HTTP有两部分组成：请求与响应，下面分别整理。

**一．HTTP请求**

**1．HTTP请求格式：**

<request line>

<headers>

<blank line>

[<request-body>]

在HTTP请求中，第一行必须是一个请求行（request line），用来说明请求类型、要访问的资源以及使用的HTTP版本。紧接着是一个首部（header）小节，用来说明服务器要使用的附加信息。在首部之后是一个空行，再此之后可以添加任意的其他数据[称之为主体（body）]。

**2．GET与POST区别**

HTTP 定义了与服务器交互的不同方法，最基本的方法是 GET 和 POST（Ajax开发，关心的只有GET请求和POST请求）。

**GET与POST方法有以下区别：**

（1）   在客户端，Get方式在通过URL提交数据，数据在URL中可以看到；POST方式，数据放置在HTML HEADER内提交。

（2）   GET方式提交的数据最多只能有1024字节，而POST则没有此限制。

（3）   安全性问题。正如在（1）中提到，使用 Get 的时候，参数会显示在地址栏上，而 Post 不会。所以，如果这些数据是中文数据而且是非敏感数据，那么使用 get；如果用户输入的数据不是中文字符而且包含敏感数据，那么还是使用 post为好。

（4）   安全的和幂等的。所谓安全的意味着该操作用于获取信息而非修改信息。幂等的意味着对同一 URL 的多个请求应该返回同样的结果。完整的定义并不像看起来那样严格。换句话说，GET 请求一般不应产生副作用。从根本上讲，其目标是当用户打开一个链接时，她可以确信从自身的角度来看没有改变资源。比如，新闻站点的头版不断更新。虽然第二次请求会返回不同的一批新闻，该操作仍然被认为是安全的和幂等的，因为它总是返回当前的新闻。反之亦然。POST 请求就不那么轻松了。POST 表示可能改变服务器上的资源的请求。仍然以新闻站点为例，读者对文章的注解应该通过 POST 请求实现，因为在注解提交之后站点已经不同了（比方说文章下面出现一条注解）。  
<http://www.cnblogs.com/stu-acer/archive/2006/08/28/488802.html>

**GET与POST方法实例：**

|  |  |
| --- | --- |
| GET实例 | POST实例 |
| GET /books/?name=Professional%20Ajax HTTP/1.1  Host: www.wrox.com  User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.6)  Gecko/20050225 Firefox/1.0.1  Connection: Keep-Alive | POST / HTTP/1.1  Host: www.wrox.com  User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.6)  Gecko/20050225 Firefox/1.0.1  Content-Type: application/x-www-form-urlencoded  Content-Length: 40  Connection: Keep-Alive       （此处空一行）  name=Professional%20Ajax&publisher=Wiley |

**3．**表单提交中get和post方式的区别归纳如下几点：

（1）get是从服务器上获取数据，post是向服务器传送数据。

（2）对于表单的提交方式，在服务器端只能用Request.QueryString来获取Get方式提交来的数据，用Post方式提交的数据只能用Request.Form来获取。

（3）一般来说，尽量避免使用Get方式提交表单，因为有可能会导致安全问题。比如说在登陆表单中用Get方式，用户输入的用户名和密码将在地址栏中暴露无遗。但是在分页程序中，用Get方式就比用Post好。

**二．HTTP响应**

**1．HTTP响应格式：**

<status line>

<headers>

<blank line>

[<response-body>]

在响应中唯一真正的区别在于第一行中用状态信息代替了请求信息。状态行（status line）通过提供一个状态码来说明所请求的资源情况。   
         
       HTTP响应实例：

|  |
| --- |
| HTTP/1.1 200 OK  Date: Sat, 31 Dec 2005 23:59:59 GMT  Content-Type: text/html;charset=ISO-8859-1  Content-Length: 122  ＜html＞  ＜head＞  ＜title＞Wrox Homepage＜/title＞  ＜/head＞  ＜body＞  ＜!-- body goes here --＞  ＜/body＞  ＜/html＞ |

**2．最常用的状态码有：**

◆200 (OK): 找到了该资源，并且一切正常。

◆304 (NOT MODIFIED): 该资源在上次请求之后没有任何修改。这通常用于浏览器的缓存机制。

◆401 (UNAUTHORIZED): 客户端无权访问该资源。这通常会使得浏览器要求用户输入用户名和密码，以登录到服务器。

◆403 (FORBIDDEN): 客户端未能获得授权。这通常是在401之后输入了不正确的用户名或密码。

◆404 (NOT FOUND): 在指定的位置不存在所申请的资源。

**POST请求**

发表于[2012-08-08](http://www.w3cmm.com/ajax/post.html)由[admin](http://www.w3cmm.com/author/admin)使用频率仅次于GET的是POST请求，通常用于向服务发送应该被保存的数据。POST请求应该把数据作为请求的主题提交，而GET请 求传统上不是这样。POST请求的主题可以包含非常多的数据，而且格式不限。在open()方法第一个参数的位置传入”post”，就可以初始化一个 POST请求，如下面的例子所示：

xhr.open("post", "example.php", ture);

发送Post请求的第二步就是向send()方法中传入某些数据。有雨XHR最初的设计主要是为了处理XML,因此可以在此传入任何想发送到服务器的字符串。

默认情况下，服务器对POST请求和提交Web表单的请求并不会一视同仁。因此，服务器端必须有程序来读取发送过来的原始数据，并从中解析出有用的 部分。不过，我们可以使用XHR来模仿表单提交：首先将Content-Type头部信息设置为application/x-www.form- urlencoded，也就是表单提交时的内容类型，其次是以适当的格式创建一个字符串。POST数据的格式与查询字符串格式相同。如果需要将页面中表单 的数据进行序列化，然后再通过XHR发送到服务器，那么就可以使用[JavaScript表单序列化](http://www.w3cmm.com/javascript/serialize-form.html)文章中的serialize()函数来创建这个字符串：

function submitData() {

var xhr = createXHR();

hr.onreadystatechange = function () {

if (xhr.readyState == 4) {

if ((xhr.status >= 200 && xhr.status < 300) || xhr.status == 304) {

alert(xhr.responseText);

} else {

alert("Request was unsuccessful: " + xhr.status);

} } };

xhr.open("post", "postexample.php", true); xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");

var form = document.getElementById("user-info"); xhr.send(serialize(form));};

这个函数可以将ID为”user-info”的表单中的数据序列化之后发送给服务器。如下面的实例PHP文件postexample.php就可以通过$\_POST取得提交的数据了：

<?php

header("Content-Type: text/plain");

echo <<<EOF

Name: {$\_POST['user-name']}Email: {$\_POST['user-email']}EOF;?>

表单提交中get和post方式的区别有5点  
1.get是从服务器上获取数据，post是向服务器传送数据。  
2.get 是把参数数据队列加到提交表单的ACTION属性所指的URL中，值和表单内各个字段一一对应，在URL中可以看到。post是通过HTTPpost机 制，将表单内各个字段与其内容放置在HTML HEADER内一起传送到ACTION属性所指的URL地址。用户看不到这个过程。  
3.对于get方式，服务器端用Request.QueryString获取变量的值，对于post方式，服务器端用Request.Form获取提交的数据。  
4.get传送的数据量较小，不能大于2KB。post传送的数据量较大，一般被默认为不受限制。但理论上，IIS4中最大量为80KB，IIS5中为100KB。  
5.get安全性非常低，post安全性较高。

**post和get请求方式的区别**

HTTP请求：get与post方法的区别  
HTTP 定义了与服务器交互的不同方法，最基本的方法是 get 和 post。事实上 get 适用于多数请求，而保留 post仅用于更新站点。根据 HTTP 规范，get 用于信息获取，而且应该是安全的和幂等的。所谓安全的意味着该操作用于获取信息而非修改信息。换句话说，get 请求一般不应产生副作用。幂等的意味着对同一 URL的多个请求应该返回同样的结果。完整的定义并不像看起来那样严格。从根本上讲，其目标是当用户打开一个链接时，她可以确信从自身的角度来看没有改变 资源。比如，新闻站点的头版不断更新。虽然第二次请求会返回不同的一批新闻，该操作仍然被认为是安全的和幂等的，因为它总是返回当前的新闻。反之亦然。 post请求就不那么轻松了。post 表示可能改变服务器上的资源的请求。仍然以新闻站点为例，读者对文章的注解应该通过 post请求实现，因为在注解提交之后站点已经不同了（比方说文章下面出现一条注解）；  
在FORM提交的时候，如果不指定Method，则默认 为get请求，Form中提交的数据将会附加在url之后，以?分开与url分开。字母数字字符原样发送，但空格转换为“+“号，其它符号转换为%XX, 其中XX为该符号以16进制表示的ASCII（或ISOLatin-1）值。get请求请提交的数据放置在HTTP请求协议头中，而post提交的数据则 放在实体数据中；  
get方式提交的数据最多只能有1024字节，而post则没有此限制。  
在表单里使用”post”和”get”有什么区别  
在Form里面，可以使用post也可以使用get。它们都是method的合法取值。但是，post和get方法在使用上至少有两点不同：  
1、get方法通过URL请求来传递用户的输入。post方法通过另外的形式。  
2、get方式的提交你需要用Request.QueryString来取得变量的值，而post方式提交时，你必须通过Request.Form来访问提交的内容。  
仔细研究下面的代码。你可以运行之来感受一下：  
代码  
〈!–两个Form只有Method属性不同–〉  
〈FORM ACTION=“getpost.[asp](http://www.knowsky.com/asp.asp)” METHOD=“get”?  
〈INPUT TYPE=“text” NAME=“Text” VALUE=“Hello World”〉〈/INPUT〉  
〈INPUT TYPE=“submit” VALUE=“Method=get”〉〈/INPUT〉  
〈/FORM〉  
〈BR〉  
〈FORM ACTION=“getpost.asp” METHOD=“post”〉  
〈INPUT TYPE=“text” NAME=“Text” VALUE=“Hello World”〉〈/INPUT〉  
〈INPUT TYPE=“submit” VALUE=“Method=post”〉〈/INPUT〉  
〈/FORM〉  
〈BR〉  
〈BR〉  
〈% If Request.QueryString(“Text”) 〈〉 ““ Then %〉  
通过get方法传递来的字符串是： “〈B〉〈%= Request.QueryString(“Text”) %〉〈/B〉“〈BR〉  
〈% End If %〉  
〈% If Request.Form(“Text”) 〈〉 ““ Then %〉  
通过post方法传递来的字符串是： “〈B〉〈%= Request.Form(“Text”) %〉〈/B〉“〈BR〉  
〈% End If %〉  
说明  
把上面的代码保存为getpost.asp，然后运行，首先测试post方法，这时候，浏览器的url并没有什么变化，返回的结果是：  
通过post方法传递来的字符串是： "Hello World"  
然后测试用get方法提交，请注意，浏览器的url变成了：  
<http://localhost/general/form/getpost.asp?Text=Hello+World>  
而返回的结果是：  
通过get方法传递来的字符串是： "Hello World"  
最后再通过post方法提交，浏览器的url还是：  
<http://localhost/general/form/getpost.asp?Text=Hello+World>  
而返回的结果变成：  
通过get方法传递来的字符串是： "Hello World"  
通过post方法传递来的字符串是： "Hello World"  
提示  
通过get方法提交数据，可能会带来安全性的问题。比如一个登陆页面。当通过get方法提交数据时，用户名和密码将出现在URL上。如果：  
１、 登陆页面可以被浏览器缓存；  
２、 其他人可以访问客户的这台机器。  
那么，别人即可以从浏览器的历史记录中，读取到此客户的账号和密码。所以，在某些情况下，get方法会带来严重的安全性问题。  
建议  
<http://www.devdao.com/>  
在Form中，建议使用post方法。  
get与post的区别2  
get：是以实体的方式得到由请求URI所指定资源的信息，如果请求URI只是一个数据产生过程，那么最终要在响应实体中返回的是处理过程的结果所指向的资源，而不是处理过程的描述。  
post：用来向目的服务器发出请求，要求它接受被附在请求后的实体，并把它当作请求队列中请求URI所指定资源的附加新子项，post被设计成用统一的方法实现下列功能：  
1：对现有资源的解释  
2：向电子公告栏、新闻组、邮件列表或类似讨论组发信息。  
3：提交数据块  
4：通过附加操作来扩展[数据库](http://www.knowsky.com/sql.asp)  
从上面描述可以看出，get是向服务器发索取数据的一种请求；而post是向服务器提交数据的一种请求，要提交的数据位于信息头后面的实体中。  
很理论化，但是很标准，method=“get”并不是从服务器上获取数据，get和post 只是发送机制不同，并不是一个取一个发！  
get方法会在IE地址栏里显示表示你提交时候所带的值；post方法不会  
1、 get是把参数数据队列加到提交表单的ACTION属性所指的URL中，值和表单内各个字段一一对应，在URL中可以看到。post是通过 HTTPpost机制，将表单内各个字段与其内容放置在HTML HEADER内一起传送到ACTION属性所指的URL地址。用户看不到这个过程。  
2、对于get方式，服务器端用Request.QueryString获取变量的值，对于post方式，服务器端用Request.Form获取提交的数据。两种方式的参数都可以用Request来获得。  
3、get传送的数据量较小，不能大于2KB。post传送的数据量较大，一般被默认为不受限制。但理论上，IIS4中最大量为80KB，IIS5中为100KB。  
4、get安全性非常低，post安全性较高。  
5、 〈form method="get" action="a.asp?b=b"〉跟〈form method="get"action="a.asp"〉是一样的，也就是说，action页面后边带的参数列表会被忽视；而 〈formmethod="post" action="a.asp?b=b"〉跟〈form method="post"action="a.asp"〉是不一样的。  
另外，get请求有如下特性：它会将数据添加到URL中，通过这种方式传递到服务器，通常利用一个问号？代表URL地址的结尾与数据参数的开端，后面的参数每一个数据参数以“名称=值”的形式出现，参数与参数之间利用一个连接符&来区分。  
post请求有如下特性：数据是放在HTTP主体中的，其组织方式不只一种，有&连接方式，也有分割符方式，可隐藏参数，传递大批数据，比较方便。  
post 地址栏不会出现一大串?bjnghfgreygt这样的东西  
如果是get，就会出现了  
1、get 方法通过 URL 请求来传递用户的数据，将表单内各字段名称与其内容，以成对的字符串连接，置于 action 属性所指程序的 url后，如[http://www.mdm.com/test.asp?name=asd&pass](http://www.mdm.com/test.asp?name=asd&password=sad)[Word](http://www.knowsky.com/article.asp?typeid=117)=sad， 数据都会直接显示在 url 上，就像用户点击一个链接一样；post 方法通过 HTTP post 机制，将表单内各字段名称与其内容放置在 HTML表头(header)内一起传送给服务器端交由 action属性能所指的程序处理，该程序会通过标准输入(stdin)方式，将表单的数据读出并加以处理  
2、 get 方式需要使用 Request.QueryString 来取得变量的值；而 post 方式通过 Request.Form 来访问提交的内容  
3、 get 方式传输的数据量非常小，一般限制在 2 KB 左右，但是执行效率却比 post 方法好；而 post方式传递的数据量相对较大，它是等待服务器来读取数据，不过也有字节限制，这是为了避免对服务器用大量数据进行恶意攻击，根据微软方面的说法，微 软对用 Request.Form() 可接收的最大数据有限制，IIS 4 中为 80 KB 字节，IIS 5 中为 100 KB 字节  
建议：除非你肯定你提交的数据可以一次性提交，否则请尽量用 post 方法  
4、 get 方式提交数据，会带来安全问题，比如一个登陆页面，通过 get 方式提交数据时，用户名和密码将出现在 URL上，如果页面可以被缓存或者其他人可以访问客户这台机器，就可以从历史记录获得该用户的帐号和密码，所以表单提交建议使用 post 方法；post方法提交的表单页面常见的问题是，该页面如果刷新的时候，会弹出一个对话框  
1、get是把参数数据队列加到提交表单的 ACTION属性所指的URL中，值和表单内各个字段一一对应，在URL中可以看到。post是通过HTTPpost机制，将表单内各个字段与其内容放置 在HTML HEADER内一起传送到ACTION属性所指的URL地址。用户看不到这个过程。  
2、 对于get方式，服务器端用Request.QueryString获取变量的值，对于post方式，服务器端用Request.Form获取提交的数据。两种方式的参数都可以用Request来获得。  
3、get传送的数据量较小，不能大于2KB。post传送的数据量较大，一般被默认为不受限制。但理论上，IIS4中最大量为80KB，IIS5中为100KB。  
4、get安全性非常低，post安全性较高。  
5、 〈form method="get" action="a.asp?b=b"〉跟〈form method="get"action="a.asp"〉是一样的，也就是说，action页面后边带的参数列表会被忽视；而 〈formmethod="post" action="a.asp?b=b"〉跟〈form method="post"action="a.asp"〉是不一样的。