

C# 系统应用之Cookie\Session基础知识及php读取Cookie\Session

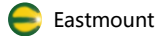
原创 Eastmount 最后发布于2014-01-26 04:11:17 阅读数 6294 ☆ 收藏

展开



Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法...



¥9.90

去订阅

本文主要是毕业设计"个人电脑使用记录清除软件"系列系统应用文章中关于Cookie方面的知识,主要从介绍Cookie的基础知识和PHP关于Cookie\Session两个会话管理机制的讲述.结合自己的PHP课程及Cookie方面的文章.提醒:由于作者写时有点晚,思路有些乱,文章论述和排版都不太满意,太杂乱无章,请海涵!

一. cookie基础知识

<一>.什么是Cookie?

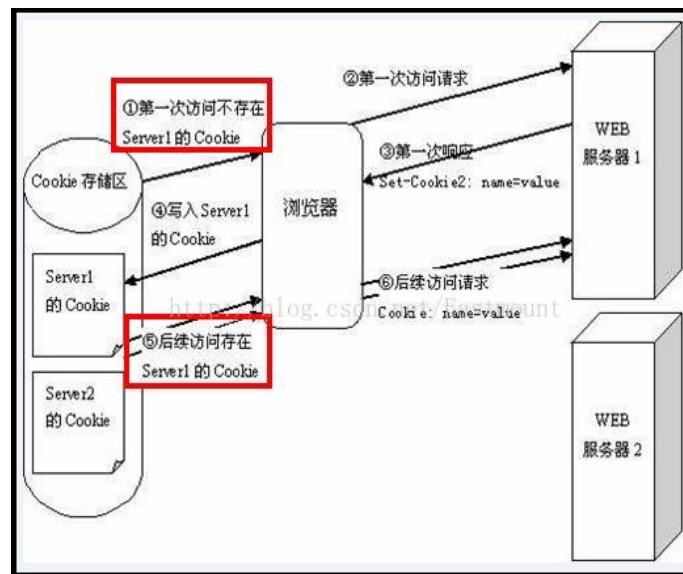
Cookie/Cookies是某些网站为了辨别用户身份而存储在本地终端(Client Side)上的数据(通常加密),它是一种在远程浏览器端储存数据并以此来跟踪和识别用户的机制.(详见维基百科<http://zh.wikipedia.org/wiki/Cookie>)

Cookie总是保存在客户端中,按在客户端的存储位置,分为内存Cookie和硬盘Cookie.其中内存Cookie由浏览器维护,保存在内存中,浏览器关闭就会消失.硬盘Cookie保存在硬盘里,除非用户手动清理或过期,将长期保存.

<二>.Cookie工作原理

Cookie是由服务器通过在HTTP响应头中加上特殊指示生成,发送给浏览器,浏览器将Cookie的value保存到某目录文件夹内,下次请求同一网站时就发送Cookie给服务器.服务器可以设置或读取Cookies中包含信息,维护用户跟服务器会话中的状态.其中,Cookies最典型的应用是判断用户登录网站信息和保存网站输入文字|选择等页面信息.

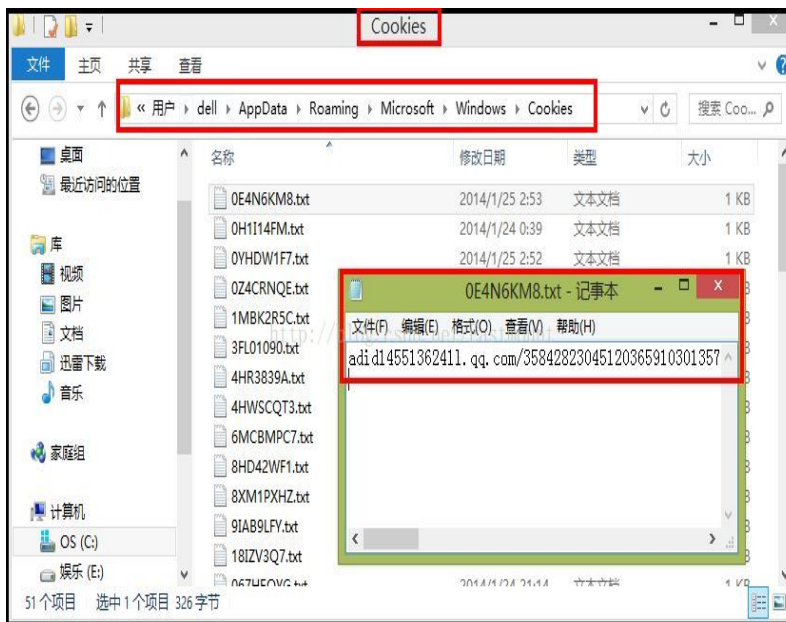
下图非常完美的诠释了Cookie机制.



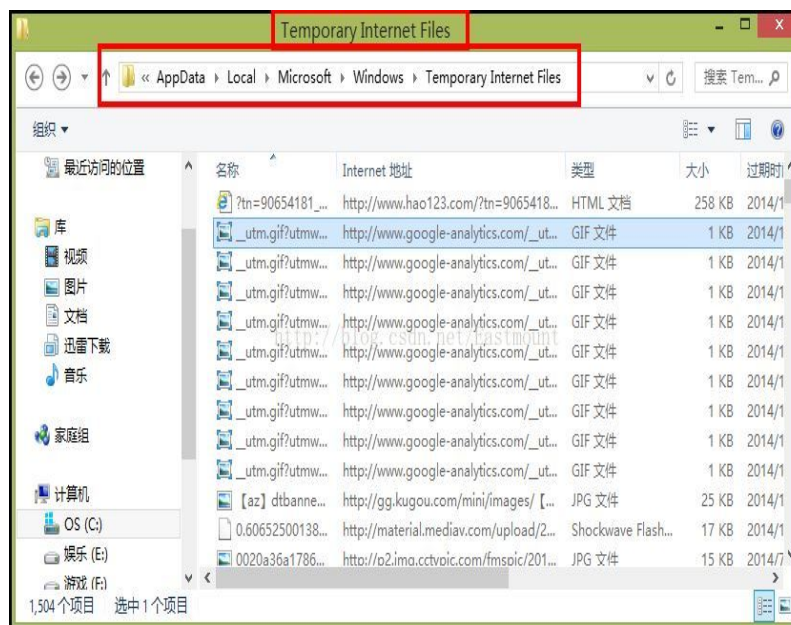
<三>.Windows文件路径(补充)

下面补充与该项目相关的浏览器历史记录和Cookie存储位置,以便历史记录清除操作(仅以自己的电脑参考).

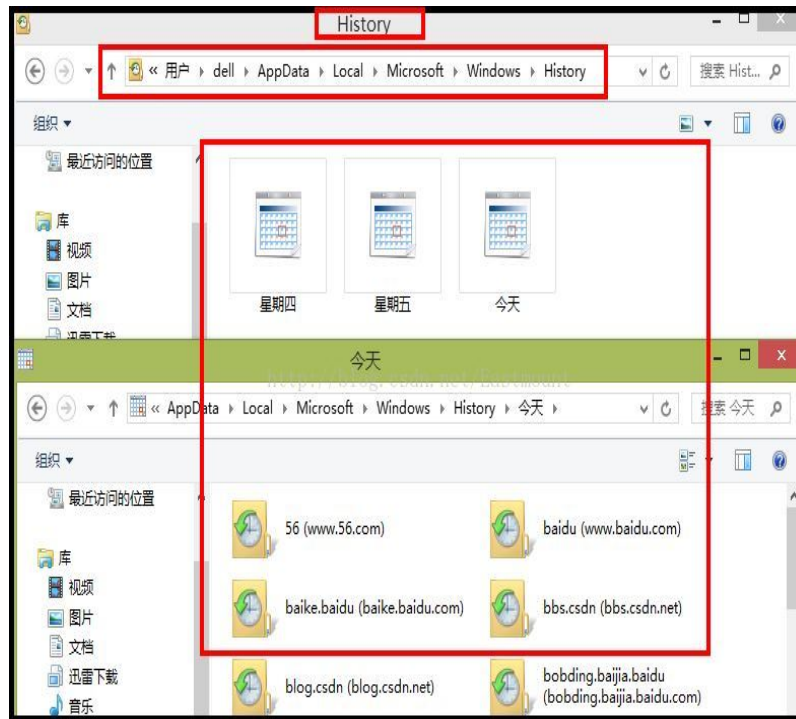
1.Windows中Cookie保存位置为"C:\Users\dell\AppData\Roaming\Microsoft\Windows\Cookies".Cookie记录用户ID、密码、浏览过的网页、停留时间等信息.如下图所示:



2.Windows中Internet临时文件位置为"C:\Users\dell\AppData\Local\Microsoft\Windows\Temporary Internet Files",它存放着最近浏览过的网页的内容(网页|图像|媒体副本等)以便以后快速查询并提高速度.如下图所示:



3.Windows中IE历史记录位置为"C:\Users\dell\AppData\Local\Microsoft\Windows\History",历史记录是存放最近时间访问过的网站地址,它以时间和站点存储.如下图所示:



二. session基础知识

<一>.什么是Session?

说到cookie,就不得不谈谈session.它究竟是又什么呢?它俩的区别是什么呢?

session(会话)是指一个终端用户与交互系统进行通信的时间间隔,通常指从进入网站到浏览器关闭所经过的时间.当程序需要为某个客户端请求创建一个session时,服务器先检查客户端的请求是否包含一个session唯一标识session ID(被存储于本地cookie中),如果包含则说明此客户端已创建过session,服务器就按session id把该session检索使用;如果不包含则为此客户端创建一个session并生成一个与session相关联的session id(它唯一标识符,不会重复又不容易被造访的长串字符串).访问网站的来客分配的唯一标识session id要么存放在客户端的cookie中(session使用过期时间设为0的cookie),要么经由URL传递.

补充:session是由服务器端存储,当用户连接服务器时会由服务器生成一个会话ID(Session ID),用该标识符来存取服务端的Session存储空间.而SessionID保存在客户端,用Cookie保存,当用户提交页面时,会将SessionID提交到服务器处理Session数据.而一旦客户端禁用Cookie,Session也会失效.

<二>.session和cookie对比

Cookie许多用途与Session相同,两种机制保存一些用户交互的信息.它们存在区别如下:

- 1.两者最大的区别是cookie存储在客户端的一小段信息,它完全存储在用户计算机上并加载到浏览器缓存中;session数据存放在服务器上.
 - 2.session使用具有一定局限性,但由于数据保存在服务器端,提供了很好的安全性;cookie使用比较方便,但不是很安全.黑客通过存储在本地的cookie进行cookie欺骗\跨站脚本攻击.(HTTP明文传输,HTTPS加密)
 - 3.由于cookie存储在本地且cookie的大小限制在4KB,对于复杂存储需求不够;由于session保存在服务器上,当访问增多时,会大量占用服务器的性能.
- (该部分参考资料:学校《LAMPWAMP架构与开发》PPT课件\文章<http://www.2cto.com/kf/201206/135471.html>)

三. PHP使用Cookie

<一>.设置cookie

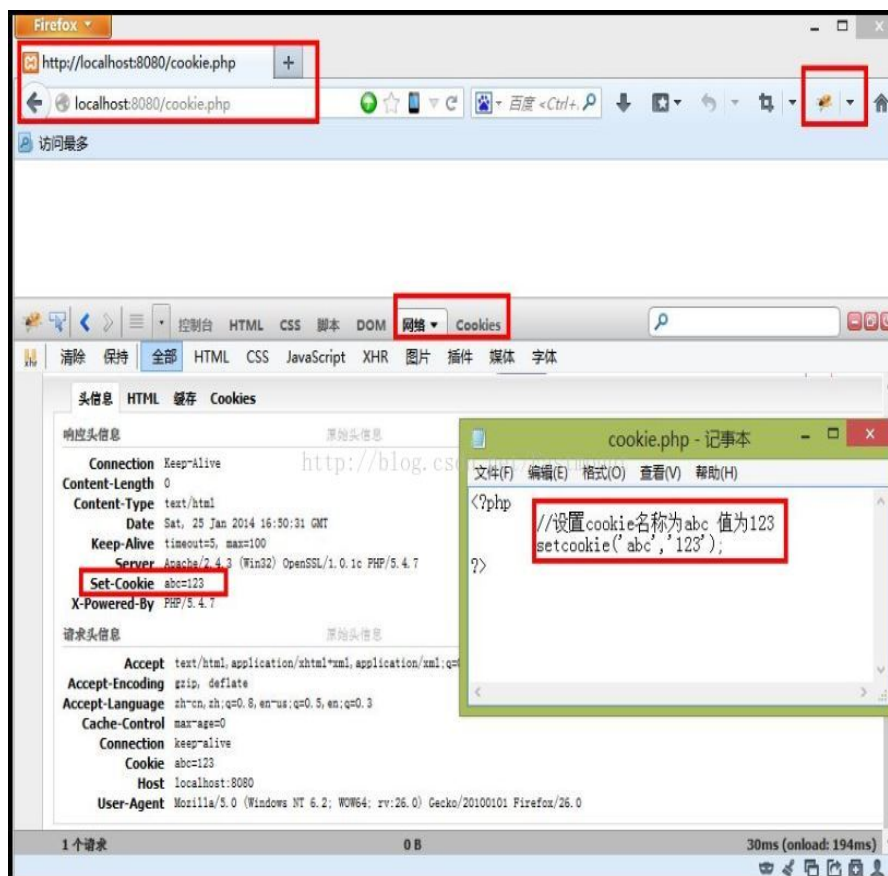
可以用setcookie()或setrawcookie()函数设置cookie,也可以通过向客户端直接发送http头来设置.其中setcookie:

```
// 函数向客户端发送一个http cookie
bool setcookie(
    string name,    // 必填 cookie变量名称
    string value,   // 必填 cookie变量值
    int expire,     // 可选 cookie有效期结束时间
    string path,    // 可选 cookie被发送到服务器的目录路径
    string domain,  // 可选 cookie域名
    int secure      // 可选 是否通过https加密链接传输cookie
)
```

设置多个cookie变量setcookie('var[a]','value');用数组表示变量,用\$_COOKIE['var']['a']来读取该COOKIE变量,亦可使用header()设置cookie,读者可自己查找资料.

```
// 设置cookie名称为abc 值为123
setcookie("abc","123");
// 设置TestCookie的有效期为3600秒 time() 返回当前时间
$value='something';
setcookie("TestCookie",$value,time()+3600);
// 设置完整的Cookie
setcookie("TestCookie",$value,time()+3600,"/forum","www.baidu.com",1);
```

使用XAMPP配置PHP环境后(配置过程可参考:<http://blog.csdn.net/eastmount/article/details/11823549>),通过Firefox火狐浏览器和Firebug插件运行结果如下图所示:



<二> 读取cookie

直接用php内置超级全局变量\$_COOKIE读取浏览器端的cookie.

```
<?php
```

```

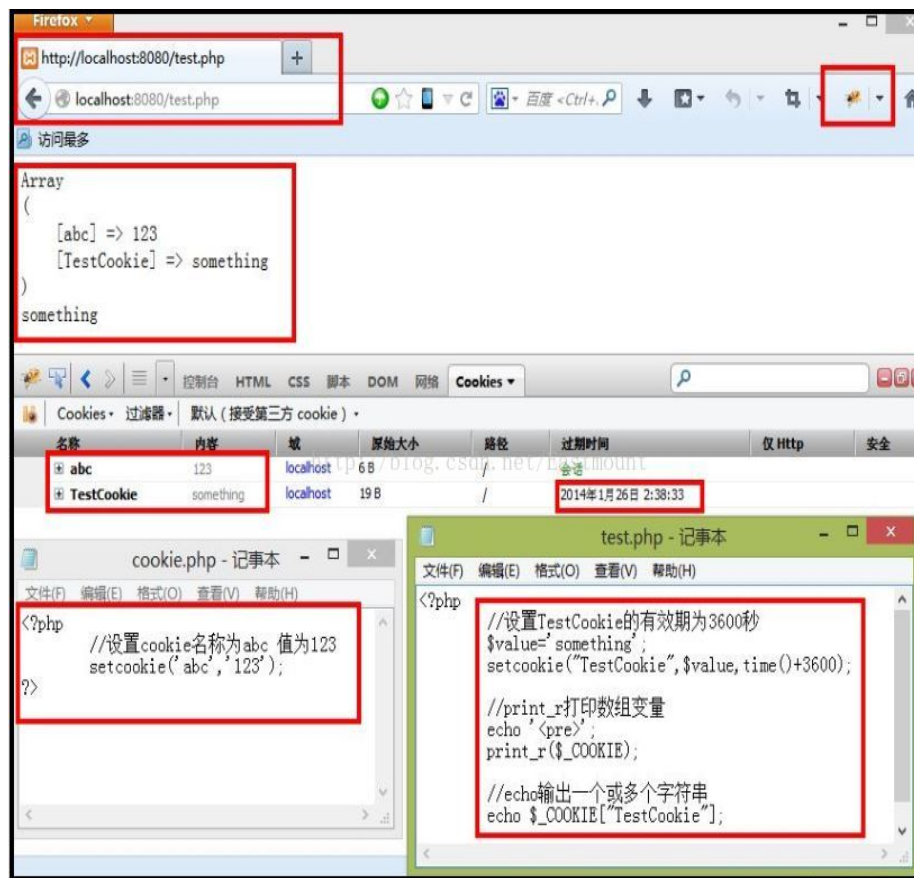
// 设置TestCookie的有效期为3600秒
                                $value='something';
setcookie("TestCookie",$value,time()+3600);

//print_r打印数组变量
echo '<pre>';
print_r($_COOKIE);

//echo输出一个或多个字符串
echo $_COOKIE["TestCookie"];
?>

```

注意:如果使用echo \$HTTP_COOKIE_VARS["TestCookie"]输出变量时,可能会提示错误:"Undefined variable: HTTP_COOKIE_VARS",因为新的版本已弃用\$HTTP_COOKIE_VARS,用\$_COOKIE替代输出.如下图所示:



可以发现使用setcookie函数设置的两个Cookie,其中Cookie名称为abc,值为123;Cookie名称为TestCookie,值为something,而且它的过期时间为1小时后,并使用print_r和echo输出.

<三>.删除cookie

只要把有效时间设为小于当前时间(失效时间),和把值设置为空.setcookie("TestCookie","",time()-1);

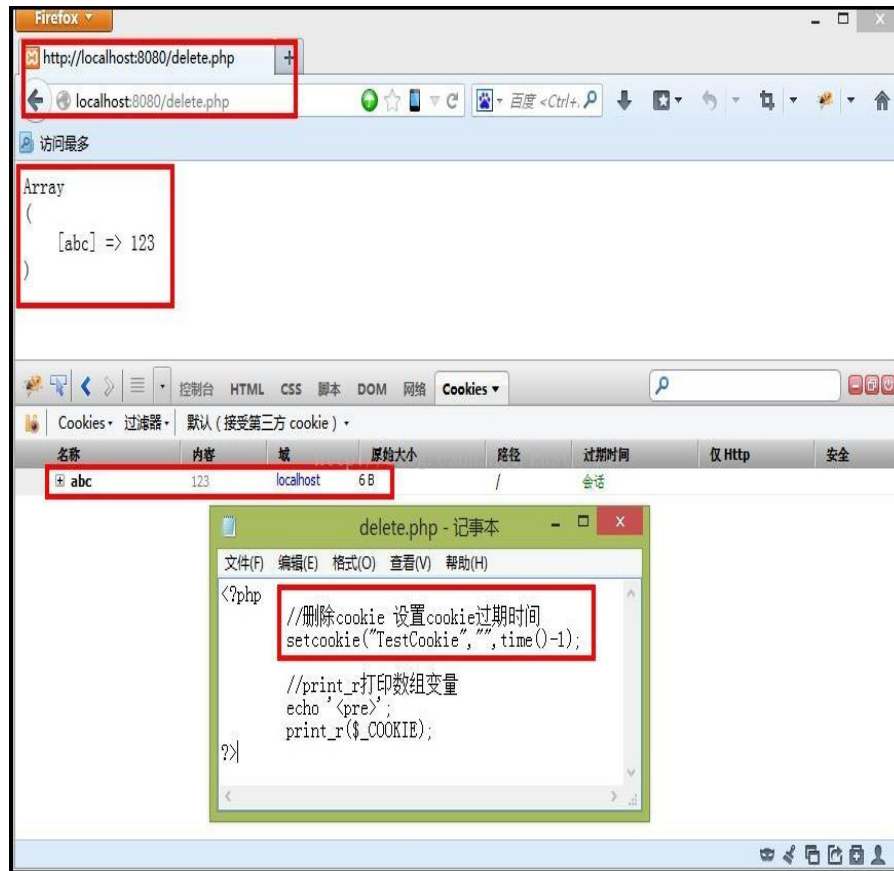
```

<?php
// 删除cookie 设置cookie过期时间
setcookie("TestCookie","",time()-1);

//print_r打印数组变量
echo '<pre>';
print_r($_COOKIE);
?>

```


结果如下图所示,可以发现TestCookie的数据已经被删除,只剩下abc的数据.



在使用PHP登录时可以使用下面这段代码,感觉比较实用的一段代码:

```
<?php
if(!isset($_COOKIE['name'])) //检测cookie变量是否设置
{
    header('location:index.php'); //跳转回主页
    die(); //退出当前脚本
}
else
{
    echo 'Hello:'. $_COOKIE['name'];
}
?>
```

(该部分参考资料:学校《LAMPWAMP架构与开发》PPT课件\博客<http://www.jb51.net/article/24871.htm>)

四. PHP使用Session

<—>.session启动

使用`session_start()`启用session,服务器端在生成session文件同时生成session ID哈希值和默认PHPSESSID的session name,并向客户端发送变量为PHPSESSID(session name),值为一个128位哈希值.服务器端将通过该cookie与客户端交互.

```
<?php
session_start(); //函数没有参数且返回值为true.最好将函数置于最先,且它之前不能有任何输出,否则会警告
?>
```

session通过cookie传送session ID,即服务器自动发送http头.当从该页面跳转到新页面并调用session_start()后,php将检查与给定ID相关的服务器段存储的session数据,如未找到则创建一个新的数据集.

<二>.session访问及读取

通过URL传送session ID, 也可以通过POST来传递session值.可以使用PHP内置的\$_SESSION变量可以方便的访问设置的session变量并给它赋值, 如\$_SESSION['XXX']='YYY'.使用echo和print_r输出session值.

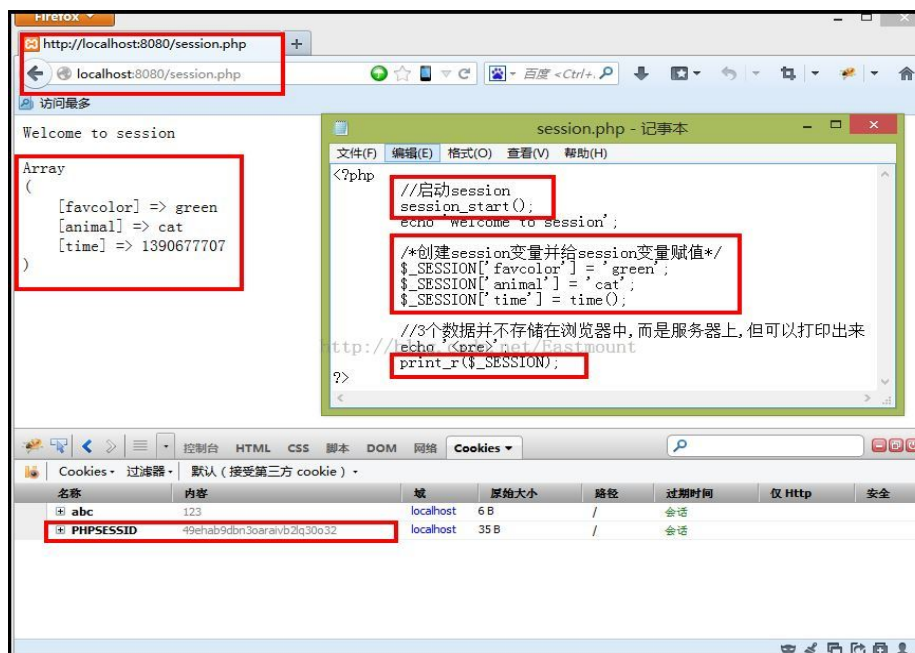
```
<?php
//启动session
session_start();
echo 'Welcome to session';

/*创建session变量并给session变量赋值*/
$_SESSION['favcolor'] = 'green';
$_SESSION['animal'] = 'cat';
$_SESSION['time'] = time();

//3个数据并不存储在浏览器中, 而是服务器上, 但可以打印出来
echo '<pre>';
print_r($_SESSION);

?>
```

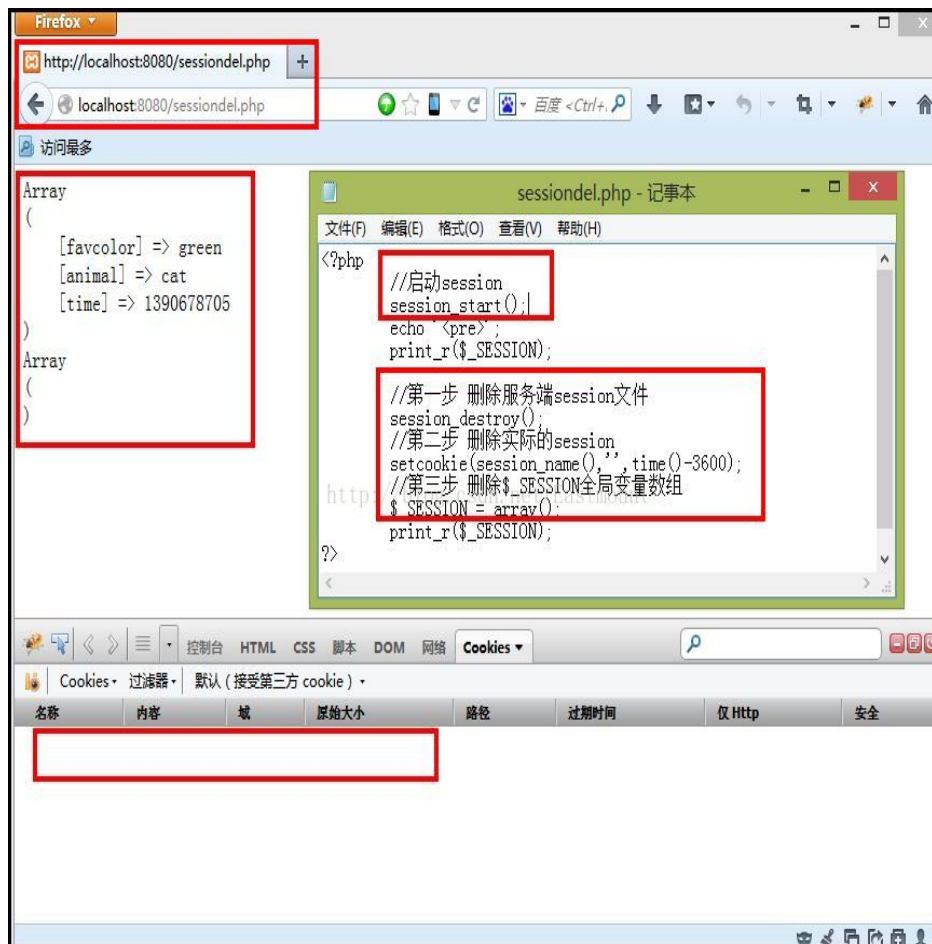
输入如下图所示,可以看到session ID对应的长串字符串:



<三>.session删除

session的删除需要三步,第一步删除服务器端的session,第二步删除实际的session,第三步删除\$_SESSION全局变量数组.代码如下:

```
// 第一步 删除服务端session文件
session_destroy();
// 第二步 删除实际的session
setcookie(session_name(), '', time()-3600);
// 第三步 删除$_SESSION全局变量数组
$_SESSION = array();
```



<四>.session安全问题

攻击者通过投入很大精力获取用户的会话ID,他们就可以在系统中拥有与此用户相同的能力,因此校验session ID的有效性是保护session安全.下面一段代码是我PHP项目时判断数据库MySQL中session用户名和密码的功能(作为自己以后的笔记):

```
<?php
//启动session
session_start();
//输入POST非空时赋值
if (!empty($_POST))
{
    $_SESSION['admin_name'] = $_POST['username'];
    $_SESSION['admin_pwd'] = $_POST['password'];
}
if($_SESSION['admin_name'] != 'admin' || $_SESSION['admin_pwd'] != 'admin')
{
    header('Location:admin_home.php');          //跳转回登录页面
}

?>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<!--在此处输出,否则会出现乱码或者先出现界面-->
<?php
//输出提示对话框
echo ("<script>alert('欢迎进入系统后台登陆界面!')</script>");

?>
<title>图书管理系统</title>
</head>
```



```
<script language="javascript">
    function show()
    {
        window.alert("还没有在后台系统添加该功能!\n前台已添加了删除|更新|详细信息功能。");
    }
</script>
<h2>
<!-- 网站的主体部分-->
<body bgcolor="black" text="white">
    ....
</body>
</h2>
</html>
```

(该部分参考资料:《LAMPWAMP架构与开发》课件\博客http://blog.sina.com.cn/s/blog_6f49a3c30100p84k.html)

五. 总结

由于该“系统应用”系列文章都是采用C#语言完成,而该篇讲述的PHP仅仅是对以前学习课程的巩固,是对后面的采用C#读取Cookie和清除Cookie等历史记录的前提,所以题目命名是C#.同时确实文章写得太晚,同时思路有些乱,本应该是几篇文章的我一篇写完了,所以请大家见谅!该文章主要来自《LAMPWAMP架构与开发》PPT课件,同时感谢我的CR老师和上面文章的作者.


同时在VC中实现历史记录全面清除-博主gisfarmer,详细介绍了采用VC清除IE临时文件\Cookies\浏览器地址栏历史记录地址\清除表单历史记录,对我的项目帮助很大.<http://blog.csdn.net/gisfarmer/article/details/4327110>

最后希望该文章对大家有所帮助,尊重作者的劳动果实,如果有错误或不足之处,请大家海涵!有点后悔当初学习PHP和数字图像处理时没多写几篇系列文章.

(By:Eastmount 2014-1-26 夜4点 <http://blog.csdn.net/eastmount>)

👍 点赞 1 ☆ 收藏 ➦ 分享 ...



Eastmount  博客专家

发布了450 篇原创文章 · 获赞 6227 · 访问量 496万+

他的留言板

关注