

CGI

June 18, 2024

1 What do you mean by Common Gateway Interface (CGI) ?

The Common Gateway Interface (CGI) is a standard protocol used to interface external application software with an information server, such as an HTTP server. It's a 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 a user requests a web page that uses CGI, the following process occurs:

1. The web server receives the request and determines that it's for a CGI program.
2. The web server runs the CGI program, passing input from the user to the program and output from the program back to the user.
3. The CGI program receives the user's request from the web server, processes it, and returns the result back to the web server.
4. The web server sends the result back to the user's web browser.

CGI can be written in any language that can read STDIN (standard input), write to STDOUT (standard output), and read environment variables, such as Perl, C, Python, or shell script languages.

However, CGI has some performance drawbacks. Every time a CGI script is executed, a new process is started on the server. For busy websites, this can significantly affect the server's performance. For this reason, CGI has largely been replaced by more efficient technologies like PHP, ASP.NET, and JSP for dynamic web content.

2 How does CGI work ?

The Common Gateway Interface (CGI) is a standard that defines how web servers can interact with scripts or programs to generate dynamic web pages. Here's a step-by-step explanation of how CGI works:

1. **User Request:** The process begins when a user clicks on a link or submits a form that points to a URL corresponding to a CGI script on a web server.
2. **Server Processing:** The web server recognizes that this URL is not a regular file, but a request for a CGI script. The server then prepares to execute the script. It does this by setting up a number of environment variables that contain information about the request, such as the client's IP address, the requested URL, any submitted form data, and more.
3. **Script Execution:** The server executes the CGI script as a separate process. This script can be written in any programming language that can read from standard input (STDIN),

write to standard output (STDOUT), and access environment variables. Common languages for CGI scripts include Perl, Python, and C.

4. **Data Collection:** The CGI script reads the environment variables and STDIN to get the information about the request. If the request was a form submission, the submitted data will be available to the script.
5. **Response Generation:** The script does whatever processing it needs to do based on the request, such as querying a database, reading or writing files, or performing calculations. It then generates an HTTP response. This response includes HTTP headers followed by the body of the response, which is typically HTML.
6. **Server Sends Response:** The web server takes the output from the CGI script and sends it back over the internet to the user's browser.
7. **Browser Displays Result:** The user's browser receives the response from the server and displays the result. If the response body was HTML, the browser will render it as a web page.

This process happens for each and every CGI request. Because a new process is created for each request, CGI can be slower and more resource-intensive than other methods of generating dynamic content, such as using a language with built-in web server support like PHP or a web application framework like Ruby on Rails or Django.

3 Example?

Here's a simple example of a CGI script written in Python. This script simply returns an HTML page that displays "Hello, World!".

```
[ ]: #!/usr/bin/env python3

print("Content-Type: text/html")    # HTML is following
print()                            # blank line, end of headers

print("<html><body><h1>Hello, World!</h1></body></html>")
```

Here's how this script works:

1. The first line is a shebang (`#!`) that tells the system to execute this script using Python 3.
2. The `print("Content-Type: text/html")` line is sending an HTTP header to the browser, telling it that the content is going to be HTML.
3. The `print()` line sends a blank line, which signifies the end of the HTTP headers.
4. The last line sends the body of the HTTP response, which is a simple HTML page that displays "Hello, World!".

To use this CGI script:

1. Save it with a `.cgi` extension, for example `hello.cgi`.
2. Make sure the script has execute permissions. You can set this with the command `chmod +x hello.cgi`.

3. Place the script in your web server's CGI directory. The location of this directory depends on your web server's configuration.
4. Access the script through your web browser by navigating to `http://yourserver.com/cgi-bin/hello.cgi` (replace `yourserver.com` with your server's domain name or IP address).

Remember, your web server must be configured to handle CGI scripts, and the directory where you place your CGI script must be configured as a CGI directory. The specifics of how to do this depend on your web server.

4 Thank You!