Cookie和session

我们为什么要用cookie和session？

他们是什么东西？

HTTP：超文本传输控制协议；

四个步骤：

1. 建立连接；
2. 发起请求；
3. 响应请求；
4. **断开连接**；

http规定，必须是客户端主动发起请求，不能是服务器主动响应内容；

HTTP协议是**无状态**；

实现会话控制：服务器端要知道你是谁

Cookie和session；

Cookie：饼干

服务器响应到客户端的标记信息：标记符号；

请求头之上会自动带上cookie信息；

Cookie是存储在客户端的（浏览器），有大小限制（4k）；

但是，客户可以完全控制，安全性比较低；

本地缓存一些数据；

对cookie进行签名：齐缝章

Session:

Session是存储在服务器之上的：内存 或 数据库，相对来讲安全性比较高；

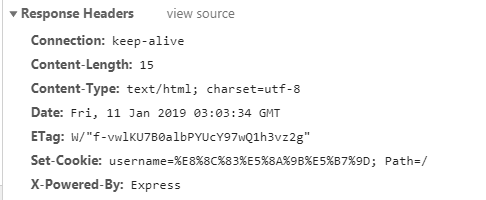
没有大小限制；

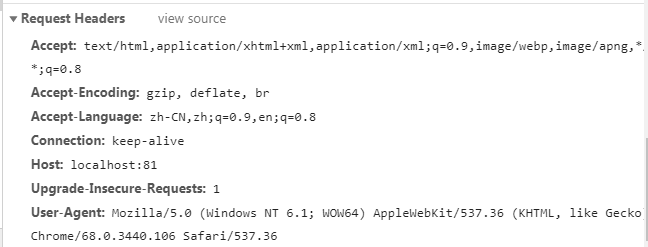
依赖于cookie：；

Nodejs的cookie实现：

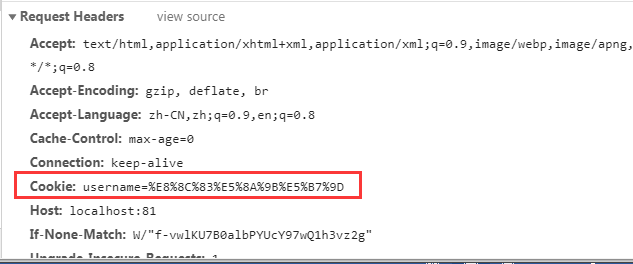
Cookie-parser

在设置cookie之前的第一次访问：





后面的2,3,4，，，，，次访问：



Session的实现：

Express-session

/\*

各参数意义：

secret：用来对session数据进行加密的字符串.这个属性值为必须指定的属性。

name：表示cookie的name，默认cookie的name是：connect.sid。

maxAge：cookie过期时间，毫秒。

resave：是指每次请求都重新设置session cookie，假设你的cookie是6000毫秒过期，每次请求都会再设置6000毫秒。

saveUninitialized： 是指无论有没有session cookie，每次请求都设置个session cookie ，默认给个标示为 connect.sid。

\*/

COOKIE和SESSION：

功能：实现了会话控制；

Cookie是存储在本地的，安全问题堪忧，里面不应该存储比较重要的数据；

有大小限制，4k；

Session是存储在服务器端，相对来讲比较安全，依赖于cookie；

localStorage：本地数据存储机制，属于H5的新功能，有效期是永久到，直到用户手动删除，存储在磁盘上的；

作用域：

Cookie的作用域是当前域名下；

Session的作用域是当前域名下；

localStorage的作用域是当前域名下；

sessionStorage的作用域是当前窗口下；

生命周期：

Cookie的生命周期是有时间限制的，可以手动设置有效期，默认是关闭浏览器就没有了；

Session的声明周期是依赖于客户端的cookie；

localStorage是永久的，直到手动删除；

sessionStorage是窗口关闭；

localStorage、sessionStorage、cookie他们都是存储在客户端的：

在请求头之上，会自动带上cookie；

做验证码：

1. 是一张图片：img；
2. 图片上的信息是随机的：Math.random；
3. 验证用户输入的信息和图片上的信息是否一致： input<====>session；

验证码是拿来干什么？

防止机器人暴力破解账号信息；

所以在验证的时候先验证验证码；