



TRUNG TÂM TIN HỌC  
Trường Đại học Khoa Học Tự Nhiên - Đại học quốc gia TP.HCM

# LẬP TRÌNH VIÊN CÔNG NGHỆ .NET

## Module 3

---

*Click vào phụ lục để chuyển tới bài cần đọc*

### Phụ lục

Bài 1 Tổng quan về Lập trình Web và ASP.NET .....	2
Bài 2 Tổ chức ASP.Net Website và xây dựng Layout .....	20
Bài 3 Thiết kế và xử lý Web Forms .....	35
Bài 4 Tạo User Controls và Kiểm tra dữ liệu trên Web Forms.	47
Bài 6 Xây dựng trang web có liên kết với CSDL .....	82
Bài 7 Quản lý Web Application .....	95
Bài 8 Hoàn chỉnh ứng dụng WEB .....	104
Bài 9 Web Services .....	109





Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

## LTV CÔNG NGHỆ .NET

### Module 3 - Bài 1. *Tổng quan về Lập trình Web và ASP.NET*

Ngành Lập trình & CSDL

<http://www.t3h.vn>

2014



2014





## Nội dung

1. **Giới thiệu về Internet và một số khái niệm**
  1. Giới thiệu về Internet
  2. Web server, Web Client
  3. Web Browser, giới thiệu một số trình duyệt web
  4. HTTP, URL, HTML
2. **Tổng quan về lập trình web: web tĩnh và web động**
3. **Giới thiệu về ASP.Net**
4. **Tổng quan về lập trình ứng dụng web**
5. **Tạo ứng dụng web đầu tiên**



## 1. Giới thiệu về Internet và một số khái niệm

- **Giới thiệu về Internet**
  - Là tập hợp các máy tính nối mạng trên thế giới trao đổi thông tin với nhau thông qua giao thức TCP/IP (Transmission Control Protocol/Internet Protocol)
  - Mang lại rất nhiều tiện ích hữu dụng cho người sử dụng, nổi bật nhất là dịch vụ thư điện tử (email), trò chuyện trực tuyến (chat) và truy tìm dữ liệu (search engine) , ...
  - Dịch vụ được dùng rất phổ biến trên Internet là World Wide Web (WWW, thường được gọi tắt là Web)





## 1. Giới thiệu về Internet và một số khái niệm

### ❑ Web server

- Máy tính lưu trữ website
- Xử lý và trả kết quả về cho Web Client khi nhận được yêu cầu

### ❑ Web Client

- Máy tính dùng để truy cập các trang web
- Có khả năng yêu cầu và nhận kết quả từ Web Server



## 1. Giới thiệu về Internet và một số khái niệm

### ❑ Web Browser – Trình duyệt

- Phần mềm dùng để xem các tài liệu hoặc tìm kiếm các tài nguyên trên World Wide Web
- Một số trình duyệt thông dụng hiện nay:
  - Internet Explorer (IE) có sẵn trong Microsoft Windows của Microsoft
  - Mozilla Firefox của Tập đoàn Mozilla
  - Chrome của Google
  - Safari của hãng Apple





# 1. Giới thiệu về Internet và một số khái niệm

## □ HTTP – HyperText Transfer Protocol

- Là giao thức chuyển giao siêu văn bản trên web
- Giao thức này là tập hợp các qui định dùng để trao đổi các tài liệu (văn bản, hình ảnh, âm thanh, video, các tập tin đa truyền thông,...) giữa Web server và Web browser

## □ URL – Uniform Resource Locator

- Đường dẫn chỉ tới một trang web cụ thể trên Internet.
- Cú pháp đầy đủ: **scheme://<host> [:port] [<path> [?<querystring>] ]**

Trong đó:

- **scheme**: loại dịch vụ Internet
- **host**: địa chỉ máy chủ chứa tài nguyên
- **port**: cổng dịch vụ trên máy chủ
- **path**: đường dẫn và tên của tập tin tài nguyên trên máy chủ
- **querystring**: các tham số được gửi kèm theo http



LTV Công Nghệ .Net (2014) – Module 3 - Bài 1. Tổng quan về lập trình web và ASP.Net

6



# 1. Giới thiệu về Internet và một số khái niệm

## □ URL – Uniform Resource Locator

Ví dụ:

<http://www.legend.net.uk/resources/gloss.html>

<http://t3h.vn/Pages/MBookView.aspx?ID=2>

[http://www.htmlib.com:80/example\\_files/file.html?Lib=3](http://www.htmlib.com:80/example_files/file.html?Lib=3)

## □ HTML – HyperText Markup Language

- Ngôn ngữ dùng để xây dựng các trang web
- Gồm các tag giúp web browser biết cách định dạng thông tin hiển thị

Ví dụ: nội dung trang web Chao.html

```
<html>
  <head><title> ASP.NET</title></head>
  <body>
    <h1>Chào mừng các bạn đến với ASP.Net </h1>
  </body>
</html>
```



LTV Công Nghệ .Net (2014) – Module 3 - Bài 1. Tổng quan về lập trình web và ASP.Net

7



## Nội dung

1. Giới thiệu về Internet và một số khái niệm
2. Tổng quan về lập trình web: web tĩnh và web động
3. Giới thiệu về ASP.Net
4. Tổng quan về lập trình ứng dụng web
5. Tạo ứng dụng web đầu tiên

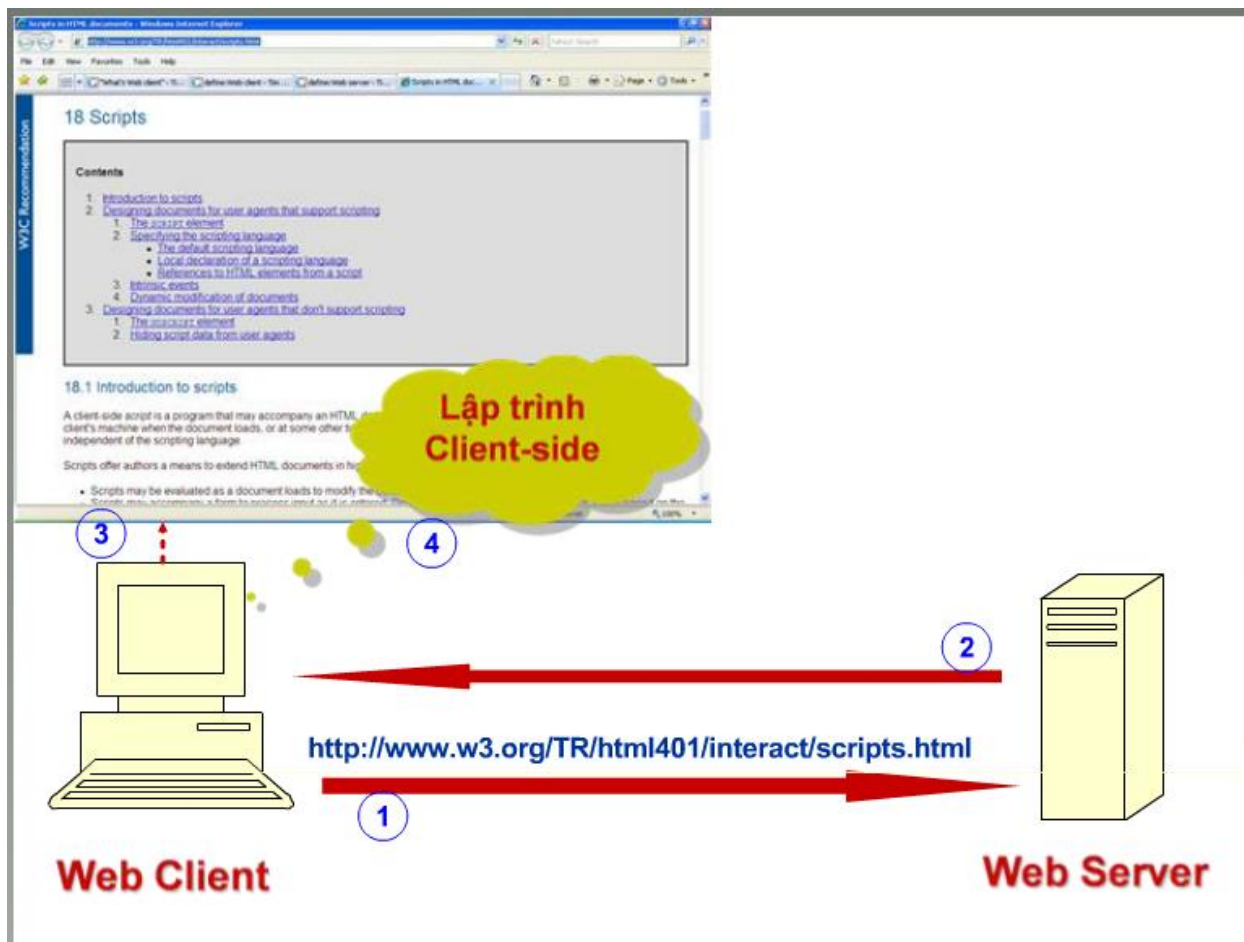


## 2. Tổng quan về lập trình Web

### □ Web tĩnh

- Là trang web được trình bày dưới dạng văn bản. Nội dung trong trang được tạo ra lúc thiết kế và không được thay đổi khi có người dùng truy cập
- Được viết bằng ngôn ngữ HTML và không có kết nối với cơ sở dữ liệu (CSDL)
- Khi có nhu cầu xử lý trên trang web tĩnh, ta dùng ngôn ngữ Client-side script → Lập trình Client-side
- Quá trình xử lý chỉ xảy ra tại Web client





## 2. Tổng quan về lập trình Web

### □ Web tĩnh

- Đặc điểm:
  - Nhanh, không mất thời gian để thể hiện kết quả lên trang web
  - Chỉ có thể thực hiện những xử lý đơn giản và độc lập
  - Không cần phải cài đặt phần mềm tại web server
  - Web ngày càng phát triển thì trong một website:
    - Có thể có các trang web tĩnh
    - Có các xử lý ở phía Client trong các trang web động

## 2. Tổng quan về lập trình Web

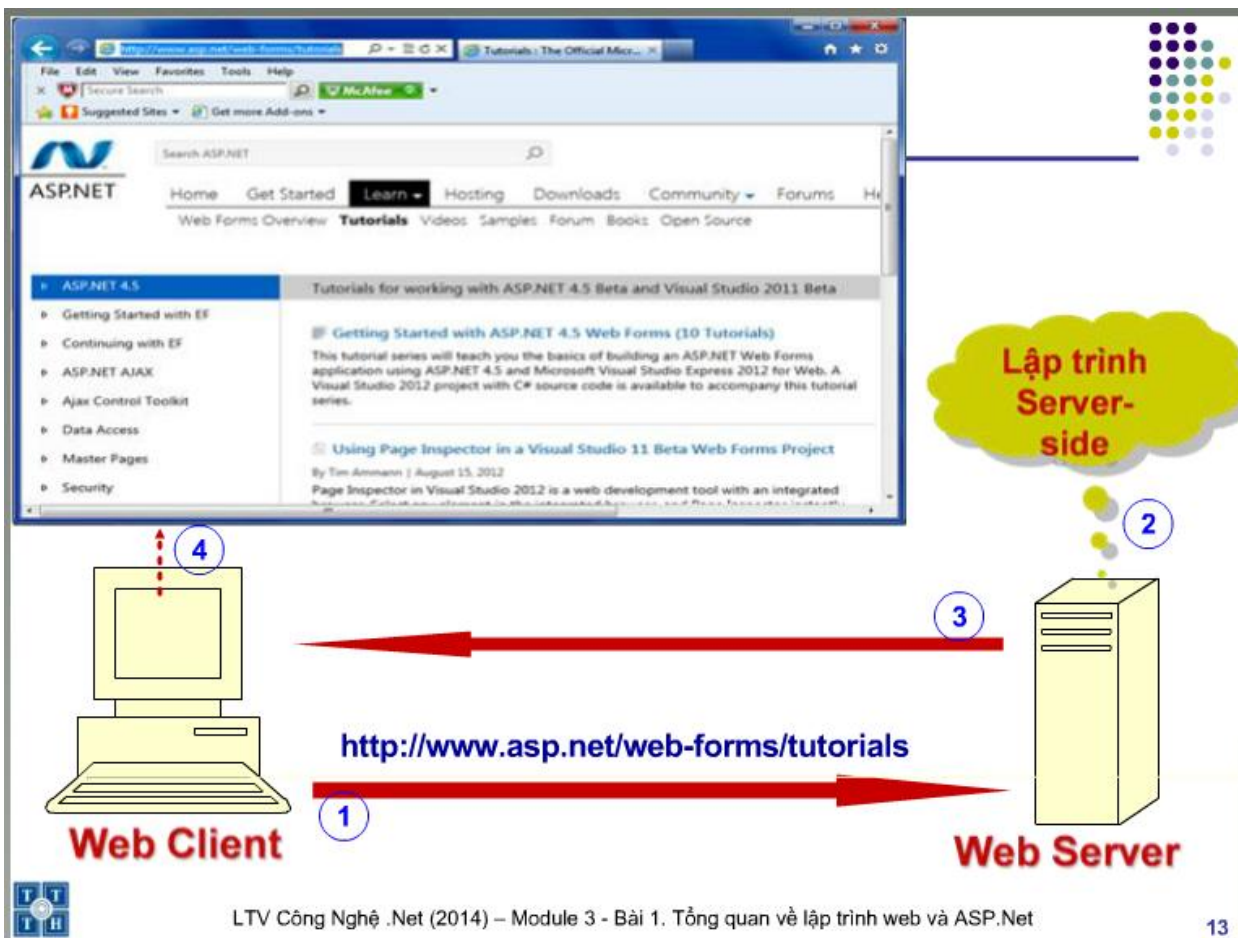
### ❑ Web động

- Là trang web có nội dung được lấy từ CSDL
- Dùng ngôn ngữ Server-side script để xử lý
- Quá trình xử lý xảy ra tại Web server
  - ➔ Lập trình Server-side
- Đặc điểm:
  - Chậm, mất thời gian để thể hiện kết quả lên trang web
  - Có khả năng thực hiện những xử lý phức tạp và truy cập cơ sở dữ liệu
  - Trang web linh động, phục vụ đa yêu cầu của người dùng
  - Phải cài đặt phần mềm tại Web server



LTV Công Nghệ .Net (2014) – Module 3 - Bài 1. Tổng quan về lập trình web và ASP.Net

12



LTV Công Nghệ .Net (2014) – Module 3 - Bài 1. Tổng quan về lập trình web và ASP.Net

13



## Nội dung

1. Giới thiệu về Internet và một số khái niệm
2. Tổng quan về lập trình web: web tĩnh và web động
3. Giới thiệu về ASP.Net
  1. ASP.Net là gì?
  2. Một số đặc điểm của ASP.Net
  3. Giới thiệu một website được tạo bằng ASP.Net
4. Tổng quan về lập trình ứng dụng web
5. Tạo ứng dụng web đầu tiên



## 3. Giới thiệu về ASP.Net

### □ ASP.Net là gì?

- Asp.Net là Active Server Pages .NET (.NET framework)
- ASP.NET là một công nghệ dùng để phát triển các ứng dụng web có nội dung động và các dịch vụ web
- ASP.Net là một kỹ thuật lập trình và phát triển ứng dụng web ở phía Server (server-side) dựa trên môi trường .Net
  - Với kỹ thuật server-side, các đoạn mã lệnh sẽ được xử lý tại server và phát sinh ra HTML, JavaScript, CSS sau đó gửi về cho trình duyệt ở phía client.





### 3. Giới thiệu về ASP.Net

#### □ Một số đặc điểm của ASP.Net

- Easy Programming Model
  - ASP.NET hỗ trợ phát triển và triển khai các ứng dụng web trên mạng toàn cầu trong một thời gian nhanh kỷ lục
    - Vì nó cung cấp cho ta một kiểu mẫu lập trình dễ dàng và gọn gàng nhất.
  - ASP.NET tự động phát sinh mã HTML cho các Server control tương ứng sao cho tối ưu với từng loại browsers hiện nay như:
    - Internet Explorer (IE), Netscape, Opera, AOL, Mozilla FireFox, Google Chrome...
- Flexible Language Options:
  - ASP.NET có khả năng giao tiếp trên 40 ngôn ngữ lập trình như:
    - C#, VB.NET, JScript.NET, MC++.NET, Smalltalk.NET, COBOL.NET, Component Pascal.NET, Eiffel.NET, Perl.NET, Mercury.NET, sdsOberon.NET, Python.NET...



LTV Công Nghệ .Net (2014) – Module 3 - Bài 1. Tổng quan về lập trình web và ASP.Net

16

### 3. Giới thiệu về ASP.Net

#### □ Một số đặc điểm của ASP.Net

- Rich Class Framework:
  - Dựa trên nền tảng vững vàng và tài nguyên phong phú .NET Framework với hơn 5000 classes bao gồm đủ thứ như: XML, data access, file upload, regular expressions, transactions, message queuing, SMTP mail, ... nên việc thiết kế các đặc tính trong một ứng dụng trở nên dễ dàng và tiện lợi hơn
- Great Tool Support:
  - Hỗ trợ thiết kế nhanh WebForm hay Services theo phương pháp “drag-drop-doubleclick” quen thuộc của nền Windows.
  - Hỗ trợ việc phát hiện và loại bỏ các lỗi sai một cách rất thuận lợi trong quá trình phát triển các ứng dụng web (support for debugging and deploying ASP.NET Web applications)



LTV Công Nghệ .Net (2014) – Module 3 - Bài 1. Tổng quan về lập trình web và ASP.Net

17



### 3. Giới thiệu về ASP.Net

#### ❑ Một số đặc điểm của ASP.Net

- Mô hình code-behind
  - Cho phép phân chia các đoạn mã server-side và HTML. Đưa ra cách viết mã linh động bằng cách để những mã lập trình *server-side code* trong một tập tin riêng biệt.
- ASP.NET được Compiled chứ không phải là Interpreted
  - Trang Asp.Net được biên dịch trước, thành tập tin DLL dưới dạng mã nhị phân → Server có thể thi hành một cách nhanh chóng.
  - ASP.NET được biên dịch dưới dạng Common Language Runtime (CLR), cho phép những người lập trình viết mã ASP.NET với bất kỳ ngôn ngữ nào được hỗ trợ bởi .NET language



### 3. Giới thiệu về ASP.Net

#### ❑ Giới thiệu một website được tạo bằng ASP.Net

- GV có thể giới thiệu bất cứ website nào được xây dựng bằng ASP.Net



## Nội dung

1. Giới thiệu về Internet và một số khái niệm
2. Tổng quan về lập trình web: client – side và server side
3. Giới thiệu về ASP.Net
4. Tổng quan về lập trình ứng dụng web
  1. Hệ thống namespace – các mô hình tổ chức ứng dụng
  2. Quá trình xử lý tập tin ASP.Net
5. Tạo ứng dụng web đầu tiên

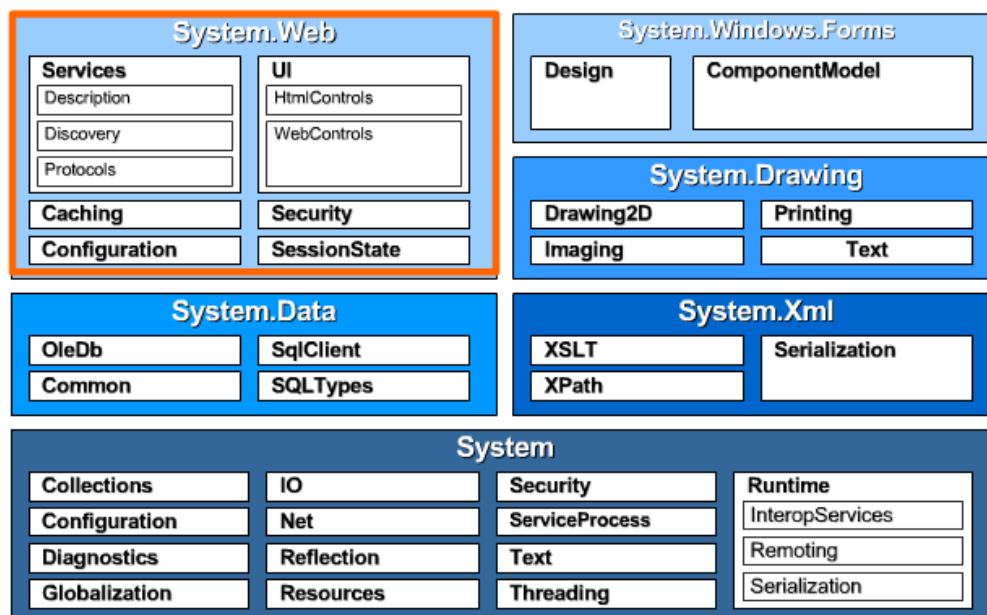


LTV Công Nghệ .Net (2014) – Module 3 - Bài 1. Tổng quan về lập trình web và ASP.Net

20

## 4. Tổng quan về lập trình ứng dụng web

### □ Hệ thống namespace trong .Net Framework



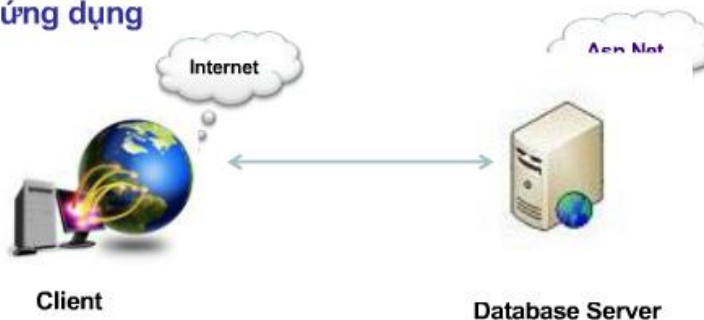
LTV Công Nghệ .Net (2014) – Module 3 - Bài 1. Tổng quan về lập trình web và ASP.Net

21

## 4. Tổng quan về lập trình ứng dụng web

### □ Các mô hình tổ chức ứng dụng

#### • Mô hình 1:



#### • Mô hình 2:



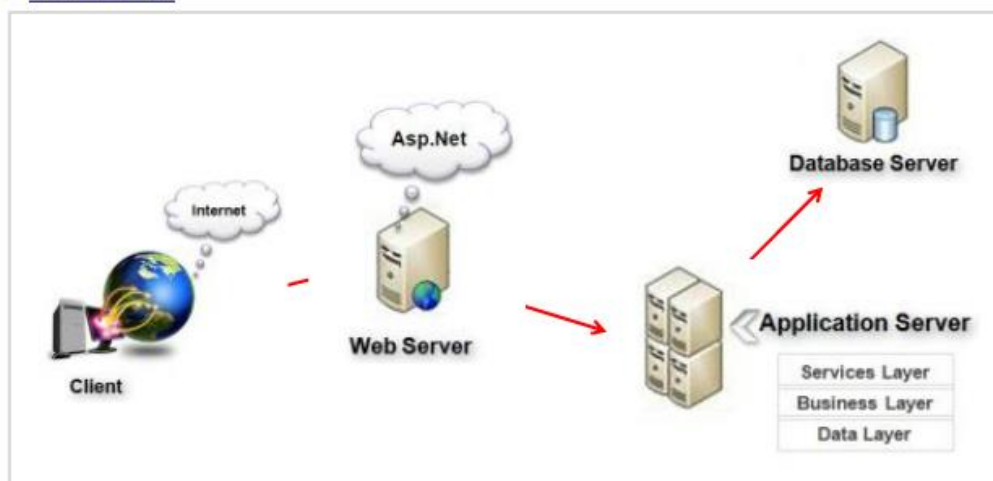
LTV Công Nghệ .Net (2014) – Module 3 - Bài 1. Tổng quan về lập trình web và ASP.Net

22

## 4. Tổng quan về lập trình ứng dụng web

### □ Các mô hình tổ chức ứng dụng

#### • Mô hình 3:



LTV Công Nghệ .Net (2014) – Module 3 - Bài 1. Tổng quan về lập trình web và ASP.Net

23

## 4. Tổng quan về lập trình ứng dụng web

- Các mô hình tổ chức ứng dụng
  - Quá trình Request - Response

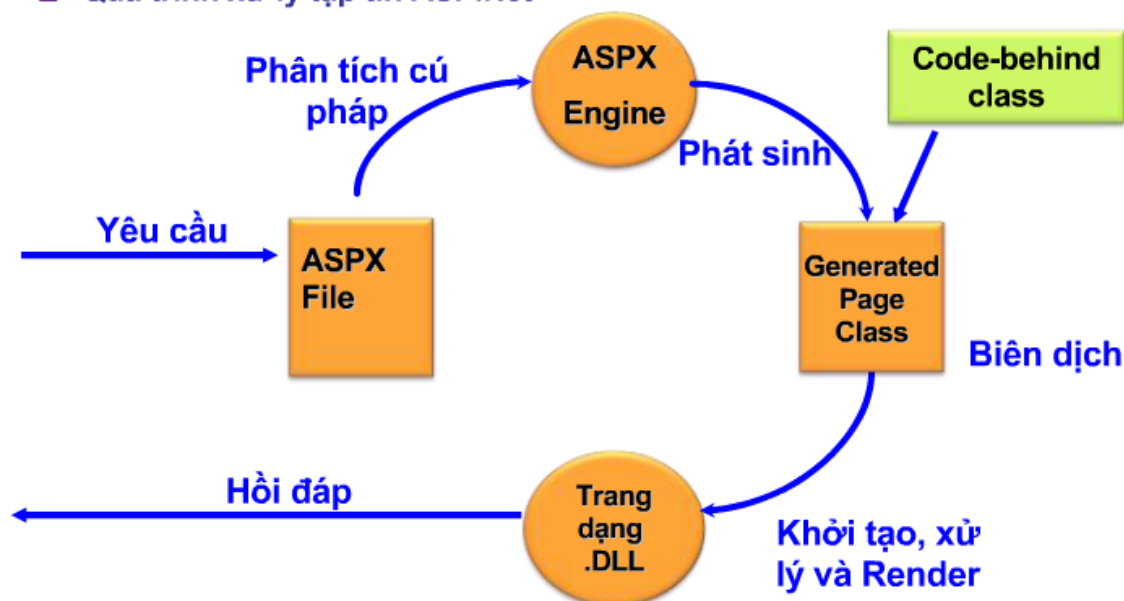


LTV Công Nghệ .Net (2014) – Module 3 - Bài 1. Tổng quan về lập trình web và ASP.Net

24

## 4. Tổng quan về lập trình ứng dụng web

- Quá trình xử lý tập tin ASP.Net

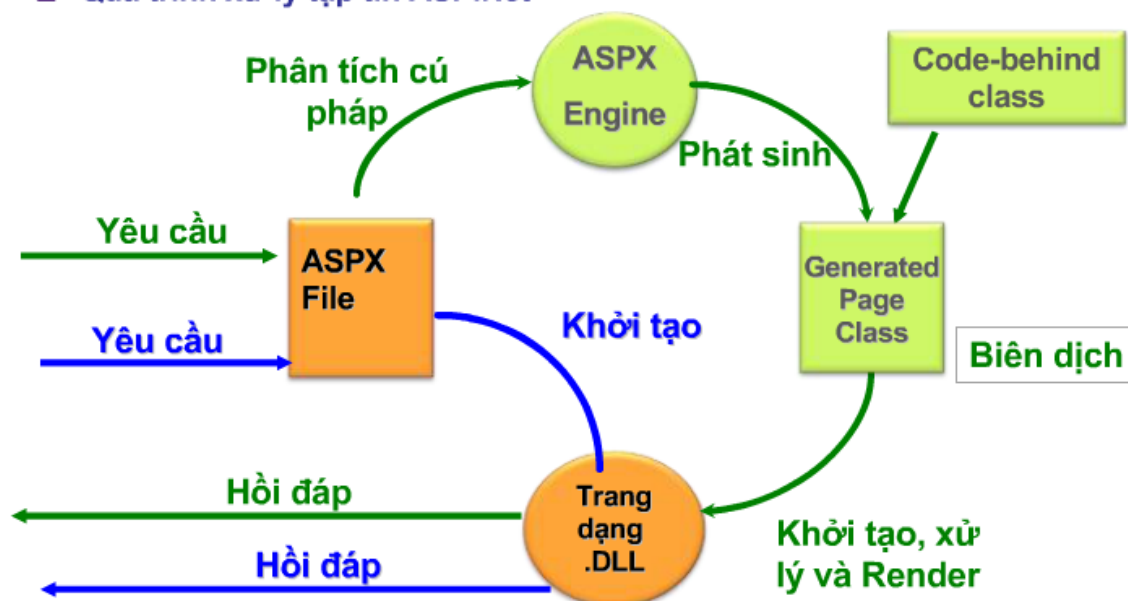


LTV Công Nghệ .Net (2014) – Module 3 - Bài 1. Tổng quan về lập trình web và ASP.Net

25

## 4. Tổng quan về lập trình ứng dụng web

### ❑ Quá trình xử lý tập tin ASP.Net



## Nội dung

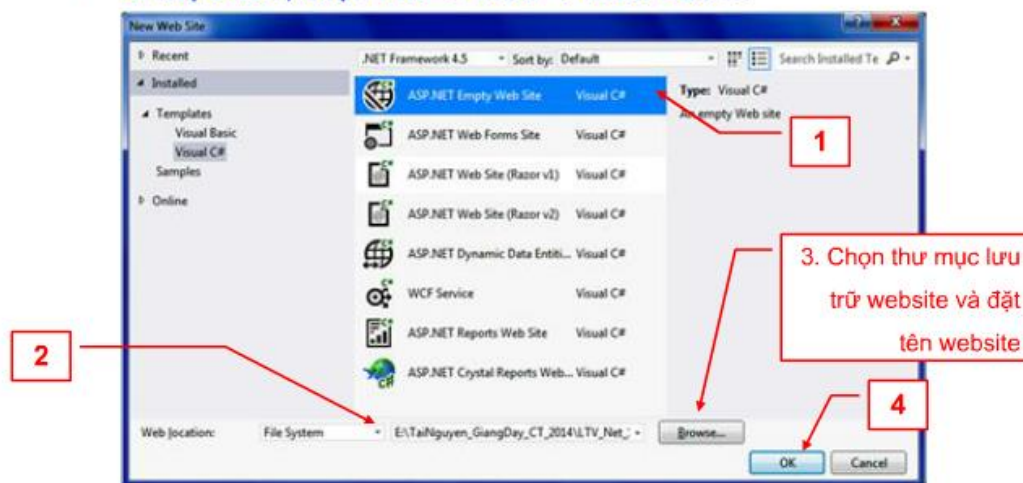
1. Giới thiệu về Internet và một số khái niệm
2. Tổng quan về lập trình web: client – side và server side
3. Giới thiệu về ASP.Net
4. Tổng quan về lập trình ứng dụng web
5. **Tạo ứng dụng web đầu tiên**
  1. Khởi động MS Visual Studio .Net
  2. Giới thiệu các thư mục lưu trữ các đối tượng
  3. Tạo một web form
  4. Đối tượng Page



## 5. Tạo ứng dụng web đầu tiên

### ❑ Khởi động MS Visual Studio .Net

- Start → Microsoft Visual Studio 2012 → Visual Studio 2012
- Tạo mới ứng dụng web:
  - Từ thực đơn, chọn **File** → **New** → **Web Site ...**



LTV Công Nghệ .Net (2014) – Module 3 - Bài 1. Tổng quan về lập trình web và ASP.Net

28

## 5. Tạo ứng dụng web đầu tiên

### ❑ Khởi động MS Visual Studio .Net

- Giới thiệu Web site types
  - **File System:** Không cần cài đặt IIS, Visual studio sẽ cung cấp web server riêng khi ta thi hành ứng dụng
  - **Local HTTP:** Phải có IIS và khai báo URL đã được ánh xạ
  - **FTP :** Sử dụng giao thức truyền file để quản lý các tập tin trong web site



LTV Công Nghệ .Net (2014) – Module 3 - Bài 1. Tổng quan về lập trình web và ASP.Net

29

## 5. Tạo ứng dụng web đầu tiên

### □ Giới thiệu các thư mục lưu trữ các đối tượng

- **App\_Data:** lưu trữ tập tin dữ liệu như .mdb, .mdf, .xml, ...
- **App\_Code:** lưu trữ các tập tin chứa source code, có phần mở rộng .vb, .cs, ...
- **App\_Themes:** lưu trữ các tập tin dùng để tạo giao diện chung theo chủ đề, thường có phần mở rộng .skin, .css hoặc các file hình ảnh
- Thao tác tạo các thư mục trên:



LTV Công Nghệ .Net (2014) – Module 3 - Bài 1. Tổng quan về lập trình web và ASP.Net

30

## 5. Tạo ứng dụng web đầu tiên

### □ Tạo một web form

- Nhấn phím phải chuột, chọn Add → Web Form
- Có phần mở rộng **.aspx** (file giao diện) và **aspx.cs** (file code-behind)
- **Xác định trang khởi động:** nhấn phím phải chuột lên trang được chọn → chọn **Set As Start Page**
- **Thi hành ứng dụng:** nhấn phím **F5**

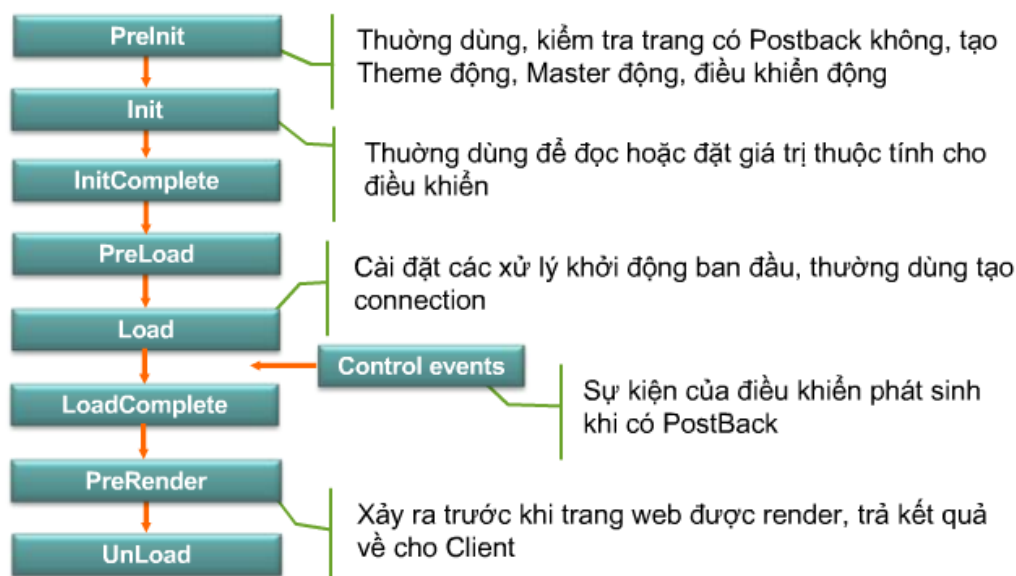




## 5. Tạo ứng dụng web đầu tiên

### □ Đối tượng page

- Các sự kiện trong vòng đời của trang ASP.Net:



LTV Công Nghệ .Net (2014) – Module 3 - Bài 1. Tổng quan về lập trình web và ASP.Net

32

## 5. Tạo ứng dụng web đầu tiên

### □ Đối tượng page

- Thuộc tính
  - **IsPostBack**: Trả về trạng thái PostBack của trang
    - False: Là lần load đầu tiên của trang
    - True: Là lần load lại do submit bởi một control trong Web Form
  - **MaintainScrollPositionOnPostBack**
    - True: Giữ nguyên vị trí mà người dùng đang đọc sau khi PostBack
    - False: Trở về vị trí đầu trang sau khi PostBack



LTV Công Nghệ .Net (2014) – Module 3 - Bài 1. Tổng quan về lập trình web và ASP.Net

33

## Thảo luận





Trường Đại học Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

## LTV CÔNG NGHỆ .NET

### Module 3 - Bài 2. *Tổ chức ASP.Net Website và xây dựng Layout*

Ngành Lập trình & CSDL

<http://www.t3h.vn>

2014



2014





## Nội dung

1. Giới thiệu ứng dụng web
2. Tổ chức CSDL để làm việc với website
3. Tổ chức lưu trữ ứng dụng web
4. Master Pages
5. Menu



## 1. Giới thiệu ứng dụng web

### ❑ Mô tả tổng quát:

- Đây là cửa hàng thời trang, chuyên bán các sản phẩm quần áo thời trang nam, nữ và phụ kiện các loại
- Mỗi sản phẩm thuộc một nhóm sản phẩm. Ví dụ:
  - Áo nơ xinh màu hồng-MS 001 thuộc nhóm Áo đầm
  - Quần jean thun – MS 002 thuộc nhóm Quần jean nữ
  - ...

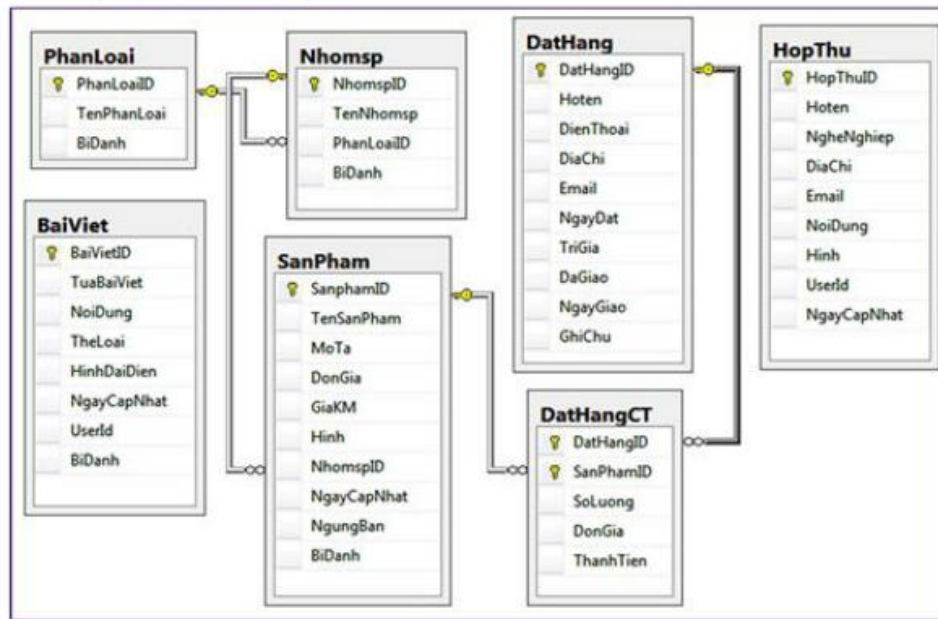
### ❑ Các yêu cầu của ứng dụng:

- Người dùng có thể Tra cứu, tìm kiếm, .... sản phẩm theo tiêu chí đa dạng
- Mua hàng qua mạng
- Đưa tin tức, sự kiện của cửa hàng lên website
- ....



# 1. Giới thiệu ứng dụng web

## □ Giới thiệu cơ sở dữ liệu:



LTV Công Nghệ .Net (2014) – Module 3 - Bài 2. Tổ chức ASP.Net website và xây dựng layout

4

## Nội dung

1. Giới thiệu ứng dụng web
2. Tổ chức CSDL để làm việc với website
  1. Tạo Entity Data Model
  2. Tạo các xử lý nghiệp vụ trong Business Logic Layer (BLL)
  3. Truy vấn dữ liệu bằng eSQL
3. Tổ chức lưu trữ ứng dụng web
4. Master Pages
5. Menu

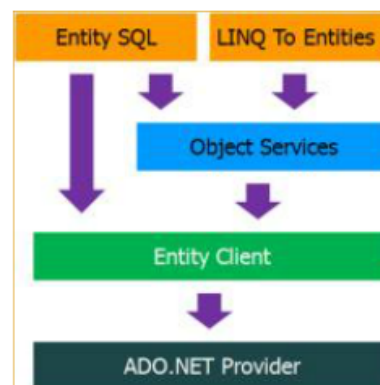


LTV Công Nghệ .Net (2014) – Module 3 - Bài 2. Tổ chức ASP.Net website và xây dựng layout

5

## 2. Tổ chức CSDL để làm việc với website

- ❑ **Tạo Entity Data Model**
  - Tạo project kiểu Library (DAL)
- ❑ **Tạo các class xử lý nghiệp vụ trong Business Logic Layer**
  - Tạo project kiểu Library (BLL)
- ❑ **Truy vấn dữ liệu bằng eSQL (Entity SQL): có 2 cách**
  - Sử dụng EntityClient Provider
  - Sử dụng Object Queries (Object Services)



LTV Công Nghệ .Net (2014) – Module 3 - Bài 2. Tổ chức ASP.Net website và xây dựng layout

6

## 2. Tổ chức CSDL để làm việc với website

- ❑ **Truy vấn dữ liệu bằng eSQL - Sử dụng EntityClient Provider**
  - Sử dụng một thành phần mới của ADO.Net, đó là **EntityClient**
  - EntityClient sẽ không giao tiếp trực tiếp với nguồn dữ liệu mà nó sẽ yêu cầu một provider cụ thể
  - EntityClient sẽ biên dịch câu truy vấn **eSQL** thành cây lệnh và cây lệnh sẽ được chuyển đến provider lưu trữ để phát sinh ra Native SQL
  - EntityClient không cụ thể hóa các đối tượng trong tập hợp kết quả. Thay vào đó, nó sẽ trả về một **DbDataReader**
  - Tương tự ADO.Net Data Provider, EntityClient sẽ sử dụng Connection, Command, ... nhưng gắn thêm tiếp đầu ngữ là "Entity"
    - **EntityConnection, EntityCommand, EntityDataReader, EntityParameter, ...**
  - Khai báo namespace **System.Data.EntityClient**



LTV Công Nghệ .Net (2014) – Module 3 - Bài 2. Tổ chức ASP.Net website và xây dựng layout

7



## 2. Tổ chức CSDL để làm việc với website

### ❑ Truy vấn dữ liệu bằng eSQL - Sử dụng EntityClient Provider

- Chuỗi kết nối (connectionString) được khai báo trong tập cấu hình

#### <connectionStrings>

```
<add name="FASHIONEntities"
connectionString="metadata=res://*/App_Code.Fashion.csdl|res://*/App_Code.Fashion.ssdl|res://*/App_Code.Fashion.msl;provider=System.Data.SqlClient;provider connection string="&quot;data source=.\\sql\\express;initial catalog=FASHION;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;;"
providerName="System.Data.EntityClient" />
```

#### </connectionStrings>

- Provider: Tên của chuẩn provider
- Metadata: Đường dẫn thư mục chứa file mô hình
- Provider Connection String: Chuỗi kết nối chuẩn



LTV Công Nghệ .Net (2014) – Module 3 - Bài 2. Tổ chức ASP.Net website và xây dựng layout

8

## 2. Tổ chức CSDL để làm việc với website



### ❑ Truy vấn dữ liệu bằng eSQL - Sử dụng EntityClient Provider

- Ví dụ:

```
using (EntityConnection conn = new EntityConnection("name=FASHIONEntities")) {
    try {
        EntityCommand cmd = conn.CreateCommand();
        cmd.CommandText = @"SELECT p.SanphamID,p.TenSanPham,d.TenNhomsps,p.DonGia
FROM FASHIONEntities.SanPhams AS p JOIN FASHIONEntities.Nhomsps AS d ON p.Nhomspld=d.Nhomspld";
        conn.Open();
        using (EntityDataReader dr = cmd.ExecuteReader(CommandBehavior.SequentialAccess)) {
            StringBuilder str = new StringBuilder();
            while (dr.Read()) {
                IExtendedDataRecord record = dr as IExtendedDataRecord;
                int id = Convert.ToInt32(record.GetValue(0)); //Đọc dữ liệu record hiện hành .....
                str.AppendLine(string.Format("{0} : {1} : {2} : {3} <br />", id, tensp, tennhom, dg)); } //end while
            IrXuat.Text = str.ToString(); } //end using
        } catch (EntityException ex) { IrXuat.Text = "Lỗi: " + ex.Message; }
        finally { conn.Dispose(); }
    }
}
```



LTV Công Nghệ .Net (2014) – Module 3 - Bài 2. Tổ chức ASP.Net website và xây dựng layout

9



## 2. Tổ chức CSDL để làm việc với website

### ❑ Truy vấn dữ liệu bằng eSQL - Sử dụng EntityClient Provider

- Các quy định khi viết chuỗi truy vấn:
  - Trong mệnh đề **Select**: sử dụng **<alias>.<tên cột>**, nếu không tạo bí danh thì dùng tên thực thể
  - Nếu chọn tất cả các cột thì dùng từ khóa **VALUE**
  - Ví dụ:

```
string chuỗi = @"SELECT p.SanphamID, p.TenSanPham, p.Hinh, p.DonGia
                FROM FASHIONEntities.SanPhams AS p WHERE p.TenSanPham Like 'A%' ";
//hoặc
string chuỗi = @"SELECT Value SanPhams
                FROM FASHIONEntities.SanPhams WHERE SanPhams.TenSanPham Like 'A%' "
```



## 2. Tổ chức CSDL để làm việc với website

### ❑ Truy vấn dữ liệu bằng eSQL - Sử dụng Object Queries(Object Services)

- Khai báo namespace **System.Data.Objects**
- Sử dụng các class:
  - **ObjectContext** : Hỗ trợ truy vấn và làm việc với các thực thể dữ liệu (và xem chúng là đối tượng). Đại diện cho CSDL và quản lý các kết nối
  - **ObjectQuery<T>**: Đại diện cho một truy vấn có định kiểu dựa trên mô hình quan niệm của ObjectContext cụ thể
  - **ObjectSet<TEntity>**: Là một tập hợp các entity. Mỗi đối tượng này tương ứng với một table





## 2. Tổ chức CSDL để làm việc với website

### ❑ Truy vấn dữ liệu bằng eSQL - Sử dụng Object Queries(Object Services)

- Ví dụ 1: Sử dụngObjectContext và ObjectQuery

```
using (ObjectContext db = new ObjectContext("name=FASHIONEntities"))
{
    //Thể hiện tất cả field, lọc các sản phẩm có Tên bắt đầu bằng ký tự A
    string chuoi = @"SELECT Value SanPhams FROM
    FASHIONEntities.SanPhams WHERE SanPhams.TenSanPham Like 'A%'";
    ObjectQuery<SanPham> query = new ObjectQuery<SanPham>(chuoi, db);
    gvSanpham.DataSource = query; //hiển thị lên trang web
    gvSanpham.DataBind();
}
```



## 2. Tổ chức CSDL để làm việc với website

### ❑ Truy vấn dữ liệu bằng eSQL - Sử dụng Object Queries(Object Services)

- Ví dụ 2: Sử dụng ObjectContext và ObjectQuery, truy vấn trả về một số cột tùy chọn

```
using (FASHIONEntities db = new FASHIONEntities())
{
    //Chuyển kiểu DbContext sang ObjectContext
    ObjectContext oContext = ((ObjectContextAdapter)db).ObjectContext;
    string chuoi = @"SELECT p.SanphamID, p.TenSanPham, p.Hinh, p.DonGia
    FROM FASHIONEntities.SanPhams AS p WHERE p.TenSanPham Like 'A%'";
    //Kiểu trả về là vô danh nên phải dùng DbDataRecord (trong System.Data.Common)
    ObjectQuery<DbDataRecord> query = new ObjectQuery<DbDataRecord>(chuoi,
    oContext);
    gvSanpham.DataSource = query;
    gvSanpham.DataBind();
}
```





## 2. Tổ chức CSDL để làm việc với website

### ❑ Truy vấn dữ liệu bằng eSQL - Sử dụng Object Queries(Object Services)

- Ví dụ 3: Sử dụng ObjectSet và các phương thức mở rộng

```
using (FASHIONEntities db = new FASHIONEntities())
{
    var oContext = ((IObjectContextAdapter)db).ObjectContext;
    ObjectSet<SanPham> obset = oContext.CreateObjectSet<SanPham>();
    ObjectQuery<DbDataRecord> results = obset
        .Select("it.SanPhamID, it.TenSanPham, it.DonGia")
        .Where("it.DonGia >= 1500000");
    gvSanpham.DataSource = results;
    gvSanpham.DataBind();
}
```

- Từ khóa **"it"** đại diện cho bí danh của lệnh hiện hành, sử dụng nó trong các phương thức mở rộng



## Nội dung



1. Giới thiệu ứng dụng web
2. Tổ chức CSDL để làm việc với website
3. Tổ chức lưu trữ ứng dụng web
4. Master Pages
5. Menu
6. Tạo Menu bằng Design (menu tĩnh)





### 3. Tổ chức lưu trữ ứng dụng web

#### ❑ Tạo các thư mục:

- App\_code
- App\_data
- Images
- Pages
- ....

(xem chi tiết trong phần Yêu cầu của bài tập)



### Nội dung



1. Giới thiệu ứng dụng web
2. Tổ chức CSDL để làm việc với website
3. Tổ chức lưu trữ ứng dụng web
4. **Master Pages**
  1. Giới thiệu
  2. Tạo Master Page
  3. Tạo Content Page
5. Menu



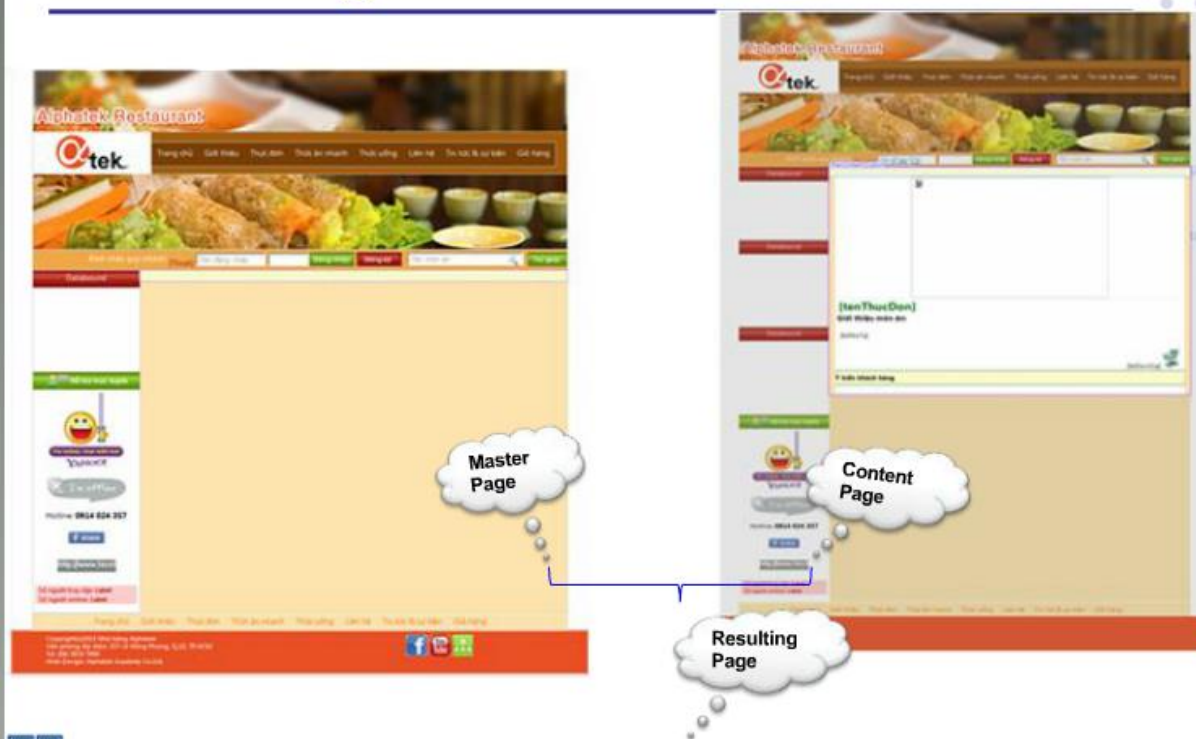
## 4. Master Pages

### ❑ Giới thiệu

- ASP.NET master là trang cho phép tạo ra một bố cục nhất quán (templates) cho các trang web trong cùng một Web site (hoặc một nhóm các trang)
- Là trang được thiết kế thường dùng để chứa các chức năng chung cho các trang trong ứng dụng web
- Trang tham chiếu đến Master Page được gọi là Content Page. Content page là trang nội dung, nó chứa phần nội dung được thiết kế chuyên biệt cho từng trang mà ta muốn hiển thị
- Nội dung thể hiện trên trình duyệt chính là sự kết hợp của Master page (.master) và Content page (.aspx)



## 4. Master Pages



## 4. Master Pages



LTV Công Nghệ .Net (2014) – Module 3 - Bài 2. Tổ chức ASP.Net website và xây dựng layout

20

## 4. Master Pages

### Master file ( \*.master)

```
<%@ Master %>

<asp:ContentPlaceHolder
ID="head" runat="server"/>

<asp:ContentPlaceHolder
ID="MainContent" runat="server" />
```

### Content file ( \*.aspx)

```
<%@ Page
MasterPageFile="~/Shared/Site.master" %>

<asp:Content ID="Content1"
ContentPlaceHolderID="Head" Runat="Server">

</asp:Content>

<asp:Content ID="Content2"
ContentPlaceHolderID="MainContent" Runat="Server">

</asp:Content>
```

### Resulting Page (\* .aspx)

Trang kết quả sẽ thể hiện trên cửa sổ trình duyệt của người dùng

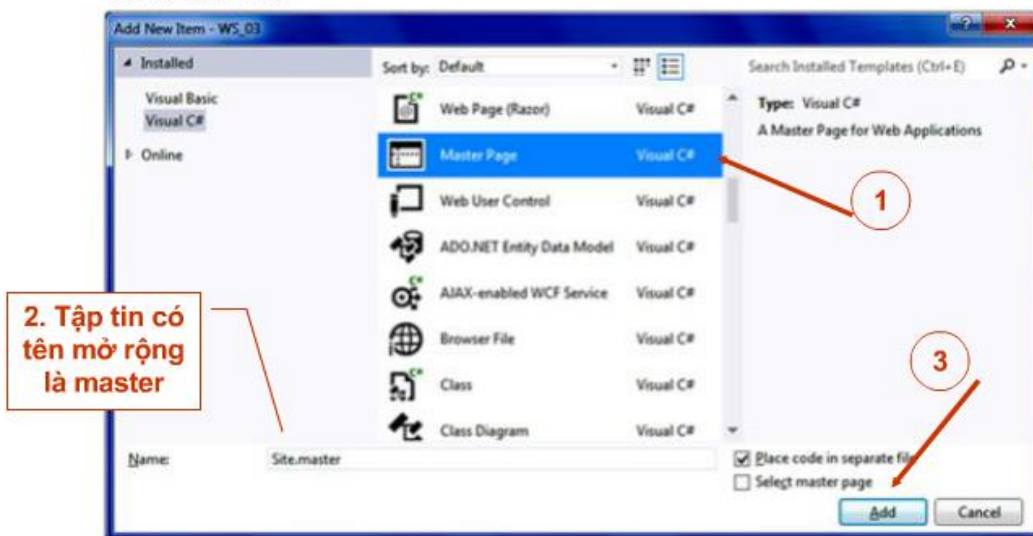
LTV Công Nghệ .Net (2014) – Module 3 - Bài 2. Tổ chức ASP.Net website và xây dựng layout

21

## 4. Master Pages

### ❑ Tạo Master Page

- Nhấn phím phải chuột lên thư mục cần chứa, chọn **Add → Add New Item ...**



LTV Công Nghệ .Net (2014) – Module 3 - Bài 2. Tổ chức ASP.Net website và xây dựng layout

22

## 4. Master Pages

### ❑ Tạo Content Page

- Bước 1: Tạo một Web Form



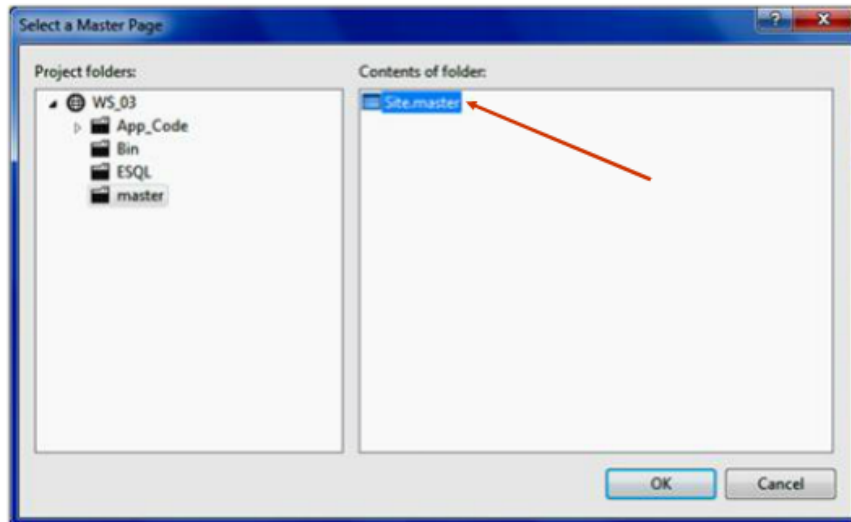
LTV Công Nghệ .Net (2014) – Module 3 - Bài 2. Tổ chức ASP.Net website và xây dựng layout

23

## 4. Master Pages

### ❑ Tạo Content Page

- Bước 2: Chọn Master page tham chiếu



## Nội dung

1. Giới thiệu ứng dụng web
2. Tổ chức CSDL để làm việc với website
3. Tổ chức lưu trữ ứng dụng web
4. Master Pages
5. Menu
  1. Giới thiệu
  2. Phân loại
  3. Thiết kế Menu



## 5. Menu

### □ Giới thiệu

- Menu thường được thiết kế trên trang Master Page
- Các chức năng trên menu thường dùng để xử lý chuyển hướng đến một trang web khác

### □ Phân loại

- Thể hiện theo chiều ngang - Horizontal Menu
- Theo chiều dọc - Vertical Menu



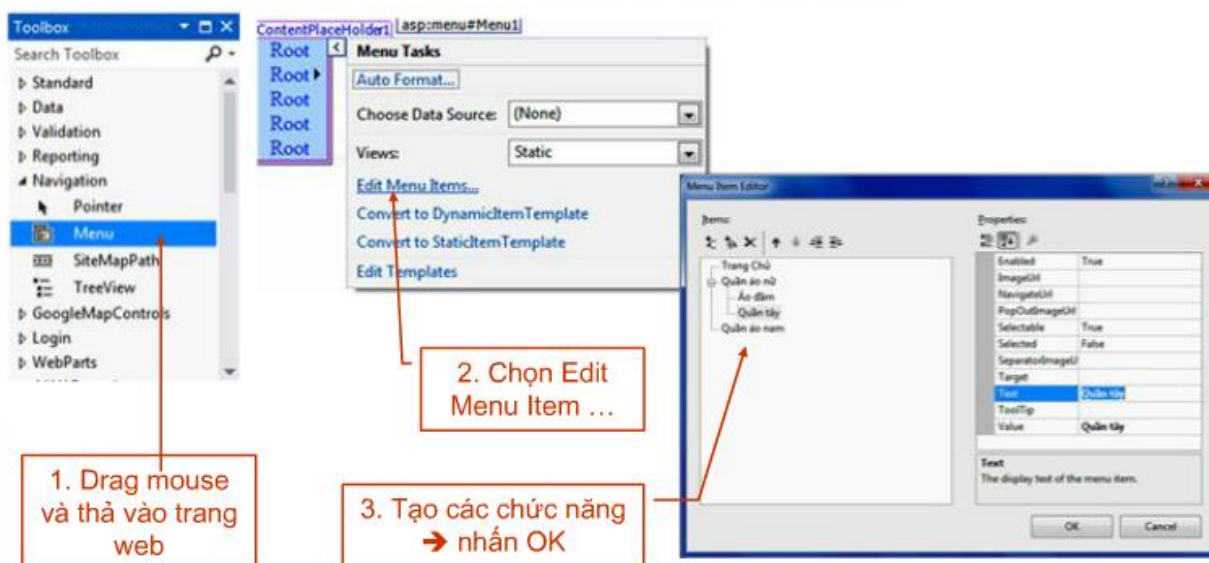
LTV Công Nghệ .Net (2014) – Module 3 - Bài 2. Tổ chức ASP.Net website và xây dựng layout

26

## 5. Menu

### □ Thiết kế Menu

- Thiết kế menu tĩnh: tạo menu bằng **Menu Item Editor**



LTV Công Nghệ .Net (2014) – Module 3 - Bài 2. Tổ chức ASP.Net website và xây dựng layout

27

## Thảo luận





Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

## LTV CÔNG NGHỆ .NET

### Module 3 - Bài 3. *Thiết kế và xử lý Web Forms*

Ngành Lập trình & CSDL

<http://www.t3h.vn>

2014



2014



## Nội dung

### 1. Tổng quan về ASP.Net Server Controls

1. Giới thiệu về các nhóm điều khiển trên thanh Toolbox
2. Các kiểu Server Control

### 2. HTML Server Controls

### 3. Web Server Controls

### 4. ViewState

### 5. Tạo và sử dụng ViewState tùy biến

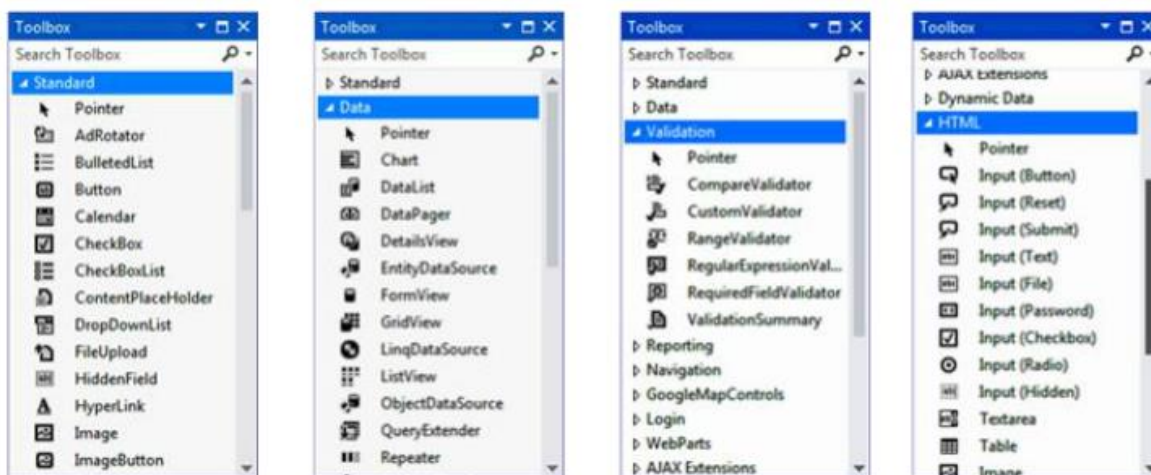


LTV Công Nghệ .Net (2014) – Module 3 - Bài 3. Thiết kế và xử lý web forms

2

## 1. Tổng quan về ASP.Net Server Controls

### □ Giới thiệu về các nhóm điều khiển trên thanh Toolbox



LTV Công Nghệ .Net (2014) – Module 3 - Bài 3. Thiết kế và xử lý web forms

3



# 1. Tổng quan về ASP.Net Server Controls

## □ Các kiểu Server Control

- HTML Server Controls:
  - Là các điều khiển được thiết kế bằng các tag HTML nhưng có khả năng được tham chiếu từ server-side-code
  - Khi nào sử dụng HTML Server Control?
    - Điều khiển cần phải có đoạn JavaScript kèm theo sự kiện
    - Có nhiều code JavaScript tham chiếu đến điều khiển đó
- Web Server Controls:
  - Là các điều khiển do ASP.Net cung cấp, viết xử lý từ server-side



# Nội dung

1. Tổng quan về ASP.Net Server Controls
2. HTML Server Controls
  1. HTML Controls
  2. Chuyển đổi HTML Controls thành HTML Server Controls
3. Web Server Controls
4. ViewState
5. Tạo và sử dụng ViewState tùy biến



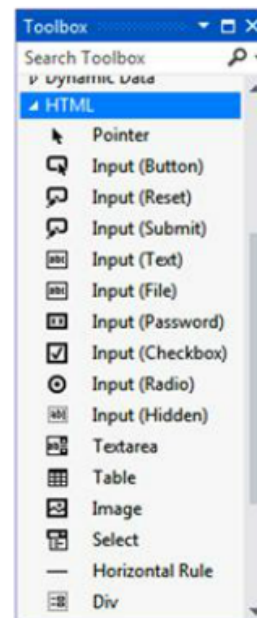
## 2. HTML Server Controls

### □ HTML Controls

- Được tạo ra từ các tag HTML tĩnh
- Thường được sử dụng lập trình ở phía client
- Thuộc namespace `System.Web.UI.HtmlControls`

### □ Chuyển đổi HTML Controls thành HTML Server Controls

- Sử dụng HTML Controls để lập trình ở phía server, ta phải:
  - Bổ sung thuộc tính `runat="server"` vào tag của điều khiển
- Điều khiển có thuộc tính `runat="server"` gọi là HTML Server Controls



## Nội dung

### 1. Tổng quan về ASP.Net Server Controls

### 2. HTML Server Controls

### 3. Web Server Controls

1. Giới thiệu
2. Các điều khiển cơ bản

### 4. ViewState

### 5. Tạo và sử dụng ViewState tùy biến





### 3. Web Server Controls

- ❑ Giới thiệu
- ❑ Label, Literal, TextBox
- ❑ Button, ImageButton, LinkButton
- ❑ Checkbox, RadioButton
- ❑ Image, HyperLink
- ❑ FileUpload
- ❑ MultiView & View



LTV Công Nghệ .Net (2014) – Module 3 - Bài 3. Thiết kế và xử lý web forms

8



### 3. Web Server Controls

- ❑ Giới thiệu
  - Web Server Controls là các đối tượng của .Net Framework
  - Được chuyển đổi sang dạng HTML tĩnh lúc thực thi
  - Thường được sử dụng lập trình ở phía server
  - Thuộc namespace `System.Web.UI.WebControls`
  - HTML Source lúc thiết kế có dạng:

**<asp:Kiểu\_điều\_khiển *Danh\_sách\_thuộc\_tính* runat="server" />**



LTV Công Nghệ .Net (2014) – Module 3 - Bài 3. Thiết kế và xử lý web forms

9



### 3. Web Server Controls

#### ❑ Label

- Dùng để hiển thị thông tin
- **ID**: Tên của điều khiển
- **Text**: Nội dung của điều khiển. Hiểu được các tag định dạng HTML
- **Ví dụ**: `lblNhan.Text = "<b>Đây là chuỗi văn bản được in đậm</b>"`

#### ❑ Literal

- Dùng để hiển thị thông tin
  - Nội dung được chèn trực tiếp vào document, không phát sinh ra thẻ HTML khi gửi kết quả đến Browser
- **Text**: Nội dung của điều khiển.
- **Mode**:
  - **Transform**: Hiểu các tag định dạng HTML
  - **Encode**: Thể hiện các tag định dạng HTML có trong nội dung như một văn bản



LTV Công Nghệ .Net (2014) – Module 3 - Bài 3. Thiết kế và xử lý web forms

10

### 3. Web Server Controls



#### ❑ TextBox

- Dùng để nhập và hiển thị dữ liệu
- **Text**: Nội dung chứa trong TextBox
- **TextMode**: Chế độ hiển thị của Textbox
  - **SingleLine**: một dòng
  - **MultiLine**: nhiều dòng
  - **Password**: hiển thị dạng mật khẩu
- **Wrap**: Cho phép tự động xuống dòng hay không

#### ❑ Image

- Dùng để hiển thị hình ảnh
- **ImageURL**: đường dẫn đến tập tin hình ảnh cần hiển thị.
- **AlternateText**: chuỗi văn bản sẽ hiển thị khi đường dẫn của thuộc tính ImageURL không tồn tại



LTV Công Nghệ .Net (2014) – Module 3 - Bài 3. Thiết kế và xử lý web forms

11

### 3. Web Server Controls

#### ❑ Button, LinkButton, ImageButton



- Sẽ PostBack về server khi được chọn
- Thuộc tính:
  - **CausesValidation** (True/False): Có kiểm tra tính hợp lệ trên trang web không
  - **OnClickClientClick**: ngôn ngữ Client-side script sẽ được thi hành ở phía client



### 3. Web Server Controls

#### ❑ Checkbox và RadioButton

- Dùng xác định mục thông tin được chọn
- Thuộc tính:
  - **Checked**: Cho biết trạng thái của điều khiển có được chọn hay không
  - **TextAlign**: Qui định vị trí hiển thị của điều khiển so với chuỗi văn bản
  - **GroupName**: Nhóm các điều khiển thành một nhóm (dành cho RadioButton)





### 3. Web Server Controls

#### □ HyperLink

- Dùng để tạo ra các liên kết siêu văn bản
- Thuộc tính:
  - **ImageURL**: Hình ảnh hiển thị trên điều khiển (ưu tiên)
  - **Text**: Nội dung hiển thị trên điều khiển
  - **NavigateUrl**: Đường dẫn cần liên kết đến
  - **Target** Dạng cửa sổ hiển thị trang kết quả
    - **\_blank**: Mở ở cửa sổ mới
    - **\_self**: Mở trên chính cửa sổ hiện hành



### 3. Web Server Controls



#### □ FileUpload

- Dùng để upload tập tin về server
- Thuộc tính
  - **HashFile** (True/False): Điều khiển có khai báo tập tin chưa
  - **FileName**: Trả về tên tập tin được upload
- Phương thức
  - **SaveAs**: upload tập tin lên Server





## 3. Web Server Controls

### □ MultiView và View

- Dạng container, dùng chứa các điều khiển khác
- MultiView gồm có nhiều View
- Thuộc tính:
  - **ActiveViewIndex**: Chỉ số của view hiện hành
- Phương thức:
  - **SetActiveView(đối\_tượng\_view)**: Thay đổi view hiện hành



## Nội dung



1. Tổng quan về ASP.Net Server Controls
2. HTML Server Controls
3. Web Server Controls
4. ViewState
  1. Giới thiệu
  2. Sử dụng thuộc tính EnableViewSate
  3. Sử dụng thuộc tính ViewSateMode
5. Tạo và sử dụng ViewState tùy biến





## 4. ViewState

### □ Giới thiệu

- Dùng để lưu lại những thông tin của trang web sau khi Web Server gửi kết quả về cho client  
→ Quản lý tình trạng ở mức client

### □ Sử dụng thuộc tính **EnableViewState**

- Cấp độ **Page**: Cho phép sử dụng đối tượng ViewState của page và các controls trong trang hay không. (True|False)
- Cấp độ **Controls**: Cho phép control sử dụng ViewState hay không.
  - Chỉ có tác dụng khi EnableViewState của Page được gán True



## 4. ViewState



### □ Sử dụng thuộc tính **ViewStateMode**

- Giúp uyển chuyển trong việc bật/tắt trạng thái của điều khiển
  - Như các điều khiển nằm trong cùng một container có thể có những trạng thái khác nhau
- Cấp độ **Page**: Sẽ ảnh hưởng đến các controls trên trang có cài đặt ViewStateMode là Inherit
  - Inherit: Chịu ảnh hưởng bởi ViewStateMode của cấp cha
  - Enabled: Sử dụng ViewState
  - Disabled: Không sử dụng ViewState

**Lưu ý:** Chỉ có tác dụng khi EnableViewState của Page được gán True





## 4. ViewState

### ❑ Sử dụng thuộc tính ViewStateMode

- Cấp độ **Controls**: Cho phép control sử dụng ViewState hay không
  - Inherit: Chịu ảnh hưởng bởi ViewStateMode của cấp cha
  - Enabled: Sử dụng ViewState
  - Disabled: Không sử dụng ViewState

Lưu ý: Chịu ảnh hưởng bởi EnableViewState của Page và của controls



## Nội dung



1. Tổng quan về ASP.Net Server Controls
2. HTML Server Controls
3. Web Server Controls
4. ViewState
5. Tạo và sử dụng ViewState tùy biến





## 5. Tạo và sử dụng ViewState tùy biến

- ❑ Gán giá trị cho ViewState:

**ViewState["KeyName"] = <Giá trị>;**

- ❑ Nhận giá trị:

**<Biến> = ViewState["KeyName"];**

- ❑ Lưu ý:

- "KeyName" có phân biệt chữ HOA/thường
- Chỉ được sử dụng khi EnableViewState của Page được gán True



## Thảo luận





Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

## LTV CÔNG NGHỆ .NET

### Module 3 - Bài 4. *Tạo User Controls và Kiểm tra dữ liệu trên Web Forms*

Ngành Lập trình & CSDL

<http://www.t3h.vn>

2014



2014





## Nội dung

1. Giới thiệu User Controls (UC)
2. Tạo và sử dụng User Controls
3. Load động User Controls
4. Tổng quan về bộ điều khiển kiểm tra dữ liệu (Validation Controls)
5. RequireFieldValidator
6. CompareValidator
7. RangeValidator
8. RegularExpression
9. CustomValidator
10. ValidationSummary



LTV Công Nghệ .Net (2014) – Module 3 - Bài 4. Tạo User Controls và Kiểm tra dữ liệu trên WebForms

2



## 1. Giới thiệu User Controls (UC)

- ☐ Khi kết hợp các control để tạo thành một control mới gọi là Web User Controls (UC)
- ☐ Sử dụng UC cũng tương tự như sử dụng thủ tục hoặc hàm  
→ Khả năng tái sử dụng UC trên các trang web



LTV Công Nghệ .Net (2014) – Module 3 - Bài 4. Tạo User Controls và Kiểm tra dữ liệu trên WebForms

3

## Nội dung

1. Giới thiệu User Controls (UC)
2. Tạo và sử dụng User Controls
  1. Tạo UC
  2. Sử dụng UC
3. Load động User Controls
4. Tổng quan về bộ điều khiển kiểm tra dữ liệu (Validation Controls)
5. RequireFieldValidator
6. CompareValidator
7. RangeValidator
8. RegularExpression
9. CustomValidator
10. ValidationSummary



LTV Công Nghệ .Net (2014) – Module 3 - Bài 4. Tạo User Controls và Kiểm tra dữ liệu trên WebForms

4

## 2. Tạo và sử dụng User Controls

### □ Tạo User Controls

- Nhấn phím phải chuột lên thư mục cần chứa, chọn **Add → Add New Item...**



LTV Công Nghệ .Net (2014) – Module 3 - Bài 4. Tạo User Controls và Kiểm tra dữ liệu trên WebForms

5



## 2. Tạo và sử dụng User Controls

### ❑ Tạo User Controls

- Tập tin lưu trữ giao diện UC có phần mở rộng là `.ascx` và lưu trữ code là `ascx.cs`
- **Thiết kế:** Giống như thiết kế trên trang `aspx`
- Điểm khác biệt với trang `aspx`:
  - Thay thế tag định hướng `<%@ Page ... %>` bằng tag `<%@ Control ... %>`
  - Không có tag `<html></html>` và `<body> </body>`

### ❑ Sử dụng User Controls: (2 cách)

- Cách 1:
  - Kéo và thả UC vào trang `aspx`.
  - Lúc đó, trong trang `aspx` sẽ xuất hiện tag định hướng `<%@ Register ... %>`



## 2. Tạo và sử dụng User Controls

### ❑ Sử dụng User Controls

- Cách 2:
  - Cấu hình trong tập tin `web.config`
  - Cú pháp:

```
<system.web><pages>
  <controls>
    <add tagPrefix="Tiếp đầu ngữ của tag"
      tagName="Tên tag"
      src="<Tên tập tin>.ascx"/>
  </controls>
</pages></system.web>
```





## 2. Tạo và sử dụng User Controls

### ❑ Sử dụng User Controls

- Cách 2:
  - Ví dụ:

```
// Khai báo trong Web.config
<controls >
  <add tagPrefix="Cuoitrang" tagName="ucFooter"
    src="~/UserControls/ucFooter.ascx"/>
</controls>

// Sử dụng trên trang .aspx
<Cuoitrang:ucFooter ID="ucFooter1" runat="server" />
```



## Nội dung



1. Giới thiệu User Controls (UC)
2. Tạo và sử dụng User Controls
3. **Load động User Controls**
  1. Khi nào load động UC?
  2. Các bước thực hiện
  3. Giới thiệu điều khiển Panel và Placeholder
4. Tổng quan về bộ điều khiển kiểm tra dữ liệu (Validation Controls)
5. RequireFieldValidator
6. CompareValidator
7. RangeValidator
8. RegularExpression
9. CustomValidator
10. ValidationSummary





### 3. Load động User Controls

#### ❑ Khi nào load động User Controls?

- Trong một trang aspx có yêu cầu sử dụng nhiều UC trên cùng một vị trí của trang
- Bài toán thường gặp:
  - Khi chuỗi tham số của đối tượng Request thay đổi

#### ❑ Các bước thực hiện trên trang aspx

- Tạo điều khiển Panel hoặc Placeholder
- Trong phương thức Page\_Load, viết đoạn code sau:

```
System.Web.UI.Control theHien = new System.Web.UI.Control();
if (điều_kiện_đúng)
    theHien = Page.LoadControl("đường_dẫn_tập_tin_UC1.ascx");
else
    theHien = Page.LoadControl("đường_dẫn_tập_tin_UC2.ascx");
Điều_khiển_Panel_hoặc_PlaceHolder.Controls.Add(theHien);
```



### 3. Load động User Controls



#### ❑ Giới thiệu điều khiển Panel và Placeholder

- Được sử dụng để chứa các điều khiển khác
- Điều khiển Panel cho phép đặt những điều khiển khác vào bên trong lúc thiết kế còn Placeholder thì không
- Xử lý thêm một điều khiển khác vào lúc thực thi:

```
Loại_điều_khiển tên_biến = new Loại_điều_khiển();
Điều_khiển_Panel_hoặc_PlaceHolder.Controls.Add(tên_biến);
```

- Ví dụ: thêm điều khiển TextBox vào Panel pnNoidung

```
TextBox txtSoA = new TextBox();
pnNoidung.Controls.Add(txtSoA);
```





## Nội dung

1. Giới thiệu User Controls (UC)
2. Tạo và sử dụng User Controls
3. Load động User Controls
4. **Tổng quan về bộ điều khiển kiểm tra dữ liệu (Validation Controls)**
  1. Giới thiệu
  2. Qui trình kiểm tra
  3. Một số thuộc tính cơ bản
5. **RequireFieldValidator**
6. **CompareValidator**
7. **RangeValidator**
8. **RegularExpression**
9. **CustomValidator**
10. **ValidationSummary**



## 4. Tổng quan về bộ điều khiển kiểm tra d.liệu

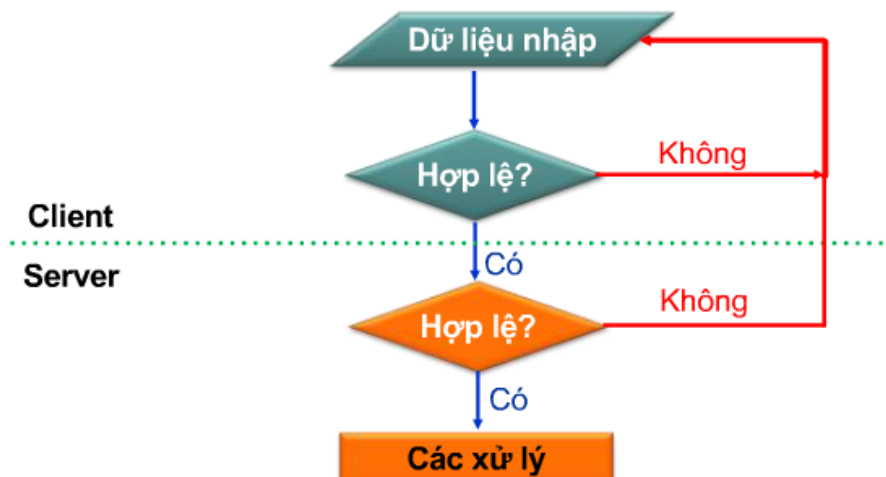
### ☐ Giới thiệu

- Sử dụng bộ điều khiển Validation, còn gọi là Validator Controls để kiểm tra tính hợp lệ của dữ liệu
- Bộ điều khiển này tự động phát sinh code kiểm tra dữ liệu ở client-side tùy thuộc vào web browser có hỗ trợ thực thi script ở client không



## 4. Tổng quan về bộ điều khiển kiểm tra d.liệu

### □ Quy trình kiểm tra



## 4. Tổng quan về bộ điều khiển kiểm tra d.liệu

### □ Một số thuộc tính cơ bản

- **ControlToValidate:** Tên điều khiển cần kiểm tra
- **Text:** Nội dung thông báo xuất hiện khi có lỗi
- **ErrorMessage:** Nội dung thông báo xuất hiện trong điều khiển ValidationSummary
- **EnableClientScript:** Cho phép thực hiện kiểm tra ở phía Client hay không (True/False)
- **SetFocusError(True/False):** Đặt con trỏ vào điều khiển khi dữ liệu không hợp lệ
- **ValidationGroup:** Tên nhóm. Nhóm các điều khiển có cùng giá trị ValidationGroup





## Nội dung

1. Giới thiệu User Controls (UC)
2. Tạo và sử dụng User Controls
3. Load động User Controls
4. Tổng quan về bộ điều khiển kiểm tra dữ liệu (Validation Controls)
5. **RequireFieldValidator**
6. CompareValidator
7. RangeValidator
8. RegularExpression
9. CustomValidator
10. ValidationSummary



## 5. RequireFieldValidator

- ☐ Kiểm tra giá trị trong điều khiển phải được nhập
- ☐ Thuộc tính **InitialValue**:
  - Giá trị khởi động
  - Giá trị nhập vào phải khác với giá trị của thuộc tính này





## Nội dung

1. Giới thiệu User Controls (UC)
2. Tạo và sử dụng User Controls
3. Load động User Controls
4. Tổng quan về bộ điều khiển kiểm tra dữ liệu (Validation Controls)
5. RequireFieldValidator
6. **CompareValidator**
7. RangeValidator
8. RegularExpression
9. CustomValidator
10. ValidationSummary



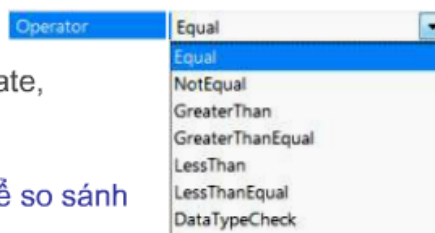
LTV Công Nghệ .Net (2014) – Module 3 - Bài 4. Tạo User Controls và Kiểm tra dữ liệu trên WebForms

18

## 6. CompareValidator



- ☐ So sánh giá trị của một điều khiển với giá trị của một điều khiển khác
- ☐ So sánh giá trị của một điều khiển với một giá trị được xác định trước
- ☐ Kiểm tra kiểu dữ liệu
- ☐ Trong trường hợp không nhập dữ liệu, điều khiển sẽ không thực hiện kiểm tra
- ☐ Thuộc tính
  - **ControlToCompare** (ưu tiên): điều khiển dùng để so sánh
  - **Operator** : toán tử so sánh
  - **Type** (String, Integer, Double, Date, Currency): kiểu dữ liệu
  - **ValueToCompare**: giá trị dùng để so sánh



LTV Công Nghệ .Net (2014) – Module 3 - Bài 4. Tạo User Controls và Kiểm tra dữ liệu trên WebForms

19



## Nội dung

1. Giới thiệu User Controls (UC)
2. Tạo và sử dụng User Controls
3. Load động User Controls
4. Tổng quan về bộ điều khiển kiểm tra dữ liệu (Validation Controls)
5. **RequireFieldValidator**
6. **CompareValidator**
7. **RangeValidator**
8. **RegularExpression**
9. **CustomValidator**
10. **ValidationSummary**



## 7. RangeValidator

- ☐ Kiểm tra giá trị của điều khiển nằm trong đoạn [min-max]
- ☐ Thuộc tính:
  - **MinimumValue**: giá trị nhỏ nhất
  - **MaximumValue**: giá trị lớn nhất
  - **Type**: xác định kiểu dữ liệu để kiểm tra (String/Integer/Double/Date/Currency)





## Nội dung

1. Giới thiệu User Controls (UC)
2. Tạo và sử dụng User Controls
3. Load động User Controls
4. Tổng quan về bộ điều khiển kiểm tra dữ liệu (Validation Controls)
5. RequireFieldValidator
6. CompareValidator
7. RangeValidator
8. **RegularExpression**
9. CustomValidator
10. ValidationSummary



## 8. RegularExpression

- ☐ Kiểm tra giá trị của điều khiển phải theo mẫu qui định
- ☐ Thuộc tính:
  - **ValidationExpression**: qui định mẫu kiểm tra dữ liệu dựa vào các ký hiệu qui định





## Nội dung

1. Giới thiệu User Controls (UC)
2. Tạo và sử dụng User Controls
3. Load động User Controls
4. Tổng quan về bộ điều khiển kiểm tra dữ liệu (Validation Controls)
5. **RequireFieldValidator**
6. **CompareValidator**
7. **RangeValidator**
8. **RegularExpression**
9. **CustomValidator**
10. **ValidationSummary**



## 9. CustomValidator



- ❑ Cho phép người dùng tự viết hàm (Function|Method) xử lý kiểm tra lỗi
- ❑ **Thuộc tính:**
  - **ValidateEmptyText(True/False)**: Có kiểm tra khi giá trị nhập là rỗng ?
  - **ClientValidationFunction**: Tên hàm kiểm tra ở mức client
- ❑ **Sự kiện:**
  - **ServerValidate**: Dùng để xử lý kiểm tra dữ liệu ở mức Server





## Nội dung

1. Giới thiệu User Controls (UC)
2. Tạo và sử dụng User Controls
3. Load động User Controls
4. Tổng quan về bộ điều khiển kiểm tra dữ liệu (Validation Controls)
5. RequireFieldValidator
6. CompareValidator
7. RangeValidator
8. RegularExpression
9. CustomValidator
10. ValidationSummary



LTV Công Nghệ .Net (2014) – Module 3 - Bài 4. Tạo User Controls và Kiểm tra dữ liệu trên WebForms

26



## 10. ValidationSummary

- ❑ **Dùng để hiển thị danh sách các lỗi đang có trên trang Web**
  - Nội dung thông báo lỗi lấy từ nội dung khai báo trong thuộc tính ErrorMessage
- ❑ **Thuộc tính**
  - **HeaderText**: Chuỗi văn bản của dòng tiêu đề
  - **ShowMessageBox(True/False)**: Thẻ hiện hộp thoại thông báo lỗi trên Browser ?
  - **ShowSummary(True/False)**: Thẻ hiện danh sách thông báo lỗi tại vị trí của điều khiển trên Browser ?



LTV Công Nghệ .Net (2014) – Module 3 - Bài 4. Tạo User Controls và Kiểm tra dữ liệu trên WebForms

27

## Thảo luận



LTV Công Nghệ .Net (2014) – Module 3 - Bài 4. Tạo User Controls và Kiểm tra dữ liệu trên WebForms

28



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

## LTV CÔNG NGHỆ .NET

Module 3 - Bài 5. *Xây dựng trang web có liên kết với CSDL (1)*

Ngành Lập trình & CSDL

<http://www.t3h.vn>



2014



## Nội dung

1. Giới thiệu tổng quan về Data Control
2. Các điều khiển thuộc nhóm ListControls
3. Điều khiển GridView
4. Tổng quan về bộ điều khiển DataSource
5. Điều khiển EntityDataSource



## 1. Giới thiệu tổng quan về Data Control

### □ Phân loại:

- Điều khiển liên kết dữ liệu (Data-Bound Controls)
  - Simple: ListBox, DropDownList, CheckBoxLayout, RadioButtonList, BulletedList (còn gọi là ListControl), AdRotator
  - Composite: GridView, DataList, DetailsView, FormView, ListView, Repeater
  - Hierarchical: Menu, TreeView
- Điều khiển nguồn dữ liệu (Data-Source Controls):
  - AccessDataSource
  - LinqDataSource
  - EntityDataSource
  - SqlDataSource
  - ....





# 1. Giới thiệu tổng quan về Data Control

- ❑ Các điều khiển data-bound có khả năng liên kết với một nguồn dữ liệu. Nguồn dữ liệu có thể là:
  - ArrayList, List, SortedList, HashTable
  - DataSet, DataTable, DataView, DataReader
  - ...
- ❑ **Thao tác liên kết dữ liệu của điều khiển Data-Bound: 2 cách**
  - **Lúc thực thi:** Thường sử dụng thuộc tính **DataSource** của điều khiển
  - **Lúc thiết kế:** Sử dụng thuộc tính **DataSourceID** để khai báo liên kết với điều khiển Data source



## Nội dung



1. Giới thiệu tổng quan về Data Control
2. Các điều khiển thuộc nhóm ListControls
  1. ListBox, DropDownList
  2. CheckBoxList, RadioButtonList
  3. BulletedList
3. Điều khiển GridView
4. Tổng quan về bộ điều khiển DataSource
5. Điều khiển EntityDataSource





## 2. Các điều khiển thuộc nhóm ListControl

### □ ListBox và DropDownList

- Dùng để hiển thị danh sách
- Thuộc tính:
  - **Items**: Chứa tập hợp các mục chọn, có thể thêm vào lúc thiết kế (Design) hoặc lúc thực thi (Run-time)
  - **Rows**: Số dòng hiển thị của ListBox
  - **SelectionMode**: Cách thức chọn các mục trong ListBox (Single/Multi)
  - **DataTextField**: Cột liên kết với nguồn dữ liệu, chứa nội dung hiển thị trên điều khiển
  - **DataValueField**: Khai báo cột chứa giá trị lưu trữ
  - **AutoPostBack**: Có tự động PostBack về Server hay không khi mục chọn của điều khiển bị thay đổi (True/False)



## 2. Các điều khiển thuộc nhóm ListControl

### □ ListBox và DropDownList

- Một số thuộc tính lúc thực thi:
  - **SelectedIndex**: Chỉ số của mục được chọn
  - **SelectedItem**: Mục được chọn (kiểu ListItem)
  - **SelectedValue**: Giá trị của mục được chọn
- Sự kiện **SelectedIndexChanged**: Xảy ra khi thay đổi mục chọn
- Một số thao tác:
  - Thêm một mục (item) vào cuối danh sách  
   Items.Add( String )  
   Items.Add( ListItem )
  - Chèn một mục tại vị trí (index) xác định  
   Items.Insert( Index, ListItem )  
   Items.Insert( Index, String )





## 2. Các điều khiển thuộc nhóm ListControl

### □ ListBox và DropDownList

- Một số thao tác:

- Xóa một mục ra khỏi danh sách
  - `Items.Remove( ListItem )`
  - `Items.Remove( String )`
  - `Items.RemoveAt( Index )`
- Xóa tất cả các mục trong danh sách: `Items.Clear()`
- Kiểm tra một mục đã có trong danh sách chưa
  - `Items.Contains( ListItem )` → trả về kiểu bool
  - `Items.IndexOf( ListItem )` → trả về kiểu int
- Lấy giá trị tổng số mục có trong danh sách: `Items.Count`



## 2. Các điều khiển thuộc nhóm ListControl

### □ CheckBoxList và RadioButtonList

- Thuộc tính
  - `RepeatColumns`: Qui định số cột hiển thị
  - `RepeatDirection`: Qui định hướng hiển thị (Vertical/Horizontal)
  - `RepeatLayout`: Qui định dạng thể hiện (Table, Flow, UnorderedList, OrderedList)
- Các thao tác thêm, xóa các mục chọn: Tương tự ListBox

### □ BulletedList

- Thuộc tính:
  - `BulletStyle`: Kiểu biểu tượng thể hiện
  - `BulletImageUrl`: URL của tập tin hình ảnh, khi `BulletStyle` là `CustomImage`
  - `DisplayMode`: Hình thức thể hiện (`Text`, `HyperLink`, `LinkButton`)
- Các thao tác thêm, xóa các mục chọn: giống ListBox





## 2. Các điều khiển thuộc nhóm ListControl

### ❑ Code liên kết nguồn dữ liệu

STT	Lệnh	Ý nghĩa dòng lệnh
1	<code>ArrayList nguonDL = new ArrayList();</code>	Khởi tạo nguồn dữ liệu
2	<code>nguonDL.Item.Add(object);</code>	Bổ sung dữ liệu
	...	
3	<code>&lt;ListControls&gt;.DataSource = nguonDL;</code>	Khai báo liên kết với nguồn DL
4	<code>&lt;ListControls&gt;.DataTextField = "cột_hiển_thị";</code>	Khai báo cột chứa nội dung hiển thị
5	<code>&lt;ListControls&gt;.DataValueField = "cột_giá_trị";</code>	Khai báo cột chứa nội dung lưu trữ
6	<code>&lt;ListControls&gt;.DataBind();</code>	Thực hiện nạp DL lên điều khiển

- Dòng 4 và 5 có thể bỏ qua, tùy vào nguồn dữ liệu



## Nội dung



1. Giới thiệu tổng quan về Data Control
2. Các điều khiển thuộc nhóm ListControls
3. Điều khiển GridView
  1. Giới thiệu
  2. Các thao tác trên GridView
4. Tổng quan về bộ điều khiển DataSource
5. Điều khiển EntityDataSource



### 3. Điều khiển GridView

#### □ Giới thiệu

- Dùng hiển thị dữ liệu dạng dòng-cột (Tabular)
- Thể hiện trên trình duyệt dưới dạng HTML Table
- Đặc điểm:
  - Hỗ trợ cơ chế sắp xếp dữ liệu, cơ chế phân trang dữ liệu
  - Có thể cập nhật dữ liệu trực tiếp trên lưới

ID	Tên sản phẩm	Đơn giá	Ngày cập nhật	Ngừng bán
6	 Áo ba lỗ dễ thương lắm đây	150,000	22/12/2013	<input type="checkbox"/>
7	 Áo nơ xinh form dài	220,000	22/12/2013	<input type="checkbox"/>
8	 Áo kiểu set đôi	190,000	16/03/2011	<input type="checkbox"/>
9	 Quần jeans Nam thời trang Hàn	480,000	17/03/2011	<input type="checkbox"/>
10	 Áo kiểu công sở Ánes_i	150,000	01/03/2011	<input type="checkbox"/>

1 2 3 4 5 6 7 8 9 10 ...



LTV Công Nghệ .Net (2014) – Module 3 - Bài 5. Xây dựng trang web có liên kết với CSDL (1)

12

### 3. Điều khiển GridView

#### □ Các thao tác

- Liên kết dữ liệu và định dạng dữ liệu:
  - Thao tác liên kết dữ liệu lúc thực thi:
 

```
<GridView>.DataSource = nguồn_dữ_liệu;
<GridView>.DataBind();
```
  - Thuộc tính định dạng:
    - `HtmlEncoded(True/False)`: Cho phép mã hóa dưới dạng HTML không?
    - `DataFormatString`: Định dạng cho giá trị của field liên kết
      - Kiểu số: {0:##,###0}
      - Kiểu ngày: {0:dd/MM/yyyy}



LTV Công Nghệ .Net (2014) – Module 3 - Bài 5. Xây dựng trang web có liên kết với CSDL (1)

13

### 3. Điều khiển GridView

#### □ Các thao tác

##### • Phân trang, sắp xếp

- Khai báo thuộc tính
  - AllowPaging: **True**
  - PageSize: Số dòng hiển thị trên 1 trang
- Sử dụng sự kiện để xử lý
  - PageIndexChanged:

ID	Tên sản phẩm	Đơn giá	Ngày cập nhật	Ngừng bán
6	 Áo ba lỗ dễ thương tím đầy	150,000	22/12/2013	<input type="checkbox"/>
7	 Áo nơ xinh form dài	220,000	22/12/2013	<input type="checkbox"/>
8	 Áo kiểu set dài	190,000	16/03/2011	<input type="checkbox"/>
9	 Quần jeans Nam thời trang Hàn	480,000	17/03/2011	<input type="checkbox"/>
10	 Áo kiểu công sở Ames_i	150,000	01/03/2011	<input type="checkbox"/>

```
<GridView>.PageIndex = e.NewPageIndex ;
```

```
//Tạo liên kết với dữ liệu
```

```
...
```

- ✓ **e.NewPageIndex**: Chỉ số trang mới được chọn
- ✓ **PageIndex**: Chỉ số trang hiện hành trên lưới



### 3. Điều khiển GridView

#### □ Các thao tác

##### • Sắp xếp dữ liệu

- Khai báo thuộc tính AllowSorting = True
- Khai báo tên cột cần sắp xếp trong thuộc tính Sort Expression
- Sử dụng sự kiện Sorting để xử lý:
  - Liên kết dữ liệu và sắp xếp theo cột cần sắp xếp
  - Giá trị **e.SortExpression** cho biết thông tin của cột được chọn sắp xếp



### 3. Điều khiển GridView

#### □ Các thao tác

##### • Thiết kế các cột tùy biến:

- Bước 1: Khi thiết kế, cột thể hiện có loại là **TemplateField**
- Bước 2: Dành cho xử lý đơn giản

- ✓ Chèn một điều khiển vào cột TemplateField, tạo liên kết dữ liệu cho điều khiển này
- ✓ Khai báo biểu thức trong ô Custom binding → Code Expression
- ✓ Ví dụ:

```
//Hiển thị cột NgungBan là "Còn hàng" hay "Ngưng bán"
(bool) Eval("NgungBan")==False?"Còn hàng":"Ngưng bán"
```

ID	Tên sản phẩm	Đơn giá	Ngày cập nhật	
1	Áo cách điện...HOT...HOT	220,000	17/03/2011	Còn hàng
2	Áo thun KOREA mới về...Kate	150,000	16/03/2011	Ngưng bán
3	Khác thun nỉn cực đẹp!!!	190,000	22/12/2013	Còn hàng
4	Áo thun Nữ xù tui cực xinh, xanh	220,000	16/03/2011	Còn hàng
5	Thun sọc cá tính!!!!	220,000	22/12/2013	Ngưng bán



LTV Công Nghệ .Net (2014) – Module 3 - Bài 5. Xây dựng trang web có liên kết với CSDL (1)

16

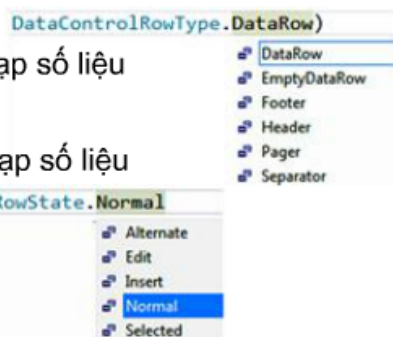
### 3. Điều khiển GridView

#### □ Các thao tác

##### • Thiết kế các cột tùy biến:

- Bước 2': Dành cho xử lý phức tạp, viết code trong sự kiện:
  - **RowDataBound**: Xảy ra cho mỗi dòng khi GridView được nạp dữ liệu (DataBinding)
    - **e.Row.RowIndex**: Chỉ số dòng hiện hành (Header/Footer=-1)
    - **e.Row.RowType**: Loại dòng đang được nạp số liệu
    - **e.Row.RowState**: Trạng thái dòng được nạp số liệu

```
e.Row.RowState= DataControlRowState.Normal
```



LTV Công Nghệ .Net (2014) – Module 3 - Bài 5. Xây dựng trang web có liên kết với CSDL (1)

17



### 3. Điều khiển GridView

#### □ Các thao tác

##### • Thiết kế các cột tùy biến:

- Bước 2': Dành cho xử lý phức tạp, viết code trong sự kiện:
  - `e.Row.FindControl("Tên điều khiển")`: Trả về tham chiếu đến điều khiển trên dòng hiện hành
  - `DataBinder.Eval(e.Row.DataItem, "Tên cột liên kết")`: Truy xuất dữ liệu của cột liên kết trên dòng hiện hành
- Ví dụ:

```
protected void gvSanPham_RowDataBound(object sender, GridViewRowEventArgs e) {
    if (e.Row.RowType == DataControlRowType.DataRow) {
        System.Web.UI.WebControls.Image Nut =
            (System.Web.UI.WebControls.Image)e.Row.FindControl("imgNgung");
        bool ngung = (bool)DataBinder.Eval(e.Row.DataItem, "NgungBan"); //hoặc
        //bool ngung = ((SanPham)e.Row.DataItem).NgungBan;
        Nut.ImageUrl = "~/Images/" + (ngung == true ? "stop.jpg" : "ready.png");
    }
}
```



LTV Công Nghệ .Net (2014) – Module 3 - Bài 5. Xây dựng trang web có liên kết với CSDL (1)

18

### 3. Điều khiển GridView



#### □ Các thao tác

- Giới thiệu 3 loại cột trong GridView: ImageField
  - `DataImageUrlField`: Chứa Image URL liên kết với field
  - `DataImageUrlFormatString`: Định dạng cho giá trị liên kết với thuộc tính `DataImageUrlField`
- Ví dụ:

Data	
DataAlternateTextField	
DataAlternateTextFormatString	
DataImageUrlField	Hình
DataImageUrlFormatString	~/Photos/{0}



LTV Công Nghệ .Net (2014) – Module 3 - Bài 5. Xây dựng trang web có liên kết với CSDL (1)

19

### 3. Điều khiển GridView

#### □ Các thao tác

- Giới thiệu 3 loại cột trong GridView: **HyperLinkField**
  - **DataNavigateUrlFields**: Tên cột chứa giá trị là URL
  - **DataNavigateUrlFormatString**: Định dạng cho giá trị liên kết với thuộc tính DataNavigateUrlFields
  - **DataTextField**: Tên cột chứa nội dung được hiển thị
  - Ví dụ:

<b>Data</b>	
DataNavigateUrlFields	SanPhamID
DataNavigateUrlFormatString	~/Pages/ChiTietSP.aspx?id={0}
DataTextField	TenSanPham
DataTextFormatString	



LTV Công Nghệ .Net (2014) – Module 3 - Bài 5. Xây dựng trang web có liên kết với CSDL (1)

20

### 3. Điều khiển GridView

#### □ Các thao tác

- Giới thiệu 3 loại cột trong GridView: **ButtonField**
  - **ButtonType**: Kiểu nút lệnh (Button/Link/Image)
  - **CommandName**: Tên kết hợp với nút lệnh
  - **DataTextField**: Cột chứa nội dung hiển thị
  - **DataTextFormatString**: Định dạng nội dung thể hiện trên nút
  - Ví dụ:

<b>Behavior</b>	
CausesValidation	False
CommandName	cmdChon
InsertVisible	True
ShowHeader	False
SortExpression	
ValidationGroup	
Visible	True
<b>Data</b>	
DataTextField	SanPhamID
DataTextFormatString	Chọn: {0}



LTV Công Nghệ .Net (2014) – Module 3 - Bài 5. Xây dựng trang web có liên kết với CSDL (1)

21



### 3. Điều khiển GridView

#### □ Các thao tác

- Giới thiệu 3 loại cột trong GridView: **ButtonField**
  - Viết xử lý trong sự kiện **RowCommand**: xảy ra khi nhấn nút lệnh
  - Ví dụ: Xuất tên sản phẩm sau khi nhấn nút Chọn

```
protected void gvw_RowCommand(object sender, GridViewCommandEventArgs e)
{
    if (e.CommandName == "cmdChon") {
        //Xác định chỉ số dòng được chọn, áp dụng với loại ButtonField
        Int16 chiso = Convert.ToInt16(e.CommandArgument);
        //Xác định dòng được chọn thông qua tập hợp Rows
        GridViewRow dong = (GridViewRow)gvSanpham.Rows[chiso];
        string Ten = dong.Cells[2].Text; //lấy giá trị ô Tên sản phẩm của dòng được chọn
        lblXuat.Text = Ten; //Xuất ra màn hình
    }
}
```



LTV Công Nghệ .Net (2014) – Module 3 - Bài 5. Xây dựng trang web có liên kết với CSDL (1)

22

### 3. Điều khiển GridView

#### □ Các thao tác

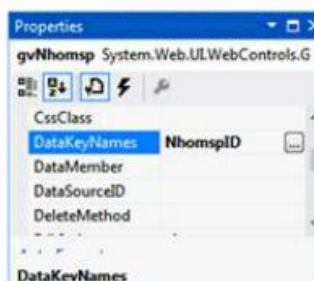
- Cập nhật dữ liệu (sửa, hủy):
  - Yêu cầu đầu tiên: Khai báo các cột dùng làm khóa cho lưới
  - Cách 1: Viết lệnh

ID	Tên nhóm sản phẩm	Phân loại ID	
1	Áo sơ mi nữ	1	Sửa Hủy Chọn
2	Áo sơ mi nam	2	Sửa Hủy Chọn
3	Áo khoác nữ	1	Ghi Không
4	Áo khoác nam	2	Sửa Hủy Chọn
5	Áo thun nam	2	Sửa Hủy Chọn
1 2 3 4 5			

**<GridView>.DataKeyNames = new string[] { "Tên field1", ... };**

Ví dụ: **gvwNhom.DataKeyNames = new string[] { "NhomspID" };**

- Cách 2: Khai báo trên cửa sổ Properties



LTV Công Nghệ .Net (2014) – Module 3 - Bài 5. Xây dựng trang web có liên kết với CSDL (1)

23



### 3. Điều khiển GridView

#### □ Các thao tác

- Cập nhật dữ liệu (sửa, hủy):
  - Bước 1 – Thiết kế: chọn loại cột thể hiện là **CommandField**

Nút lệnh	CommandName	Ý nghĩa sử dụng
Edit	Edit	Chuyển sang chế độ hiệu chỉnh (Edit) thông tin
Update	Update	Cập nhật thông tin về CSDL
Cancel	Cancel	Khôi phục dữ liệu ban đầu và trở về trạng thái <b>ReadOnly</b>
Delete	Delete	Hủy mẫu tin khỏi CSDL
Select	Select	Chọn dòng làm việc



### 3. Điều khiển GridView



#### □ Các thao tác

- Cập nhật dữ liệu (sửa, hủy):
  - Bước 2 – Xử lý: Sử dụng các sự kiện sau:

Sự kiện phát sinh	Ứng với nút lệnh có Commandname
RowEditing	Edit
RowUpdating	Update
RowCancelingEdit	Cancel
RowDeleting	Delete
SelectedIndexChanged	Select
SelectedIndexChanged	Select (sau khi chọn)
RowCommand	Tùy chọn





### 3. Điều khiển GridView

#### □ Các thao tác

- Cập nhật dữ liệu (sửa, hủy):
  - [Bước 2 – Xử lý](#): Sử dụng các sự kiện sau:

Sự kiện phát sinh	Ứng với nút lệnh có CommandName	Ý nghĩa sử dụng các sự kiện
RowEditing	<b>Edit</b>	Xảy ra khi nhấn nút Edit, trước khi GridView chuyển sang chế độ edit
RowUpdating	<b>Update</b>	Xảy ra khi nhấn nút Update, trước khi GridView cập nhật
RowCancelingEdit	<b>Cancel</b>	Xảy ra khi nhấn nút Cancel, trước khi GridView thoát khỏi chế độ edit
RowDeleting	<b>Delete</b>	Xảy ra khi nhấn nút Delete, trước khi GridView xóa dòng
SelectedIndexChanged	<b>Select</b>	Khi nhấn nút Select
SelectedIndexChanged	<b>Select</b>	Khi dòng chọn đã được xác định
RowCommand	<b>Tùy chọn</b>	



### 3. Điều khiển GridView

#### □ Các thao tác

- Cập nhật dữ liệu (sửa, hủy):
  - [Bước 2 – Xử lý](#): kết hợp với các thuộc tính sau:
    - `<GridView>.EditIndex`: Xác định chỉ số dòng được sửa
    - `<GridView>.DataKeys[i].Value`: Truy xuất giá trị khóa tại dòng có chỉ số i
    - `<GridView>.DataKeys[i].Values[n]`: Truy xuất giá trị n khóa tại dòng có chỉ số i, trong trường hợp khóa là một tập hợp của nhiều cột
    - `<GridView>.EnablePersistedSelection`: Giữ nguyên dòng được chọn khi sang trang mới





### 3. Điều khiển GridView

#### □ Các thao tác

- Cập nhật dữ liệu (sửa, hủy):
  - Bước 2 – Xử lý: kết hợp với các thuộc tính sau:
    - Thuộc tính của tham số **e** trong sự kiện **RowEditing**
      - **NewEditIndex**: Chỉ số của dòng đang chuyển sang chế độ chỉnh sửa (Edit)
    - Thuộc tính của tham số **e** trong sự kiện **RowUpdating**
      - **RowIndex**: Chỉ số của dòng hiện hành
      - **Rows[ i ].Cells[ j ].Controls[n]**: Tham chiếu đến một điều khiển trong một cell



### Nội dung



1. Giới thiệu tổng quan về Data Control
2. Các điều khiển thuộc nhóm ListControls
3. Điều khiển GridView
4. Tổng quan về bộ điều khiển DataSource
5. Điều khiển EntityDataSource





## 4. Tổng quan về bộ điều khiển DataSource

- ❑ Là các điều khiển dùng để kết nối và truy xuất dữ liệu từ các nguồn CSDL như: MsAccess, Sql Server, ..... Nó sẽ hỗ trợ các thao tác cho các điều khiển liên kết (binding) vào nó
  - ❑ Khi các điều khiển binding vào nó thì sẽ được hỗ trợ các tính năng sau:
    - Hỗ trợ phân trang
    - Hỗ trợ sắp xếp tăng/giảm tự động
    - Cập nhật dữ liệu
- ➔ Tất cả các xử lý trên gần như không cần viết code



## 4. Tổng quan về bộ điều khiển DataSource

- ❑ **Các điều khiển DataSource**
  - **AccessDataSource**: Kết nối đến tập tin CSDL MS Access
  - **SqlDataSource** : Kết nối đến CSDL SQL Server, Oracle, ...
  - **XmlDataSource** : Kết nối đến tập tin XML
  - **ObjectDataSource** : Kết nối đến một object (class, ...) hoặc Typed-DataSet đặt trong thư mục Bin | App\_Code
  - **SitemapDataSource**: Liên kết đến tập tin .sitemap
  - **LinqDataSource**: Kết nối đến dữ liệu theo kỹ thuật LINQToSQL
  - **EntityDataSource**: Kết nối đến dữ liệu theo kỹ thuật LINQToEntity





## Nội dung

1. Giới thiệu tổng quan về Data Control
2. Các điều khiển thuộc nhóm ListControls
3. Điều khiển GridView
4. Tổng quan về bộ điều khiển DataSource
5. Điều khiển EntityDataSource
  1. Giới thiệu
  2. Thao tác cấu hình
  3. Các thuộc tính cơ bản
  4. Liên kết dữ liệu với GridView
  5. Một số các thao tác
  6. Cập nhật dữ liệu



## 5. Điều khiển EntityDataSource

### □ Giới thiệu

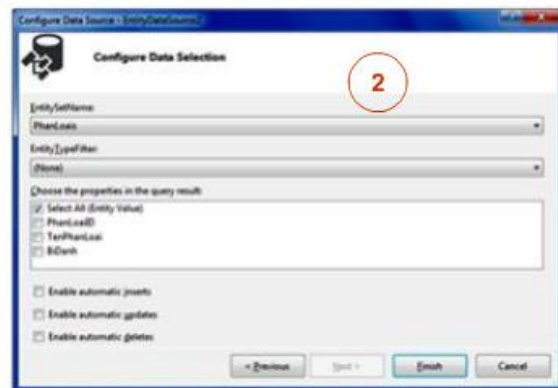
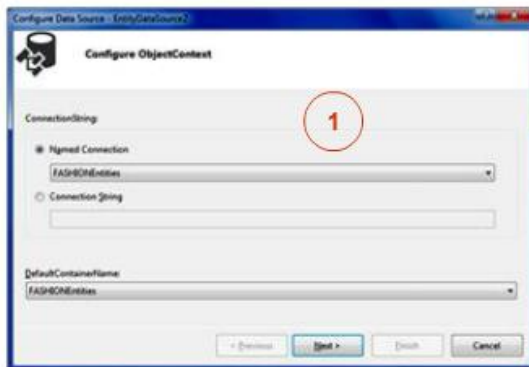
- EntityDataSource dùng để liên kết với dữ liệu từ một CSDL quan hệ mà Entity Framework cho phép
- Có khả năng lọc, sắp xếp và nhóm dữ liệu
- EntityDataSource sẽ tự động thực hiện cập nhật dữ liệu dựa trên mô hình thực thể dữ liệu (EDM) của Entity Framework



## 5. Điều khiển EntityDataSource

### ❑ Thao tác cấu hình

- Bước 1: Cấu hìnhObjectContext (Khai báo Connection & EDM)



- Bước 2: Chọn nguồn dữ liệu → Click vào nút **Finish**



## 5. Điều khiển EntityDataSource

### ❑ Các thuộc tính cơ bản

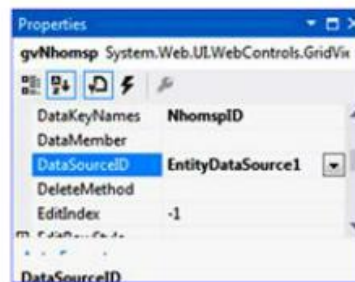
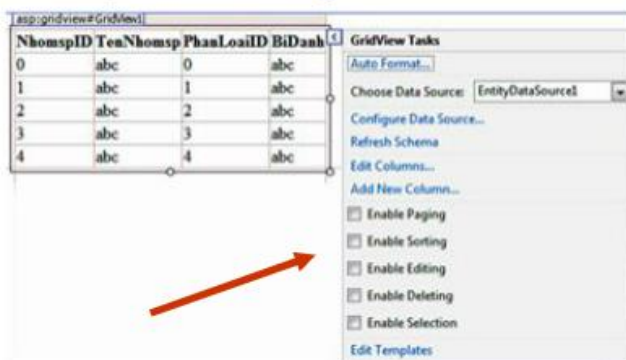
- **ConnectionString**: Khai báo kết nối
- **DefaultContainerName**: Khai báo tên EDM sử dụng
- **EntitySetName**: Khai báo tên entity
- **Select**: Các cột thông tin cần rút trích
- **Where**: Điều kiện lọc
- **GroupBy**: Thông tin gom nhóm dữ liệu
- **OrderBy**: Thông tin sắp xếp dữ liệu
- **EnableDelete**: (True) Cho phép hủy
- **EnableInsert**: (True) Cho phép thêm
- **EnableUpdate**: (True) Cho phép sửa



## 5. Điều khiển EntityDataSource

### ❑ Liên kết dữ liệu với GridView

- Gán giá trị cho thuộc tính **DataSourceID**
- Bật các chức năng cần thiết



## 5. Điều khiển EntityDataSource

### ❑ Sắp xếp dữ liệu

- Sử dụng thuộc tính:
  - **OrderBy**: Khai báo các cột cần sắp xếp
  - **AutoSort**: **True**

### ❑ Chọn lựa dữ liệu

- Sử dụng thuộc tính **Select**, khai báo theo các tùy chọn sau:
  - Để trống nếu trả về tất cả các cột
  - Chọn một số cột thì sử dụng ngôn ngữ Entity SQL
  - Ví dụ: `it.[TenSanPham], it.[DonGia], it.[MoTa], ...`





## 5. Điều khiển EntityDataSource

### ❑ Lọc dữ liệu

- Sử dụng thuộc tính **Where**, khai báo biểu thức điều kiện:
  - Có thể sử dụng các toán tử AND hoặc OR
  - Có thể sử dụng tham số trong biểu thức
  - Ví dụ:

```
it.TenSanPham Like "%jean%"
//Hoặc có tham số
it.SanPhamID = @Ma
```



## 5. Điều khiển EntityDataSource

### ❑ Nhóm và thống kê dữ liệu

- Sử dụng thuộc tính:
  - GroupBy**: Khai báo các cột cần nhóm
  - Sử dụng các hàm sau trong thuộc tính **Select**: **Sum(cột)**, **Count()**, **Min(cột)**, **Max(cột)**, **Average(cột)**
  - Ví dụ:

```
GroupBy: it.NhomspID, it.[Nhomsp].TenNhomsp
Select: it.NhomspID, it.Nhomsp.TenNhomsp,
Count(it.SanPhamID) As Sosp, Sum(it.DonGia) As TongTien,
Avg(it.DonGia) As DGTB, Min(it.DonGia) As DGMin , ...
```

### ❑ Thêm, xóa, sửa và cập nhật dữ liệu

- Bật các thuộc tính: **EnableDelete**, **EnableInsert**, **EnableUpdate**



## Thảo luận



LTV Công Nghệ .Net (2014) – Module 3 - Bài 5. Xây dựng trang web có liên kết với CSDL (1)

40



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

## LTV CÔNG NGHỆ .NET

Module 3 - Bài 6. *Xây dựng trang web có liên  
kết với CSDL (2)*

Ngành Lập trình & CSDL

<http://www.t3h.vn>

2014



2014



## Nội dung

1. Giới thiệu điều khiển DetailsView
2. Các thao tác trên DetailsView
3. Giới thiệu điều khiển ListView
4. Các thao tác trên ListView



LTV Công Nghệ .Net (2014) – Module 3 - Bài 6. Xây dựng trang web có liên kết với CSDL (2)

2

## 1. Giới thiệu điều khiển DetailsView

- ☐ Dùng hiển thị dữ liệu dạng cột (Columnar)
- ☐ Thể hiện trên trình duyệt dưới dạng HTML Table
- ☐ Đặc điểm:
  - Hỗ trợ cơ chế phân trang dữ liệu
  - Cho phép cập nhật dữ liệu trực tiếp
  - Cho phép kết hợp với GridView để tạo ra dạng Master-Detail

Cập nhật nhóm sản phẩm	
ID	11
Tên nhóm sản phẩm	Quần kaki nữ
Phân loại ID	1
Bí danh	quan-kaki-nu
<a href="#">Sửa</a>   <a href="#">Hủy</a>   <a href="#">Thêm</a>	
<< ... 11 12 13 14 15 16 17 18 19 20 ... >>	



LTV Công Nghệ .Net (2014) – Module 3 - Bài 6. Xây dựng trang web có liên kết với CSDL (2)

3



## Nội dung

1. Giới thiệu điều khiển DetailsView
2. Các thao tác trên DetailsView
  1. Liên kết dữ liệu và định dạng dữ liệu
  2. Phân trang
  3. Thiết kế các cột tùy biến
  4. Cập nhật dữ liệu
  5. Tạo dạng Master-Detail
3. Giới thiệu điều khiển ListView
4. Các thao tác trên ListView



LTV Công Nghệ .Net (2014) – Module 3 - Bài 6. Xây dựng trang web có liên kết với CSDL (2)

4



## 2. Các thao tác trên DetailsView

- ❑ Các thao tác:
- Liên kết dữ liệu và định dạng dữ liệu
  - Phân trang
  - Thiết kế các cột tùy biến
- ➔ Tất cả đều tương tự điều khiển GridView



LTV Công Nghệ .Net (2014) – Module 3 - Bài 6. Xây dựng trang web có liên kết với CSDL (2)

5



## 2. Các thao tác trên DetailsView

### ❑ Cập nhật dữ liệu

- **Thiết kế:** Sử dụng bộ nút CommandField giống như GridView
- **Xử lý:** Tương tự GridView, chỉ khác nhau về tên sự kiện và một số thuộc tính

Sự kiện phát sinh	Ứng với nút lệnh có CommandName	Ý nghĩa sử dụng các sự kiện
ItemCommand	<b>Edit, New, Cancel</b> Và những nút có tên tùy chọn	Sửa, Thêm, Không
ModeChanging		Chuyển sang trạng thái khác (Edit, Insert, Read-Only)
ItemUpdating	<b>Cancel</b>	Cập nhật vào CSDL
ItemInserting	<b>Delete</b>	Thêm mẫu tin mới vào CSDL
ItemDeleting	<b>Select</b>	Hủy mẫu tin khỏi CSDL



LTV Công Nghệ .Net (2014) – Module 3 - Bài 6. Xây dựng trang web có liên kết với CSDL (2)

6

## 2. Các thao tác trên DetailsView

### ❑ Tạo dạng Master-Detail

ID	Tên sản phẩm	Thông tin chi tiết của sản phẩm
1	<a href="#">Áo cách điệu HOT...HOT</a>	ID 4
2	<a href="#">Áo thun KOREA mới về...Kute</a>	Tên sản phẩm Áo thun Nữ xi tin cực xinh, xanh
3	<a href="#">Khoe thun nón cực đẹp!!!</a>	Mô tả Chất Liệu Thành Phẩm : 100%Cotton
4	<a href="#">Áo thun Nữ xi tin cực xinh, xanh</a>	Đơn giá 220,000
5	<a href="#">Thun sọc cá tính!!!!</a>	Giá khuyến mãi 0
6	<a href="#">Áo ba lỗ dễ thương lắm đây</a>	Hình 18G03478.jpg
7	<a href="#">Áo nơ xinh form dài</a>	Nhóm sản phẩm ID 6
8	<a href="#">Áo kiểu set đôi</a>	Ngày cập nhật 16/03/2011
9	<a href="#">Quần jeans Nam thời trang Hàn</a>	Ngừng bán <input type="checkbox"/>
10	<a href="#">Áo kiểu công sở Ánes i</a>	Bí danh Ao-thun-Nu-xi-tin-cuc-xinh-xanh

Master: dùng GridView

Detail: dùng DetailsView



LTV Công Nghệ .Net (2014) – Module 3 - Bài 6. Xây dựng trang web có liên kết với CSDL (2)

7

## 2. Các thao tác trên DetailsView

### ❑ Tạo dạng Master-Detail

- Bước 1: tạo EntityDataSource1 dùng làm nguồn cho GridView
- Bước 2: tạo GridView dùng làm Master (đặt ID=**gvSanpham**) và kết nguồn vào  
(Nên tạo một nút lệnh xử lý chọn dòng)
- Bước 3: tạo EntityDataSource2 dùng làm nguồn cho DetailsView.
  - Khai báo thuộc tính **Where**
- Bước 4: tạo DetailsView dùng làm Detail và kết nguồn vào



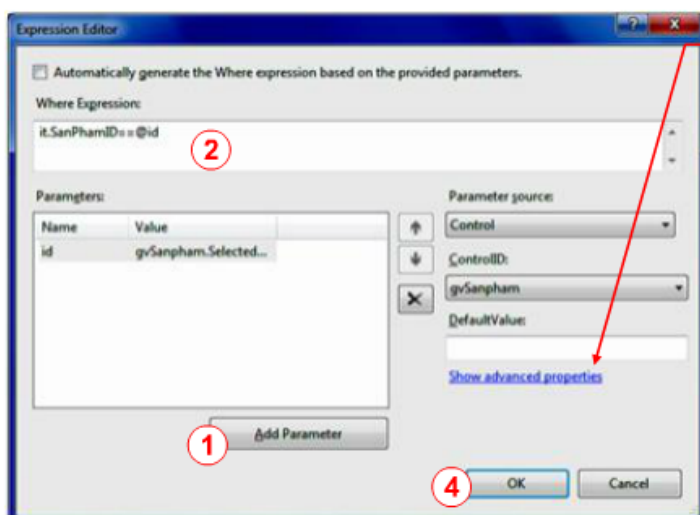
LTV Công Nghệ .Net (2014) – Module 3 - Bài 6. Xây dựng trang web có liên kết với CSDL (2)

8

## 2. Các thao tác trên DetailsView

### ❑ Tạo dạng Master-Detail

- Cửa sổ khai báo thuộc tính **Where**



3. Click vào để khai báo kiểu dữ liệu cho tham số



LTV Công Nghệ .Net (2014) – Module 3 - Bài 6. Xây dựng trang web có liên kết với CSDL (2)

9



## Nội dung

1. Giới thiệu điều khiển DetailsView
2. Các thao tác trên DetailsView
3. Giới thiệu điều khiển ListView
4. Các thao tác trên ListView



## 3. Giới thiệu điều khiển ListView

- ☐ Thể hiện dữ liệu theo các dạng: lưới (Grid), nhóm (titled), danh sách (BulletedList, flow), 1 dòng thể hiện trong 1 cột (Single Row - dạng này ít dùng)
- ☐ Nổi bật là cho phép thể hiện dữ liệu theo nhóm
- ☐ Đặc điểm:
  - Hỗ trợ cơ chế sắp xếp dữ liệu
  - Hỗ trợ cơ chế phân trang dữ liệu
  - Cho phép cập nhật, thêm và xóa dữ liệu



## Nội dung

1. Giới thiệu điều khiển DetailsView
2. Các thao tác trên DetailsView
3. Giới thiệu điều khiển ListView
4. Các thao tác trên ListView
  1. Liên kết dữ liệu, chọn Layout và định dạng
  2. Phân trang
  3. Sắp xếp
  4. Cập nhật dữ liệu

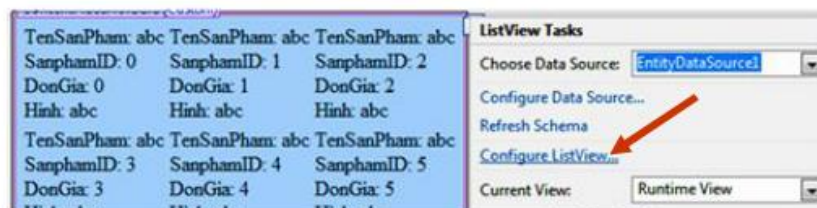


LTV Công Nghệ .Net (2014) – Module 3 - Bài 6. Xây dựng trang web có liên kết với CSDL (2)

12

## 4. Các thao tác trên ListView

- Liên kết dữ liệu, chọn Layout và định dạng
  - Liên kết dữ liệu
    - Giống như các điều khiển Data-Bound khác
  - Chọn Layout



- Tùy theo Layout được chọn, Asp.Net sẽ phát sinh ra mã HTML tương ứng:
  - Dạng Grid, Tiled, Single Row: phát sinh ra tag <Table>
  - Dạng Bulleted List: phát sinh ra tag <ul>, <li>
  - Dạng Flow: phát sinh tag <span>



LTV Công Nghệ .Net (2014) – Module 3 - Bài 6. Xây dựng trang web có liên kết với CSDL (2)

13



## 4. Các thao tác trên ListView

### □ Liên kết dữ liệu, chọn Layout và định dạng

- Hoặc có thể mở cửa sổ source viết code, sử dụng các thành phần sau:
  - **LayoutTemplate**: Định nghĩa bố cục cho điều khiển, có `<tr ID="itemPlaceholder" runat="server"> </tr>` là nơi chứa các nội dung từ [ItemTemplate](#) hoặc [GroupTemplate](#)
  - **ItemTemplate**: Chứa dữ liệu liên kết của dòng có chỉ số lẻ
  - **AlternatingItemTemplate**: Chứa dữ liệu liên kết của dòng chẵn
  - **EmptyDataTemplate**: Nội dung khi không có dữ liệu binding
  - **InsertItemTemplate**: Chứa các điều khiển hiển thị khi thêm mới
  - **EditItemTemplate**: Chứa các điều khiển hiển thị khi sửa
  - **SelectedItemTemplate**: Chứa dữ liệu liên kết khi dòng được chọn, có thể định dạng riêng để phân biệt với các dòng khác



LTV Công Nghệ .Net (2014) – Module 3 - Bài 6. Xây dựng trang web có liên kết với CSDL (2)

14

## 4. Các thao tác trên ListView

### □ Liên kết dữ liệu, chọn Layout và định dạng

- Hoặc có thể mở cửa sổ source viết code, sử dụng các thành phần sau:

```
<asp:ListView ID="ListView1" runat="server" DataSourceID="EntityDataSource1">
  <AlternatingItemTemplate>...</AlternatingItemTemplate>
  <EditItemTemplate>...</EditItemTemplate>
  <EmptyDataTemplate>...</EmptyDataTemplate>
  <InsertItemTemplate>...</InsertItemTemplate>
  <ItemTemplate>...</ItemTemplate>
  <LayoutTemplate>...</LayoutTemplate>
  <SelectedItemTemplate>...</SelectedItemTemplate>
</asp:ListView>
```



LTV Công Nghệ .Net (2014) – Module 3 - Bài 6. Xây dựng trang web có liên kết với CSDL (2)

15

## 4. Các thao tác trên ListView

### □ Liên kết dữ liệu, chọn Layout và định dạng

```

<asp:ListView ID="ListView1" runat="server" DataSourceID="EntityDataSource1">
  <ItemTemplate>
    <tr style="">
      <td><asp:Label ID="PhanLoaiIDLabel" runat="server" Text="<%= Eval("PhanLoaiID") %>" /></td>
      <td><asp:Label ID="TenPhanLoaiLabel" runat="server" Text="<%= Eval("TenPhanLoai") %>" /></td>
    </tr>
  </ItemTemplate>
  <LayoutTemplate>
    <table runat="server">
      <tr runat="server">
        <td runat="server">
          <table id="itemPlaceholderContainer" runat="server" border="0" style="">
            <tr runat="server" style="">
              <th runat="server">Phân loại ID</th>
              <th runat="server">Tên phân loại</th>
            </tr>
            <tr id="itemPlaceholder" runat="server">
              <td><div></div></td>
              <td><div></div></td>
            </tr>
          </table>
        </td>
      </tr>
      <tr runat="server">
        <td runat="server" style=""></td>
      </tr>
    </table>
  </LayoutTemplate>
</asp:ListView>

```

Bổ sung các style vào đây

Phân loại ID	Tên phân loại
1	Quần áo nữ
2	Quần áo nam
3	Phụ kiện thời trang



LTV Công Nghệ .Net (2014) – Module 3 - Bài 6. Xây dựng trang web có liên kết với CSDL (2)

16

## 4. Các thao tác trên ListView

### □ Liên kết dữ liệu, chọn Layout và định dạng

- Nếu trình bày dạng Tiled, thì bổ sung thêm item: **<GroupTemplate>**
  - Và khai báo thuộc tính:
 
$$\text{GroupItemCount} = \text{<Số cột trong 1 nhóm>}$$
- Định dạng dữ liệu
  - Mở cửa sổ Source hiệu chỉnh trực tiếp:
    - Sử dụng CSS: định dạng giao diện
    - Sử dụng các tag HTML để tạo Layout tùy biến
    - Sử dụng cú pháp Binding để liên kết dữ liệu

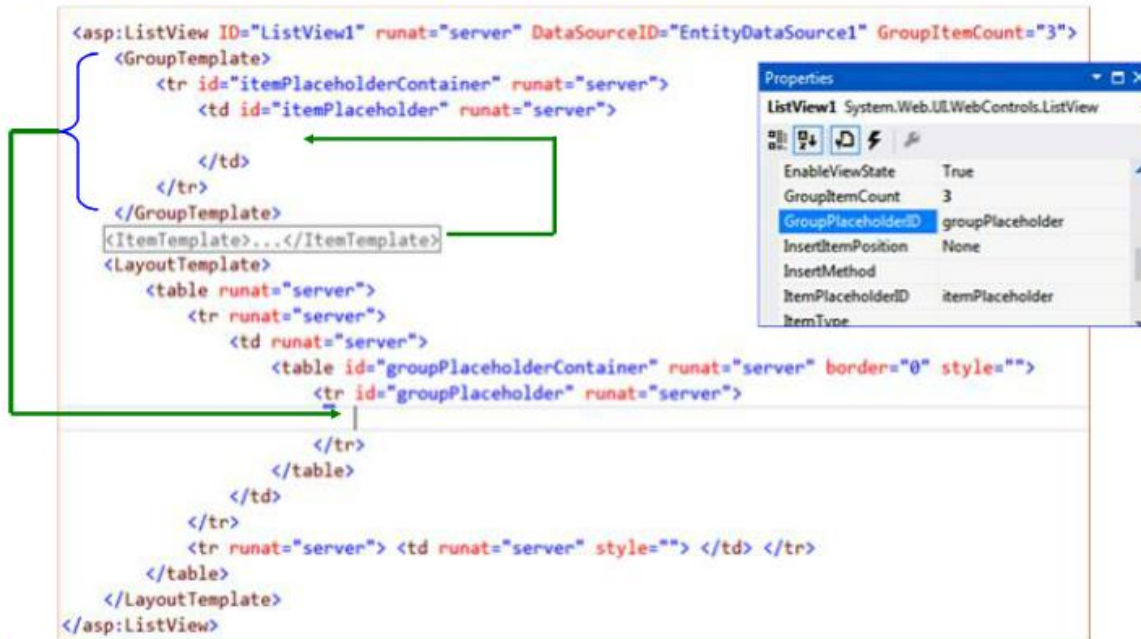


LTV Công Nghệ .Net (2014) – Module 3 - Bài 6. Xây dựng trang web có liên kết với CSDL (2)

17

## 4. Các thao tác trên ListView

### ❑ Liên kết dữ liệu, chọn Layout và định dạng



The screenshot shows the ASP.NET code for a `ListView` control and its Properties window. The code defines a `GroupTemplate` and an `ItemTemplate` using a `table` layout. The `GroupTemplate` contains a `tr` with an `td` that holds the `ItemTemplate`. The `ItemTemplate` is a `table` with a single `tr` and `td` containing the `groupPlaceholder`. The Properties window shows the `ListView1` control with the following properties:

- `EnableViewState`: True
- `GroupItemCount`: 3
- `GroupPlaceholderID`: groupPlaceholder
- `InsertItemPosition`: None
- `InsertMethod`: None
- `ItemPlaceholderID`: itemPlaceholder
- `ItemType`: None

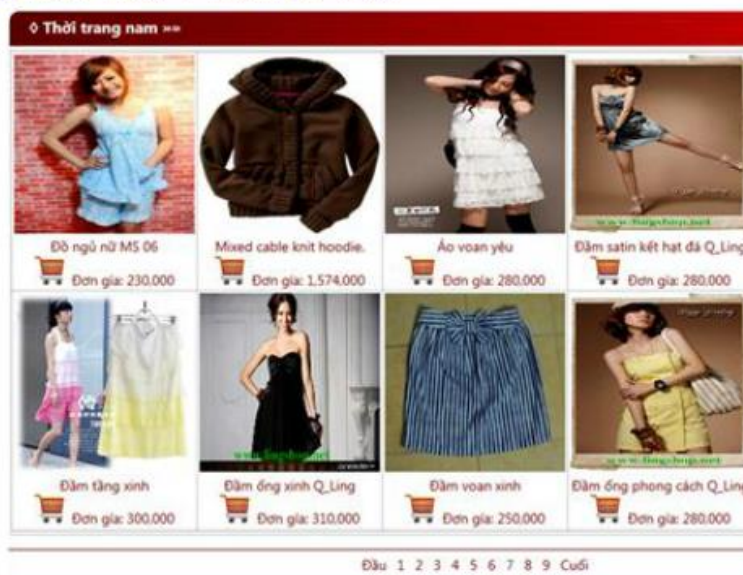
LTV Công Nghệ .Net (2014) – Module 3 - Bài 6. Xây dựng trang web có liên kết với CSDL (2)

18

## 4. Các thao tác trên ListView

### ❑ Liên kết dữ liệu, chọn Layout và định dạng

- Ví dụ: mẫu Tiled, 1 nhóm có 4 cột



LTV Công Nghệ .Net (2014) – Module 3 - Bài 6. Xây dựng trang web có liên kết với CSDL (2)

19



## 4. Các thao tác trên ListView

### □ Sắp xếp

- Bổ sung nút lệnh(Button, LinkButton, ImageButton) vào **LayoutTemplate** và khai báo thuộc tính :
  - `CommandName = "Sort"`
  - `CommandArgument = "Tên field muốn sắp xếp"`
  - Ví dụ: tạo các nút sắp xếp trên dòng tiêu đề, sử dụng LinkButton thay thế cho Label

```
<th runat="server">
    <asp:LinkButton ID="InkID" runat="server" CommandName="Sort"
    CommandArgument="PhanLoaiID">Phân loại ID</asp:LinkButton>
</th>
<th runat="server">
    <asp:LinkButton ID="InkTen" runat="server" CommandName="Sort"
    CommandArgument="TenPhanLoai">Tên Phân loại</asp:LinkButton>
</th>
```



## 4. Các thao tác trên ListView

### □ Phân trang

- Sử dụng điều khiển **DataPager**, khai báo các thuộc tính:
  - `PagedControlID` : khai báo ID của điều khiển được phân trang  
Lưu ý:Nếu DataPager đặt trong ListView thì không cần khai báo thuộc tính này
  - `PageSize` : số dòng trên 1 trang
  - `Fields` : Khai báo các dạng nút phân trang

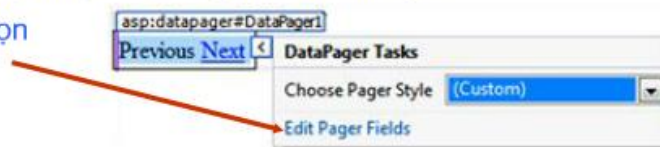


## 4. Các thao tác trên ListView

### □ Phân trang

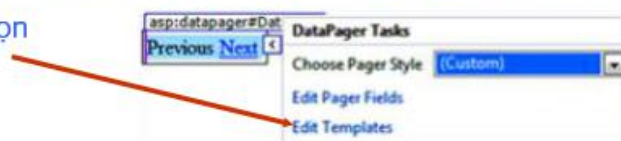
- Bổ sung template riêng vào DataPager

- Chọn

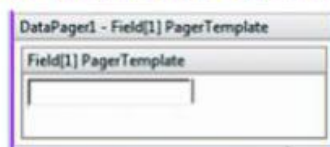


- Chọn loại Template Pager Field → click vào nút OK

- Chọn



- Bổ sung điều khiển vào thành phần PagerTemplate



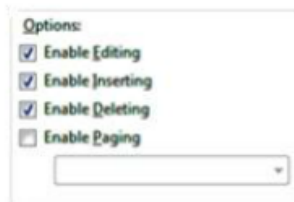
LTV Công Nghệ .Net (2014) – Module 3 - Bài 6. Xây dựng trang web có liên kết với CSDL (2)

22

## 4. Các thao tác trên ListView

### □ Cập nhật dữ liệu

- Khi cấu hình, bật các chức năng sau:



→ Tạo ra các nút Thêm, Sửa, Hủy

- Các sự kiện và code xử lý tương tự như GridView và DetailsView

### □ Nếu bổ sung nút **Select** thì sẽ viết code ở cửa sổ Source với CommandName ="Select"



LTV Công Nghệ .Net (2014) – Module 3 - Bài 6. Xây dựng trang web có liên kết với CSDL (2)

23

## Thảo luận



LTV Công Nghệ .Net (2014) – Module 3 - Bài 6. Xây dựng trang web có liên kết với CSDL (2)

24



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

## LTV CÔNG NGHỆ .NET

### Module 3 - Bài 7. *Quản lý Web Application*

Ngành Lập trình & CSDL

<http://www.t3h.vn>

2014



2014





## Nội dung

### 1. Tập tin quản lý và cấu hình ứng dụng web

1. Tập tin Global.asax
2. Tập tin Web.config

### 2. Các đối tượng quản lý ứng dụng web



LTV Công Nghệ .Net (2014) – Module 3 - Bài 7. Quản lý Web Application

2



## 1. Tập tin quản lý và cấu hình ứng dụng Web

### □ Tập tin Global.asax

- Giới thiệu
  - Là một tập tin tùy chọn có chứa code để quản lý các sự kiện ở cấp Application và Session
    - Khai báo và khởi tạo cho biến Application, Session trong các sự kiện này
  - Được đặt trong thư mục gốc của ứng dụng ASP.NET
  - Một project chỉ có một tập tin Global.asax



LTV Công Nghệ .Net (2014) – Module 3 - Bài 7. Quản lý Web Application

3

# 1. Tập tin quản lý và cấu hình ứng dụng Web

## ❑ Tập tin Global.asax

- Thao tác tạo: Từ project, chọn **Add** → **Add new Item...**



LTV Công Nghệ .Net (2014) – Module 3 - Bài 7. Quản lý Web Application

4

# 1. Tập tin quản lý và cấu hình ứng dụng Web

## ❑ Tập tin Global.asax

- Các sự kiện trong tập tin

```
<%@ Application Language="C#" %>
<script runat="server">
    void Application_Start(object sender, EventArgs e) {
        //code được xử lý khi ứng dụng khởi động
    }
    void Application_End(object sender, EventArgs e) {
        // code được xử lý khi đóng ứng dụng
    }
    void Application_Error(object sender, EventArgs e) {
        // code được xử lý khi ứng dụng bị lỗi
    }
    void Session_Start(object sender, EventArgs e) {
        // code được xử lý khi có một Session mới
    }
    void Session_End(object sender, EventArgs e) {
        // code được xử lý khi kết thúc một Session
    }
</script>
```



LTV Công Nghệ .Net (2014) – Module 3 - Bài 7. Quản lý Web Application

5

# 1. Tập tin quản lý và cấu hình ứng dụng Web

## ❑ Tập tin Web.config

- Dùng để khai báo các thông số mặc định cho ứng dụng, có dạng XML
- Chứa các thông tin cấu hình riêng của ứng dụng Web
- Trong mỗi ứng dụng và các thư mục con của ứng dụng Web có thể có tập tin Web.config riêng
  - Trong một project được phép có nhiều tập tin Web.config
- Cấu trúc tập tin ban đầu khi mới tạo website

```
<?xml version="1.0"?>
<!--
  For more information on how to configure your
  ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.5" />
    <httpRuntime targetFramework="4.5" />
  </system.web>
</configuration>
```



LTV Công Nghệ .Net (2014) – Module 3 - Bài 7. Quản lý Web Application

6

# 1. Tập tin quản lý và cấu hình ứng dụng Web

## ❑ Tập tin Web.config

- Làm việc với tập tin
  - Bổ sung thông tin tùy chọn riêng cho ứng dụng

```
<appSettings>
  <add key="Tên khóa" value="Giá trị" />
</appSettings>
```

- Truy xuất thông tin
  - Sử dụng Namespace: System.Configuration

```
<Biến>=ConfigurationManager.AppSettings["Tên khóa"];
```



LTV Công Nghệ .Net (2014) – Module 3 - Bài 7. Quản lý Web Application

7



## Nội dung

1. Tập tin quản lý và cấu hình ứng dụng web
2. Các đối tượng quản lý ứng dụng web
  1. Đối tượng Request
  2. Đối tượng Response
  3. Đối tượng Session
  4. Đối tượng Application
  5. Đối tượng Server
  6. Đối tượng Cookies



## 2. Các đối tượng quản lý ứng dụng Web

### □ Đối tượng Request

- Dùng để nhận giá trị từ Client gửi về cho Web server
- Thuộc tính **QueryString**: cho phép nhận các giá trị truyền qua chuỗi tham số
- Cú pháp:

```
Biến = Request.QueryString["Tên_tham_số"];
```

**URL** → **http: // <host> [: <port>] [ <path> [? <QueryString> ] ]**

- Ví dụ:

```
http://t3h.vn/Pages/MBookView.aspx?ID=2&Ch=3
```

→ Khi truy xuất

```
string id, muc;
```

```
id = Server.HtmlEncode(Request.QueryString["ID"]); // 2
```

```
muc = Request.QueryString["Ch"]; // 3
```





## 2. Các đối tượng quản lý ứng dụng Web

### ❑ Đối tượng Response

- Được sử dụng để giao tiếp với Client (gửi kết quả đến Client)
- Quản lý và điều phối thông tin từ Web Server đến trình duyệt của người dùng
- Phương thức:
  - **Write**: in một chuỗi ra trang web, không thể qui định vị trí hiển thị của chuỗi
  - **Redirect**: gửi thông điệp yêu cầu Web Browser truy cập đến một địa chỉ khác

### • Ví dụ:

```
Response.Write("Chào các bạn!");
Response.Redirect("~/Pages/TrangChu.aspx") ;
```



## 2. Các đối tượng quản lý ứng dụng Web

### ❑ Đối tượng Server

- Được sử dụng để cung cấp thông tin của Web Server cho ứng dụng
- Phương thức:
  - **Transfer("URL")**: ngừng thi hành trang hiện hành, gửi yêu cầu mới đến trang khác
  - **MapPath()**: Trả về đường dẫn vật lý tương ứng với đường dẫn ảo trên Web Server
    - Đường dẫn đến thư mục của trang hiện hành  
`Server.MapPath("");`
    - Đường dẫn đến thư mục root của IIS: C:\inetpub\wwwroot\  
`Server.MapPath("~/");`





## 2. Các đối tượng quản lý ứng dụng Web

### ❑ Đối tượng Application

- Đối tượng toàn cục, quản lý toàn bộ ứng dụng Web
- Thông tin được lưu trữ trong đối tượng Application được “hiểu” ở tất cả trang aspx trong suốt thời gian “sống” của ứng dụng
- Chỉ bị đóng/hủy khi tắt Web Server
- Tạo biến Application:

```
Application.Lock();
Application["tên_biến"] = giá_trị;
Application.Unlock();
```

- Lấy giá trị:

```
<Biến> = Application["<tên biến>"];
```

- Lưu ý: <tên biến> phân biệt chữ HOA/thường



## 2. Các đối tượng quản lý ứng dụng Web

### ❑ Đối tượng Session

- Lưu trữ thông tin trong một phiên làm việc cụ thể
- Được tạo ra khi người dùng kết nối đến Web Server lần đầu tiên
- Thuộc tính Timeout(phút): thời gian duy trì Session, nếu người dùng không tương tác với Web Server (default=20')
- Phương thức Abandon: hủy Session và giải phóng tài nguyên cho Web Server
- Tạo biến Session: **Session["Tên\_biến"] = Giá\_trị;**
- Lấy giá trị từ biến Session: **Biến = Session["Tên\_biến"];**
  - Lưu ý:
    - Tên\_biến phân biệt chữ HOA/thường
    - Biến được lưu trữ trong Session không bị loại bỏ khi chuyển đổi giữa các trang





## 2. Các đối tượng quản lý ứng dụng Web

### ❑ Đối tượng Cookies

- Dùng để lưu trữ thông tin của người dùng tại máy Client
- Thêm Cookies:

**Response.Cookies.Add(đối\_tượng\_HttpCookie) ;**

- Lấy giá trị từ Cookies:

**Request.Cookies["Tên Cookie"].Value;**



## 2. Các đối tượng quản lý ứng dụng Web

### ❑ Đối tượng Cookies

- Ví dụ:

*// Khai báo Cookies*

```
HttpCookie cookTen = new HttpCookie("Ten_dn");
cookTen.Value = "Pham-Thien-Thanh";
cookTen.Expires = DateTime.Today.AddDays(3); // hết hạn
Response.Cookies.Add(cookTen);           // thêm vào
client
```

*// Truy xuất*

```
HttpCookie cookTen = Request.Cookies["Ten_dn"];
if (cookTen != null)
```

**Xuat\_Cook.Text = cookTen.Value;**



## Thảo luận



LTV Công Nghệ .Net (2014) – Module 3 - Bài 7. Quản lý Web Application

16



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

## LTV CÔNG NGHỆ .NET

Module 3 - Bài 8. *Hoàn chỉnh ứng dụng Web*

Ngành Lập trình & CSDL

<http://www.t3h.vn>

2014



2014





## Nội dung

1. Cài đặt jQuery Plugin
  1. Tổng quan về jQuery
  2. Cài đặt jQuery Plugin
2. Cài đặt Plugin của Facebook



## 1. Cài đặt jQuery Plugin

- Tổng quan về jQuery
  - Thông điệp của jQuery





# 1. Cài đặt jQuery Plugin

## □ Tổng quan về jQuery

- jQuery là gì?
  - jQuery là một thư viện JavaScript nhỏ gọn và có nhiều tính năng phong phú, xử lý nhanh
  - Làm việc với hầu hết các chuẩn mới (như HTML5, CSS3)
  - Với một kho API dễ dàng sử dụng, nó làm cho mọi thứ trở nên đơn giản hơn như:
    - Duyệt và thao tác trên DOM
    - Xử lý sự kiện, các thao tác về animation(hình ảnh động) và Ajax
  - Các API này làm việc được trên nhiều trình duyệt (cross-browser).
  - Với sự kết hợp của tính linh hoạt và khả năng mở rộng, jQuery đã làm thay đổi cách viết JavaScript của hàng triệu người



# 1. Cài đặt jQuery Plugin

## □ Tổng quan về jQuery

- jQuery Plugin là gì?
  - Là một phương pháp mới được sử dụng để mở rộng đối tượng nguyên mẫu của jQuery
    - Tạo ra các bài toán thường dùng, như: hiệu ứng giao diện, hiệu ứng menu, animation, ....
  - Hỗ trợ cho người lập trình lẫn người thiết kế web tạo ra các giao diện chuyên nghiệp mà không tốn nhiều thời gian
  - Có hàng ngàn jQuery Plugin trên Internet

## □ Cài đặt jQuery Plugin

- Giáo viên hướng dẫn cụ thể các jQuery có thể cài đặt trong hệ thống bài tập



## Nội dung

1. Cài đặt Plugin jQuery
2. Cài đặt Plugin của Facebook



## 1. Cài đặt Plugin của Facebook

- ❑ Một vài plugin của Facebook đòi hỏi bạn phải có **App ID**, do đó ta nên đăng ký tạo ứng dụng trên Facebook để nhận App ID, giúp dễ dàng cài đặt các chức năng của FB
- ❑ Cài đặt:
  - Tạo nút **mở Fanpage**
  - Tạo chức năng **Comment**

(Giáo viên hướng dẫn trực tiếp)



## Thảo luận



LTV Công Nghệ .Net (2014) – Module 3 - Bài 8. Hoàn chỉnh ứng dụng Web

8



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
TRUNG TÂM TIN HỌC

## LTV CÔNG NGHỆ .NET

### Module 3 - Bài 9. *Web Services*

Ngành Lập trình & CSDL

<http://www.t3h.vn>

2014



2014





## Nội dung

### 1. Tìm hiểu về Web Services (WS)

1. Web Services là gì?
2. Web Services hoạt động như thế nào?

### 2. Xây dựng Web Services

### 3. Sử dụng Web Services



LTV Công Nghệ .Net (2014) – Module 3 - Bài 9. Web Services

2



## 1. Tìm hiểu về Web Services

### □ Web Services là gì?

- Là một hệ thống phần mềm được thiết kế dùng để hỗ trợ sự tương tác giữa các máy tính trên mạng
  - Các phần mềm được viết bởi những ngôn ngữ lập trình khác nhau và chạy trên những nền tảng khác nhau có thể sử dụng dịch vụ Web để chuyển đổi dữ liệu thông qua mạng Internet theo cách giao tiếp tương tự bên trong một máy tính
- Có khả năng chạy trên những hệ điều hành và framework khác nhau
- Mục tiêu của Web services (WS) là cung cấp một số chức năng của nó (có thể là của một cá nhân hay một tổ chức)
- Với sự phát triển và lớn mạnh của Internet, dịch vụ Web thật sự là một công nghệ đáng được quan tâm để giảm chi phí và độ phức tạp trong tích hợp và phát triển hệ thống.



LTV Công Nghệ .Net (2014) – Module 3 - Bài 9. Web Services

3



## 1. Tìm hiểu về Web Services

### ❑ Web Services hoạt động như thế nào?

- WS có 3 thành phần cơ bản: SOAP, WSDL và UDDI
  - **SOAP (Simple Object Access Protocol)**: là một giao thức truyền thông dùng để liên lạc và gửi thông điệp giữa các ứng dụng trên Internet
    - Có nền tảng dựa trên XML
    - Có khả năng xuyên qua bức tường lửa (Firewall).
  - **WSDL (Web Services Description Language)**: là ngôn ngữ dựa trên ngôn ngữ XML, dùng để mô tả WS và cách thức truy xuất nó
  - **UDDI (Universal Description, Discovery and Integration)**: là một dịch vụ danh bạ, nơi mà cá nhân hoặc tổ chức có thể đăng ký và tìm kiếm WS



## 1. Tìm hiểu về Web Services

### ❑ Web Services hoạt động như thế nào?

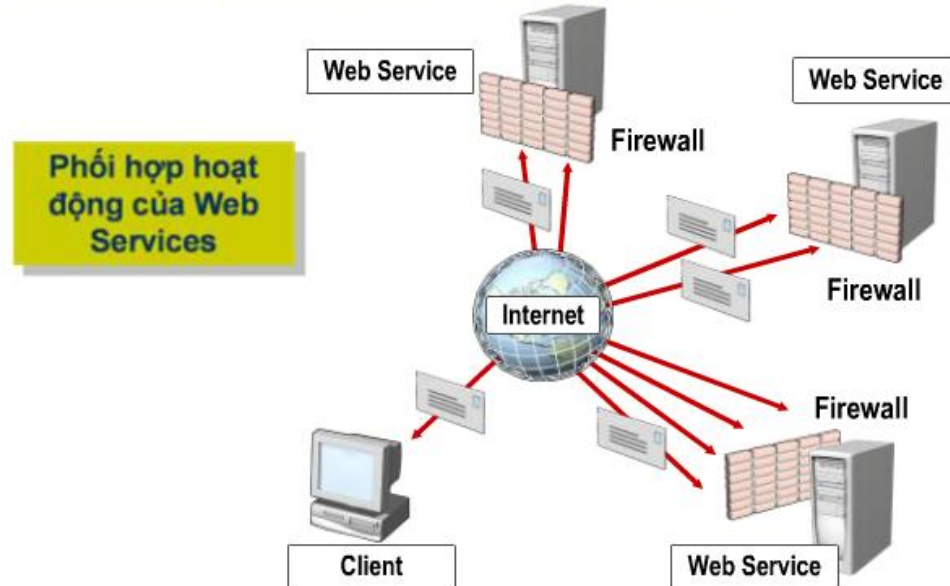
- WS có 3 thành phần cơ bản: SOAP, WSDL và UDDI



# 1. Tìm hiểu về Web Services

## ❑ Web Services hoạt động như thế nào?

- WS có 3 thành phần cơ bản: SOAP, WSDL và UDDI



LTV Công Nghệ .Net (2014) – Module 3 - Bài 9. Web Services

6

## Nội dung

1. Tìm hiểu về Web Services (WS)
2. Xây dựng Web Services
  1. Tạo Web Services trong Visual Studio .Net
  2. Kiểm tra Web Services
3. Sử dụng Web Services

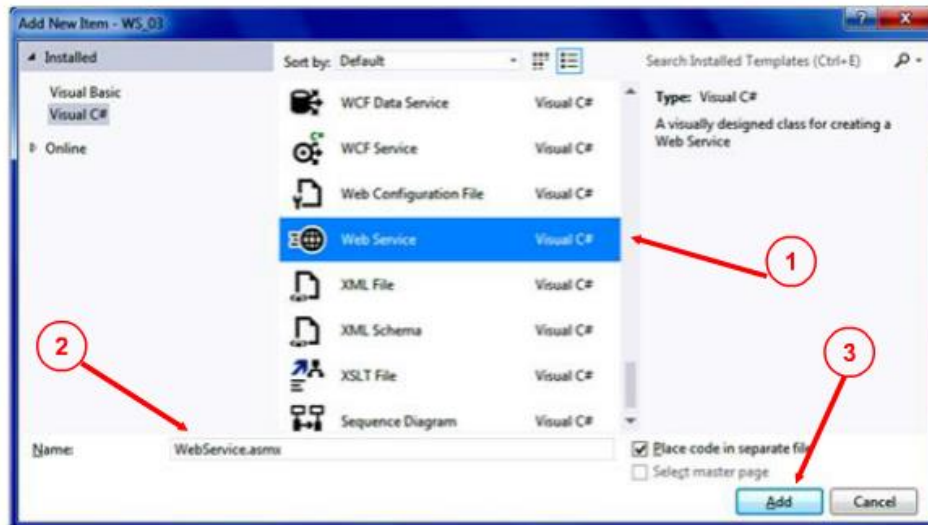
LTV Công Nghệ .Net (2014) – Module 3 - Bài 9. Web Services

7

## 2. Xây dựng Web Services

### ❑ Tạo Web Services trong Visual Studio .Net

- Nhấn phím phải chuột vào thư mục cần chứa WS, chọn **Add → Add New Item...**



LTV Công Nghệ .Net (2014) – Module 3 - Bài 9. Web Services

8

## 2. Xây dựng Web Services

### ❑ Tạo Web Services trong Visual Studio .Net

- Tập tin lưu trữ có phần mở rộng là **asmx**
- Tập tin Code behind **.cs** được đặt trong thư mục **App\_code**
- Các phương thức tạo ra luôn có thuộc tính **[WebMethod]** đứng trước nó

LTV Công Nghệ .Net (2014) – Module 3 - Bài 9. Web Services

9

## 2. Xây dựng Web Services

### ❑ Tạo Web Services trong Visual Studio .Net

Xuất hiện cửa sổ viết code

Phương thức cung cấp sẵn

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.Services;
6 /// <summary>
7 /// Summary description for WS_Sanpham
8 /// </summary>
9 [WebService(Namespace = "http://tempuri.org/")]
10 [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
11 // To allow this Web Service to be called from script, using ASP.NET
12 // [System.Web.Script.Services.ScriptService]
13 public class WS_Sanpham : System.Web.Services.WebService {
14     public WS_Sanpham () {
15         // Uncomment the following line if using designed components
16         // InitializeComponent();
17     }
18     [WebMethod]
19     public string HelloWorld() {
20         return "Hello World";
21     }
22 }

```



## 2. Xây dựng Web Services

### ❑ Tạo Web Services trong Visual Studio .Net

- Bổ sung thêm phương thức DocSanPham

```

//Đọc sản phẩm có ID truyền vào, trả về chuỗi Tên sản phẩm
[WebMethod]
public string DocSanPham(int pID)
{
    string ten = "";
    using(FASHIONEEntities db = new FASHIONEEntities())
    {
        SanPham entity= db.SanPhams.FirstOrDefault(x => x.SanphamID == pID);
        if (entity!=null)
            ten = entity.TenSanPham;
    }
    return ten;
}

```

### ❑ Kiểm tra Web Services

- Chọn đối tượng <tên WS>.asmx, nhấn F5
- Kết quả thể hiện có dạng XML





## Nội dung

1. Tìm hiểu về Web Services (WS)
2. Xây dựng Web Services
3. Sử dụng Web Services
  1. Sử dụng WS do người dùng xây dựng
  2. Sử dụng WS do các website cụ thể cung cấp miễn phí



## 3. Sử dụng Web Services

- Sử dụng Web Services do người dùng xây dựng
  - Muốn sử dụng Web services (WS), Client cần phải biết WS đó hỗ trợ những phương thức nào, phương thức có bao nhiêu tham số, kết quả trả về, ...
  - Các thông tin cần biết này được mô tả bởi tài liệu **WSDL**



### 3. Sử dụng Web Services

#### ❑ Sử dụng Web Services do người dùng xây dựng

- Ví dụ 1: truy xuất thông tin của phương thức DocSanPham

**WS\_Sanpham**

The following operations are supported. For a formal definition, please review the [Service Description](#).

- DocSanPham
- HelloWorld

This web service is using <http://tempuri.org/> as its base URI.

Recommendation: Change the default namespace to public.

Each XML Web service needs a unique namespace in order to be uniquely identified.

**Click**

```
<?xml version="1.0" encoding="UTF-8"?>
<string
  xmlns="http://tempuri.org/">Khoác
  thun nón cực đẹp!!!</string>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://tempuri.org/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://tempuri.org/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/">
  <wsdl:types>
    <wsdl:message name="HelloWorldSoapIn">
      <wsdl:sequence base="tns:HelloWorldSoapIn">
        <wsdl:element base="tns:HelloWorldSoapIn" name="HelloWorldSoapIn"/>
      </wsdl:sequence>
    </wsdl:message>
    <wsdl:message name="HelloWorldSoapOut">
      <wsdl:sequence base="tns:HelloWorldSoapOut">
        <wsdl:element base="tns:HelloWorldSoapOut" name="HelloWorldSoapOut"/>
      </wsdl:sequence>
    </wsdl:message>
    <wsdl:message name="DocSanPhamSoapIn">
      <wsdl:sequence base="tns:DocSanPhamSoapIn">
        <wsdl:element base="tns:DocSanPhamSoapIn" name="DocSanPhamSoapIn"/>
      </wsdl:sequence>
    </wsdl:message>
    <wsdl:message name="DocSanPhamSoapOut">
      <wsdl:sequence base="tns:DocSanPhamSoapOut">
        <wsdl:element base="tns:DocSanPhamSoapOut" name="DocSanPhamSoapOut"/>
      </wsdl:sequence>
    </wsdl:message>
    <wsdl:portType name="WS_SanphamSoap">
      <wsdl:operation name="HelloWorld">
        <wsdl:input message="tns:HelloWorldSoapIn"/>
        <wsdl:output message="tns:HelloWorldSoapOut"/>
      </wsdl:operation>
      <wsdl:operation name="DocSanPham">
        <wsdl:input message="tns:DocSanPhamSoapIn"/>
        <wsdl:output message="tns:DocSanPhamSoapOut"/>
      </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="WS_SanphamSoap" type="tns:WS_SanphamSoap">
      <wsdl:soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
      <wsdl:operation name="HelloWorld" binding="tns:WS_SanphamSoap" type="tns:HelloWorldSoapIn tns:HelloWorldSoapOut"/>
      <wsdl:operation name="DocSanPham" binding="tns:WS_SanphamSoap" type="tns:DocSanPhamSoapIn tns:DocSanPhamSoapOut"/>
    </wsdl:binding>
  </wsdl:definitions>
```

LTV Công Nghệ .Net (2014) – Module 3 - Bài 9. Web Services

14

### 3. Sử dụng Web Services

#### ❑ Sử dụng Web Services do người dùng xây dựng

- Ví dụ 2: Sử dụng Web Services trên 1 trang web

```
protected void btnDoc_Click(object sender, EventArgs e)
{
    WS_Sanpham ws = new WS_Sanpham();
    lblTen.Text = ws.DocSanPham(1);
}
```

LTV Công Nghệ .Net (2014) – Module 3 - Bài 9. Web Services

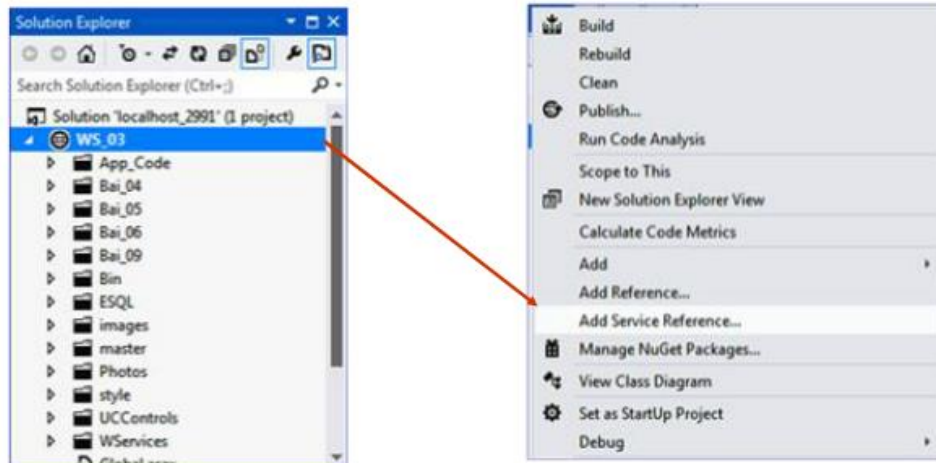
15

### 3. Sử dụng Web Services

- ❑ Sử dụng Web Services do các website cụ thể cung cấp miễn phí

- **Bước 1:** Bổ sung WS vào project

- Nhấn phím phải chuột lên project, chọn **Add Service Reference**



LTV Công Nghệ .Net (2014) – Module 3 - Bài 9. Web Services

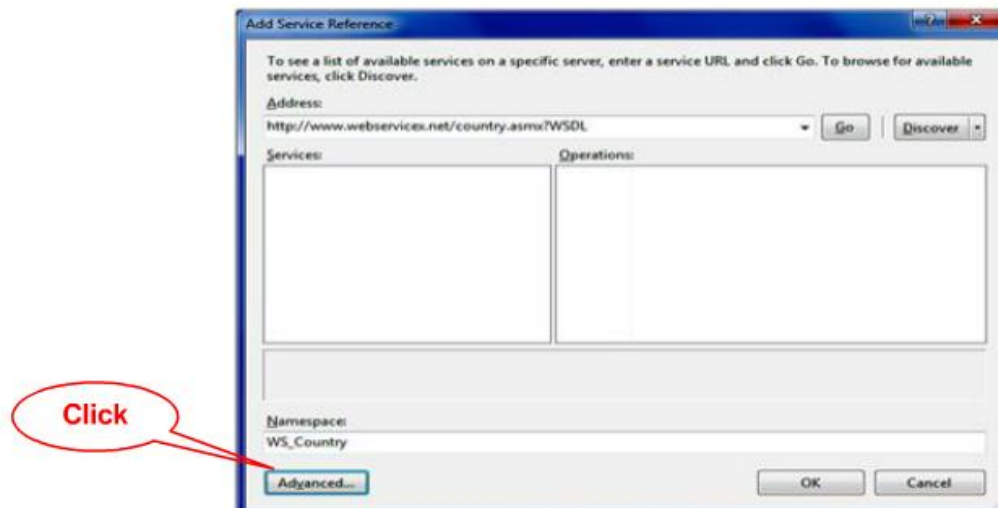
16

### 3. Sử dụng Web Services

- ❑ Sử dụng Web Services do các website cụ thể cung cấp miễn phí

- **Bước 1:** Bổ sung WS vào project

- Nhấn vào nút **Advanced**



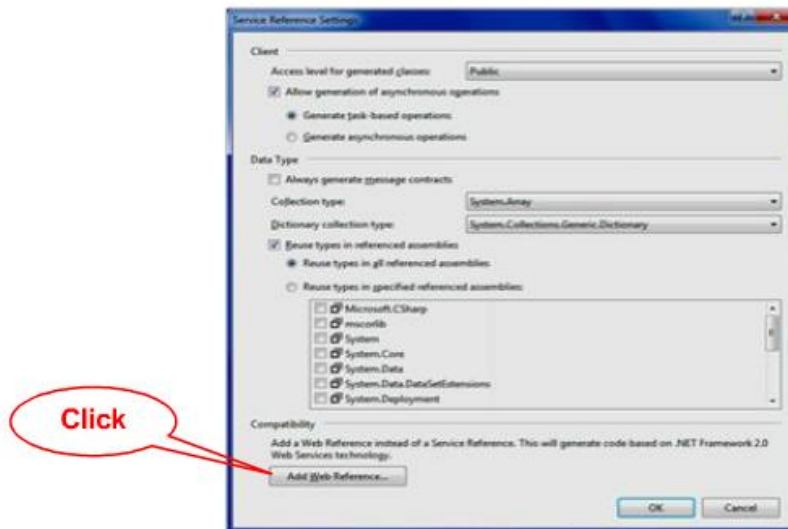
LTV Công Nghệ .Net (2014) – Module 3 - Bài 9. Web Services

17

### 3. Sử dụng Web Services

❑ Sử dụng Web Services do các website cụ thể cung cấp miễn phí

- **Bước 1:** Bổ sung WS vào project
  - Nhấn vào nút **Add Web References ...**



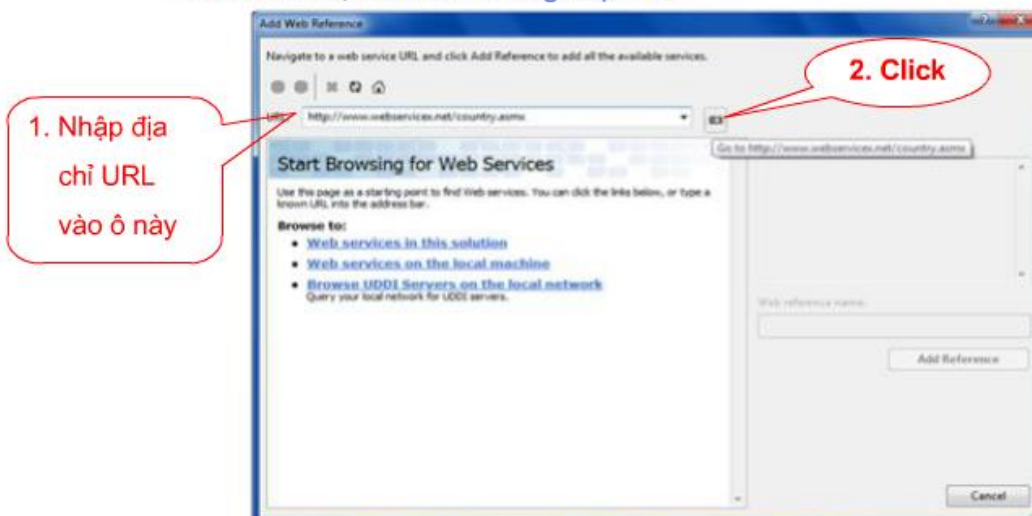
LTV Công Nghệ .Net (2014) – Module 3 - Bài 9. Web Services

18

### 3. Sử dụng Web Services

❑ Sử dụng Web Services do các website cụ thể cung cấp miễn phí

- **Bước 1:** Bổ sung WS vào project
  - Khai báo địa chỉ URL cung cấp WS



LTV Công Nghệ .Net (2014) – Module 3 - Bài 9. Web Services

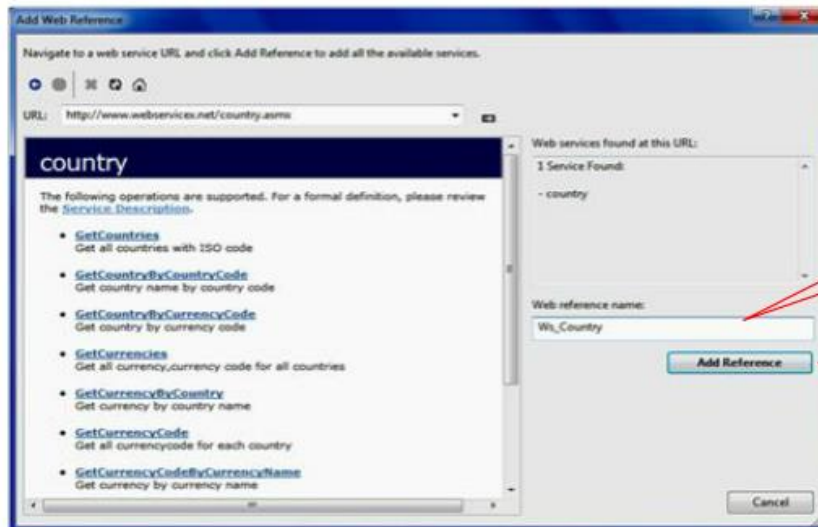
19

### 3. Sử dụng Web Services

- ❑ Sử dụng Web Services do các website cụ thể cung cấp miễn phí

- **Bước 1:** Bổ sung WS vào project

- Đặt tên Web services



LTV Công Nghệ .Net (2014) – Module 3 - Bài 9. Web Services

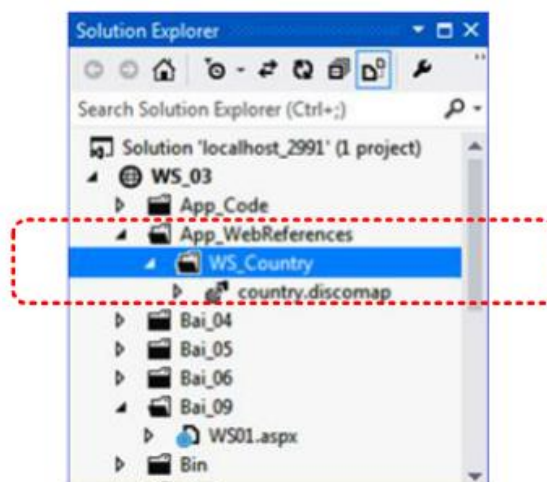
20

### 3. Sử dụng Web Services

- ❑ Sử dụng Web Services do các website cụ thể cung cấp miễn phí

- **Bước 1:** Bổ sung WS vào project

- Kết quả sau khi bổ sung WS vào project



LTV Công Nghệ .Net (2014) – Module 3 - Bài 9. Web Services

21



### 3. Sử dụng Web Services

#### ❑ Sử dụng Web Services do các website cụ thể cung cấp miễn phí

##### • Bước 2: Sử dụng Web Services

- Khai báo và khởi tạo các lớp mà WS đã cung cấp
- Sử dụng các phương thức của WS để truy xuất các thông tin cần thiết
- Ví dụ 1: truy xuất tên các nước

```
WS_Country.country ws = new WS_Country.country();
lblTen.Text = ws.GetCountries(); // kết quả trả về dưới dạng XML
//Xuất ra điều khiển TextBox sẽ thấy nội dung XML
```



### 3. Sử dụng Web Services

#### ❑ Sử dụng Web Services do các website cụ thể cung cấp miễn phí

##### • Bước 2: Sử dụng Web Services

- Ví dụ 2: truy xuất danh sách các thành phố lớn ở Việt Nam. Giả sử đã tạo WS tên WS\_Weather

```
//Đọc Web Services, lưu vào nộiDung
WS_Weather.GlobalWeather ws2 = new WS_Weather.GlobalWeather();
string noiDung = "<?xml version='1.0' encoding='utf-8' ?>";
noiDung += ws2.GetCitiesByCountry("Viet Nam");

//Ghi noiDung vào tài liệu XML
string tapTin = Server.MapPath("~/Bai_09/Country.xml");
File.WriteAllText(tapTin, noiDung);

//Đọc tài liệu XML vào DataSet
DataSet ds = new DataSet();
ds.ReadXml(tapTin);

// Xuất DataSet ra GridView
if (ds.Tables.Count > 0) {
    gvNuoc.DataSource = ds.Tables[0];
    gvNuoc.DataBind(); }
}
```

Country	City
Viet Nam	Da Nang
Viet Nam	Ha Noi
Viet Nam	Nha Trang
Viet Nam	Hue
Viet Nam	Pleiku City
Viet Nam	Quy Nhon
Viet Nam	Ho Chi Minh
Viet Nam	Vinh



## Thảo luận

