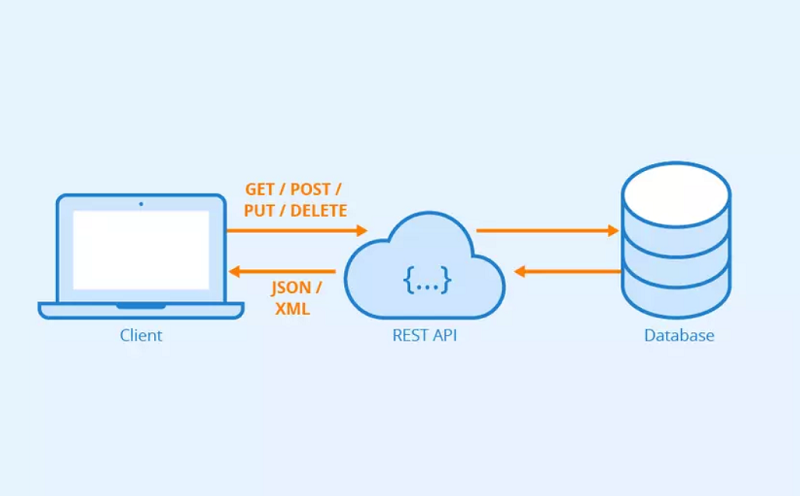
**1. REST API**

REST (REpresentational State Transfer) là một kiến trúc phần mềm được tạo ra để hướng dẫn thiết kế và phát triển kiến trúc cho World Wide Web. Kiến trúc REST nhấn mạnh khả năng mở rộng của các tương tác giữa các thành phần, giao diện thống nhất, triển khai độc lập của các thành phần và tạo ra kiến trúc phân lớp để tạo điều kiện thuận lợi cho các thành phần bộ nhớ đệm nhằm giảm độ trễ do người dùng nhận thấy, thực thi bảo mật và đóng gói các hệ thống kế thừa.



REST thường hoạt động chủ yếu dựa vào những giao thức HTTP, các cơ sở hoạt động cơ bản nêu trên sẽ được sử dụng những phương thức HTTP. REST API chia nhỏ một transaction ra thành nhiều module nhỏ khác nhau. Mỗi một module sẽ giải quyết một phần công việc trong transaction đó. Việc chia nhỏ thành các module này giúp các nhà phát triển có thể linh hoạt xử lý hơn. Tuy nhiên, việc chia nhỏ này cũng tạo ra không ít thách thức trong quá trình thiết kế REST API từ đầu đều không đảm bảo bằng SOAP (stateless).

Các HTTP method:

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.

**HTTP response status codes**: HTTP response status codes cho biết một yêu cầu HTTP cụ thể đã được hoàn tất thành công hay chưa. Các câu trả lời được nhóm thành năm lớp:

**Informational** responses (100–199)

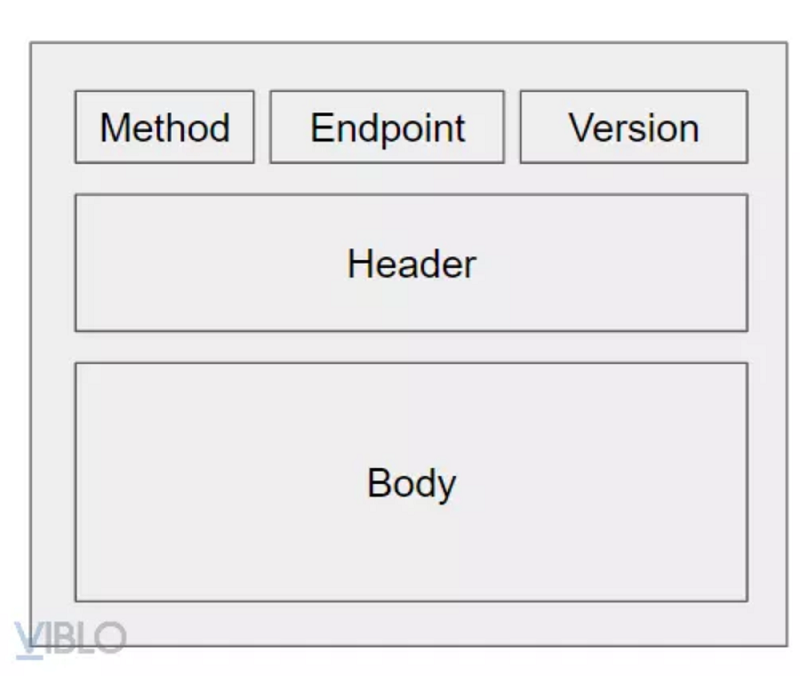
**Successful** responses (200–299)

**Redirection** messages (300–399)

**Client error** responses (400–499)

**Server error** responses (500–599)

**Cấu trúc thông điệp của Rest**



**Rest Request** gồm 4 thành phần chính:

* Method - CRUD : POST, PUT, GET, DELETE
* Endpoint - Chứa URI ( phổ biến nhất là URL )
* Header - Chứa thông tin liên quan đến client và server như tên, trạng thái, authentication, ip, …
* Body - Chứa dữ liệu

**Rest Response** - Server gửi đại diện dữ liệu thường được viết ở dạng XML, JSON, …

**Ưu điểm**

REST thích ứng tốt với các công nghệ web mới, dễ dàng sử dụng và bảo trì. Tách biệt rõ ràng giữa server và client, chỉ giao tiếp thông qua các phương thức HTTP và URI (đọc lại bài viết REST 101). Các thông tin có thể được lưu lại phía client → tránh phải gọi đến resource nhiều lần. Có thể sử dụng bất kỳ cấu trúc nào (XML, JSON, hoặc cấu trúc do server và client quy ước với nhau).

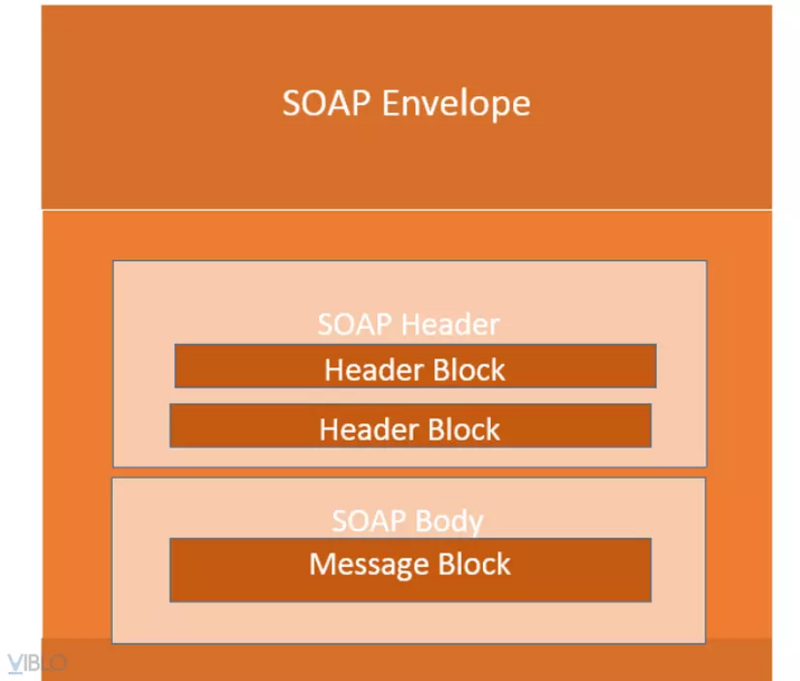
**Nhược điểm**

Chỉ hoạt động trên các giao thức HTTP. Việc bảo mật và xác thực có thể không đảm bảo bằng SOAP (stateless).

**2. SOAP**

SOAP (Simple Object Access Protocol): Là một đặc tả giao thức nhắn tin để trao đổi thông tin có cấu trúc trong việc triển khai các dịch vụ web trong mạng máy tính. Nó sử dụng Tập thông tin XML cho định dạng thông báo và dựa trên các giao thức lớp ứng dụng, thường là Giao thức truyền siêu văn bản (HTTP) hoặc Giao thức truyền thư đơngiản (SMTP), để đàm phán và truyền thông.

Khả năng mở rộng (bảo mật và WS-Addressing là một trong những phần mở rộng đang được phát triển). Tính trung lập (SOAP có thể hoạt động trên bất kỳ giao thức nào như HTTP, SMTP , TCP , UDP). Độc lập (SOAP cho phép bất kỳ mô hình lập trình nào).



SOAP Envelope: Xác định tài liệu XML dưới dạng thông báo SOAP. SOAP Header: Chứa thông tin tiêu đề. SOAP Body: Chứa thông tin cuộc gọi và phản hồi. SOAP Fault: Cung cấp thông tin về các lỗi đã xảy ra trong khi xử lý thư.(Không cần thiết)

**Ưu điểm:**

* Tính trung lập của ngôn ngữ: SOAP có thể được phát triển bằng bất kỳ ngôn ngữ nào.
* Khả năng tương tác và tính độc lập của nền tảng: SOAP có thể được triển khai bằng bất kỳ ngôn ngữ nào và có thể được thực thi trong bất kỳ nền tảng nào.
* Tính đơn giản: Thông báo SOAP có định dạng XML rất đơn giản.
* Khả năng mở rộng: SOAP sử dụng giao thức HTTP để truyền tải do đó nó trở nên có khả năng mở rộng.

**Nhược điểm:**

Chậm: SOAP sử dụng định dạng XML cần được phân tích cú pháp và cũng dài hơn. Sự phụ thuộc của WSDL: Nó phụ thuộc vào Web Services Description Language và không có bất kỳ cơ chế tiêu chuẩn hóa nào để khám phá động các dịch vụ.