Cookies／session

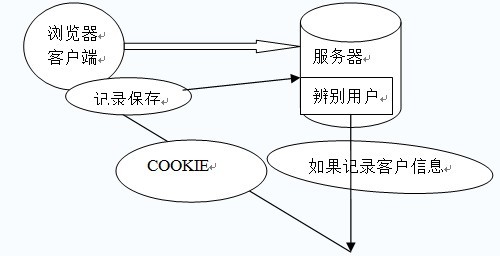
为什么从一个页面跳到另外一个不用重新登录。

每一次登录后都dispatcher.lua文件产生一个临时的cookies和一个stoke（在浏览器关闭后实效），这个cookies中记录了当前的用户信息。stoke是在dispatcher中产生的一个随机数。切换菜单的时候，若是cookies中的用户信息是没有经过授权的，或是stoke的值跟服务器中的值不相同时，就会转跳到登录页面。

1. Cookies/session是个啥？

详解请参看：<http://blog.csdn.net/fangaoxin/article/details/6952954/>

简单的说，会话（Session）跟踪是Web程序中常用的技术，用来跟踪用户的整个会话。常用的会话跟踪技术是Cookie与Session。Cookie通过在客户端记录信息确定用户身份，Session通过在服务器端记录信息确定用户身份。



主要涉及到两个文件，dispatcher.lua文件用来产生随机数stoke和cookies。

sauth.lua文件用来检验用户的合法性

**源码**

|  |
| --- |
| local sess = ctx.authsession  local verifytoken = false  if not sess then  sess = luci.http.getcookie("sysauth")  sess = sess and sess:match("^[a-f0-9]\*$")  verifytoken = true  end |

sess为服务器上的session的value，当服务器上没有该session的值时候，从浏览器的cookies中name为“sysauth”获取value

在未登录的情况下，服务器端的session和浏览器端的cookies都为空，此时verifytoken = true

然后

|  |
| --- |
| local eu = http.getenv("HTTP\_AUTH\_USER")  local ep = http.getenv("HTTP\_AUTH\_PASS")  if eu and ep and luci.sys.user.checkpasswd(eu, ep) then  authen = function() return eu end  end |

从页面中获取输入的用户名和密码，并检查。如果用户名和密码正确则

|  |
| --- |
| if not sess then  local token = luci.sys.uniqueid(16)  sauth.reap()  sauth.write(sid, {  user=user,  token=token,  secret=luci.sys.uniqueid(16)  })  ctx.urltoken.stok = token  end  luci.http.header("Set-Cookie", "sysauth=" .. sid.."; path="..build\_url())  ctx.authsession = sid  ctx.authuser = user |

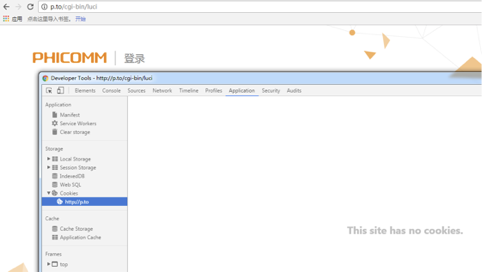
生成一个随机数token并赋值给服务器端的session的urltoken（stok）。然后服务器向浏览器发送一个cookie，名字为“sysauth”，value是一个table类型的，包括user=user,

token=token,

secret=luci.sys.uniqueid(16)以及path。

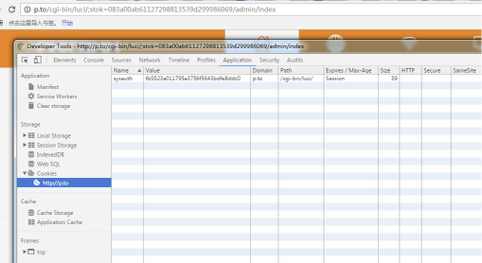
同时在session端记录

1.未登录状态



由于这时候还没有用户信息，所以此时服务器端的session或是浏览器端的cookies都是空。

2. 登录成功后



产生一个name为“sysauth”的cookies，在url写入一个stok。

另外如果记住密码的话会产生另一个cookies

