

## Bài 15. THỰC HÀNH

### Nội dung:

1. Ajax là gì.....	1
2. Hoạt động của ajax.....	1
3. Ajax trong ASP.NET MVC .....	5

### 1. Ajax là gì

AJAX, viết tắt từ Asynchronous JavaScript and XML (JavaScript và XML không đồng bộ), là bộ công cụ cho phép tăng tốc độ ứng dụng web bằng cách chia nhỏ dữ liệu và chỉ tải những thông tin thay đổi thay vì tải toàn bộ trang web.

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 dù 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.

### 2. Hoạt động của ajax

### Trình duyệt

Khi một sự kiện xuất hiện:

- Tạo đối tượng XMLHttpRequest
- Gửi đi đối tượng HttpRequest



**Internet**



### Server

- Xử lý HttpRequest
- Tạo đối tượng response và gửi dữ liệu trở lại cho trình duyệt.



### Trình duyệt

- Sử dụng Javascript để xử lý và trả về dữ liệu.
- **Cập nhật** nội dung trang web.

### Ví dụ 1: Xử lý form dùng Javascript

```

<!DOCTYPE html>
<html dir="ltr" lang="en-US">
  <head>
    <meta charset="UTF-8" />
    <title>Load Text Into Multiple Tags Using JavaScript</title>
    <script type="text/javascript">
      function loadText() {
        document.getElementsByTagName("p")[0].innerHTML = "Hello World";
        document.getElementsByTagName("p")[1].innerHTML = "Goodbye World";
      }
    </script>
  </head>
  <body>
    <input type='button' onclick='loadText()' value='Load Some Copy!'/>
    <br />
    <p></p>
    <p></p>
  </body>
</html>

```

Kết quả thực hiện:



Load Some Copy!

Hello World

Goodbye World

Mỗi khi button được click thì sẽ chạy function loadText() và nội dung của 2 thẻ p ở phía dưới được cập nhật giá trị mới là 'Hello World' và 'Goodbye World'.

### Ví dụ 2: Xử lý form dùng Javascript và jquery

```
<!DOCTYPE html>
<html dir="ltr" lang="en-US">
  <head>
    <meta charset="UTF-8" />
    <title>Load Text Into A Tag With jQuery</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.4.4/jquery.js"
type="text/javascript"></script>
    <script type="text/javascript">
      $(document).ready(function() {
        $("input").click(function() {
          $('p').html("Hello World");
        });
      });
    </script>
  </head>
  <body>
    <input type='button' value='Load Some Copy!'/>
    <br />
    <p></p>
    <p></p>
  </body>
</html>
```

### Ví dụ 3: Sử dụng Ajax và jQuery

File exampleajax.html là minh họa sử dụng công nghệ ajax: khi click vào button 'Load some copy' thì nội dung file ajaxCopy.txt sẽ hiển thị ngay phía dưới button.

```
<!DOCTYPE html>
<html dir="ltr" lang="en-US">
  <head>
    <meta charset="UTF-8" />
    <title>AJAX Detection And Data Loading With jQuery</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.4.4/jquery.js"
type="text/javascript"></script>
    <script type="text/javascript">
      $(document).ready(function() {
        $("input").click(function() {
          $('#words').load("ajaxCopy.txt");
        });
      });
    </script>
  </head>
```

```
<body>
  <input type='button' value='Load Some Copy!'/>
  <br /><br />
  <div id="words"></div>
</body>
</html>
```

Kết quả là:

Load Some Copy!

AJAX, viết tắt từ Asynchronous JavaScript and XML (JavaScript và XML không đồng bộ), là bộ công cụ cho phép tăng tốc độ ứng dụng web bằng cách cắt nhỏ dữ liệu và chỉ hiển thị những gì cần thiết, thay vì tải đi tải lại toàn bộ trang web. AJAX không phải một công nghệ đơn lẻ mà là sự kết hợp một nhóm công nghệ với nhau. Trong đó, HTML và CSS đóng vai hiển thị dữ liệu, mô hình DOM trình bày thông tin động, đối tượng XMLHttpRequest trao đổi dữ liệu không đồng bộ với máy chủ web, còn XML là định dạng chủ yếu cho dữ liệu truyền. Đây đều là công nghệ sẵn có nhưng Javascript đã lắp ráp chúng lại để tạo nên những trang web có dữ liệu được tải lên rất nhanh.

Như vậy chúng ta thấy: phần id=words được cập nhật dữ liệu là dữ liệu từ file ajaxCopy.txt còn các thông tin khác vẫn giữ nguyên.

Chú ý: file ajaxCopy.txt được đặt trên server cùng thư mục với file exampleajax.html.

Ví dụ 4: Sử dụng HTML5, Ajax và jQuery

```
<!DOCTYPE html>
<html dir="ltr" lang="en-US">
  <head>
    <meta charset="UTF-8" />
    <title>AJAX Detection And Data Loading Using New School jQuery & HTML5:
    Example # 3</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.4.4/jquery.js"
    type="text/javascript"></script>
    <script type="text/javascript">
      $(document).ready(function() {
        $("button").click(function() {
          $('#words').load("ajaxCopy.txt");
        });
      });
    </script>
  </head>
  <body>
    <button type="button">Load Some Copy!</button>
    <br /><br />
    <article>
      <section id="words">
      </section>
    </article>
  </body>
</html>
```

Kết quả thực hiện tương tự như ví dụ 3.

### 3. Ajax trong ASP.NET MVC

Trong .NET, chúng ta có thể gọi code phía server theo 2 cách:

- jQuery AJAX
- ASP.NET AJAX

#### 3.1. JQuery Ajax

Chúng ta xem xét 3 phương thức jQuery AJAX sau đây:

- \$.ajax() thực hiện việc AJAX request không đồng bộ.
- \$.get() Lấy dữ liệu từ server sử dụng AJAX HTTP GET request.
- \$.post() Lấy dữ liệu từ server sử dụng AJAX HTTP POST request.

Có nhiều loại tham số khác nhau cho phương thức **\$.ajax()**, sau đây là một số loại tham số thường dùng:

- async
- type
- url
- data
- datatype
- contentType
- success
- error

Trong đó:

async	Là false nếu request được gửi là đồng bộ. Giá trị mặc định là true. Nếu bạn đặt là false thì, request của bạn sẽ chặn việc thực thi các đoạn mã khác cho đến khi nhận được phản hồi (response).	async: <b>false</b>
type	Là kiểu của HTTP Request, thường là "POST" hoặc "GET". Mặc định là "GET". Một số dạng khác của request là "PUT" hoặc "DELETE", nhưng các dạng này không được hỗ trợ bởi một số web browsers.	type: <b>"POST"</b>
url	Là URL của request	url: <b>"/Home/JqAJAX"</b>
data	Là dữ liệu được gửi lên server. Nó có thể là đối tượng hoặc query string.	data: JSON.stringify(model_data)

dataType	Là kiểu dữ liệu bạn mong chờ nhận được từ server. Mặc định, JQuery sẽ tìm kiếm dạng MIME của response nếu không có dataType được chỉ định. Các dạng dataType có thể là text, xml, json, script, html jsonp.	dataType: "json"
contentType	Loại content type của dữ liệu được gửi tới server. Mặc định là 'application/x-www-form-urlencoded'.	contentType: 'application/json; charset=utf-8'
success	Một hàm sẽ được gọi (callback) nếu request thành công. Hàm nhận response data (được chuyển đổi thành đối tượng JavaScript nếu dataType là JSON), cũng như trạng thái text của request và raw request object.	success: function (result) { \$('#result').html(result); ; }
error	Một hàm sẽ được gọi (callback) nếu request có lỗi. Hàm sẽ nhận được raw request object và trạng thái text của request.	error: function (result) { alert('Error occurred!!'); }

### Ví dụ 1: Cộng 2 số nguyên

Chúng ta có đoạn code Html và JQuery Ajax trong file **.cshtml** như sau:

```
<h2>Ajax Demo</h2>

<h4>Number 1:<input type="text" id="num1" /></h4>
<h4>Number 2:<input type="text" id="num2" /> </h4>
<input type="button" id="b1" value="Calculate" />

<div id="rs" style="font-size:large"></div>

<script src="~/Scripts/jquery-3.3.1.js"></script>

<script>
    $(document).ready(function () {
        $("#b1").click(function(){
            var n1 = $("#num1").val();
            var n2 = $("#num2").val();
            $.ajax({
                url: '@Url.Action("Math", "AjaxDemo2")?n1=' + n1 + '&n2=' + n2,
                success: function (data) {
                    if (data.status == "ok") {
                        $("#rs").html(data.result);
                    }
                }
            });
        });
    });
</script>
```

*function*(data): trả về đối tượng Json, có 2 thuộc tính là status và result như dưới đây.

**File Controller:**

```
public class AjaxDemo2Controller : Controller
{
    // GET: AjaxDemo2
    public ActionResult Index()
    {
        return View();
    }
    public ActionResult Math(int n1, int n2)
    {
        var x = n1 + n2;
        var result = n1 + " + " + n2 + " = " + x;
        var data = new { status = "ok", result = result };
        return Json(data, JsonRequestBehavior.AllowGet);
    }
}
```

**Kết quả thực hiện:**

### Ajax Demo

Number 1:

Number 2:

120 + 130 = 250

*Chú ý khi chúng ta **click** vào button Calculate, trang web **đường như đứng yên**, nhìn lên thanh địa chỉ của trình duyệt sẽ không thấy trang bị **load** lại.*