

Buổi 9**Request, Response, Session, Cookie**

Trong khuôn khổ tài liệu buổi 9, chúng ta sử dụng CSDL WebBanHang:

- File [WebBanHang] Mo ta CSDL.png: Lược đồ mô tả cấu trúc bảng, khóa ngoại.
- File [WebBanHang] CSDL.sql: Phát sinh CSDL.

I. Request

Request là 1 đối tượng thuộc class `System.Web.HttpRequest` chứa thông tin từ trình duyệt (*browser*) gửi lên máy chủ. Thông tin này có thể được gửi thông qua dạng GET hoặc POST.

Lưu ý: Mặc định ASP.NET gửi thông qua dạng POST.

1. Gửi thông tin dạng GET

Thông tin được gửi dưới dạng GET sẽ được truyền thẳng vào đường dẫn (*URL*) của HTTP Request theo từng cặp `name=value`, phân cách nhau bởi dấu `&`. Do đó, thông tin này có thể được lưu vào bộ nhớ đệm (*cache*) của trình duyệt, lưu lại dưới dạng bookmark, history, bị giới hạn về độ dài và không có tính bảo mật:

`http://www.google.com/search?q=asp&lang=vi&filetype=pdf`

Đối tượng Request cung cấp thuộc tính `QueryString` là 1 mảng chứa tất cả những cặp `name-value` của chuỗi GET. Để lấy giá trị từ chuỗi GET, chúng ta sử dụng cú pháp sau:

`Request.QueryString["<keyName>"]`

Ví dụ: Để lấy giá trị *pdf* từ đường dẫn ở trên, chúng ta dùng câu lệnh sau:

`string fileType = Request.QueryString["filetype"];`

2. Gửi thông tin dạng POST

Thông tin được gửi dưới dạng POST sẽ được truyền vào phần nội dung của HTTP Request. Do đó, thông tin này không thể được lưu vào bộ nhớ đệm (*cache*) của trình duyệt, không thể lưu lại dưới dạng bookmark, history, không bị giới hạn về độ dài và có tính bảo mật cao.

Với thông tin được gửi dưới dạng POST, ASP.NET WebForm đã tự động chuyển thành các thuộc tính của các control, do đó chúng ta sẽ truy cập thông tin qua các thuộc tính này. Ngoài ra, có thể truy cập trực tiếp bằng đối tượng Request.

Ví dụ: Có 1 textbox ASP.NET và 1 textbox HTML như sau:

`<asp:TextBox ID="txt1" runat="server"></asp:TextBox>`

`<input type="text" name="txt2" runat="server" />`

- Để lấy giá trị textbox 1, chúng ta có thể dùng cú pháp `txt1.Text` hoặc `Request["txt1"]`.
- Để lấy giá trị textbox 2, chúng ta có thể dùng cú pháp `Request["txt2"]`.

3. Các thuộc tính, phương thức khác

Đối tượng Request cung cấp một số thuộc tính và phương thức thường dùng như sau:

- Thuộc tính `Cookies`¹ là 1 Collection chứa tất cả các cookie của trang hiện tại.
- Thuộc tính `RawUrl` cho biết đường dẫn của trang web đang gửi request.
- Phương thức `MapPath()` cho biết đường dẫn tuyệt đối của 1 đường dẫn ảo nào đó.

¹ Sẽ được giải thích kĩ hơn ở phần III

II. Response

Response là 1 đối tượng thuộc class `System.Web.HttpResponse` chứa thông tin từ máy chủ gửi trả về cho trình duyệt sau khi đã xử lý yêu cầu.

Đối tượng Response cung cấp một số thuộc tính và phương thức thường dùng như sau:

- Thuộc tính `Cookies2` là 1 `Collection` chứa tất cả các cookie của trang hiện tại.
- Phương thức `Redirect()` dùng để chuyển trang đến 1 URL.
- Phương thức `Write()` dùng để ghi nội dung vào trang web ở client.

III. Cookie

Trong các ứng dụng web, việc lưu lại thông tin của người dùng xuyên suốt các trang web trong cùng 1 ứng dụng web là vấn đề rất quan trọng. Nếu như thông tin không được lưu xuyên suốt như vậy, thì dữ liệu của các trang web sẽ là rời rạc với nhau, ví dụ: người dùng đã đăng nhập vào ứng dụng từ trang `DangNhap.aspx`, nhưng khi qua các trang khác như `index.aspx`, `DSSanPham.aspx`... thì trạng thái đăng nhập sẽ bị mất.

Để lưu lại thông tin xuyên suốt các trang web trong cùng 1 ứng dụng web, chúng ta có thể sử dụng *Session* và *Cookie*.

Cookie là thông tin được lưu ở máy client dưới dạng văn bản theo từng cặp `key-value`, có thời gian hiệu lực (*expiration time*) nhất định. Sau khi hết thời gian này, cookie sẽ được trình duyệt tự động xóa.

Mỗi khi trình duyệt và máy chủ tương tác với nhau thông qua request và response, những cookie có liên quan tới trang web hiện tại sẽ được gửi kèm với request và response này. Đó là lý do vì sao đối tượng `Request` và `Response` ở phần I và II đều có thuộc tính `Cookies` chứa tất cả cookie của trang hiện tại.

Một số thông tin có thể lưu lại bằng cookie:

- Thông tin tài khoản (ngoại trừ mật khẩu và các thông tin cần bảo mật).
- Trạng thái đăng nhập.
- Các tùy chỉnh, thiết lập của người dùng với ứng dụng.
- Các sản phẩm người dùng thêm vào giỏ hàng mà chưa thanh toán.
- ...

Do cookie được lưu ở client và không có tính bảo mật, người dùng có thể dễ dàng xem và thay đổi thông tin của cookie, do đó, không được phép lưu những thông tin cần bảo mật như mật khẩu, số thẻ ngân hàng... bằng cookie.

1. Tạo cookie

Để tạo 1 cookie có tên là `"TenTK"` và giá trị là `"admin"`, thời gian hiệu lực là 2 tuần, chúng ta sử dụng đoạn code sau:

```
HttpCookie c = new HttpCookie("TenTK");  
c.Value = "admin";  
c.Expires = DateTime.Now.AddDays(14);  
Response.Cookies.Add(c);
```

² Sẽ được giải thích kĩ hơn ở phần III

2. Lấy giá trị cookie

Để lấy giá trị của cookie có tên là "TenTK", chúng ta sử dụng đoạn code sau:

```
HttpCookie c = Request.Cookies["TenTK"];
if (c != null)
{
    string user = c.Value;
}
```

3. Thay đổi giá trị cookie

Cookie có thể được ghi đè nếu trùng tên. Do đó, để thay đổi giá trị của cookie có tên là "TenTK", chúng ta sử dụng đoạn code sau:

```
HttpCookie c = Request.Cookies["TenTK"];
if (c != null)
{
    c.Value = "guest";
    Response.Cookies.Add(c);
}
```

4. Xóa cookie

Cookie sẽ tự động bị xóa khi hết thời gian hiệu lực. Do đó, để thay đổi giá trị của cookie có tên là "TenTK", chúng ta sử dụng đoạn code sau:

```
HttpCookie c = Request.Cookies["TenTK"];
if (c != null)
{
    c.Expires = DateTime.Now.AddDays(-1);
    Response.Cookies.Add(c);
}
```

Lưu ý: Để lấy giá trị cookie từ client gửi lên, chúng ta dùng đối tượng Request. Để tạo, thay đổi, xóa cookie và lưu lại kết quả xuống client, chúng ta dùng đối tượng Response.

IV. Session

Session là 1 phiên làm việc của người dùng, bắt đầu từ khi người dùng gửi request đầu tiên cho đến khi đóng trình duyệt.

Mỗi Session được máy chủ cấp cho 1 SessionID lưu trên server và 1 cookie có lưu SessionID này tại client để quản lý session nào tương ứng với máy nào. Toàn bộ thông tin của session sẽ được lưu ở server.

Các thao tác với session đều được thực hiện thông qua đối tượng Session. Đối tượng này là 1 Collection chứa các thông tin theo từng cặp key-value:

```
// Ghi giá trị vào session
Session["TenTK"] = "admin";
// Thay đổi giá trị session
Session["TenTK"] = "guest";
// Lấy giá trị session
string user = Session["TenTK"];
// Xóa session
Session.Clear();
// Hủy bỏ session và tạo session mới
Session.Abandon();
```

Điểm khác nhau giữa phương thức `Clear()` và `Abandon()` của đối tượng `Session`:

- Phương thức `Clear()` chỉ xóa các cặp key-value của session nhưng vẫn tiếp tục sử dụng session đó, tức là `SessionID` không đổi.
- Phương thức `Abandon()` hủy bỏ session và tạo session mới. Do đó, `SessionID` cũ sẽ bị xóa và server sẽ cấp `SessionID` mới. Ngoài ra, phương thức này còn kích hoạt sự kiện `Session_End()`.

V. So sánh Cookie và Session

Cookie	Session
Có tính toàn cục. Dùng để lưu dữ liệu của người dùng xuyên suốt tất cả các trang web trong cùng 1 ứng dụng web.	
Lưu ở client	Lưu ở server
Tính bảo mật thấp	Tính bảo mật cao

VI. Bài tập

Thực hiện các yêu cầu dưới đây:

1. Tạo `MasterPage` gồm 1 dòng tiêu đề "*Web bán hàng*", 1 dòng lời chào "*Xin chào <họ tên>*" và nút *Đăng xuất*, trong đó:
 - `<họ tên>` là họ tên của người dùng đang đăng nhập. Nếu người dùng chưa đăng nhập thì hiển thị "*Xin chào Khách*".
 - Nút *Đăng xuất* chỉ hiển thị nếu người dùng đã đăng nhập.
 - Khi nhấn nút *Đăng xuất*, hủy bỏ trạng thái đăng nhập của người dùng đồng thời refresh lại trang web.
 - Tất cả các trang con đều sử dụng `MasterPage` này.
2. Tạo trang `DangNhap.aspx` cho phép người dùng đăng nhập tài khoản. Kiểm tra thông tin tài khoản từ `CSDL`, nếu đúng thì lưu lại trạng thái đăng nhập (bằng cookie hoặc session) và chuyển sang trang `DSSanPham.aspx`, nếu sai thì thông báo "*Đăng nhập thất bại*".
3. Tạo trang `DSSanPham.aspx` hiển thị danh sách tất cả sản phẩm trong cửa hàng (bằng `DataList` hoặc `Repeater`). Mỗi sản phẩm gồm ảnh minh họa, tên sản phẩm, giá tiền và nút *Thêm vào GH*, trong đó:
 - Khi click vào ảnh minh họa hoặc tên sản phẩm, chuyển sang trang `ChiTietSP.aspx`, hiển thị thông tin chi tiết của sản phẩm vừa chọn.
 - Khi nhấn nút *Thêm vào GH*, thêm thông tin sản phẩm đó vào giỏ hàng. Lưu ý kiểm tra sản phẩm có tồn tại trong giỏ hàng hay chưa.
4. Tạo trang `ChiTietSP.aspx` hiển thị tất cả thông tin của 1 sản phẩm bất kỳ (gợi ý: sử dụng chuỗi `GET`), đồng thời kèm theo 1 textbox để nhập số lượng và nút *Thêm vào GH*, trong đó:
 - Chỉ cho phép nhập số vào textbox số lượng.
 - Xử lý nút *Thêm vào GH* tương tự như trang `DSSanPham.aspx`.