**第23章 PHP中使用会话控制**

**1. 什么是SESSION会话控制**

HTTP是无状态协议，即，HTTP协议没有一个内建机制来维护两个事务之间的状态。当一个用户在请求一个页面后再请求另外一个页面时，HTTP将无法告诉我们，这两个请求来自同一个用户。

会话控制：在网站中，根据一个会话跟踪用户, 从而做到对用户登录的支持，根据个人喜好显示相应的内容，记录用户的行为，实现购物车。

使用$\_SESSION超级全局变量

**2. 理解基本的会话功能**

php的会话，是通过唯一的会话ID来驱动的，会话ID是一个加密的随机数字，由PHP生成，在会话的生命周期内，保存在客户端。它可以保存在用户机器的cookie里，或通过URL在网络上传递。

会话ID，就像一把钥匙，它允许我们注册一些变量（会话变量），内容保存在服务器端的普通文件中，会话ID保存在客户端。如果会话ID在客户端可见，就可以访问该会话保存在服务器上的会话变量。类似于超市的储物柜机制。



**2.1 什么是cookie**

cookie其实是一小段信息，它可以由脚本在客户端机器保存。可以通过发送一个包含特定数据，并且具有如下格式的HTTP标题头，从而在用户机器设置一个cookie。**setcookie(name,value,expire,path,domain,secure)**

当浏览器连接一个URL时，首先要搜索当地保存的cookie。如果有任何与正在连接的URL相关的cookie，浏览器将它提交到服务器。

cookie的工作流程是：由服务器发送给浏览器一个cookie

以后再次访问其同级目录下、子目录下的URL，由浏览器带着cookie过来，服务器检测其cookie信息。

设置： setcookie()

读取：$\_COOKIE[] 超级全局数组

**2.2 通过PHP手动设置cookie**

**函数格式： setcookie(name,value,expire,path,domain,secure)**

**如：**

**setcookie('mycookie','value');**

**当用户访问站点中的下一页或重载当前页时，可以通过**

**名为$\_COOKIE['mycookie']来实现。**

**注意：cookie标题头，必须在发送其他标题头之前发送，否则无效。**

**2.3 在会话中使用cookie**

**一些浏览器不接受cookie，一些用户可能将其浏览器的cookie功能关闭了，PHP会话控制使用cookie/URL双模式。**

**session\_get\_cookie\_params()**

**session\_set\_cookie\_params()**

**2.4 存储会话ID**

**可以设置一个cookie来保存 SESSION\_ID，也可以将SESSION\_ID添加到URL中。**

**3. 实现简单的会话**

使用会话的基本步骤：  
**1.开始一个会话 2，注册会话变量 3.使用会话变量 4. 注销变量并销毁会话**

**session的工作流程：**



**3.1 开始一个会话**

**1）session\_start()**

**在脚本开始的地方使用此函数。该函数将检查是否已经有一个会话ID存在。**

如果不存在，就创建一个，并且使其能够通过超级全局数组$\_SESSION进行访问。

如果已经存在，将这个已经注册的会话变量载入，以便使用。

**2）php.ini中的session.auto\_start选项，自动启动一个会话。**

**3.2 注册一个会话变量**

**$\_SESSION['myvar'] =5; 就可以注册一个会话变量。**

**php.ini中的gc\_maxlifetime指令，默认为1440秒，超过时间，会话将被当成垃圾回收。**

**3.3 使用会话变量**

**if(isset($\_SESSION['myvar']))...**

**3.4 注销变量与销毁会话**

**unset($\_SESSION['myvar'])**

**要销毁所有的会话变量，可以：**

**$\_SESSION = array();**

**使用完一个会话后，首先应该注销所有的变量，然后：session\_destroy();**

**清除会话ID.**

**4. 创建一个简单的会话例子**

page1.php——启动一个会话并注册一个变量

**————————————————————————————————**

**<?php**

**session\_start();**

**$\_SESSION['sess\_var'] = "Hello world!";**

**echo 'the content of $\_SESSION [\'sess\_var\' ] is ' . $\_SESSION['sess\_var'].'<br/>';**

**?>**

**<a href = 'page2.php'> next page</a>**

**————————————————————————————————**



**$\_SESSION['sess\_var'] = "Hello world!"; 该变量已经注册，并设置了值，该变量的最终值，可以被后续页面使用，需要再次调用session\_start()函数，该变量才会被载入。**

**page2.php——访问一个会话并注销它**

**————————————————————————————————**

**<?php**

**session\_start();**

**$\_SESSION['sess\_var'] = "Hello world!";**

**echo 'the content of $\_SESSION [\'sess\_var\' ] is ' . $\_SESSION['sess\_var'].'<br/>';**

**unset($\_SESSION['sess\_var']) ;**

**?>**

**<a href = 'page3.php'> next page</a>**

**————————————————————————————————**

**session\_start();之后，可以获取该变量和所存储的值。使用之后，必须注销该变量。会话仍然存在，但变量被注销。**

**page2.php——结束会话**

**————————————————————————————————**

**<?php**

**session\_start();**

**echo 'the content of $\_SESSION [\'sess\_var\' ] is ' . $\_SESSION['sess\_var'].'<br/>';**

**session\_destroy();**

**?>**

**————————————————————————————————**



**5. 配置会话控制**

**php.ini中有一组会话配置选项，可以设置。**

**6. 通过会话控制，实现身份验证举例：**

**3个脚本，第一个是 auth\_main.php 它提供一个登录表单，并为站点成员提供了身份验证。第二个是members\_only.php 只向成功登录的成员显示信息。第三个是logout.php 用户退出登录。**

**authmain.php——身份验证应用程序的主体部分**

**————————————————————————————————**

**<?php**

**session\_start();**

**if ((!isset($\_POST['userid'])) && (!isset($\_POST['password']))) {**

**//if theuser has just tried to log in**

**$userid = $\_POST['userid'];**

**$password = $\_POST['password'];**

**// connect to mysql**

**$db\_conn = mysqli\_connect("localhost", "webauth", "webauth",'auth');**

**if(mysqli\_connect\_errno()) {**

**echo 'Connection to database failed : ' . mysqli\_connect\_error();**

**exit;**

**}**

**$query = "select \* from authorized\_users where name = '$userid' and password = sha1('$password') " ;**

**$result = $db\_conn->query($query);**

**if($result>num\_rows) {**

**// if they are in the database register the user id**

**$\_SESSION['valid\_user'] = $userid ;**

**}**

**$db\_conn->close();**

**}**

**?>**

**<html lang="en">**

**<head>**

**<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">**

**<title>Document</title>**

**</head>**

**<body>**

**<h1>Home page</h1>**

**<?php**

**if(isset($\_SESSION['valid\_user'])){**

**echo "you are logged in as: ".$\_SESSION['valid\_user'] ."<br/>" ;**

**echo '<a href = "logout.php"> Log out </a><br/>';**

**} else {**

**if(isset($userid)){**

**// if they've tried and failed to log in**

**echo 'could not log you in . <br/>' ;**

**} else {**

**//thery have not tried to log in yet or have logged out**

**echo 'you are not logged in .<br/>' ;**

**}**

**?>**

**<form action="authmain.php" method = "post">**

**<table>**

**<tr>**

**<td>Userid:</td>**

**<td><input type="text" name="userid"></td>**

**</tr>**

**<tr>**

**<td>Password:</td>**

**<td><input type="password" name="password"></td>**

**</tr>**

**<tr>**

**<td colspan="2" align ="center"><input type="submit" value = "log in"></td>**

**</tr>**

**</table>**

**</form>**

**<?php**

**}**

**?>**

**<br/>**

**<a href="members\_only.php">Members section</a>**

**</body>**

**</html>**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**程序功能：如果某人登录成功，就注册一个 $\_SESSION['valid\_user'] 的会话变量，该变量包含用户的ID。**

**实现步骤：表单点击提交按钮，脚本将从顶端开始重新执行。这时，我们有了用于验证的用户名和密码， $\_POST['userid']; $\_POST['password'];中。**

**进入身份验证模块**

**连接到mysql数据库并检查用户ID和密码，如果它们与数据库中的数据匹配，将创建变量$\_SESSION['valid\_user'] ，它包含该用户的ID，因此，我们知道是谁登录进来了，并对其进行跟踪。**

**已经知道该用户是谁，则提供退出的选项**

**如果登录失败，提供一个错误信息。**

members\_only.php —— 网站的有效用户检查部分，可确定是否是有效成员

**————————————————————————————————**

**<?php**

**session\_start();**

**echo '<h1>Members only</h1>';**

**//check session variable**

**if(isset($\_SESSION['valid\_user'])){**

**echo '<p>you are logged in as ' .$\_SESSION['valid\_user'] . '</p>';**

**echo '<p>members only content goes here</p>';**

**} else{**

**echo '<p>you are not logged in .'</p>';**

**echo '<p>only logged in members may see this page.</p>';**

**}**

**echo '<a href="authmain.php">back to main page</a>';**

**?>**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**启动会话，检查session变量是否被设置，显示成员内容，否则，告诉他尚未通过身份验证。**

**logout.php——该脚本注销会话变量，并销毁会话**

**————————————————————————————————**

**<?php**

**session\_start();**

**$old\_user = $\_SESSION['valid\_user'] ;**

**unset($\_SESSION['valid\_user'] );**

**session\_destroy();**

**?>**

**<html>**

**<body>**

**<h1>log out</h1>**

**<?php**

**if(!empty($old\_user)){**

**echo "logged out.<br/>";**

**}else{**

**echo 'you were not logged in ,and so have not been logged out. <br/>';**

**?>**

**<a hrer="authmain.php'>back to main page</a>**

**</body>**

**</html>**

**————————————————————————————————**