HTML Encoding in .NET

Available Methods

James Jardine

2012

Table of Contents

Abstract

Cross Site Scripting is a serious vulnerability and it is difficult from both a developer and code reviewer standpoint to understand the different options for HTML output encoding. In addition to the multiple functions available within the .Net framework, there are third-party libraries also available. The purpose of this document is to describe the different methods available within .Net and the characters that get encoded. This document was created prior to the release of .Net 4.5 and is just observations made about the different encoding methods identified. This does not represent all possible ways to encode HTML output.

Warning: It is important to note that the focus is strictly on HTML encoding and does not cover attribute, javascript, css, or other data contexts.

Framework Methods

System.Security.SecurityElement.Escape

This method is more relative to XML elements than it is for HTML elements. The table below shows the characters that are encoded and the values they are replaced with.

Invalid Character	Replaced With
<	&lt;
>	&gt;
"	&quot;
&	&amp;
'	&apos; **

** It is important to note that this method uses &apos; to replace the apostrophe (') character. This is valid within an XML context, but not recommended for HTML. It appears that in Firefox this will be rendered correctly. In Internet Explorer, however, it renders &*apos;* to the screen which is not the desired output.

System.Net.WebUtility.HTMLEncode (4.0+ Only)

The table below shows the characters that are encoded and the values they are replaced with. This method was added in .Net 4.0.

Invalid Character	Replaced With
<	&lt;
>	&gt;
"	&quot;
&	&amp;
'	&#39;

System.Web.HttpUtility.HTMLEncode

The table below shows the characters that are encoded and the values they are replaced with.

Invalid Character	Replaced With
<	&lt;
>	&gt;
"	&quot;
&	&amp;
'	&#39; (.Net 4.0 Only)

In 4.0 the logic was changed so that developers could replace the default encoding method to use their own. By default, this will call System.Net.WebUtility.HTMLEncode() under the covers. If another encoder has been defined in the web.config then that method will be called.

System.Web.HttpServerUtility.HTMLEncode

This method makes a single call to System.Web.HttpUtility.HTMLEncode.

Server.HTMLEncode

The table below shows the characters that are encoded and the values they are replaced with.

Invalid Character	Replaced With
<	&lt;
>	&gt;
"	&quot;
&	&amp;
'	&#39; (.Net 4.0 Only)

System.Web.Security.AntiXss.HTMLEncode

This method was added in version 4.5 of the .Net framework. It is the same as the Web Protection library listed below except it is included by default in the framework now. There is no need for the external library.

External Methods

Microsoft.Security.Application.Encoder.HTMLEncode

The Web Protection Library, formerly known as the AntiXSS library, provides as set of functions to encode data to protect against XSS. Unlike the built in framework methods, this library uses a white-list technique. In a white-list, there is a large list of acceptable characters that are left alone and the rest are encoded.

Pro: Better protection against possible future exploits.

Con: Can cause a performance impact.

The Web Protection Library can be downloaded from: http://wpl.codeplex.com/.

** In .Net 4.5 Developer Preview, AntiXSS has been included in the framework under the System.Web.Security.AntiXss namespace.

ESAPI

This library is an open source project on OWASP. The purpose is to provide an enterprise library that can be used by a large audience. You can check it out at:

https://www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API#tab=.NET

Inline Code Block

New to .Net 4.0 – This works to auto-encode data between the tags. Remember that this only performs HTML encoding, so if this block is in another context it may not provide protection against cross-site scripting.

```
<%: textToEncode %>
```

Example Usage:

```
<span><%: txtUserName.Text %></span>
```

New to .Net 4.5 – This works for data binding context to auto-encode the data. Remember that this only performs HTML encoding, so if this block is in another context it may not provide protection against cross-site scripting.

```
<%#: textToEncode %>
```

ASP.NET HTML Encoding Reference

System.Security.SecurityElement.Escape

Invalid Character	Replaced With
<	&lt;
>	&gt;
"	&quot;
&	&amp;
'	&apos;

Server.HTMLEncode

Invalid Character	Replaced With
<	&lt;
>	&gt;
"	&quot;
&	&amp;
'	&#39; * .Net 4.0

System.Net.WebUtility.HTMLEncode

Invalid Character	Replaced With
<	&lt;
>	&gt;
"	&quot;
&	&amp;
'	&#39; * .Net 4.0

System.Web.HttpServerUtility.HTMLEncode

This calls the System.Web.HttpUtility.HtmlEncode method.

System.Web.HttpUtility.HTMLEncode**

Invalid Character	Replaced With
<	&lt;
>	&gt;
"	&quot;
&	&amp;
'	&#39; * .Net 4.0

Inline Code Blocks

<%:texttoencode%> (.Net 4.0)

<%#:databoundtext%> (.Net 4.5)

** In 4.0, by default, this calls System.Net.WebUtility.HTMLEncode. Otherwise, it calls the HTMLEncode method of the defined HTTPEncoder class defined in the Web.Config.