

KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG  
BỘ MÔN MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

# LẬP TRÌNH WEB



## Chương 7: AJAX

Biên soạn : ThS. HÀ DUY AN

Cập nhật : ThS. NGUYỄN CAO HỒNG NGỌC

# Nội dung



- Giới thiệu về AJAX
- XMLHttpRequest – Create Object
- XMLHttpRequest – Request
- XMLHttpRequest – Response
- XMLHttpRequest – readyState
- HTTP Header
- Link tham khảo thêm

# Giới thiệu về AJAX



## ■ AJAX = Asynchronous Javascript and XML

- AJAX không phải là một ngôn ngữ lập trình mới, nó cung cấp một hướng tiếp cận mới cho các ứng dụng web dựa trên các chuẩn có sẵn.
- Đối với các ứng dụng web truyền thống, nếu muốn cập nhật nội dung thông tin trên trang web thì bắt buộc trình duyệt phải tải lại toàn bộ trang web gây lãng phí và tốn thời gian chờ đợi không cần thiết của người dùng.
- AJAX cho phép tạo ra các trang web có khả năng cập nhật lại một phần nội dung mà không cần tải lại toàn bộ trang.
- Một số ứng dụng sử dụng AJAX: Google Maps, Gmail, Youtube, Facebook,...

# Giới thiệu về AJAX (tt)



- **AJAX hoạt động thế nào?**
  - AJAX hoạt động dựa trên các chuẩn Internet, nó là sự kết hợp của các kỹ thuật sau:
    - XMLHttpRequest object: trao đổi dữ liệu với server
    - JavaScript/DOM: dùng để hiển thị dữ liệu, tương tác với người dùng
    - CSS: trình bày dữ liệu
    - XML: định dạng dữ liệu cho mục đích truyền dữ liệu

# Giới thiệu về AJAX (tt)

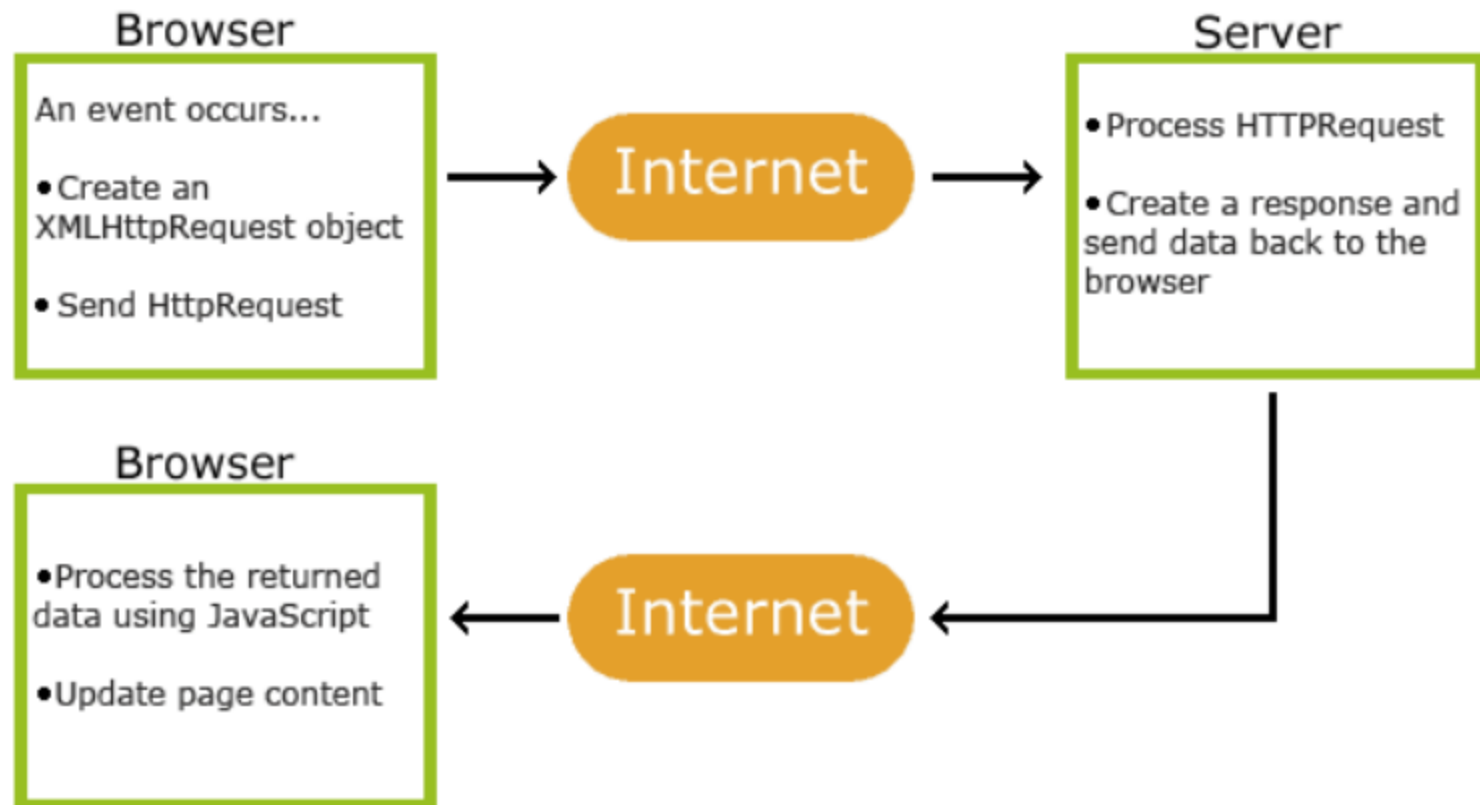


- AJAX hoạt động thế nào? (tt)
  - Với kỹ thuật AJAX dữ liệu được truyền giữa browser và server có thể được ẩn bên dưới browser, có 2 chế độ hoạt động:
    - Đồng bộ: trình duyệt sẽ chờ cho đến khi server trả về dữ liệu trước khi các lệnh JavaScript tiếp theo được thực thi
    - Bất đồng bộ: các lệnh JavaScript tiếp theo sẽ được thực thi sau khi yêu cầu được gửi mà không chờ kết quả trả về từ server

# Giới thiệu về AJAX (tt)



## ■ AJAX hoạt động thế nào?



Nguồn: [http://www.w3schools.com/ajax/ajax\\_intro.asp](http://www.w3schools.com/ajax/ajax_intro.asp)

# XMLHttpRequest – Create Object



- Trọng tâm của kỹ thuật AJAX là việc sử dụng đối tượng: XMLHttpRequest để gửi các yêu cầu và nhận kết quả trả về từ server
- Hầu hết các trình duyệt đều hỗ trợ đối tượng XMLHttpRequest, riêng IE5 và IE6 dùng ActiveXObject
- Tạo một đối tượng XMLHttpRequest:

```
var xmlhttp;  
if (window.XMLHttpRequest) { // code for IE7+, Firefox, Chrome, Opera, Safari  
    xmlhttp=new XMLHttpRequest();  
} else { // code for IE6, IE5  
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");  
}
```



# XMLHttpRequest – Request



- ***open(method,url,async)***: dùng để xác định phương thức gửi dữ liệu, url và việc xử lý dữ liệu đồng bộ hay bất đồng bộ
  - *method*: phương thức gửi dữ liệu: GET hay POST
  - *url*: vị trí của tập tin trên server
  - *async*: true (bất đồng bộ) hay false (đồng bộ)
- ***send(string)***: gửi yêu cầu đến server
  - *string*: dữ liệu được gửi trong phần thân của thông điệp, chỉ dùng với phương thức POST
- Sau khi dữ liệu được gửi đi và đối tượng nhận được thông điệp trả lời từ phía server thì các thông tin về thông điệp trả lời này có thể được truy xuất thông qua các thuộc tính của đối tượng XMLHttpRequest



# XMLHttpRequest – Request (tt)



```
xmlhttp.open("GET", "example.php?t="+Math.random(), true);  
xmlhttp.send();
```

```
xmlhttp.open("POST", "example.php", true);  
xmlhttp.setRequestHeader("Content-type", "application/x-www-form-  
urlencoded");  
xmlhttp.send("fname=Henry&lname=Ford");
```

# XMLHttpRequest – Response



- Để nhận thông điệp trả lời từ server, sử dụng 2 thuộc tính sau của đối tượng XMLHttpRequest:
  - *responseText*: nội dung (phần thân) của thông điệp trả lời
  - *responseXML*: chứa một XML DOM document nếu thông điệp trả về có kiểu nội dung (content type) là "text/xml" hay "application/xml"

```
<!DOCTYPE html><html><head>
<script>
function loadXMLDoc() {
var xmlhttp;
if (window.XMLHttpRequest) { // code for IE7+, Firefox, Chrome, Opera, Safari
    xmlhttp=new XMLHttpRequest();
} else { // code for IE6, IE5
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
xmlhttp.onreadystatechange=function() {
    if (xmlhttp.readyState==4 && xmlhttp.status==200) {
        document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
    }
}
xmlhttp.open("GET","ajax_info.txt",true);
xmlhttp.send();
}
</script>
</head><body>
<div id="myDiv"><h2>Let AJAX change this text</h2></div>
<button type="button" onclick="loadXMLDoc()">Change Content</button>
</body></html>
```

# XMLHttpRequest – readyState (tt)



- Mỗi khi thuộc tính readyState thay đổi giá trị thì sự kiện onreadystatechange được phát sinh => cho phép viết các script kiểm tra giá trị của thuộc tính readyState và xử lý dữ liệu gửi về từ server
- 3 thuộc tính quan trọng của đối tượng XMLHttpRequest
  - **onreadystatechange**: chứa hàm (hay tên hàm) được gọi mỗi khi thuộc tính readyState thay đổi giá trị
  - **readyState**: xác định trạng thái của XMLHttpRequest (xem slide tiếp theo)
  - **status**: mã trạng thái của thông điệp trả lời
    - 200: “OK”
    - 403: “Forbidden”
    - 404: “Not found”

# XMLHttpRequest – readyState



- *readyState*: xác định giai đoạn trong chu kỳ request/response đang được kích hoạt, các giá trị có thể có của thuộc tính này:
  - 0: chưa được khởi tạo. Phương thức open() chưa được gọi
  - 1: đã khởi tạo. Phương thức open() đã thực thi nhưng send() chưa được gọi
  - 2: đã gửi. Phương thức send() đã được gọi, nhưng chưa nhận được thông điệp trả lời
  - 3: đang nhận dữ liệu
  - 4: hoàn tất. Đã nhận được tất cả dữ liệu trả về từ server

# HTTP Header



- `setRequestHeader("header", "value")` — thêm các header field trong thông điệp yêu cầu của đối tượng `XMLHttpRequest`  
**VD:** `xhr.setRequestHeader("MyHeader", "MyValue");`
- `getResponseHeader("header")` — lấy về giá trị của một header field trong thông điệp trả lời  
**VD:** `var headerValue xhr.getResponseHeader("content-type");`
- `getAllResponseHeaders()` — hàm trả về một chuỗi chứa tất cả các header field trong thông điệp trả lời

**VD:** `var allHeaders=xhr.getAllResponseHeaders()`

Kết quả của câu lệnh là biến `allHeaders` có giá trị là một chuỗi có định dạng giống như sau:

```
ETag: "5000000003362e-36-4ad6ee0f34206"  
Content-Length: 54  
Keep-Alive: timeout=5, max=100  
Content-Type: text/plain  
Last-Modified: Wed, 21 Sep 2011 07:53:58 GMT
```

# Link tham khảo thêm



- 
- [https://www.w3schools.com/js/js\\_ajax\\_php.asp](https://www.w3schools.com/js/js_ajax_php.asp)
  - [http://www.w3schools.com/php/php\\_ajax\\_database.asp](http://www.w3schools.com/php/php_ajax_database.asp)
  - [http://www.w3schools.com/php/php\\_ajax\\_livesearch.asp](http://www.w3schools.com/php/php_ajax_livesearch.asp)
  - [http://www.w3schools.com/php/php\\_ajax\\_poll.asp](http://www.w3schools.com/php/php_ajax_poll.asp)



# HẾT

