

Re: Special Characters in Query String

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2005-05/msg01626.html>

- *From:* "Juan T. Llibre" <nomailreplies@xxxxxxxxxxxxx>
 - *Date:* Mon, 9 May 2005 17:48:09 -0400
-

Hi, Joerg.

Take a look at the result :

<http://asp.net.do/test/utf-8.aspx>

(Browsing it with IE6 set to use utf-8 results in the same.)

Now, look at the same code, using iso-8859-1 :

<http://asp.net.do/test3/iso-8859-1.aspx>

(Browsing with IE6 set to "Auto-select", results in Western European getting selected.)

I've had numerous problems with utf-8, all of which result in common characters in spanish not getting displayed.

Using iso-8859-1 gets rid of the problems.

Juan T. Llibre
ASP.NET MVP
<http://asp.net.do/foros/>
Foros de ASP.NET en Español
Ven, y hablemos de ASP.NET...

"Joerg Jooss" <news-reply@xxxxxxxxxxxxx> wrote in message news:xn0e2241bm7cozb002@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

> Juan T. Llibre wrote:

>

>> re:

>> > assuming you serve mostly content based on Western European

>> > languages

>>

>> That's a mighty big assumption to make, don't you think ?

>

> No, not all. It just doesn't make sense to use UTF-8 from a bandwidth

Re: Special Characters in Query String

> perspective once you need to serve a lot of content in on other
> languages or scripts, as one character may require up to six bytes.
>
>> Even so, using UTF-8, I haven't found a way
>> to display characters in the high-ascii 128-255 range,
>> which several Western European languages require.
>
> Then you've been doing something wrong. Let me quote the Unicode
> standard document:
>
> "The Unicode Standard provides 1,114,112 code points, most of which are
> available for encoding of characters. The majority of the common
> characters used in the major languages of the world are encoded in the
> first 65,536 code points, also known as the Basic Multilingual Plane
> (BMP). The overall capacity for more than a million characters is more
> than sufficient for all known character encoding requirements,
> including full coverage of all minority and historic scripts of the
> world."
>
> There's no civilized 8 bit encoding that cannot be replaced by Unicode
> ;-)
>
>> I've been able to do that by using iso-8859-1.
>
> If you can display your characters with ISO-8859-1, you have
> accidentally or willingly switched the response encoding.
>
>> Can you post a sample, using utf-8,
>> which displays characters in the high-ascii 128-255 range ?
>
> Let's avoid the errors of the past — there's no such thing as Hi ASCII
> or 8 bit ASCII. US-ASCII and all its localized clones (ISO-646-xx) are
> 7 bit. ISO-8859-x, Windows-125x are built "on top of" US-ASCII.
>
> If you want to see UTF-8 in real live, feel free to visit my homepage
> which is running dasBlog and serves content in UTF-8.
>
>> I'd be a bit more liable to believe you if you did.
>>
>> Specifically, if you could show me how to display the
>> characters ñ, Ñ, ÿ, ŷ, á, é, í, ó, and ú with utf-8, I'd be grateful.
>
> OK, do the following:
>
> 1. Create a new WebForm in a new ASP.NET project. Make sure that your
> web.config's <globalization/> looks like this:
> <globalization
> requestEncoding="utf-8"
> responseEncoding="utf-8" />
>
> 2. Add a Label control to the WebForm, call it "label" and set its text

Re: Special Characters in Query String

> in the property control to the empty string.
>
> 3. Implement the Page_Load method like this:
> this.label.Text = "ñ, Ñ, ÿ, ç, á, é, í, ó, and ú";
>
> 4. Run the WebForm — it should display the text given above.
>
> And that's pretty much it.
>
> See, once you have the characters in a string object and the page is
> not rendered correctly, one of the following errors may have occurred:
>
> -- Your browser is configured to use a fixed encoding, which does not
> match and is not compatible with the real encoding (like ISO-8859-1
> vs. UTF-8 for non-ASCII content). This will lead to weird or missing
> characters in web pages (responses).
>
> -- Neither the HTTP response nor the HTML source specify the character
> encoding. In that case, the browser must guess, and of course it can
> guess wrong. This should never happen with any decent web application
> technology. ASP.NET for example sends a proper
> Content-Type: text/html; charset=utf-8
> HTTP header.
>
> Then there's the case that you have a build-time error. In the example
> given above, I've hardcoded the string in my source file. The ASP.NET
> page processor needs to know the source file's encoding to decode these
> characters correctly — that's what the fileEncoding attribute of the
> <globalization/> element does. If you'd changed that attribute to an
> incompatible one or one that cannot represent a given character, that
> particular character would be already missing in the resulting string
> object. This isn't usually a problem as display text belongs into
> satellite assemblies anyway, but for simple applications this needs to
> be kept in mind.
>
> I hope this helps.
>
> Cheers,
> --
> <http://www.joergjooss.de>
> <mailto:news-reply@xxxxxxxxxxxxxx>

-
- *Follow-Ups:*
 - ◆ ***Re: Special Characters in Query String***
 - ◇ *From: Joerg Jooss*

- **References:**

- ◆ **Special Characters in Query String**
 - ◇ From: SMG
- ◆ **Re: Special Characters in Query String**
 - ◇ From: Kim Bach Petersen
- ◆ **Re: Special Characters in Query String**
 - ◇ From: Joerg Jooss
- ◆ **Re: Special Characters in Query String**
 - ◇ From: Kim Bach Petersen
- ◆ **Re: Special Characters in Query String**
 - ◇ From: Juan T. Llibre
- ◆ **Re: Special Characters in Query String**
 - ◇ From: Joerg Jooss
- ◆ **Re: Special Characters in Query String**
 - ◇ From: Juan T. Llibre
- ◆ **Re: Special Characters in Query String**
 - ◇ From: Joerg Jooss

- Prev by Date: **Re: Help .. Button does not function properly.**
- Next by Date: **ASP.NET compile model detail**
- Previous by thread: **Re: Special Characters in Query String**
- Next by thread: **Re: Special Characters in Query String**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**