

# HTTP

## EC512

### Spring 2015

# HTTP

- HTTP is the standard protocol used between a web browser and a web server.
- It is standardized by the World Wide Web Consortium, W3C [www.w3c.org](http://www.w3c.org)
- The latest standard documentation can be found at [www.w3.org/Protocols/HTTP/1.1/rfc2616.pdf](http://www.w3.org/Protocols/HTTP/1.1/rfc2616.pdf)
- **Assignment** - read the HTTP standard document at least carefully enough to use it as a future reference.

# HTTP Operation

- The browser and web server communicate with a simple *request/response* mechanism.
- When you perform some action in your browser (open a URL, activate a link etc.) a request header is generated that is sent to the web server along with a possible *body* of information.
- The web server sends back a response header with the document you requested in the *body* of the response. Normally this is your HTML page.
- Images and other HTML elements are requested and responded to in a similar manner.
- A MIME type in the header specifies the type of the information in the body.

# Fiddler

- You can't look at requests and responses in your browser
- View source only shows the HTML
- A means to capture the raw data is needed
- Fiddler 2, the free tool from [www.fiddlertool.com](http://www.fiddlertool.com) is a great utility
- You should install it on your system
- Demo in class

# Request Header

The screenshot shows the 'Request Headers' tab in a web browser's developer tools. The request method is 'GET / HTTP/1.1'. The headers are organized into two sections: 'Client' and 'Transport'.

**Client**

- Accept: \*/\*
- Accept-Language: en-us
- UA-CPU: x86
- Accept-Encoding: gzip, deflate
- User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET CLR

**Transport**

- Host: 192.168.1.103
- Proxy-Connection: Keep-Alive

# Response Header

Transformer Headers TextView ImageView HexView Auth Caching Privacy Raw XML

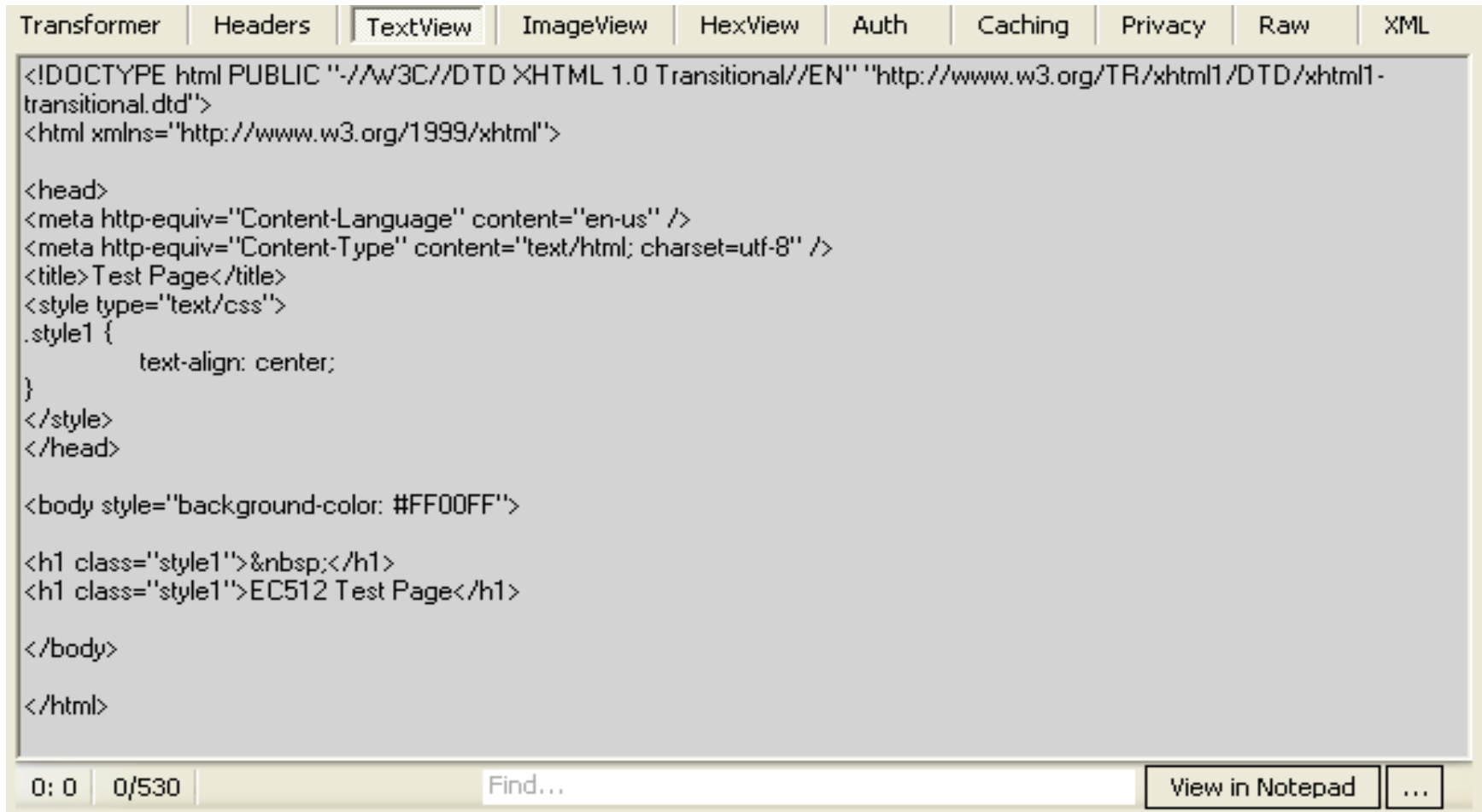
Response Headers [Raw] [Header Definitions]

HTTP/1.1 200 OK

- [-] **Cache**
  - ... Date: Sat, 26 Jan 2008 23:25:55 GMT
- [-] **Entity**
  - ... Content-Type: text/html
  - ... Last-Modified: Sat, 26 Jan 2008 22:40:59 GMT
  - ... ETag: "58577d856c60c81:d02"
  - ... Content-Length: 555
- [-] **Miscellaneous**
  - ... Server: Microsoft-IIS/5.1
  - ... X-Powered-By: ASP.NET
  - ... Content-Location: http://192.168.1.103/Default.htm
  - ... Accept-Ranges: bytes

MIME Type

# Response Data



The image shows a screenshot of a web browser's developer tools interface, specifically the 'Raw' tab. The main content area displays the raw HTML response data. The HTML includes a DOCTYPE declaration, a meta tag for content language (en-us), a meta tag for content type (text/html; charset=utf-8), a title 'Test Page', and a style block for a class named 'style1' with a text-align: center property. The body contains a background-color of #FF00FF, a space character, and the text 'EC512 Test Page'.

```
<!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">

<head>
<meta http-equiv="Content-Language" content="en-us" />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Test Page</title>
<style type="text/css">
.style1 {
    text-align: center;
}
</style>
</head>

<body style="background-color: #FF00FF">

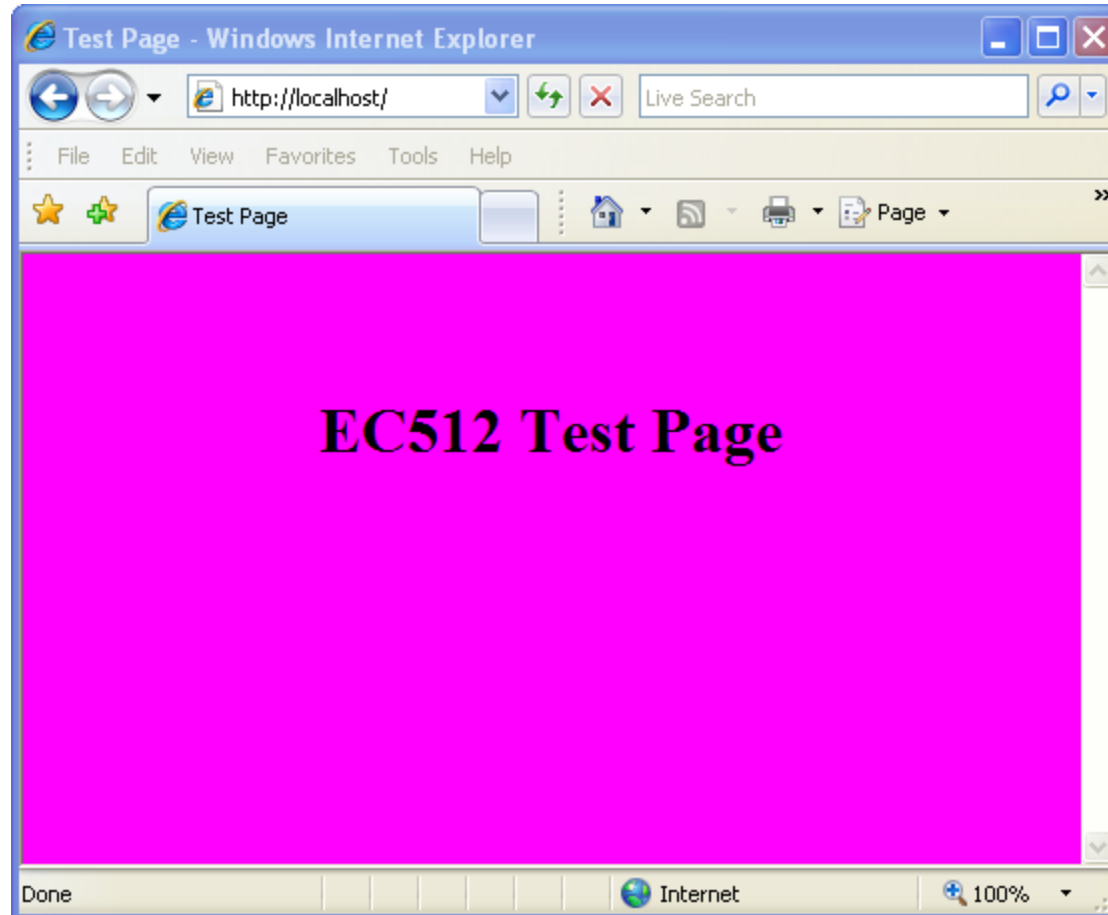
<h1 class="style1">&nbsp;</h1>
<h1 class="style1">EC512 Test Page</h1>

</body>

</html>
```

0: 0 | 0/530 | Find... | View in Notepad | ...

# The Page





# The Common Gateway Interface CGI

- CGI is the fundamental way in which HTTP is used to communicate with server side applications, e.g., Perl scripts, ASP, ASP.NET etc.
- CGI is very simple.
- The HTML form is central to the operation of CGI.
- The data in a form is passed to the server using one of two methods:
  1. GET Method
  2. POST Method
- The following example shows the use of CGI without form arguments.
- It is written in C although other languages can be used.

# Simple CGI Program

(prints environment variables)

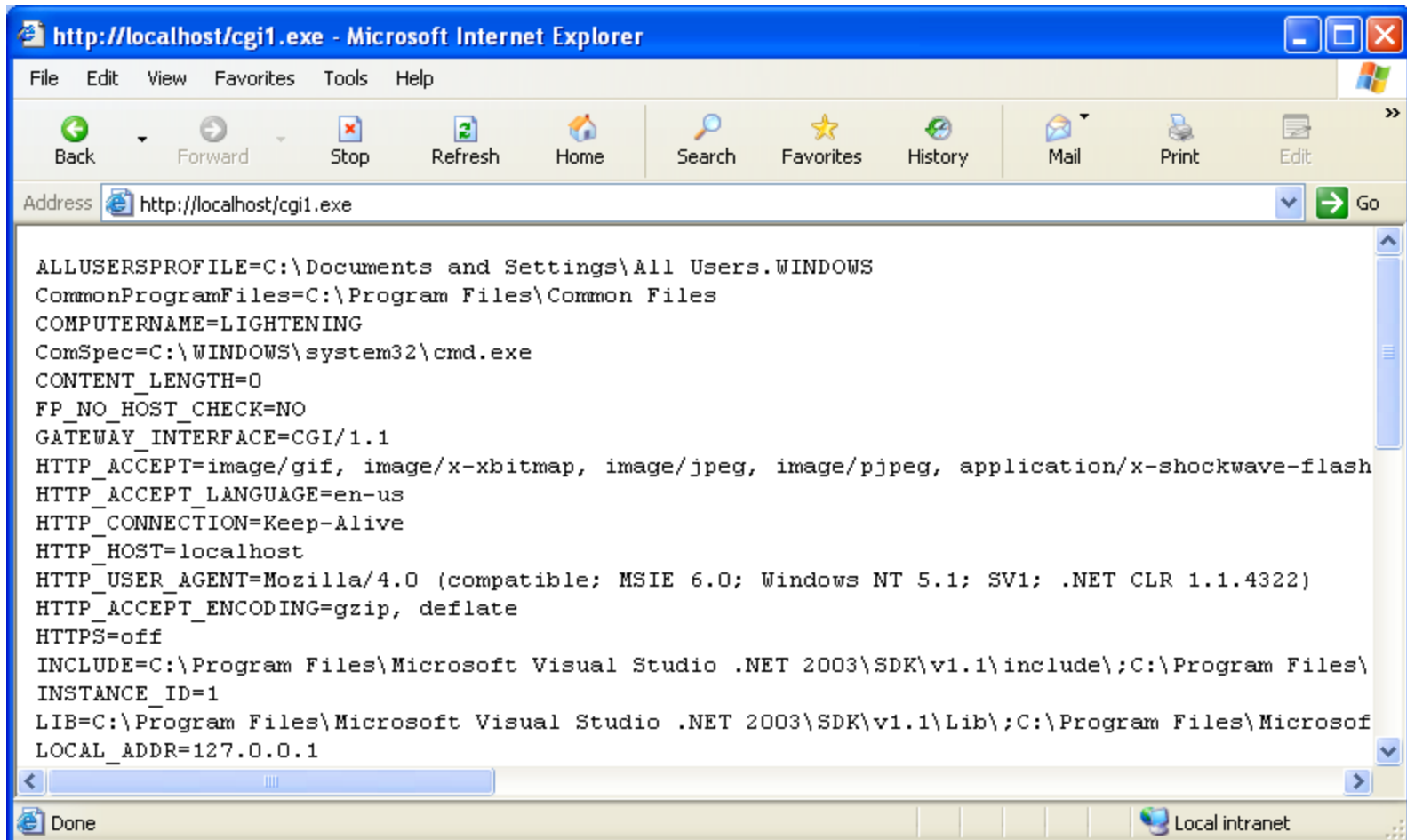
```
#include <stdio.h>

int main(int argc, char* argv[], char* envp[])
{
    int i=0;

    printf("Content-type: text/plain\n\n");

    while(envp[i])
    {
        printf("%s\n", envp[i]);
        ++i;
    }
    return 0;
}
```

# Output

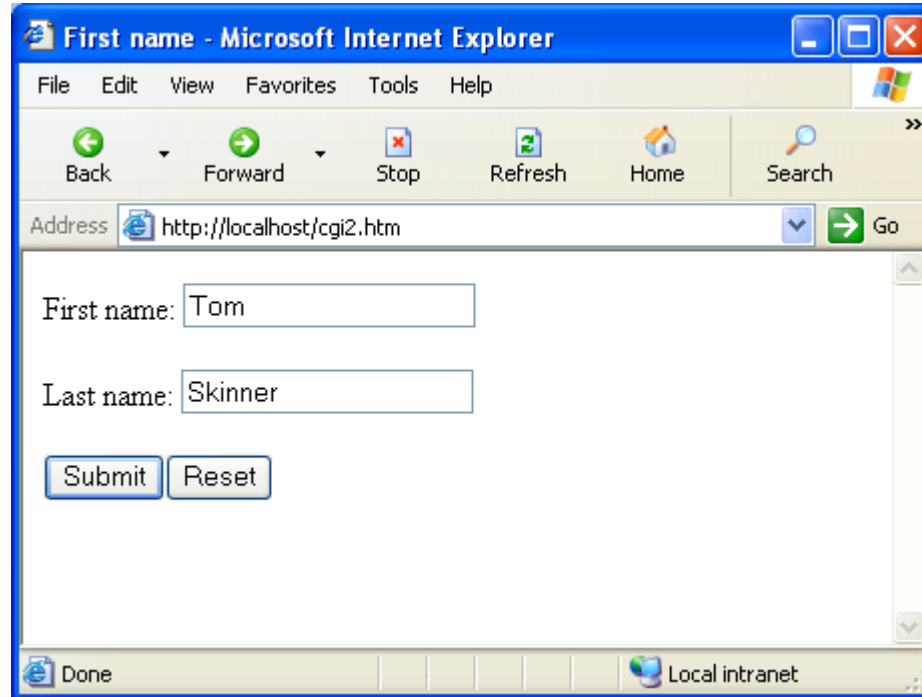


```
ALLUSERSPROFILE=C:\Documents and Settings\All Users\WINDOWS
CommonProgramFiles=C:\Program Files\Common Files
COMPUTERNAME=LIGHTENING
ComSpec=C:\WINDOWS\system32\cmd.exe
CONTENT_LENGTH=0
FP_NO_HOST_CHECK=NO
GATEWAY_INTERFACE=CGI/1.1
HTTP_ACCEPT=image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash
HTTP_ACCEPT_LANGUAGE=en-us
HTTP_CONNECTION=Keep-Alive
HTTP_HOST=localhost
HTTP_USER_AGENT=Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)
HTTP_ACCEPT_ENCODING=gzip, deflate
HTTPS=off
INCLUDE=C:\Program Files\Microsoft Visual Studio .NET 2003\SDK\v1.1\include\;C:\Program Files\
INSTANCE_ID=1
LIB=C:\Program Files\Microsoft Visual Studio .NET 2003\SDK\v1.1\Lib\;C:\Program Files\Microsof
LOCAL_ADDR=127.0.0.1
```

# A Simple Form

```
<form method="GET" action="cgi2.exe">
  <p>First name: <input type="text" name="first"
    size="20"></p>
  <p>Last name: <input type="text" name="last"
    size="20"></p>
  <p><input type="submit" value="Submit" name="B1">
  <input type="reset" value="Reset" name="B2"></p>
</form>
```

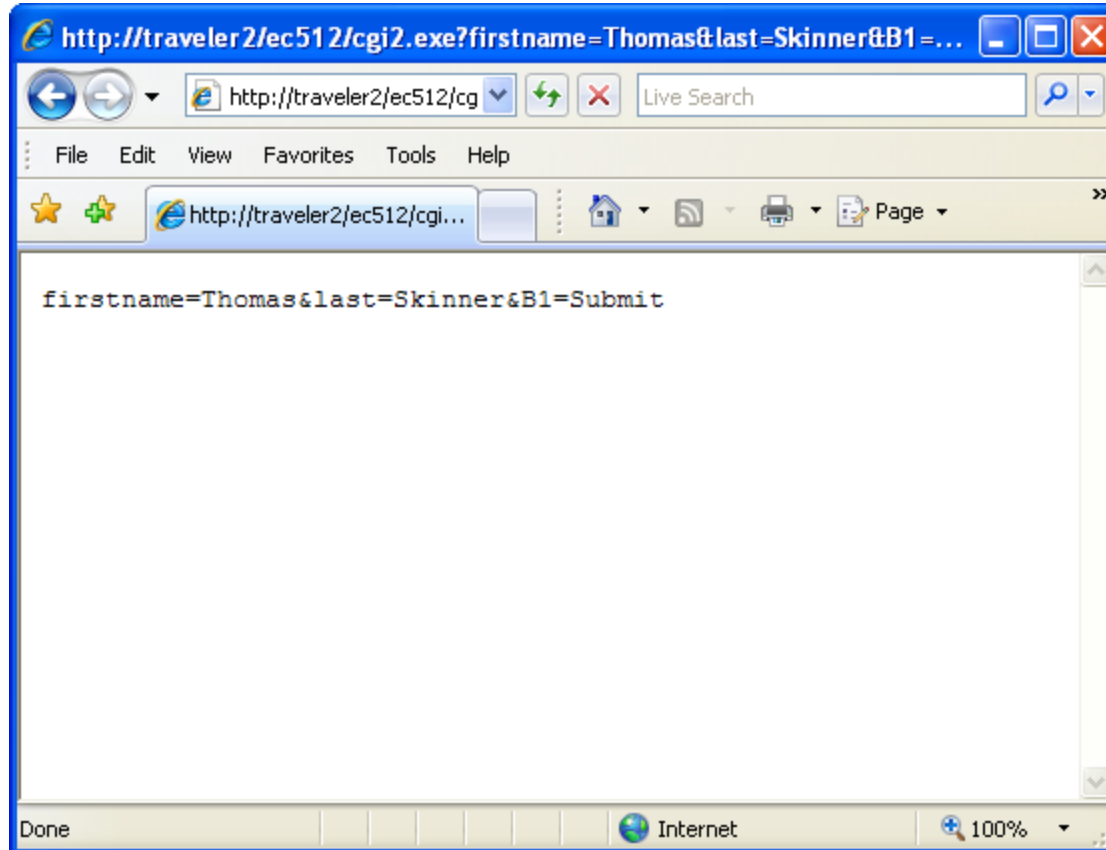
# Simple Form Display



# Print Query String

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char* argv[])
{
    char* s;
    s=getenv("QUERY_STRING");
    printf("Content-type: text/plain\n\n");
    printf("%s\n", s);
    return 0;
}
```

# Query String Display



# A Look at the Headers

The screenshot shows the 'Request Headers' section of a web browser's developer tools. The tabs at the top are Headers, TextView, WebForms, HexView, Auth, Raw, and XML. The main content area displays the following information:

**Request Headers** [Raw] [Header Definitions]

GET /ec512/cgi2.exe?firstname=Thomas&last=Skinner&B1=Submit HTTP/1.1

- Client**
  - Accept: \*/\*
  - Accept-Language: en-us
  - UA-CPU: x86
  - Accept-Encoding: gzip, deflate
  - User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET CLR
- Transport**
  - Proxy-Connection: Keep-Alive
  - Host: traveler2
- Miscellaneous**
  - Referer: http://traveler2/ec512/cgi2.htm



Transformer Headers TextView ImageView HexView Auth Caching Privacy Raw XML

**Response Headers** [\[ Raw \]](#) [\[ Header Definitions \]](#)

HTTP/1.1 200 OK

- Transport**
  - Connection: close
- Cache**
  - Date: Sun, 27 Jan 2008 15:11:41 GMT
- Entity**
  - Content-type: text/plain
- Miscellaneous**
  - Server: Microsoft-IIS/5.1
  - X-Powered-By: ASP.NET

Transformer Headers Text View ImageView HexView Auth Caching Privacy Raw XML

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.1
Date: Sun, 27 Jan 2008 15:11:41 GMT
X-Powered-By: ASP.NET
Connection: close
Content-type: text/plain

firstname=Thomas&last=Skinner&B1=Submit
```

Find... View in Notepad

# Get and Post Methods

- With the GET method arguments are passed as an extended URL.
- Some characters need to be *escaped*.
- The POST method passes arguments in the body of the request.
- Both methods are commonly used and the particular method is specified in the <FORM> tag.
- POST method args are passed to the CGI program via standard input.

# A Simple Form

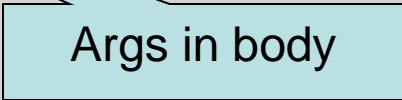
```
<form method="POST" action="cgi3.exe">  
  <p>First name: <input type="text" name="first"  
    size="20"></p>  
  <p>Last name: <input type="text" name="last"  
    size="20"></p>  
  <p><input type="submit" value="Submit" name="B1">  
  <input type="reset" value="Reset" name="B2"></p>  
</form>
```

# CGI program to echo POST method data (cgi3.exe)

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char* argv[])
{
    char s[1024];
    char* ss;
    printf("Content-type: text/plain\n\n");
    ss=getenv("CONTENT_LENGTH");
    if(ss)
    {
        int count=atoi(ss);
        for(int i=0; i<count; ++i)
            putchar(getchar());
    }
    return 0;
}
```

```
Headers | TextView | WebForms | HexView | Auth | Raw | XML
POST /ec512/cgi2.exe HTTP/1.1
Accept: */*
Referer: http://traveler2/ec512/cgi2.htm
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
UA-CPU: x86
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322;
.NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.0.04506.648; .NET CLR
3.5.21022)
Proxy-Connection: Keep-Alive
Content-Length: 39
Host: traveler2
Pragma: no-cache

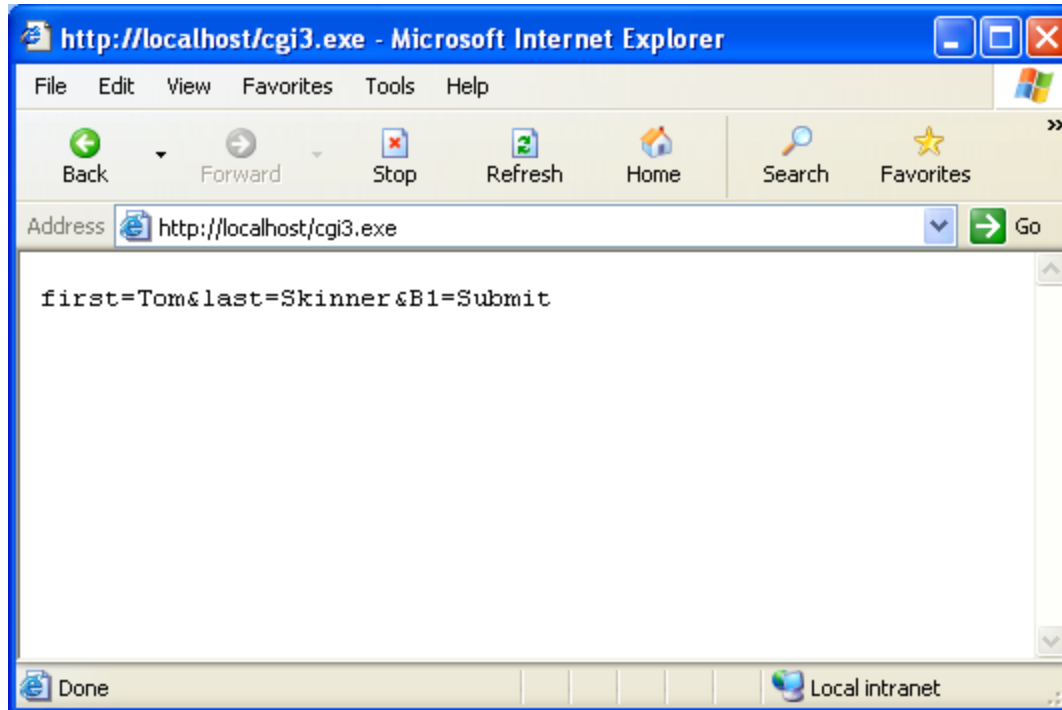
firstname=Thomas&last=Skinner&B1=Submit
```



Args in body

Hex Find... View in Notepad

# Earlier form using POST method and cgi3.exe as the action



# New Uses of HTTP

- In recent years HTTP has been used as a transport protocol not exclusively for displaying web pages in web browsers.
- HTTP can be used to upload and download files.
- HTTP can be used to effectively provide network shares, WebDAV etc. See <http://en.wikipedia.org/wiki/WebDAV>
- Remote desktop type applications.
- HTTP provides transport for XML/SOAP web services.
- Central to all these uses is that a web server with extended capabilities is involved.