

AJAX - Tương lai của ứng dụng Web

Bài 10: Đối tượng XMLHttpRequest - Giới thiệu.

Bây giờ ta sẽ xét vấn đề trọng tâm của Ajax: đối tượng XMLHttpRequest. XMLHttpRequest được đưa ra ban đầu trong Internet Explorer 5 như là một ActiveX component. Nó chỉ hoạt động trong Internet Explorer điều này làm cho số ít các nhà phát triển làm việc với XMLHttpRequest, cho tới khi nó trở thành một chuẩn không chính thức trong Mozilla 1.0 và Safari 1.2. Một điều rất quan trọng cần chú ý là XMLHttpRequest không là chuẩn của W3C, mặc dù rất nhiều chức năng của nó được đề xuất trong các đặc tả: DOM Level 3 Load and Save Specification.

Vì XMLHttpRequest không phải là một chuẩn, nên có những sự khác biệt nhỏ trong các trình duyệt về hỗ trợ đối tượng này, tuy nhiên hầu hết các phương thức và thuộc tính của nó đều được hỗ trợ. Hiện thời, Firefox, Safari, Opera, Konqueror, và Internet Explorer tất cả đều hỗ trợ XMLHttpRequest tương tự nhau. Nếu một số lượng người dùng đáng kể vẫn truy cập website với các trình duyệt phiên bản cũ, thì phải cân nhắc về sự lựa chọn công nghệ sẽ ứng dụng.

Giới thiệu về XMLHttpRequest

Trước tiên cần tạo một đối tượng XMLHttpRequest bằng JavaScript trước khi sử dụng đối tượng này để gửi request và xử lý các response. XMLHttpRequest chưa là chuẩn của W3C, vì thế phải dùng JavaScript theo nhiều cách để tạo một thể hiện của XMLHttpRequest. Internet Explorer thực thi XMLHttpRequest như một đối tượng ActiveX, và các trình duyệt khác như Firefox, Safari, và Opera thực thi nó như một đối tượng JavaScript nguyên thủy. Vì sự khác nhau này, đoạn mã JavaScript phải chứa các cấu trúc logic để tạo một thể hiện của XMLHttpRequest dùng kỹ thuật ActiveX hay đối tượng JavaScript nguyên thủy.

Rất may mắn, trong trường hợp này không cần phải viết mã một cách tỉ mỉ để xác định kiểu trình duyệt và làm thế nào để tạo một thể hiện của đối tượng XMLHttpRequest. Công việc chỉ đơn giản là kiểm tra sự hỗ trợ đối tượng ActiveX của trình duyệt. Nếu hỗ trợ ActiveX, thì tạo một đối tượng XMLHttpRequest dùng ActiveX. Trường hợp khác, tạo đối tượng này bằng kỹ thuật tạo đối tượng JavaScript nguyên thủy. Đoạn mã JavaScript sau dễ dàng tạo ra các thể hiện của đối tượng XMLHttpRequest mà không quan tâm tới trình duyệt.

Tạo một thể hiện của đối tượng XMLHttpRequest:

Trích:

```
var xmlhttp;  
function createXMLHttpRequest() {
```

```
if (window.ActiveXObject) {  
xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");  
}  
else if (window.XMLHttpRequest) {  
xmlHttp = new XMLHttpRequest();  
}  
}
```

Việc tạo một đối tượng XMLHttpRequest khá là đơn giản. Trước tiên, tạo một biến toàn cục (giả sử có tên là) xmlHttp để lưu trữ một tham chiếu đến đối tượng. Phương thức createXMLHttpRequest thực hiện việc tạo một thể hiện của XMLHttpRequest. Đoạn chương trình này chứa một câu lệnh rẽ nhánh đơn giản để xác định cách thức tạo một đối tượng.

Lệnh gọi window.ActiveXObject sẽ trả về một biến kiểu object hay giá trị null, tương đương với điều kiện true hay false của câu lệnh if, đây là dấu hiệu chỉ ra cho ta biết trình duyệt hỗ trợ ActiveX control, đó là Internet Explorer. Nếu thế, XMLHttpRequest được tạo ra là một thể hiện của ActiveXObject, truyền một chỉ báo kiểu string để xác định loại đối tượng ActiveX cần tạo. Trong thể hiện này, ta cung cấp tham số Microsoft.XMLHTTP cho hàm tạo, để chỉ ra rằng cần tạo một thể hiện của XMLHttpRequest.

Nếu lệnh gọi tới window.ActiveXObject không được thực thi (điều kiện false của lệnh if), lệnh JavaScript sẽ rẽ nhánh tương ứng với lệnh else, để xác định rằng trình duyệt thực thi XMLHttpRequest dưới dạng một đối tượng JavaScript nguyên thủy. Nếu window.XMLHttpRequest tồn tại, sau đó một thể hiện của XMLHttpRequest được tạo ra.

Kiểu dữ liệu XMLHttpRequest của JavaScript tương thích với rất nhiều trình duyệt khác nhau, có thể truy cập các thuộc tính và phương thức của một thể hiện của XMLHttpRequest mà không cần quan tâm tới cách tạo các thể hiện này. Điều này làm cho việc phát triển các ứng dụng đơn giản hơn và làm cho JavaScript không phụ thuộc vào trình duyệt cụ thể.

Bài sau chúng ta sẽ đi vào phân tích các đặc tính của XMLHttpRequest.

Bài 11: Đối tượng XMLHttpRequest - Phân tích các đặc tính - Phương thức và thuộc tính

Phân tích các đặc tính của XMLHttpRequest:

Các phương thức và thuộc tính

Danh mục sau chỉ ra các phương thức điển hình của đối tượng XMLHttpRequest:

abort() Hủy request hiện thời.

getAllResponseHeaders() Trả về tất cả các response header cho HTTP request dưới dạng cặp key/value.

getResponseHeader("header") Trả về giá trị kiểu string của header xác định.

open("method", "url") Thiết lập giai đoạn cho một lời gọi tới server. Tham số của method có thể là GET, POST, hay PUT. Tham số url có thể là quan hệ hay trực tiếp. Phương thức này còn có 3 tham số tùy chọn.

send(content) Gửi request tới server.


setRequestHeader("header", "value") Thiết lập header xác định cho giá trị cung cấp. open() phải được gọi trước khi cố gắng thiết lập bất kỳ một header nào.

Bây giờ ta xét cụ thể các phương thức này:

- **void open(string method, string url, boolean async, string username, string password):** Phương thức này thiết lập một phiên gọi tới server. Ý nghĩa của nó này là khởi tạo một request. Nó có hai tham số yêu cầu và ba tham số tùy chọn; ta phải cung cấp đặc tả của phương thức được triệu gọi (GET, POST, hay PUT) và địa chỉ URL của tài nguyên được gọi. Cũng có thể truyền một biến chỉ báo kiểu Boolean, xác định việc truyền là bất đồng bộ - giá trị mặc định là true, nghĩa là các request luôn bất đồng bộ theo mặc định. Nếu truyền một giá trị false, quá trình xử lý sẽ phải đợi cho đến khi có phản hồi từ server. Việc truyền dữ liệu bất đồng bộ là một lợi thế của việc dùng Ajax, vì vậy thiết lập tham số này ở giá trị false đã thay đổi mục đích việc sử dụng XMLHttpRequest. Ta thấy nó rất hữu ích trong một số trường hợp chẳng hạn như chứng thực người dùng (validating user) trong khi trang web của không hề bị thay đổi. Hai tham số cuối tùy chọn, cho phép sử dụng username và password.
- **void send(content):** Phương thức này thực hiện gửi request tới server. Nếu request được khai báo bất đồng bộ, kết quả được trả về ngay, còn không nó sẽ đợi cho tới khi nhận được response từ server. Tham số có thể là một thể hiện của một đối tượng DOM, một luồng dữ liệu vào (input stream), hay một kiểu string. Nội dung truyền cho phương thức này được gửi đi như một phần trong request.
- **void setRequestHeader(string header, string value):** Phương thức thiết lập giá trị cho phần header đã cho trong HTTP request. Tham số bao gồm một biến kiểu string biểu thị header được thiết lập và một string khác biểu diễn giá trị để thay thế trong header. Chú ý là nó cần phải được gọi trước khi gọi đến open(). Trong hầu hết các phương thức thì cặp open() và send() thường xuyên được gọi.
- **void abort():** Phương thức này rất dễ hiểu, nó sẽ hủy request.
- **string getAllResponseHeaders():** Trả về một biến kiểu string chứa response header của HTTP request. Header bao gồm các trường Content-Length, Date, và URI.
- **string getResponseHeader(string header):** Phương thức này tương tự như getAllResponseHeaders(); ngoại trừ việc nó nhận một tham số biểu diễn giá trị xác định header mà ta cần đón bắt, giá trị trả về cũng có kiểu string.

Ngoài những phương thức chuẩn này, đối tượng XMLHttpRequest còn có các thuộc tính được liệt kê sau đây. Anh em chú ý ta sẽ phải sử dụng các thuộc tính mở rộng này khi làm việc với XMLHttpRequest.

onreadystatechange Bộ xử lý sự kiện cho một sự kiện phát sinh mỗi khi có sự thay đổi trạng thái.

readyState Trạng thái của request. Có 5 giá trị 
0 = uninitialized, 1 = loading, 2 = loaded, 3 = interactive, và 4 = complete.

responseText Response trả về từ server dưới dạng string.

responseXML Response trả về từ server dưới dạng XML. Đối tượng này có thể được phân tách và khảo sát như một đối tượng tài liệu DOM.

status Mã trạng thái HTTP từ server (chẳng hạn 200 nếu không có lỗi, 404 cho lỗi Not Found, ...).

statusText Thông điệp của mã trạng thái HTTP (chẳng hạn OK hay Not Found, ...).