

25-cookie

▼ 基本使用

- 服务器响应 HTTP 请求时返回的头部中包含的信息
HTTP/1.1 200 OK
Content-type: text/html
Set-Cookie: name=value
Other-header: other-header-value
- 之后的相关(特定域)请求都会在头部带上服务器返回的 cookie 信息
GET /index.js1 http/1.1
Cookie: name=value
Other-header: other-header-value

▼ 限制

- cookie 与特定域绑定 —— cookie 会与请求一起发送到创建它的域
- 总共不超过 300 个 cookie
- 每个 cookie 不超过 4096 字节 (4M)
- 每个域不超过20个 cookie
- 每个域不超过 89120 字节 (80M)

▼ 组成

▼ 名称

- 必须经过 URL 编码
- 不区分大小写
- 最佳实践：区分大小写，因为一些服务器软件可能这样对待它们

▼ 值

- 必须经过 URL 编码
- 字符串值

▼ 域

- 默认为设置 cookie 的域
- ▼ 不包含子域
 - www.wrox.com
- ▼ 包含子域
 - .wrox.com 表示对 wrox.com 的所有子域有效

▼ 路径

- ▼ 请求 URL 中包含指定路径才会把 cookie 发送给服务器
 - ▼ `http://wrox.com/books/`
 - 访问在 `http://www.wrox.com/` 下的页面不会发送 cookie

▼ 过期时间

- 默认情况下，浏览器会话结束后删除所有 cookie
- ▼ 不发送到服务器的时间点
 - 设置为过去的时间会立即删除 cookie
- ▼ GMT 格式
 - `Wdy, DD-Mon-YYYY HH:MM:SS GMT`

▼ 安全标志

- ▼ 只在使用 SSL 安全连接的情况下才把 cookie 发送到服务器
 - `https://www.wrox.com` 发
 - `http://www.wrox.com` 不发

▼ JS 中的 cookie

▼ BOM 接口: `document.cookie`

▼ 读取

- 直接访问 `document.cookie`

```
cookieStart = document.cookie.indexOf(cookieName);  
  
cookieEnd = document.cookie.indexOf(';', cookieName);  
  
cookieValue =  
decodeURIComponent(document.cookie.substring(cookieStart +  
cookieName.length, cookieEnd));
```

▼ 设置

- 直接对 `document.cookie` 进行赋值
- 赋值时不会覆盖之前存在的任何 cookie
- 如果设置的名称已存在，则更新该 cookie

▼ 格式

- `name=value; expires=expiration_time; path=domain_path;
domain=domain_name; secure`

▼ 对名/值进行编码

- encodeURIComponent()

- ▼ 对名/值进行解码

- decodeURIComponent()

- ▼ 注意

- ▼ HTTP-only 的 cookie

- 可以在浏览器设置，也可以在服务器设置
- 只能在服务器读取

- ▼ 在 cookie 中保存大量信息可能会影响特定域浏览器请求的性能

- 保存的 cookie 越大，请求完成的时间越长
- cookie 任何人都可能获得，应该避免在 cookie 中存储重要或敏感信息