

Web programming in C ++

Common Gateway Interface or CGI is a set of standards that define how information is exchanged between a Web Server and a Custom Script.

What is CGI?

Common Gateway Interface or CGI is a set of standards that define how information is exchanged between a Web Server and a Custom Script.

CGI Specification currently maintained by NCSA and NCSA defines CGI as follows: Common Gateway Interface or CGI is a standard for external gateway programs to Interface with information from the Server such as HTTP Server .

The current version is CGI / 1.1 and CGI / 1.2 is in development.

Web Browsing

To understand the concept of CGI, we see what happens when we click on a hyperlink to a specific Webpage or URL.

Your browser contacts the HTTP Web Server and requests a URL, for example, filename.

Web Server will parse that URL and will search for filename. If it finds the requested file, the Web Server sends the file back to the browser; otherwise, it sends an error message indicating that you requested a wrong file.

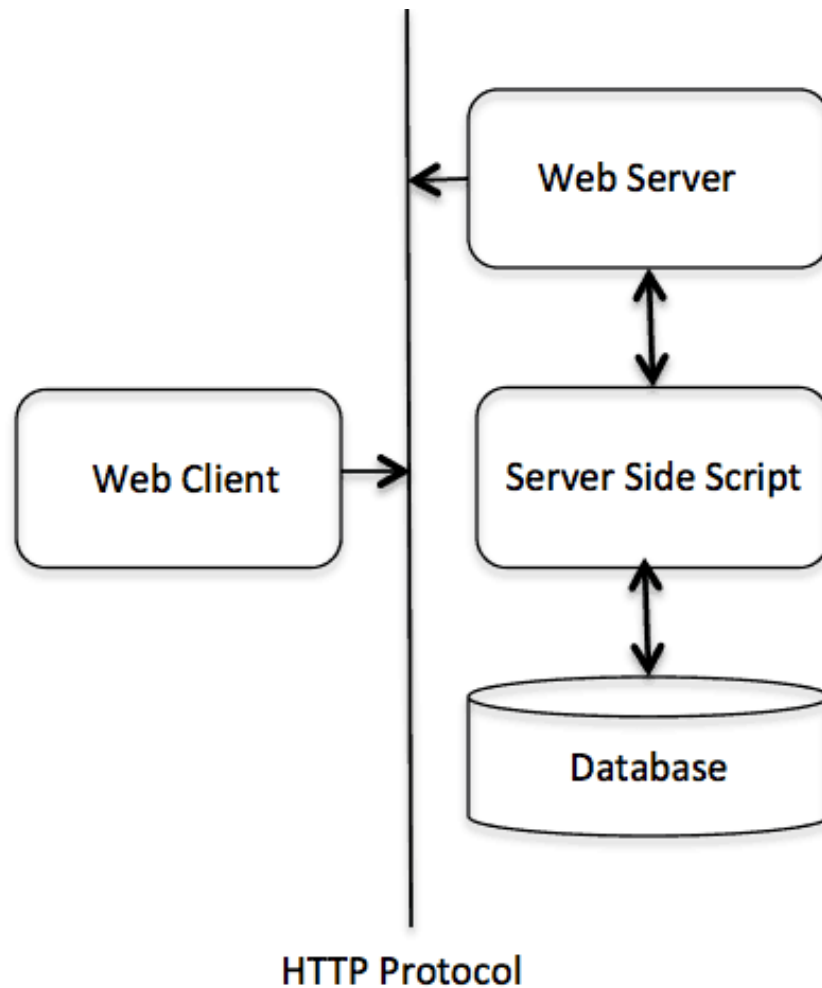
The Web browser receives feedback from the Web Server and displays the received file or error message based on the received response.

However, it is possible to set up HTTP Server in such a way that whenever a file in a particular directory is requested, the file is not sent back; instead it is executed as a program and the output is generated from the program sent to the browser to display.

Common Gateway Interface or CGI is a standard protocol for applications (called CGI programs or CGI scripts) capable of interacting with Web Server and with Client. These CGI programs can be written in Python, PERL, Shell, C or C ++ .

CGI structure diagram

Below is a diagram of a simple structure of CGI:



Configure Web Server

Before you continue with CGI programming, make sure your Web Server supports CGI and it is configured to handle CGI Program. All CGI programs to be executed by HTTP Server are kept in a pre-configured directory. This directory is called CGI directory, and by convention, it is named `/var/www/cgi-bin`. By convention, CGI files have a `.cgi` extension, so they are executable in C++.

By default, Apache Web Server is configured to run CGI Program in `/var/www/cgi-bin`. If you want to specify any other directory to run your CGI script, then you must modify the following area in the file `httpd.conf`:

```
span="">  "/var/www/cgi-bin" > AllowOverride None Options ExecCGI Order allow,
span="">  "/var/www/cgi-bin" > Options All
```

First CGI program

Consider the content of the following C++ program:

```
#include using namespace std ; int main () { cout << "Content-type:text/html
```

Hello World! Day is my first CGI program

```
n" ; cout << "n" ; cout << "n" ; return 0 ; }
```

Compile the above code and name it `cplusplus.cgi`. This file is kept in the `/var/www/cgi-bin` directory and it has the following content. Before running your CGI Program, make sure that you have switched the file mode by using the `chmod 755 cplusplus.cgi` command on UNIX to make the file executable. The result is:

Hello World! Day is my first CGI program

The C++ program is a simple program that is writing output on the `STDOUT` file, for example: the screen. There is an important feature available so the first line is printed as `Content-type: text/html`. This line sends back to the browser and determines the content type to be displayed on the browser screen. You now understand the basics of CGI and you can write many complex CGI Programs using C++. A CGI Program in C++ can interact with any other peripheral systems such as RDBMS, to exchange information.

HTTP Header

The **Content-type: text/html** line is part of the HTTP Header, which is sent to the browser. All HTTP Headers have the following form:

```
HTTP Field Name : Noi dung trong Vi du : Content - type : text / html
```

Here are some important HTTP Headers, which you will use regularly in CGI Programming.

Header	Description
Content-type:	A MIME string that defines the format of the file being returned. Example: <code>Content-type: text/html</code>
Expires:	Date The date the information expires. It should be used by the browser to decide when a page should refresh. A valid date string should be in the format <code>01 Jan 1998 12:00:00 GMT</code>
Location:	URL URL that should be returned instead of the requested URL. You can use this field to redirect a request to any file
Last-modified:	Date Last modified date of Content-length
source:	N Length, in bytes, of the data being returned. The browser uses this value to report the estimated download time for a file
Set-Cookie:	String Set the cookie passed through the <i>string</i>

Environment variable of CGI

All CGI Programs will have access to the following environment variables. These variables play an important role while writing any CGI Program:

Variable name	Description
CONTENT_TYPE	Data type of content. Used when the Client sends the attached content to the Server. For example: file upload
CONTENT_LENGTH	Length of query information. It is only available for HTTP_COOKIE POST requests. Returns the cookies set in the form that are the key / value pairs
HTTP_USER_AGENT	The request-header field is the User-Agent containing the information about the User Agent that started the request. It is the name of the web browser
PATH_INFO	Is the path for CGI script
QUERY_STRING	URL encoding information is sent with GET
REMOTE_ADDR	method The IP address of the remote host creates that request. It can be useful for log or authentication
REMOTE_HOST	purposes. The full name of the host creates that request. If this information is not available, REMOTE_ADDR can be used to retrieve the
REQUEST_METHOD	IP address. The method used to create the request. Common methods are GET and POST
SCRIPT_FILENAME	Path full to CGI script
SCRIPT_NAME	Name of CGI script
server_name	Hostname or IP address of Server
SERVER_SOFTWARE	Name and version of software that Server is running

The following is a small CGI program to list all CGI variables.

```
#include <string>
#include <string.h>
using namespace std; const string ENV [ 24 ] = { "COMSPEC"
; cout << "n" ; cout << "n" ; return 0 ; }
```

CGI library in C ++

With real-world examples, you will need to perform many operations by your CGI Program. There is a CGI library written for the C ++ program, which you can download from: <ftp://ftp.gnu.org/gnu/cgicc/> and follow these steps to install this library:

```
$tar xzf cgicc - X . X . X . tar . gz $cd cgicc - X . X . X / $ ./ configure
```

You can check the Documentation available at: [C ++ CGI Lib Documentation](#).

POST and GET methods

You have a number of situations when you need to transfer some information from the browser to the Web Server and finally to your CGI Program. The two methods used most often by the browser to transmit this information to Web Server are GET methods and POST methods.

Transfer information using the GET method

GET method of encrypted user information is appended to the request page. Is the page and encrypted information differentiated by characters? as follows:

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

The GET method is the default method for transmitting information from the browser to the Web Server and it creates a long string that appears in the browser's Location: box. Never use the GET method if you have a password or other sensitive information to transmit to the Server. The GET method has a size limit and you can transmit 1024 characters in a Request String.

When using the GET method, the information transmitted by the HTTP Header field is QUERY_STRING and will be accessible in CGI Program via the QUERY_STRING environment variable.

You can transmit information by concatenating a sequence of key / value pairs along with any URL or you can use HTML tags to transmit information using the GET method.

For example, the URL simply uses the GET method

This is a simple URL that will pass two values ??to the hello_get.py program using the method.

Below is the program to create the CGI Program cpp_get.cgi to handle the input provided by the web browser. We are using the CGI library in C ++, to make it easier to access the transmitted information:

```
#include <string>
#include <string.h>
#include <string>
#include <string.h>
#include <string>
#include <string.h>
#include <string>
#include <string.h>
n" ; fi = formData . getElement ( "last_name" ); if ( ! fi -> isEmpty () && fi
n" ; cout << "n" ; cout << "n" ; return 0 ; }
```

Now, compile the above program as follows:

```
$g ++ - o cpp_get . cgi cpp_get . cpp - lcgicc
```

Create `cpp_get.cgi` and place it in your CGI directory.

Example of FORM using the GET method

The following simple example will pass two values ??using HTML FORM and submit button. We are using the same CGI Program as `cpp_get.cgi` to handle this input.

```
span=""> action = "/cgi-bin/cpp_get.cgi" method = "get" > First Name:   
span=""> type = "text" name = "first_name" >  
span=""> /> Last Name:  span=""> type = "text" name = "last_name" />  

```

Transfer information using POST method

A more reliable general method for passing information to a CGI Program is POST. It encapsulates information in the same way as GET, but instead of sending it in the form of a text string after a sign? in the URL, it is sent in the form of a separate message. Notice this to CGI Script in the standard input form.

Use CGI Program `cpp_get.cgi` to handle POST method. We use the same example above, which will pass two values ??using the HTML FORM and Submit button, but this time with the POST method as follows:

```
span=""> action = "/cgi-bin/cpp_get.cgi" method = "post" > First Name:   
span=""> type = "text" name = "first_name" >  
span=""> /> Last Name:  span=""> type = "text" name = "last_name" />  

```

Pass Checkbox Data to CGI Program

Checkbox is used when more than one option is needed to be selected:

The following HTML code is an example for a form with two checkboxes:

```
span=""> action = "/cgi-bin/cpp_checkbox.cgi" method = "POST" target = "_blank" >  
 span=""> type = "checkbox" name = "maths" value = "on" /> Maths  
 span=""> type = "checkbox" name = "physics" value = "on" /> Physics  

```

The following C ++ program will create `cpp_checkbox.cgi` to handle the input provided by the browser via the checkbox:

```
#include #include #include #include #include #include #include #include #include
n" ; physics_flag = formData . queryCheckbox ( "physics" ); if ( physics_flag
n" ; cout < "n" ; cout < "n" ; return 0 ; }
```

Transfer Radiobutton Data to CGI Program

Radio Button is used when only one option is needed to choose.

The following HTML code is an example of a form with two radio buttons:

```
span=""> action = "/cgi-bin/cpp_radiobutton.cgi" method = "post" target = "_blank" >
 span=""> type = "radio" name = "subject" value = "maths" checked = "checked"
/> Maths  span=""> type = "radio" name = "subject" value = "physics" /> Physics


```

The following C ++ program will create **cpp_radiobutton.cgi** to handle the input provided by the web browser through the radio buttons:

```
#include #include #include #include #include #include #include #include #include
n" ; cout < "n" ; cout < "n" ; return 0 ; }
```

Transmit Text Area Data to CGI Program

The TEXTAREA element is used when multi-line text must be passed to the CGI Program.

The following HTML code is an example for a form with a TEXTAREA box:

```
span=""> action = "/cgi-bin/cpp_textarea.cgi" method = "post" target = "_blank" >
< span=""> name = "textcontent" cols = "40" rows = "4" > Type your text
here. <input type="submit" value = "Submit" /> </pre>
<p style="text-align: justify;">The following C ++ program will create
```

```
<pre class="prettyprint notranslate prettyprinted"> <form< span="">  
action = "/cgi-bin/cpp_dropdown.cgi" method = "post" target = "_blank" >  
<select< span=""> name = "dropdown" > <option< span=""> value = "Maths"
```

```
Cookie:Domain=www.tutorialspoint.com;rn" ; cout << "Set-  
Cookie:Path=/perl;n" ; cout << "Content-type:text/htmlrnrn" ; cout <<  
"n" ; cout << "n" ; cout << "Cookies in CGI" ; cout << "n" ;
```

```
you can try the code above with your server.</p>
<p style="text-align: justify;">This is the program
<strong>cpp_uploadfile.cpp</strong> to process file upload:</p>
```

You finished reading the article "**Web programming in C ++**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.
