**Lập Trình Web**

**Lab 10: Tìm hiểu về Webservice**

**Bài 1**

1. REST

- **Định nghĩa**:

* + REST (REpresentational State Transfer) được đưa ra vào năm 2000, trong luận văn tiến sĩ của Roy Thomas Fielding (đồng sáng lập giao thức HTTP). Nó là một dạng chuyển đổi cấu trúc dữ liệu, là một phong cách kiến ​​trúc cho việc thiết kế các ứng dụng có kết nối. Nó sử dụng 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.
    - API (Application Programming Interface) là giao diện lập trình ứng dụng giúp tạo ra các phương thức kết nối với các thư viện và ứng dụng khác nhau.
    - REST API là một ứng dụng chuyển đổi cấu trúc dữ liệu có các phương thức để kết nối với các thư viện và ứng dụng khác. REST API không được xem là một công nghệ, nó là một giải pháp để tạo ra các ứng dụng web services thay thế cho các kiểu khác như SOAP, WSDL (Web Service Definition Language),…

- **Ưu điểm**:

* + **Hiệu năng**: Các thành phần đảm bảo được việc giao tiếp theo đúng một quy ước giúp hệ thống có thể vận hành tốt hơn.
  + **Tính khả biến**: Với hệ thống cần thay đổi các tài nguyên liên tục, sử dụng REST với việc tạo request đơn giản sẽ giúp mọi việc chuyển trở nên đơn giản hơn.
  + **Tính mở rộng**: Các hệ thống REST có khả năng mở rộng rất cao nhờ sự tách biệt giữa các thành phần và các quy ước giao tiếp được quy định sẵn.
  + **Tính linh hoạt**: Việc chuẩn hóa interface giúp hệ thống trở nên linh hoạt hơn, có thể sử dụng cho nhiều nền tảng khác nhau, mobile, web, …
  + **Trong sáng**: Trong giao tiếp giữa các thành phần, các request trở nên rất rõ ràng, dễ hiểu.
  + **Đơn giản**: Xây dựng rất đơn giản, ví dụ việc [khai báo resource trong rails](https://guides.rubyonrails.org/routing.html) đã giúp xây dựng các uri cho resource.
  + **Tính tin cậy**: Khó để xảy ra lỗi trong giao tiếp giữa các thành phần gây sụp đổ hệ thống.

**- Nhược điểm**:

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

1. SOAP

- **Định nghĩa**:

* SOAP (Simple Object Access Protocol) là một giao thức do W3C định nghĩa. SOAP áp dụng XML để xác định dữ liệu dạng văn bản (plain text) qua HTTP và SMTP. Web Service dùng SOAP trong quá trình truyền tải dữ liệu. SOAP không phụ thuộc vào ngôn ngữ lập trình hay bất kỳ nền tảng nào vì nó dùng XML.

**- Ưu điểm**:

* + Không phụ thuộc vào ngôn ngữ , nền tảng, hay phương thức truyền tải.
  + Hoạt động tốt trong môi trường doanh nghiệp phân tán lớn, phức tạp, đòi hỏi tình bảo mật cao:
    - Dịch vụ tài chính (Paypal API)
    - Thanh toán trực tuyến (Saleforce API)
    - Viễn thông (Clickatell SMS API)
  + Được tiêu chuẩn hóa
  + Cung cấp khả năng mở rộng đáng kể trước khi xây dựng dưới dạng các tiêu chuẩn Web Service.
  + Tự động hóa khi được sử dụng với các sản phẩm ngôn ngữ nhất định
* Nhược điểm
  + Việc duy trì trạng thái (stateful) khiến tiêu tốn tài nguyên cho các metadata.
  + Khó sử dụng và không phổ biến bằng REST trong các ứng dụng web hay mobile thông thường.
  + Chỉ sử dụng XML