HTTP Session Management
CS183 Hypermedia and the Web

Cookies

A mechanism for adding state to HTTP requests. Permits name, value pairs to be preserved across multiple HTTP requests.

Basic approach:

Server sends a Set-Cookie HTTP header in its response. Value of the header includes name, value pairs. The client stores the name, value pairs, and proactively sends them to the server (in the Cookie header) with every request thereafter.

How Cookies are set in PHP:

Use built-in setcookie() function. Most simple form is:

setcookie(name, value)

It is also possible to specify when the cookie will expire, the URL path and domain for which it applies (defaults to those of request URI).

Accessing Cookie values:

$local_variable = $_COOKIE[name]

To delete a cookie, use setcookie with the same cookie name, and an expiration value in the past.

Sessions

One problem with cookies is that it is possible for browsers to refuse to accept cookies. Additionally, it adds network overhead to send lots of information back and forth. There are also limits to the amount of information that can be sent, and some information you just don’t want to save on the client’s computer.

The solution: store session information on the server, and have the client only store an identifier for its information as stored on the server. The identifier is known as a session ID. The session ID is stored using a cookie (can be passed as a GET parameter as well), and the server then uses the session ID to retrieve the information it has stored on the server.

To start a session:
session_start()

Creates a session Cookie.

Then, can create, access, and modify session variables:

$_SESSION[session_var_name] = value;

$local_variable = $_SESSION[session_var_name];

Can check if session variable is set by using isset();

To end a session:

session_destroy();