

THE RESURGENCE OF WEBFORMS

“I’M NOT DEAD YET!”

Philip Japikse (@skimedic)

skimedic@outlook.com

www.skimedic.com/blog

MVP, MCSD.Net, MCDBA, CSM, CSP

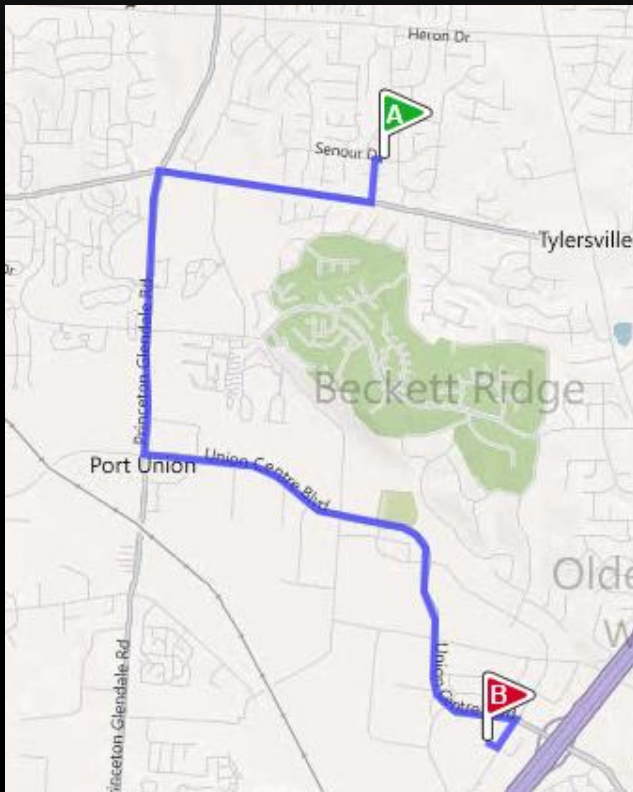


WHO AM I?

- Developer, Author, Teacher
- Microsoft MVP, ASPInsider, MCSD, MCDBA, CSM, CSP
- Founder, Agile Conferences, Inc.
- President, Cincinnati .NET User's Group
- Co-host, Hallway Conversations
 - www.hallwayconversations.com

ROUTING

ROUTING



- Directs Incoming URLs to correct Pages
- Creates Outgoing URLs
- SEO Optimized

SETTING UP ROUTING

```
var settings = new FriendlyUrlSettings();  
settings.AutoRedirectMode = RedirectMode.Permanent;  
routes.EnableFriendlyUrls(settings);  
  
routes.MapPageRoute("CustomerDefault", "CustomerList",  
"~/CustomerPages/Customers.aspx");  
  
routes.MapPageRoute("CustomerDetails", "CustomerDetails/{id}",  
"~/CustomerPages/CustomerDetails.aspx", true,  
new RouteValueDictionary { { "id", "0" } });
```

USING ROUTING

In Markup:

```
<a href="#">#: GetRouteUrl("CustomerDetails",new  
{id=Item.Id} ) %>">
```

In Code:

```
Response.RedirectToRoute("CustomerDefault");
```

DATA BINDING

MAGIC STRINGS == EVIL!



<http://www.fightersgeneration.com/np9/characters/darthvader-flames.jpg>

STRONGLY TYPED == GOOD



http://mobilelocalsocial.com/wp-content/uploads/2010/04/tumblr_kpkvqIiNIX71qz7I0ao1_500-e1272645432268.jpg

STRONGLY TYPED DATA CONTROLS

```
<asp:Repeater ID="CustomerRepeater" runat="server"
>
  <ItemType="DataAccess.Customer" >
    <itemtemplate>
      <li>
        Title:<%# Item.Title %>
        First Name:<%# Item.FirstName %>
        Last Name:<%# Item.LastName %>
        Phone Number:<%# Item.PhoneNumber %>
      </li>
    </ItemTemplate>
  </asp:Repeater>
```

LIST BINDING – SELECTING DATA

```
<asp:Repeater ID="CustomerRepeater" runat="server"
ItemType="DataAccess.Customer" SelectMethod="GetData">
  <ItemTemplate>
    <li>
      Title:<%# Item.Title %>
      First Name:<%# Item.FirstName %>
      Last Name:<%# Item.LastName %>
      Phone Number:<%# Item.PhoneNumber %>
    </li>
  </ItemTemplate>
</asp:Repeater>
```

```
public IEnumerable<DataAccess.Customer> GetData()
{ return CustomerRepository.GetAll(); }
```

DEMO

Binding

VALUE PROVIDERS - QUERY STRING

```
public IQueryable<Customer> GetCustomers(  
    [QueryString]string keyword)  
{  
    return CustomerRepository.GetByName(keyword);  
}
```

```
public IQueryable<Customer> GetCustomers(  
    [QueryString("name")]string keyword)  
{  
    return CustomerRepository.GetByName(keyword);  
}
```

VALUE PROVIDERS - CONTROLS

```
public IQueryable<Customer> GetCustomers(  
    [Control("customers")]int? id)  
{  
    if (id.HasValue)  
    {  
        return CustomerRepository.GetOne(id);  
    }  
    else  
    {  
        return CustomerRepository.GetAll();  
    }  
}
```

VALUE PROVIDER OPTIONS

- Controls
 - Query string values
 - View Data
 - Session State
 - Cookies
 - Posted Form data
 - View State
 - Custom
-

DEMO

Filtering

TWO WAY BINDING

```
<EditItemTemplate>
FirstName:
<asp:TextBox ID="FirstNameTextBox" runat="server"
Text='<%= BindItem.FirstName %>' />
LastName:
<asp:TextBox ID="LastNameTextBox" runat="server"
Text='<%= BindItem.LastName %>' />
Title:
<asp:TextBox ID="TitleTextBox" runat="server" Text='<%=
BindItem.Title %>' />
</EditItemTemplate>
```

DEMO

Updating data

MODEL BINDING

ELIMINATES ITERATING FORM VALUES

```
customer.EmailAddress =  
    Request.Form["ctl00$MainContent$fvDataBinding$email"];  
  
=====
```

```
public void UpdateCustomer(int id)  
{  
    using (var db = new ProductsContext())  
    {  
        var customer = db.Customers.Find(id);  
        TryUpdateModel(customer);  
        if (ModelState.IsValid)  
        {  
            db.SaveChanges();  
        }  
    }  
}
```

IMPLICIT MODEL BINDING

```
public void SaveCustomer(Customer customer)
{
    if (ModelState.IsValid)
    {
        using (var db = new WebFormsLab.Model.ProductsContext())
        {
            db.Customers.Add(customer);
            try
            {
                //throw new Exception();
                db.SaveChanges();
            }
            //catch (DbUpdateException)
            catch (Exception)
            {
                ModelState.AddModelError("", "An error occurred saving the data");
                return;
            }
            Response.Redirect("~/Customers.aspx");
        }
    }
}
```

DEMO

Model Binding

DATA VALIDATION

MARKUP BASED ANNOTATIONS

```
<asp:RequiredFieldValidator runat="server"  
ControlToValidate="firstName"  
ErrorMessage="Please enter a value for First  
Name" ForeColor="Red" />
```


MICROSOFT AJAX VALIDATIONS

```
<span id="MainContent_fvDataBinding_ctl02" style="color:Red;visibility:hidden;">Please enter a value for First Name</span>
```

```
<script type="text/javascript">
//
var MainContent_fvDataBinding_ctl02 = document.all ?
document.all["MainContent_fvDataBinding_ctl02"] :
document.getElementById("MainContent_fvDataBinding_ctl02");
MainContent_fvDataBinding_ctl02.controltovalidate =
"MainContent_fvDataBinding_firstName";
MainContent_fvDataBinding_ctl02.errorMessage = "Please enter a
value for First Name";
MainContent_fvDataBinding_ctl02.evaluationfunction =
"RequiredFieldValidatorEvaluateIsValid";
MainContent_fvDataBinding_ctl02.initialvalue = "";
//]]&gt;
&lt;/script&gt;</pre></div>
```

UNOBTRUSIVE VALIDATION

- Reduces page size by reducing inline JavaScript
 - `ValidationSettings.UnobtrusiveValidationMode`
 - Set in:
 - `Web.Config`
 - `Global.asax`
 - Page class
-

ADD SCRIPTMANAGER TO GLOBAL.ASAX

```
ScriptManager.ScriptResourceMapping
```

```
.AddDefinition("jquery",  
    new ScriptResourceDefinition  
    {  
        Path = "~/scripts/jquery-1.4.1.min.js",  
        DebugPath = "~/scripts/jquery-1.4.1.js",  
        CdnPath =  
"http://ajax.microsoft.com/ajax/jQuery/jquery-  
1.4.1.min.js",  
        CdnDebugPath =  
"http://ajax.microsoft.com/ajax/jQuery/jquery-1.4.1.js"  
    });
```

JQUERY BASED VALIDATIONS

```
<span data-val-controltovalidate=
"MainContent_fvDataBinding_firstName"
data-val-errormessage=
"Please enter a value for First Name" id="MainContent_fvDataBinding_ctl02" data-
val="true"
data-val-evaluationfunction=
"RequiredFieldValidatorEvaluateIsValid"
data-val-initialvalue=""
style="color:Red;visibility:hidden;">Please enter a
value for First Name</span>
```

SOME DATA ANNOTATIONS

- CreditCard
 - Phone
 - EmailAddress
 - Range
 - Compare
 - Url
 - FileExtensions
 - Required
 - Key
 - RegularExpression
-

ADD MODEL DATA ANNOTATIONS

```
public class Customer
{
    [Key]
    public int Id { get; set; }
    [Required]
    public string FirstName { get; set; }
    [Required]
    public string LastName { get; set; }
    [Range(0, 130)]
    public int Age { get; set; }
    public Address Address { get; set; }
    [Phone]
    public string DaytimePhone { get; set; }
    [EmailAddress, StringLength(256)]
    public string EmailAddress { get; set; }
}
```

CHECK FOR MODEL VALIDITY

```
public void SaveCustomer(Customer customer)
{
    if (ModelState.IsValid)
    {
        using (var db = new
WebFormsLab.Model.ProductsContext())
        {
            db.Customers.Add(customer);
            db.SaveChanges();
            Response.Redirect("~/Customers.aspx");
        }
    }
}
```

VALIDATION CONTROLS

```
<asp:ValidationSummary runat="server"  
ShowModelStateErrors="true"  
ForeColor="Red" HeaderText="Please  
check the following errors:" />
```

```
<asp:ModelErrorMessage  
ModelStateKey="phone" runat="server"  
ForeColor="Red" />
```


RESULT: UI GOODNESS

CUSTOMER DETAILS

First Name:
 The FirstName field is required.

Last Name:
 The LastName field is required.

Age:
 The Age field is required.

E-Mail:

Phone:

Please check the following errors:

- The FirstName field is required.
- The LastName field is required.
- The Age field is required.

ADDING ERRORS TO MODELSTATE

```
try
{
//throw exception to demonstrate technique
    throw new Exception();
    db.SaveChanges();
}
//Should have better error handling 😊
catch (Exception)
{
    ModelState.AddModelError("", "An error occurred saving
the data");
    return;
}
Response.Redirect("~/Customers.aspx");
```

DEMO

Validation

REQUEST VALIDATION

REQUEST VALIDATION FEATURES

- Integrates AntiXSS library
 - HtmlEncode, HtmlFormUrlEncode, HtmlAttributeEncode
 - XmlAttributesEncode, XmlEncode
 - UrlEncode and UrlPathEncode
 - CssEncode
- Allow for selected validations
 - Deferred (“lazy”) request validation
 - `Context.Request.Unvalidated.Form[“key”]`

ENABLING 4.5 REQUEST VALIDATION

```
<system.web>  
    ...  
    <httpRuntime  
requestValidationMode="4.5"  
encoderType="System.Web.Security.AntiXss.  
AntiXssEncoder, System.Web,  
Version=4.0.0.0, Culture=neutral,  
PublicKeyToken=b03f5f7f11d50a3a" />  
    ...  
</system.web>
```

FIELD LEVEL CONTROL

```
<asp:TextBox runat="server"  
ID="firstName" Text='<%#:  
BindItem.FirstName %>'  
ValidateRequestMode="Disabled" />  
<!-- Disabled, Enabled, Inherited -->
```

DEMO

Request Validation

ASync

ASYNC READING/WRITING REQUESTS

- Builds on ASP.NET 4 GetBufferlessInputStream
- Asych reads with BeginRead/EndRead
- Allows for reading in chunks
- Asynchronous Flushing with BeginFlush/EndFlush

AWAIT, ASYNC MODULES & HANDLERS

- HTTP Modules
 - New helper method
 - EventHandlerTaskAsyncHelper
 - New delegate type
 - TaskEventHandler
 - HTTP Handlers
 - New base type
 - HttpTaskAsyncHandler
 - ASP.NET automatically integrates with ASP.NET pipeline
-

ENABLING ASYNC

- Web.config

```
<add key="aspnet:UseTaskFriendlySynchronizationContext" value="true"/>
```

- Page directive

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/Site.Master" AutoEventWireup="true"
    CodeBehind="ProductDetails.aspx.cs"
Inherits="WebFormsLab.ProductDetails"
    Async="true" %>
```

EXECUTE ASYNC TASK

```
RegisterAsyncTask(new PageAsyncTask(async (t) =>
{
    using (var wc = new WebClient())
    {
        //Instead of wc.DownloadFile
        await wc.DownloadFileTaskAsync(
            imageUrl, Server.MapPath(product.ImagePath));
    }
}));
```

DEMO

Async

HTML5 UPDATES

HTML5 UPDATES

- TextBox TextMode
 - Now supports email, datetime, etc.
- FileUpload supports multiple files
 - Browser Dependent
- Validator controls support HTML5 elements
- Addition of runat="server" in URL paths
 - Supports `<video runat="server" src="~/file.wmv"/>`
- UpdatePanel supports HTML5 Update Fields

CORE ASP.NET UPDATES

ADDITIONAL GOODIES

- Support for WebSockets Protocol
 - Bundling and Minification
 - Performance Improvements
 - Key Performance Factors
 - Sharing Common Assemblies
 - Multi-core JIT compilation
 - Tuning Garbage Collection
 - Prefetching for Web Applications
-

QUESTIONS?



CONTACT ME

- skimedic@gmail.com
- www.skimedic.com/blog
- www.twitter.com/skimedic
- www.hallwayconversations.com
- www.about.me/skimedic

Thank
You!