

弹指一挥间

好好做事，学习待人 (大数据分析/.NET/JAVA)技术交流QQ:860280456; .NET/JAVA技术交流群:192028174

博客园 :: 首页 :: 新随笔 :: 联系 :: 订阅  :: 管理 288 Posts :: 4 Stories :: 70 Comments :: 0 Trackbacks

公告

昵称: v.e.n.u.s

园龄: 7年1个月

粉丝: 96

关注: 8

+加关注

最新随笔

1. MYSQL数据库高可用方案探究
2. [MSSQL]表变量和临时表的区别
3. [Memcached] telnet命令
4. Redis在C#中的使用及Redis的封装
5. [Memcached]分布式缓存系统Memcached在Asp.net下的应用
6. Redis与Memcached的区别
7. 程序员职业规划
8. [转]别转管理，十年程序员老鸟给新手的几条忠告
9. 技术人员的未来：做技术还是做管理？
10. Hadoop相关基础知识

我的标签

svn(1)
Winform(1)

随笔分类(295)

.NET(21)
.NET面试笔试题(4)
[ORM](2)
[WP](29)
[多线程](4)
[通讯](3)
[正则表达式](1)
【.NET B/S】(12)
【.NET C/S】(21)
【.NET MVC】(12)
【C# 基础】(10)
【C# 相关用法】(3)
【Framework】(2)
【IHttpHandler】(7)
【Java 基础】(5)
【Java】(1)
【SQL_MSSQL】(28)
【SQL_MySQL】(7)
【SQL_Oracle】(16)

encodeURIComponent来解决URL传递时的中文问题

在AJAX浏览器来进行发送数据时，一般它所默认的都是UTF-8的编码。

使用JQUERY中所提供的方法来做操作

encodeURIComponent

```
function verify() {
```

```
    //解决中文乱麻问题的方法1，页面端发出的数据作一次
```

```
    encodeURIComponent,
```

```
    服务器段使用进行转移成UTF-8;
```

```
    //解决中文乱麻问题的方法2，页面端发出的数据作两次
```

```
    encodeURIComponent,
```

这个做的好处在于，不管浏览器用户在页面来设置编码，服务器所采用的编码来做一次URLencode转换成UTF-8.

```
    var url = "AJAXServer?name=" +
```

```
    encodeURIComponent($("#userName").val());
```

```
    url = convertURL(url);
```

```
    $.get(url,null,function(data){
```

```
        $("#result").html(data);
```

```
    });
```

```
}
```

说明：1 .encodeURIComponent函数主要是来对URI来做转码，它默认是采用的UTF-8的编码。

2. UTF-8编码的格式：一个汉字来三个字节构成，每一个字节会转换成16进制的编码，同时添加上%号。

这里要做两次encodeURIComponent的原因以及说明：

具体说明其中具体的原理分析如下，假设页面端输入的中文是一个“中”，按照下面步骤进行解码

1.第一次encodeURIComponent，按照utf-8方式获取字节数组变成[-28,-72-83]，对字节码数组进行遍历，把每个字节转化成对应的16进制数，这样就变成了[E4,B8,AD],最后变成[%E4,%B8,%AD]

2.第二次encodeURIComponent，把数组最后变成[%25E4,%25B8,%25AD]然后就把处理后的数据[%25E4,%25B8,%25AD]发往服务器端，当应用服务器调用getParameter方法，getParameter方法会去向应用服务器请求参数

应用服务器最初获得的就是发送来的[%25E4,%25B8,%25AD]，应用服务器会对这个数据进行URLdecode操作，URLdecode操作和encodeURIComponent操作是相反的操作，处理结果就是[%E4,%B8,%AD]，并把这个值返回给getParameter方法

【SQL_数据库优化】(1)
 【Unity_Scripts】(1)
 【编码规范】(4)
 【程序设计】(2)
 【工具使用】(9)
 【缓存】(8)
 【架构之路】(11)
 【前端】(12)
 【设计模式】(1)
 【算法】加密解密及其他(4)
 【源代码管理】(4)

Android(3)

I/O(1)

IIS(2)

Json(1)

Memcached(2)

SEO(10)

Unity3D(3)

WCF(6)

WPF(1)

XML(5)

编码转码(2)

常查系列(1)

网络协议(1)

杂谈(9)

职场(3)

随笔档案(288)

2017年3月 (2)

2017年2月 (17)

2016年12月 (2)

2016年9月 (2)

2016年7月 (1)

2016年6月 (3)

2016年5月 (3)

2016年4月 (1)

2016年2月 (1)

2016年1月 (2)

2015年12月 (1)

2015年11月 (7)

2015年9月 (2)

2015年8月 (2)

2015年7月 (3)

2015年6月 (5)

2015年5月 (5)

2015年4月 (11)

2015年3月 (7)

2015年2月 (3)

2015年1月 (42)

2014年12月 (1)

2014年11月 (3)

2014年10月 (40)

2014年9月 (5)

2014年8月 (13)

2014年7月 (10)

2014年6月 (8)

2014年4月 (5)

2014年3月 (10)

2013年10月 (3)

2013年9月 (1)

2013年8月 (1)

然后再在服务器端中调用相应的URL转码方法或者是函数 就可以把数据还原成最初页面发送过来的中文“中”了。

JS、C#编码解码

escape不编码字符有**69**个：*****，**+**，**-**，**.**，**/**，**@**，**_**，**0-9**，**a-z**，**A-Z**

encodeURIComponent不编码字符有**82**个：**!**，**#**，**\$**，**&**，**'**，**(**，**)**，*****，**+**，**,**，**-**，**.**，**/**，**:**，**;**，**=**，**?**，**@**，**_**，**~**，**0-9**，**a-z**，**A-Z**

encodeURIComponentComponent不编码字符有**71**个：**!**，**'**，**(**，**)**，*****，**-**，**.**，**_**，**~**，**0-9**，**a-z**，**A-Z**

1.

JS: escape :

- **js**使用数据时可以使用**escape**
- 例如：搜藏中**history**纪录。
- **0-255**以外的**unicode**值进行编码时输出**%u******格式，其它情况下**escape**，**encodeURIComponent**，**encodeURIComponentComponent**编码结果相同。

解码使用：unescape

C#:

HttpUtility.UrlEncode

HttpUtility.UrlDecode

2.

JS: encodeURIComponent :

- 进行**url**跳转时可以整体使用**encodeURIComponent**
- 例如：
Location.href=encodeURIComponent("http://cang.baidu.com/do/s?word=百度&ct=21");

解码使用decodeURI();

C#: decodeURIComponent

3.

JS: encodeURIComponent :

- 传递参数时需要使用**encodeURIComponent**，这样组合的**url**才不会被**#**等特殊字符截断。
- 例如：**<script language="javascript">document.write('退出');</script>**

2013年7月 (2)
2013年5月 (5)
2013年4月 (7)
2013年3月 (20)
2013年2月 (13)
2013年1月 (7)
2012年10月 (4)
2012年9月 (3)
2012年8月 (2)
2012年4月 (3)

文章分类(2)
.NET概念(1)
交通(1)

相册
.NET
MSSQL

积分与排名
积分 - 137252
排名 - 1616

最新评论
1. Re:通过Ajax post Json
类型的数据到Controller
感谢,解决了我的问题
--你是不是想家了
2. Re:WS+MQ+WCF+EF(Code First)
赞
--禅道
3. Re:CodeSmith 7.01破
解下载
完美激活
--爱学习的小蜗牛
4. Re:CodeSmith 7.01破
解下载
老是提示 激活码无效, 按照
你的方法也不行啊
--华季
5. Re:[Ajax] 使用Ajax异步
上传图片文件(非Form表单提
交)
@半壁江山这个是服务端代
码, 代码部署在哪里, 就能上
传到哪里 (除io读写权限限制
外) ...
--v.e.n.u.s

阅读排行榜
1. SVN使用教程(152582)
2. 读写文本(.txt)文件 .NET(23534)
3. [Ajax] 使用Ajax异步上
传图片文件(非Form表单提
交)(17427)
4. ORACLE基本语法(1458
6)
5. CodeSmith 7.01破解下

解码使用decodeURIComponent()

C#:

[HttpContext.Current.]Server.UrlDecode

[HttpContext.Current.]Server.UrlEncode

URL编码与两次encodeURIComponent

当使用地址栏提交查询参数时，如果不编码，非英文字符会按照操作系统的字符集进行编码提交到服务器，服务器会按照配置的字符集进行解码，所以如果两者不一致就会导致乱码。

encodeURIComponent函数采用UTF-8对URL进行编码，所以如果服务器在进行解码时使用的是其他的编码方式就会出现乱码，默认的服务器配置的解码字符集都不是UTF-8，所以大部分情况下地址栏提交中文查询参数时会产生乱码；针对这种情况，可以连续使用两次encodeURIComponent在客户端(主要指浏览器)对非英文字符进行编码，然后在服务端使用java.net.URLDecoder(String."UTF-8")解码，即可得到正确的中文。

如果只进行一次encodeURIComponent，得到的是UTF-8形式的URL，服务器端通过request.getParameter()解码查询参数(通常是iso-8859-1)就会得到乱码。

如果进行两次encodeURIComponent,第一次编码得到的是UTF-8形式的URL，第二次编码得到的依然是UTF-8形式的URL，但是在效果上相当于首先进行了一次UTF-8编码(此时已经全部转换为ASCII字符)，再进行了一次iso-8859-1编码，因为对英文字符来说UTF-8编码和ISO-8859-1编码的效果相同。在服务器端，首先通过request.getParameter()自动进行第一次解码(可能是gb2312,gbk,utf-8,iso-8859-1等字符集,对结果无影响)得到ascii字符，然后再使用UTF-8进行第二次解码，通常使用java.net.URLDecoder("", "UTF-8")方法。

两次编码两次解码的过程为：

UTF-8编码->UTF-8(iso-8859-1)编码->iso-8859-1解码->UTF-8解码，编码和解码的过程是对称的，所以不会出现乱码。

encodeURIComponent函数主要是来对URI来做转码，它默认是采用的UTF-8的编码。
· UTF-8编码的格式:一个汉字来三个字节构成，每一个字节会转换成16进制的编码，同时添加上%号。

假设页面端输入的中文是一个“中”，按照下面步骤进行解码

- 1.第一次encodeURIComponent，按照utf-8方式获取字节数组变成[-28,-72-83]，对字节码数组进行遍历，把每个字节转化成对应的16进制数，这样就变成了[E4,B8,AD],最后变成[中,中,中] 此时已经没有了多字节字符，全部是单字节字符。
- 2、第二次encodeURIComponent，进行编码，会把%看成一个转义字符，并不编码%以后字符，会把%编码成%。把数组最后变成[中,中,中]然后就把处理后的数据[中,中,中]发往服务器端，

当应用服务器调用**getParameter**方法，**getParameter**方法会去向应用服务器请求参数

应用服务器最初获得的就是发送来的[]，应用服务器会对这个数据进行**URLdecode**操作，应用服务器进行解码的这一次，不管是按照**UTF-8**，还是**GBK**，还是**ISO-8859**，都能得到[]，因为都会把**%**解析成**%**。并把这个值返回给**getParameter**方法

3、再用**UTF-8**解码一次，就得到"中"了。

想想看，如果不编码两次，当服务器自动解码的时候，假如是按照**ISO-8859**去解码**UTF-8**编码的东西，就是会出现乱码。

```
1. document.authorityForm.action =
    basePath3+"User_viewUser.do?
    id="+id+"&roleName="+encodeURIComponent(encodeURIComponent(roleName))+"&roleType="+roleType;
```

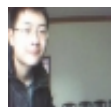
```
document.authorityForm.action = basePath3+"User_viewUser.do?
id="+id+"&roleName="+encodeURIComponent(encodeURIComponent(roleName))+"&roleType="+roleType;
```

JAVA后台：

```
1. roleName =
    java.net.URLDecoder.decode(getRequest().getParameter("roleName"),"UTF-8");
```

```
roleName =
    java.net.URLDecoder.decode(getRequest().getParameter("roleName"),
    "UTF-8");
```

分类：【.NET B/S】



v.e.n.u.s

关注 - 8

粉丝 - 96

+加关注

0

0

« 上一篇: mysql常用函数

» 下一篇: MSMQ消息队列 用法

posted on 2015-09-22 17:22 v.e.n.u.s 阅读(10109) 评论(0) 编辑 收藏

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) 网站首页。

最新**IT**新闻：

- 罗永浩：彪悍、情怀、理想主义 很多标签都与我无关
 - Facebook将推免费版企业协作聊天工具 用价格死磕Slack
 - 亚马逊今年视频内容投入将达45亿美元 缩小与Netflix差距
 - VR市场迟到的谷歌拼命追赶 挖来对手平台的开发牛人
 - 高盛说下一个风口在太空：20年内行业规模达数万亿美元
- » 更多新闻...

最新知识库文章：

- 技术文章的阅读姿势
 - 马拉松式学习与技术人员的成长性
 - 程序员的“认知失调”
 - 为什么有的人工作多年还是老样子
 - 也许，这样理解HTTPS更容易
- » 更多知识库文章...

Copyright @ v.e.n.u.s
Powered by: .Text and ASP.NET
Theme by: .NET Monster