

关于 HTTP GET/POST 请求参数长度最大值的一个理解误区

分类: Java 2014-09-24 14:02:11

1. Get方法长度限制

Http Get方法提交的数据大小长度并没有限制，HTTP协议规范没有对URL长度进行限制。这个限制是特定的浏览器及服务器对它的限制。

如：IE对URL长度的限制是2083字节(2K+35)。

下面就是对各种浏览器和服务器的最大处理能力做一些说明。

Microsoft Internet Explorer (Browser)

IE浏览器对URL的最大限制为2083个字符，如果超过这个数字，提交按钮没有任何反应。

Firefox (Browser)

对于Firefox浏览器URL的长度限制为65,536个字符。

Safari (Browser)

URL最大长度限制为 80,000个字符。

Opera (Browser)

URL最大长度限制为190,000个字符。

Google (chrome)

URL最大长度限制为8182个字符。

Apache (Server)

能接受最大url长度为8,192个字符。

Microsoft Internet Information Server(IIS)

能接受最大url的长度为16,384个字符。

通过上面的数据可知，为了让所有的用户都能正常浏览，URL最好不要超过IE的最大长度限制(2083个字符)，当然，如果URL不直接提供给用户，而是提供给程序调用，这时的长度就只受Web服务器影响了。

注：对于中文的传递，最终会为urlencode后的编码形式进行传递，如果浏览器的编码为UTF8的话，一个汉字最终编码后的字符长度为9个字符。

因此如果使用的 GET 方法，最大长度等于URL最大长度减去实际路径中的字符数。

2. POST方法长度限制

理论上讲，POST是没有大小限制的。HTTP协议规范也没有进行大小限制，起限制作用的是服务器的处理程序的处理能力。

如：在Tomcat下取消POST大小的限制（Tomcat默认2M）；

打开tomcat目录下的conf目录，打开server.xml 文件，修改

```
debug="0"
acceptCount="100"
connectionTimeout="20000"
disableUploadTimeout="true"
port="8080"
redirectPort="8443"
enableLookups="false"
minSpareThreads="25"
maxSpareThreads="75"
maxThreads="150"
maxPostSize="0"
URIEncoding="GBK"
>
```

增加红色字体部分 `maxPostSize="0"` (设为0是取消POST的大小限制)

刚看到群里又有同学在说 HTTP 协议下的 Get 请求参数长度是有大小限制的, 最大不能超过 XX, 而 Post 是无限制的, 看到这里, 我想他们定是看多了一些以讹传讹的博客或者书籍, 导致一种理解上的误区:

- 1、首先即使有长度限制, 也是限制的是整个 URI 长度, 而不仅仅是你的参数值数据长度。
- 2、HTTP 协议从未规定 GET/POST 的请求长度限制是多少。

The HTTP protocol does not place any a priori limit on the length of a URI. Servers MUST be able to handle the URI of any resource they serve, and SHOULD be able to handle URIs of unbounded length if they provide GET-based forms that could generate such URIs. A server SHOULD return 414 (Request-URI Too Long) status if a URI is longer than the server can handle (see section 10.4.15).

Note: Servers ought to be cautious about depending on URI lengths above 255 bytes, because some older client or proxy implementations might not properly support these lengths.

- 3、所谓的请求长度限制是由浏览器和 web 服务器决定和设置的, 各种浏览器和 web 服务器的设定均不一样, 这依赖于各个浏览器厂家的规定或者可以根据 web 服务器的处理能力来设定。

The limit is in MSIE and Safari about 2KB, in Opera about 4KB and in Firefox about 8KB, (255 bytes if we count very old browsers) . We may thus assume that 8KB is the maximum possible length and that 2KB is a more affordable length to rely on at the server side and that 255 bytes is the safest length to assume that the entire URL will come in.

If the limit is exceeded in either the browser or the server, most will just truncate the characters outside the limit without any warning. Some servers however may send a HTTP 414 error. If you need to send large data, then better use POST instead of GET. Its limit is much higher, but more dependent on the server used than the client. Usually up to around 2GB is allowed by the average webserver. This is also configurable somewhere in the server settings. The average server will display a server-specific error/exception when the POST limit is exceeded, usually as HTTP 500 error.

HTTP 1.1 defines Status Code 414 Request-URI Too Long for the cases where a server-defined limit is reached. You can see further details on RFC 2616. For the case of client-defined limits, there is no sense on the server returning something, because the server won't receive the request at all.

The server is refusing to service the request because the Request-URI is longer than the server is willing to interpret. This rare condition is only likely to occur when a client has improperly converted a POST request to a GET request with long query information, when the client has descended into a URI "black hole" of redirection (e.g., a redirected URI prefix that points to a suffix of itself), or when the server is under attack by a client attempting to exploit security holes present in some servers using fixed-length buffers for reading or manipulating the Request-URI.

附 GET VS POST:

1、多数浏览器对于POST采用两阶段发送数据的，先发送请求头，再发送请求体，即使参数再少再短，也会被分成两个步骤来发送（相对于GET），也就是第一步发送header数据，第二步再发送body部分。HTTP是应用层的协议，而在传输层有些情况TCP会出现两次连结的过程，HTTP协议本身不保存状态信息，一次请求一次响应。对于TCP而言，通信次数越多反而靠性越低，能在一次连结中传输完需要的消息是最可靠的，尽量使用GET请求来减少网络耗时。如果通信时间增加，这段时间客户端与服务器端一直保持连接状态，在服务器侧负载可能会增加，可靠性会下降。

Tips: 关于这点你可以参考: Yahoo网站性能优化指南之服务器篇

<http://segmentfault.com/a/1190000000353790>

<http://developer.yahoo.com/performance/rules.html>

<http://blogread.cn/it/article/6100?f=wb> YSLOW法则中, 为什么yahoo推荐用GET代替POST?

上面这篇文章介绍了 wireshark 抓包验证 post 两次发包, get 一次发包的全过程, 推荐阅读。

2、GET请求能够被cache，GET请求能够被保存在浏览器的浏览历史里面（密码等重要数据GET提交，别人查看历史记录，就可以直接看到这些私密数据）POST不进行缓存。

3、GET参数是带在URL后面，传统IE中URL的最大可用长度为2048字符，其他浏览器对URL长度限制实现上有所不同。POST请求无长度限制（目前理论上是这样的）。

4、GET提交的数据大小，不同浏览器的限制不同，一般在2k-8K之间，POST提交数据比较大，大小靠服务器的设定值限制，而且某些数据只能用 POST 方法「携带」，比如 file。

5、全部用POST不是十分合理，最好先把请求按功能和场景分下类，对数据请求频繁，数据不敏感且数据量在普通浏览器最小限定的2k范围内，这样的情况使用GET。其他地方使用POST。

6、GET 的本质是「得」，而 POST 的本质是「给」。而且，GET 是「幂等」的，在这一点上，GET 被认为是「安全的」。但实际上 server 端也可以用作资源更新，但是这种用法违反了约定，容易造成 CSRF（跨站请求伪造）。

REF:

maximum length of HTTP GET request?

<http://stackoverflow.com/questions/2659952/maximum-length-of-http-get-request>

<http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.15> Request-URI Too Long

<http://www.w3.org/Protocols/rfc2616/rfc2616-sec3.html#sec3.2.1> General Syntax

<http://www.cnblogs.com/xiaotaomaomao/articles/986070.html>

<http://www.cnblogs.com/TankXiao/archive/2012/02/13/2342672.html> HTTP协议详解

post方式相比get安全,携带数据更大,我准备所有数据都用post方式获取,这样好吗?

<http://segmentfault.com/q/1010000000213082>

<http://www.cnblogs.com/hydddd/archive/2009/04/09/1432744.html> 浅谈CSRF攻击方式