

HTTP协议

1.简介

Http协议定义了很多与服务器交互的方法，是一个无状态的请求/响应协议。最基本的有4种，分别是GET,POST,PUT,DELETE.

一个URL地址用于描述一个网络上的资源，而HTTP中的GET, POST, PUT, DELETE就对应着对这个资源的查，改，增，删4个操作。

我们最常见的就是GET和POST了。GET一般用于获取/查询资源信息，而POST一般用于更新资源信息.

2.特点

(1) HTTP是无连接：无连接的含义是限制每次连接只处理一个请求。服务器处理完客户的请求，并收到客户的应答后，即断开连接。采用这种方式可以节省传输时间。

(2) HTTP是媒体独立的：只要客户端和服务器知道如何处理的数据内容，任何类型的数据都可以通过HTTP发送。客户端以及服务器指定使用适合的MIME-type内容类型。

(3) HTTP是无状态：无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则它必须重传，这样可能导致每次连接传送的数据量增大。另一方面，在服务器不需要先前信息时它的应答就较快。

HTTP请求报文

HTTP请求报文由3部分组成（请求行+请求头+请求体）：



图 15-4 HTTP 请求报文

请求行：

①是请求方法，GET和POST是最常见的HTTP方法，除此以外还包括DELETE、HEAD、OPTIONS、PUT、TRACE。

②为请求对应的URL地址，它和报文头的Host属性组成完整的请求URL。

③是协议名称及版本号。

请求头：

④是HTTP的报文头，报文头包含若干个属性，格式为“属性名:属性值”，服务端据此获取客户端的信息。

与缓存相关的规则信息，均包含在header中

请求体：

⑤是报文体，它将一个页面表单中的组件值通过param1=value1¶m2=value2的键值对形式编码成一个格式化串，它承载多个请求参数的数据。不但报文体可以传递请求参数，请求URL也可以通过类似于“/chapter15/user.html?param1=value1¶m2=value2”的方式传递请求参数。

HTTP请求报文头属性

Accept

请求报文可通过一个“Accept”报文头属性告诉服务端 客户端接受什么类型的响应。

如下报文头相当于告诉服务端，客户端能够接受的响应类型仅为纯文本数据啊，你别发其它什么图片啊，视频啊过来，

1. Accept:text/plain

Accept属性的值可以为一个或多个MIME类型的值（描述消息内容类型的因特网标准，消息能包含文本、图像、音频、视频以及其他应用程序专用的数据）

cookie

客户端的Cookie就是通过这个报文头属性传给服务端的哦！如下所示：

```
1. Cookie: $Version=1; Skin=new;jsessionId=5F4771183629C9834F8382E23
```

服务端是怎么知道客户端的多个请求是隶属于一个Session呢？注意到后台的那个jsessionid = 5F4771183629C9834F8382E23木有？原来就是通过HTTP请求报文头的Cookie属性的jsessionid的值关联起来的！（当然也可以通过重写URL的方式将会话ID附带在每个URL的后面哦）。

Referer

表示这个请求是从哪个URL过来的，假如你通过google搜索出一个商家的广告页面，你对这个广告页面感兴趣，鼠标一点发送一个请求报文到商家的网站，这个请求报文的Referer报文头属性值就是http://www.google.com。

Cache-Control

对缓存进行控制，如一个请求希望响应返回的内容在客户端要被缓存一年，或不希望被缓存就可以通过这个报文头达到目的。

HTTP响应报文

HTTP的响应报文也由三部分组成（响应行+响应头+响应体）



响应行：

- ①报文协议及版本；
- ②状态码及状态描述；

响应头：

- ③响应报文头，也是由多个属性组成；

响应体：

- ④响应报文体，即我们真正要的“干货”

响应状态码

和请求报文相比，响应报文多了一个“响应状态码”，它以“清晰明确”的语言告诉客户端本次请求的处理结果。

HTTP的响应状态码由5段组成：

- 1xx 消息，一般是告诉客户端，请求已经收到了，正在处理，别急...
- 2xx 处理成功，一般表示：请求收悉、我明白你要的、请求已受理、已经处理完成等信息。
- 3xx 重定向到其它地方。它让客户端再发起一个请求以完成整个处理。
- 4xx 处理发生错误，责任在客户端，如客户端的请求一个不存在的资源，客户端未被授权，禁止访问等。

- 5xx 处理发生错误，责任在服务端，如服务端抛出异常，路由出错，HTTP版本不支持等。

以下是几个常见的状态码：

200 OK

你最希望看到的，即处理成功！

303 See Other

我把你redirect到其它的页面，目标的URL通过响应报文头的Location告诉你。

304 Not Modified

告诉客户端，你请求的这个资源至你上次取得后，并没有更改，你直接用你本地的缓存吧，我很忙哦，你能不能少来烦我啊！

404 Not Found

你最不希望看到的，即找不到页面。如你在google上找到一个页面，点击这个链接返回404，表示这个页面已经被网站删除了，google那边的记录只是美好的回忆。

500 Internal Server Error

看到这个错误，你就应该查查服务端的日志了，肯定抛出了一堆异常，别睡了，起来改BUG去吧！

◆**200 (OK):** 找到了该资源，并且一切正常。

◆**302/307：**临时重定向，指出请求的文档已被临时移动到别处，此文档的新的url在location响应头中给出

◆**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。

常见的HTTP响应报文头属性

Cache-Control

响应输出到客户端后，服务端通过该报文头属告诉客户端如何控制响应内容的缓存。

常见的取值有private、public、no-cache、max-age，no-store，默认为private。

private: 客户端可以缓存

public: 客户端和代理服务器都可缓存（前端的同学，可以认为public和private是一样的）

max-age=xxx: 缓存的内容将在 xxx 秒后失效

no-cache: 需要使用对比缓存来验证缓存数据

no-store: 所有内容都不会缓存

默认为private，缓存时间为31536000秒（365天）也就是说，在365天内再次请求这条数据，都会直接获取缓存数据库中的数据，直接使用。

ETag

一个代表响应服务端资源（如页面）版本的报文头属性，如果某个服务端资源发生了变化了，这个ETag就会相应发生变化。它是Cache-Control的有益补充，可以让客户端“更智能”地处理什么时候要从服务端取资源，什么时候可以直接从缓存中返回响应。

Location

我们在JSP中让页面Redirect到一个某个A页面中，其实是让客户端再发一个请求到A页面，这个需要Redirect到的A页面的URL，其实就是通过响应报文头的Location属性告知客户端的，如下的报文头属性，将使客户端redirect到iteye的首页中：

[javascript] [view plain copy](#)

```
1. Location: http://www.iteye.com
```

Set-Cookie

服务端可以设置客户端的Cookie，其原理就是通过这个响应报文头属性实现的：

[javascript] [view plain copy](#)

```
1. Set-Cookie: UserID=JohnDoe; Max-Age=3600; Version=1
```

cookie机制：

客户端请求服务器，如果服务器需要记录该用户状态，就使用response向客户端浏览器颁发一个Cookie。客户端浏览器会把Cookie保存起来。当浏览器再请求该网站时，浏览器把请求的网址连同该Cookie一同提交给服务器。服务器检查该Cookie，以此来辨认用户状态。服务器还可以根据需要修改Cookie的内容。

Cookie的maxAge决定着Cookie的有效期，单位为秒（Second）。Cookie中通过getMaxAge()方法与setMaxAge(int maxAge)方法来读写maxAge属性。

如果maxAge属性为正数，则表示该Cookie会在maxAge秒之后自动失效。

如果maxAge为负数，则表示该Cookie仅在本浏览器窗口以及本窗口打开的子窗口内有效，关闭窗口后该Cookie即失效。

如果maxAge为0，则表示删除该Cookie。

Cookie并不提供修改、删除操作。如果要修改某个Cookie，只需要新建一个同名的Cookie，添加到response中覆盖原来的Cookie。

如果要删除某个Cookie，只需要新建一个同名的Cookie，并将maxAge设置为0，并添加到response中覆盖原来的Cookie。

```
Cookie cookie = new Cookie("username","helloweenvsfei"); // 新建Cookie
```

```
cookie.setMaxAge(0);
```

负数

```
response.addCookie(cookie);
```

户端

```
// 设置生命周期为0，不能为
```

```
// 必须执行这一句 输出到客
```