Sesiones

Introducción

¿Por qué usar sesiones?

HTTP es un protocolo de comunicación para la transferencia de páginas web. Esto significa que su función es transportar datos (hipertexto) entre un Servidor Web (Host) y un Cliente Web (Browser).

Cuando un usuario navega dentro de un mismo sitio web puede provocar decenas de peticiones (requerimientos) al Servidor Web donde se encuentra alojado el sitio. Como mínimo cada cambio de página es una nueva petición. El Web Server no sabe (o no le importa) si dos requerimientos provienen del mismo usuario o no. Cada requerimiento es instanciado como una nueva petición de alguno de los tantos usuarios que pueden estar utilizando el sitio concurrentemente.

Las sesiones son utilizadas para crear un ambiente de variables en la memoria del Web Server por cada usuario (según la configuración del Hosting pueden ser almacenados en disco no Ram). El servidor puede almacenar diferentes estados y conservar esta información entre las distintas peticiones del usuario, o sea, el contenido de las variables persiste mientras que el usuario navega en diferentes páginas del sitio. (aún cuando ocurran largos intervalos de tiempo entre un requerimiento y otro).

Dicho en otras palabras, el protocolo de transferencia de hipertexto (HTTP) define las reglas utilizadas para transferir el texto, gráficos, vídeo y todos los otros datos a través de la World Wide Web. Se trata de un protocolo sin estado, lo que significa que cada solicitud se procesa sin ningún conocimiento de las peticiones anteriores o de las futuras. El manejo de sesión es una solución inteligente a este problema de indiferencia entre una petición y otra, ambas hechas por el mismo usuario. Esto se logra mediante la asignación de un atributo de identificación único a cada visitante (usuario) del sitio, conocido como el identificador de sesión (SID).

Identificador de Sesión - SID

Las sesiones son mantenidas a través de una identificación única (session_id) transferida en cada requerimiento. Este identificador se conserva en ambos extremos de la comunicación (Cliente y Servidor). Del lado del cliente, el navegador la almacena usualmente en una cookie, pero también puede ser transferida desde el Servidor por parámetros como un argumento de la dirección URL de la página.

Gestión de una sesión

PHP administra los diferentes SID de dos maneras:

- Cookies
- URL rewriting

Cookies

Uno de los medios para la gestión de la información del usuario se constituye en el método original de la utilización de una cookie. Cuando un usuario visita un sitio web, el servidor almacena información sobre el usuario en una cookie y la envía al navegador para que éste la guarde (se debe tener en cuenta que son los navegadores los que almacenan la información de una cookie).

A medida que el usuario realiza una solicitud a otra página (del mismo sitio), el servidor recupera la información de usuario y la utiliza, por ejemplo, para personalizar la página. Sin embargo, en lugar de almacenar las preferencias del usuario en la cookie, se almacena el SID. A medida que el cliente navega a través del sitio, el SID se recupera y los diversos temas correlacionados con el SID se suministran con fines de uso dentro de la página.

Es importante tener en cuenta que la posibilidad de utilizar cookies depende del cliente, así como que los programas estén preparados para la posibilidad de que el usuario tenga desactivado el soporte para cookies.

Por URL rewriting:

El segundo método utilizado para la propagación del SID consiste simplemente en añadir el SID en cada URL que se encuentra dentro de la página solicitada. Éste resulta una reproducción automática del SID. Este método, conocido como la reescritura de URL, elimina la posibilidad de que la función de gestión de sesiones pudiera ser negada si el cliente desactiva las cookies. Sin embargo, este método tiene sus inconvenientes. En primer lugar, la reescritura de URL no permite la persistencia entre las sesiones debido a que el proceso de añadir automáticamente un SID a la URL no continúa una vez que el usuario sale de su sitio. En segundo lugar, nada detiene que un usuario copie la URL y la envíe por correo a otro usuario. Siempre que la sesión no haya expirado puede continuar sobre otra estación de trabajo. Se debe considerar la posibilidad de caos que podría producirse si dos usuarios tienen que navegar simultáneamente usando la misma sesión. Por estas razones, se recomienda la metodología basada en cookies.

Características

- No requiere instalación. El soporte para sesiones está habilitado por defecto en PHP.
- Es la forma de mantener valores entre llamadas a distintas páginas.
- Se utiliza mayormente para identificar a un usuario logueado (autentificado) dentro de un sitio Web.
- Son una forma de mantener ciertos datos de un visitante entre los diversos accesos a una web. De esta forma, se pueden crear aplicaciones donde se le muestra al usuario datos personalizados de acuerdo a las preferencias o acciones del mismo.
- A cada usuario que entra al sitio se le asigna automáticamente un identificador de sesión.
- Este identificador llamado SID (Session ID) está asociado a una única sesión y ayuda a identificar el contenido de la sesión actual. Esta variable se propaga automáticamente a través de Cookies o mediante URL.
- Las sesiones tienen un tiempo de vida limitado.
- La sesión caduca según un tiempo establecido por el administrador del sitio Web o simplemente cuando el usuario cierra la ventana del browser.
- Para establecer manualmente la cantidad de minutos de vida de una sesión se debe modificar la opción session.cookie_lifetime dentro del archivo php.ini

Funciones de PHP para interactuar con Sesiones

Construcción de variables de sesión

- Para poder utilizarlas se debe (siempre) comenzar la sesión con la función session_start()
- Esta función deberá ser agregada en todas las páginas que manejen variables de sesión.
- Dicha función crea una nueva sesión o continúa con una existente si ha sido creada anteriormente.
- Las variables de sesión deben "registrarse".
- Para esto, se agregan a un arreglo denominado \$_SESSION[]
- Ejemplo de uso:

```
session_start();

$_SESSION['usuario'] = "root";
```

Identificación de la sesión

- Toda sesión tiene un número de identificación
- Para ver el identificador de sesión se utiliza la función session_id()
- Ejemplo de uso:

```
session_start();
echo "Session ID: " . session_id();
```

Destrucción de una sesión

- Las variables de sesión pueden ser destruidas en forma manual
- Generalmente se destruyen cuando un usuario hace un "log out" de un sitio web
- La función utilizada para destruir una sesión es session_destroy()
- Previamente se deberán eliminar los valores del arreglo que contiene los datos de la sesión
- Esta eliminación se realiza con la función session_unset()
- Ejemplo de uso

```
session_start();
session_unset();
session_destroy();
```

Crear variables de sesión

Las variables de sesión se crean utilizando la variable global \$_SESSION. Este es un arreglo dinámico que se redimensiona con cada variable que se desea agregar o quitar.

```
session_start();

$_SESSION["variable"] = "valor";

Se podría mejorar el código si se verifica la existencia de la variable de sesión. Si no existe se le asigna un valor.

session_start();
```

Autentificación de Usuarios a un Sitio web con uso de Sesiones

Logueo al Sitio

}

• Crear un archivo con el nombre: Logueo.php

if (!isset(\$_SESSION["country"])) {

\$_SESSION["variable"] = "valor";

- Armar un formulario de logueo en el cual se ingrese el usuario y la contraseña.
- El método del formulario debe ser POST ya que se guarda información confidencial.
- El atributo "action" debe apuntar a logueo.php o simplemente no especificarlo y por defecto apuntará a la misma pagina donde se encuentra el formulario.



Autentificación y creación de la sesión:

• Una vez cargado el usuario y contraseña al presionar el botón enviar, se ejecutará el evento submit y los datos del formulario serán enviados a la misma página (logueo.php)

- Luego, se deberá acceder a través de una consulta SQL y corroborar si existe en la base de datos un registro con el usuario y contraseña declarados en el formulario. Si existe un registro, entonces se le permitirá ingresar al sitio redireccionando a la página a Listado.php. En caso contrario, imprimir en la misma pantalla de logueo un mensaje de error informando que el usuario o la contraseña son incorrectas.
- Nota: para corroborar si la pagina logueo.php fue llamada por segunda vez producto del evento submit del formulario se debe utilizar la siguiente sentencia al haber presionado el botón enviar:

```
if ( $_SERVER[REQUEST_METHOD] = = "POST" )
{
...
}
```

Cerrar sesión: Pagina Desconectarse

- Desarrollar la pagina desconectarse.php
- La misma deberá destruir la sesión activa y redireccionar a la página logueo.php