HTTP
Getting What We Ask For

• HTTP: How Browsers & Servers Communicate
  • HTTP 1.1: http://www.w3.org/Protocols/
  • TCP Connection, usually over port 80
  • Text Based Instructions
  • Simple Verbs
    • GET, POST, PUT, DELETE, HEAD, CONNECT, OPTIONS, TRACE
  • Optional Headers

• Basic GET Example

<table>
<thead>
<tr>
<th>GET / HTTP/1.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host: <a href="http://www.example.com">www.example.com</a></td>
</tr>
</tbody>
</table>

• HOST header is required for HTTP/1.1

• Two CRLF to indicate the request has finished
  • CRLF = \r\n  Although most Web Servers will accept \n
"Although the line terminator for the start-line and header fields is the sequence CRLF, a recipient MAY recognize a single LF as a line terminator and ignore any preceding CR."


• Verbs and HTTP versions are Case Sensitive

<table>
<thead>
<tr>
<th>GET / HTTP/1.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host: example.com</td>
</tr>
<tr>
<td>HTTP/1.1 501 Not Implemented</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GET / http/1.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host: example.com</td>
</tr>
<tr>
<td>HTTP/1.0 505 HTTP Version Not Supported</td>
</tr>
</tbody>
</table>

• Headers are not Case Sensitive

<table>
<thead>
<tr>
<th>GET / HTTP/1.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host: exAMPle.com</td>
</tr>
<tr>
<td>HTTP/1.1 200 OK</td>
</tr>
</tbody>
</table>
Basic HTTP Example

GET / HTTP/1.1
host: example.com

HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: max-age=604800
Content-Type: text/html
Date: Mon, 21 Jul 2014 05:04:02 GMT
Etag: "359670651"
Expires: Mon, 28 Jul 2014 05:04:02 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (cpm/F858)
X-Cache: HIT
x-ec-custom-error: 1
Content-Length: 1270

<!doctype html>
<html>
<head>
<title>Example Domain</title>
</head>
<body>
<div>
<h1>Example Domain</h1>
<p>This domain is established to be used for illustrative examples in documents. You may use this domain in examples without prior coordination or asking for permission.</p>
<p><a href="http://www.iana.org/domains/example">More information...</a></p>
</div>
</body>
</html>

Command Line Basics

Program Name: telnet
Program Arguments:

> telnet example.com 80

Command Prompt: (don't type this part)
Press the return key at the end to run the program

HTTP With Telnet

- **telnet** is a very simple program that basically opens a TCP connection to a host
- Key parts: **host** and **port**
We typed in this stuff

Local `telnet` program prints this

Remote server sends this back

This part is NOT part of an HTTP request! This is just setting up `telnet` to issue a request

This part is the HTTP request

HTTP With Telnet

Telnet Example

Telnet On a Mac

- Applications folder → Utilities Folder
  - Terminal

- Just type in `telnet example.com 80` and hit return
Telnet On a Windows

- Windows ships with telnet, but it's disabled.
- Search: "enable telnet on windows X"
  - where X is your version of Windows.
- Here's a great explanation for Windows 10

Telnet On Linux

- If you're running Linux on your desktop, you already know how to use telnet. 😎

```
curl -v http://example.com
```

Request

Response Headers

Response Body

```
<!doctype html>
<html>
<head>
<title>Example Domain</title>
</head>
<body>

</body>
</html>
```

curl Example
Examine Requests in Chrome

Response Codes

- Informational: 1xx
- Successful: 2xx
  - 200 OK
- Redirection: 3xx
  - 301 Moved
- Client Error: 4xx
  - 404 Not Found
- Server Error: 5xx
  - 500 Internal Server Error

http://tools.ietf.org/html/rfc7231#page-4

HTTP/2.0

- New binary method of allowing multiple requests through a single TCP socket
- More of a change to how the protocol is implemented on the wire than in the concepts of how the protocol works
- Advanced topic, if you’re interested in more details:
- Otherwise, just know its a thing