

## 060010606 DSE12 Open Source Web Technology

*The **basic objective** of the material is to supplement teaching and discussion in the classroom. Student is required to go for extra reading in the subject through library work. Document contains syllabus's chapter wise document and practise list which student have to submit as per instruction*

### Unit: 4 State Management

- ◆ Overview and Techniques
- ◆ Dealing with Cookies
- ◆ Session Management
- ◆ Advantages and Disadvantages of State Management Techniques
- ◆ Global Variables

#### Cookies

A cookie is a small file with the maximum size of 4KB that the web server stores on the client computer. They are typically used to keeping track of information such as a username that the site can retrieve to personalize the page when the user visits the website next time. A cookie can only be read from the domain that it has been issued from. Cookies are usually set in an HTTP header but JavaScript can also set a cookie directly on a browser.

**Cookies** are **transferred** between **server** and **client** according to **http**.

PHP supports http cookies.

Cookies can also be thought of as tickets used to identify clients and their orders.

#### **Uses of cookie**

Cookies are often used to perform following tasks:

- **Session management:** Cookies are widely used to manage user sessions. For example, when you use an online shopping cart, you keep adding items in the cart and finally when you checkout, all of those items are added to the list of items you have purchased. This can be achieved using cookies.
- **User identification:** Once a user visits a webpage, using cookies, that user can be remembered. And later on, depending upon the search/visit pattern of the user, content which the user likely to be visited are served. A good example of this is 'Retargetting'. A concept used in online marketing, where depending upon the user's choice of content, advertisements of the relevant product, which the user may buy, are served.
- **Tracking / Analytics:** Cookies are used to track the user. Which, in turn, is used to analyse and serve various kind of data of great value, like location,

## Babu Madhav Institute of Information Technology, UTU

technologies (e.g. browser, OS) form where the user visited, how long (s)he stayed on various pages etc.

To set a cookie in PHP, the **setcookie()** function is used. The setcookie() function needs to be called prior to any output generated by the script otherwise the cookie will not be set.

Parameter	Description
<b>name</b>	(Required). Specifies the name of the cookie
<b>value</b>	(Required). Specifies the value of the cookie
<b>expire</b>	(Optional). Specifies when the cookie expires. e.g. <b>time()+3600*24*30</b> will set the cookie to expire in <b>30 days</b> . If this parameter is not set, the cookie will expire at the end of the session (when the browser closes).
<b>path</b>	(Optional). Specifies the server path of the cookie.* If set to <b>"/"</b> , the cookie will be available within the entire domain. If set to <b>"/phptest/"</b> , the cookie will only be available within the test directory and all sub-directories of <b>phptest</b> . The default value is the current directory that the cookie is being set in.
<b>domain</b>	(Optional). Specifies the domain name of the cookie. To make the cookie available on all subdomains of example.com then you'd set it to ".example.com". Setting it to <a href="http://www.example.com">www.example.com</a> will make the cookie only available in the www subdomain
<b>secure</b>	(Optional). Specifies whether or not the cookie should only be transmitted over a secure <b>HTTPS</b> connection. <b>TRUE</b> indicates that the <u>cookie will only be set</u> if a secure connection exists. Default is <b>FALSE</b> . set to '1' to transmit in HTTPS

### Creating Cookie:

1. setcookie("Subject", "Open Source Web Technology", time()+60\*60\*24\*1);
2. setcookie("AnotherCookie", \$value, time()+3600);

### Reading Cookie:

## Babu Madhav Institute of Information Technology, UTU

To access a cookie received from a client, use the PHP `$_COOKIE` superglobal array. It is an associative array that contains a record of all the cookies values sent by the browser in the current request. The records are stored as a list where cookie name is used as the key. `$_COOKIE` array can contain depends on the memory size set in `php.ini`. The default value is 1GB.

```
<?php
foreach ($_COOKIE as $key=>$val) {
    print $key . " => " . $val . "<br/>";
}
?>
```

It is always advisable to check whether a cookie is set or not before accessing its value. Therefore to check whether a cookie is set or not, the PHP `isset()` function is used.

```
if(isset($_COOKIE["Auction_Item"])){}
```

### **Deleting Cookie:**

The `setcookie()` function can be used to delete a cookie. For deleting a cookie, the `setcookie()` function is called by passing the cookie name and other arguments or empty strings but however this time, the expiration date is required to be set in the past

```
<?php
setcookie("Subject", "", time()-60);
or
setcookie($cookie_name,"",time() - 3600);
?>
```

### **Note:**

1. If the expiration time of the cookie is set to 0, or omitted, the cookie will expire at the end of the session i.e. when the browser closes.
2. The same path, domain, and other arguments should be passed that were used to create the cookie in order to ensure that the correct cookie is deleted.

### **Update Cookie:**

Actually, there is not a way to update a cookie. Just set (again) this cookie using `setcookie`.

```
$cookie_name = 'Subject;;
```

```
$cookie_value = 'Subject12';
```

```
setcookie($cookie_name,$cookie_value,time() + (86400 * 30), '/'); // 86400 = 1 day
```

### **Where the cookie store?**

- *Depends on the browser...*
- *e.g., firefox/mozilla under /home/a*
  - *Look for cookies.txt in .mozilla directory*

## Babu Madhav Institute of Information Technology, UTU

- Usually under:
  - /home/a      /.mozilla/firefox/asdkfljy.default
- Cookie is stored only if there is an expiry date
- Otherwise it is deleted when leaving browser
- Persistent only if an expiry date is set

Hidden Form Field	Query String	Cookie	Session

--	--	--	--

## **Session:**

- A session is a global variable stored on the server.
- Each session is assigned a unique id which is used to retrieve stored values.
- Whenever a session is created, a cookie containing the unique session id is stored on the user's computer and returned with every request to the server. If the client browser does not support cookies, the unique php session id is displayed in the URL
- Sessions have the capacity to store relatively large data compared to cookies.
- The session values are automatically deleted when the browser is closed. If you want to store the values permanently, then you should store them in the database.
- Just like the \$\_COOKIE array variable, session variables are stored in the \$\_SESSION array variable. Just like cookies, the session must be started before any HTML tags.
- You want to store important information such as the user id more securely on the server where malicious users cannot temper with them.
- You want to pass values from one page to another.
- You want the alternative to cookies on browsers that do not support cookies.
- You want to store global variables in an efficient and more secure way compared to passing them in the URL
- You are developing an application such as a shopping cart that has to temporary store information with a capacity larger than 4KB.

## **Why and when to use Sessions?**

- You want to store important information such as the user id more securely on the server where malicious users cannot temper with them.
- You want to pass values from one page to another.
- You want the alternative to cookies on browsers that do not support cookies.
- You want to store global variables in an efficient and more secure way compared to passing them in the URL
- You are developing an application such as a shopping cart that has to temporary store information with a capacity larger than 4KB.

## **Creating a Session**

```
<?php

session_start(); //start the PHP_session function

if(isset($_SESSION['page_count']))
{
    $_SESSION['page_count'] += 1;
}
else
{
    $_SESSION['page_count'] = 1;
}
echo 'You are visitor number ' . $_SESSION['page_count'];

?>
```

## **Destroying Session Variables**

The session\_destroy() function is used to destroy the whole Php session variables.

If you want to destroy only a session single item, you use the unset() function.

Session\_destroy removes all the session data including cookies associated with the session.

Unset only frees the individual session variables

The code below illustrates how to use both methods.

```
<?php

session_destroy(); //destroy entire session

?>
<?php

unset($_SESSION['product']); //destroy product session item

?>
```