Chapter 12: Cookies & Sessions

PHP and MySQL • CIS 86 • Mission College

Tonight's agenda: Chapter 12

- Cookies
  - Setting → Reading → Deleting
- Sessions
  - Starting → Reading → Ending → Destroying

We will not cover this topic:

- HTTP Authenticaion
  - Storing user names and passwords
  - Salting passwords

Example

- Hit counter should count unique site visitors, not just page counts.
- We can prevent the same visitor from triggering the hit counter multiple times by setting a cookie that indicates they've already visited the site in the past.

Communication between web pages?

- We call the problem of communicating between one page and another a problem of *maintaining state.*
- There are five ways we can communicate information on one page to the next page the user visits:
  1. URL tokens
  2. Hidden form fields
  3. Cookies
  4. Sessions
  5. HTML5 Local Storage (not usable by PHP)

HTML5 Local Storage

- HTML5 Local Storage lets you store small amounts of information on a user's computer disk.
- Because the information is stored on the local computer's disk, it is not available to PHP code, which runs on the server.
- But it is available to JavaScript, which runs on the client computer.

URL Tokens

- We can add tokens like `?page=17` or `?visited=yes` to the URLs of pages we visit.
- Problem: we have to manually keep track of which tokens we will attach to the URL at any given point in their web visit. We don’t know what order they will visit the pages on our site.

Hidden Form Fields

- We can use hidden form fields when the user is on a page that has a form.
- See page 275 (3rd Edition, Chapter 11)
- Page ??? (4th Edition, Chapter 10)
- Example: wizard (a form that has several pages).
- Problem: we can't use hidden form fields if the page does not have a form or if the page is not itself a form handler.

Cookies

- Cookies are small pieces of information that the server can request be placed on the user's computer.
- This is part of the HTTP protocol.
- Problem: The amount of information that can be stored is limited, and the number of cookies you can store is limited.
- Problem: A cookie write can only be requested when the web page headers are sent from the server to the client. A cookie cannot be written at any other time during display of a web page.

Sessions

- A session is a piece of information stored on the server paired with an identifying piece of information stored as a cookie on the client (user) computer.
- Sessions have the advantage that they can store much more data on the server than they can on the user's computer.
- Sessions have the advantage that they store only a tiny piece of information on the user's computer.
- Problem: the program may store sensitive information on the server in the cloud, and it could become compromised.

Cookie transactions

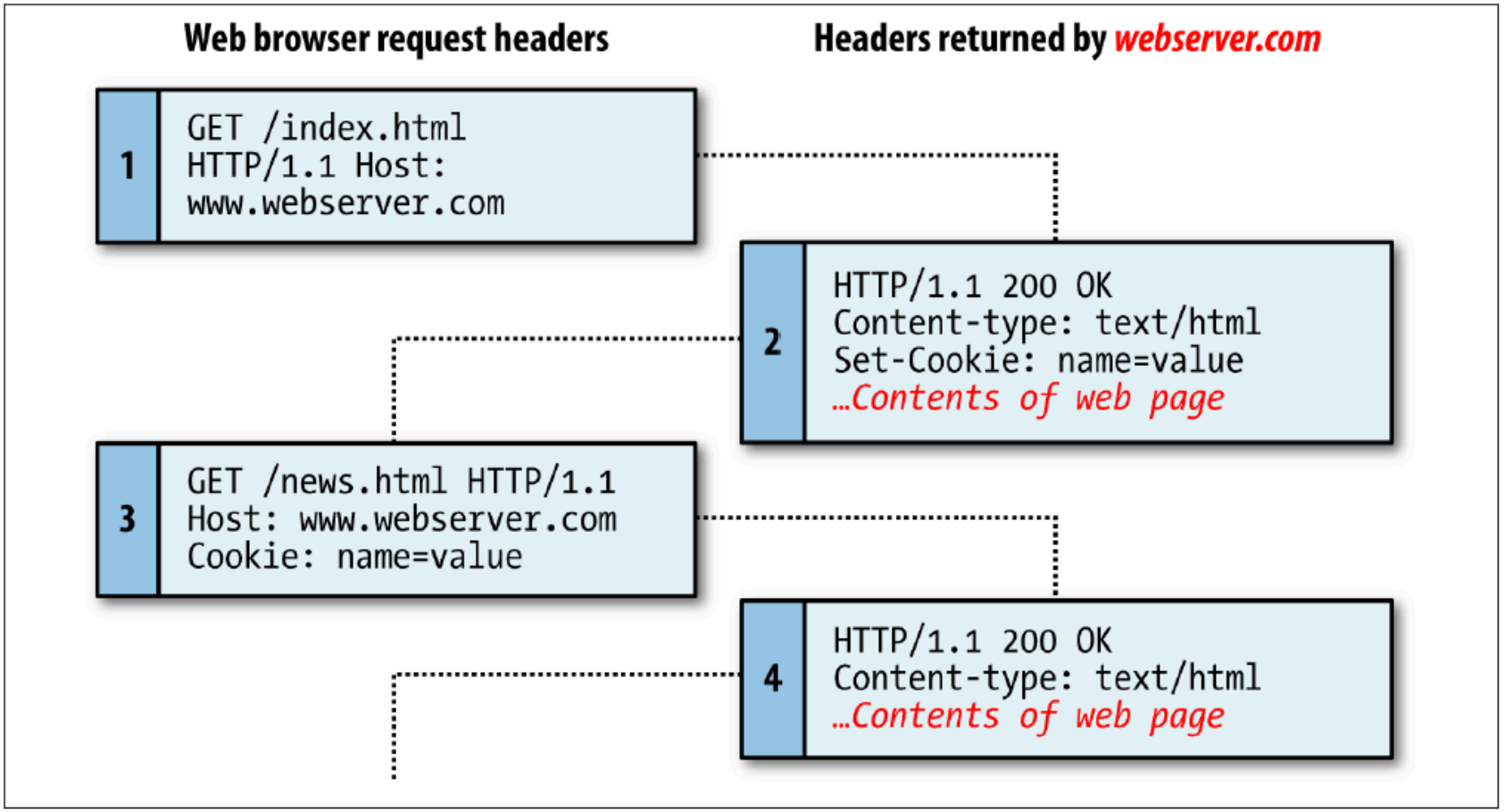

Figure 13-1. A browser/server request/response dialog with cookies

Cookie parameters

Parameter	Required?	Description
name	required	a key for looking up the cookie
value	required	the value or contents of the cookie
expire	optional	UNIX timestamp
path	optional	folders where the cookie applies
domain	optional	domains and subdomains where the cookie applies
secure	optional	whether the cookie must use a secure https:// connection
httponly	optional	whether cookie is available from http (PHP) only

Setting a cookie

```
setcookie (
      'visited',
      'yes',
      time() + 60 * 60 * 24 * 7,
      '/');
```

- The cookie name is `visited`
- The cookie value is `yes`
- The expiration date is 7 days from now
- The path is the entire web site (root)

Reading a cookie

```
if (isset ($_COOKIE['visited'])) {
    $visited = $_COOKIE['visited'];
}
else {
    $visited = 'no';
}
```

Note this uses the `$_COOKIE` system array, which is an associative array similar to `$_GET`, `$_POST`, and `$_FILES`.

Deleting a cookie

- To delete a cookie, set it to a date in the past.

```
setcookie (
      'visited',
      'no',
      time() - 2592000,
      '/');
```

- 2592000 seconds is one month.

Handy Numbers

- 100,000 seconds is a little more than a day.
- 1,000,000 seconds is a little more than 10 days. (I use this number in place of one week.)
- 3,000,000 seconds is about a month.

Starting a session

- Page 299

```
session_start();
```

Setting session variables

```
$_SESSION['username'] = $un;
$_SESSION['password'] = $pw;
```

Retrieving session variables

```
$un = $_SESSION['username'];
$pw = $_SESSION['password'];
```

Ending a session

```
session_destroy();
```

Destroying a session

Or, to be more certain, you can empty the session array and remove its cookie also:

```
session_start();
$_SESSION = array ();
setcookie (session_name(), “”,
    time() – 2592000, “/”);
session_destroy();
```