Misc JavaScript Tips

Last updated by | Kalin Bowden | Nov 21, 2023 at 12:44 PM CST

Misc JavaScript Tips

Table of Contents

ASP.NET javascript

Preventing Server OnClick from Being Fired on the Client

Returning true/false in the OnClientClick of an ASP.NET control works to prevent the postback OnClick from happening - but the return needs to be done just right.

Can't return a value directly from a function, it has to be like this: (the default is true so no need for that)

HTML Element Attribute Event Handler

```
OnClientClick="if (bthUpdateClick() == false) return (false);"
```

JavaScript to Handle Preventing the OnClick Event

```javascript
function bthUpdateClick() {
    var multiSelect = document.getElementById("ddlMultiSelect");
    if (multiSelect.value != "Yes" && multiSelect.value != "No")
    {
        alert('Please choose Multi Select for this Group to be Yes or No.\r\nThis determines whether m
        return false;
    }

    return true;
}
```

Preventing the Browser from Caching JavaScript Resource Files

Useful for when getting javascript file to refresh when Ctrl-F5 isn't doing it

```html
<script type="text/javascript" src="assets/js/forms/ImportPDFTemplate.js?refresh=<%#Eval("Guid.NewGuid().T
```

Retrieving Custom Element Attributes

Useful for fetching ASP.NET custom attributes from the code behind

```javascript
var paragraphX = buttons.attr("paragraphX");
var paragraphY = buttons.attr("paragraphY");
```
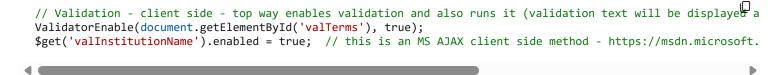
Using the Jquery Included with Telerik

[Use the Jquery included with Telerik](#)

```html
<telerik:RadScriptManager ID="VRRadScriptManager" EnableScriptCombine="false" runat="server">
    <Scripts>
            <asp:ScriptReference Assembly="Telerik.Web.UI" Name="Telerik.Web.UI.Common.Core.js" />
            <asp:ScriptReference Assembly="Telerik.Web.UI" Name="Telerik.Web.UI.Common.jQuery.js" />
    </Scripts>
</telerik:RadScriptManager>
```

Referencing JQuery Using JQuery Include with Telerik

```javascript
// Instead of $ use $telerik.$
$telerik.$('.rwCloseButton');

// from within a RadWindow popup the JS is like this - although from debug console the native without ",parent
$telerik.$('.rwCloseButton',parent.document).hide();
Page.ClientScript.RegisterStartupScript(GetType(), "HideCloseButton", "<script language='javascript'>$telerik.
```
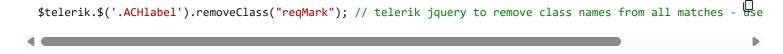
Client-Side Enabling of Validators

```
// Validation - client side - top way enables validation and also runs it (validation text will be displayed a
ValidatorEnable(document.getElementById('valTerms'), true);
$get('valInstitutionName').enabled = true;  // this is an MS AJAX client side method - https://msdn.microsoft.
```

Telerik JQuery to Remove Class Names from All Matches

```
$telerik.$('.ACHlabel').removeClass("reqMark"); // telerik jquery to remove class names from all matches - use
```

Telerik JQuery to Disable/Enable Entry Fields

```
$telerik.$('.ACHentry').prop("disabled", false) // telerik jquery to disable/enable entry fields
```

Hiding a Validator Validation Message

Raw js to hide a validation message (no more validation messages will show for the validator until it's unhidden - this is mostly useless?) - the .enabled one from $get above works better.

```
document.getElementById('valInstitutionName').style.visibility = 'hidden';    // raw js to hide a validation m
```

Hide RadWindow Close Button

Telerik jquery script to hide the "close" button on a RadWindow popup - useful if the close button was wanted until a certain point.

```
// telerik jquery script to hide the "close" button on a RadWindow popup - useful if the close button was want
Page.ClientScript.RegisterStartupScript(GetType(), "HideCloseButton", "<script language='javascript'>$telerik.
```

Client-Side HTML Escaping

```javascript
// client-side HTML escape
var entityMap = {
  '&': '&amp;',
  '<': '&lt;',
  '>': '&gt;',
  '"': '&quot;',
  "'": '&#39;',
  '/': '&#x2F;',
  '`': '&#x60;',
  '=': '&#x3D;'
};

function escapeHtml (string) {
  return String(string).replace(/[&<>"'`=\/]/g, function (s) {
    return entityMap[s];
  });
}

// Can also set properties directly using jquery and they will be escaped
// before:
// <div class="someClass">text</div>
var someHtmlString = "<script>alert('hi!');</script>";

// set a DIV's text:
$("div.someClass").text(someHtmlString);
// after:
// <div class="someClass">&lt;script&gt;alert('hi!');&lt;/script&gt;</div>

// get the text in a string:
var escaped = $("<div>").text(someHtmlString).html();
// value:
// &lt;script&gt;alert('hi!');&lt;/script&gt;

// to select an element whose attribute has a value like this, don't escape the escaped chars, just escape the
// <td title="&lt;b>Development City&lt;/b> (Pending), UMN Purchasing">
$telerik.$('td[title^=\\<b\\>]')

// this will not work to set HTML into an attribute, it will get escaped
$telerik.$('td[title^=\\<b\\>]').attr('title','<b>bold this</b> and not this')
```

Client-Side Objects

Object	Data Type	Description
Page_IsValid	Boolean	Indicates whether the page is currently valid. The validation scripts keep this up to date at all times.
Page_Validators	Array of elements	This is an array containing all of the validators on the page.
Page_ValidationActive	Boolean	Indicates whether validation should take place. Set this variable to False to turn off validation programmatically.
isvalid	Boolean	This is a property on each client validator indicating whether it is currently valid.
Page_ValidationSummaries	Array of elements	This is an array containing all of the validation summaries on the page.

Client-Side APIs

ValidatorValidate(val)
: Takes a client-validator as input. Makes the validator check its input and update its display.

```
ValidatorValidate(document.getElementById('valProvide'));
```

ValidatorEnable(val, enable)
: Takes a client-validator and a Boolean value. Enables or disables a client validator. Being disabled will stop it from evaluating and it will always appear valid.

```
ValidatorEnable(document.getElementById('valRoutingNumber'), false)  // will disable the validator and hide it
```

ValidatorHookupControl(control, val)
: Takes an input HTML element and a client-validator. Modifies or creates the element's change event so that it updates the validator when changed. This can be useful for custom validators that depend on multiple input values.

```
ValidatorHookupControlID('controlid','validatorid')  // I think this and above may allow one to work with 2 co
```

Page_ClientValidate(val)
: Takes validation group as input and validate all validators of the group and returns bool.

ValidationSummaryOnSubmit(validationGroup)
: Parameters: validationGroup: string, Used to show Validation Summary of this group (if not supplied it will work on all ValidationSummary in page)

ValidationSummaryOnSubmit

How To Call:

- ValidationSummaryOnSubmit();
- ValidationSummaryOnSubmit('grpAdd');
  - It is used to show ValidationSummary of particular group(if specified). Suppose you have called ValidatorEnable of a ValidationControl and want to show ValidationSummary on failure, you can use function ValidationSummaryOnSubmit to achieve this.
    - not sure if this applies in any way, wrote attempting to fix a duplicating client side validator issue which was resolved by removing some ajax

```javascript
/* clear out duplicate validators, unsure why this is happening but sometimes duplicates are being created in
   and it's causing duplicate validation messages to show in the validation summary */
var validatorNames = [];
for (var validatorIndex = 0; validatorIndex < Page_Validators.length; validatorIndex++) {
        if (validatorNames.includes(Page_Validators[validatorIndex].id)) {
                console.log('Disposing: ' + Page_Validators[validatorIndex].id);
                Page_Validators[validatorIndex].dispose();
        }
        else {
                validatorNames.push(Page_Validators[validatorIndex].id);
        }
}

// Hide all validators if they are named like validator1, validator2, etc, could also enable/disable - better
var n = 0;
var z = '';
for (var i = 0; i < Page_Validators.length; i++) {
    n += 1;
    z = 'ctl00_MainContent_validator' + n;
    document.getElementById(z).style.visibility = 'hidden';
}

function EnableValidators() {
        Page_Validators.forEach(EnableValidator);
}
function EnableValidator(item, index, arr) {
        //ValidatorEnable(item, true);   // this way works fine first cycle when starting from disabled (doesn
        item.enabled = true;
}

function DisableValidators() {
        Page_Validators.forEach(DisableValidator);
}
function DisableValidator(item, index, arr) {
        //ValidatorEnable(item, false);   // this way is probably ok either way
        item.enabled = false;
}


// Client side validation - checkbox example when the checkbox must be checked
<asp:CheckBox ID="cbIAgree" ClientIDMode="Static" Text="I agree that credits can be made to this account" Text
<asp:CustomValidator ID="valTerms" ClientIDMode="Static" ErrorMessage="Please accept the terms..." ClientValid
        <span class="flaticon-fieldError_Error"></span>
</asp:CustomValidator>

function ValidateTerms(src, args) {
        var cbx = document.getElementById('cbIAgree');
        args.IsValid = cbx.checked;
}

// For trimming there is a function provided by the asp.net validation scripts - does left and right trim
textBox1Value = ValidatorTrim(textBox1Value);
textBox2Value = ValidatorTrim(textBox2Value);
```

RadToolBar Manual PostBack from JavaScript

```javascript
// RadToolBar manual postback from JS - pass toolbar item number, 0-indexed
__doPostBack("<%=rtb.UniqueID %>", "0");
```

Javascript Delay

setTimeout waits for X duration and then calls method f()
setInterval calls method f() every X durations

```javascript
// Javascript delay
setTimeout(() => { console.log("Now!"); }, 2000);
```

Forcing Page Level JavaScript to Wait for All External JavaScript Files to Load Before Executing

Add the defer attribute to the script tag to force it to wait for all external JavaScript to load before executing any JavaScript

```html
<script defer type="text/javascript">
```