# Khái niệm Ajax

AJAX, là viết tắt của Asynchronous JavaScript and XML, (tạm dịch là JavaScript và XML không đồng bộ), không phải là ngôn ngữ lập trình.

Ajax chỉ sử dụng kết hợp

* Một đối tượng XMLHttpRequest được tích hợp sẵn của trình duyệt (để yêu cầu dữ liệu từ một máy chủ web)
* JavaScript và HTML DOM (để hiển thị hoặc sử dụng dữ liệu)

Vậy khi làm việc với Ajax chúng ta cần hiểu được hai phần quan trọng là XMLHttpRequest và JavaScript, HTML

Các ứng dụng AJAX có thể sử dụng để truyền tải dữ liệu XML, văn bản thuần hoặc văn bản JSON.

AJAX cho phép các website được cập nhật không đồng bộ bằng cách trao đổi dữ liệu với web server.

# Mục đích Ajax

Ajax cho phép:

* Update một phần web page mà không cần tải lại trang.
* Tạo request tới server khi mà trang đã tải xong
* Nhận dữ liệu từ server khi mà trang đã tải xong
* Gửi dữ liệu tới server đằng sau phía giao diện.

Như vậy khi ta sử dụng Ajax ta sẽ tạo một request và nhận lại response. Sử dụng response để thay đổi hiển thị mà không phải tải lại trang. Đó là mục đích của lớn nhất của Ajax

# Hoạt đông của Ajax

Sau đây là hình ảnh mô tả hoạt động của Ajax:



Nguồn ảnh: www.w3schools.com

Sau đây là mô tả hoạt động của Ajax như sau:

1. Người dùng thao tác với trình duyệt. Click button
2. Một đối tượng XMLHttpRequest được tạo bởi JavaScript
3. Gửi đi XMLHttpRequest tới máy chủ web qua internet
4. Máy chủ xử lý
5. Máy chủ gửi lại response về cho web page trên máy người dùng
6. Response đọc từ JavaScript
7. Web page được update bởi JavaScript từ kết quả đọc được ở bước 6

# Thành phần trong AJAX

## XMLHttpRequest

Tất cả trình duyệt đềuhỗ trợ XMLHttpRequest.

Một đối tượng XMLHttpRequest giúp trao đổi dữ liệu với máy chủ web phía sau giao diện.

Có nghĩa là không cần tải lại trang web mà có thể thay đổi nội dung một phần trang web bằng cách dùng XMLHttpRequest.

### Các phương thức của XMLHttpRequest

Sau đây là các phương thức khi ta sử dụng XMLHttpRequest

|  |  |
| --- | --- |
| Method | Mô tả |
| new XMLHttpRequest() | Tạo một đối tượng XMLHttpRequest() |
| abort() | Hủy bỏ request hiện tại |
| getAllResponseHeaders | Trả về thông tin header |
| getResponseHeader() | Trả thông tin riêng của header |
| open(method,url,async,user,psw) | Khởi tạo một request với các thông số  method: GET hoặc POST  url địa chỉ file  async: = true là không đồng bộ, false là đồng bộ  user: username  psw: mật khẩu |
| send() | Sử dụng cho method là GET  Gửi request tới server |
| send(string) | Sử dụng cho method là POST  Gửi request tới server |
|  | Thêm một cặp label-value vào header khi gửi. |

### Các thuộc tính của đối tượng XMLHttpRequest

Các thuôc tính của đối tượng XMLHttpRequest như sau:

|  |  |
| --- | --- |
| Property | Mô tả |
| onreadystatechange | Định nghĩa một method sẽ được gọi khi “readyState” thay đổi |
| readyState | Giữ trạng thái hiện tại của XMLHttpRequest.  0: request không được khởi tạo  1: kết nối máy chủ chưa được thiết lập  2: đã nhận request  3: request đang được xử lý  4: yêu cầu kết thúc và response đã có |
| responseText | Dữ liệu response trả về ở dạng string |
| responseXML | Dữ liệu response trả về ở dạng XML |
| status | Trả về status-number của request  200: "OK"  403: "Forbidden"  404: "Not Found"  …………  Đây là các thông số của Http Messages. |
| statusText | Trả về status-text (ví dụ "OK" or "Not Found") |

## Một số kiểu dữ liệu trong AJAX

### XML

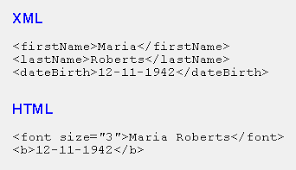
Xml là một ngôn ngữ đánh dấu có cấu trúc giống HTML có cấu trúc đóng mở thẻ

Ngoài ra XML còn có thể tạo ra các ngôn ngữ đánh dấu khác.

XML là một kiểu dữ liệu truyền trên internet mà chúng ta sử dụng phổ biến hiện nay.

Ajax là Asynchronous JavaScript and XML nghe có vẻ Ajax chỉ làm việc với XML. Tên gây ra nhầm lẫn rằng Ajax chi sử dụng với XML nhưng Ajax không phải như vậy nó còn làm việc với Json, Text,..

Bạn không cần phải hiểu và nắm rõ XML vẫn có thể làm việc với Ajax. Tuy nhiên làm tốt và hiểu hơn các bạn nên có hiểu biết một chút về XML



Cấu trúc của HTML khá XML là rất giống nhau

Ví dụ về XML

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8" ?>  <employees>  <employee>  <firstName>John</firstName> <lastName>Doe</lastName>  </employee>  <employee>  <firstName>Anna</firstName> <lastName>Smith</lastName>  </employee>  </employees> |

Cấu trúc XML như sau:

* <?xml version="1.0" encoding="UTF-8"?> là dòng head của file
* <persons> được gọi là root-node
* Nội dung trong root-node là từng node trong file

### JSON

JSON có cấu trúc theo dạng key-value.

Ví dụ về JSON

|  |
| --- |
| {  "employees": {  "employee": [  {  "firstName": "John",  "lastName": "Doe"  },  {  "firstName": "Anna",  "lastName": "Smith"  }  ]  }  } |

Dữ liệu mảng json trên và mảng XML ở trên là giống nhau

Đây là link chuyển đổi giữa XML và JSON:

<http://www.utilities-online.info/xmltojson/#.WvvtGIiFPDc>

Về cơ bản JSON và XML là hai kiểu dữ liệu phổ biển để mô tả dữ liệu

Tuy vậy có những ưu nhược điểm và được so sánh mục sau

### So sánh JSON và XML với Ajax

Giống nhau JSON và Ajax như sau:

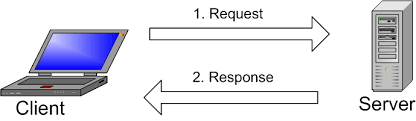
* đều có thể được sử dụng để nhận dữ liệu từ máy chủ web
* Dùng mô tả dữ liệu mà mắt người đều đọc được
* Có các cấp bậc dữ liệu
* Các ngôn ngữ lập trình đều hỗ trợ
* Đều có thể gắn với XML HttpRequest

Sự khác nhau giữ JSON và XML

|  |  |  |
| --- | --- | --- |
| Đặc điểm | JSON | XML |
| Tốc độ | Nhanh hơn do JSON được hỗ trợ bởi JS | Chậm hơn do khó phân tích hơn |
| Cách sử dụng | Sử dụng JSON.parse() để tạo ra đối tượng để sử dụng | Sử dụng XML DOM  để phân tích thành đối tượng |
| Cấu trúc | Ngắn hơn  Đọc viết nhanh hơn | Dài hơn  Đọc và viết phức tạp |
| Phân tích | Phân tích đơn giản | Phân tích khó khăn hầu như các ngôn ngữ đều phải xây dựng một bộ phân tích. |

# Kết hợp AJAX với các ngôn ngữ lập trình

Với mô hình Client và Server như sau:



Phía Server phải sử dụng những ngôn ngữ lập trình để:

* Xử lý yêu cầu người dùng
* Nhận yêu cầu từ XML HTTP Request
* Gửi về dữ liệu Response cho Ajax bằng dữ liệu XML, JSON, Text..

Để như vậy, thường các ngôn ngữ lập trình sẽ xây dựng những phương thức để tạo ra JSON, XML, Text,..tương ứng với request.

Phía Client sử dụng Ajax như sau:

* Xây dựng mô hình giao diện tương ứng với cấu trúc JSON, XML để hiển thị cho người dùng
* Xây dựng các kịch bản sử dụng Ajax