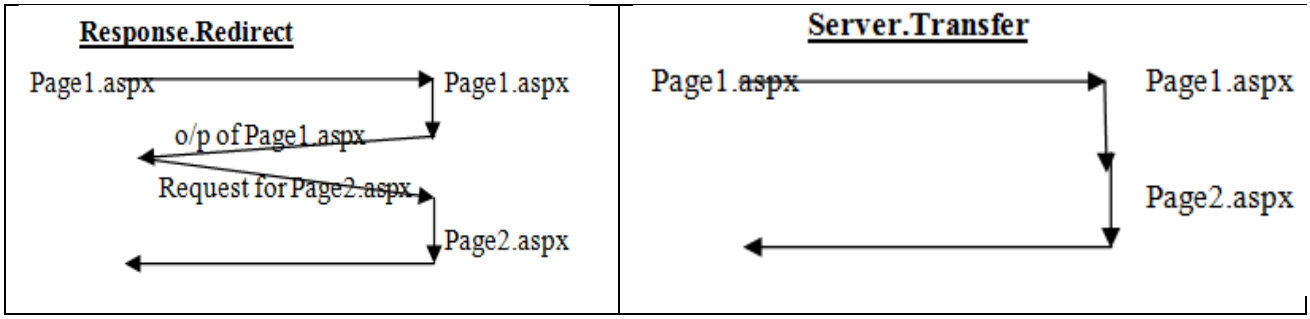
Agenda

1. *Redirecting Options*
2. *Response.Redirect*
3. *Server.Transfer*
4. *Cross Page Postback*
5. *Determine how the Page was Invoked*
6. *Passing Values between Pages*

Redirecting Options

You will often want to redirect users to other pages as part of your Web application. ASP.NET provides the following ways for you to build redirection into your Web pages

1. Hyperlink
2. Response.Redirect
3. Server.Transfer
4. IButtonControl.CrossPagePostBack



HyperLink:

- For navigation without any additional processing, as in menus or lists of links.
- When navigation should be under user control
- Does not pass current page information to the target page.
- Redirects to any page, not just pages in the same Web application.
- Enables you to share information between pages using query string or session state.

Response.Redirect

- For conditional navigation. For example, use this option if the application must determine which page to navigate to based on data provided by the user.
- Performs a new HTTP GET request on the target page. Passes the query string (if any) to the target page. In Internet Explorer, the size of the query string is limited to 2,048 characters.
- Enables you to redirect to any page, not just pages in the same Web application.

Server.Transfer

- For conditional navigation.
- Transfers control to a new page that renders in place of the source page.
- Redirects only to target pages that are in the same Web application as the source page.
- Enables you to read values and public properties from source page.
- Best used in situations where the URL is hidden from the user.
- Does not update browser information with information about the target page. Pressing the refresh or back buttons in the browser can result in unexpected behavior.

Cross-page Postback (IButtonControl.CrossPagePostBack)

- To pass current page information to the target page (as in multi-page forms).
- When navigation should be under user control.

- Redirects to any page, not just pages in the same Web application.
- Enables the target page to read public properties of the source page if the pages are in the same Web application.

To determine how an ASP.NET Web page was invoked	
Invocation method	**Property values**
Hyperlink **Response.Redirect**	○ IsPostBack is set to false. ○ PreviousPage is set to null
Postback	○ IsPostBack is set to true. ○ PreviousPage is set to null
Server.Transfer	○ IsPostBack is set to false. ○ PreviousPage references the source page. ○ PreviousPage.IsCrossPagePostBack that is referenced in the PreviousPage is set to false
Cross-page posting	○ IsPostBack is set to false. ○ PreviousPage references the source page. ○ PreviousPage.IsCrossPagePostBack that is referenced in the PreviousPage is set to true.

Note: **Page.IsCrossPagePostBack** is always false.

Passing Values between Pages

1. If Source and Target Page are in same Web Application
    a. Get control information in the target page from controls in the source page
    b. Create public properties in the source page and access the property values in the target page
    c. Add data to Session in Source Page and retrieve the same in target page
2. If Source and Target Page are in different Web Application
    a. Using QueryString – Request.QueryString (Redirect)
    b. HTTP Post information – Request.Form (Cross-page Postback)

Example

By Default every WebForm Posts the form to itself but if Button's **PostBackUrl** is set to a Url of another page then the Form is Posted to that Url instead of itself.

Page1.aspx

```
<asp:TextBox ID="txtDemo" runat="server"></asp:TextBox><br />
<asp:Button ID="btnRedirect" runat="server" Text="Response.Redirect" OnClick="btnRedirect_Click" />
<asp:Button ID="btnTransfer" runat="server" Text="Server.Transfer" OnClick="btnTransfer_Click" />
<a href="Page2.aspx">Page2.aspx</a>
<asp:Button ID="btnCrossPagePostback" runat="server" PostBackUrl="~/Page2.aspx" Text="Cross Page Postback" />
```

Page1.aspx.cs

```
protected void btnRedirect_Click(object sender, EventArgs e)
{
   string demo = txtDemo.Text;
   Response.Redirect("Page2.aspx?demo=" + demo);
}
protected void btnTransfer_Click(object sender, EventArgs e)
{
   Context.Items.Add("demo", txtDemo.Text);
   Server.Transfer("Page2.aspx");
}
public string DemoText
{
   get
   {
      return txtDemo.Text;
   }
}
```

Page2.aspx

On Top of ASPX Page write the following:

```
<%@PreviousPageType VirtualPath="~/Page1.aspx"%>
```

In Form tag:

```
<asp:Label ID="lblDemo" runat="server"></asp:Label>
```

Program to display command line argument values

```
public partial class Page2 : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
      if (PreviousPage == null)
      {
         //Response.Redirect or Hyperlink is used to come to this page...
         string demo = Request.QueryString["demo"];
         if (demo != null)
            lblDemo.Text = "REDIRECT: " + demo;

      }
```

```
    else if (PreviousPage.IsCrossPagePostBack)
    {
      //Crosspage Postback has been used on Page1
      TextBox txt = (TextBox)PreviousPage.FindControl("txtDemo");
      lblDemo.Text = "CROSSPAGEPOSTBACK: " + txt.Text;
      //OR
      //<%@ PreviousPageType VirtualPath="~/Page1.aspx" %> in Page2.aspx
      lblDemo.Text = "CROSSPAGEPOSTBACK: " + PreviousPage.DemoText;
    }
    else
    {
      //Server.Transfer has been used on Page1
      lblDemo.Text = "Transfer: " + Context.Items["demo"].ToString();
    }
  }
}
```

Notes:

<%@**PreviousPageType VirtualPath**="~/Page1.aspx"%>

- If the above directive is not written in ASPX page, then the datatype of the property **this.PreviousPage** is "**System.Web.UI.Page**" but if this directive is present in "Page2.aspx" then the datatype of "**this.PreviousPage**" is dynamic class inherited from "Page1" i.e. "page1_aspx".
- After adding this directive to Page2.aspx, we can access the public properties of the class Page1 of "Page1.aspx.cs" file. This can be done using "**PreviousPage**" property of "Page" class.

  Example: **PreviousPage.DemoText**
- Once the above directive is used in Page2, Crosspostback to Page2.aspx is allowed only from Page1.aspx .