

关于 cookie:

1. 是什么?

本质就是一个【字符串】，里面包含着浏览器和服务器沟通的信息（交互时产生的信息）。
存储的形式以：【key-value】的形式存储。

浏览器会自动携带该网站的 cookie，只要是该网站下的 cookie，全部携带。

2. 分类:

① 会话 cookie（关闭浏览器后，会话 cookie 会自动消失，会话 cookie 存储在浏览器运行的那块【内存】上）。

② 持久化 cookie：（看过期时间，一旦到了过期时间，自动销毁，存储在用户的硬盘上，备注：如果没有到过期时间，同时用户清理了浏览器的缓存，持久化 cookie 也会消失）。

3. 工作原理:

—当浏览器第一次请求服务器的时候，服务器可能返回一个或多个 cookie 给浏览器

—浏览器判断 cookie 种类

—会话 cookie：存储在浏览器运行的那块内存上

—持久化 cookie：存储在用户的硬盘上

—以后请求该网站的时候，自动携带上该网站的所有 cookie（无法进行干预）

—服务器拿到之前自己“种”下 cookie，分析里面的内容，校验 cookie 的合法性，根据 cookie 里保存的内容，进行具体的业务逻辑。

4. 应用:

解决 http 无状态 的问题（例子：7 天免登录，一般来说不会单独使用 cookie，一般配合后台的 session 存储使用）

5. 不同的语言、不同的后端架构 cookie 的具体语法是不一样的，但是 cookie 原理和工作过程是不变的。

备注：cookie 不一定只由服务器生成，前端同样可以生成 cookie，但是前端生成的 cookie 几乎没有意义。

6. 对比浏览器的本地存储:

1. localStorage:

- (1). 保存的数据，只要用户不清除，一直存在
- (2). 作为一个中转人，实现跨页签通信。
- (3). 保存数据的大小：5MB - 10MB

2. sessionStorage:

- (1). 保存的数据，关闭浏览器就消失
- (3). 保存数据的大小：5MB - 10MB

3. cookie:

- (1). 分类：会话 cookie——关闭浏览器消失、持久化 cookie——到过期时间消失
- (2). 保存数据的大小：4K —— 8K
- (3). 主要用于解决 http 无状态（一般配合后端的 session 会话存储使用）
- (4). 浏览器请求服务器时，会自动携带该网站的所有 cookie

关于 session:

1. 是什么?

标准来说, session 这个单词指的是会话。

(1). 前端通过浏览器去查看 cookie 的时候, 会发现有些 cookie 的过期时间是: session, 意味着该 cookie 是会话 cookie。

(2). 后端人员常常把【session 会话存储】简称为: session 存储, 或者更简单的称为: session

2. 特点:

1. 存在于服务端

2. 存储的是浏览器和服务器之间沟通产生的一些信息

3. 默认 session 的存储在服务器的内存中, 每当一个新客户端发来请求, 服务器都会新开辟出一块空间, 供 session 会话存储使用。

4. 工作流程:

—第一次浏览器请求服务器的时候, 服务器会开辟出一块内存空间, 供 session 会话存储使用。

区别 browser 的
session storage

—返回响应的时候, 会自动返回一个 cookie(有时会返回多个, 为了安全), cookie 里包含着, 上一步产生会话存储“容器”的编号(id)

—以后请求的时候, 会自动携带这个 cookie, 给服务器。

—服务器从该 cookie 中拿到对应的 session 的 id, 去服务器中匹配。

—服务器会根据匹配信息, 决定下一步逻辑

5. 备注: [1]. 一般来说 cookie 一定会配合 session 使用。

[2]. 服务端一般会做 session 的持久化, 防止由于服务器重启, 造成 session 的丢失。

[3]. session 什么时候销毁?

(1). 服务器没有做 session 的持久化的同时, 服务器重启了。

(2). 给客户端种下的那个用于保存 session 编号的 cookie 销毁了, 随之服务器保存的 session 销毁(不管是否做了 session 的持久化)。

(3). 用户主动在网页上点击了“注销” “退出登录”等等按钮。