

REVIEW ARTICLE



ISSN: 2321-7758

COMMON GATEWAY INTERFACE (CGI)

DHRUV PAHUJA, ABHISHEK MAHAJAN, AKASH VERMA

CSE Department, Dronacharya College of Engg.

Article Received: 21/10/2014

Article Revised on: 06/11/2014

Article Accepted on:08/11/2014



ENGINEERS
MAKE A WORLD OF DIFFERENCE

International Journal of
Engineering
Research-Online



ABSTRACT

Common gateway interface (CGI) is the first technology that has been created to enable interactivity on the World Wide Web. Despite its relative age and a number of competing technologies, CGI remains a viable tool for creating dynamic Web pages. This chapter introduces CGI technology from a historic perspective. Further, this chapter examines the architecture of a CGI script and illustrates it with several examples written in Perl. The chapter concludes with an overview of different applications of CGI scripts and future prospects of this technology. This perl library uses perl5 objects to make it easy to create Web fill-out forms and parse their contents. This package defines CGI objects, entities that contain the values of the current query string and other state variables. Using a CGI object's methods, you can examine keywords and parameters passed to your script, and create forms whose initial values are taken from the current query.

Keywords-active server page, application programming interface (API), CGI script, dynamic web pages, Perl, CGI forms, CGI Environment, Configuring the Server

©KY Publications

INTRODUCTION

The common gateway interface (CGI) is a standard way for a Web server to pass a Web user's request to an application program and to receive data back to forward to the user. When the user requests a Web page (for example, by clicking on a highlighted word or entering a Web site address), the server sends back the requested page. However, when a user fills out a form on a Web page and sends it in, it usually needs to be processed by an application program. The Web server typically passes the form information to a small application program that processes the data and may send back a confirmation message. This method or convention for passing data back and forth between the server and the application is called the common gateway interface (CGI). It is part of the Web's Hypertext Transfer Protocol (HTTP).

If you are creating a Web site and want a CGI application to get control, you specify the name of the application in the uniform resource locator (URL) that you code in an HTML file. This URL can be specified as part of the FORMS tags if you are creating a form. For example, you might code:

<FORM METHOD=POST ACTION=http://www.mybiz.com/cgi-bin/formprog.pl>

and the server at "mybiz.com" would pass control to the CGI application called "formprog.pl" to record the entered data and return a confirmation message. (The ".pl" indicates a program written in PERL but other languages could have been used.)

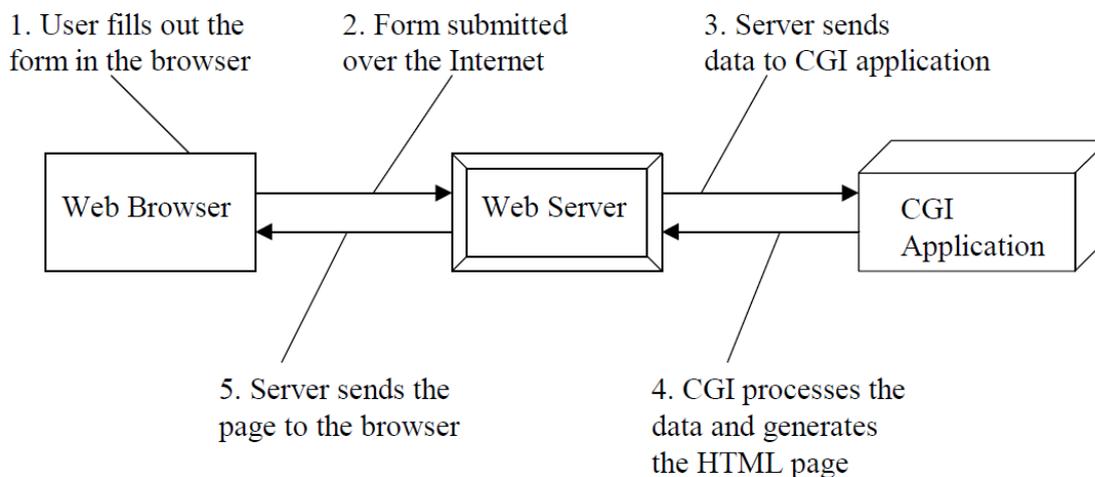
The common gateway interface provides a consistent way for data to be passed from the user's request to the application program and back to the user. This means that the person who writes the application program can make sure it gets used no matter which operating system the server uses (PC, Macintosh, UNIX, OS/390, or others). It's simply a basic way for information to be passed from the Web server about your request to the application program and back again.

Because the interface is consistent, a programmer can write a CGI application in a number of different languages. The most popular languages for CGI applications are: C, C++, Java, and PERL.[3]

CGI FORMS

A CGI form is a way for an Internet user to interact dynamically with a Web server. An HTML page that contains a form may use a CGI program to process the form's data once it has been submitted. For example: a user visits a Web page looking for specific information on a topic and finds that the Web page offers an e-mail newsletter service. In order for the user to receive the newsletter, he must fill out a CGI form with all of his applicable information. The form is then sent to the server using CGI, and the server processes the information that was sent in the form in order to add the user to its database and initiate the user's newsletter service.[4]

CGI REQUEST PROCESSING



Using CGI, an intranet programmer can write an application that searches a database and displays the result in HTML format. CGI is used to allow people to fill out corporate forms on an intranet, and have that information entered into a database.

1. For CGI to be used, an intranet programmer first writes a program or a script. Since interpreted languages are easier to debug, modify, and maintain than compiled programs, they are used more frequently than compiled programs. Perl is probably the most popular language for writing scripts. Languages such as C, C++, or FORTRAN can be used to access CGI as well after they have been compiled.
2. Next, the script or compiled program is put on an intranet Web server in a special directory (usually called /cgi-bin) which holds all the CGI programs and scripts. For security reasons, programs put in other directories won't run. If multiple directory placements were allowed, it would be difficult for

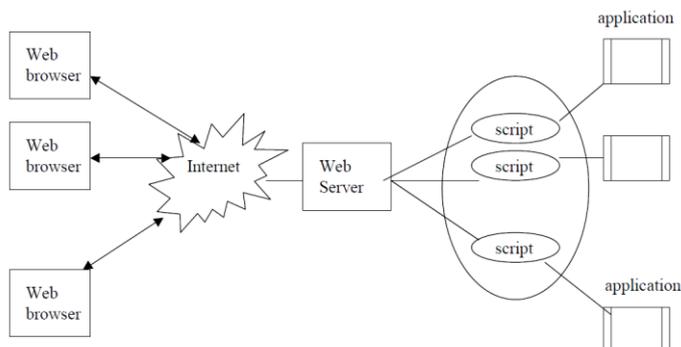
intranet maintainers to track them all and therefore realize if an unauthorized user posted a rogue CGI program.

3. After posting the CGI program to /cgi-bin, it is linked to the URL in HTML on an intranet Web page.
4. When someone clicks on the URL, the server launches (via HTML's GET or POST) the CGI program residing on the Web server. If, for example, the CGI program's function is for searching a database, the CGI program could send an HTML form to the client. From the client, the data on the completed query form is sent back to the CGI program using STDIN or environment variables. The data is formatted as encoded name/value pairs.
5. The CGI program contacts the database and requests the information. The database sends the information to the CGI program. The information can be in a variety of formats, such as text, graphics, sound and video files, and URLs. The CGI program returns the results to the server (via STDOUT) which, in turn, would send it on to the browser.
6. The CGI program also formats the data, for example, taking the information and putting it into HTML format so that the user can read it using a Web browser. The user can use that HTML page as they can any other.

CGI ENVIRONMENT

1. CGI scripts generally executed with limited permission
2. CGI Scripts are given predefined environment variables that provide information about web server and client
 - For Perl, it's available through %ENV hash
3. Example of Environment variables
 - SERVER_NAME: The servers hostname/ip
 - SERVER_PROTOCOL: The name and version of the protocol.
 - SERVER_PORT: The port number to which request was sent
 - REQUEST_METHOD: The method with witch the request was made
 - PATH_INFO:
 - SCRIPT_NAME
 - QUERY_STRING
 - REMOTE_HOST
 - REMOTE_ADDR
 - AUTH_TYPE: if the server supports user authentication
 - REMOTE_USER: If the server supports users authentication, and the script is protected, this is the username they have authenticated as.
 - CONTENT_TYPE
 - CONTENT_LENGTH

CGI Based Architecture



ADVANTAGES

1. Platform independence: Most web servers support CGI, including UNIX: Apache, Netscape, NCSA, and CERN
Windows NT: Netscape, Microsoft IIS, and O'Reilly Website
Macintosh: WebStar
2. Language independence: (Perl, TCL, C, C++ Visual Basic, AppletScript, Java)
3. Scalability: The simplicity of the CGI interface means that it is extremely scalable.

DISADVANTAGES

1. Performance
Every time a user requests a CGI script, the server must launch the CGI program, which takes processor time. When the CGI program is written in an interpreted language like Perl, then the program must run the entire Perl interpreter, and compile the program before it can be run, which takes even more processor time. For busy web sites running complex applications, the performance problem becomes critical issue.
2. It is free. Not easy to sell the web server products with free software.
3. Not good to interact with database.(connecting issue)
4. Not flexible or powerful to generate dynamic HTML pages.

CONFIGURING THE SERVER

Before you can run CGI programs on your server, certain parameters in the server configuration files must be modified. If you are using either the NCSA or CERN HTTP server, you need to first set the *ServerRoot* directive in the *httpd.conf* file to point to the directory where the server software is located:

```
ServerRoot /usr/local/etc/httpd
```

Running CGI Scripts

On the NCSA server, the *ScriptAlias* directive in the server resource map file (*srm.conf*) indicates the directory where the CGI scripts are placed.

```
ScriptAlias /cgi-bin/ /usr/local/etc/httpd/cgi-bin/
```

For example, if a user accesses the URL:

```
http://your_host.com/cgi-bin/welcome
```

the local program:

```
/usr/local/etc/httpd/cgi-bin/welcome
```

will be executed by the server. You can have multiple directories to hold CGI scripts:

```
ScriptAlias /cgi-bin/ /usr/local/etc/httpd/cgi-bin/
```

```
ScriptAlias /my-cgi-bin/ /usr/local/etc/httpd/my-cgi-bin/
```

You might wonder why all CGI programs must be placed in distinct directories. The most important reason for this is system security. By having all the programs in one place, a server administrator can control and monitor all the programs being run on the system. However, there are directives that allow programs to be run outside of these directories, based on the file extension. The following directives, when placed in the *srm.conf* configuration file, allow the server to execute files containing *.pl*, *.sh*, or *.cgi* extensions.

```
AddType application/x-httpd-cgi .pl .sh .cgi
```

However, this could be very dangerous! By globally enabling all files ending in certain extensions, there is a risk that novice programmers might write programs that violate system security (e.g., printing the contents of important system files to standard output).[6]

On the CERN server, setting up the CGI directory is done in the *httpd.conf* file, using the following syntax:

```
Exec /cgi-bin/* /usr/local/etc/httpd/cgi-bin
```

CONCLUSION

The common gateway interface has literally revolutionized the way we use the Internet. CGI Scripts were the first technology to enable interactivity on the web. Without CGI Scripts and all other technologies spawned by it, we would still be using static web pages instead of interactive web applications. CGI scripts

provide the mechanism for the web server respond to the client requests by generating Web Pages with customized and personalized content. The common gateway interface is a platform independent and transparent to the client browsers. CGI scripts can be written in many programming languages capable of producing command-line executable that can generate HTML output.

REFERENCES

- [1] The Internet Encyclopaedia, Volume 2 edited by Hossein Bidgoli
- [2] http://en.wikipedia.org/wiki/Common_Gateway_Interface
- [3] <http://searchsoa.techtarget.com/definition/common-gateway-interface>
- [4] <http://www.webopedia.com/TERM/C/CGI.html>
- [5] <http://webdesign.about.com/od/cgi/a/aa021599.htm>
- [6] Shishir Gundavaram, CGI Programming on the World Wide Web, O'Reilly Open Books Project, 1st Edition March 1996
- [7] <http://www.abelvisp.co.uk/enduser-features.html>
- [8] <http://www.bytepile.com/definitions-c.php>