Postback & Response.Redirect

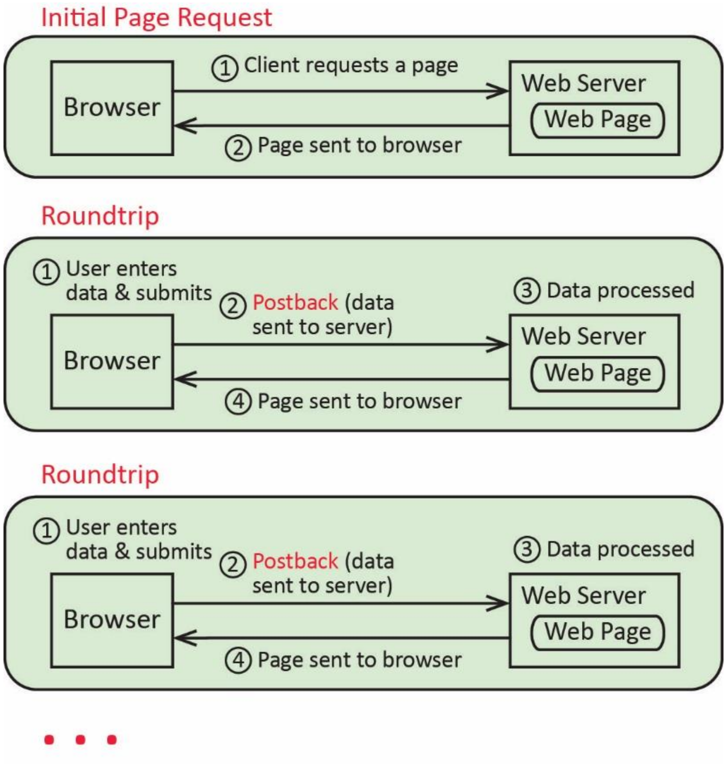
Postback & Roundtrip

1. Definitions

   a. Postback – A user interacts with a page and then presses a button (or other control) to submit the data. This is called a *postback.* We say the data is *posted-back* to the server. Usually a page on the browser postsback to the same page on the server.

   b. The page on the server that has been posted-back to server processes the data that was submitted and then sends the (modified) page back to the browser.

   c. Round-trip – A postback followed by the server processing the submission and returning data (HTML) to the browser.



2. Suppose we have a page named *Postback.aspx.* Consider the following:

   a. Every ASP.NET page has exactly one HTML Form. For example, when the page resides on the server, the form tag will look like this:

      ```
      <form id="form1" runat="server">
      ```

      The *runat* attribute specifies that the form will be processed on the server. When a postback occurs, the server removes the *runat* attribute (because that is not an HTML attribute) and inserts the *method* and *action* attributes. The value of the *action* attribute specifies the page to postback to, in this (and most) cases, the same page. For example, if the page is *Postback.aspx,* then it will postback to this same page:

      ```
      <form method="post" action="./Postback.aspx" id="form1">
      ```

      Note: the "`./`" indicates the current folder. In this case, I have the page in a subfolder named, *pages.* By default, ASP.NET will look for the page in the root folder. Thus, since the page is in a subfolder, the "`./`" prefix says to look in the current folder. All this occurs automatically.

   b. Your page is a subclass of System.Web.UI.Page. The ":" in the signature below is equivalent to "extends" in Java. We will discuss classes, partial classes, inheritance, interfaces later. For example, the code-behind file, *Postback.aspx.cs* will define the class:

      ```
      public partial class Postback : System.Web.UI.Page
      ```

   c. When a postback occurs, an instance of the *Postback* class is created, which represents your page. Code inside this class can reference this instance by using the keyword *this* (just like Java), or more commonly by using the alias *Page*.

   d. Sometimes it is useful to detect whether it is the first time on a page or if it is a postback. Your page class, *Postback* inherits a Boolean property, *IsPostBack* which returns *true* if the request is a postback and *false* otherwise. If *IsPostBack* is *false,* this means that this is the first time this page has been requested.

      ```
      protected void Page_Load(object sender, System.EventArgs e) {
         if (!Page.IsPostBack) {
                // First time on page
         }
         Else {
                // Postback
         }
      }
      ```

3. Why is this useful? The simple answer is that there are times when you want to do a certain thing only the first time on a page and perhaps other things when it is a postback.

   Example – Suppose you have a DropDownList that is populated from a database. The first time a user visits the page, you will need to query the database and bind the results into the drop down. On postbacks you don't need to do this again because the data will persist automatically. Remember, data that is in a Form is retained between postbacks. An HTTP Request packages up the data and sends it back to the server. Thus, there is no need to reload the data (usually).

Response.Redirect

4. One way to transfer control to a new page is:

   ```
   Response.Redirect("newPage.aspx");
   ```

   To reload the current page:

   ```
   Response.Redirect(Request.RawUrl);
   ```

5. A Response.Redirect is always a "first-time" on a page. In other words, if you postback to the same page multiple times, and then,

   ```
   Response.Redirect(Request.RawUrl);
   ```

   It will be the "first-time" on the page, even though the page has never changed on the browser.

6. If you Response.Redirect to a new page, and then use the back arrow on your browse to return to the previous page, then this is neither a postback nor a "first-time"; you are simply pulling the previous page from cache on the browser, so no trip to the server is made.

   PostBack will haunt you until you *get it*.