

¿Qué es CGI?

El Common Gateway Interface, o CGI, es un conjunto de normas que definen cómo se intercambia información entre el servidor web y un script personalizado.

Las especificaciones de CGI están actualmente mantenidos por la NCSA y NCSA define CGI es el siguiente:

El Common Gateway Interface, o CGI, es un estándar para los programas actua como una puerta de enlace externos para interactuar con los servidores de información, tales como los servidores HTTP.

La versión actual es CGI/1.1 y CGI/1.2 está en progreso.

Navegacion Por Internet:

Para entender el concepto de CGI, vamos a ver lo que sucede cuando hacemos clic en un hipervínculo para navegar por una página web en particular o URL.

su buscador contacta al servidor web HTTP y la demanda de la ie URL. nombre de archivo.

Servidor Web analizará la URL y buscará el nombre del archivo. Si encuentra el archivo solicitado entonces el servidor web envía el archivo nuevamente en el navegador como respuesta sino no lo envía un mensaje de error que indica que ha solicitado un archivo incorrecto.

Navegador Web tiene respuesta del servidor web y se muestra ya sea el archivo recibido o mensaje de error en base a la respuesta recibida.

Sin embargo, es posible configurar el servidor HTTP de tal manera que cada vez que se solicita un archivo en un directorio determinado, ese archivo no es enviado de vuelta; sino que se ejecuta como un programa, y la salida producida por el programa es enviado de vuelta a su navegador para visualizar.

El Common Gateway Interface (CGI) es un protocolo estándar para permitir que las aplicaciones (llamados programas CGI o scripts CGI) para interactuar con servidores Web y con los clientes.

Estos programas CGI pueden ser un escrito en Python, Perl, Shell, C o C ++, etc.

Configuración del servidor Web

Antes de continuar con CGI de programación, asegúrese de que su servidor web es compatible con CGI y está configurado para gestionar los programas CGI. Todos los programas CGI a ejecutar por el servidor HTTP se guardan en un directorio configurado previamente. Este directorio se llama directorio CGI y por convención que se denomina como / var / www / cgi-bin. Por convención archivos CGI tendrán extensión como. Cgi, aunque son C ++ ejecutable.

Por defecto, el servidor Web Apache está configurado para ejecutar programas CGI en / var / www / cgi-bin. Si desea especificar cualquier otro directorio para ejecutar las secuencias de comandos CGI, es posible modificar la siguiente sección en el archivo httpd.conf:

```
<Directory "/var/www/cgi-bin">
    AllowOverride None
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>
```

```
<Directory "/var/www/cgi-bin">
Options All
</Directory>
```

Aquí, por supuesto que usted tiene hasta Servidor Web y funcionando con éxito y que son capaces de ejecutar cualquier otro programa CGI como Perl o Shell etc.

Primer Programa CGI

Considere el siguiente contenido de un programa en C + ±:

```
#include <iostream>
using namespace std;

int main ()
{
    cout << "Content-type:text/html\r\n\r\n";
    cout << "<html>\n";
    cout << "<head>\n";
    cout << "<title>Hello World - First CGI Program</title>\n";
    cout << "</head>\n";
    cout << "<body>\n";
    cout << "<h2>Hello World! This is my first CGI program</h2>\n";
    cout << "</body>\n";
    cout << "</html>\n";

    return 0;
}
```

Compilar el código anterior y el nombre del archivo ejecutable como cplusplus.cgi. Este archivo se guarda en el directorio / var / www / cgi-bin y se ha contenido siguiente. Antes de ejecutar el programa CGI asegurarse de que tiene el modo de cambio de archivo usando el comando chmod 755 cplusplus.cgi UNIX para que el archivo ejecutable. Ahora bien, si se hace clic cplusplus.cgi entonces esto genera el siguiente resultado:

Hello World! This is my first CGI program

el ejemplo de arriba C + + es un programa sencillo programa que está escribiendo su salida en archivo STDOUT es decir.salida por pantalla. Hay una característica importante y adicional disponible, que es la primera línea que se imprimirá Content-type: text / html \ r \ n \ r \ n. Esta línea se envía de vuelta al navegador y especificar el tipo de contenido que se mostrará en la pantalla del navegador. Ahora debe haber entendido el concepto básico de CGI y usted puede escribir muchos programas CGI complicados usando Python. Un programa CGI + C + puede interactuar con cualquier otro sistema externo, tales como RDBMS, para intercambiar información.

HTTP Header

La línea Content-type: text / html \ r \ n \ r \ n es parte de la cabecera HTTP, que se envía al navegador para entender el contenido. Todo el encabezado HTTP será el siguiente formulario
HTTP Field Name: Field Content

For Example
Content-type: text/html\r\n\r\n

Hay algunas otras cabeceras HTTP importantes, que va a utilizar con frecuencia en su programación CGI.

Header descripcion:

Content-type: Una cadena MIME define el formato del archivo que se devuelve. Ejemplo es Content-type: text / html

Expires: Date: La fecha de la información deja de ser válido. Esto debería ser usado por el navegador para decidir cuándo una página debe ser renovado. Una cadena de fecha válida debe estar en el formato de 01 de enero 1998 12:00:00 GMT.

Location: URL: La URL que debe ser devuelta en lugar de la URL solicitada. Puede utilizar esta presentada para redirigir una solicitud a cualquier archivo.

Last-modified: Date: La fecha de la última modificación del recurso.

Content-length: N Longitud, en bytes, de los datos que se devuelve. El navegador utiliza este valor para informar del tiempo de descarga estimado para un archivo.

Set-Cookie: String Establecer la cookie pasado por la cadena.

Variables de entorno CGI

Todo el programa CGI tendrá acceso a las siguientes variables de entorno. Estas variables juegan un papel importante al escribir cualquier programa CGI.

Nombre de la variable descripción:

CONTENT_TYPE: El tipo de datos del contenido. Se utiliza cuando el cliente está enviando contenidos conectados al servidor. Por ejemplo subida de archivo etc

CONTENT_LENGTH :La longitud de la información de consulta. Está disponible sólo para peticiones POST.

HTTP_COOKIE :Vuelva a poner las cookies creadas en forma del par clave y valor .

HTTP_USER_AGENT :El campo de encabezado de solicitud User-Agent contiene información acerca de la aplicación del usuario que origina la solicitud. Su nombre del navegador web.

PATH_INFO :la ruta para el script CGI.

QUERY_STRING :La información con codificación URL que se envía con la solicitud de método GET.

REMOTE_ADDR :La dirección IP del host remoto que realiza la solicitud. Esto puede ser útil para el registro o para fines de autenticación.

REMOTE_HOST:El nombre completo de la máquina que lo pide. Si esta información no está disponible, entonces REMOTE_ADDR se puede utilizar para obtener la dirección IR.

REQUEST_METHOD :El método utilizado para realizar la solicitud. Los métodos más comunes son GET y POST.

SCRIPT_FILENAME :La ruta completa al script CGI.

SCRIPT_NAME :el nombre del scrip cgi.

SERVER_NAME :el nombre del servidor hostname or IP Address

SERVER_SOFTWARE :El nombre y la versión del software que se ejecuta el servidor.

Aquí hay una pequeña programa CGI hacer una lista de todas las variables CGI. Haga clic en este enlace para ver el resultado Consigue el entorno.

C++ CGI Library

Para ver ejemplos reales, lo que tendría que hacer muchas operaciones por el programa CGI. Hay una biblioteca CGI escrito para C + + programa que se puede descargar desde <ftp://ftp.gnu.org/gnu/cgicc/> y siguiendo los siguientes pasos para instalar la biblioteca:

GET and POST Methods

Usted debe haber encontrado con muchas situaciones en las que usted necesita para pasar un poco de información desde el navegador al servidor web y en última instancia a su programa de CGI. Lo más frecuente navegador utiliza dos métodos para pasar esta información al servidor web. Estos métodos son el método GET y el método POST.

Pasar información utilizando el método GET:

El método GET envía la información de usuario codificada anexa a la solicitud de la página. La página y la información codificada se separan por el carácter de la siguiente manera:

<http://www.test.com/cgi-bin/cpp.cgi?key1=value1&key2=value2>

El método GET es el método por defecto para pasar información desde el navegador al servidor web y se produce una cadena larga que aparece en la ubicación de su navegador: caja. Nunca utilice el método GET si tienes contraseña u otra información sensible para pasar al servidor. El método GET tiene limitación de tamaño y se puede transmitir hasta 1024 caracteres en una cadena de solicitud.

Puede pasar información concatenando simplemente pares de clave y valor mucho con cualquier URL o puede usar etiquetas HTML <FORM> para pasar información utilizando el método GET.

Simple URL Example : Get Method

He aquí una simple URL que pasará dos valores para hello_get.py programa utilizando el método GET.

A continuación se muestra el programa para generar el programa CGI cpp_get.cgi para manejar la entrada dada por el navegador web. Vamos a usar C + + biblioteca CGI que hace que sea muy fácil acceder a la información que pasa:

Using Cookies in CGI

Protocolo HTTP es un protocolo sin estado. Pero para que un sitio web comercial que se requiere para mantener la información de la sesión entre las diferentes páginas. Por ejemplo, un registro de usuario termina después de completar muchas páginas. Pero, ¿cómo mantener la información de la sesión del usuario a través de todas las páginas web.

En muchas situaciones, el uso de las cookies es el método más eficaz de recordar y el seguimiento de las preferencias, compras, comisiones y demás información necesaria para una mejor experiencia de los visitantes o las estadísticas del sitio.

¿Cómo funciona? El servidor envía algunos datos al navegador del visitante en forma de una COOKIE . El navegador puede aceptar la cookie. Si lo hace, se almacena como un registro de texto sin formato en el disco duro del visitante. Ahora, cuando el visitante llega a otra página en su sitio, la cookie está disponible para su recuperación. Una vez recuperado, el servidor sabe / recuerda lo que estaba almacenado.

Las cookies son un registro de datos de texto sin formato de 5 campos de longitud variable:

Expires : La fecha de la cookie caducará. Si está en blanco, la cookie expira cuando el visitante cierra el navegador.

Domain :El nombre de dominio de su sitio.

Path:La ruta de acceso al directorio de la página web que se establece la cookie. Esto puede estar en blanco si desea recuperar la cookie desde cualquier directorio o página.

Secure:Si este campo contiene la palabra "seguro", entonces la cookie sólo puede ser recuperado con un servidor seguro. Si este campo está vacío, no existe esta restricción.

Name= Value :Las cookies se establecen y se recuperan en forma de pares de clave y valor.

Configuración de cookies

Esto es muy fácil de enviar las cookies en el navegador. Estas cookies se enviarán junto con la cabecera HTTP antes de Content-type presentada. Asumiendo que desea establecer Usuario y Contraseña como cookies. Así que ajuste las cookies se hará de la siguiente manera

A partir de este ejemplo, debe haber entendido cómo establecer cookies. Utilizamos encabezado HTTP Set-Cookie para establecer cookies.

Aquí, es opcional para establecer los atributos cookies como Expire, Dominio y Path. Es notable que las cookies se establecen antes de enviar la línea mágica "Content-type: text / html \r \n \r \n".

Compilar encima programa para producir setcookies.cgi, y tratar de crear cookies usando la siguiente enlace. Creará cuatro cookies en su computadora:

Recuperando cookies Esto es muy fácil de recuperar todas las cookies establecidas. Las cookies se almacenan en el entorno CGI HTTP_COOKIE variable y habrán siguiente formulario.

key1=value1;key2=value2;key3=value3...

Ejemplo para setear las cookies

```
#include <iostream>

using namespace std;

int main ()
{
    cout << "Set-Cookie:UserID=XYZ;\r\n";
    cout << "Set-Cookie:Password=XYZ123;\r\n";
    cout << "Set-Cookie:Domain=www.tutorialspoint.com;\r\n";
    cout << "Set-Cookie:Path=/perl;\n";
    cout << "Content-type:text/html\r\n\r\n";

    cout << "<html>\n";
    cout << "<head>\n";
    cout << "<title>Cookies in CGI</title>\n";
    cout << "</head>\n";
    cout << "<body>\n";

    cout << "Setting cookies" << endl;

    cout << "<br/>\n";
    cout << "</body>\n";
    cout << "</html>\n";

    return 0;
}
```

sigo buscando iNfo para recuperar jajaj

Cookies

En las dos secciones anteriores, hablamos de dos maneras en que la información puede ser pasado entre los programas (o ejecuciones del mismo programa) a través de un navegador. su sección se centra en el almacenamiento de la información de estado en el equipo cliente con las cookies. Las cookies son esencialmente pequeños archivos de texto que un servidor web envía a su navegador y que a continuación, guarda las cookies en su ordenador. Muchos sitios Web usan cookies para rastrear el progreso de los usuarios a través de su sitio (como en una aplicación de carro de la compra) o para ayudar a personalizar el sitio para un usuario individual. Las cookies no pueden entrar en el equipo, ni pueden borrar su disco duro. Sin embargo, pueden ser utilizados para identificar a los usuarios y llevar un registro de la frecuencia con el usuario visita un sitio o lo que los usuarios compran en un sitio. Por esta razón, las cookies son considerados como un problema de seguridad y privacidad. Navegadores web populares proporcionan soporte para cookies. Estos navegadores también permiten a los usuarios que se preocupan por su privacidad y seguridad para desactivar este soporte. La mayoría de los principales sitios Web usan cookies. Como programador, usted debe ser consciente de la posibilidad de que las cookies pueden ser desactivadas por sus clientes. Figuras 19.15-19.17 utilizan cookies para almacenar y manipular la información acerca de un usuario. Figura 19.15. Documento XHTML que contiene un formulario para enviar los datos al servidor

(Este elemento se muestra en las páginas 943 a 944 en la versión impresa)

La figura 19.15 es una página XHTML que contiene un formulario en el que los valores han de ser de entrada. el form pasa la información a través de post a writecookie.cgi (Fig. 19.16).

Este script CGI recupera los datos contenidos en la variable de entorno CONTENT_LENGTH. Línea 24 de la figura. 19.16 declara e inicializa cadena expira para almacenar la fecha de caducidad de la cookie, lo que determina el tiempo que la cookie reside en la máquina del cliente.

Este valor puede ser una cadena, como el que en este ejemplo, o puede ser un valor relativo. Por ejemplo, "30 d" establece la cookie caduca después de 30 días.

A los efectos de este capítulo de la fecha de vencimiento se establece deliberadamente a expirar en el año 2010 para asegurar que el programa se ejecutará correctamente en el futuro.

Puede establecer la fecha de caducidad de este ejemplo para cualquier fecha futura cuando sea necesario. El explorador elimina las cookies cuando caducan.

Figura 19.16. Escribir una cookie.

(Este elemento se muestra en las páginas 944 a 946 en la versión impresa)

Después de obtener los datos del formulario, el programa crea una cookie (líneas 54-56). En este ejemplo, se crea una cookie que almacena una línea de texto que contiene los pares de nombre y valor de los datos publicados, delimitados por un carácter de dos puntos (:). que la línea debe ser de salida antes de la cabecera se escribe en el cliente. La línea de texto comienza con el Set-Cookie: HEADER, lo que indica que el navegador debe almacenar los datos de entrada en una cookie. Hemos establecido tres atributos de la cookie: un par nombre-valor que contiene los datos que se almacenan, un par nombre-valor que contiene la fecha de vencimiento y un par nombre-valor que contiene la dirección URL del servidor de dominio para el que (por ejemplo, www.deitel.com) la cookie es válida. Para este ejemplo, el path no se establece en cualquier valor, por lo que la cookie puede leer desde cualquier servidor en el dominio del servidor que originalmente escribió la cookie.

Tenga en cuenta que estos pares de nombre y valor se separan con punto y coma (;). Utilizamos solamente caracteres de dos puntos dentro de nuestros datos de las cookies para no entrar en conflicto con el formato de la Set-Cookie: cabecera.(header)

Cuando entramos en los mismos datos que está representada en la fig. 19.15, líneas 54-56 almacenar los datos de "Name = Zoeage: 24color: Rojo" para la cookie. Líneas 59-76 envían una página Web

que indica que la cookie se ha escrito para el cliente.

Navegadores web almacenan la información de la cookie de manera específica del proveedor. Por ejemplo, Internet Explorer almacena las cookies de Microsoft como archivos de texto en el directorio de archivos temporales de Internet en la máquina del cliente. Netscape almacena sus galletas en un único archivo denominado cookies.txt

Figura 19.17 lee la cookie escrito en la figura. 19.16 y muestra la información almacenada. Cuando un cliente envía una solicitud a un servidor, el navegador Web cliente localiza las cookies previamente escritos por ese servidor.

Estas cookies son enviados por el navegador de nuevo al servidor como parte de la solicitud. En el servidor, en la variable de entorno HTTP_COOKIE variable que almacena las cookies del cliente.

La línea 20 llama a la función getenv con la variable de entorno HTTP_COOKIE como parámetro y almacena el valor devuelto en datastring.

Los pares nombre-valor se descodifican y se almacenan en cadenas (líneas 2334) de acuerdo con el esquema de codificación utilizado en la fig. 19.16. Líneas 3655 salida el contenido de la cookie como una página Web.

Figura 19.17. Programa para leer las cookies enviadas desde el ordenador del cliente.

(Este elemento se muestra en las páginas 947 a 948 en la versión impresa).

LAS COOCKIES presentan un riesgo de seguridad. Si los usuarios no autorizados tengan acceso a una computadora, pueden examinar los discos y ver los archivos locales, que incluyen cookies. Por esta razón, los datos confidenciales, como contraseñas, números de la seguridad social y números de tarjetas de crédito, nunca deben ser almacenados en las cookies.