

AJAX - Tương lai của ứng dụng Web

AJAX là gì ?

AJAX (tiếng Anh), viết tắt cho "Asynchronous JavaScript and XML", nghĩa là "JavaScript và XML không đồng bộ". Từ *Ajax* được ông Jesse James Garrett tạo ra và dùng lần đầu tiên vào tháng 2 năm 2005 để chỉ kỹ thuật này, mặc dù các hỗ trợ cho *Ajax* đã có trên các chương trình duyệt từ 10 năm trước. Ajax là một kỹ thuật phát triển web có tính tương tác cao bằng cách kết hợp các ngôn ngữ:

- HTML (hoặc XHTML) với CSS trong việc hiển thị thông tin
- Mô hình DOM (Document Object Model), được thực hiện thông qua JavaScript, nhằm hiển thị thông tin động và tương tác với những thông tin được hiển thị
- Đối tượng XMLHttpRequest để trao đổi dữ liệu một cách không đồng bộ với máy chủ web. (Mặc dù, việc trao đổi này có thể được thực hiện với nhiều định dạng như HTML, văn bản thường, JSON và thậm chí EBML, nhưng XML là ngôn ngữ thường được sử dụng).
- XML thường là định dạng cho dữ liệu truyền, mặc dầu bất cứ định dạng nào cũng có thể dùng, bao gồm HTML định dạng trước, văn bản thuần (plain text), JSON và ngay cả EBML.

Giống như DHTML, LAMP hay SPA, Ajax tự nó không phải là một công nghệ mà là một thuật ngữ mô tả việc sử dụng kết hợp một nhóm nhiều công nghệ với nhau. Trong thực tế, các công nghệ dẫn xuất hoặc kết hợp dựa trên Ajax như AFLAX cũng đã xuất hiện.

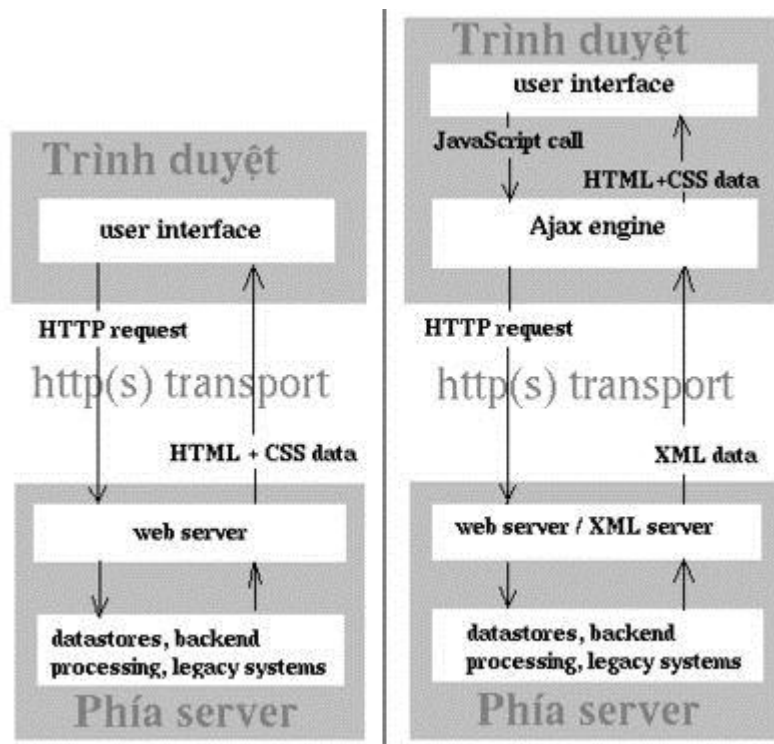
So sánh với các ứng dụng web truyền thống

Tương tác đồng bộ trong ứng dụng web truyền thống (trên) và dị bộ trong ứng dụng AJAX
Adaptive Path

Hiểu nôm na: *Điểm khác biệt cơ bản nhất của công nghệ này là việc xử lý thông tin được thực hiện trên máy thân chủ (Client) thay vì máy phục vụ (Server) như cách truyền thống. Máy chủ chỉ làm một việc đơn giản là nhận thông tin từ máy khách và trả các dữ liệu về cho máy khách. Máy khách xử lý sơ bộ thông tin của người dùng nhập vào, sau đó chuyển về máy chủ rồi nhận dữ liệu từ máy chủ và xử lý để hiển thị cho người dùng.*

Các ứng dụng Ajax phần lớn trông giống như thể chúng được đặt trên máy của người sử dụng hơn là được đặt trên một máy phục vụ thông qua Internet. Lý do: các trang được cập nhật nhưng không nạp lại (*refresh*) toàn bộ. "Mọi thao tác của người sử dụng sẽ gửi mẫu của một lời gọi JavaScript tới bộ xử lý (*engine*) Ajax thay vì tạo ra một yêu cầu HTTP (*HTTP request*)", [Jesse James Garrett](#) đã ghi như vậy trong bài luận đầu tiên định nghĩa về thuật ngữ này. "Mọi đáp ứng

cho thao tác của người sử dụng sẽ không cần truy vấn tới máy phục vụ – ví dụ như việc kiểm tra một cách đơn giản sự hợp lệ của dữ liệu, sửa đổi dữ liệu trong bộ nhớ và thậm chí một vài thao tác duyệt trang – bộ xử lý Ajax tự nó đảm nhận trách nhiệm này. Nếu bộ xử lý cần gì từ máy phục vụ để đáp ứng – như khi nó gửi dữ liệu để xử lý, tải về bổ sung các mã giao diện hay nhận về dữ liệu mới – nó sẽ thực hiện các yêu cầu tới máy phục vụ một cách không đồng bộ, thông thường sử dụng XML, mà không làm gián đoạn sự tương tác của người sử dụng với ứng dụng web".



So sánh ứng dụng web truyền thống (trái) với AJAX
Adaptive Path

Các ứng dụng truyền thống về bản chất là gửi dữ liệu từ các form, được nhập bởi người sử dụng, tới một máy phục vụ web. Máy phục vụ web sẽ trả lời bằng việc gửi về một trang web mới. Do máy phục vụ phải tạo ra một trang web mới mỗi lần như vậy nên các ứng dụng chạy chậm và "lúng túng" hơn.

Mặt khác, các ứng dụng Ajax có thể gửi các yêu cầu tới máy phục vụ web để nhận về chỉ những dữ liệu cần thiết, thông qua việc dùng SOAP hoặc một vài dịch vụ web dựa trên nền tảng XML cục bộ khác. Trên máy thân chủ (*client*), JavaScript sẽ xử lý các đáp ứng của máy chủ. Kết quả là trang web được hiển thị nhanh hơn vì lượng dữ liệu trao đổi giữa máy chủ và trình duyệt web giảm đi rất nhiều. Thời gian xử lý của máy chủ web cũng vì thế mà được giảm theo vì phần lớn thời gian xử lý được thực hiện trên máy khách của người dùng.

Có thể xem xét một website nhiếp ảnh cho phép người sử dụng nhập các tiêu đề như một ví dụ về cách làm việc của Ajax. Với một ứng dụng web truyền thống, toàn bộ trang web bao gồm cả

các ảnh cần được nạp lại. Với các công nghệ Ajax, DHTML có thể thay thế chỉ những đoạn tiêu đề và kết quả là người dùng có những giao dịch "mượt mà" đáng quan tâm.

Các trình duyệt hỗ trợ Ajax

- Apple Safari từ 1.2 trở lên
- Konqueror
- Microsoft Internet Explorer từ 4.0 trở lên
- Mozilla/Mozilla Firefox từ 1.0 trở lên
- Netscape từ 7.1 trở lên
- Opera từ 8.0 trở lên
- GreenBrowser 4.3

Tham khảo

- [Ajax: A New Approach to Web Applications](#), bởi [Jesse James Garrett](#). Bài báo đầu tiên về Ajax
- [AJAX:Getting Started](#) Bài báo của Mozilla.org.

Xem thêm

- [Web 2.0](#)

Ví dụ

- [Flickr](#)
- [Google Local](#) (trước đây [Google Maps](#))
- [Windows Live](#)

Ajax của người Việt:

- [Địa danh Việt Nam](#) : Website với trên 48000 địa danh Việt Nam trên bản đồ trực tuyến có ảnh vệ tinh. Sử dụng công nghệ AJAX và Google Maps API.
- **Thắng Cảnh Đẹp** [<http://thangcanhdep.com>]: sử dụng 100% AJAX, kết hợp với Google MAP API.
- **PHP basic** [<http://phpbasic.com>]: sử dụng AJAX + diễn đàn thảo luận
- [Một ứng dụng web truyền thống](#) xử lý sự tương tác với người dùng so với một ứng dụng Ajax
- [Các thành phần khác nhau](#) của các ứng dụng Ajax tương tác với nhau so với các ứng dụng web truyền thống

Công cụ

- [ASP.NET AJAX](#), toolkit Ajax của [Microsoft](#) cho ASP.NET
- [Anthem.NET](#), toolkit AJAX của Jason Diamond cho ASP.NET
- [Dojo Toolkit](#), toolkit Ajax/DHTML
- [Prototype](#), khuôn khổ mã nguồn mở
- [Sajax](#), toolkit Ajax đơn giản 1
- [Rialto](#) (*Rich Internet AppLication TOolkit*)
- [ZK](#), Ajax but no Javascript.
- [jQuery](#), Toolkit xử lý javascript cho web page (support Ajax mạnh)
- [Mootools](#), Một Toolkit khác xử lý javascript cho web page (support Ajax mạnh)

AJAX Toàn Tập - Bài Mở Đầu

Quá trình phát triển các công nghệ ứng dụng WEB.

Trước khi tìm hiểu về Ajax, chúng ta cùng xem xét quá trình phát triển các công nghệ Web, nguyên nhân và hoàn cảnh xuất hiện công nghệ Ajax.

Quá trình phát triển các công nghệ trong ứng dụng Web

Ban đầu, các trang Web là tĩnh; người dùng gửi yêu cầu một tài nguyên nào đó, và server sẽ trả về tài nguyên đó. Các trang Web không có gì hơn là một văn bản được định dạng và phân tán. Đối với các trình duyệt, thì các trang Web tĩnh không phải là các vấn đề khó khăn, và trang Web lúc đầu chỉ để thông tin về các sự kiện, địa chỉ, hay lịch làm việc qua Internet mà thôi, chưa có sự tương tác qua các trang Web. Năm 1990, Tim Berners-Lee, tại CERN, đã sáng chế ra HTML (Hyper Text Markup Language), ngôn ngữ đánh dấu siêu văn bản. HTML rất đơn giản và dễ dùng, và nó trở thành một ngôn ngữ rất phổ biến và cơ bản.

Tuy nhiên, không lâu sau đó, nhu cầu về các trang Web động, có sự tương tác ngày một tăng, chính vì thế sự ra đời các công nghệ Web động là một điều tất yếu. Sau đây là một số công nghệ Web động cơ bản:

1. CGI

Giải pháp đầu tiên để làm các trang Web động là Common Gateway Interface (CGI). CGI cho phép tạo các chương trình chạy khi người dùng gửi các yêu cầu. Giả sử khi cần hiển thị các các mục để bán trên Web site – với một CGI script ta có thể truy nhập cơ sở dữ liệu sản phẩm và hiển thị kết quả. Sử dụng các form HTML đơn giản và các CGI script, có thể tạo các “cửa hàng” ảo cho phép bán sản phẩm cho khách hàng qua một trình duyệt. CGI script có thể được viết bằng

một số ngôn ngữ từ Perl cho đến Visual Basic.

Tuy nhiên, CGI không phải là cách an toàn cho các trang Web động. Với CGI, người khác có thể chạy chương trình trên hệ thống. Vì thế có thể chạy các chương trình không mong muốn gây tổn hại hệ thống. Nhưng dù vậy, cho đến hôm nay thì CGI vẫn còn được sử dụng.

2. Applet

Tháng 5/1995, John Gage của hãng Sun và Andressen (nay thuộc Netscape Communications Corporation) đã công bố một ngôn ngữ lập trình mới có tên Java. Netscape Navigator đã hỗ trợ ngôn ngữ mới này, và một con đường mới cho các trang Web động được mở ra, kỷ nguyên của applet bắt đầu.

Applet cho phép các nhà phát triển viết các ứng dụng nhỏ nhúng vào trang Web. Khi người dùng sử dụng một trình duyệt hỗ trợ Java, họ có thể chạy các applet trong trình duyệt trên nền máy ảo Java Virtual Machine (JVM). Dù rằng applet làm được nhiều điều song nó cũng có một số nhược điểm: thường bị chặn bởi việc đọc và ghi các file hệ thống, không thể tải các thư viện, hoặc đôi khi không thể thực thi trên phía client. Bù lại những hạn chế trên, applet được chạy trên một mô hình bảo mật kiểu sandbox bảo vệ người dùng khỏi các đoạn mã nguy hiểm.

Có những lúc applet được sử dụng rất nhiều, nhưng nó cũng có những vấn đề nảy sinh: đó là sự phụ thuộc vào máy ảo Java JVM, các applet chỉ thực thi khi có môi trường thích hợp được cài đặt phía client, hơn nữa tốc độ của các applet là tương đối chậm vì thế applet không phải là giải pháp tối ưu cho Web động.

3. JavaScript

Cùng thời gian này, Netscape đã tạo ra một ngôn ngữ kịch bản gọi là JavaScript. JavaScript được thiết kế để việc phát triển dễ dàng hơn cho các nhà thiết kế Web và các lập trình viên không thành thạo Java. (Microsoft cũng có một ngôn ngữ kịch bản gọi là VBScript). JavaScript ngay lập tức trở thành một phương pháp hiệu quả để tạo ra các trang Web động.

Việc người ta coi các trang như là một đối tượng đã làm nảy sinh một khái niệm mới gọi là Document Object Model (DOM). Lúc đầu thì JavaScript và DOM có một sự kết hợp chặt chẽ nhưng sau đó chúng được phân tách. DOM hoàn toàn là cách biểu diễn hướng đối tượng của trang Web và nó có thể được sửa đổi với các ngôn ngữ kịch bản bất kỳ như JavaScript hay VBScript.

Tổ chức World Wide Web Consortium (W3C) đã chuẩn hóa DOM, trong khi European Computer Manufacturers Association (ECMA) phê duyệt JavaScript dưới dạng đặc tả ECMAScript.

4. JSP/Servlet, ASP và PHP

Cùng với Java, Sun đồng thời đưa ra một công nghệ mới gọi là servlet. Các đoạn mã Java sẽ không chạy phía client như với applet; chúng sẽ được chạy trên một ứng dụng phía server. Servlet cũng đồng thời phục vụ các CGI script. Servlet là một bước tiến lớn, nó đưa ra một thư viện hàm API trên Java và một thư viện hoàn chỉnh để thao tác trên giao thức HTTP. JavaServer Page (JSP) là một công nghệ lập trình Web của Sun, cùng với nó là một công nghệ khác của Microsoft - Active Server Pages (ASP), JSP là công nghệ đòi hỏi một trình chủ hiểu được Java.

Microsoft đã nghiên cứu các nhược điểm của servlet và tạo ra ASP dễ dàng hơn để thiết kế các

trang web động. Microsoft thêm các bộ công cụ rất mạnh và sự tích hợp rất hoàn hảo với các Web server. JSP và ASP có những nét tương đương vì chúng đều được thiết kế để phân tách qua trình xử lý khỏi quá trình biểu diễn. Có sự khác biệt về kỹ thuật, song cả hai đều cho phép các nhà thiết kế Web tập trung vào cách bố trí (layout) trong khi các nhà phát triển phần mềm thì tập trung vào các kỹ thuật lập trình logic.

Tất nhiên Microsoft và Sun không độc quyền ở các giải pháp phía server. Còn có các công nghệ khác, trong đó phải kể đến là PHP (Hypertext Preprocessor) cho tới Cold Fusion. Các công nghệ này cung cấp các bộ công cụ rất mạnh cho các nhà phát triển.

5. Flash

Năm 1996, FutureWave đã đưa ra sản phẩm FutureSplash Animator. Sau đó FutureWave thuộc sở hữu của Macromedia, và công ty này đưa ra sản phẩm Flash. Flash cho phép các nhà thiết kế tạo các ứng dụng hoạt họa và linh động. Flash không đòi hỏi các kỹ năng lập trình cao cấp và rất dễ học. Cũng giống như các nhiều giải pháp khác Flash yêu cầu phần mềm phía client. Chẳng hạn như gói Shockwave Player plug-in có thể được tích hợp trong một số hệ điều hành hay trình duyệt.

6. DHTML

Khi Microsoft và Netscape đưa ra các version 4 của các trình duyệt của họ, thì các nhà phát triển Web có một lựa chọn mới: Dynamic HTML (DHTML). DHTML không phải là một chuẩn của W3C; nó giống một bộ công cụ thương mại hơn. Trong thực tế nó là một tập hợp gồm HTML, Cascading Style Sheets (CSS), JavaScript, và DOM. Tập hợp các công nghệ trên cho phép các nhà phát triển sửa đổi nội dung và cấu trúc của một trang Web một cách nhanh chóng. Tuy nhiên, DHTML yêu cầu sự hỗ trợ từ các trình duyệt. Mặc dù cả Internet Explorer và Netscape hỗ trợ DHTML, nhưng các thể hiện của chúng là khác nhau, các nhà phát triển cần phải biết được loại trình duyệt nào mà phía client dùng. DHTML thật sự là một bước tiến mới, nhưng nó vẫn cần một sự qui chuẩn để phát triển. Hiện nay DHTML vẫn đang trên con đường phát triển mạnh.

7. XML

Kể từ khi ra đời vào giữa năm 1990, eXtensible Markup Language (XML) của W3C dẫn xuất của SGML đã trở nên rất phổ biến. XML có mặt ở khắp nơi, Microsoft Office 12 cũng sẽ hỗ trợ định dạng file XML.

Ngày nay chúng ta có rất nhiều dạng dẫn xuất của XML cho các ứng dụng Web (tất nhiên là có cả XHTML): XUL của Mozilla; XAMJ, một sản phẩm mã nguồn mở trên nền Java; MXML từ Macromedia; và XAML của Microsoft.

Bài 0 (tiếp): Web 2.0 và công nghệ AJAX.

Công nghệ Web thế hệ thứ hai – Web 2.0

Được xem là một cuộc cách mạng trên thế giới mạng, thế hệ web mới có những thay đổi quan trọng không chỉ ở nền tảng công nghệ mà còn cả ở cách thức sử dụng - hình thành nên môi trường cộng đồng, ở đó mọi người cùng tham gia đóng góp cho xã hội “ảo” chứ không chỉ “duyet và xem”.

Web 2.0 là gì? Làm sao phân biệt đâu là Web 1.0 đâu là Web 2.0? Thuật ngữ “Web 2.0” đang trở nên thịnh hành. Thực chất, Web 2.0 có nghĩa là sử dụng web đúng với bản chất và khả năng của nó.

Mục tiêu đầu tiên của những người tiên phong xây dựng Internet là nhằm kết nối các nhà nghiên cứu và các máy tính của họ với nhau để có thể chia sẻ thông tin hiệu quả. Khi bổ sung World Wide Web (năm 1990), Tim Berners-Lee cũng nhằm mục tiêu tạo phương tiện cho phép người dùng tự do đưa thông tin lên Internet và dễ dàng chia sẻ với mọi người (trình duyệt web đầu tiên do Berners-Lee viết bao gồm cả công cụ soạn thảo trang web). Tuy nhiên, sau đó web đã phát triển theo hướng hơi khác mục tiêu ban đầu.

Tuy có một số ngoại lệ nhưng thế giới Web 1.0 (thế hệ web trước Web 2.0) chủ yếu gồm các website “đóng” của các hãng thông tấn hay các công ty nhằm mục đích tiếp cận độc giả hay khách hàng hiệu quả hơn. Nó là phương tiện phát tin hơn là phương tiện chia sẻ thông tin. Chỉ đến gần đây, với sự xuất hiện của nhiều kỹ thuật mới như blog (hay weblog), wiki... web mới trở nên có tính cộng đồng (và cộng tác) hơn và trở nên gần hơn với sự kỳ vọng và khả năng thực sự của nó.

Khái niệm Web 2.0 đầu tiên được Dale Dougherty, phó chủ tịch của O'Reilly Media, đưa ra tại hội thảo Web 2.0 lần thứ nhất do O'Reilly Media và MediaLive International tổ chức vào tháng 10/2004. Dougherty không đưa ra định nghĩa mà chỉ dùng các ví dụ so sánh phân biệt Web 1.0 và Web 2.0: “DoubleClick là Web 1.0; Google AdSense là Web 2.0. Ofoto là Web 1.0; Flickr là Web 2.0. Britannica Online là Web 1.0; Wikipedia là Web 2.0. v.v...”.

Sau đó Tim O'Reilly, chủ tịch kiêm giám đốc điều hành O'Reilly Media, đã đúc kết lại 7 đặc tính của Web 2.0:

1. Web có vai trò nền tảng, có thể chạy mọi ứng dụng
2. Tập hợp trí tuệ cộng đồng
3. Dữ liệu có vai trò then chốt
4. Phần mềm được cung cấp ở dạng dịch vụ web và được cập nhật không ngừng
5. Phát triển ứng dụng dễ dàng và nhanh chóng
6. Phần mềm có thể chạy trên nhiều thiết bị
7. Giao diện ứng dụng phong phú

Thoạt đầu, Web 2.0 được chú trọng tới yếu tố công nghệ, nhấn mạnh tới vai trò nền tảng ứng dụng. Nhưng đến hội thảo Web 2.0 lần 2 tổ chức vào tháng 10/2005, Web 2.0 được nhấn mạnh đến tính chất sâu xa hơn – yếu tố cộng đồng.