



COMP 4021

Internet Computing

PHP Sessions

Dr. Kenneth LEUNG

Slides created by Dr. David ROSSITER

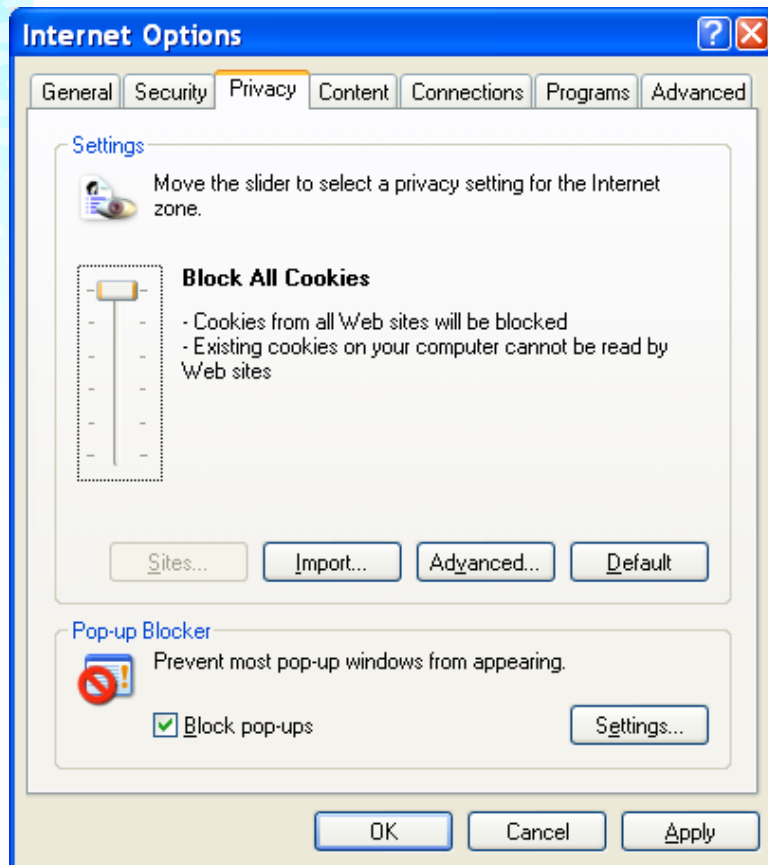
What is a 'Session'?

- HTTP is a *stateless* protocol - each request/response is self-contained, there is no 'memory' from one access to another
- If there is no state then the browser has to send authentication information (i.e. name and password) every single time the user accesses a page on that web site
- For example, this is what happens every time you access a page on the COMP4021 web site
- Using a state is crucial for many Web applications

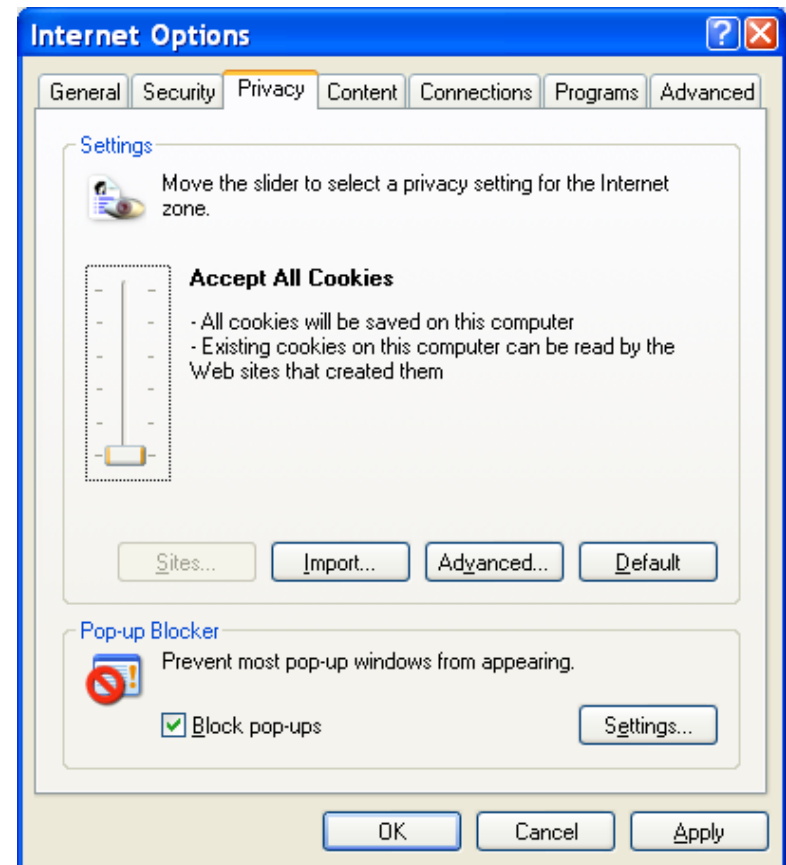
Using Cookies

- Traditionally, one way to create a kind of state is to use cookies
- However, a user may block cookies on their computer
- Alternatively, perhaps a browser is being used which does not support cookies (i.e. a browser with high security settings, some mobile phone browsers)

Example Security Settings (IE)



...



High 'privacy' – all cookies blocked

Low 'privacy' - all cookies accepted

Session IDs

- PHP (and other server side languages) can use sessions that make states easy to handle so that you can successfully preserve and use variables among web pages
- Every session will have a unique id, although this is invisible to the code
- Session ids are stored on the server
- To activate a session, include `session_start()`; at the beginning of your script, before PHP has done any printing

Registering a Session Variable

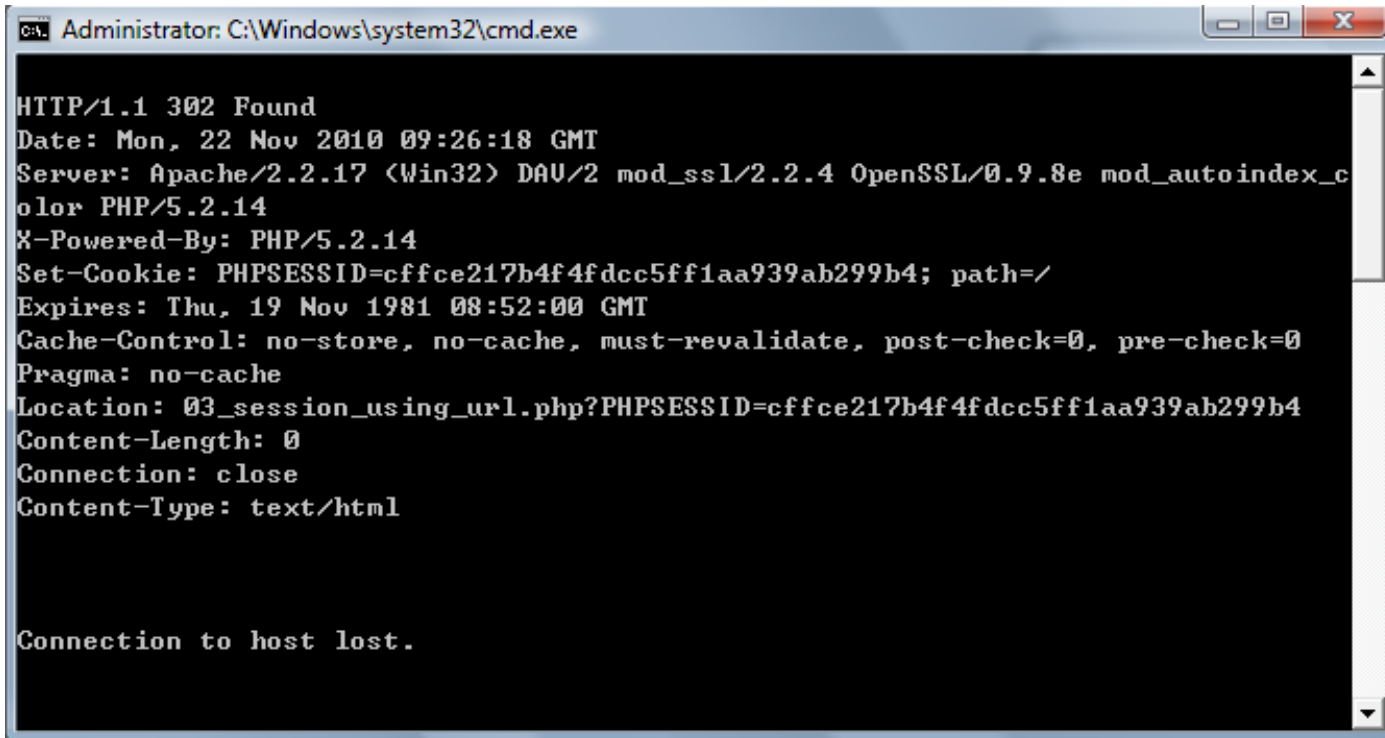
- This code transfers session ID using cookies
- If cookies don't work, then this code doesn't work

```
<?php
    session_start();

    if ( !isset( $_SESSION['count'] ) ) { // If does not exist, create it
        $_SESSION['count'] = 0;
    } else {
        $_SESSION['count']++;           // Increment count by one
    }
    echo "You have visited here ".
        $_SESSION['count']." time(s).";
?>
```

Example Output using PHP

Session ID is passed using cookie



```
Administrator: C:\Windows\system32\cmd.exe

HTTP/1.1 302 Found
Date: Mon, 22 Nov 2010 09:26:18 GMT
Server: Apache/2.2.17 (Win32) DAV/2 mod_ssl/2.2.4 OpenSSL/0.9.8e mod_autoindex_color PHP/5.2.14
X-Powered-By: PHP/5.2.14
Set-Cookie: PHPSESSID=cffce217b4f4fdcc5ff1aa939ab299b4; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Location: 03_session_using_url.php?PHPSESSID=cffce217b4f4fdcc5ff1aa939ab299b4
Content-Length: 0
Connection: close
Content-Type: text/html

Connection to host lost.
```

Tell the browser to re-load the same program again, this time passing the Session ID to the program so that it can access and manipulate it

Unregistering a Session Variable

```
<?php
    session_start();

    // Dump the session variable
    unset($_SESSION['count']) ;
?>
```


Passing the Session ID

- There are two methods to propagate a session ID
 1. Cookies
 2. GET method (passed as a URL parameter)
- Cookies are best, but they are not always available
- There is an alternative way which uses the URL
- The session module supports both methods

Example Using URL

- This code transfers session ID using the GET method parameter in the URL

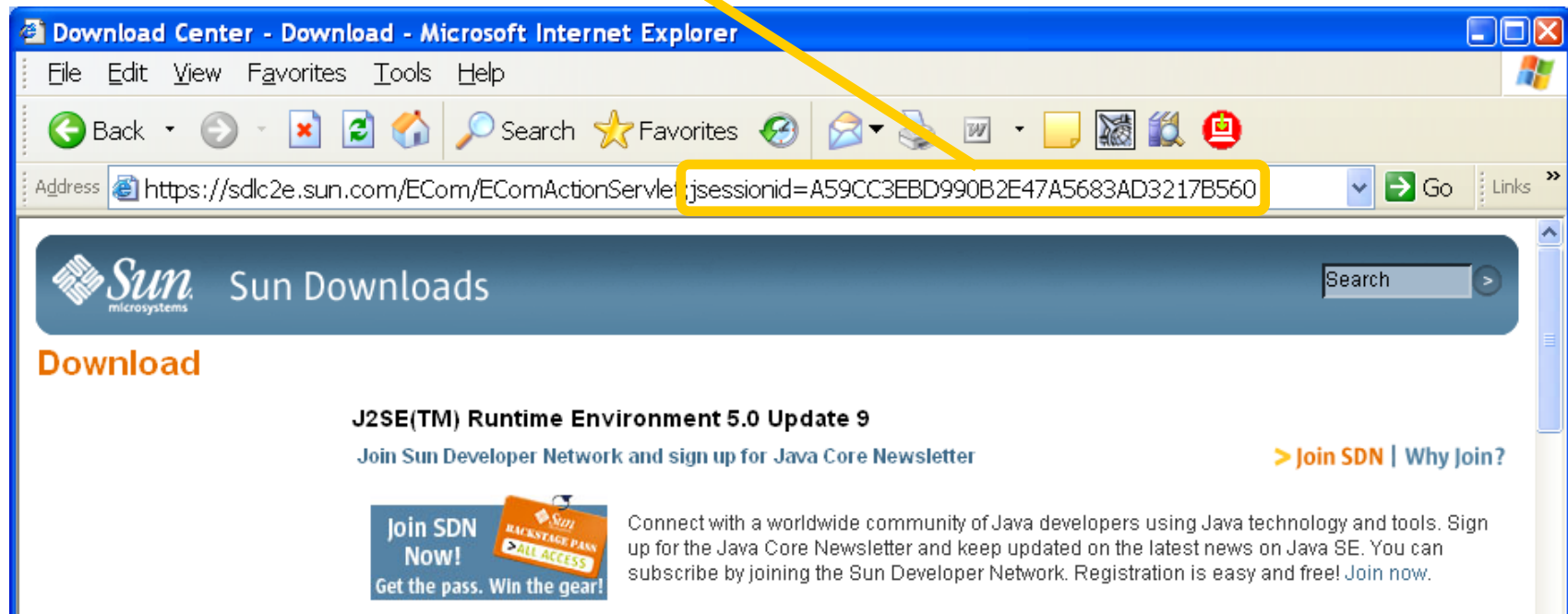
```
<?php
// This will check for both cookie and URL methods
session_start();

if (!isset($_SESSION['count'])) {
    header("Location: 03_session_using_url.php?".SID);
    $_SESSION['count'] = 0;
} else {
    echo "You have visited here ".$_SESSION['count']." time(s)";
    $_SESSION['count']++; }
?>
```

03_session_using_url.php

Using URL to pass Session ID

- An example of using URL to pass session ID in a website developed using Java



Generate Session ID

- If the cookie has no session information, the PHP program will generate a new session ID
 - E.g. you can browse *03_session_using_url.php* without adding the session ID in the URL
- Then the PHP program will tell the browser to reload itself with the new session ID