Trouble encoding a u umlaut with in a .Net http handler

Asked 16 years ago Modified 9 years, 2 months ago Viewed 5k times

7

I have a JavaScript request going to a ASP.Net (2.0) HTTP handler which passes the request to a java web service. In this system special characters, such as those with an accent do not get passed on correctly.

E.G.

- Human input: `Düsseldorf`
- becomes a JavaScript asynch request to `http://site/serviceproxy.ashx?q=D%FCsseldorf`, which is valid in ISO-8859-1 as well as in UTF-8 as far as I can tell. (unless it's %c3%bc in UTF-8)
- `HttpContext.Current.Request.QueryString.Get("q")` returns `D�sseldorf` which is where trouble begins.
- but `HttpUtility.UrlEncode(HttpContext.Current.Request.QueryString.Get("q"), Encoding.GetEncoding("ISO-8859-1"))` returns `D%3fsseldorf` (a '?')
- and `HttpUtility.UrlEncode(HttpContext.Current.Request.QueryString.Get("q"), Encoding.UTF8)` returns `D%ef%bfsseldorf`

So it the value doesn't get decoded nor re-encoded correctly to be passed on to the java service.

- Notice `HttpContext.Current.Request.Url.Query` is `?q=D%FCsseldorf&output=json&from=1&to=10`
- while `HttpContext.Current.Request.QueryString.ToString()` is `q=D%ufffdsseldorf&output=json&from=1&to=10`

Why is this, and how can I tell the `HttpContext` to honor the request headers which include:

```
Content-Type=application/x-www-form-urlencoded;+charset=UTF-8
```

and decode the URL's `QueryString` using the UTF-8 charset.

Addendum: As the answer notes, the trouble lies not so much in the decoding as the encoding; using `escape()` in JavaScript does not escape according to UTF-8, while

using `encodeURIComponent()` does.

asp.net javascript character-encoding diacritics encodeuricomponent

Share
Improve this question
Follow

edited Oct 15, 2015 at 12:23

marc_s
753k ● 183 ● 1.4k ● 1.5k

asked Nov 25, 2008 at 22:26

dlamblin
45.3k ● 22 ● 104 ● 144

2 Answers

Sorted by: Highest score (default)

7

I don't know what the default character encoding used by your server (IIS?) is, or if it can be changed, but I can tell you a few things that might help.

0xFC is the ISO-8859-1 encoding for ü. While the Unicode code point is U+00FC, when encoded with UTF-8, this requires two bytes, and becomes 0xC3 0xBC.

If a UTF-8 decoder were to see the illegal byte sequence 0xFC, it would decode it as a Unicode "replacement character", U+FFFD, and pick up where it saw the beginning of another valid byte sequence, in this case 's'.

The reason you get `%3f` is that '?' is the "replacement character" for the Latin character set, similar to  in the Unicode character set.

I believe what you're seeing is the client encoding with ISO-8859-1, but the server is decoding with UTF-8. As soon as it hits the server, your data is corrupted. I recommend that you modify the client to use UTF-8 encoding; it should be requesting `http://site/serviceproxy.ashx?q=D%C3%BCsseldorf`

It sounds like you are constructing these URLs from JavaScript, so you should use the `encodeURI` and `encodeURIComponent` functions, not `escape`.

Share
Improve this answer
Follow

edited Nov 26, 2008 at 1:06

answered Nov 26, 2008 at 1:00

erickson
269k ● 59 ● 401 ● 497

> And here I was looking for a way to tell escape() I wanted to use UTF-8. I didn't think to look at encodeURIComponent(). Also I misunderstood UTF-8 and thought %FC might be valid for both. – dlamblin Nov 26, 2008 at 16:43

I am getting the same problem with an ASP.NET generic handler when the URL is typed directly into IE8. Characters are being sent through as char 65533, and yet I do

1

have IE8 set to

```
[x] Send UTF-8 URLs.
```

In my scenario, I'm debugging an HTTP handler in Visual Studio and typing the address of the handler directly into the browser:

```
http://localhost/myHandler.ashx?term=xxxxxx
```

and then stepping through the code. The client will be passing UTF-8 encoded URLs, but is there a way to debug the code when IE8 running on the development machine is the client?

Share Improve this answer Follow

answered Jan 21, 2010 at 19:59

Tim

23 • 4