**HTTP请求报文**是ISO-8864-1编码的，URL或消息体中含中文则是把其按对应编码转成16进制数，以%开头，空格用+

[**http请求的编码和解码问题**](http://wade6.iteye.com/blog/1744463)

**博客分类：**

* [web开发](http://wade6.iteye.com/category/191657)

本文在前人整理的基础上进行了二次加工，将一些问题追本溯源。

**一、字符编码**

计算机中的一切计算都是用二进制进行的，具体原因参考后面的链接；

早期操作计算机需要学会二进制，操作成本比较大；

为了更容易的操作计算机，专家们想将人们容易理解的数字、字母、常用符号等引入计算机，这就需要设置一个固定的二进制来代表这些字符；

而具体用哪个二进制表示哪个符号呢，当然每个人都可以约定自己的一套标准，这个标准就叫字符编码。

如果大家要想互相通讯而不造成混乱，那么大家就必须使用相同的编码规则；

于是最早美国有关的标准化组织就出台了现在的ASCII编码（美国标准信息交换码），统一规定了上述常用符号用哪个二进制数来表示。

**二、字符编码集**

各种符号（文字）与计算机二进制码之间对应关系的映射表

**三、常见字符集编码**

常见的字符编码有ASCII、GB2312、GBK、Unicode、UTF-8等。

**四、浏览器对请求的编码**

一般来说，URL只能使用英文字母、阿拉伯数字和某些标点符号（ASCII码范围），不能使用其他文字和符号。

但是URL 常常会包含 ASCII 码范围之外的字符，所以URL 必须转换为有效的 ASCII 格式才能正确使用。

URL使用"%"其后跟随两位的十六进制数来替换非 ASCII 字符。URL 不能包含空格，URL 编码通常使用 + 来替换空格。

但是这个跟在%后面的十六进制数是多少，是由编码时使用的编码方式决定的。

例如：

url：www.baidu.com/s?wd=春节；

因为URL中有汉字，所以“春节”这两个中文在转换后才能作为正确的URL使用；

但是在不同的字符集中，这两个字的编码不同，例如：

* “春”和“节”的utf-8编码分别是“E6 98 A5”和“E8 8A 82”，那么应该编码成www.baidu.com/s?wd=%E6%98%A5%E8%8A%82；
* “春”和“节”的GB2312编码分别是“B4 BA”和“BD DA”，那么应该编码成www.baidu.com/s?wd=%B4%BA%BD%DA；

问题是，RFC1738（规定URL格式的标准）没有规定这种情况的具体的编码方法，而是交给应用程序（浏览器）自己决定。

这导致不同浏览器的“URL编码”成为了一个混乱的领域，下面来看一下。

注意，下面讨论的编码对象是ASCII码范围以外的字符。ASCII码范围的字符不需要编码。

**1、地址栏中的Get请求的url**

* 在中文环境中，IE默认对URL的Path部分会采用UTF-8来进行编码，参数部分采用GBK来编码；
* FireFox对URL的Path和参数都采用GBK进行编码。

**2、页面中链接的Get请求url**

* 在中文环境中，IE对URL的Path部分都会采用UTF-8来进行编码，而参数部分会根据当前页面的编码确定；
* FireFox对URL的Path和参数都会采用当前页面的字符集来进行编码。

**3、浏览器做普通Post提交url**

* 采用当前页面的字符集进行编码（浏览器大多数时候会根据Http返回头中的Content-Type来确定当前页面的字符集，如果我们想要设定表单提交时所用的字符集编码方式，可以用设定Http返回头中Content-Type的值来实现）。

**4、浏览器做Ajax Get提交url**

* 根据浏览器的不同，编码时采用的字符集也是不同的；
* FireFox始终采用UTF-8；
* IE会根据本地浏览器的字符集配置来确定采用的字符集。

**5、浏览器做Ajax Post提交url**

* 始终会采用UTF-8进行字符集编码(它决定于XMLHttpRequest的实现)。

**6、总结**

由于不同浏览器对get请求的编码方式不同，所以web应用的访问路径及get请求的参数（直接在url中输入的情况下）基本都使用数字和字母表示，很少用中文等其他ASCII码范围之外的字符，以免出现乱码问题。

如果是返回页面中的get请求，服务端肯定会根据自己的情况进行合适的编码，无需关心中文等问题。

**五、web容器对请求的解码**

web服务器接收到客户端的请求后，会将其内容转给web容器来处理；

因为接到的请求path（url）是编码过的二进制流，所以在处理前会将其转换成ASCII码（应该是由web容器转换，这个不清楚，谁转都一样）；

但是请求中可能还有部分参数和消息体的数据是经过编码的（例如中文字符被编码），这里就涉及到对请求内容和参数进行解码的问题。

**1、解码对象**

根据对HTTP协议，GET请求只有头信息，POST请求包括头信息和消息体；

这里将解码的对象分为两个：头参数和消息体。

**2、容器的默认解码字符集**

对于这两个部分，大多数容器都默认以ISO-8859-1方式进行解码。

**3、自定义解码字符集**

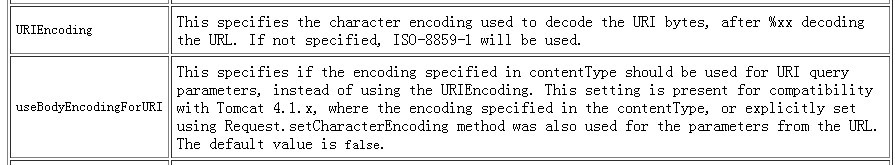
如果不想使用默认的字符集进行解码，web容器允许自定义解码的字符集；

**1）头参数解码设定**

以servlet容器为例，通过修改连接器配置来设定头参数的解码方式；

修改tomcat的server.xml文件，增加下面的属性就可以设置头参数的解码方式；

以下是Tomcat连接器中两个相关配置项的定义（Server.xml中的Connector配置项 http://tomcat.apache.org/tomcat-6.0-doc/config/http.html）



**2）消息体的解码设定**

在获取消息体参数前，可以通过调用request.setCharacterEncoding(charset)来设置消息体的解码字符集。

具体情况可参见[Servlet学习整理（四）—— ServletRequest和ServletResponse](http://wade6.iteye.com/blog/1736299)的“请求数据的编码”

**参考：**

http://blog.sina.com.cn/s/blog\_70313b0501015ae6.html  计算机为什么只识别二进制

http://ansjsun.iteye.com/blog/1477598   字符编码发展史

http://blog.sina.com.cn/s/blog\_812fb97901011e13.html  在Servlet技术中设置客户端和服务器端编码的问题