# Cookie

## 会话技术

### 什么是会话？

人与人之间的交谈可以算会话吧？这个挺容易理解吧，我说一句，你说一句，直到谈话结束。那对web开发过程中，怎么理解会话呢？比如用户打开一个浏览器，访问同一个服务器的多个web资源，直到最后用户关闭浏览器，整个过程就称为一个会话。

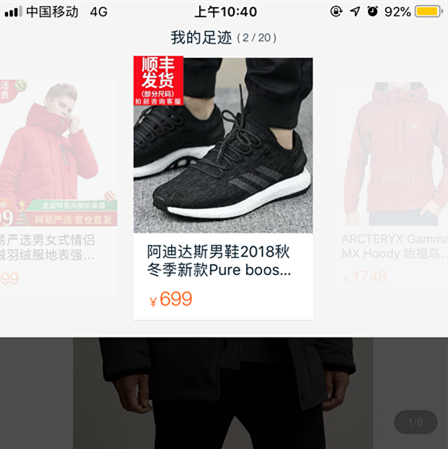
### 会话解决什么问题？

那会话有什么用呢？可以解决什么问题呢？比如说我连续点击某个链接，tomcat会记录下你点击了多少次吗？是不是不会啊？或者说我在网站上的浏览记录，点击了哪些链接，这些服务器是不是也没法帮我们去统计啊？

因为**http请求是无状态的**。什么叫无状态呢？（面试会问）客户端再次访问服务器时，服务器不知道这个客户端是否之前访问过服务器。也就无法存储一些浏览过程中产生的数据信息。

会话技术就是为了解决这些浏览器和服务器交互过程中，产生的数据。

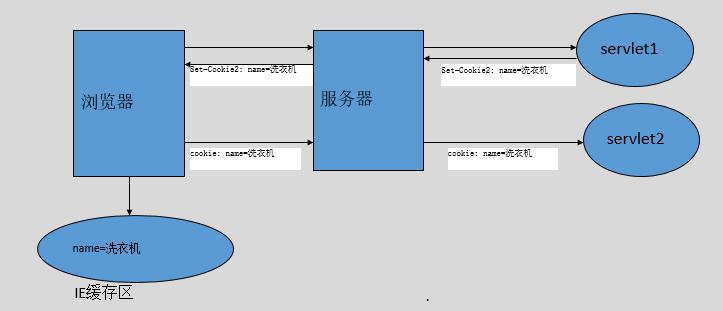
比如说购物时，加入购物车里面的商品。还有浏览商品时，浏览历史，或者说足迹等。



## Cookie

### 原理

服务器程序把每个用户的数据以cookie的形式写给用户各自的浏览器。当用户使用浏览器再去访问服务器中的web资源时，就会带着各自的数据去访问。这样，WEB服务器就能通过cookie去识别用户了。web资源处理的就是用户各自的数据了。这段话怎么理解呢？比如说你去理发店理发， 然后办了一张充值卡，下次你再去的时候，你拿着你的充值卡，人家是不是就知道你曾经来过，以及什么时间使用了多少钱等等，是理发还是烫发等等。



### Cookie API

#### 如何创建cookie



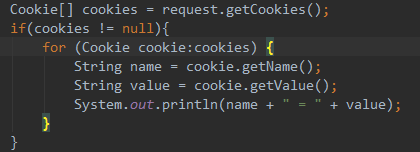
#### 如何将cookie发回给浏览器





响应报文多一个响应头，Set-Cookie

#### 再次请求服务器接收cookie

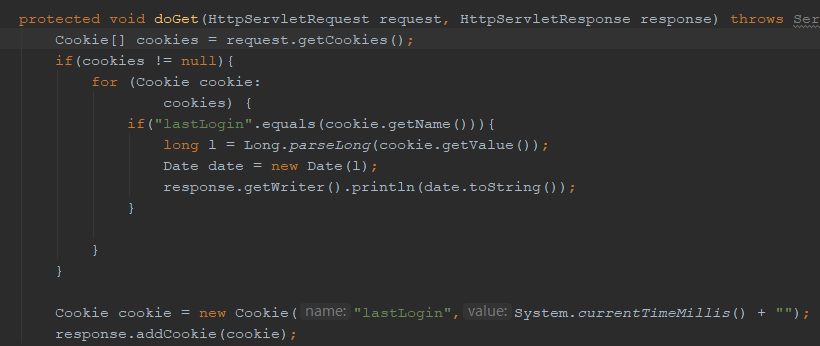


需要注意的是，每一个cookie对象是不是只能存储一对信息啊，同时，键值对都是string类型。

这个时候，我们新建一个1.html，访问这个页面，会带上cookie吗？也会带上cookie。

什么时候不带呢？访问其他域名或者ip是不是就不带啊？

### 获取上次访问时间



### Cookie有效期以及保存位置

cookie setMaxAge

A positive value indicates that the cookie will expire after that many seconds have passed. Note that the value is the *maximum* age when the cookie will expire, not the cookie's current age.

A negative value means that the cookie is not stored persistently and will be deleted when the Web browser exits. A zero value causes the cookie to be deleted.

没有设置setMaxAge的时候，默认保存在内存中；关闭浏览器，不可见



关闭浏览器，重新点击链接：



设置了有效时间，则保存在硬盘中。

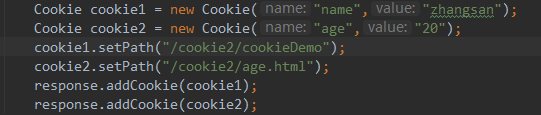
### 不同浏览器是否可以共享cookie

不能

### 设置cookie的path

前我们设置的cookie，是不是访问域名下所有的地址都会带上cookie啊，那我们如果想仅部分路径带上cookie，其他路径不带上。比如说，前端的js，css等文件，这些是不是不需要带上cookie啊？

<http://localhost:8080/cookie2/cookieDemo>





再次访问<http://localhost:8080/cookie2/cookieDemo>



访问<http://localhost:8080/cookie2/age.html>



setPath是除了域名（ip + port）以外的东西。

需要注意的是假如一个cookie的path信息是

localhost/app1/abc/cookie,它能否区分一个应用名为app1，后面带abc路径名以及一个应用名为/，路径为/app1/abc？不能

setDomain设置除自身域名之外的cookie，chrome会阻止，不保存。

### cookie细节

* 一个Cookie只能标识一种信息，它至少含有一个标识该信息的名称（NAME）和设置值（VALUE）value也必须是字符串类型。
* 一个WEB站点可以给一个WEB浏览器发送多个Cookie，一个WEB浏览器也可以存储多个WEB站点提供的Cookie。
* 浏览器一般只允许存放300个Cookie，每个站点最多存放20-50个Cookie，每个Cookie的大小限制为4KB。
* 如果创建了一个cookie，并将他发送到浏览器，默认情况下它是一个会话级别的cookie（即存储在浏览器的内存中），用户退出浏览器之后即被删除。若希望浏览器将该cookie存储在磁盘上，则需要使用maxAge，并给出一个以秒为单位的时间。将最大时效设为0则是命令浏览器删除该cookie。
* 注意，删除cookie时，path必须一致，否则不会删除

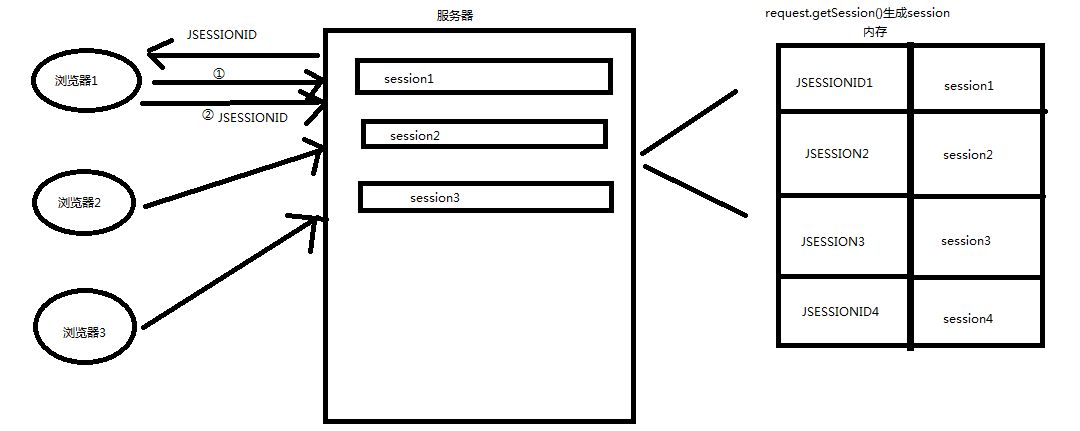
# Session

cookie是**客户端技术**，程序把每个用户的数据以cookie的形式写给用户各自的浏览器。当用户使用浏览器再去访问服务器中的web资源时，就会带着各自的数据去。这样，web资源处理的就是用户各自的数据了。

鉴于cookie保存在浏览器端，且大小，个数有一定限制，因此又出现了另外一种保存的形式，session。

## 什么是Session

Session是**服务器技术**，利用这个技术，服务器在运行时可以为每一个用户的浏览器创建一个其独享的HttpSession对象，由于session为用户浏览器独享，所以用户在访问服务器的web资源时，可以把各自的数据放在各自的session中，当用户再去访问服务器中的其它web资源时，其它web资源再从用户各自的session中取出数据为用户服务。通俗的讲，以去超市购物为例，cookie的话是不是类似每个超市给每个人一张卡，但是卡可能会被别人拿走，session的话，类似超市根据你的信息，比如手机号，指纹等给你设置了一个针对你自己的会员标识。存放在超市是不是就安全多了。



## session的简单使用

### 获取session

|  |  |
| --- | --- |
| [HttpSession](mk:@MSITStore:F:\\王道训练营%20JAVA方向%20第七期\\第二阶段%20JavaWeb\\servlet-api.chm::/jsdk23/javax/servlet/http/HttpSession.html) | [getSession](mk:@MSITStore:F:\\王道训练营%20JAVA方向%20第七期\\第二阶段%20JavaWeb\\servlet-api.chm::/jsdk23/javax/servlet/http/HttpServletRequest.html" \l "getSession())() 无论如何会给你一个session对象 Returns the current session associated with this request, or if the request does not have a session, creates one. |
| [HttpSession](mk:@MSITStore:F:\\王道训练营%20JAVA方向%20第七期\\第二阶段%20JavaWeb\\servlet-api.chm::/jsdk23/javax/servlet/http/HttpSession.html) | [getSession](mk:@MSITStore:F:\\王道训练营%20JAVA方向%20第七期\\第二阶段%20JavaWeb\\servlet-api.chm::/jsdk23/javax/servlet/http/HttpServletRequest.html" \l "getSession(boolean))(boolean create)  Returns the current HttpSession associated with this request or, if there is no current session and create is true, returns a new session.  If create is false ，there is no current session return null; |

[getSession](mk:@MSITStore:F:\\王道训练营%20JAVA方向%20第七期\\第二阶段%20JavaWeb\\servlet-api.chm::/jsdk23/javax/servlet/http/HttpServletRequest.html" \l "getSession())()，有session对象，则返回一个对象；没有，则新建一个对象。

[getSession](mk:@MSITStore:F:\\王道训练营%20JAVA方向%20第七期\\第二阶段%20JavaWeb\\servlet-api.chm::/jsdk23/javax/servlet/http/HttpServletRequest.html" \l "getSession(boolean))(boolean create)当create为true时，和上面的API相同，当为false时，没有则不会新创建。它是依据什么判断有无session 的呢？是不是通过JSESSIONID啊，如果有则去内存中检索JSESSIONID对应的session名称，通过request.getSession()返回；如果没有，则新创建一个session。

这个验证是不是也很容易啊？



通过比较Session的id（JSESSIONID）。

### session的简单使用

#### setAttribute

#### getAttribute

#### removeAttribute

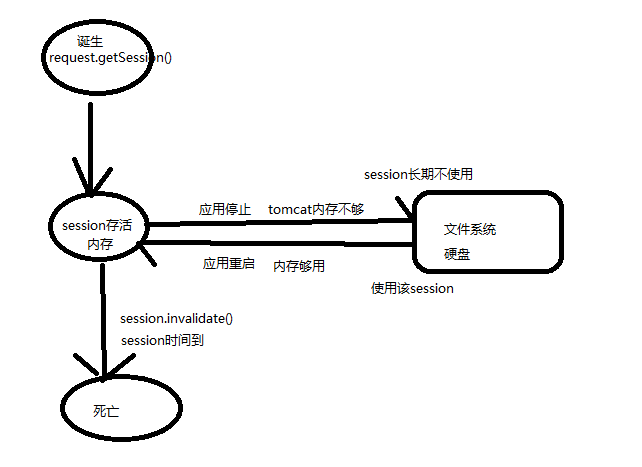
### 回顾一下request域，context域，session域

request域的作用范围较小，仅可以在同一请求内，即转发和包含的时候。响应返回，生命周期结束。

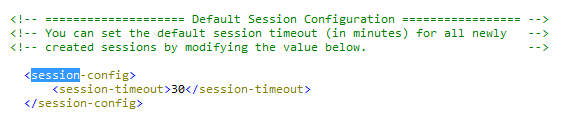
context域，作用范围广，不区分用户，任何servlet均可取值。保存多久？

session域，作用范围较广，区分用户，仅可取出自己的数据。伴随整个应用。

### session的生命周期



通过第一次调用request.getSession()创建session。那由谁创建的呢？由tomcat创建的吧。那session死亡的形式有什么呢？可以通过session主动调用invalidate方法，将session杀死。但是在我们之前的演示中，我们并没有调用这个API，那是不是说session就会一直存活呢？你比如说淘宝等大型电商网站，内存对于他们来说是不是非常稀缺的资源啊，你觉得淘宝会将你的登录信息一直保存一年左右吗？是不是不太可能啊？session经过一段时间以后，会自动结束生命周期，就像生物的寿命完结，自然死亡一样。那session的寿命有多久呢？我们看一下tomcat的conf/web.xml文件，有一个session的配置信息。



同时也可以利用manager页面管理session。

应用停止的时候，session会序列化到硬盘中，存储路径

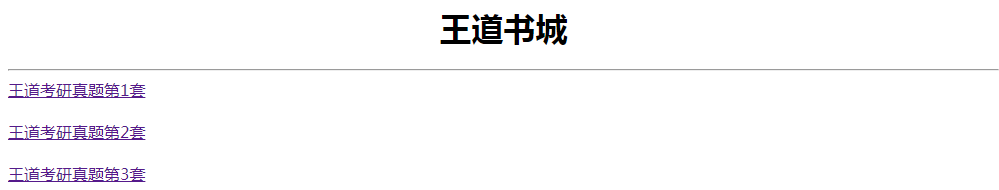
\.IntelliJIdea2018.2\system\tomcat\Tomcat\_8\_0\_53\_CookieSession\_3\work\Catalina\localhost\session

### 购物车案例

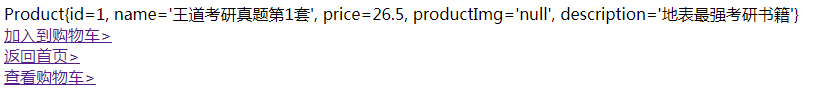
通过一个非常简易的案例，让大家了解一下session的使用。

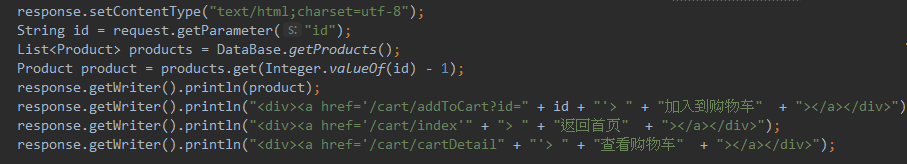


首页展示了10本书，通过一个模拟的数据库装载好书，然后打印到浏览器。

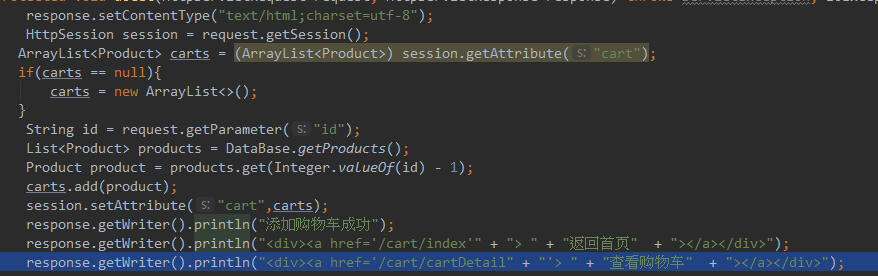


点击进入商品详情页面：



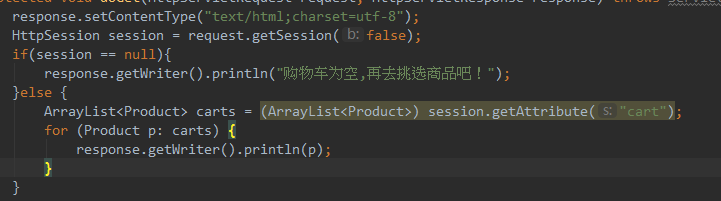


加入到购物车：



session中存放一个list，list用来存储商品集合信息。

查看购物车：



浏览器禁用cookie，session的解决办法：





重写这条href。