

# 5

## Lab

# AJAX & REST API

**Thực hành Lập trình Ứng dụng Mạng**  
**Lớp NT109**

GVHD: Đỗ Thị Hương Lan

Học kỳ II – Năm học 2020-2021

**Lưu hành nội bộ**

## 1. TỔNG QUAN

### 1. Mục tiêu

- Tìm hiểu khái niệm, cơ chế hoạt động, lợi ích của AJAX.
- Tìm hiểu khái niệm, cơ chế hoạt động, lợi ích của REST API.
- Cách xây dựng một ứng dụng sử dụng AJAX và REST API.
- Lập trình một số tính năng kết hợp AJAX và REST API.

### 2. Môi trường

- Ngôn ngữ xử lý server và client: tự chọn
- Database: tự chọn.
- IDE: phù hợp với ngôn ngữ backend, frontend lựa chọn, ...

### 3. Kiến thức liên quan

- Sinh viên cần nắm kiến thức nền tảng về xử lý server side.
- Có kiến thức cơ bản về xử lý client side.
- Nắm được 1 ngôn ngữ lập trình xử lý server side và client side.
- Tham khảo tại w3school.com

## B. KIẾN THỨC NỀN TẢNG

### 1) AJAX

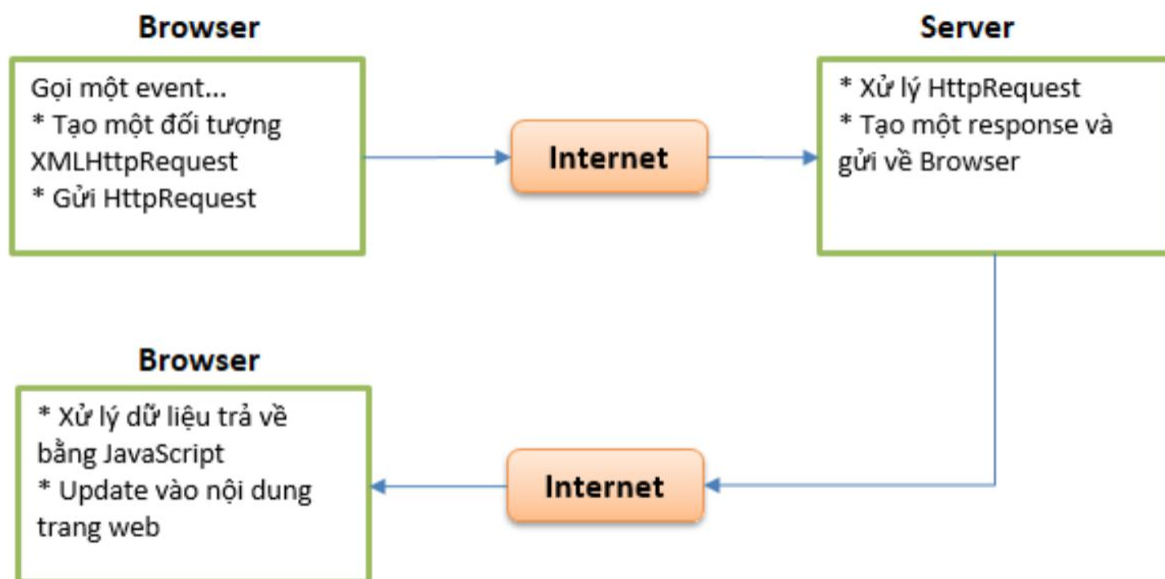
#### a) Khái niệm

- Ajax là viết tắt của cụm từ "Asynchronous JavaScript And XML".
- Ajax không phải là một ngôn ngữ lập trình hay công cụ.



- Ajax là **phương thức trao đổi dữ liệu với máy chủ và cập nhật các phần của trang web - mà không cần tải lại toàn bộ trang.**
- Ajax chính nó là một thuật ngữ chung cho các kỹ thuật JavaScript khác nhau được sử dụng để kết nối với máy chủ web tự động mà không nhất thiết phải tải lại trang. Về mặt kỹ thuật, nó đề cập đến việc sử dụng các đối tượng XMLHttpRequest để tương tác với một máy chủ web động thông qua JavaScript.

### b) Cơ chế hoạt động



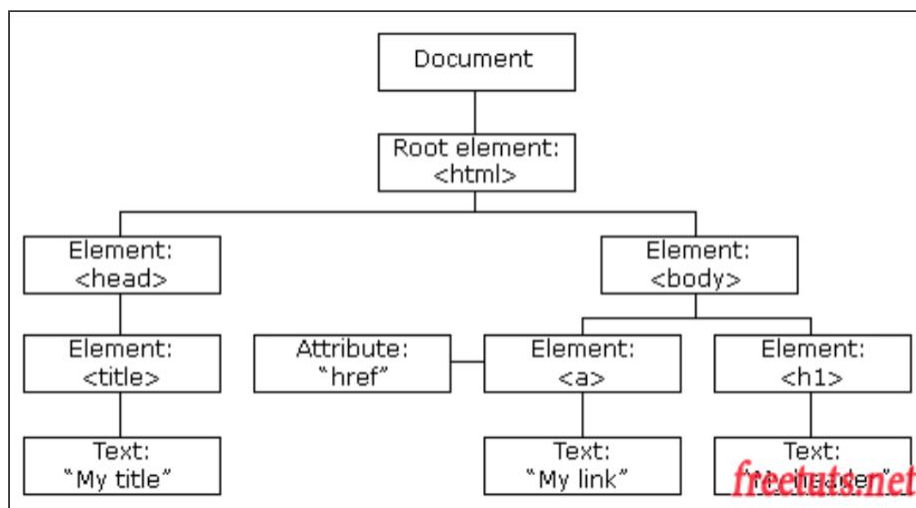
### c) Lợi ích

- **Callbacks:** Ajax được sử dụng để thực hiện một cuộc gọi lại, thực hiện việc truy xuất và / hoặc lưu dữ liệu mà không gửi toàn bộ trang trở lại máy chủ. Bằng cách gửi lại một phần trang web đến máy chủ, việc sử dụng mạng được giảm thiểu và các hoạt động diễn ra nhanh hơn. Trong các trang web bằng thông hạn chế, điều này có thể cải thiện đáng kể hiệu suất mạng. Dữ liệu được gửi đến và đi từ máy chủ một cách tối thiểu.
- **Thực hiện các cuộc gọi không đồng bộ:** Ajax cho phép bạn thực hiện các cuộc gọi không đồng bộ đến một máy chủ web. Điều này cho phép trình duyệt của người dùng tránh phải chờ tất cả dữ liệu đến trước khi cho phép người dùng hành động một lần nữa.

- **Thân thiện với người dùng:** Vì không phải post lại trang lên server, các ứng dụng hỗ trợ Ajax sẽ luôn nhanh hơn và thân thiện với người dùng hơn.
- **Tăng tốc độ:** Mục đích chính của Ajax là cải thiện tốc độ, hiệu suất và khả năng sử dụng của một ứng dụng web. Một ví dụ tuyệt vời của Ajax là tính năng xếp hạng phim trên Netflix. Người dùng đánh giá một bộ phim và xếp hạng cá nhân của họ cho bộ phim đó sẽ được lưu vào cơ sở dữ liệu của họ mà không cần chờ trang làm mới hoặc tải lại.

#### d) Các công nghệ AJAX

- AJAX không phải là một công nghệ mà là **một nhóm các công nghệ liên quan đến nhau**. Các công nghệ AJAX bao gồm:
  - HTML/XHTML và CSS
  - DOM
  - XML và JSON
  - XMLHttpRequest
  - JavaScript



#### e) Đối tượng XMLHttpRequest trong AJAX

- Đối tượng *XMLHttpRequest* (viết tắt là *XHR*) là một API có thể được sử dụng bởi JavaScript, Jscript, VBScript và một số ngôn ngữ phát triển ứng dụng Web khác.
- Đối tượng *XMLHttpRequest* trong AJAX được sử dụng cho việc giao tiếp không đồng bộ giữa máy khách và máy chủ.
- Nó thực hiện các hoạt động sau:
  - Gửi dữ liệu từ máy khách đến máy chủ dưới nền (background).
  - Nhận dữ liệu từ máy chủ.
  - Cập nhật trang web mà không cần tải lại trang web.
- Tạo đối tượng *XMLHttpRequest* trong AJAX

```
1 <script>
2   request = new XMLHttpRequest();
3 </script>
```

```
<!DOCTYPE html>
<html>
<body>

<h2>The XMLHttpRequest Object</h2>

<p id="demo">Let AJAX change this text.</p>

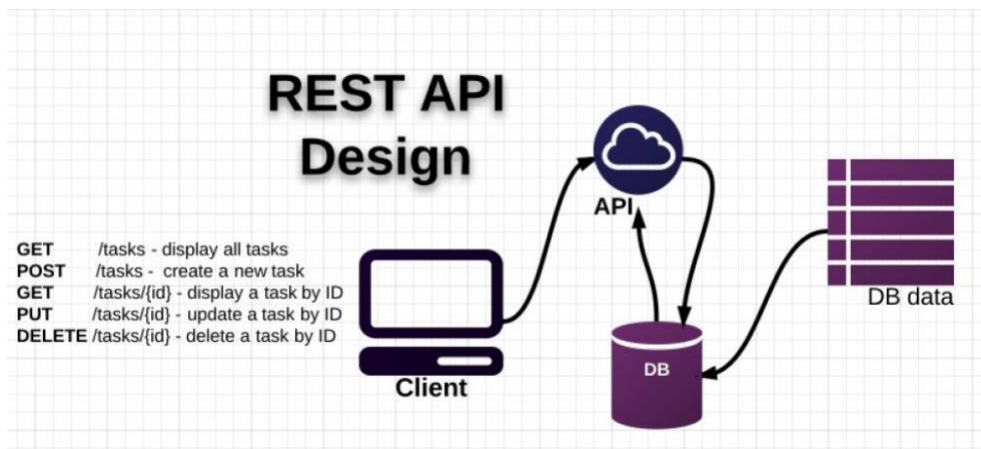
<button type="button" onclick="loadDoc()">Change Content</button>

<script>
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML = this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
</script>

</body>
</html>
```

## 2. REST API

### a) Khái niệm



- API (Application Programming Interface) 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.

- REST (**RE**presentational **S**tate **T**ransfer) 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.

- RESTful API là một tiêu chuẩn (giải pháp) 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.

- 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.

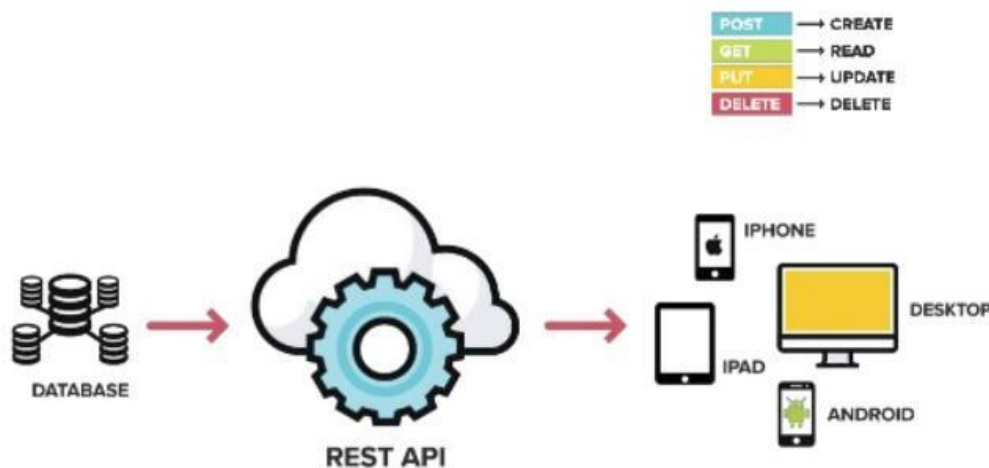
### b) Resource

- Quản lý resource (tài nguyên) là một phần quan trọng và chiếm phần lớn trong việc phát triển website. Trong đó resource của các website khác nhau có thể sẽ khác nhau. Với các trang mạng xã hội như Facebook thì resource thường là danh sách người dùng (user hoặc account), danh sách các bài viết (post hoặc article), các ảnh được đăng (photo hoặc image), các trang fanpage (fanpage)... Đối với một trang chia sẻ ảnh như Instagram thì resource có thể là các ảnh được đăng (photo), danh sách người dùng (user)... Đối với các trang bán hàng thì resource có thể là danh sách sản phẩm (product), danh sách các người bán (seller), danh sách khách hàng (user hay customer)...

- Việc quản lý resource của một website bao gồm 4 tác vụ chính:
  - Tạo mới một resource (create)
  - Lấy thông tin một resource (read)
  - Cập nhật một resource (update)
  - Xóa một resource (delete)
- Có rất nhiều cách khác nhau để xây dựng một trang web thực hiện 4 tác vụ trên. Ví dụ với một trang blog chạy dưới tên miền là `http://my-blog.xyz` thì để xem nội dung một bài viết (post) với ID là 123 bạn có thể làm theo một trong các cách sau:
  - Gửi một request tới URL `http://my-blog.xyz/posts?id=123` với HTTP method là GET
  - Gửi một request tới URL `http://my-blog.xyz/posts/123` với HTTP method là GET
  - Gửi một request tới URL `http://my-blog.xyz/action=view_post&id=123` với HTTP method là GET
  - Gửi một request tới URL `http://my-blog.xyz/view_post&id=123` với HTTP method là GET
  - Gửi một request tới URL `http://my-blog.xyz/posts?id=123` với HTTP method là POST

- Gửi một request tới URL `http://my-blog.xyz/posts/123` với HTTP method là POST
- ...
- Và sau một thời gian dài thì người ta đã thống nhất ra các tiêu chuẩn khác nhau để thực hiện việc quản lý resource. Các tiêu chuẩn này (hay còn được gọi là Web API hoặc HTTP API) quy định một cách thống nhất việc quản lý các resource của web. Và RESTful là một trong các web API được sử dụng phổ biến ngày nay.
- Phân lớp hệ thống : trong hệ thống REST chia tách các thành phần hệ thống theo từng lớp, mỗi lớp chỉ sử dụng lớp ở dưới nó và giao tiếp với lớp ở ngay trên nó mà thôi. Điều này giúp giảm độ phức tạp của hệ thống, giúp các thành phần tách biệt nhau từ đó dễ dàng mở rộng từng thành phần.

### c) Ưu điểm



- Giúp cho ứng dụng rõ ràng hơn
- REST URL đại diện cho resource chứ không phải hành động
- Dữ liệu được trả về với nhiều định dạng khác nhau như: xml, html, json....
- Code đơn giản và ngắn gọn.
- REST chú trọng vào tài nguyên của hệ thống



## C. THỰC HÀNH

**Mục tiêu:** Xây dựng API cho hệ thống và cài đặt một số chức năng sử dụng AJAX hỗ trợ tải bộ phận

### **Yêu cầu 1: Thiết kế Server Resources chuẩn RESTFul API.**

- Tạo mã nguồn cho API (có thể nằm chung với hệ thống đang chạy)
- Data gửi nhận dạng JSON
- Sử dụng các công cụ (Curl, Wget, Postman, ...) để kiểm tra hoạt động của các RESTFul API.

### **Yêu cầu 2: Sử dụng Ajax để tương tác với Server thông qua API, Web page**

- Sử dụng JavaScript thuần hoặc thư viện JQuery để gửi nhận request.
  - Gửi/Lấy dữ liệu và hiển thị.
  - Có thể lựa chọn 3 trong các chức năng sau để triển khai (tùy vào Dự án)
    - + Thêm sản phẩm vào giỏ hàng, cập nhật giỏ hàng
    - + Like, đánh giá, bình chọn sản phẩm, bài viết, đối tượng ...
    - + Kiểm tra sự tồn tại của tài khoản hỗ trợ việc đăng ký tài khoản
    - + Xử lý Form Đăng nhập ở đầu trang
    - + Bình luận
    - + Tải thêm bài viết, sản phẩm (Load more)
    - + Gửi feedback
    - + Gợi ý tỉnh thành, quận huyện, xã cho phần địa chỉ
    - + Gợi ý tìm kiếm (Liệt kê danh sách gợi ý khi search)
    - + Thêm/Xóa/Sửa/Cập nhật bài viết, sản phẩm, nội dung ở trang quản lý
- ....

## D. YÊU CẦU & ĐÁNH GIÁ

### 1. Yêu cầu

- Sinh viên thực hành và nộp bài theo **nhóm** tại website môn học theo thời gian quy định
- Bài nộp bao gồm:
  - Toàn bộ **Source-code** của các bài tập liên quan
  - File PDF chứa thông tin nhóm, hình ảnh screenshot demo hoặc Video Demo (Không nén)

Toàn bộ project đặt vào 1 file nén (.rar/.zip) với tên theo quy tắc sau:

**Lab3-MSSV1-MSSV2-MSSV3 ...**

Ví dụ: *Lab3-16520067-17520010-18521000*

### 2. Đánh giá kết quả

Tiêu chí	Chuyên cần (A)	Điểm bài lab (B)	Kết quả
Điểm	Có mặt = 10 Vắng = 7	Tối đa = 10 Không nộp bài = 0	<b>Điểm = (A + B)/2</b>
Lưu ý	<b>Vắng + Không nộp bài : 0</b>		

**HẾT**