

BẢY BƯỚC THIẾT KẾ TRANG WEB ẨM TƯỢNG VÀ HIỆU QUẢ

Mặc dù việc thiết kế trang web đòi hỏi phải có nhiều khả năng về mỹ thuật, tuy nhiên vẫn có thể đưa ra một quá trình thiết kế để có thể giúp bạn tăng khả năng tạo ra các trang web hiệu quả và ấn tượng. Có thể nhiều người sẽ cho rằng việc đưa ra các bước trong quá trình thiết kế sẽ làm mất đi quá trình sáng tạo, điều này có lẽ đúng đối với những nhà thiết kế giỏi, những người có quá trình thiết kế và sáng tạo của riêng họ. Nhưng với những người không được đào tạo bài bản và thực hành nhiều thì việc nghiên cứu xem người khác thiết kế và ứng dụng các bước đó như thế nào có thể sẽ giúp ích rất nhiều cho họ.

Bước 1 : Phân tích người dùng và tự đánh giá chính mình

Nếu bạn không biết người dùng dự định là ai, thì tất cả việc thiết kế, cho dù có được thực hiện kỹ lưỡng đến đâu cũng chỉ dẫn đến thất bại. Bạn cần phải biết các thông tin về người dùng như trình độ, sở thích, các lĩnh vực quan tâm, cấu hình trang thiết bị, phần mềm, ... để tránh đưa ra một trang web vô tích sự.

Bạn cũng cần phải phân tích các mối quan tâm và khả năng của chính bạn. Bạn có khả năng thiết kế các trang web có hiệu quả và ấn tượng không? Bạn có đủ trình độ chuyên môn để tạo ra được các trang có lượng thông tin phong phú dựa trên các tài nguyên sẵn có không?

Sau đây là một số kỹ thuật giúp cho bước này :

1. Mô tả mục tiêu: Hãy xác định chính xác mục tiêu của trang này một cách ngắn gọn. Mục tiêu cần được mô tả một cách súc tích, rõ ràng, không quá rườm rà, chi tiết.
2. Xác định vấn đề giải quyết: Từ mô tả mục tiêu ở bước trên, nêu ra các vấn đề cần giải quyết để đạt được mục tiêu, tóm tắt phương pháp giải quyết, ...
3. Xác định người dùng: Liệt kê các đặc điểm của khách hàng như tuổi tác, nghề nghiệp, giới tính, thu nhập, trình độ, vùng cư trú, cấu hình trang thiết bị, phần mềm, ... vào một danh sách để phân tích và xử lý sau này.
4. Liệt kê các nguồn tài nguyên: Bạn có sẵn những gì để hoàn thành công việc cả về mặt trang thiết bị, công cụ phần mềm, ... và cả về trình độ chuyên môn ? Bạn có thể làm được những gì, và bạn sẽ nhờ giúp đỡ những gì ?
5. Xây dựng bảng tiến độ thực hiện: Xác định thời gian cần để hoàn thành sản phẩm với các tài nguyên sẵn có, thời gian cần để thực hiện từng bước của quá trình, ...

Bước 2 : Thiết kế các chức năng và cấu trúc trang

Có thể lúc này bạn rất muốn ngồi ngay vào máy và bắt tay vào việc xây dựng trang web nhưng đừng vội! Hãy dành thời gian cho việc thiết kế các chức năng và cấu trúc của các trang chính, vì

đây là bước quan trọng nhất trong quá trình thiết kế. Sau đây là một số hướng dẫn để thực hiện bước này :

1. Chọn cách làm việc sao cho có thể phác thảo thiết kế một cách thoải mái: Bạn có thể dùng bút để vẽ sơ đồ trên giấy, hay có thể dùng các chương trình máy tính để phác thảo. Tuy nhiên việc sử dụng các chương trình máy tính có thể sẽ làm hạn chế năng suất làm việc vì các công cụ có sẵn thường bị giới hạn.
2. Việc thiết kế nên đi từ trừu tượng đến cụ thể: Việc đưa ra các chi tiết ngay từ đầu có thể sẽ làm mất đi tổng quan của vấn đề. Phải xác định khung của chức năng trước rồi sau đó mới lựa chọn nội dung để điền vào.

Bước 3 : Tìm cách trình bày ấn tượng và hiệu quả

Ngay cả một cấu trúc tốt nhất cũng sẽ thất bại nếu việc trình bày thông tin không trực quan và ấn tượng. Một cách trình bày có ấn tượng và hiệu quả được đánh giá không chỉ bằng cách trông nó như thế nào mà còn xem nó đóng góp như thế nào vào quá trình đạt mục tiêu ban đầu.

Sau đây là một số cách để tìm ra nguồn cung cấp cho các trình bày tốt:

1. Đưa ra càng nhiều cách trình bày từ khả năng của chính bạn. Luôn quan sát và sưu tập các trình bày tốt đã đoạt giải, đã được nhiều người công nhận, hay các trình bày mà bạn thích, ...
2. Luôn cập nhật các thay đổi về công nghệ web. Bạn nên luôn có các tài liệu mới nhất về HTML, cũng như các thông tin về các dạng tập tin và các thiết bị mới được hỗ trợ bởi các nhà sản xuất.
3. Luôn ghi nhớ: Đối tượng đánh giá cách trình bày là người dùng chứ không phải bạn.
4. Thử càng nhiều giải pháp càng tốt và hãy ghi nhận các nhận xét, phản hồi của những người cộng tác để hoàn chỉnh thiết kế.

Bước 4 : Xây dựng nội dung

Là một người thiết kế trang web, bạn có thể có hoặc không chịu trách nhiệm tạo nội dung (như văn bản, hình ảnh, âm thanh, video, ..). Vì việc tạo nội dung thường không thể đợi đến lúc thiết kế hình thành, bạn có thể tiến hành các bước sau để đảm bảo rằng nội dung và thiết kế của bạn là tương thích với nhau:

1. Sửa đổi, hiệu chỉnh các nội dung đã có hoặc các nội dung mà bạn có quyền sửa.
2. Xin hỗ trợ và cố vấn của những chuyên gia đối với các chủ đề ngoài lĩnh vực chuyên môn của bạn.
3. Thiết lập đường dây liên lạc giữa bạn (người thiết kế) và những người tạo nội dung. Đưa ra các qui ước, các đặc tả cho nội dung như môi trường hỗ trợ, định dạng tập tin, cách nén, qui ước đặt tên tập tin, ...

4. Đảm bảo càng nhiều thông tin càng tốt. Cần phải có sự kết hợp chặt chẽ giữa văn bản và đồ họa, hình ảnh, âm thanh trong nội dung.
5. Tạo một cấu trúc cây thư mục hợp lý cho nội dung và thường xuyên sao lưu để đảm bảo an toàn.

Bước 5 : Thiết kế và kiểm tra khung trang web

Trong khi đang tiến hành xây dựng nội dung, đây là lúc kiểm tra các chức năng và cấu trúc được xây dựng trong bước 2 xem nó hoạt động như thế nào. Đây là bước mà bạn chuyển các mô tả về chức năng, về thiết kế ban đầu sang một thể hiện là các trang web cụ thể. Sau đây là một số hướng dẫn để thực hiện bước này :

1. Liên lạc với người quản trị server để xem việc tổ chức các tập tin như thế nào và các đặc tả nào có sẵn. Cho người quản trị biết các loại tập tin nào mà bạn đang sử dụng chưa được hỗ trợ.
2. Sử dụng các liên kết trong các trang tới các cấu trúc thư mục tương tự như cấu trúc thư mục trên server.
3. Ghi nhận các ảnh thường được dùng trong việc truy xuất các trang thông thường để đưa vào cache. Bằng cách này bạn có thể tăng tốc độ truy xuất các trang.
4. Thử nghiệm trên server để kiểm tra xem nó hoạt động đúng như thiết kế hay không.

Bước 6 : Đưa nội dung vào

Trong trường hợp tốt nhất, các khung dành cho văn bản và đồ họa sẽ được điền vào bằng nội dung thực sự của nó một cách dễ dàng và ăn khớp. Tuy nhiên điều này hiếm khi xảy ra vì một lẽ, hình ảnh và văn bản đưa vào có thể không vừa với khung thiết kế dành cho nó như dự định ban đầu. Cần phải thêm một số thao tác nữa mới có thể thực hiện xong chuyện này.

Để việc đưa nội dung vào thật sự đơn giản, ăn khớp, cần phải giữ mối liên lạc tốt giữa các thành viên liên quan như người thiết kế, người minh họa, người viết nội dung, người biên tập, và người quản trị server, ... Sau đây là một số hướng dẫn cho việc thực hiện tốt bước này :

1. Trước tiên hãy cho các trang hoạt động cục bộ, riêng lẻ để dễ kiểm tra, hiệu chỉnh, đánh giá, ...
2. Làm việc theo module, nghĩa là cho nhóm các trang liên quan nhau hoạt động trôi chảy trước khi mở rộng ra.
3. Đừng ngại thay đổi một quyết định thiết kế trước đó. Có thể bạn đã giả định sai, hoặc là công nghệ đã thay đổi vào lúc đưa nội dung vào, ...

Bước 7 : Kiểm tra và đánh giá

Các trang hiệu quả nhất là kết quả của việc thiết kế và đánh giá cẩn thận. Một web site trị giá nửa triệu đô la có thể có đến 70% tổng chi phí dành cho việc thiết kế và đánh giá. Sau đây là một số bước để thực hiện việc này:

1. Kiểm tra hoạt động của các liên kết nội bộ và các nguồn tài nguyên.
2. Kiểm tra độ chính xác của các liên kết ngoại. Không có gì tệ hơn là các liên kết với các trang bên ngoài không còn tồn tại nữa, hoặc là được chuyển đến nơi khác, hoặc là không còn phù hợp nữa.
3. Thử các trang với nhiều trình duyệt khác nhau. Thực hiện điều này để kiểm tra tính tương thích của trang với các trình duyệt, xem thử thiết kế trang đã tận dụng hết các hỗ trợ của trình duyệt chưa, ...
4. Thử các trang bằng nhiều cách kết nối khác nhau. Thử xem việc hiển thị các trang có ảnh hưởng như thế nào nếu kết nối bằng mạng cục bộ, đường kết nối tốc độ cao, đường điện thoại, ...
5. Thử các trang ở tình trạng mức độ truy cập cao. Nếu server của bạn chạy tốt trong các giờ cao điểm thì những giờ khác có thể chấp nhận được.
6. Thử các trang với nhiều dạng người dùng khác nhau. Nếu trang của bạn đề cập về các mối quan tâm chung thì hãy tranh thủ thử trang web với những đồng sự, bạn bè, ... Hãy ghi chú và quan sát. Có thể bạn sẽ không cần thay đổi phiên bản của trang web nhưng bạn sẽ cần các thông tin vì trang liên tục được cập nhật hóa.

Trên đây là các bước để giúp bạn có thể tạo các trang web tốt. Chúc các bạn thành công.

Lê Đình Duy

ldduy@fit.hcmuns.edu.vn

LẬP TRÌNH WEB ASP VỚI TIẾNG VIỆT UNICODE

Lê Đình Duy

Khoa CNTT - ĐHKHTN Tp. HCM

ldduy@fit.hcmuns.edu.vn

08.2002

1. Một số khái niệm căn bản về biểu diễn kí tự bên trong máy tính

1.1. Khái niệm về điểm mã, đơn vị mã, bảng mã

Về mặt bản chất, máy tính chỉ làm việc với các con số, do đó để biểu diễn các kí tự trên máy tính cần phải có một qui ước nhất quán giữa các kí tự cần biểu diễn và các con số tương ứng mà máy tính xử lí. Qui ước này được thể hiện qua các bước sau:

- Chọn tập các kí tự cần mã hóa (character set).
- Gán cho mỗi kí tự cần mã hóa một giá trị nguyên không âm, gọi là điểm mã (code point).
- Chuyển các điểm mã thành dãy các đơn vị mã (code units) để phục vụ cho việc lưu trữ và mã hóa. Một đơn vị mã là một đơn vị của bộ nhớ, có thể là 8, 16, hay 32 bit. Các điểm mã không nhất thiết phải có cùng số đơn vị mã.

Tập hợp những điểm mã của một tập các kí tự được gọi là một trang mã (code page) hay còn gọi là bảng mã hay bộ mã. Như vậy khi nói về một bảng mã, chúng ta quan tâm đến hai điều chính, số lượng các kí tự được mã hóa, và cách mã hóa chúng thành các đơn vị mã.

Lấy ví dụ bảng mã ASCII, tập kí tự cần mã hóa có 128 kí tự bao gồm các kí tự tiếng Anh, kí tự số, kí tự tiền tệ Anh, Mỹ và các kí tự điều khiển hệ thống ngoại vi. Các điểm mã có giá trị nằm trong khoảng từ 0-127. Mỗi điểm mã được mã hóa bằng đúng một đơn vị mã 8 bit, có nghĩa là đúng một byte.

Việc quyết định chọn cách mã hóa như thế nào sẽ quyết định số lượng kí tự được mã hóa. Ví dụ, nếu chọn cách mã hóa các điểm mã bằng đúng một đơn vị mã 8-bit thì số lượng điểm mã của một bảng mã (tạm gọi là bảng mã 8 bit) chỉ có thể tối đa là 256.

Do bảng mã ASCII không đủ để biểu diễn các kí tự của các ngôn ngữ khác, ví dụ như tiếng Việt, nên Microsoft đã mở rộng bảng mã ASCII bằng cách sử dụng 128 điểm mã có giá trị từ 128-255 để mã hóa cho các kí tự ngoài ASCII này. Tuy nhiên do chỉ có 128 điểm mã, trong khi số lượng các kí tự của các ngôn ngữ khác nhiều hơn, nên Microsoft đã tạo ra nhiều bảng mã khác nhau cho từng loại ngôn ngữ [1]. Ví dụ:

code page	1250	1251	1252	1253	1254	1258	etc.,
upper 128	Eastern Europe	Cyrillic	West Euro ANSI	Greek	Turkish	Vietnamese	etc.,
lower 128	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	etc.,

Tuy nhiên trong từng bảng mã này, không phải tất cả các kí tự của một ngôn ngữ đều có trong bảng mã. Hay nói chính xác hơn là không phải tất cả các kí tự đều được biểu diễn bằng duy nhất một điểm mã. Lấy ví dụ tiếng Việt chúng ta có 134 kí tự tổ hợp từ 28 chữ cái và 5 dấu thanh. Do chỉ có 128 điểm mã nên bảng mã windows-1258 dành cho tiếng Việt biểu diễn một số kí tự thành hai điểm mã liên tiếp, một điểm mã dành cho kí tự cơ sở và một điểm mã dành cho dấu thanh. Ví dụ: kí tự “é” được biểu diễn bằng hai điểm mã

tương ứng với các kí tự ê và kí tự dấu sắc: ê = ê + ´. Cách biểu diễn như vậy được gọi là cách biểu diễn tách rời (decomposed) mà thuật ngữ chúng ta hay gọi là tổ hợp.

Bảng mã TCVN3-ABC dùng 134 điểm mã để biểu diễn hết các kí tự tiếng Việt, chính điều này đã dẫn đến phải sử dụng một số điểm mã của bảng mã ASCII. Đây chính là lí do mà các trang web sử dụng bảng mã này không hiển thị được kí tự ư trong các trình duyệt Internet Explorer 5.0 trở lên. Cách biểu diễn như vậy được gọi là cách biểu diễn kết hợp sẵn (precomposed) mà thuật ngữ chúng ta hay gọi là dựng sẵn.

1.2. Bảng mã Unicode

Về mặt bản chất các bảng mã trên của Windows là bảng mã 8-bit, nghĩa là mỗi điểm mã được mã hóa bằng đúng một đơn vị mã 8-bit. Chính điều này đã giới hạn số lượng các kí tự được mã chỉ là 256. Do đó trong một văn bản không thể cùng hiển thị nhiều kí tự của các ngôn ngữ khác nhau được.

Unicode ra đời nhằm thống nhất chung các kí tự của mọi ngôn ngữ trong một bảng mã duy nhất [2]. Hai vấn đề nên lưu ý khi đề cập đến thuật ngữ Unicode đó là:

- Tập kí tự mà Unicode biểu diễn: ở đây muốn nói đến tập kí tự và cách ánh xạ các kí tự bằng các điểm mã tương ứng.
- Cách mã hóa các điểm mã thành các đơn vị mã.

Unicode dùng 16 bit để biểu diễn các điểm mã, do đó nó có thể biểu diễn được đến 65,536 kí tự có điểm mã nằm trong khoảng từ 0-65,535. Do vậy với Unicode người ta có thể biểu diễn được hầu hết các kí tự của các ngôn ngữ.

Cách đơn giản nhất để mã hóa các kí tự Unicode là biểu diễn mỗi điểm mã bằng đúng một đơn vị mã 16-bit. Đây chính là cách mã hóa nguyên thủy của Unicode trong phiên bản 2.0 được ISO/IEC chuẩn hóa thành ISO/IEC 10646 hay còn gọi là UCS-2. Tuy nhiên, để tương thích với các hệ thống xử lí trước khi Unicode ra đời cũng như tối ưu hóa trong quá trình lưu trữ và truyền dữ liệu, người ta dùng các cách khác nhau để mã hóa các điểm mã thành các đơn vị mã. Mỗi cách mã hóa như vậy được gọi là một dạng biến đổi của Unicode (UTF – Unicode Transformation Format). Thông dụng nhất hiện nay là UTF-8 và UTF-16 dùng dãy các đơn vị mã có độ dài khác nhau để mã hóa các điểm mã. UTF-8 dùng 1 đến 4 đơn vị mã 8-bit trong khi UTF-16 dùng 1 đến 2 đơn vị mã 16-bit để mã hóa. Ví dụ sau minh họa cách mã hóa của UTF-8:

- 128 kí tự đầu tiên của Unicode từ điểm mã U+0000 đến U+007F, được mã hóa thành 1 byte.
- Từ điểm mã U+0080 đến U+07FF, được mã hóa thành 2 byte.
- Từ điểm mã U+0800 đến U+FFFF, được mã hóa thành 3 byte.
- Từ điểm mã U+0800 đến U+FFFF, được mã hóa thành 4 byte.

Như vậy khi đề cập đến Unicode trong lập trình, cần phải xác định rõ chúng ta dùng bảng mã Unicode theo dạng biến đổi nào: UCS-2, UTF-8, hay UTF-16, ... UCS-2 được dùng trong các hệ quản trị cơ sở dữ liệu như SQL Server 7.0/2000, Microsoft Access 2000, UTF-8 thường được dùng trong các ứng dụng web, trong khi UTF-16 lại được dùng trong các hệ thống như Windows 2000/XP, Java, ...

2. Lập trình web với tiếng Việt Unicode

2.1. Chỉ định bảng mã dùng trong trang web

Khi một trang web được server chuyển xuống cho client, trình duyệt sẽ dùng thông tin về bảng mã mà trang web đó sử dụng để chuyển dãy các byte trong tài liệu đó thành các kí tự tương ứng để hiển thị lên màn hình. Ngoài ra, một khi dữ liệu trong các FORM được gửi đi sau khi người dùng submit, trình duyệt cũng sẽ căn cứ vào bảng mã này để chuyển đổi dữ liệu khi truyền đi. Ví dụ, nếu trang web được chỉ định dùng bảng

mã windows-1252 thì khi FORM được submit, dữ liệu sẽ được mã hóa theo bảng mã này cho dù trước đó trong các hộp điều khiển của FORM, dữ liệu được gõ dưới dạng Unicode [3].

Việc chỉ định bảng mã có vai trò rất quan trọng trong việc hiển thị đúng nội dung mà người thiết kế mong muốn, bởi vì nếu không chỉ định bảng mã được dùng trong trang web hiện hành một cách rõ ràng, trình duyệt sẽ sử dụng bảng mã mặc định. Ví dụ, nếu dữ liệu chuyển đến cho trang web là E1 BB 81, nếu chỉ định bảng mã là UTF-8 thì 3 byte này chính là biểu diễn mã của kí tự “è” trong khi nếu hệ thống dùng bảng mã mặc định, ví dụ như windows-1252, thì 3 byte này lại được xem như là biểu diễn 3 kí tự khác nhau và sẽ được hiển thị là “á»□”.

Để chỉ định bảng mã mà trang web hiện hành sử dụng, ta dùng tag META với thuộc tính HTTP-EQUIV được gán là Content-Type, và chỉ định tên của bảng mã được dùng trong thuộc tính CONTENT (Thông tin về các bảng mã được dùng trên Windows có thể xem tại [4]). Trong ví dụ sau, tag META được dùng để chỉ định bảng mã windows-1252 cho một trang web:

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; CHARSET=windows-1252">
```

Để yêu cầu trình duyệt sử dụng một bảng mã cho toàn bộ trang web, ta phải đặt tag META này trước tag BODY. Thông thường là đặt tag META này trong tag HEAD như ví dụ sau:

```
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; CHARSET=windows-1252">
<TITLE>New Page 1</TITLE>
</HEAD>
<BODY>
...
</BODY>
</HTML>
```

Trong trường hợp bảng mã được chỉ định không có khả năng biểu diễn được tất cả các kí tự của trang web, người ta phải dùng đến số tham chiếu của kí tự (NCRs - numerical character references). Số tham chiếu kí tự là điểm mã của kí tự Unicode tương ứng mà nó biểu diễn. Số tham chiếu kí tự có hai dạng thập phân và thập lục phân. Dạng thập phân có cú pháp là “&#D;”, với D là số thập phân. Dạng thập lục phân có cú pháp là “&#xH;”, với H là số thập lục phân. Ví dụ: å và å là các số tham chiếu của kí tự “a” trong bảng mã Unicode. Một khi gặp số tham chiếu của kí tự, trình duyệt sẽ tham chiếu trực tiếp đến kí tự có điểm mã tương ứng trong bảng mã Unicode mà không sử dụng đến bảng mã được chỉ định hiện hành [5].

Lấy ví dụ một trang web được mã hóa với bảng mã windows-1252, lúc đó để hiển thị đoạn văn bản: “Tiếng Việt”, dữ liệu cho trang web phải là “Tiếng Việt”, trong đó ế và ệ lần lượt là các số tham chiếu của các kí tự “é” và “ệ” trong bảng mã Unicode.

Điều này cho phép giải thích tại sao, các trang web không dùng bảng mã UTF-8, ví dụ như windows-1252, vẫn có thể hiển thị được các kí tự Unicode không thuộc bảng mã đó hay khi chuyển đổi từ bảng mã UTF-8 sang windows-1252, MS FrontPage 2000 lại tự động thêm vào các số tham chiếu kí tự theo cách trên.

2.2. Hoạt động của webserver

Khi trình duyệt yêu cầu một trang .asp, trình xử lí trang asp tại webserver sẽ thông dịch các mã lệnh ở trong trang web này và gửi kết quả về cho trình duyệt. Thông thường, lệnh Response.Write được dùng cho các kết xuất từ các hằng chuỗi hay từ các biến ra màn hình. Ví dụ như:

```

...
<%
    Response.Write "Chào mừng bạn đến với trang web này" 'in một hằng chuỗi
    Response.Write rs("TEN_NV") 'in dữ liệu của một biến, ví dụ như là một trường của recordset
%>
...

```

Để yêu cầu webserver mã hóa các dữ liệu trong các hằng chuỗi và biến theo bảng mã sẽ được dùng để hiển thị tại client, ta cần phải đặt thuộc tính CodePage về bảng mã tương ứng. Các lệnh trong ví dụ sau sẽ yêu cầu webserver mã hóa các chuỗi dữ liệu theo bảng mã UTF-8 (Thông tin về các codepage tương ứng với các bảng mã xem tại [\[4\]](#)):

```

<%Session.CodePage=65001%> // Dùng cho toàn bộ các trang trong Session hiện hành
<%@CodePage=65001%> // Dùng cho trang hiện hành

```

Lấy ví dụ trong trường hợp dùng cơ sở dữ liệu SQL Server 7.0, dữ liệu được trả về từ các câu truy vấn theo bảng mã UCS-2. Nếu ta chỉ định CodePage là 65001, webserver sẽ tự động chuyển dữ liệu từ UCS-2 sang UTF-8, ngược lại nếu không chỉ định thuộc tính CodePage, webserver sẽ chuyển dữ liệu đó đến client theo bảng mã mặc định (ví dụ như windows-1252). Điều này giải thích cho trường hợp một số trang web asp hiển thị không đúng dữ liệu Unicode được lưu trong các cơ sở dữ liệu như SQL Server 7.0/2000, MS Access 2000.

Ngoài ra, các trang asp có sử dụng đoạn mã lệnh thiết lập CodePage là 65001 phải được lưu theo định dạng tương ứng là UTF-8 [\[6\]](#).

Như vậy, việc thiết lập thuộc tính CodePage trong trang asp sẽ giúp cho webserver hiểu được các dữ liệu được lưu trong các cơ sở dữ liệu, hằng chuỗi kí tự, ... theo bảng mã nào để mã hóa (encode) nó trước khi chuyển đến cho trình duyệt. Việc chỉ định bảng mã dùng trong trang web bằng tag META sẽ giúp cho trình duyệt diễn dịch (decode) dữ liệu được chuyển đến từ webserver đúng nhất khi hiển thị [\[7\]](#).

2.3. Các bước cơ bản của lập trình web asp sử dụng tiếng Việt Unicode

- Soạn và lưu trữ tập tin .asp dưới dạng mã hóa UTF-8.
- Trong các tập tin asp, chèn các đoạn mã chỉ định cho web server và trình duyệt xử lí dữ liệu trong trang web như là UTF-8. Các đoạn mã này phải đặt ở đầu trang asp. Sử dụng ví dụ mẫu sau:

```

<%@CODEPAGE=65001%>
<%Session.Codepage=65001%>
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; CHARSET=UTF-8">
</HEAD>
<BODY>
</BODY>
</HTML>

```

- Sử dụng các hệ quản trị CSDL hỗ trợ Unicode như SQLServer 7.0/2000, MS Access 2000. Nếu dùng SQL Server thì phải khai báo kiểu dữ liệu cho các trường lưu dữ liệu Unicode là NCHAR, NVARCHAR, NTEXT, ... Các kiểu dữ liệu như TEXT, MEMO, HYPERLINK trong MS Access 2000 mặc định là hỗ trợ lưu dữ liệu Unicode.
- Truy xuất cơ sở dữ liệu thông qua JScript/VBScript/ODBC.
- Khi làm việc trên hệ quản trị CSDL SQL Server 7.0/2000, nếu dùng các hằng chuỗi trong các câu lệnh SQL, phải thêm tiếp đầu ngữ N (bắt buộc là chữ in hoa) vào [\[8\]](#). Nếu không sử dụng tiếp đầu

ngữ này, SQL Server sẽ tự động chuyển chuỗi dữ liệu sang bảng mã mặc định hiện hành trước khi sử dụng nó trong các thao tác cập nhật CSDL. Ví dụ, nếu bạn dùng câu lệnh sau: INSERT INTO SINHVIEN(TEN_SV) VALUES('Trần Nam Hải') thì hằng chuỗi dữ liệu 'Trần Nam Hải' sẽ được SQL Server xem như là chuỗi kí tự thường chứ không phải là chuỗi Unicode. Điều này sẽ dẫn đến hậu quả là dữ liệu sẽ được lưu trữ không chính xác. Ví dụ như dữ liệu của kí tự "à" trong chuỗi trên là E1 BA A7, sẽ được lưu thành 3 kí tự khác nhau. Trong khi đó nếu dùng câu lệnh INSERT INTO SINHVIEN(TEN_SV) VALUES(N'Trần Nam Hải') thì 3 byte E1 BA A7 sẽ được xem như là một kí tự khi lưu xuống [9].

3. Hỗ trợ Unicode của các phần mềm

3.1. Các phần mềm hỗ trợ soạn thảo trang web

- Visual Studio.NET, Notepad, MS FrontPage2002: Hỗ trợ lưu tập tin dưới dạng UTF-8
- Visual InterDev 6.0: Nếu trong trang asp ta sử dụng các hằng chuỗi được gõ vào dưới dạng Unicode, ví dụ như: Response.Write "Chào mừng bạn " thì lúc lưu tập tin, chương trình sẽ phát hiện ra trong trang asp này có xuất hiện kí tự Unicode và yêu cầu lưu xuống dưới dạng Unicode, nếu không các kí tự Unicode sẽ bị mất. Tuy nhiên, nếu chọn lưu dưới dạng Unicode thì chương trình sẽ lưu tập tin này dưới dạng mã hóa UCS-2. Hiện nay webserver IIS không thể xử lí được trang asp này [10]. Do đó không nên dùng Visual InterDev 6.0 để soạn thảo các trang asp trong các ứng dụng Unicode tiếng Việt.
- Các phần mềm thông dụng hỗ trợ gõ tiếng Việt Unicode: UniKey, VietKey.

3.2. Các phần mềm hệ thống khác

- SQL Server 7.0/2000 và MS Access 2000 hỗ trợ Unicode. Với mỗi kí tự Unicode, hệ thống sẽ sử dụng bảng mã UCS-2 để lưu trữ, nghĩa là dùng cố định 2 byte cho một kí tự. SQL 6.5 và MS Access 97 không hỗ trợ Unicode.
- IIS 5.0 không thể đọc được các tập tin lưu dưới dạng UCS-2 [10], không hỗ trợ CodePage của bảng mã UTF-16 là 1200 [11]. IIS 4.0 không hỗ trợ CodePage của bảng mã UTF-8 là 65001 [11].

Tóm lại

Unicode ra đời nhằm khắc phục hạn chế về số lượng kí tự được mã hóa của các bảng mã 8-bit trước đó, cho phép mọi ngôn ngữ có thể sử dụng chung một bảng mã duy nhất. Do vấn đề tương thích trong lưu trữ và truyền dữ liệu mà Unicode có các dạng mã hóa khác nhau như UCS-2, UTF-8, UTF-16. UTF-8 là dạng mã hóa Unicode thông dụng nhất trong các ứng dụng web hiện nay.

Để viết các ứng dụng web dùng tiếng Việt Unicode, cần chọn các phần mềm soạn thảo hỗ trợ lưu trữ tập tin dưới dạng mã hóa UTF-8 như Visual Studio.NET, MS FrontPage2000, NotePad, ... ; sử dụng các hệ quản trị CSDL hỗ trợ Unicode như SQL Server 7.0/2000, MS Access 2000, ...; đặt các đoạn mã chỉ định bảng mã mà webserver và trình duyệt dùng để mã hóa và giải mã dữ liệu.

Tài liệu trích dẫn

1. <http://www.microsoft.com/globaldev/articles/unicode.asp>
2. <http://www.unicode.org/unicode/standard/principles.html>
3. <http://support.microsoft.com/default.aspx?scid=kb;en-us;Q303612>
4. <http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/reference/charsets/charset4.asp>
5. <http://www.w3.org/TR/html401/charset.html#code-position>

6. <http://support.microsoft.com/default.aspx?scid=KB;EN-US;Q295063&>
7. <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/iisref/html/psdk/asp/vbob150l.asp>
8. <http://support.microsoft.com/default.aspx?scid=kb;en-us;Q239530>
9. <http://support.microsoft.com/default.aspx?scid=kb;en-us;q232580>
10. <http://support.microsoft.com/default.aspx?scid=kb;en-us;Q245000>
11. [http://support.microsoft.com/default.aspx?scid=kb;\[LN\];Q254313](http://support.microsoft.com/default.aspx?scid=kb;[LN];Q254313)

HẾT

KẾT XUẤT DỮ LIỆU RA DẠNG EXCEL TỪ TRANG ASP

Thông thường, các ứng dụng web kết xuất dữ liệu của các báo cáo ra dưới dạng bảng biểu. Sẽ rất cần thiết nếu người dùng cũng nhận được dữ liệu này dưới dạng Excel để có thể sử dụng cho các mục đích khác.

Kỹ thuật để đạt được mục đích này khá đơn giản. Ý tưởng chính của kỹ thuật này là sử dụng thuộc tính ContentType của đối tượng Response trong ASP và thực hiện theo các bước tuần tự sau:

Bước 1: Chỉ định dữ liệu sẽ được chuyển đi theo định dạng Excel

Đơn giản chỉ cần dùng câu lệnh: Response.ContentType = "application/vnd.ms-excel". Thông thường, cần phải đặt câu lệnh Response.Buffer = True và Response.Clear trước câu lệnh này để đảm bảo dữ liệu được chuyển xuống client chính xác.

Bước 2: Kết xuất dữ liệu dưới dạng bảng theo cách làm thông thường.

Dòng đầu tiên của bảng chứa tên của các cột sẽ được hiển thị trong tập tin excel.

Hãy xem ví dụ minh họa sau:

Data2Excel.asp

```
<%
    Response.Buffer = True
    Response.Clear
    ' thiết lập định dạng sẽ kết xuất là Excel
    Response.ContentType = "application/vnd.ms-excel"

    ' kết nối với CSDL
    strDSN = "DRIVER=Microsoft Access Driver (*.mdb);DBQ="
    ' đường dẫn tương đối đến tập tin CSDL
    strDSN = strDSN & Server.MapPath("myDB.mdb")
    set Conn = Server.CreateObject("ADODB.Connection")
    Conn.Open strDSN
    strSQL = "SELECT * FROM myTable"
    set rs = Conn.Execute(strSQL)
    rowstart = "<tr>"
    rowend = "</tr>"
    cellstart = "<td>"
    cellend = "</td>"
    Response.Write "<TABLE border=1>"
    ' dòng đầu tiên in tên các cột
    Response.Write rowstart
    For i = 0 to rs.Fields.Count -1
        Response.Write cellstart & "<b>" & rs.Fields(i).name & "</b>" & cellend
    Next
    Response.Write rowend
    ' in dữ liệu của từng dòng
    Do while not rs.EOF
        Response.Write rowstart
        For i = 0 to rs.Fields.Count -1
```

```

Response.Write cellstart & rs.Fields(i)& cellend
Next
Response.Write rowend
rs.MoveNext
Loop
rs.Close
set rs = Nothing
Conn.Close
set Conn = nothing
Response.Write "</TABLE>"
Response.End
%>

```

Bàn luận:

Ưu điểm của cách làm trên là cho phép bạn sử dụng các đoạn mã sẵn có lúc kết xuất dữ liệu ra dạng bảng theo cách thông thường để chuyển sang định dạng Excel. Tuy nhiên việc kết xuất dữ liệu theo định dạng Excel như trong ví dụ trên có thể chiếm tài nguyên của webserver đặc biệt khi dữ liệu lớn do đó chỉ nên dùng cách này nếu trang này không được sử dụng thường xuyên.

Nếu muốn kết xuất dữ liệu lớn và thực hiện thường xuyên, ta có thể kết xuất thông qua định dạng CSV (Comma-Separated Values) để tối ưu hơn. Các tập tin theo định dạng CSV là các tập tin văn bản mà dữ liệu trong các cột được ngăn cách với nhau bởi dấu phẩy “,” (comma), rất thường được dùng cho việc trao đổi dữ liệu giữa các hệ quản trị CSDL và các chương trình bảng tính như Excel. Ví dụ, nếu bạn mở một tập tin csv có nội dung như sau trong Excel, ta sẽ nhận được một bảng 3 dòng, 3 cột:

Doe,John,944-7077

Johnson,Mary,370-3920

Smith,Abigail,299-3958

Để chuyển dữ liệu sang định dạng CSV, vẫn với cách làm tương tự bằng cách thay đổi thuộc tính ContentType về dạng “application/csv” và thêm dòng lệnh sau để yêu cầu trình duyệt hiển thị hộp thoại tải tập tin về: Response.AddHeader "Content-Disposition", "filename=mydata.csv;". Sau đó, thay vì định dạng dữ liệu dưới dạng bảng, ta định dạng dữ liệu theo dạng dữ liệu các cột được phân cách với nhau bằng dấu phẩy “,”. Xem ví dụ minh họa sau:

```

Data2CSV.asp
<%
Response.Buffer = True
Response.Clear
' thiết lập định dạng sẽ kết xuất là Excel
Response.ContentType = "application/csv"
Response.AddHeader "Content-Disposition", "filename=mydata.csv;"

' kết nối với CSDL
strDSN = "DRIVER=Microsoft Access Driver (*.mdb);DBQ="

```

```

' đường dẫn tương đối đến tập tin CSDL
strDSN = strDSN & Server.MapPath("myDB.mdb")
set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open strDSN
strSQL = "SELECT * FROM myTable"
set rs = Conn.Execute(strSQL)
' dòng đầu tiên in tên các cột
For i = 0 to rs.Fields.Count -1
    Response.Write rs.Fields(i).name & ", "
Next
Response.Write vbNewLine ' kết thúc một dòng dữ liệu
' in dữ liệu của từng dòng
Do while not rs.EOF
    For i = 0 to rs.Fields.Count -1
        Response.Write rs.Fields(i)& " , "
    Next
    Response.Write vbNewLine ' kết thúc một dòng dữ liệu
    rs.MoveNext
Loop
rs.Close
set rs = Nothing
Conn.Close
set Conn = nothing
Response.End
%>

```

Hạn chế của định dạng này là do dữ liệu chỉ là các kí tự ASCII nên sẽ không dùng được trong trường hợp dữ liệu sử dụng Unicode, ngoài ra cần phải có cách xử lí thích hợp trong trường hợp dữ liệu trong các cột có dấu phẩy “,”. Excel xử lí trường hợp này bằng cách thay đặt toàn bộ dữ liệu trong dấu “””. Ví dụ nếu bạn có dữ liệu là **Abc, xyz** thì dữ liệu sẽ được đổi thành **“Abc, xyz”**

Tham khảo thêm tại:

http://www.web-savant.com/users/kathi/asp/samples/tut/Export_to_Excel.asp

http://gethelp.devx.com/techtips/asp_pro/10min/10min0699.asp

Lê Đình Duy – ldduy@fit.hcmuns.edu.vn

HIỂN THỊ HÌNH ẢNH TỪ CƠ SỞ DỮ LIỆU TRONG CÁC ỨNG DỤNG WEB

Lê Đình Duy

Khoa CNTT – ĐHKHTN Tp.HCM

ldduy@fit.hcmuns.edu.vn

12.2002

1. Giới thiệu

Việc lưu trữ và hiển thị hình ảnh từ CSDL trong các ứng dụng web là cần thiết. Ví dụ như các ứng dụng quản lí hồ sơ của nhân viên, ngoài thông tin bằng văn bản như Họ tên, địa chỉ, bằng cấp, ... sẽ rất cần thiết nếu có thêm dữ liệu về ảnh của nhân viên. Hay trong các ứng dụng bán hàng, bên cạnh thông tin về sản phẩm như Tên sản phẩm, phân loại sản phẩm, giá cả, ... hình ảnh trực quan về sản phẩm cũng rất cần thiết để giúp cho người dùng dễ dàng hơn khi chọn lựa, ...

Thông thường người ta dùng một trong hai cách để đạt được mục đích này. Cách thứ nhất là lưu trữ tập tin hình ảnh trên một thư mục riêng, trong CSDL ngoài dữ liệu văn bản chỉ lưu đường dẫn đến tập tin hình ảnh. Cách thứ hai là lưu trữ cả dữ liệu hình ảnh và văn bản trong cùng một bản ghi trong CSDL.

Cách tiếp cận thứ nhất thường được dùng hơn vì CSDL có kích thước nhỏ và thao tác hiển thị khá đơn giản thông qua tag IMG với thuộc tính SRC được gán bằng thông tin về đường dẫn đến tập tin hình ảnh đã có trong CSDL. Hạn chế của cách tiếp cận này là đòi hỏi dữ liệu ảnh phải được lưu trữ trên webserver hoặc ở một máy tính nào đó mà webserver có thể truy cập được. Trong các ứng dụng mà cơ sở dữ liệu có thể được sao lưu nhiều bản để phân tán, "rủi ro" sẽ xảy ra nếu các tập tin hình ảnh không được sao lưu theo đúng đường dẫn sẵn có.

Cách tiếp cận thứ hai do lưu trữ hình ảnh trong CSDL như là dữ liệu nhị phân nên sẽ dẫn đến hạn chế là làm cho kích thước của CSDL tăng lên đáng kể. Nhưng bù lại, dữ liệu hình ảnh và văn bản ở chung một nơi nên có thể dùng cho các CSDL được sao lưu nhiều nơi. Để lưu trữ dữ liệu hình ảnh trên các hệ quản trị CSDL ta phải dùng các kiểu dữ liệu dạng nhị phân cho nó. Ví dụ, trong MS SQL Server là kiểu dữ liệu image, trong MS Access là kiểu dữ liệu OLE Object.

2. Hiển thị hình ảnh có trong CSDL

Nếu dùng cách tiếp cận thứ nhất, ta sẽ tạo ra trong bảng dữ liệu một trường tên chẳng hạn là IMAGE_URL để lưu trữ đường dẫn đến tập tin hình ảnh, ví dụ như là: images/id1234.gif. Để hiển thị hình ảnh này trong trang web, đơn giản chỉ cần dùng tag IMG với thuộc tính SRC được gán bằng dữ liệu trong trường ImageURL như: `Response.Write("")`.

Nếu dùng cách tiếp cận thứ hai, ta sẽ tạo ra trong bảng dữ liệu một trường có tên chẳng hạn là APP_IMGDATA để lưu trữ dữ liệu hình ảnh dưới dạng nhị phân. Sau đó, để hiển thị hình ảnh này lên, ta thực hiện tuần tự các bước sau:

- Đọc dữ liệu hình ảnh lưu trong ImgData vào recordset.
- Gán thuộc tính ContentType của đối tượng Response tương ứng với định dạng ảnh lưu trữ. Nếu lưu tập tin dưới dạng .gif, ta đặt `Response.ContentType="image/gif"`.
- Sử dụng hàm `Response.BinaryWrite` để ghi nội dung dữ liệu của hình ảnh ra.

Giả sử ta dùng CSDL là MS Access với tập tin CSDL là DB_USERS được lưu trong thư mục APP_DB, bảng dữ liệu APP_USERS được dùng để lưu thông tin của người dùng và hình ảnh tương ứng. Đoạn mã sau của tập tin showimage.asp minh họa các bước trên:

```
showimage.asp
<%
    ' kết nối với CSDL
    strDSN = "DRIVER=Microsoft Access Driver (*.mdb);DBQ="
    ' đường dẫn tương đối đến tập tin CSDL
    strDSN = strDSN & Server.MapPath("app_db/db_users.mdb")
    set Conn = Server.CreateObject("ADODB.Connection")
    Conn.Open strDSN
    ' giả sử cần hiển thị hình ảnh lưu trong bản ghi
    ' có trường APP_USERID bằng vUserID
    strSQL = "SELECT * FROM APP_USERS WHERE "
    strSQL = strSQL & " APP_USERID = " & vUserID
    ' đọc dữ liệu hình ảnh vào recordset
    Set rs = Conn.Execute(strSQL)
    ' gán thuộc tính ContentType là image/gif
    Response.ContentType = "image/gif"
    ' ghi dữ liệu hình ảnh ra tập tin chuyển xuống client
    Response.BinaryWrite(rs("APP_IMGDATA"))
    rs.close
    set rs = nothing
    Conn.close
    set Conn = nothing
%>
```

Với đoạn mã trên do ta đã đặt nội dung chuyển xuống cho trình duyệt là image/gif nên không thể vừa cùng hiển thị dữ liệu văn bản vừa hiển thị hình ảnh được.

3. Tải tập tin ảnh lên CSDL

Để có thể tải các tập tin dữ liệu lên server, thông thường ta dùng các component đã được viết sẵn để hỗ trợ cho việc này. Một trong các component cung cấp miễn phí là [aspSmartUpload](http://www.aspsmart.com/aspSmartUpload/) (<http://www.aspsmart.com/aspSmartUpload/>). Hai vấn đề cần lưu ý khi sử dụng các component dạng này là form trong trang dành cho người dùng nhập phải được đặt thuộc tính ENCTYPE là multipart/form-data và trang xử lý upload phải dùng đối tượng Form của các component để lấy dữ liệu về thay cho Request.Form. Với aspSmartUpload ta có thể tải đồng thời nhiều tập tin lên server, thậm chí có thể hạn chế kích thước tập tin, kiểu tập tin, ... sẽ được dùng để tải. Ví dụ sau minh họa việc tải dữ liệu lên CSDL bằng cách dùng tập tin upload.htm cho phép người dùng chỉ định tập tin cần upload, tập tin upload.asp dùng để lưu dữ liệu tập tin cần upload vào một trường trong bảng CSDL hoặc lưu thành một tập tin trong thư mục nào đó:

```
upload.htm
<H1>aspSmartUpload : Sample </H1>
<HR>

<FORM method="POST" action="upload.asp" enctype="multipart/form-data">
    <input type="FILE" name="FILE1" ><br>
    <input type="submit" value="Upload">
</FORM>
```

```

upload.asp
<%
' tạo đối tượng aspSmartUpload
Set mySmartUpload = Server.CreateObject("aspSmartUpload.SmartUpload")
' upload tập tin
mySmartUpload.Upload
' kết nối với CSDL
strDSN = "DRIVER=Microsoft Access Driver (*.mdb);DBQ="
' đường dẫn tương đối đến tập tin CSDL
strDSN = strDSN & Server.MapPath("app_db/db_users.mdb")
set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open strDSN

set rs = Server.CreateObject("ADODB.Recordset")
set rs.ActiveConnection = Conn
rs.Source = "app_users"
rs.LockType = 3
rs.Open
rs.close
For each file In mySmartUpload.Files
    If NOT file.IsMissing then ' nếu tập tin tồn tại
        rs.AddNew
        ' lưu dưới dạng một field trong CSDL
        file.FileToField rs.Fields("APP_IMGDATA")
        ' lưu dưới dạng một tập tin trên thư mục của server
        ' với đường dẫn tuyệt đối
        file.SaveAs("c:\temp\" & file.FileName)
        ' lưu dưới dạng một tập tin trên thư mục của server
        ' với đường dẫn tương đối so với thư mục webroot
        file.SaveAs("dbimages/upload" & file.FileName)
        ...
        rs.Update
    End if
Next
rs.close
set rs = nothing
Conn.close
set Conn = nothing
%>

```

4. Kết luận

Bài viết vừa trình bày các hai thao tác cơ bản cho việc quản lí hình ảnh trong cơ sở dữ liệu của các ứng dụng web. Thao tác thứ nhất liên quan đến việc chọn hình thức lưu trữ dữ liệu ảnh, thao tác thứ hai liên quan đến việc tải các hình ảnh lên server. Lưu trữ dữ liệu ảnh dưới dạng là một trường cơ sở dữ liệu sẽ làm cho kích thước cơ sở dữ liệu lớn, việc xử lí khá phức tạp nhưng có thuận lợi trong trường hợp ứng dụng được thiết kế cho việc sử dụng phân tán. Đơn giản và thường được dùng hơn cả là lưu trữ các tập tin hình ảnh trong một thư mục định trước trên server và lưu trữ đường dẫn đến tập tin này trong cơ sở dữ liệu.

BẮT ĐẦU HỌC LẬP TRÌNH ỨNG DỤNG WEB VỚI ASP

Lê Đình Duy
Khoa CNTT – ĐHKHTN Tp.HCM
ldduy@fit.hcmuns.edu.vn
07.2002

1. Giới thiệu ASP

Microsoft Active Server Pages (ASP) là môi trường lập trình ứng dụng phía server (server side scripting) hỗ trợ mạnh trong việc xây dựng các ứng dụng Web. Các ứng dụng ASP có thể làm việc với bất kỳ cơ sở dữ liệu nào tương thích với ODBC như SQL, Access, Oracle, Informix,... đồng thời rất dễ viết và sửa đổi. Hơn nữa nó có thể tích hợp các công nghệ sẵn có của Microsoft như COM, ... một cách dễ dàng.

Để có thể chạy được các trang web viết bằng ASP, cần phải có webserver hỗ trợ ASP. Microsoft IIS và Personal Web Server (PWS) trên Win95,98,NT hay Internet Information Server (IIS) trên Windows2000 là các webserver của Microsoft hỗ trợ ASP. Trong trường hợp webserver không phải của Microsoft, hay hệ điều hành không phải là Windows mà là Unix, Linux, cần phải cài đặt một thư viện hỗ trợ ASP. Thông dụng nhất là Sun Chili!Soft (<http://www.chillisoft.com>)

Để soạn thảo các trang ASP, ta có thể dùng bất cứ phần mềm soạn thảo văn bản nào, ví dụ như Notepad. Thông dụng và dễ dùng thường là Visual InterDev trong bộ Microsoft Studio.

Ngoài ra với ứng dụng có liên kết với cơ sở dữ liệu, cần phải cài đặt thêm các phần mềm cơ sở dữ liệu như Access, SQL, Oracle, ... Phần mềm cơ sở dữ liệu đơn giản nhất cho người mới bắt đầu là Access.

Để có thể viết ứng dụng web bằng ASP, cần phải biết các kiến thức cơ bản sau:

- Kiến thức về thiết kế web, HTML để giúp thiết kế các trang web.
- Kiến thức về các ngôn ngữ lập trình VB script, Java script. VB Script là ngôn ngữ lập trình thông dụng cho ASP.
- Thông thường các ứng dụng web có liên quan nhiều đến việc quản lí, truy xuất, cập nhật cơ sở dữ liệu nên cần phải nắm thêm kiến thức về cú pháp các câu truy vấn SQL, kiến thức về kết nối và lập trình cơ sở dữ liệu với ADO.

2. Một số kiến thức cơ bản về ASP

Một trang ASP thường có một số đặc điểm sau:

- Là một tập tin văn bản (text file) có phần mở rộng là **.asp**: Phần mở rộng này sẽ giúp webserver yêu cầu trình xử lí trang asp (ASP engine) trước khi trả về cho trình duyệt.
- Ngôn ngữ script thông dụng nhất dùng để viết các mã của ASP là VBScript. Ngoài ra, ta cũng có thể viết các mã bằng các ngôn ngữ như JScript, Perl, Python, ... nếu trên webserver có cài đặt các bộ xử lí ngôn ngữ này (script engine).
- Các đoạn mã viết trong trang ASP sẽ được các bộ xử lí ngôn ngữ trên webserver xử lí tuần tự từ trên xuống dưới. Kết quả của việc xử lí này là trả về trang HTML cho webserver và webserver sẽ gửi trang này về cho trình duyệt. Đó là lí do tại sao, tại trình duyệt ta không thể thấy được các đoạn mã chương trình đã được viết trong trang ASP.
- Một trang ASP thông thường gồm có 4 thành phần:
 - o Dữ liệu văn bản (text)

- Các tag HTML
- Các đoạn mã chương trình phía client đặt trong cặp tag <SCRIPT> và </SCRIPT>
- Mã chương trình ASP được đặt trong cặp tag <% và %>:

Ba thành phần ban đầu là cấu trúc của một trang HTML thông thường, do đó có thể xem một trang ASP là một trang HTML được nhúng thêm phần xử lý viết bằng mã ASP (VBScript, JScript, Perl, ...). Ví dụ sau minh họa một trang ASP, dữ liệu văn bản là "Welcome to my website. Today is:", các tag HTML là <P>, , ... và đoạn mã chương trình đặt giữa <% và %>

```
<HTML>
<BODY>
<P> <B>Welcome to my website</B>. Today is
<%
    Response.Write Date()
%>
</BODY>
</HTML>
```

3. Một số sách, website tham khảo

- Thiết kế và Lập trình ứng dụng web bằng ASP – Lê Đình Duy – NXB Thống kê, 2001
- Xây dựng trang web động với ASP – Nhóm tác giả ELICOM - 2001
- ASP Databases – Nhóm tác giả SAIGONBOOK - 2001
- Professional Active Server Pages 3.0 – Alex Homer et al - 1999
- MSDN – Active Server Pages Tutorial
- <http://www.learnasp.com>
- <http://www.4guysfromrolla.com>
- <http://www.15seconds.com>
- <http://asp.superexpert.com>
- <http://www.aspfaqs.com>

PHƯƠNG PHÁP HỖ TRỢ GIỚI HẠN TRUY CẬP TRANG WEB

Lê Đình Duy

Khoa CNTT – ĐHKHTN Tp.HCM

ldduy@fit.hcmuns.edu.vn

09.2002

1. Giới thiệu

Thông thường, trong các ứng dụng web, người thiết kế muốn giới hạn sự truy cập đến một số trang web thông qua việc chứng thực người dùng (authentication) nhằm mục đích cho phép những người có quyền thực sự mới được phép truy cập và thực hiện một số trang web nào đó. Ví dụ các trang web dùng cho việc cập nhật CSDL từ xa chỉ cho phép người quản trị thực hiện hay trong các ứng dụng như diễn đàn thảo luận, thông thường các trang gửi bài mới chỉ cho phép những người đã đăng kí thực hiện mà thôi, ...

Để đạt được mục đích này, có hai cách tiếp cận:

- Dùng chức năng bảo mật của hệ thống: Cách này giới hạn quyền truy cập đến các trang web cần bảo vệ bằng quyền trên hệ thống tập tin NTFS. Ví dụ, nếu muốn giới hạn quyền truy cập đến tập tin admin.asp, ta xác lập quyền cho một người dùng nào đó được quyền đọc, thi hành mà thôi. Cách này có hạn chế là người dùng trang web phải có tài khoản trên server. Điều này sẽ thực sự khó khăn khi đa số các ứng dụng web thường được hosting tại các server của các ISP.
- Dùng các đoạn mã chương trình tự viết: Cách này sử dụng cookies (thông qua biến kiểu Session) kết hợp với CSDL về người dùng để làm việc này! Cách làm này cho phép đáp ứng khá hoàn hảo nhu cầu bảo mật các trang web và tương thích dễ dàng trong trường hợp hosting ở các server khác nhau.

2. Bảo vệ bằng các đoạn mã chương trình tự viết

Ý tưởng chính của cách làm này là ta sẽ dùng một biến Session có kiểu là boolean để lưu thông tin về người dùng đã được chứng thực hay chưa. Giả sử ta đặt tên cho biến này là blLoginOK, giá trị True sẽ tương ứng với người dùng đã được chứng thực và ngược lại.

Việc chứng thực người dùng sẽ được thông qua một trang đăng nhập (ví dụ là trang login.htm). Trang này sẽ yêu cầu người dùng nhập thông tin về tên đăng nhập và mật khẩu. Sau đó một đoạn mã (trang login.asp) sẽ được dùng để kiểm tra thông tin người dùng vừa nhập có trùng khớp với dữ liệu được lưu trữ trên CSDL hay không. Nếu thông tin trùng khớp, giá trị của biến blLoginOK sẽ được chuyển thành True (lưu ý ta phải thiết lập biến blLoginOK có giá trị mặc định là False).

Trong các trang cần hạn chế truy cập, ta chỉ cần kiểm tra giá trị của biến này là True hay False. Nếu giá trị là True, người dùng sẽ được phép thực hiện tiếp các đoạn mã tiếp theo của trang, còn ngược lại, ta sẽ thông báo yêu cầu người dùng chứng thực thông qua một trang đăng nhập trước khi tiếp tục.

Các bước thực hiện tuần tự như sau:

Bước 1: Tạo cơ sở dữ liệu chứa thông tin về người dùng

Giả sử ta dùng MS Access để tạo cơ sở dữ liệu có tên là DB_USERS.MDB, trong đó ta tạo một bảng dữ liệu có tên là APP_USERS. Hai trường chính của bảng dữ liệu này là APP_USERNAME và APP_PASSWORD. Nếu đặt APP_USERNAME như là khóa chính thì một người dùng sẽ được xác định bằng một tên đăng nhập

duy nhất. Tất nhiên, ta có thể tạo thêm các trường khác để quản lí như Họ Tên, Địa chỉ Email (có thể sẽ cần để gửi email khi quên mật khẩu), Lần đăng nhập cuối cùng, Thời gian sử dụng hệ thống, ... Sau khi tạo xong, giả sử tập tin này được lưu tại thư mục APP_DB.

Bước 2: Tạo trang đăng nhập login.htm để yêu cầu người dùng nhập thông tin về tên đăng nhập và mật khẩu:

```
Login.htm
<FORM action="login.asp" method="POST">
Username: <input type="text" name="fmUserName" size="20"><br>
Password: <input type="password" name="fmPassword" size="20"><br>
<input type="submit" value="Submit" name="btnSubmit">
<input type="reset" value="Reset" name="btnReset">
</FORM>
```

Bước 3: Tạo trang login.asp để kiểm tra thông tin người dùng vừa nhập có trùng khớp với thông tin có sẵn trên CSDL hay không. Nếu trùng khớp, giá trị biến blLoginOK sẽ được chuyển thành True.

```
Login.asp
<%
    On Error Resume Next
    vUserName = Request.Form("fmUserName")
    ' Thay thế dấu nhảy đơn ' thành hai dấu nhảy đơn để tránh lỗi SQL injection
    vUserName = Replace(vUserName, "'", "'")
    vPassword = Request.Form("fmPassword")
    vPassword = Replace(vPassword, "'", "'")

    strDSN = "DRIVER=Microsoft Access Driver (*.mdb);DBQ="
    ' đường dẫn tương đối đến tập tin CSDL
    strDSN = strDSN & Server.MapPath("app_db/db_users.mdb")
    set Conn = Server.CreateObject("ADODB.Connection")
    Conn.Open strDSN

    strSQL = "SELECT * FROM APP_USERS WHERE "
    strSQL = strSQL & " APP_USERNAME = " & "'" & vUserName & "'"
    strSQL = strSQL & " AND " & "APP_PASSWORD = " & "'" & vPassword & "'"
    Set rs = Conn.Execute(strSQL)

    if rs.eof then ' người dùng không hợp lệ
        Response.Redirect("login.htm")
    else
        Session("blLoginOK") = True
    end if
    set rs = nothing
    set Conn = nothing
%>
```

Bước 4: Trong các trang web ví dụ như Admin.asp mà ta chỉ muốn những người đã được chứng thực mới được quyền sử dụng, đặt đoạn mã kiểm tra biến blLoginOK là True hay False ngay đầu trang:

```
Admin.asp
<%
    if (Session("blLoginOK") <> True) then
        Response.Redirect("login.htm")
    end if
%>
...
```

3. Kết luận

Nhu cầu hạn chế người dùng truy cập đến một số trang web nào đó trong ứng dụng là một nhu cầu thường xuyên khi xây dựng các ứng dụng. Bằng cách sử dụng biến Session và CSDL của người dùng cùng với các trang login.htm, login.asp, ta có thể đạt được mục đích trên một cách dễ dàng.

BẢO VỆ CƠ SỞ DỮ LIỆU ACCESS TRONG CÁC ỨNG DỤNG WEB

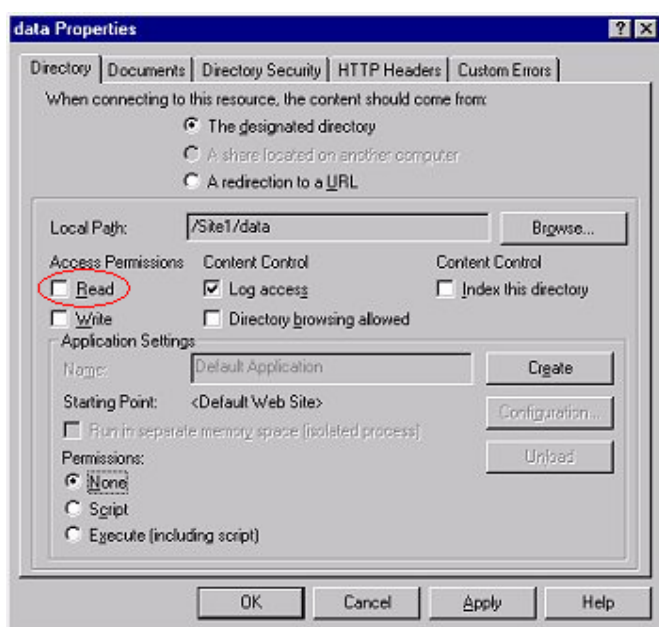
Các ứng dụng web sử dụng CSDL Access thường hay đặt tập tin CSDL .mdb vào một thư mục có thể truy cập được từ web, ví dụ như: D:\inetpub\wwwroot\myDB.mdb. Điều nguy hiểm nhất theo cách làm thông thường này là nếu người dùng biết được hay đoán được đường dẫn đến tập tin .mdb, họ có thể tải tập tin CSDL đó về và toàn bộ thông tin lưu trữ trên CSDL bị đánh cắp.

Để bảo vệ CSDL Access trong các ứng dụng web, nên kết hợp các phương án an toàn sau:

Phương án 1: Đặt tập tin CSDL .mdb vào thư mục được không được quyền truy cập từ Web.

Giả sử ta có website có thư mục webroot là D:\inetpub\wwwroot\. Thư mục chứa tập tin CSDL ví dụ là D:\inetpub\wwwroot\Site1\data\myDB.mdb. Mặc định nếu người dùng đoán được đường dẫn này: <http://www.yourserver.com/site1/data/myDB.mdb>, họ có thể tải được tập tin CSDL này về bởi vì thông thường các tập tin trong thư mục này được thiết lập quyền Read.

Để hạn chế không cho phép người dùng tải tập tin CSDL về, ta sẽ bỏ quyền Read được thiết lập trong thư mục này bằng cách dùng tiện ích Internet Service Manager.



Thao tác này không ảnh hưởng gì đến việc các đoạn mã ASP truy cập đến CSDL do thiết lập này được đặt ở mức webserver chứ không phải ở mức hệ thống tập tin NTFS. Nghĩa là các đoạn mã ASP vẫn hoạt động bình thường như trước. Điểm khác duy nhất là người dùng không thể tải được tập tin CSDL dù biết đường dẫn đến nó mà thôi.

Phương án 2: Đặt tập tin CSDL .mdb tại nơi mà chỉ truy cập được ở mức server-side

Ý tưởng chính của phương án này là đặt tập tin CSDL trong một thư mục có cấp cao hơn thư mục webroot của webserver. Ví dụ, nếu thư mục D:\inetpub\wwwroot\ là webroot của webserver, ta có thể tạo một thư mục private đặt tại D:\inetpub\private và đặt tập tin CSDL vào đây. Bằng cách này,

người dùng client không thể nào truy cập đến thư mục private này để tải CSDL về. Lúc này, đường dẫn đến tập tin CSDL trong chuỗi DSN sẽ được chỉnh lại như sau:

- Nếu dùng đường dẫn tuyệt đối: sFileName = "D:\inetpub\private"
- Nếu dùng đường dẫn tương đối:
 sFileName = Server.MapPath("/") ' trả về giá trị D:\inetpub\wwwroot
 sFileName = Replace(sFileName, "wwwroot", "private")
 sFileName = sFileName & "myDB.mdb"

Lê Đình Duy – ldduy@fit.hcmuns.edu.vn

BẢO VỆ ỨNG DỤNG WEB CHỐNG TẤN CÔNG KIỂU SQL INJECTION

Lê Đình Duy

Khoa CNTT – ĐHKHTN Tp.HCM

ldduy@fit.hcmuns.edu.vn

11.2002

1. SQL Injection là gì?

Việc thiết kế và đưa vào hoạt động một website luôn đòi hỏi các nhà phát triển phải quan tâm đến các vấn đề về an toàn, bảo mật nhằm giảm thiểu tối đa khả năng bị tấn công từ các tin tặc. Tuy nhiên, thông thường các nhà phát triển đã dồn trọng tâm vào các vấn đề an toàn trong việc chọn hệ điều hành, hệ quản trị CSDL, webserver sẽ chạy ứng dụng, ... Ví dụ, người ta thường quan tâm nhiều đến các lỗ hổng về an toàn trên IIS hơn là quan tâm đến các đoạn mã của ứng dụng có tiềm ẩn các lỗ hổng nghiêm trọng hay không. Một trong số các lỗ hổng này đó là SQL injection attack.

SQL injection là một kĩ thuật cho phép những kẻ tấn công thi hành các câu lệnh truy vấn SQL bất hợp pháp (không được người phát triển lường trước) bằng cách lợi dụng lỗ hổng trong việc kiểm tra dữ liệu nhập trong các ứng dụng web. Hậu quả của nó rất tai hại vì nó cho phép những kẻ tấn công có thể thực hiện các thao tác xóa, hiệu chỉnh, ... do có toàn quyền trên cơ sở dữ liệu của ứng dụng. Lỗi này thường xảy ra trên các ứng dụng web có dữ liệu được quản lí bằng các hệ quản trị CSDL như SQL Server, Oracle, DB2, Sysbase.

Xét một ví dụ điển hình, thông thường để cho phép người dùng truy cập vào các trang web được bảo mật, hệ thống thường xây dựng trang đăng nhập để yêu cầu người dùng nhập thông tin về tên đăng nhập và mật khẩu. Sau khi người dùng nhập thông tin vào, hệ thống sẽ kiểm tra tên đăng nhập và mật khẩu có hợp lệ hay không để quyết định cho phép hay từ chối thực hiện tiếp.

Trong trường hợp này, người ta có thể dùng 2 trang, một trang HTML để hiển thị form nhập liệu và một trang ASP dùng để xử lí thông tin nhập từ phía người dùng. Ví dụ:

Login.htm

```
<form action="ExecLogin.asp" method="post">
  Username: <input type="text" name="txtUsername"><br>
  Password: <input type="password" name="txtPassword"><br>
  <input type="submit">
</form>
```

ExecLogin.asp

```
<%
Dim p_strUsername, p_strPassword, objRS, strSQL

p_strUsername = Request.Form("txtUsername")
p_strPassword = Request.Form("txtPassword")

strSQL = "SELECT * FROM tblUsers " & _
        "WHERE Username=" & p_strUsername & _
        " and Password=" & p_strPassword & ""

Set objRS = Server.CreateObject("ADODB.Recordset")
objRS.Open strSQL, "DSN=..."

If (objRS.EOF) Then
  Response.Write "Invalid login."
Else
  Response.Write "You are logged in as " & objRS("Username")
```

```
End If
Set objRS = Nothing
%>
```

Thoạt nhìn, đoạn mã trong trang ExecLogin.asp dường như không chứa bất cứ một lỗ hổng về an toàn nào. Người dùng không thể đăng nhập mà không có tên đăng nhập và mật khẩu hợp lệ. Tuy nhiên, đoạn mã này thực sự không an toàn và là tiền đề cho một SQL injection attack. Đặc biệt, chỗ sơ hở nằm ở chỗ dữ liệu nhập vào từ người dùng được dùng để xây dựng trực tiếp câu lệnh truy vấn SQL. Chính điều này cho phép những kẻ tấn công có thể điều khiển câu truy vấn sẽ được thực hiện.

Ví dụ, nếu người dùng nhập chuỗi sau vào trong cả 2 ô nhập liệu username/password của trang Login.htm: **' or ' = ' .** Lúc này, câu truy vấn sẽ được gọi thực hiện là:

```
SELECT * FROM tblUsers WHERE Username=" or "=" and Password = " or "="
```

Câu truy vấn này là hợp lệ và sẽ trả về tất cả các bản ghi của tblUsers và đoạn mã tiếp theo xử lí người dùng đăng nhập bất hợp pháp này như là người dùng đăng nhập hợp lệ.

Một ví dụ khác của SQL injection attack nữa là khi các trang web sử dụng dữ liệu nhập vào theo dạng querystring (bằng cách gõ cặp tham số và giá trị trực tiếp trên thanh địa chỉ hoặc dùng form với thuộc tính ACTION là GET). Ví dụ sau minh họa một trang ASP nhận dữ liệu cho biến ID thông qua querystring và phát sinh nội dung của trang đó dựa trên ID:

```
<%
Dim p_IngID, objRS, strSQL
p_IngID = Request("ID")

strSQL = "SELECT * FROM tblArticles WHERE ID=" & p_IngID

Set objRS = Server.CreateObject("ADODB.Recordset")
objRS.Open strSQL, "DSN=..."

If (Not objRS.EOF) Then Response.Write objRS("ArticleContent")

Set objRS = Nothing
%>
```

Trong các tình huống thông thường, đoạn mã này hiển thị nội dung của article có ID trùng với ID được chuyển đến cho nó dưới dạng querystring. Ví dụ, trang này có thể được gọi như sau:

<http://www.example.com/Article.asp?ID=1055>, để hiển thị nội dung của article có ID là 1055.

Giống như ví dụ đăng nhập ở trước, đoạn mã này để lộ sơ hở cho một SQL injection attack. Kẻ tấn công có thể thay thế một ID hợp lệ bằng cách gán ID cho một giá trị khác, để thực hiện một lệnh SQL bất hợp pháp, ví dụ như: **0 or 1=1** (nghĩa là, <http://www.example.com/Article.asp?ID=0 or 1=1>).

Câu truy vấn SQL lúc này sẽ trả về tất cả các article từ bảng dữ liệu vì nó sẽ thực hiện câu lệnh:

```
SELECT * FROM tblArticles WHERE ID=0 or 1=1
```

Tất nhiên ví dụ này dường như không có gì nguy hiểm, nhưng hãy thử tưởng tượng kẻ tấn công có thể xóa toàn bộ CSDL bằng cách chèn vào các đoạn lệnh nguy hiểm như lệnh DELETE. Tất cả chỉ là đơn giản thay đổi chuỗi gán dữ liệu cho ID, ví dụ như:

<http://www.example.com/Article.asp?ID=1055>; DELETE FROM tblArticles.

2. Các tác hại và cách phòng tránh

Tác hại từ SQL Injection attack tùy thuộc vào môi trường và cách cấu hình hệ thống. Nếu ứng dụng sử dụng quyền dbo (quyền của người sở hữu CSDL - owner) khi thao tác dữ liệu, nó có thể xóa toàn bộ các bảng dữ liệu, tạo các bảng dữ liệu mới, ... Nếu ứng dụng sử dụng quyền sa (quyền quản trị hệ thống), nó có thể điều khiển toàn bộ hệ quản trị CSDL và với quyền hạn rộng lớn như vậy nó có thể tạo ra các tài khoản người dùng bất hợp pháp để điều khiển hệ thống của bạn.

Để phòng tránh các nguy cơ có thể xảy ra, hãy bảo vệ các câu truy vấn SQL là bằng cách kiểm soát chặt chẽ tất cả các dữ liệu nhập nhận được từ đối tượng Request (Request, Request.QueryString, Request.Form, Request.Cookies, and Request.ServerVariables).

- Trong trường hợp dữ liệu nhập vào là chuỗi, như trong ví dụ 1, lỗi xuất phát từ việc có dấu nháy đơn trong dữ liệu. Để tránh điều này, thay thế các dấu nháy đơn bằng hàm Replace để thay thế bằng 2 dấu nháy đơn:

```
p_strUsername = Replace(Request.Form("txtUsername"), "'", "'")  
p_strPassword = Replace(Request.Form("txtPassword"), "'", "'")
```

- Trong trường hợp dữ liệu nhập vào là số, như trong ví dụ 2, lỗi xuất phát từ việc thay thế một giá trị được tiên đoán là dữ liệu số bằng chuỗi chứa câu lệnh SQL bất hợp pháp. Để tránh điều này, đơn giản hãy kiểm tra dữ liệu có đúng kiểu hay không:

```
p_lngID = CLng(Request("ID"))
```

Như vậy, nếu người dùng truyền vào một chuỗi, hàm này sẽ trả về lỗi ngay lập tức.

Ngoài ra để tránh các nguy cơ từ SQL Injection attack, nên chú ý loại bỏ bất kì thông tin kĩ thuật nào chứa trong thông điệp chuyển xuống cho người dùng khi ứng dụng có lỗi. Các thông báo lỗi thông thường tiết lộ các chi tiết kĩ thuật có thể cho phép kẻ tấn công biết được điểm yếu của hệ thống.

Cuối cùng, để giới hạn mức độ của SQL Injection attack, nên kiểm soát chặt chẽ và giới hạn quyền xử lí dữ liệu đến tài khoản người dùng mà ứng dụng web đang sử dụng. Các ứng dụng thông thường nên tránh dùng đến các quyền như dbo hay sa. Quyền càng bị hạn chế, thiệt hại càng ít.

Các tài liệu tham khảo

SQL Injection FAQ: <http://www.sqlsecurity.com/DesktopDefault.aspx?tabindex=2&tabid=3>

Advanced SQL Injection : http://www.nextgenss.com/papers/advanced_sql_injection.pdf

Preventing SQL Injection: http://www.owasp.org/asac/input_validation/sql.shtml

Biên dịch từ: <http://www.4guysfromrolla.com/webtech/061902-1.shtml>