Web Applications Development I

Lecture 9: Cookies and Sessions

Outline

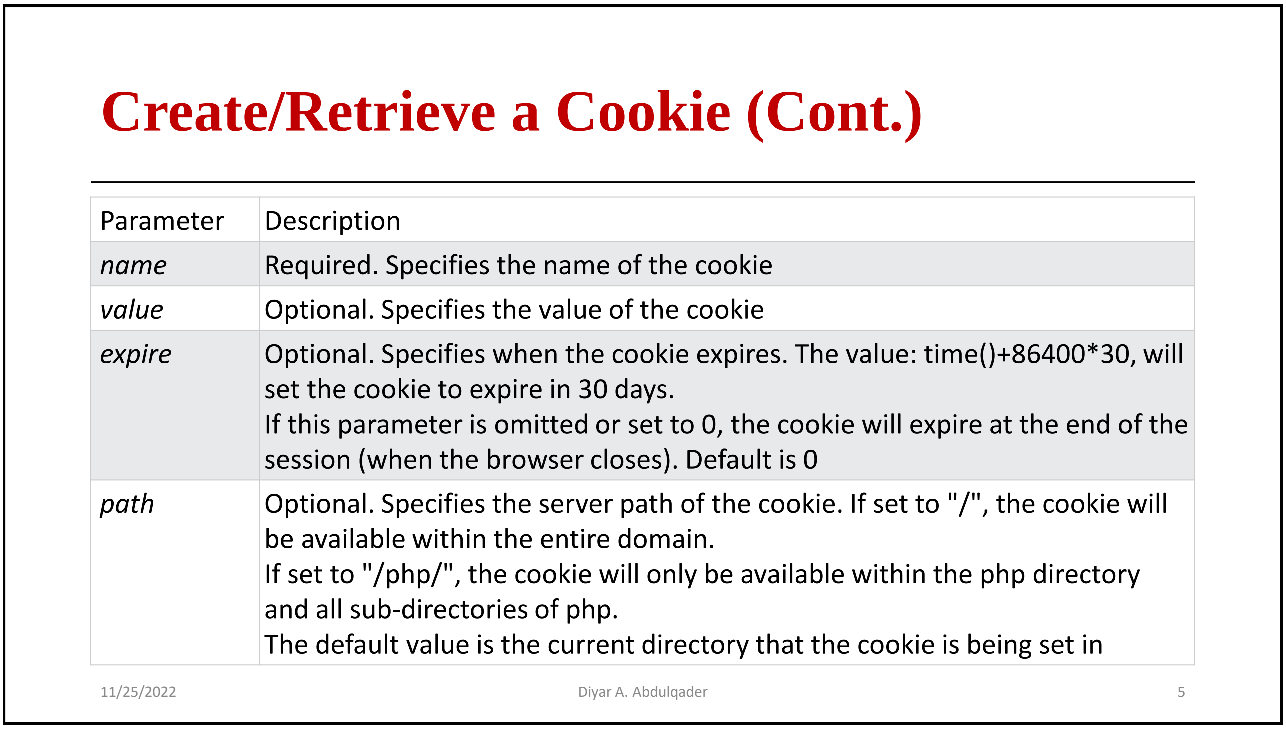
- ➢What is a Cookie?
- ➢Create/Retrieve a Cookie
- ➢Modify/Delete a Cookie
- ➢What is a PHP Session?
- ➢Start a PHP Session
- ➢Get PHP Session Variable Values
- ➢Modify or Destroy a PHP Session Variable
- ➢Sessions without cookies
- ➢Page Redirection

What is a Cookie?

- A cookie is often used to identify a user.
- A cookie is a small file that the server embeds on the user's computer.
- Each time the same computer requests a page with a browser, it will send the cookie too.
- With PHP, you can both create and retrieve cookie values.

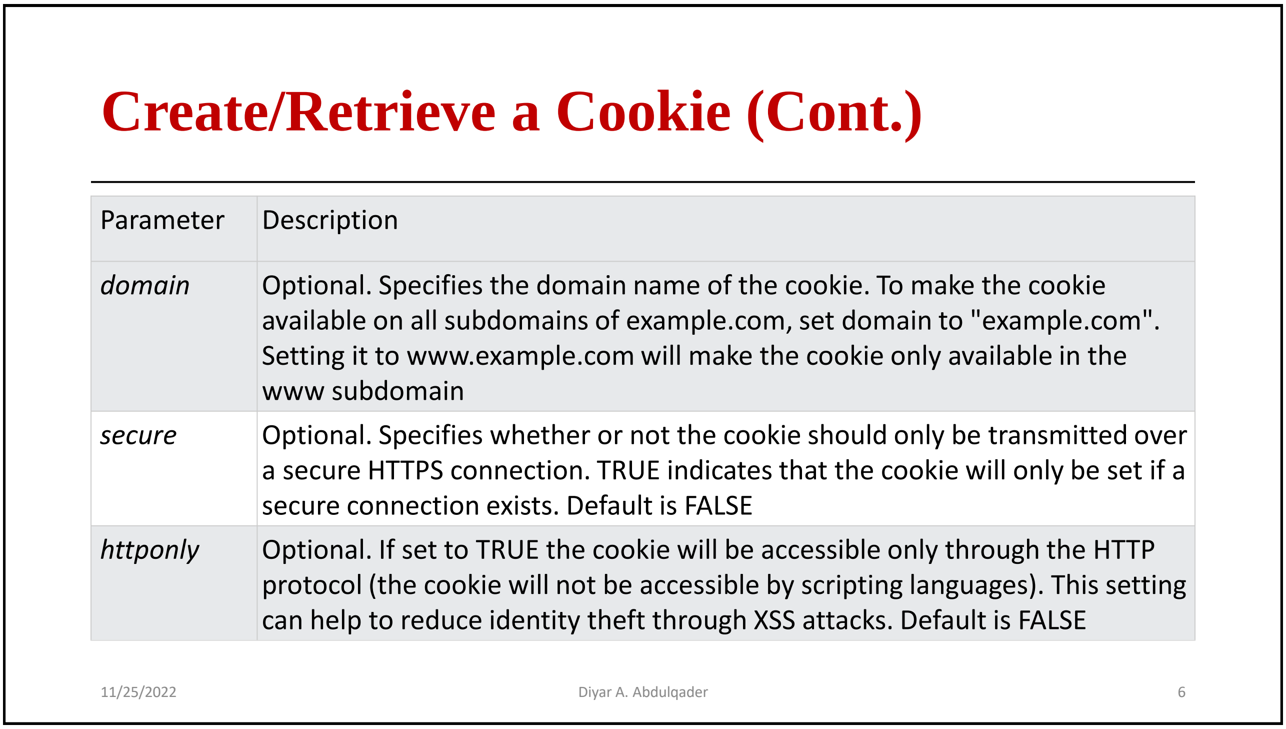
Create/Retrieve a Cookie

- A cookie is **created** with the **setcookie**() function.
- The setcookie() function defines a cookie to be sent along with the rest of the HTTP headers.
- The name of the cookie is automatically assigned to a variable of the same name.
- For example, if a cookie was sent with the name "user", a variable is automatically created called $user, containing the cookie value.
- **Syntax**:
  setcookie(*name, value, expire, path, domain, secure, httponly*);
- Only the name parameter is required. All other parameters are optional.



Create/Retrieve a Cookie (Cont.)

Parameter	Description
name	Required. Specifies the name of the cookie
value	Optional. Specifies the value of the cookie
expire	Optional. Specifies when the cookie expires. The value: time()+86400*30, will set the cookie to expire in 30 days. If this parameter is omitted or set to 0, the cookie will expire at the end of the session (when the browser closes). Default is 0
path	Optional. Specifies the server path of the cookie. If set to "/", the cookie will be available within the entire domain. If set to "/php/", the cookie will only be available within the php directory and all sub-directories of php. The default value is the current directory that the cookie is being set in



Create/Retrieve a Cookie (Cont.)

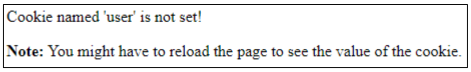
Parameter	Description
domain	Optional. Specifies the domain name of the cookie. To make the cookie available on all subdomains of example.com, set domain to "example.com". Setting it to www.example.com will make the cookie only available in the www subdomain
secure	Optional. Specifies whether or not the cookie should only be transmitted over a secure HTTPS connection. TRUE indicates that the cookie will only be set if a secure connection exists. Default is FALSE
httponly	Optional. If set to TRUE the cookie will be accessible only through the HTTP protocol (the cookie will not be accessible by scripting languages). This setting can help to reduce identity theft through XSS attacks. Default is FALSE

Create/Retrieve a Cookie (Cont.)

- We then **retrieve** the value of the cookie "user" (using the global variable $_COOKIE).
- We also use the isset() function to find out if the cookie is set.

```
<?php
$cookie_name = "user";
$cookie_value = "John
Doe";
setcookie($cookie_name,
$cookie_value, time() +
(86400 * 30), "/"); //
86400 = 1 day
?>
<html>
<body>
```

```
<?php
if(!isset($_COOKIE[$cookie_name])) {
  echo "Cookie named '" . $cookie_name . "' is not
set!";
} else {
  echo "Cookie '" . $cookie_name . "' is set!<br>";
  echo "Value is: " . $_COOKIE[$cookie_name];
}
?>
</body>
</html>
```



```
Cookie named 'user' is not set!

Note: You might have to reload the page to see the value of the cookie.
```

Modify/Delete a Cookie

- **Note**: The setcookie() function must appear BEFORE the <html> tag.
- To **modify** a cookie, just set (again) the cookie using the setcookie() function:
- To **delete** a cookie, use the setcookie() function with an expiration date in the past:

```
<?php
// set the expiration date to one hour ago
setcookie("user", "", time() - 3600);
?>
<html>
<body>
<?php echo "Cookie 'user' is deleted."; ?>
</body>
</html>
```

More Example

Pageone.php

```
<?php
setcookie("user", "diyar",
time() + (60 * 5), "/");

setcookie("age", "21", time()
+ (60 * 5), "/");
?>
<html>
<body>
<a href="pagetwo.php">Page
Two</a>
</body>
</html>
```

Output

Page Two

Pagetwo.php

```
<html>
<body>
<?php
if(!isset($_COOKIE["user"])) {
    echo "Cookie named user is not set!";
} else {
    echo "Cookie user is set!<br>";
    echo "User is: " . $_COOKIE["user"];
    echo "Age is: " . $_COOKIE["age"];
}
?>
</body>
</html>
```

Output

```
Cookie user is set!
User is: diyar
Ageis: diyar
```

What is a PHP Session?

- A session is a way to store information (in variables) to be used across multiple pages.
- Unlike a cookie, the information is not stored on the users computer.
- A session creates a file in a temporary directory on the server where registered session variables and their values are stored.
- This data will be available to all pages on the site during that visit (e.g. username, favorite color, etc).
- By default, session variables last until the user closes the browser.
- The location of the temporary file is determined by a setting in the **php.ini** file called **session.save_path**.
- Before using any session variable make sure you have setup this path.

Start a PHP Session

- When a session is started following things happen:
  - PHP first creates a unique identifier for that particular session which is a random string of 32 hexadecimal numbers such as 3c7foj34c3jj973hjkop2fc937e3443.
  - A cookie called **PHPSESSID** is automatically sent to the user's computer to store unique session identification string.
  - A file is automatically created on the server in the designated temporary directory and bears the name of the unique identifier prefixed by sess_ sess_3c7foj34c3jj973hjkop2fc937e3443.
- A session ends when the user loses the browser or after leaving the site, the server will terminate the session after a predetermined period of time, commonly 30 minutes duration.

Start a PHP Session (Cont.)

- A session is started with the session_start() function.
- Session variables are set with the PHP global variable: $_SESSION.
- Note: The session_start() function must be the very first thing in your document. Before any HTML tags.

session1.php

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>
```

```
<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>
</body></html>
```

Get PHP Session Variable Values

- We create another page called "session2.php".
- From this page, we will access the session information we set on the first page ("session1.php").
- Notice that session variables are not passed individually to each new page,
- instead they are retrieved from the session we open at the beginning of each page (session_start()).
- Also notice that all session variable values are stored in the global $_SESSION variable.

Get PHP Session Variable Values (Cont.)

```
<?php session_start(); ?>
<!DOCTYPE html>
<html>
<body>
<?php
// Echo session variables that were set on previous page
echo "Favorite color is " . $_SESSION["favcolor"] . ".<br>";
echo "Favorite animal is " . $_SESSION["favanimal"] . ".";
?>
</body>
</html>
```

Modify or Destroy a PHP Session Variable

- To change a session variable, just overwrite it: $_SESSION["favcolor"] = "yellow";
- To remove all global session variables and destroy the session, use `session_unset()` and `session_destroy()`.
- Example:

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>
```

```
<?php
// remove all session variables
session_unset();
// destroy the session
session_destroy();
?>
</body>
</html>
```

Sessions without cookies

- There may be a case when a user does not allow to store cookies on their machine. So there is another method to send session ID to the browser.
- Alternatively, you can use the session_id which is defined if the session started.
- Thus, you can embed it unconditionally into URLs.
- session_id() is used to get or set the session id for the current session.
- To set session_id() needs to be called before session_start().
- session_id() returns the session id for the current session or the empty string ("") if there is no current session (no current session id exists). On failure, false is returned.

Sessions without cookies (Cont.)

Pageone.php

```
<?php
session_start();
$_SESSION["user"]= "diyar";
?>
<html>
<body>
<a href="pagetwo.php
?phpsid=<?php echo
htmlspecialchars(session_id()
); ?>">Page Two</a>
</body>
</html>
```

Output

```
Page Two
```

Pagetwo.php

```
<?php
session_id($_GET["phpsid"]);
session_start();
echo "Your Sesion user is: " .
$_SESSION['user'];
?>
<html>
<body>

</body>
</html>
```

Output

```
Your Sesion user is:  diyar
```

Page Redirection

- You can redirect your user to some other page.
- Example:

```
<?php header("Location: http://www.example.com/"); ?>
```

- Please note that Location starts with capital L, some browsers might not redirect if small l is used.
- Furthermore, you can add some redirection interval by using the following code:

```
<?php
header("Refresh: 5; url=http://www.example.com"); //will redirect after 5 seconds
 ?>
```

- redirect a web page using JavaScript

```
<?php echo "<script>window.location.href='test.php';</script>"; ?>
```