# 认知Cookie

## 什么是Cookie?

cookie的出现是为了解决http1.0无状态的问题，用来标识一个http会话的，一般cookie会和服务器上的session配套使用。Cookie存在浏览器中，每次发送请求，会把相应的cookie带到服务端，第一次发起请求，服务端会生成一个cookie返回给浏览器。

此处多说一下，随着http协议的发展，http逐渐支持持久连接，比如http1.1。

关于http1.0和http1.1的知识，可以参考此篇文章：

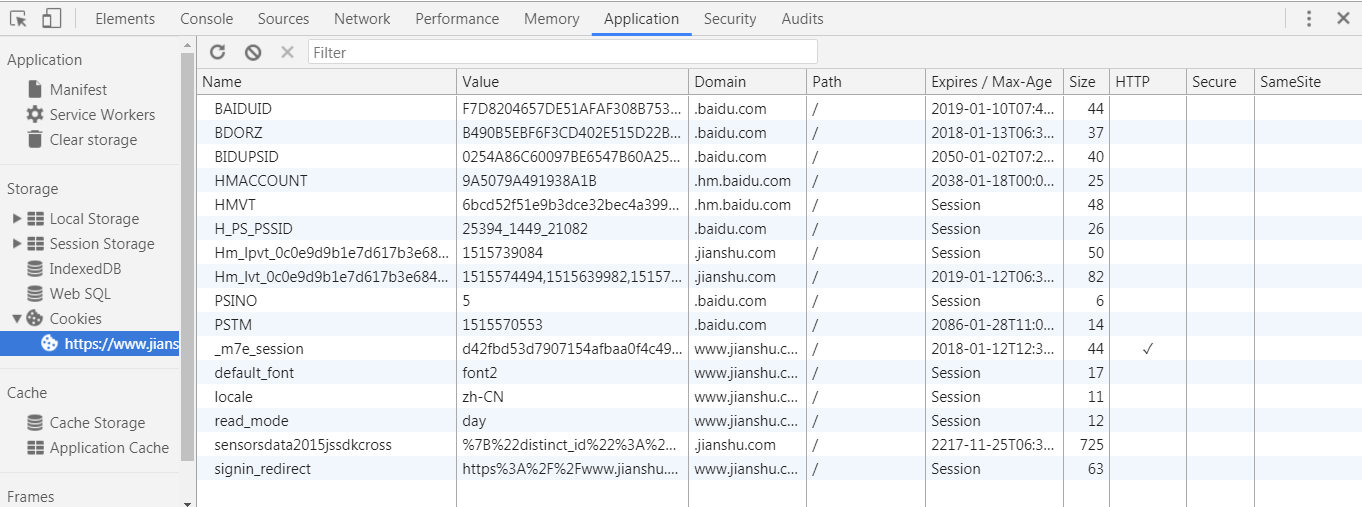
<https://www.jianshu.com/p/52d86558ca57>

## Cookie的属性



此图是Chrome浏览器存储cookie展示的属性。查看的路径：

设置->高级->内容设置->Cookie->查看所有 Cookie 和网站数据。



或者在开发者工具中也可以看到。

**Name**：一个cookie的名称。

**Value**：一个cookie的值。

**Domain**：可以访问此cookie的域名，可以设置完整的域名或者非顶级域名。

**Path**：可以访问此cookie的页面路径，比如domain是.abc.com,path是/test，那么只有/test路径下的页面可以读取此cookie。

**Expires/Max-Age**：字段为cookie的超时时间，若设置其值为一个时间，那么当到达此时间后，此cookie失效。不设置的话默认值是Session，意思是cookie会和session一起失效。当浏览器关闭(不是浏览器标签页，而是整个浏览器) 后，此cookie失效。

**Size**：此cookie的大小。

**Http**：cookie的httponly属性。若此属性为true，则只有在http请求头中会带有此cookie的信息，而不能通过document.cookie来访问此cookie。

**Secure**：字段 设置是否只能通过https来传递此条cookie。

## Cookie作用域

测试Cookie的作用域需要弄几个域名，修改

C:\Windows\System32\drivers\etc\hosts文件，将本机ip映射出四个域名，如下：

127.0.0.1 wx1.abc.com

127.0.0.1 wx2.abc.com

127.0.0.1 wx1.com

127.0.0.1 wx2.com

前两个是一级域名（abc.com）相同,二级域名（wx1、wx2）不同，后两个是一级域名不同（关于域名的分级有其它说法，此处不做解释）。然后我们再写两个jsp，一个用于设置Cookie，另一个用于显示Cookie。

setCookie.jsp

<%

Cookie cookie = new Cookie("test\_key","test\_value");

response.addCookie(cookie);

%>

getCookie.jsp

<%

Cookie[] cookies = request.getCookies();

if(cookies !=null){

for(Cookie cookie : cookies){

out.println(cookie.getName()+"-"+cookie.getValue()+"<br/>");

}

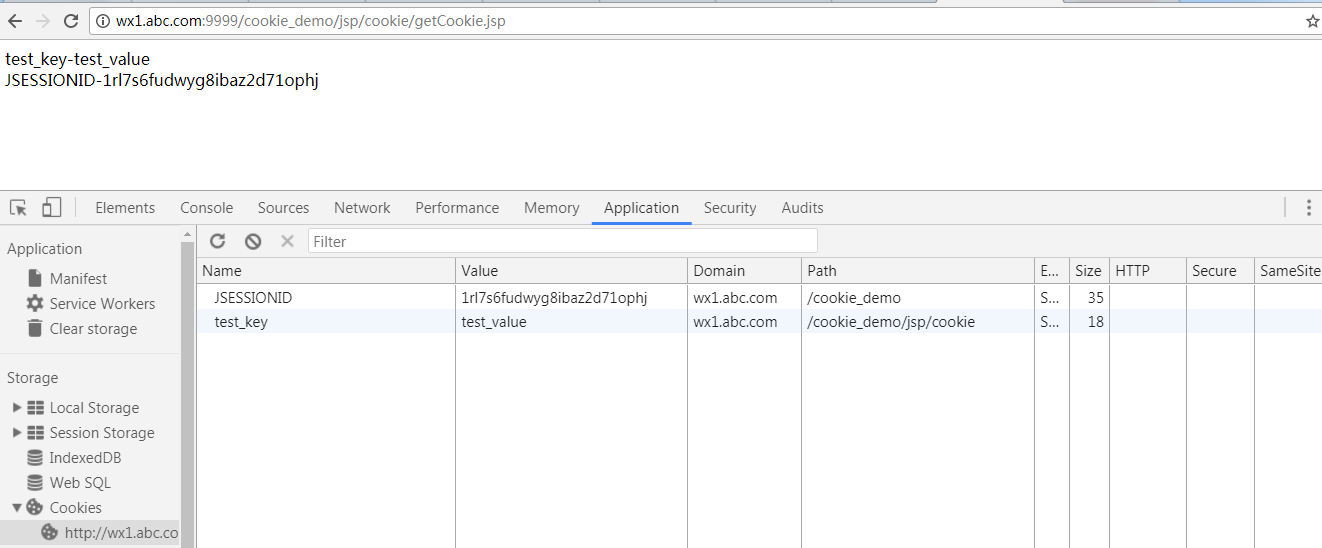
}

%>

测试一：

先访问：<http://wx1.abc.com:9999/cookie_demo/jsp/cookie/setCookie.jsp>

再访问：<http://wx1.abc.com:9999/cookie_demo/jsp/cookie/getCookie.jsp>



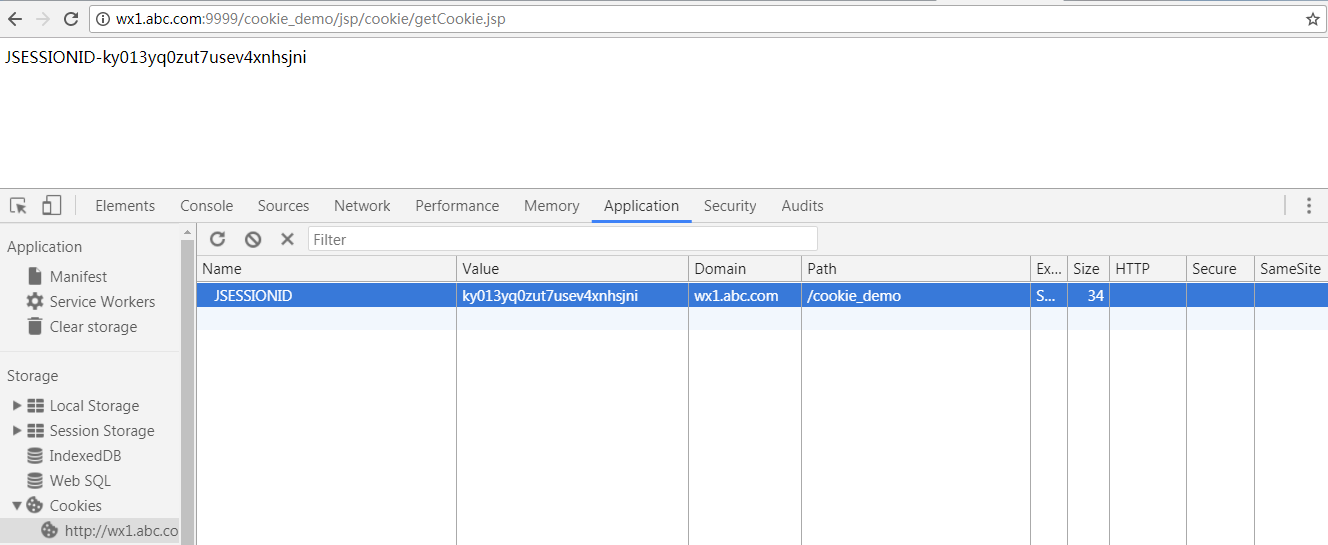
默认domain域名为当前域名，**path默认为当前uri路径，session对应的cookie的path为项目名称**。Path路径的更改可以通过cookie.setPath("/")进行设置。

测试二：

在setCookie.jsp中增加如下代码：

cookie.setDomain(".com");

清除浏览器cookie缓存，先访问setCookie页面，再访问getCookie页面。



发现设置顶级域名，浏览器不会将cookie写入浏览器。

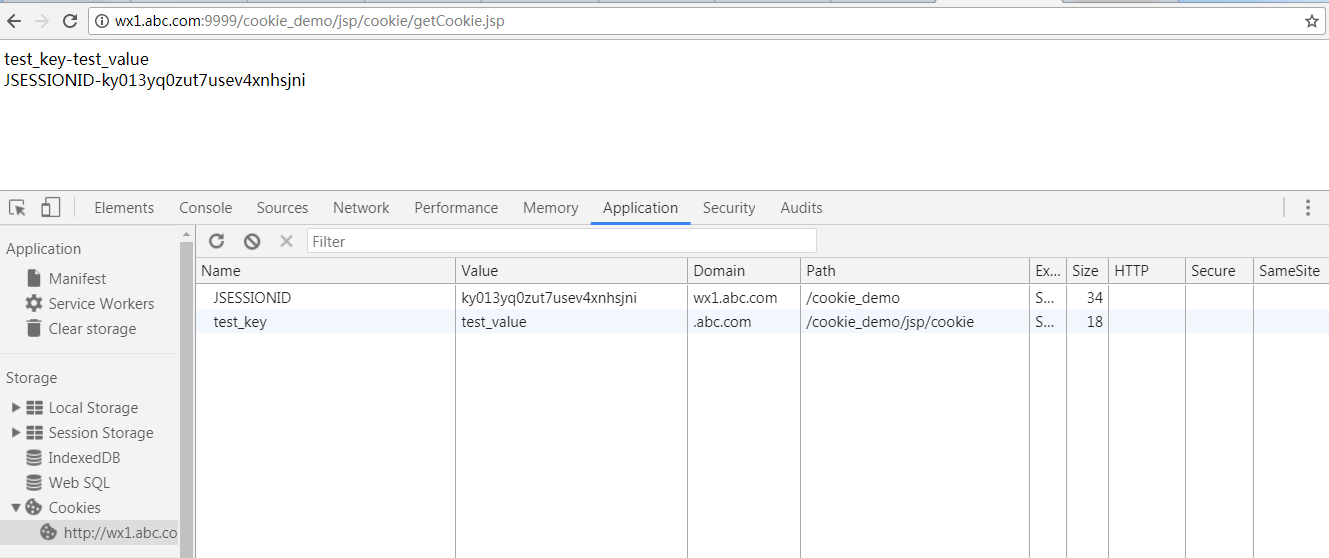
测试三：

在setCookie.jsp中增加如下代码：

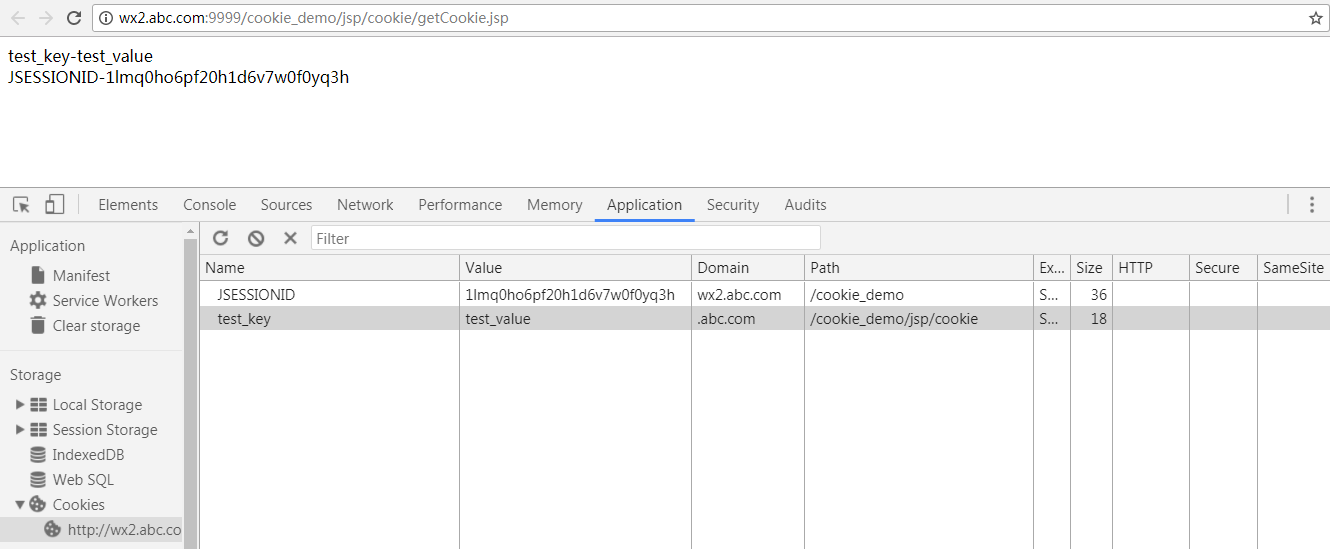
cookie.setDomain("abc.com");

先访问：<http://wx1.abc.com:9999/cookie_demo/jsp/cookie/setCookie.jsp>

再访问：<http://wx1.abc.com:9999/cookie_demo/jsp/cookie/getCookie.jsp>



再访问：<http://wx2.abc.com:9999/cookie_demo/jsp/cookie/getCookie.jsp>



可以得出一个结论：cookie作用域为父级域名时，所有子级域名都可以得到该cookie。

测试四：

在setCookie.jsp中增加如下代码：

cookie.setDomain("wx1.com");

先访问：[http://wx1.com:9999/cookie\_demo/jsp/cookie/setCookie.jsp](http://wx1.abc.com:9999/cookie_demo/jsp/cookie/setCookie.jsp)

在访问：[http://wx2.com:9999/cookie\_demo/jsp/cookie/getCookie.jsp](http://wx2.abc.com:9999/cookie_demo/jsp/cookie/getCookie.jsp)

此时在cookie中没有看到test\_key的值，说明：cookie不能跨域名设置。

**综上对cookie域名的总结：**

1. **Cookie默认作用域为当前域名。**
2. **Cookie作用域为父级域名时，所有子域名都可以得到该cookie（可实现同域名单点登录）。**
3. **Cookie不能跨域名设置，直接会被浏览器忽略。**
4. **Cookie不能将域名设置成顶级域名，直接会被浏览器忽略。**

## Cookie路径

测试一：

setCookie.jsp设置成如下：

<%

Cookie cookie = new Cookie("test\_key","test\_value");

cookie.setPath("/abc");

response.addCookie(cookie);

%>

先访问：<http://wx1.abc.com:9999/cookie_demo/jsp/cookie/setCookie.jsp>

在访问：<http://wx1.abc.com:9999/cookie_demo/jsp/cookie/getCookie.jsp>

发现cookie值没有设置到浏览器中。

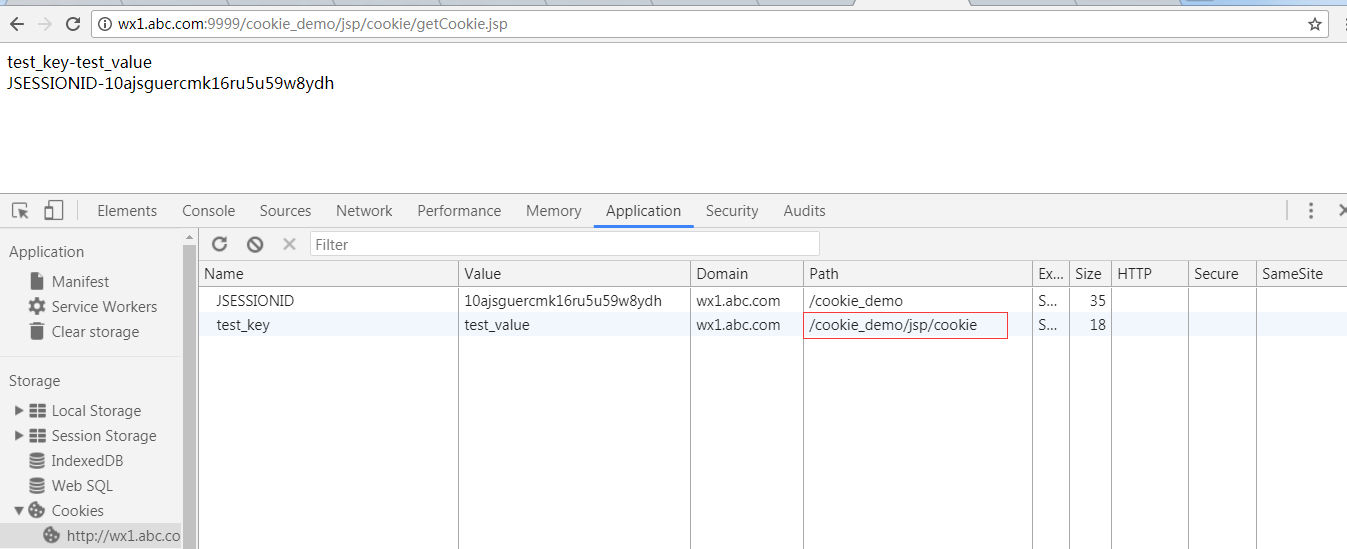
测试二：

在setCookie.jsp中，修改成如下代码：

cookie.setPath("cookie\_demo/jsp");

先访问：<http://wx1.abc.com:9999/cookie_demo/jsp/cookie/setCookie.jsp>

在访问：<http://wx1.abc.com:9999/cookie_demo/jsp/cookie/getCookie.jsp>

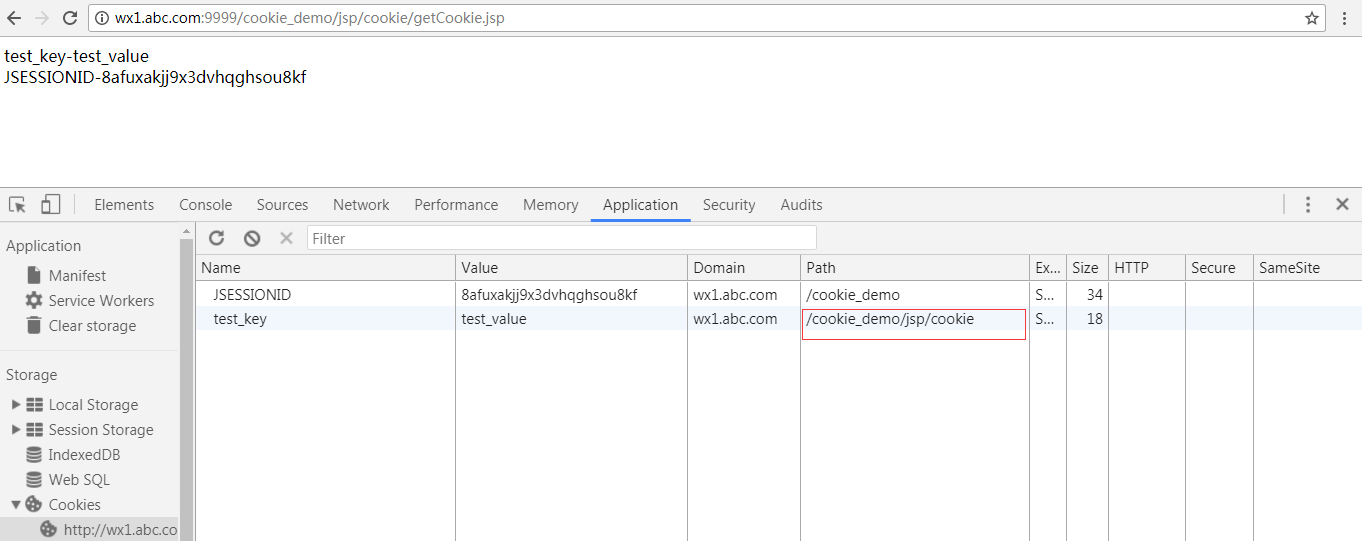


在setCookie.jsp中，修改成如下代码：

cookie.setPath("abc");

先访问：<http://wx1.abc.com:9999/cookie_demo/jsp/cookie/setCookie.jsp>

在访问：<http://wx1.abc.com:9999/cookie_demo/jsp/cookie/getCookie.jsp>

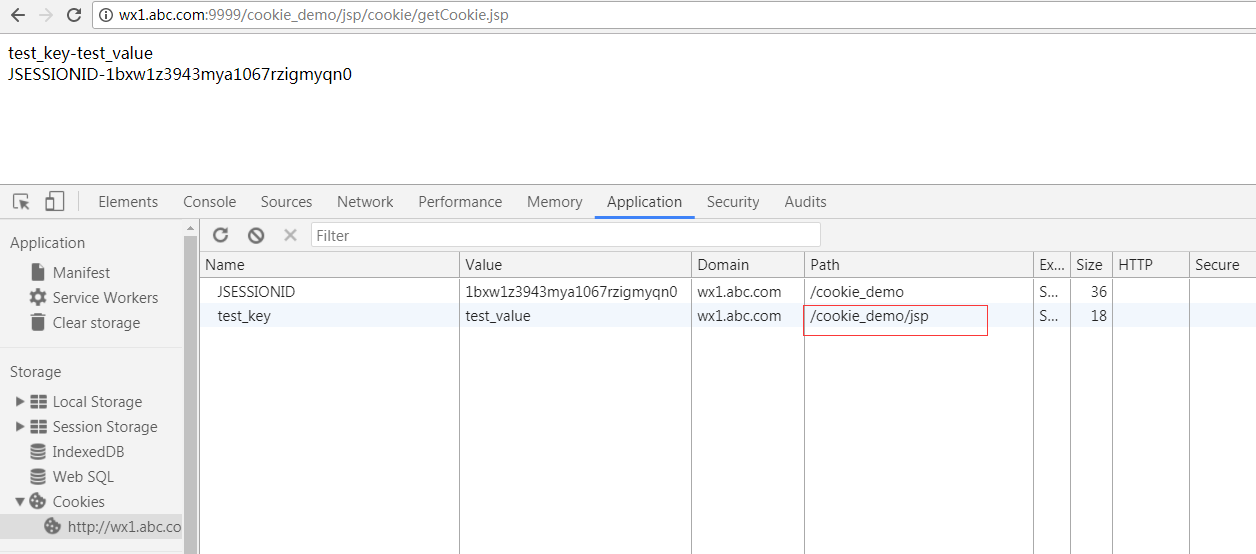


在setCookie.jsp中，修改成如下代码：

cookie.setPath("/cookie\_demo/jsp");

先访问：<http://wx1.abc.com:9999/cookie_demo/jsp/cookie/setCookie.jsp>

在访问：<http://wx1.abc.com:9999/cookie_demo/jsp/cookie/getCookie.jsp>



从以上实验可以发现，cookie中path的设置必须以“/”开头，如果不是以“/”开始，都是非法路径，cookie设置的path会使用默认规则，也就是当前的uri路径。

**cookie路径的总结：**

1. **Path默认为当前uri路径，session对应的cookie的path为项目名称。**
2. **Path的值必须以“/”开始，否则路径非法，设置使用默认规则。**
3. **Path设置的路径必须是uri的子路径，否则浏览器会默认忽略此cookie。**

## Cookie总结

一个http发送请求，能携带哪些cookie信息，是由cookie中的domain和path两个属性决定的。一般的规则是：**先匹配domain，再匹配path，domain必须是当前请求的域名或者子域名，path必须是当前请求URI路径或者子路径**。如果二者都匹配，这个cookie才能被携带。

参考的文章：

<https://www.cnblogs.com/tzyy/p/4151291.html>

<http://blog.csdn.net/ghsau/article/details/20395681>