



链滴

HTTP 协议请求方法和状态码 & HTTP 协议 Header 相关

作者: [someone38063](#)

原文链接: <https://ld246.com/article/1558354835946>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

URL 格式

`schema://host[:port#]/path.../[?query-string][#anchor]`

-

-

schema 指低层使用的协议（例如 http, https, ftp）

-
-

host HTTP 服务器的 IP 地址或者 Domain name

-
-

port# HTTP 服务器的端口默认是 80，可省略，如果使用其他端口需要指明

-
-

path 访问资源的路径

-
-

query-string 发送给 HTTP 服务器的数据

-
-

anchor 锚

-

URL 中的锚点

-

-

是一种超链接，只是他是页面内部的超链接

-
-

假如有一个网页很长，里面的内容可以分为 N 个部分，这样的话，就可以在网页的顶部设置一锚点，以便浏览者单击相应的锚点，快速到达本页内相应的位置，可以不必在很长的网页中自行寻找

-
-

锚点在 URL 的最右边，前面有一个字符"#"

-

HTTP 请求方法

-

-

GET 请求指定的页面信息并返回实体主体

-
-

GET 是最常见的方法，用于获取资源，常用于服务器查询某些信息，打开网页一般都是 GET 方，因为要从 Web 服务器获取信息

-
-

带参数的 GET 方法

-
-

浏览器也可以在 GET 方法中把数据传给服务器，数据放在 URL 的 '?' 后面

-
-

将查询字符串追加到 URL 末尾，一边将信息发送给服务器，这种方式叫 **查询字符串** 或 **Query String**

<p>例如在百度中搜索某 URL:

<code>https://.baidu.com/swd=abc&rsv_spt=1&rsv_iqid=0x99d541420020da12∓issp=1&f=8&rsv_bp=1&rsv_idx=2&ie=utf8&rqlang=cn&tnbaiduhome_pg&rsv_enter=1&oq=sssss&inputT=388&rsv_t=b2a8td06KxF5SxzU1ebZevafl6twuMdpqDWeQzErgjzdTQaXQmWpuclHX%2F%2BGcTlkZmM&rsv_su3=10&rsv_sug1=4&rsv_sug7=100&rsv_pq=8459df0b0009d665&rsv_sug=0&rsv_sug4=7296</code></p>

<p>查询字符串以"名=值"这样的形式出现, 多个名值之间用字符"&"隔开</p>

<p>在 Fiddler 中的 WebForms 选项卡可以更清楚地看到 GET 方法中的查询字符串参数</p>

<p>POST 向指定资源提交数据并进行处理请求(例如提交表单或上传文件), 数据被包含在请求体中。OST 请求可能会导致新的资源的建立和/或对已有资源的修改</p>

<p>POST 方法通常用来把表单中填好的数据发送给服务器</p>

<p>使用 WebForms 选项卡可以更清楚地看到 Body 主体里面的内容</p>

<p>HEAD 类似 GET 请求, 只不过返回的响应中没有具体内容, 用于获取报文头</p>

<p>PUT 从客户端向服务器传送的数据取代指定文档的内容</p>

<p>DELECT 请求服务器删除指定的内容</p>

<h2 id="GET和POST方法的区别">GET 和 POST 方法的区别</h2>
<p>(1)GET 提交的数据会放在 URL 后, 以'?'分割 URL 和传输数据, 参数之间以'&'相连, 例如
<code>EditPosts.aspx?name=test1&id=123456</code>, POST 方法是把提交的数据放在 HTTP 包的 Body 中</p>
<p>(2)GET 提交的数据大小有限制(因为浏览器对 URL 的长度有限制), 而 POST 方法提交的数据没有限制</p>
<p>(3)GET 方法需要 Reuquest.QueryString 来取得变量的值, 而 POST 方法通过 Request.Form 来获取变量的值</p>
<p>(4)GET 方法提交数据会带来安全问题, 比如登陆一个页面通过 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这台机器, 就可以从历史记录中获取用户名的账号和密码</p>
<h2 id="HTTP状态码">HTTP 状态码</h2>

<p>每个 HTTP Response 报文都会携带一个状态码, 用于告诉客户端请求是否成功, 状态码是一个 3 位数字的代码</p>

HTTP 状态码存在 HTTP 中的响应报文中，其作用是 Web 服务器用来告诉客户端发生了什么事

HTTP 响应报文中的第一行，由 HTTP 协议版本号，状态码，状态消息 3 部分组成

状态码分类

状态码	已定义范围	分类
1XX	100~101	信息提示，表示请求已被成功接收，继续处理
2XX	200~206	成功，表示请求已被成功接收，理解，接受
3XX	300~305	重定向，要完成请求，必须进行更进一步的处理
4XX	400~415	客户端错误，请求有语法错误或请求无法实现
5XX	500~505	服务器错误，服务器未能实现合法的请求

常见的状态码

名称	释义
----	----

<td>200</td>	<td>OK: 服务器成功处理了请求</td>
</tr>	
<tr>	
<td>301/302</td>	<td>Move Permanently(重定向): 请求的 URL 已移走。Response 中应该包
</tr>	一个 Location URL, 说明资源现在所处的位置</td>
<tr>	
<td>304</td>	<td>Not Modified(未修改): 客户端的缓存资源是最新的, 需要客户端使用缓
</tr>	
<tr>	
<td>404</td>	<td>Not Found: 未找到资源</td>
</tr>	
<tr>	
<td>401</td>	<td>禁止访问</td>
</tr>	
<tr>	
<td>501</td>	<td>Internal Server Error: 服务器遇到一个错误, 使其无法对请求提供服务</
</tr>	d>
</tbody>	
</table>	
	
	
<p>200(OK)</p>	
	
	
<p>最常见的成功响应状态代码, 它表明该请求成功地完成, 所请求的资源成功的发送回客户端</p>	
	
	
<p>204(No Content,没有内容)</p>	
	
	
<p>返回的 HTTP 响应中只有一些 Header 和状态行, 没有 Response Body</p>	
	
	
<p>作用: </p>	
	
	
<p>再不获取资源的情况下了解资源的情况(比如判断其类型)</p>	
	
	
<p>通过查看 HTTP 响应中的状态码查看某个对象是否存在</p>	
	
	
<p>通过查看 Header 测试资源是否被修改</p>	
	
	

<p>(浏览器访问 ditu.google.cn, 会捕获到很多 204)</p>

<p>206(Partial Content, 部分内容)</p>

<p>206 状态码代表服务器已经成功处理了部分 GET 请求(只有发送 GET 方法的 HTTP 请求, Web 服务器才可能返回 206)</p>

<p>206 的应用场景如下</p>

<p>FlashGet、迅雷或 HTTP 下载工具都是使用 206 状态码来实现断点续传的</p>

<p>将一个大文档分解为多个下载同时下载, 比如在线看视频(在线视频中, 边看边下载, 原理就是浏览器发送很多 Request, Web 服务器返回 206 状态码来实现的)</p>

<p>例如 GET 方法的 HTTP 请求, Header 中包含 Range: bytes=5303296-5336063(意思就是请得到 5303296~5336063 之间的数据)</p>

<p>301(Moved Permanently)</p>

<p>表示请求的网页已经永久性的转移到另一个地址</p>

<p>使用情况</p>

<p>防止用户数错域名: 比如 Google 担心用户输错域名, 就买下其他类似域名比如 go0gle.com, 后重定向到 www.google.com</p>

<p>网站更换域名, 比如京东以前的域名是 www.360buy.com, 现在域名是 www.jd.com</p>

<p>有多个权重不错的域名, 需要把所有的权重都传递到新域名上, 这就需要 301 重定向。如果不置 301, 把多个域名绑定在一个主机头上, 会被搜索引擎认为是两个相同的站点, 不利于网站的排名。帮规定的域名越多, 内容重复度越高, 排名越低。</p>

<p>302(Found)</p>

<p>当我们访问一个 URL 时, 服务器要我们访问另一个资源, 这时候浏览器会继续发送一个 HTTP 请求访问新的资源。</p>

<p>实例：在未登录状态下，直接访问需要登录才能访问的页面，会被服务器返回 302，跳转到登录面。</p>

<p>301 和 302 区别</p>

<p>在语法上一模一样，都是在 location 中返回新的 URL，区别在于：</p>

<p>301 表示旧地址的资源已经被永久移除了(资源不可访问)，搜索引擎会把权重算到新地址</p>

<p>302 表示旧地址资源还在(仍然可以访问)，这个重定向只是临时地从旧地址跳转到新地址，搜索引擎会把权重算到旧地址</p>

<p>304(Not Modified)</p>

<p>304 状态码表示上次的文档已经被缓存了，还可以继续使用，304 的响应是没有 Body 的</p>

<p>如果不想使用缓存，可以用 <code>Ctrl+F5</code> 强制刷新页面</p>

<p>400(Bad Request)</p>

<p>表示客户端请求有语法错误，发送的 HTTP 请求中的数据有错误(如表单有错误，Cookie 有错误。不能被服务器所理解</p>

<p>401(Unauthorized)</p>

<p>指未授权错误。有些网页采用的是 HTTP 基本认证(Basic Authentication),需要在 HTTP 请求中上 Authorization，否则服务器会返回状态码 401</p>

<p>403(Forbidden)</p>

<p>表示 Web 客户端发送的请求被 Web 服务器拒绝了。如果服务器向说明为什么拒绝请求，可以在 Body 中描述原因，但这个状态码通常表示服务器不想说明拒绝原因</p>

<p>404(Not Found)</p>

<p>当你访问一个 URL，这个 URL 的域名是正确的，但是资源不存在，服务器就会返回 404 状态码

告诉浏览器资源不存在(意味着输错了 URL)</p>

<p>500(Internal Server Error)</p>

<p>状态码 500 表示服务器内部错误，出现错误的原因很多：代码错误，数据库连接语句错误，程
内部抛出异常，空指针错误</p>

<p>503(Server Unavailable)</p>

<p>表示服务器暂时不可用。由于服务器维护或过载，服务器当时无法处理请求，这个状况是临时的
并在一段时间后恢复</p>

<h2 id="Fiddler中设置断点修改HTTP请求">Fiddler 中设置断点修改 HTTP 请求</h2>
<h2 id="全局断点">全局断点</h2>

<p>点击菜单栏中的 Rules > Automatic Breakpoint > Before Requests ,或者【F11】，这
方法会拦截所有的会话</p>

<p>取消全局断点，单击 Rules > Automatic Breakpoint > Disabled，或者【Shift + F11】
</p>

<h2 id="单个断点">单个断点</h2>

<p>在 Fiddler 左下角的 QuickExec 命令行中输入命令 <code>bpu www.baidu.com</code>,这
方法只会拦截 <a href="https://ld246.com/forward?goto=http%3A%2F%2Fwww.baidu.com" ta
get="_blank" rel="nofollow ugc">www.baidu.com </p>

<p>想要消除消除单个断点，在命令行中输入 <code>bpu</code> </p>

<h3 id="实例-Fiddler修改HTTP请求">实例，Fiddler 修改 HTTP 请求</h3>

浏览器想要访问 <a href="https://ld246.com/forward?goto=http%3A%2F%2Fwww.163.co
" target="_blank" rel="nofollow ugc">www.163.com，则通过 Fiddler 修改 HTTP 请求，
浏览器去访问 <a href="https://ld246.com/forward?goto=http%3A%2F%2Fwww.cnblogs.com"
arget="_blank" rel="nofollow ugc">www.cnblogs.com

<p>(1)在菜单栏单击 Rules > Automatic Breakpoint > Before Requests</p>
<p>(2)打开浏览器，输入 <code>www.163.com</code>，这时候你会发现任务栏的 Fiddler 图标
闪烁，这说明 Fiddler 拦截住了 HTTP 请求</p>
<p>(3)回到 Fiddler 界面，菜单栏单击单击 Rules > Automatic Breakpoint > Disabled(因
已经拦截住想要的 HTTP 请求，其他 HTTP 请求不需要拦截了)</p>
<p>(4)被拦截的 HTTP 请求有一个红色的 T 图标，选中所需要修改的 HTTP 请求，选中 “Inspector
” 面板，使用 “Raw” 选项卡修改</p>

(5)把 GET 'url' url 改为"[http://www.cnblogs.com/tankxiao](https://ld246.com/forward?goto=http%3A%2F%2Fwww.cnblogs.com%2Ftankxiao%2522)",同时把 HOST 改为"www.cnblogs.com",然后单击绿色的"Run to Completion"按钮放行</p>

(6)返回浏览器,此时会发现浏览器打开的是 cnblogs 的页面了,如果单击黄色按钮"Break on Response",则会继续拦截这个 HTTP 请求的响应</p>

Fiddler 设置断点修改 HTTP 响应</h2>

全局断点->全局断点</h2>

-

点击菜单栏中的 Rules > Automatic Breakpoint > After Response ,这种方法会中断所有的会话</p>

-
-

取消全局断点,单击 Rules > Automatic Breakpoint > Disabled, </p>

-

单个断点->单个断点</h2>

-

在命令行中输入"bpafter www.baidu.com",这种方法只会中断 [www.baidu.com](https://ld246.com/forward?goto=http%3A%2F%2Fwww.baidu.com)</p>

-
-

想要消除单个断点,可以在命令行中输入"bpafter"</p>

-

Fiddler 修改网页的标题</h2>

(1)启动 Fiddler,进行单个断点: "bpafter url"或全局断点"After Response"</p>

(2)打开浏览器输入'url'</p>

(3)在 Fiddler 选中该'url',在 Inspector 面板中选择 Raw 选项卡</p>

(4)在 HTML 代码中修改标签,然后单击"RUN to Completion"放行</p>

#伪造 Refere</p>

如果没有 Refere,网站会认为是盗链</p>

(1)启动 Fiddler,设置全局断点"Before Requests"</p>

(2)打开浏览器输入'url'</p>

(3)这 Fiddler 会拦截该 Request,在 Raw 选项卡中修改 HTTP 请求,添加"Refere:url"</p>

(4)单击绿色的"Run to Completion"</p>

#Web 页面简单的性能测试</p>

-

一个网页的加载速度跟父请求和子请求都是有关系的</p>

-
-

子请求出现了 404 或者 500 之类的错误,会严重影响整个网页的加载速度</p>

-
-

子请求的响应速度慢也会影响网页加载的速度</p>

-
-

Statistics 是一个详情和数据统计面板,显示每条 HTTP 请求的具体统计信息,在 Over Elapsed 中能看到 HTTP 响应返回所需要的响应的的时间</p>

-

