

Client-side disabling of Buttons

Contributed by Marc
17.02.2008

In Webapps I often ran into the problem that users clicked a submit button more then once. This was because the user did not get immediatly feedback like he does in fat-client applications. To reduce these situations I was looking for some code to disable a ASP.Net button right after the user clicked on it. I found some snippets on the web built a little helper-class to reuse this code and simplify its usage.

To disable a button instantly after a user clicked it JScript on the client (browser) is needed as every server-side code needs a roundtrip to the server in one or another way what will result in above delays we don't want anymore.

To add this client-side JScript code to a ASP.Net webcontrols button I wrote the following helper class. Usage of this class is really simple: just call

Usage example:

```
ButtonHelper.SetSingleClickButton(mypage, mybutton);
```

ButtonHelper.cs:

```
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

namespace iDev.Web
{
    ///
    /// Helperclass for ASP.net buttons.
    ///
    public class ButtonHelper
    {
        public ButtonHelper()
        {}

        ///
        ///Add client-side JScript to disable button after it was clicked.
        ///
        public static void SetSingleClickButton(Page page, Button button)
        {
            // Check pre-conditions
            if (page == null) throw new ArgumentNullException("page");
            if (button == null) throw new ArgumentNullException("button");

            button.Attributes.Add("onclick", "javascript:" +
                button.ClientID + ".disabled=true;" +
                page.GetPostBackEventReference(button));
        }
    }
}
```