representational State Transfer, là một kiến trúc phần mềm nhằm hướng dẫn quá trình thiết kế và phát triển kiến trúc trên World Wide Web. Điểm chính của kiến trúc này là tăng cường khả năng mở rộng của giao tiếp giữa các thành phần, cung cấp giao diện thống nhất, đảm bảo sự độc lập triển khai của các thành phần và xây dựng kiến trúc phân lớp để hỗ trợ bộ nhớ đệm, giảm độ trễ đối với người dùng, thực hiện bảo mật và tạo điều kiện thuận lợi cho hệ thống kế thừa.



REST thường sử dụng giao thức HTTP làm cơ sở, trong đó các phương thức HTTP chính được áp dụng. REST API phân chia mỗi giao dịch thành các module nhỏ, mỗi module đảm nhận một phần công việc cụ thể. Quá trình chia nhỏ này mang lại tính linh hoạt cho nhà phát triển, nhưng đồng thời đặt ra những thách thức trong quá trình thiết kế API so với việc sử dụng SOAP, mặc dù REST là stateless.

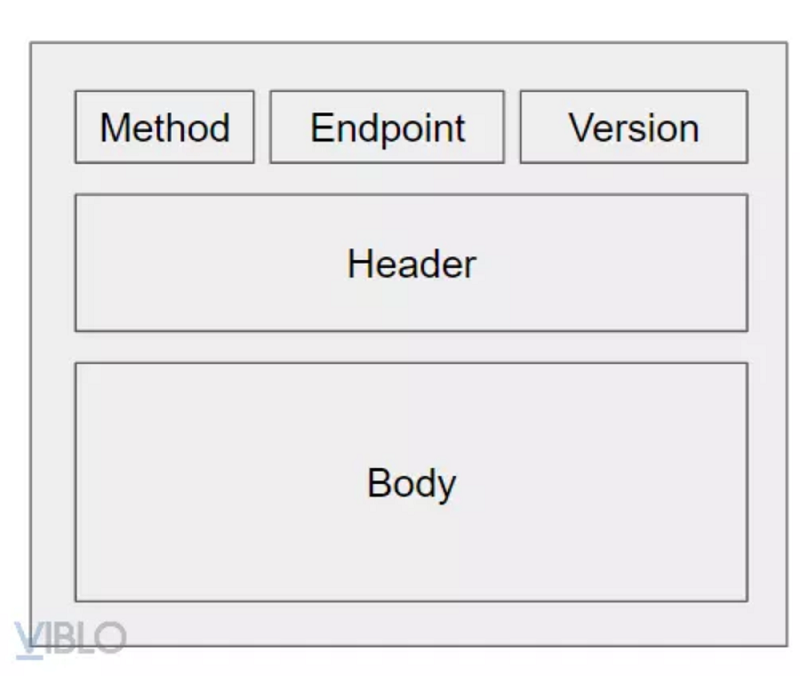
Các phương thức HTTP chính trong REST bao gồm:

* GET (SELECT): Trả về một Resource hoặc danh sách Resource.
* POST (CREATE): Tạo mới một Resource.
* PUT (UPDATE): Cập nhật thông tin cho Resource.
* DELETE (DELETE): Xóa một Resource.

HTTP response status codes được sử dụng để biểu thị kết quả của một yêu cầu HTTP cụ thể và được chia thành năm loại:

* Informational responses (100–199): Thông tin.
* Successful responses (200–299): Thành công.
* Redirection messages (300–399): Chuyển hướng.
* Client error responses (400–499): Lỗi từ phía client.
* Server error responses (500–599): Lỗi từ phía server.

**Cấu trúc thông điệp của Rest**



REST Request bao gồm bốn yếu tố quan trọng:

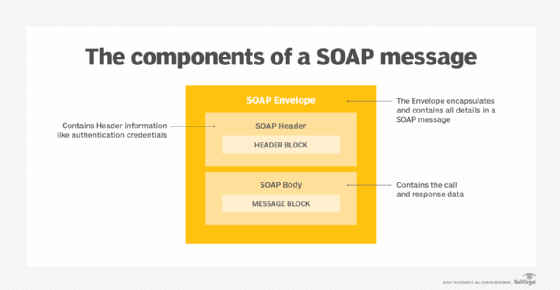
* Method - CRUD (Create, Read, Update, Delete): Xác định hành động cụ thể mà client muốn thực hiện trên resource. Các phương thức phổ biến bao gồm POST (tạo mới), PUT (cập nhật), GET (đọc), DELETE (xóa).
* Endpoint: Là URI (Uniform Resource Identifier) hoặc URL (Uniform Resource Locator) mà client muốn tương tác. Đây là địa chỉ của resource trên server.
* Header: Chứa các thông tin liên quan đến giao tiếp giữa client và server, như tên, trạng thái, thông tin xác thực, địa chỉ IP, và nhiều thông số khác.
* Body: Chứa dữ liệu được gửi đi kèm với request, thường được sử dụng trong các phương thức như POST và PUT. Dữ liệu này có thể ở định dạng XML, JSON hoặc các định dạng khác do client và server thỏa thuận.

REST Response từ server thường được biểu diễn trong các định dạng như XML hoặc JSON.

Ưu điểm của REST bao gồm khả năng tương thích tốt với các công nghệ web mới, dễ sử dụng và bảo trì. Nó tách biệt rõ ràng giữa server và client, chỉ sử dụng các phương thức HTTP và URI. Thêm vào đó, thông tin có thể được lưu lại phía client, giảm số lần phải truy cập resource.

Tuy nhiên, REST chỉ hoạt động trên giao thức HTTP và có thể không cung cấp mức độ bảo mật và xác thực như SOAP (stateless).

**2. SOAP (Simple Object Access Protocol)** là một chuẩn giao thức nhắn tin được sử dụng để trao đổi thông tin có cấu trúc trong việc triển khai các dịch vụ web trên mạng máy tính. Giao thức này sử dụng định dạng XML cho các thông báo và thường dựa trên các giao thức lớp ứng dụng như HTTP hoặc SMTP để đàm phán và truyền thông.



SOAP có khả năng mở rộng, trong đó bảo mật và WS-Addressing là một số phần mở rộng đang được phát triển. Nó cũng thể hiện tính trung lập, có thể hoạt động trên nhiều loại giao thức như HTTP, SMTP, TCP, và UDP. Điều này giúp SOAP độc lập, cho phép tích hợp với bất kỳ mô hình lập trình nào.

Cấu trúc chính của một thông điệp SOAP bao gồm SOAP Envelope để xác định tài liệu XML, SOAP Header chứa thông tin tiêu đề, SOAP Body chứa nội dung của cuộc gọi và phản hồi, và có thể có SOAP Fault để cung cấp thông tin về lỗi xảy ra.

Ưu điểm của SOAP bao gồm tính trung lập ngôn ngữ, khả năng tương tác và độc lập nền tảng, tính đơn giản với định dạng XML, và khả năng mở rộng thông qua sự sử dụng giao thức HTTP.

Tuy nhiên, SOAP cũng có nhược điểm, bao gồm tốc độ chậm do định dạng XML cần phải được phân tích cú pháp, và sự phụ thuộc vào WSDL mà không có cơ chế tiêu chuẩn hóa để khám phá động các dịch vụ.