

Web Server Design

Lecture 9 – Server-Side Execution

Old Dominion University

Department of Computer Science

CS 431/531 Fall 2019

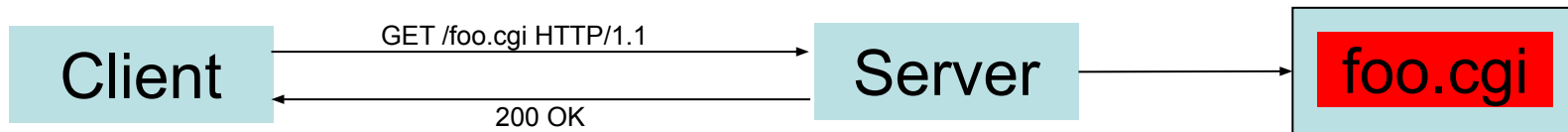
Sawood Alam <salam@cs.odu.edu>

2019-10-24

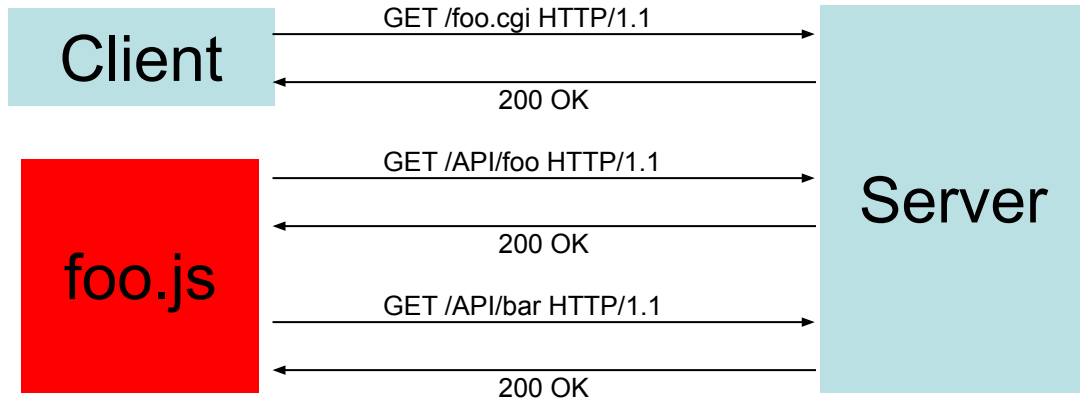
Original slides by Michael L. Nelson

Common Gateway Interface (CGI)

- A method for remotely invoking executable programs on a server
 - A long-time convention
 - <http://hoohoo.ncsa.uiuc.edu/cgi/>
 - Finally defined in RFC 3875



Cf. Client-Side Approach



CGI Invocation

- How Apache does it:
 - http://httpd.apache.org/docs/current/mod/mod_cgi.html
- We'll live slightly more dangerously:
 - Any executable (non-directory) file can be invoked as CGI with:
 - POST
 - GET w/ query string
 - e.g. `/a/b/c.cgi?var1=foo&var2=bar`

CGI Operation

- The CGI program is responsible for returning (on STDOUT) some combination of its own headers:
 - Content-type
 - Location
 - Status
 - and other locally-defined headers
- Script-returned headers are:
 - Collected by the server
 - Processed; e.g.:
 - “Location” -> HTTP/1.1 302 Found
 - Status -> HTTP response code line
 - Combined with the server’s headers
- Resulting combination of headers are returned to the client

Partial vs. Non-Parsed Headers

- The approach in the prior slide is what is known as "partial headers" (a combination of the headers from CGI script + the server)
- You can also have your script be responsible for all of the headers, in non-parsed-headers (nph) mode.
 - Somewhat outdated, but possible
 - See: http://docstore.mik.ua/orelly/linux/cgi/ch03_03.htm
 - We will not do nph scripts for A4

Status + Custom Header

```
$ cat status.cgi
#!/usr/bin/perl

print "Status: 678 This is not a real HTTP status code\n";
print "X-This-Header-Is-Madeup: foo=bar\n\n";
```

```
$ telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
GET /~mln/teaching/cs595-s12/cgi/status.cgi HTTP/1.1
Host: www.cs.odu.edu

HTTP/1.1 678 This is not a real HTTP status code
Date: Tue, 03 Apr 2012 18:01:58 GMT
Server: Apache/2.2.17 (Unix) PHP/5.3.5 mod_ssl/2.2.17 OpenSSL/0.9.8q
X-This-Header-Is-Madeup: foo=bar
Content-Length: 0
Content-Type: text/plain

Connection closed by foreign host.
```

Status With an Entity

```
$ cat status-entity.cgi
#!/usr/bin/perl

print "Status: 678 This is not a real HTTP status code\n";
print "X-This-Header-Is-Madeup: foo=bar\n";
print "Content-type: text/html\n\n";
print "this is not a header, this is part of the entity...\n"
```

```
$ curl -i
http://www.cs.odu.edu/~mln/teaching/cs595-s12/cgi/status-entity.cgi
HTTP/1.1 678 This is not a real HTTP status code
Date: Tue, 03 Apr 2012 18:11:57 GMT
Server: Apache/2.2.17 (Unix) PHP/5.3.5 mod_ssl/2.2.17 OpenSSL/0.9.8q
X-This-Header-Is-Madeup: foo=bar
Content-Length: 52
Content-Type: text/html

this is not a header, this is part of the entity...
```


Location

```
$ cat location.cgi
#!/usr/bin/perl

print "Location: http://www.cs.odu.edu/~mln/\n\n";
```

```
$ telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
GET /~mln/teaching/cs595-s06/cgi/location.cgi HTTP/1.1
Host: www.cs.odu.edu
```

```
HTTP/1.1 302 Found
Date: Mon, 24 Apr 2006 14:40:31 GMT
Server: Apache/2
Location: http://www.cs.odu.edu/~mln/
Content-Length: 277
Content-Type: text/html; charset=iso-8859-1
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>302 Found</title>
</head><body>
<h1>Found</h1>
<p>The document has moved <a href="http://www.cs.odu.edu/~mln/">here</a>.</p>
<hr>
<address>Apache/2 Server at www.cs.odu.edu Port 80</address>
</body></html>
```

Note how the entity is automatically constructed

Location Overrides the Entity...

```
$ cat location-entity.cgi
#!/usr/bin/perl

print "Location: http://www.cs.odu.edu/~mln/\n";
print "Content-type: text/plain\n\n";
print "this will never get printed..."
```

```
$ curl -i http://www.cs.odu.edu/~mln/teaching/cs595-s12/cgi/location-entity.cgi
HTTP/1.1 302 Found
Date: Tue, 03 Apr 2012 18:27:33 GMT
Server: Apache/2.2.17 (Unix) PHP/5.3.5 mod_ssl/2.2.17 OpenSSL/0.9.8q
Location: http://www.cs.odu.edu/~mln/
Content-Length: 329
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>302 Found</title>
</head><body>
<h1>Found</h1>
<p>The document has moved <a href="http://www.cs.odu.edu/~mln/">here</a>.</p>
<hr>
<address>Apache/2.2.17 (Unix) PHP/5.3.5 mod_ssl/2.2.17 OpenSSL/0.9.8q Server at
www.cs.odu.edu Port 80</address>
</body></html>
```

Content-type

```
$ cat ls.cgi
#!/usr/bin/perl

print "Content-type: text/plain\n\n";
$ls = `ls -alR`;
print "$ls\n";
```

```
$ telnet www.cs.odu.edu 80
Trying 128.82.4.2...
Connected to xenon.cs.odu.edu.
Escape character is '^]'.
HEAD /~mln/teaching/cs595-s07/cgi/ls.cgi HTTP/1.1
Connection: close
Host: www.cs.odu.edu
```

```
HTTP/1.1 200 OK
Date: Mon, 09 Apr 2007 13:31:12 GMT
Server: Apache/2.2.0
Connection: close
Content-Type: text/plain
```

```
Connection closed by foreign host.
```

Note how status 200 OK
is automatically constructed

CGI Environment

Section 4.1, RFC 3875

- AUTH_TYPE
- CONTENT_LENGTH
- CONTENT_TYPE
- GATEWAY_INTERFACE
- PATH_INFO
- PATH_TRANSLATED
- QUERY_STRING
- REMOTE_ADDR
- REMOTE_HOST
- REMOTE_IDENT
- REMOTE_USER
- REQUEST_METHOD
- SCRIPT_NAME
- SERVER_NAME
- SERVER_PORT
- SERVER_PROTOCOL
- SERVER_SOFTWARE

Current cs.odu.edu – not so much

```
$ curl -i https://www.cs.odu.edu/~mln/teaching/cs595-s12/cgi/env.cgi
```

```
HTTP/1.1 200 OK
```

```
Server: nginx
```

```
Date: Wed, 14 Nov 2018 02:44:15 GMT
```

```
Content-Type: text/html
```

```
Transfer-Encoding: chunked
```

```
Connection: keep-alive
```

```
Vary: Accept-Encoding
```

```
Front-End-Https: on
```

```
SSL_SESSION_ID = b1305bf2f47d8510d877d44e8c976c84b9899a03a171100d34f438c347729cd3 <br>
```

```
SCRIPT_NAME = /~mln/teaching/cs595-s12/cgi/env.cgi <br>
```

```
SSL_PROTOCOL = TLSv1.2 <br>
```

```
REQUEST_METHOD = GET <br>
```

```
HTTP_ACCEPT = */* <br>
```

```
SSL_COMPRESS_METHOD = NULL <br>
```

```
SCRIPT_FILENAME = /home/mln/secure_html/teaching/cs595-s12/cgi/env.cgi <br>
```

```
REQUEST_SCHEME = https <br>
```

```
SSL_CLIENT_VERIFY = NONE <br>
```

```
SSL_VERSION_INTERFACE = mod_ssl/2.4.10 <br>
```

```
SSL_VERSION_LIBRARY = OpenSSL/1.0.1 <br>
```

```
SERVER_SOFTWARE = Apache <br>
```

```
SSL_SERVER_I_DN_CN = web-home-2.cs.odu.edu <br>
```

```
QUERY_STRING = <br>
```

```
REMOTE_PORT = 45668 <br>
```

```
HTTP_USER_AGENT = curl/7.30.0 <br>
```

```
SERVER_SIGNATURE = <address>Apache Server at www.cs.odu.edu Port 443</address>  
<br>
```

```
SSL_SERVER_S_DN_CN = web-home-2.cs.odu.edu <br>
```

```
HTTP_X_SCHEME = https <br>
```

```
HTTP_X_REAL_IP = 70.177.203.225 <br>
```

```
PATH = /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin <br>
```

```
[continued from previous column]
```

```
SSL_SERVER_A_SIG = sha1WithRSAEncryption <br>
```

```
SSL_SECURE_RENEG = true <br>
```

```
GATEWAY_INTERFACE = CGI/1.1 <br>
```

```
SSL_SESSION_RESUMED = Resumed <br>
```

```
HTTPS = on <br>
```

```
SSL_CIPHER_USEKEYSIZE = 256 <br>
```

```
SSL_CIPHER_ALGKEYSIZE = 256 <br>
```

```
DOCUMENT_ROOT = /var/www/html <br>
```

```
SSL_SERVER_M_SERIAL = D9EFF3EC8A1F19C5 <br>
```

```
SSL_CIPHER_EXPORT = false <br>
```

```
SERVER_NAME = www.cs.odu.edu <br>
```

```
SSL_SERVER_S_DN = CN=web-home-2.cs.odu.edu <br>
```

```
SERVER_ADMIN = [no address given] <br>
```

```
HTTP_CONNECTION = close <br>
```

```
SSL_SERVER_V_END = Jun 8 19:41:33 2023 GMT <br>
```

```
CONTEXT_PREFIX = /~mln <br>
```

```
SSL_SERVER_V_START = Jun 10 19:41:33 2013 GMT <br>
```

```
HTTP_X_FORWARDED_PROTO = https <br>
```

```
SERVER_PORT = 443 <br>
```

```
SSL_SERVER_A_KEY = rsaEncryption <br>
```

```
REMOTE_ADDR = 128.82.4.81 <br>
```

```
CONTEXT_DOCUMENT_ROOT = /home/mln/secure_html <br>
```

```
SSL_CIPHER = ECDHE-RSA-AES256-GCM-SHA384 <br>
```

```
SERVER_PROTOCOL = HTTP/1.0 <br>
```

```
HTTP_X_FORWARDED_FOR = 70.177.203.225 <br>
```

```
REQUEST_URI = /~mln/teaching/cs595-s12/cgi/env.cgi <br>
```

```
SSL_SERVER_M_VERSION = 1 <br>
```

```
SERVER_ADDR = 172.18.8.46 <br>
```

```
SSL_SERVER_I_DN = CN=web-home-2.cs.odu.edu <br>
```

```
HTTP_HOST = www.cs.odu.edu <br>
```

<https://www.cs.odu.edu/~mln/teaching/cs595-s12/cgi/env.cgi>

Others still do the right thing

```
$ curl -i -H "Referer: http://www.cgi101.com/book/ch3"
http://www.cgi101.com/book/ch3/env.cgi
HTTP/1.1 200 OK
Date: Wed, 14 Nov 2018 02:47:25 GMT
Server: Apache/2.4.25 (Debian)
Vary: Accept-Encoding
Content-Length: 1272
Content-Type: text/html; charset=ISO-8859-1

<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en-US"
xml:lang="en-US">
<head>
<title>Environment</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"
/>
</head>
<body>
CONTEXT_DOCUMENT_ROOT = /home/cgi101/public_html<br>
CONTEXT_PREFIX = <br>
DOCUMENT_ROOT = /home/cgi101/public_html<br>
GATEWAY_INTERFACE = CGI/1.1<br>
HTTP_ACCEPT = /*<br>
HTTP_HOST = www.cgi101.com<br>
HTTP_REFERER = http://www.cgi101.com/book/ch3<br>
HTTP_USER_AGENT = curl/7.30.0<br>
PATH = /usr/local/bin:/usr/bin:/bin<br>
QUERY_STRING = <br>
```

```
[continued from previous column]
REMOTE_ADDR = 70.177.203.225<br>
REMOTE_PORT = 55124<br>
REQUEST_METHOD = GET<br>
REQUEST_SCHEME = http<br>
REQUEST_URI = /book/ch3/env.cgi<br>
SCRIPT_FILENAME =
/home/cgi101/public_html/book/ch3/env.cgi<br>
SCRIPT_NAME = /book/ch3/env.cgi<br>
SERVER_ADDR = 45.79.7.121<br>
SERVER_ADMIN = kira@lightsphere.com<br>
SERVER_NAME = www.cgi101.com<br>
SERVER_PORT = 80<br>
SERVER_PROTOCOL = HTTP/1.1<br>
SERVER_SIGNATURE = <address>Apache/2.4.25 (Debian)
Server at www.cgi101.com Port 80</address>
<br>
SERVER_SOFTWARE = Apache/2.4.25 (Debian)<br>
UNIQUE_ID = W@uMvX8AAQEAAHgseREAAAAK<br>

</body>
```

<http://www.cgi101.com/book/ch3/>

How to Customize the Environment?

- C:
 - `setenv()`
 - `fork()` & `execve()`
- Perl:
 - `set %ENV`
 - `fork()` & `exec()`
- Python:
 - `os.environ`
 - `os.fork()` & `os.execve()`
- Others? Please share in the discussions!

ENV & CGI Examples

```
#!/usr/bin/perl

print "Content-type: text/html\n\n";

foreach $key (keys (%ENV)) {
    print "$key = $ENV{$key} <br>\n";
}

while (<STDIN>) {
    print "$_<br>\n";
}
```

{GET, POST}

X

{multipart/form-data, application/x-form-www-urlencoded}

<http://www.cs.odu.edu/~mln/teaching/cs595-s12/cgi/>