COMP 4021

Internet Computing

PHP Sessions

What is a ‘Session’?

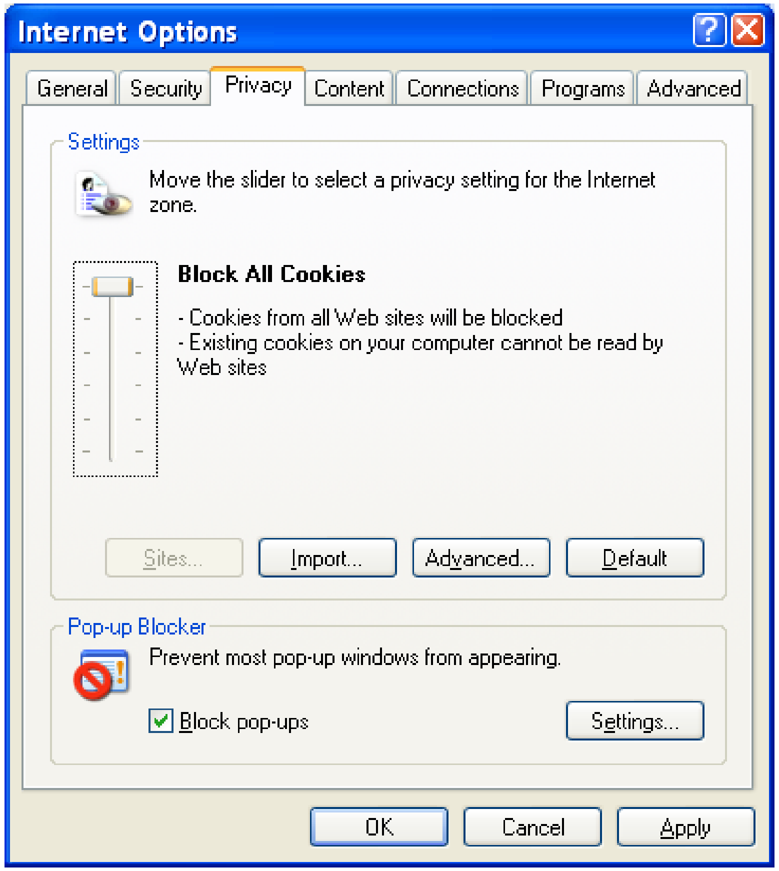
- HTTP is a *stateless* protocol - requests/responses are independent; there is no ‘memory’ from one access to another
- If there is no state then the browser has to send authentication information (i.e. name and password) every time the user accesses a page on that web site
- “States” are crucial for many Web applications

-- We mention this when we study cookies --

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)


Internet Options

General | Security | Privacy | Content | Connections | Programs | Advanced

Settings

Move the slider to select a privacy setting for the Internet zone.

Block All Cookies

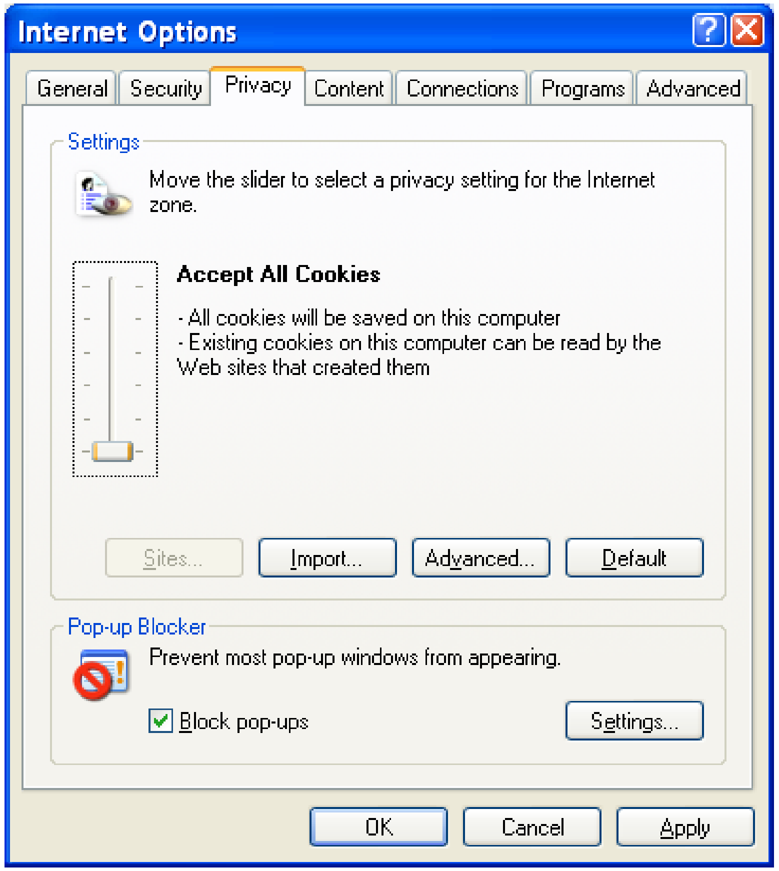
- Cookies from all Web sites will be blocked
- Existing cookies on your computer cannot be read by Web sites

Pop-up Blocker

Prevent most pop-up windows from appearing.

✓ Block pop-ups

. . .


Internet Options

General | Security | Privacy | Content | Connections | Programs | Advanced

Settings

Move the slider to select a privacy setting for the Internet zone.

Accept All Cookies

- All cookies will be saved on this computer
- Existing cookies on this computer can be read by the Web sites that created them

Pop-up Blocker

Prevent most pop-up windows from appearing.

✓ Block pop-ups

High 'privacy' – all cookies blocked

Low 'privacy' - all cookies accepted

Session IDs

- PHP (and other server side languages) can use sessions to identify a sequence of user interactions
- Session can be used to:
  - track anonymous users (no user login needed) for analytics
  - Identify authenticated users so they can access protected pages until logout or session expired
  - Create variables that can be shared between web pages within the same session (e.g., “visit count” in the next slide)
- Create a session using *session_start();* at the beginning of your PHP script, before PHP has done any printing
  - After a session is created successfully, you have:
    - A unique session ID, which can be obtained by session_id(), stored on server
    - A super global array $_session[] storing session related data, e.g., $_session["visit_count"], $_session["last_visit_time"], etc.

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(); // create session ID and $_session global variable

   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).";  // "." is string concatenate

?>
```

Unregistering a Session Variable

```
<?php
    session_start();

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

session_unset(): Erase all session variables and data
session_destroy(): Destroy session data without destroying session variables

Passing Session ID: Cookies vs URL parameters

- There are two methods to propagate a session ID
  1. Cookies: Cookies are simple to use, but …
     - Cookies are not always available
     - Cookies are stored on browser and open to attack
  2. Session ID is passed as a URL parameter using GET method
- PHP session module supports both methods
  - Use cookies first, if unsuccessful, use GET parameter

session_start() and Session IDs

- A session ID must be at least 128 bits long (why so long?)
- What does session_start() do?
  - **If** session has been created (session ID in browser cookie or in URL parameter)
  - **then** load session data and continue
  - **else** generate session ID and set browser cookie, create $_SESSION

    PHP super global array
    - **If** session cookie does not exist (i.e., cookie disabled; use URL redirect)
    - **then** PHP constant SID set to session ID (PHPSESSID=cffce217b4…),
    - **else** SID set to empty (no need because cookie can do the job)

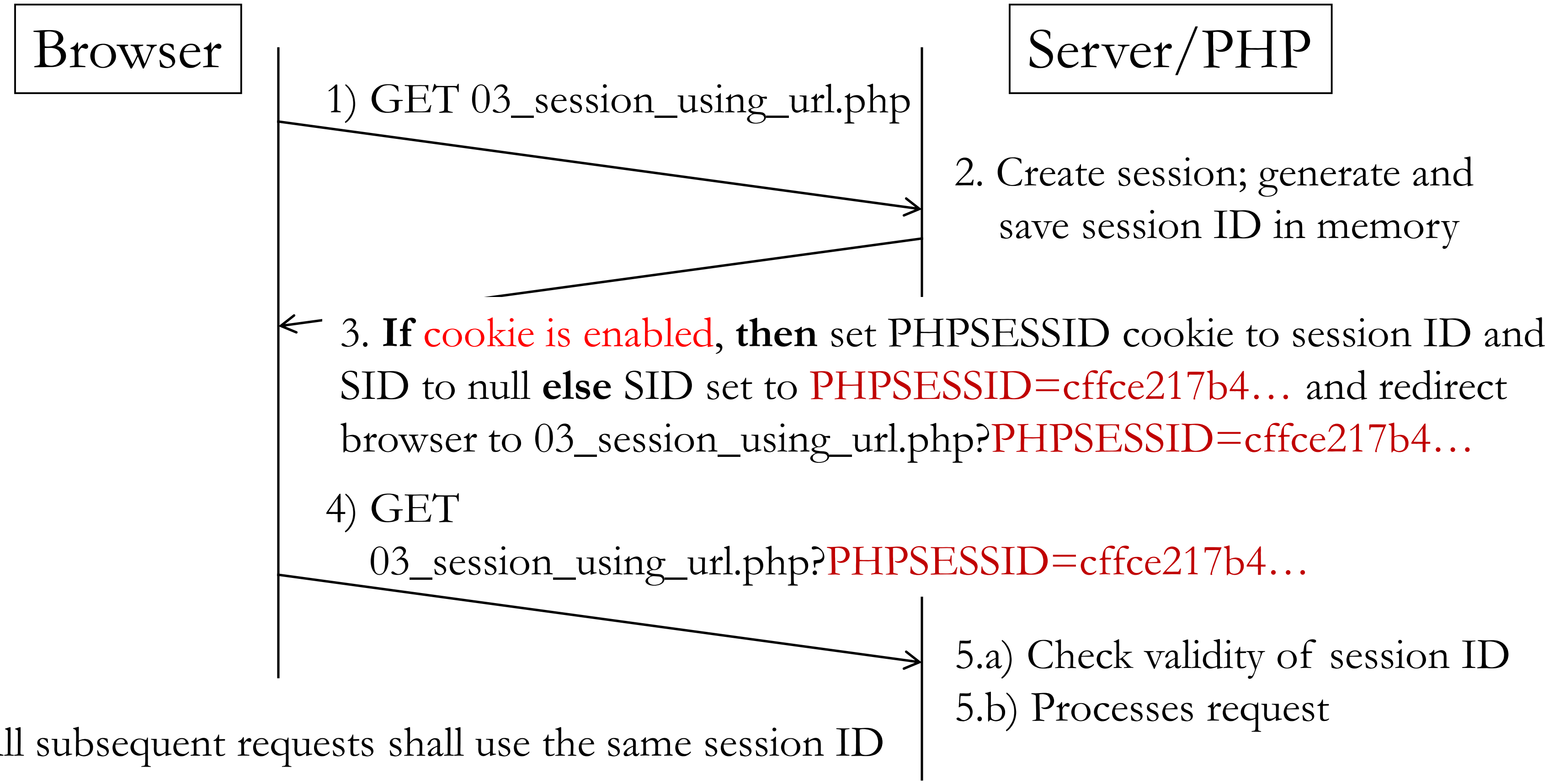
- The first GET is: GET 03_session_using_url.php
- Depending on whether cookies or redirect is used, subsequent GETs are either

  GET 03_session_using_url.php?

  or

  GET 03_session_using_url.php?PHPSESSID=cffce217b4…

A Possible Interaction Scenario



1) GET 03_session_using_url.php

2. Create session; generate and save session ID in memory

3. **If** cookie is enabled, **then** set PHPSESSID cookie to session ID and SID to null **else** SID set to PHPSESSID=cffce217b4… and redirect browser to 03_session_using_url.php?PHPSESSID=cffce217b4…

4) GET 03_session_using_url.php?PHPSESSID=cffce217b4…

5.a) Check validity of session ID
5.b) Processes request

All subsequent requests shall use the same session ID

Example Using URL

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

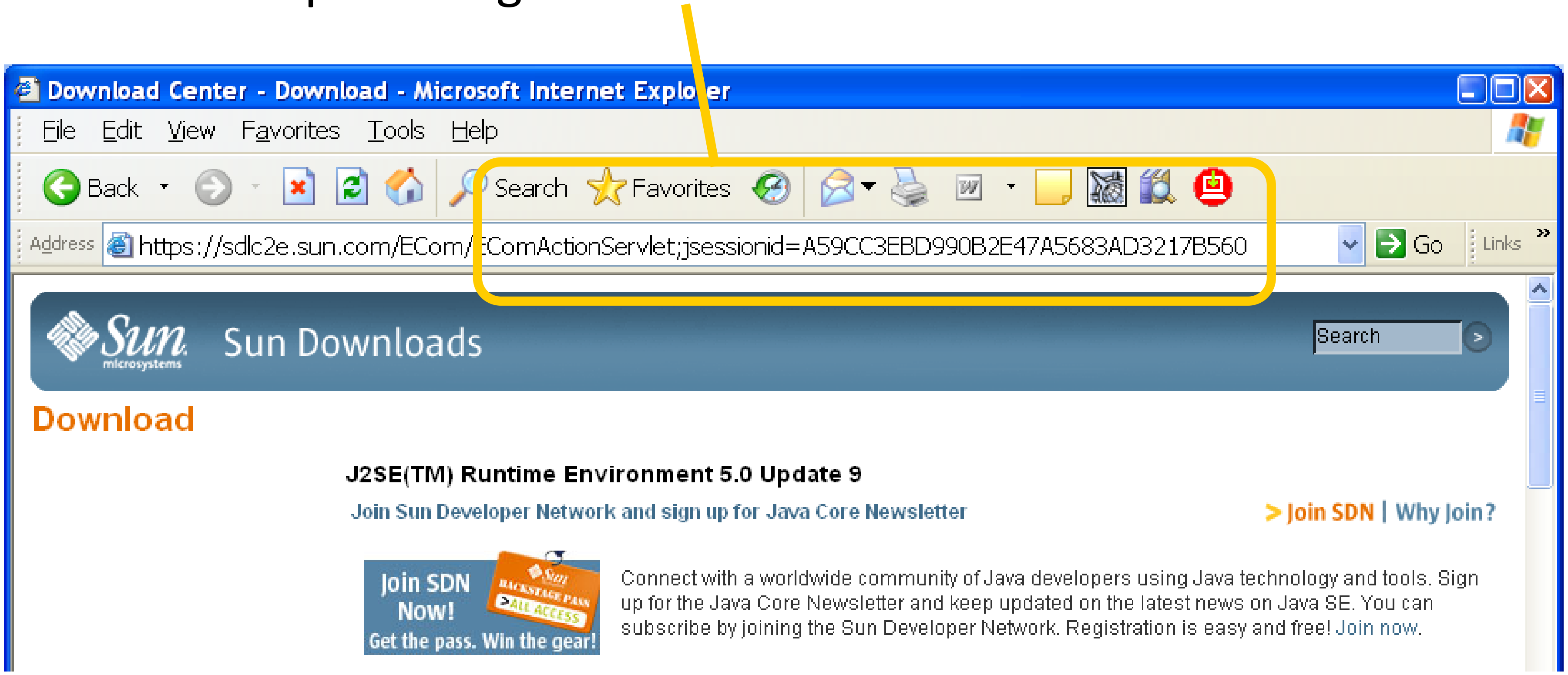
```
<?php
 // This will check for both cookie and URL methods
 session_start();  // Step 2

 if (!isset($_SESSION['count'])) {
   header("Location: 03_session_using_url.php?".SID);  // Step 3
   $_SESSION['count'] = 0;
 } else {
   echo "You have visited here ".$_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


Download Center - Download - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go Links
Sun microsystems Sun Downloads
Search

https://sdlc2e.sun.com/ECom/EComActionServlet;jsessionid=A59CC3EBD990B2E47A5683AD3217B560

Download

J2SE(TM) Runtime Environment 5.0 Update 9

Join Sun Developer Network and sign up for Java Core Newsletter

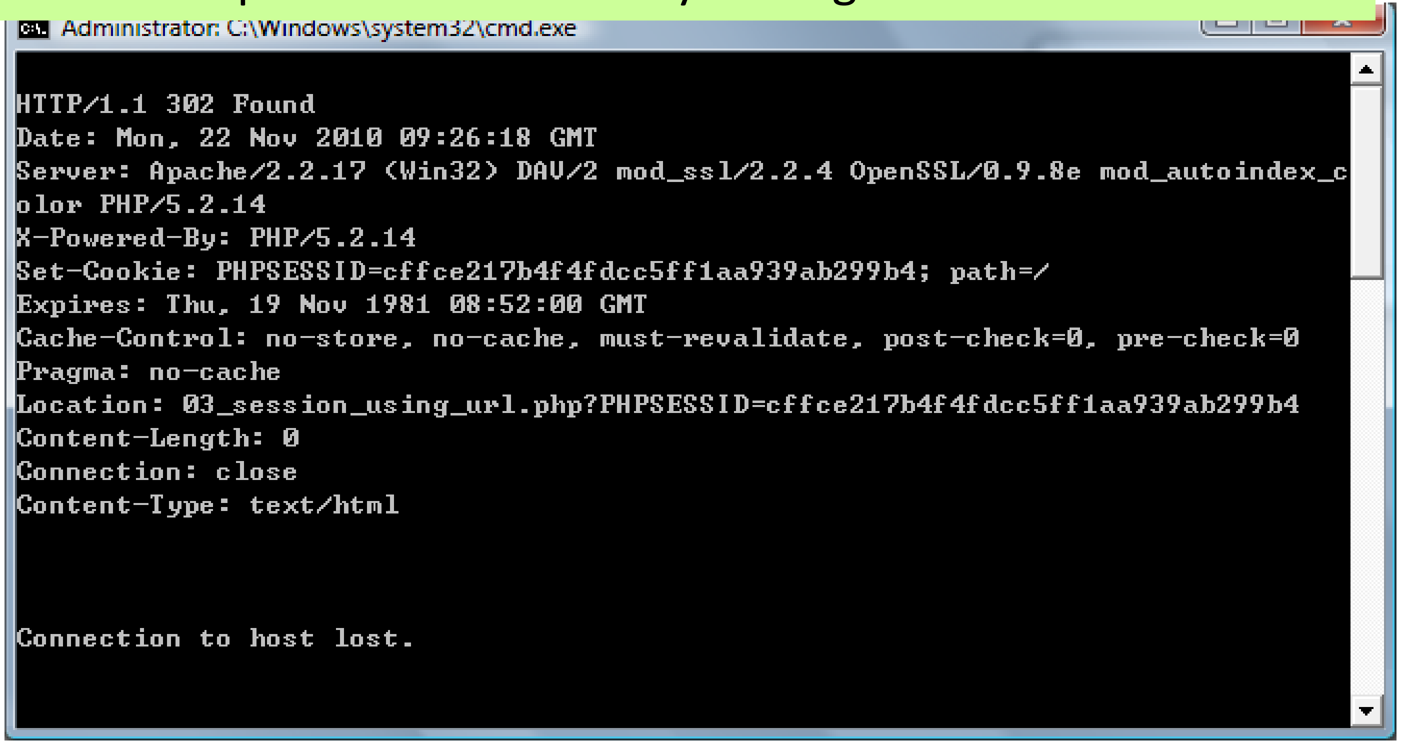
> Join SDN | Why Join?

Join SDN Now! Get the pass. Win the gear!

Connect with a worldwide community of Java developers using Java technology and tools. Sign up for the Java Core Newsletter and keep updated on the latest news on Java SE. You can subscribe by joining the Sun Developer Network. Registration is easy and free! Join now.

Example Output using PHP

Session ID is passed to browser by setting the PHPSESSID 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_c
olor 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

Take Home Message

- Session is important for security reason (time out a login)
- It is also important for web analytics, e.g., a session can be treated as a ‘visit’
- PHP provides some handy functions for handling sessions