链滴

Http 简单介绍

作者：goker

原文链接：https://ld246.com/article/1606804403522

来源网站：链滴

<h2 id="Http简单介绍">Http 简单介绍</h2>
<code>Http</code> 是无状态（无法记录用户操作、状态、信息记录，只能通过 <code>cook
e、session</code> 这些会话机制）。
<code>http常用端口：80,8080</code>，<code>https(s=TLS)常用端口：443</code>。
<h2 id="一--http版本">一. http 版本</h2>
<code>http2.0：</code> 建立 <code>tcp</code> 后，并行发送多次请求，并行返回多次
应。
<code>http1.1：</code> 长连接 <code>(keep-alive)</code>：仅一次 <code>tcp</code>
接，可以发起多次 <code>http</code> 请求。
<code>http1.0：</code> 短连接：一次 <code>tcp</code> 连接，一次 <code>http</code>
请求（相当于每次和你说话，都要打招呼，你好）。
<h2 id="二--Http工作原理">二. Http 工作原理</h2>
<code>浏览器请求-->浏览器缓存-->dns-->tcp连接-->http请求-->http响应-
>tcp断</code>
请求/响应交互模型
用户点击 <code>url</code> 的链接之后，浏览器和 <code>web</code> 服务执行以下动作:<br

(1). 浏览器分析链接中的 <code>url</code>
(2). 浏览器向 <code>dns请求</code> 解析域名为 <code>ip地址</code>
(3). <code>dns解析ip</code> 返回给浏览器
(4). 流量器和服务器建立 <code>tcp连接</code>（默认 <code>80端口</code>，三次握手）<b
>
(5). 浏览器发送请求：<code>GET /index.html</code>
(6). 服务器响应，将 <code>index.html</code> 发送给浏览器
(7). 关闭连接，<code>释放tcp</code> 连接（四次挥手）
(8). 浏览器显示 <code>index.html</code> 的内容
<h2 id="三--Http报文分析">三. Http 报文分析</h2>
<code>General</code>（基本）：
<code>Request URL: https://www.baidu.com/</code>
<code>Request Method: GET</code>
<code>Status Code: 200 OK</code> 状态码
<code>Remote Address: 180.101.49.12:44</code>
<h3 id="3-1--请求">3.1. 请求</h3>
<code>GET / HTTP/1.1</code>
<code>request header</code>（请求头）

<code>Accpet:text/html：</code> 请求类型
<code>Accpet-Encoding:gzip,deflate：</code> 压缩
<code>Accept-Language: zh-CN,zh：</code> 语言
<code>Cache-Control: no-cache：</code> 缓存控制
<code>Connection: keep-alive：</code> 长连接
<code>Host: www.baidu.com：</code> 请求域名
<code>Pragma: no-cache：</code> 没有缓存
<code>User-Agent: Mozilla/5.0 ：</code> 用户信息，浏览器、设备

<h3 id="3-2--响应">3.2. 响应</h3>
<code>HTTP/1.1 200 OK</code>
<code>response header</code>（响应头）
<code>status: 200：</code> 状态码

200：成功
301：永久重定向，跳转（浏览器会记住跳转的地址，下次直接跳转）
302：临时跳转
307：内部跳转

304：走本地缓存
400：客户端错误
401：认证错误
403：找不主页，默认返回 index.html，权限不足
404：找不到页面，路径不对
500：服务端错误
502：找不到后端主机
503：服务器过载
504：超时

<code>Accept-Ranges: bytes：</code> 类型大小
<code>Cache-Control: private：</code> 缓存控制，受保护
<code>Connection: keep-alive：</code> 长连接
<code>Content-Encoding: gzip：</code> 压缩
<code>Content-Type: text/html;charset=utf-8：</code> 返回内容类型
<code>Date: Fri, 06 Nov 2020 06:09:28 GMT：</code> 服务器时间，东八区
<code>Expires: Fri, 06 Nov 2020 06:08:48 GMT：</code> 浏览器缓存有效性持续时间
<code>Last-modified: Mon, 26 Oct 2020 15:53:28 GMT：</code> 上次修改时间，做缓存
<code>Server: BWS/1.1：</code> 使用 web 软件版本
<code>Referer：</code> 记录上一次过来的页面域名，我是从那个过来的。
<code>Location：</code> 重定向到那
内容主体 ……
<h2 id="四--补充内容-get-post区别---转发-重定向区别-">四. 补充内容(get/post 区别)、(转发/定向区别)</h2>
<h3 id="4-1--get-post区别">4.1. get/post 区别</h3>
(1). 请求的内容和作用不一样
<code>get</code> 用于请求指定的页面信息，并返回实体主体。
<code>post</code> 用于提交表单、上传文件，数据是包含在请求体中的。
(2). 请求的参数直接在浏览器地址栏中显示？
<code>get</code> 会，参数是通过 <code>url</code> 传递。
<code>post</code> 不会，参数是通过 <code>Request body</code>。
(3). 发送次数不一样
<code>get</code> 会产生一个数据包：浏览器会先 <code>http header</code> 和 <code>data</code> 一起发送出去，服务器响应 <code>200</code>（返回数据）。
<code>post</code> 会产生两个数据包(2 次)：两次发包的优点，可以保证数据的完整性；
浏览器会先发送 <code>header</code>，服务器响应 <code>100</code>(临时请求) <code>continue</code>，在发送 <code>data</code>，服务器响应 <code>200</code>（返回数据）。
(4). 存储的容量不一样
<code>get</code> 存储的参数不能超过 <code>2k</code>
<code>post</code> 没有限制
<h3 id="4-2--转发-重定向区别">4.2. 转发/重定向区别</h3>
(1). <code>dispatcher</code> 转发：能进入 <code>WEB-INF</code> 目录下，能共享数，不能跳转外网，浏览器地址栏不会发生改变。
(2). <code>redirect</code> 重定向：不能进入 <code>WEB-INF</code> 目录下，不能共享数，能跳转外网，浏览器地址会发生改变。