HTTP by **Hand**

Exploring HTTP/1.x

Looking forward to HTTP/2

@bantic
Cory Forsyth

201 Created

We build 🛸-age apps with Ember.js. We take teams from 🛵 to 🏎 in no time flat.

Why by hand?

- Why let browsers have all the fun?
- HTTP is human-scale

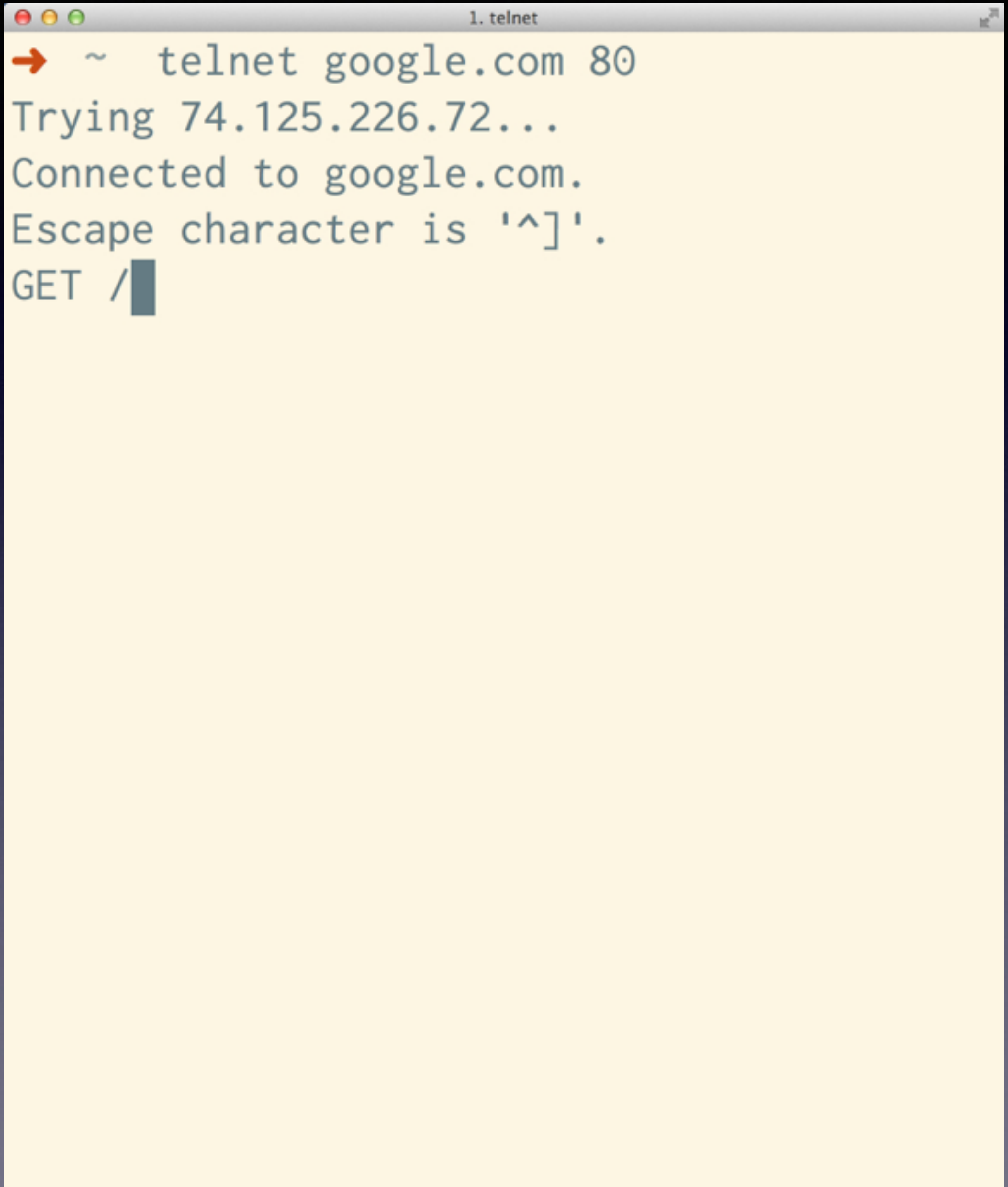
How we (a)buse HTTP

- Asset host sharding
- Concatenation
- Spriting

What is HTTP?

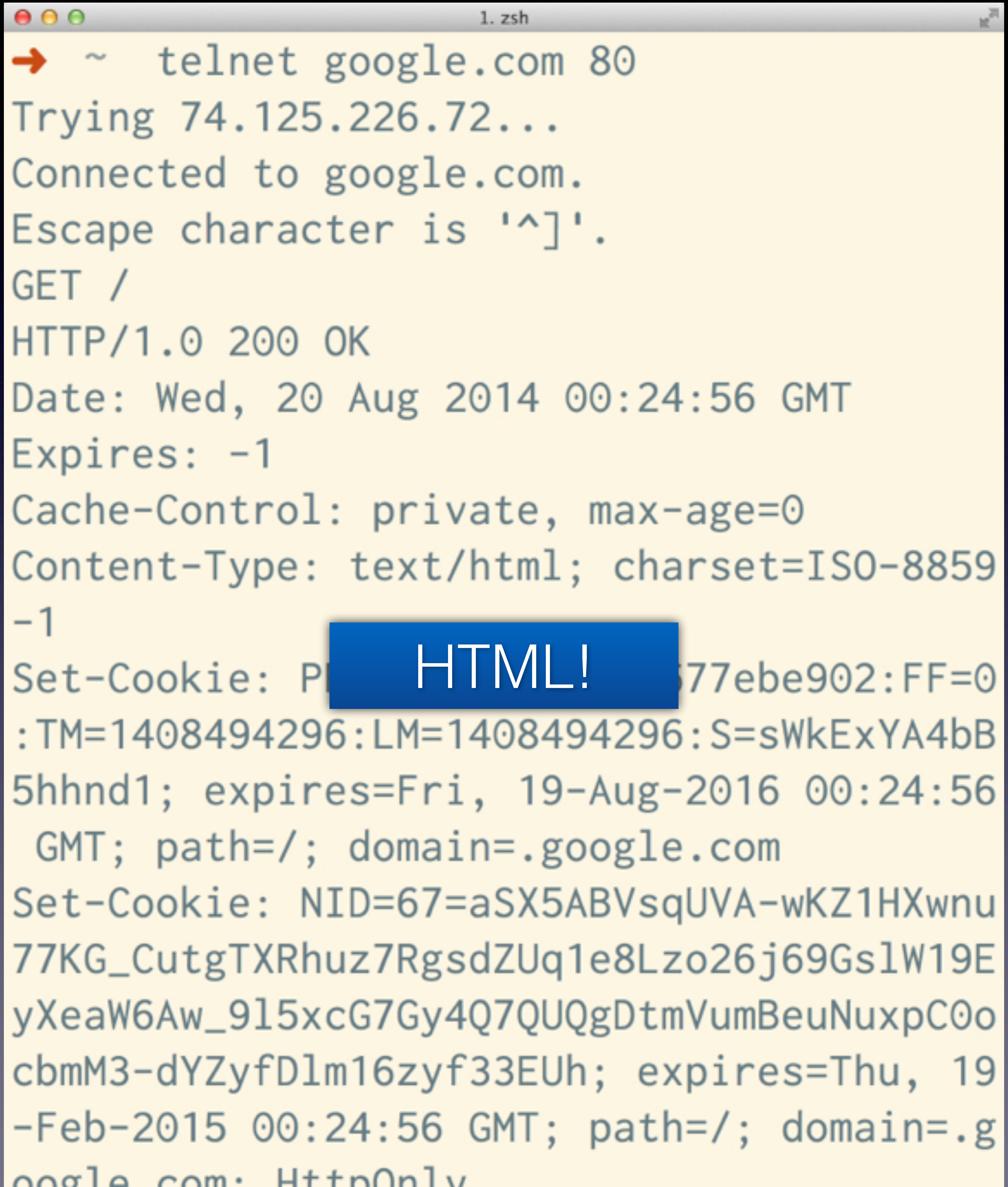
Let’s get some HTML

HTTP/0.9



1. telnet

```
➜  ~  telnet google.com 80
Trying 74.125.226.72...
Connected to google.com.
Escape character is '^]'.
GET /
```



```
➜  ~  telnet google.com 80
Trying 74.125.226.72...
Connected to google.com.
Escape character is '^]'.
GET /
HTTP/1.0 200 OK
Date: Wed, 20 Aug 2014 00:24:56 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859
-1
Set-Cookie: P                  77ebe902:FF=0
:TM=1408494296:LM=1408494296:S=sWkExYA4bB
5hhnd1; expires=Fri, 19-Aug-2016 00:24:56
 GMT; path=/; domain=.google.com
Set-Cookie: NID=67=aSX5ABVsqUVA-wKZ1HXwnu
77KG_CutgTXRhuz7RgsdZUq1e8Lzo26j69GslW19E
yXeaW6Aw_915xcG7Gy4Q7QUQgDtmVumBeuNuxpC0o
cbmM3-dYZyfDlm16zyf33EUh; expires=Thu, 19
-Feb-2015 00:24:56 GMT; path=/; domain=.g
oogle.com; HttpOnly
```

HTML!

HTTP/0.9

One-line Request Format

Not really a spec

```
GET /
```

Let’s get some HTML

HTTP/1.0

HTTP/1.0 Spec

Request

```
Request  =  Request-Line                          ; Section 5.1
             *(( general-header                   ; Section 4.5
                 | request-header                 ; Section 5.3
                 | entity-header ) CRLF)          ; Section 7.1
                                CRLF
                 [ message-body ]                 ; Section 4.3
```

HTTP/1.0 Spec

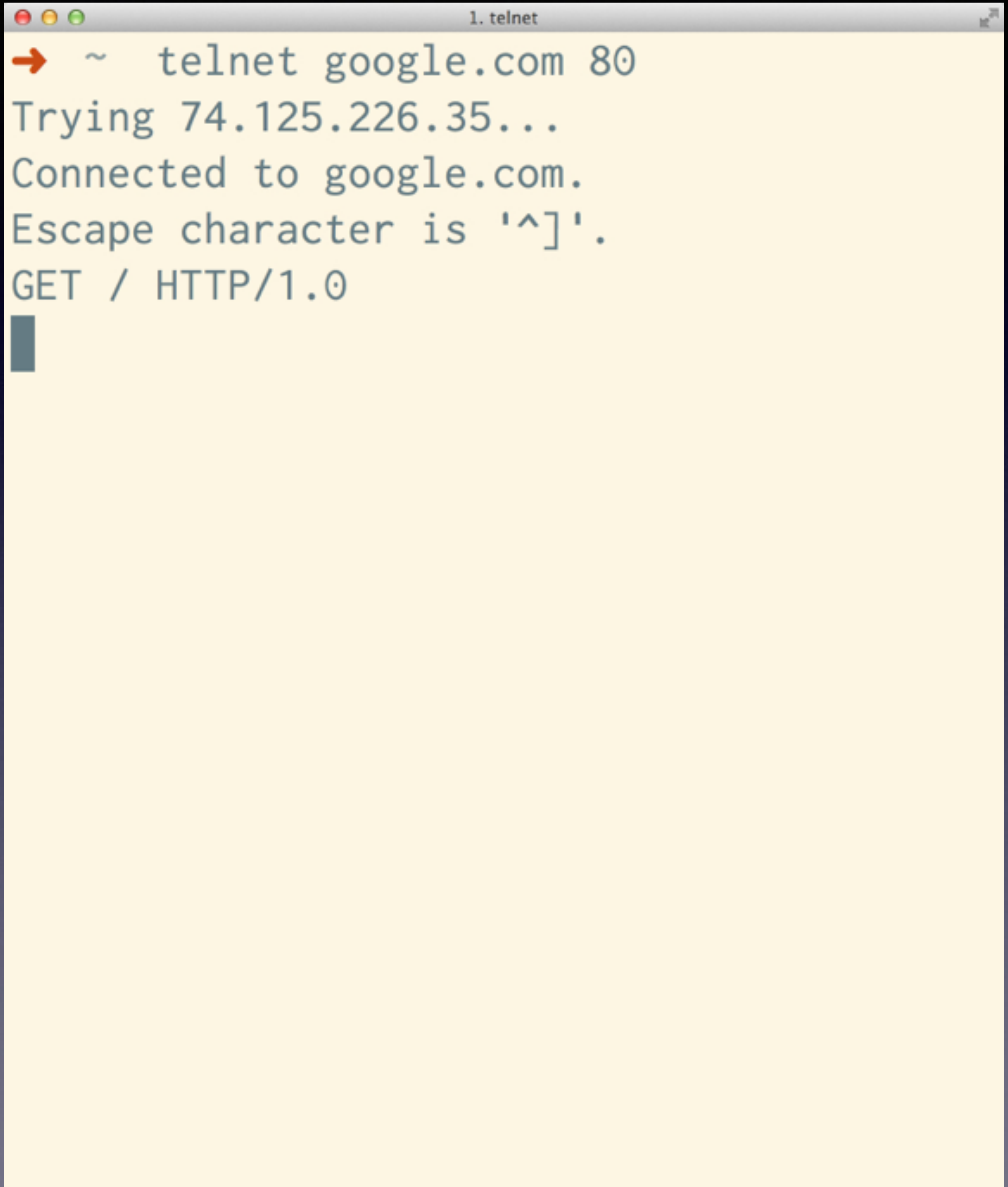
Request-Line

```
Request-Line   = Method SP Request-URI SP HTTP-Version CRLF
```

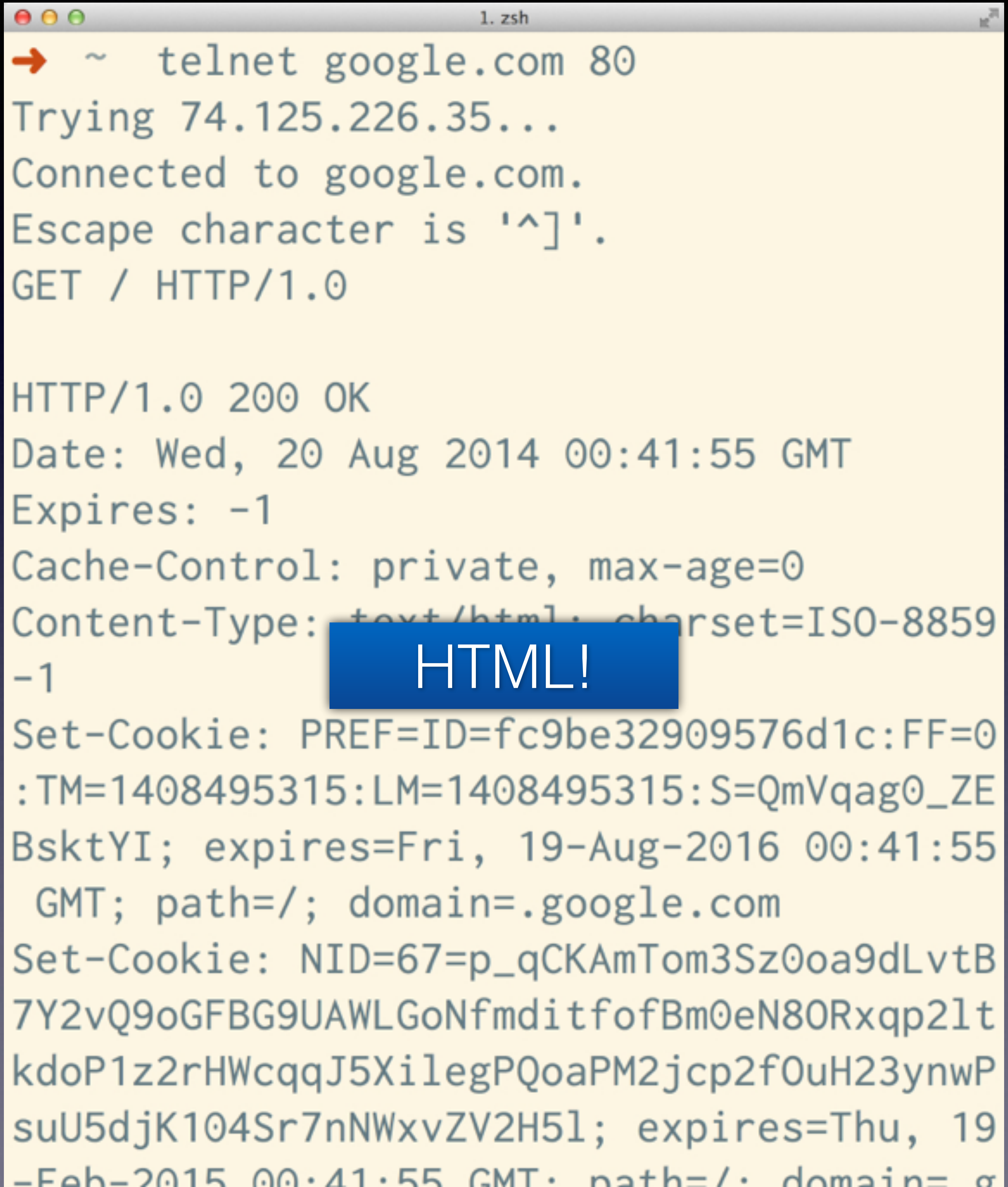
HTTP/1.0 Spec

Request-Line

Request-Line = GET / HTTP/1.0

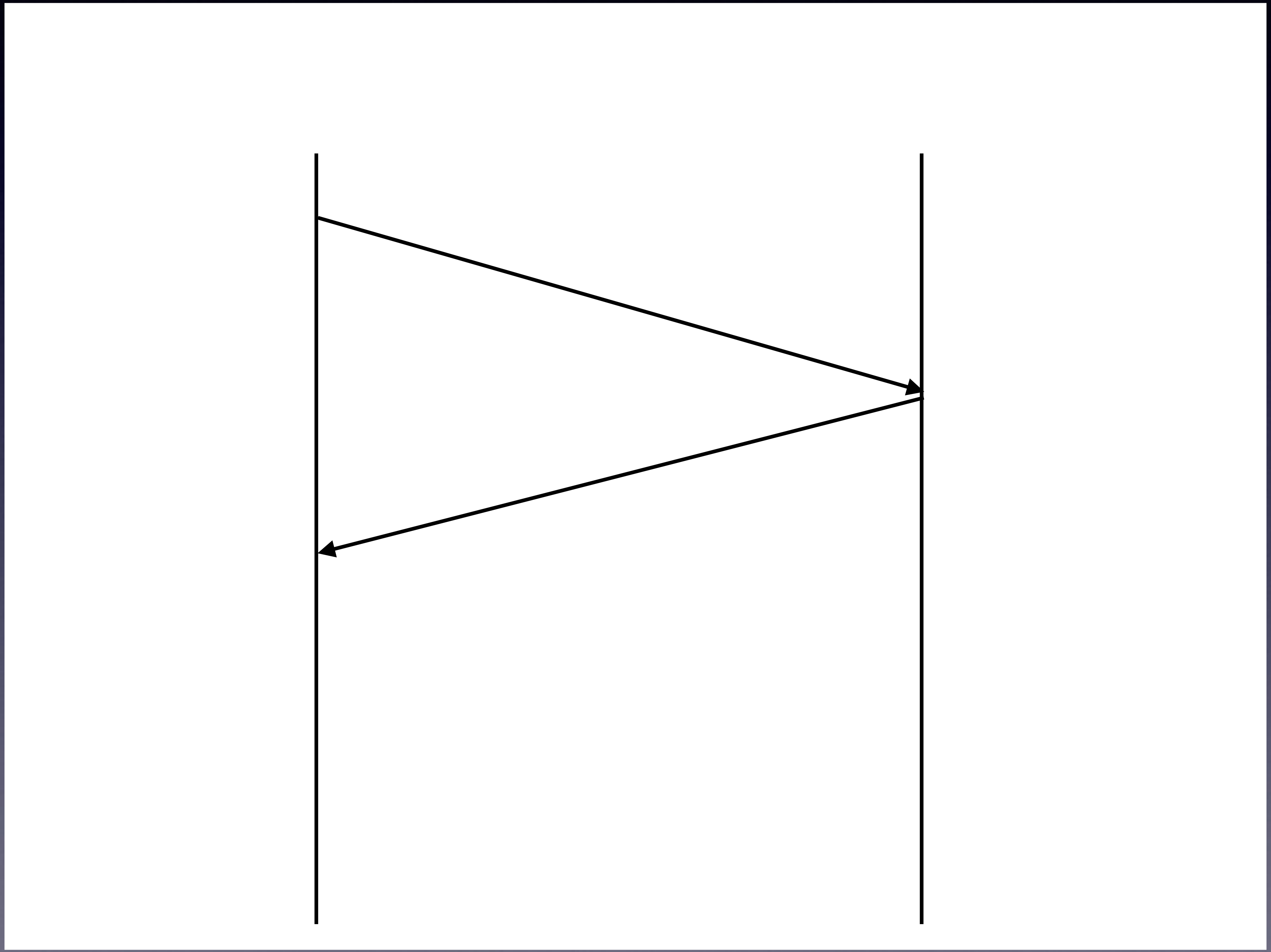


```
➜  ~  telnet google.com 80
Trying 74.125.226.35...
Connected to google.com.
Escape character is '^]'.
GET / HTTP/1.0

```

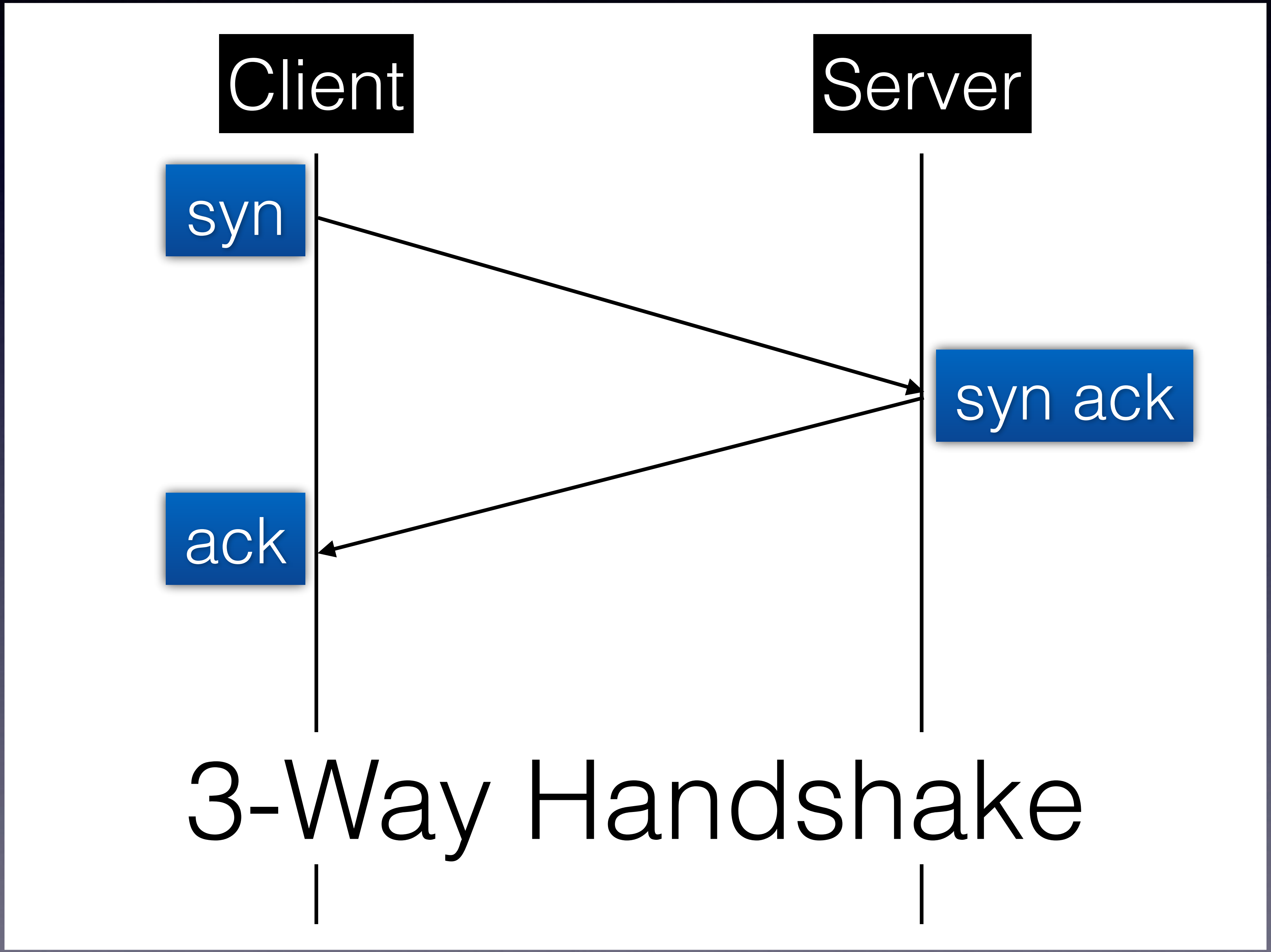


```
➜  ~  telnet google.com 80
Trying 74.125.226.35...
Connected to google.com.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.0 200 OK
Date: Wed, 20 Aug 2014 00:41:55 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859
-1
Set-Cookie: PREF=ID=fc9be32909576d1c:FF=0
:TM=1408495315:LM=1408495315:S=QmVqag0_ZE
BsktYI; expires=Fri, 19-Aug-2016 00:41:55
 GMT; path=/; domain=.google.com
Set-Cookie: NID=67=p_qCKAmTom3Sz0oa9dLvtB
7Y2vQ9oGFBG9UAWLGoNfmditfofBm0eN8ORxqp2lt
kdoP1z2rHWcqqJ5XilegPQoaPM2jcp2f0uH23ynwP
suU5djK104Sr7nNWxvZV2H5l; expires=Thu, 19
-Feb-2015 00:41:55 GMT; path=/; domain=.g
```

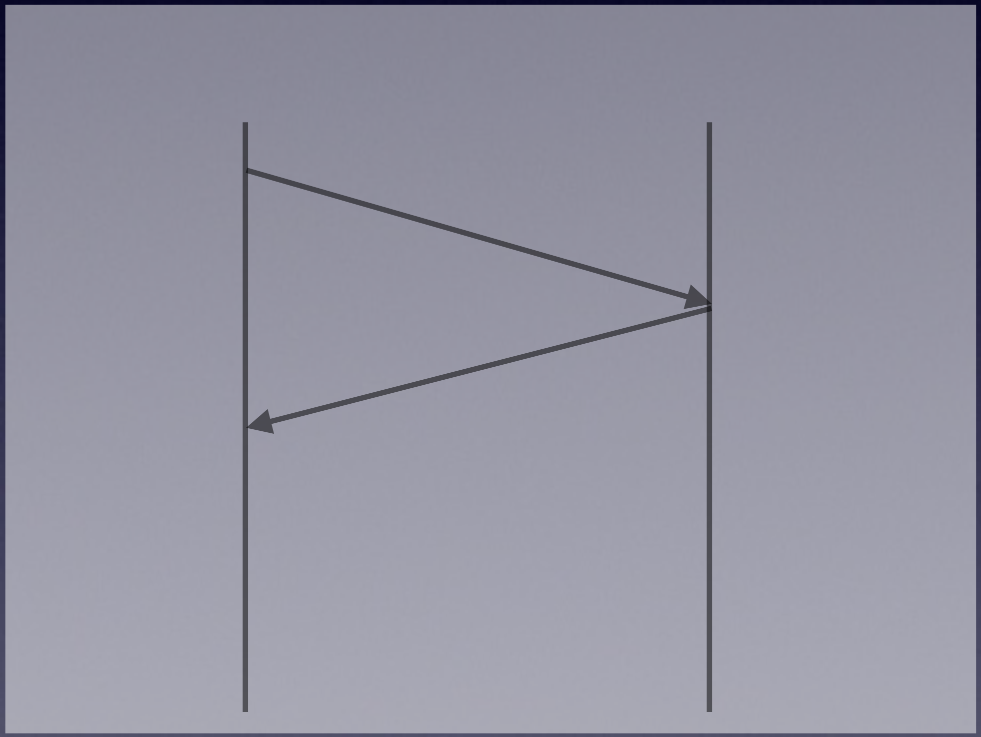
Quick Aside: TCP



Quick Aside: TCP

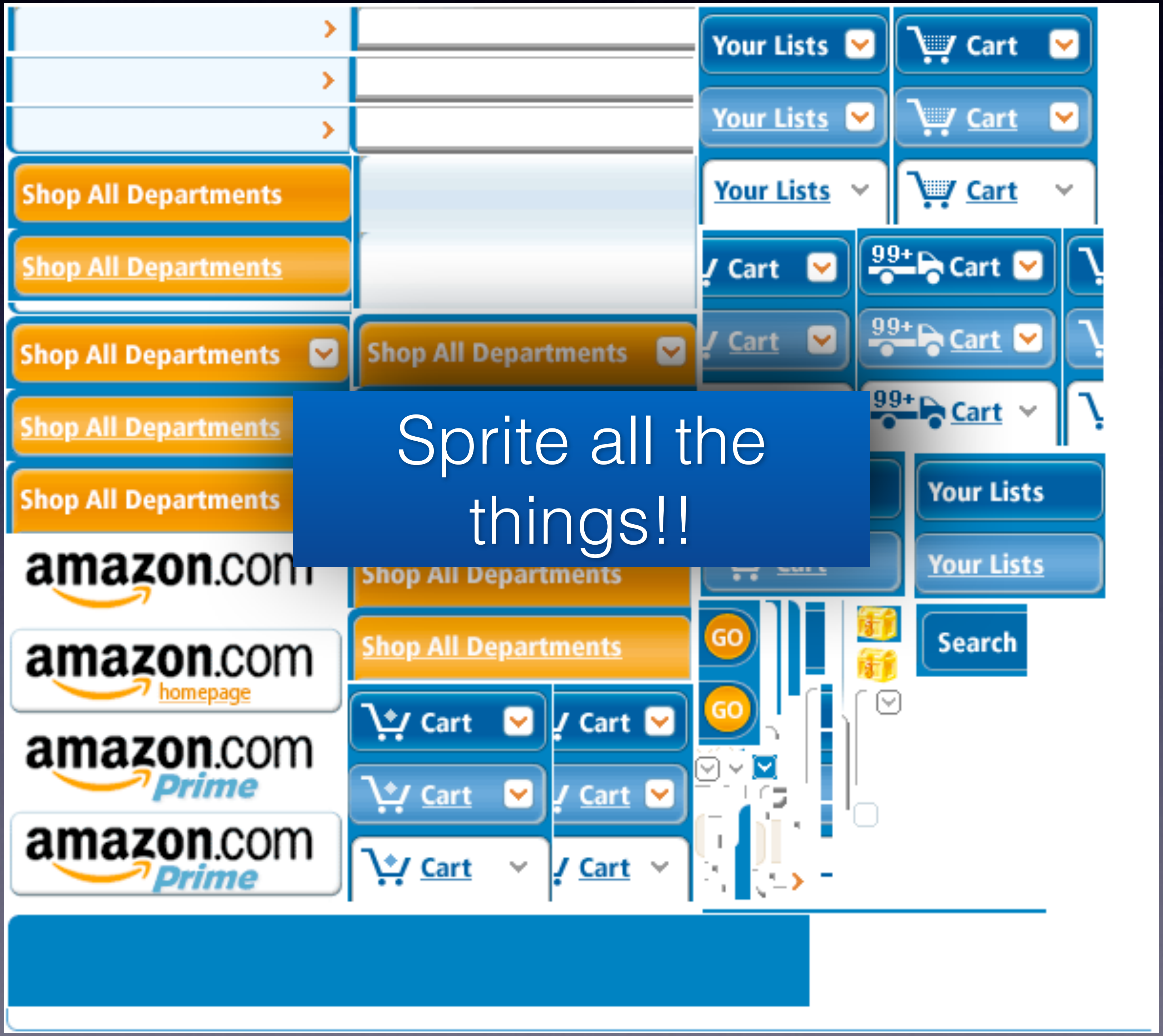


Quick Aside: TCP



- Minimum 1 Round Trip Per Request
- Can’t make speed of light faster
- How can we avoid this latency?

How we (a)buse HTTP



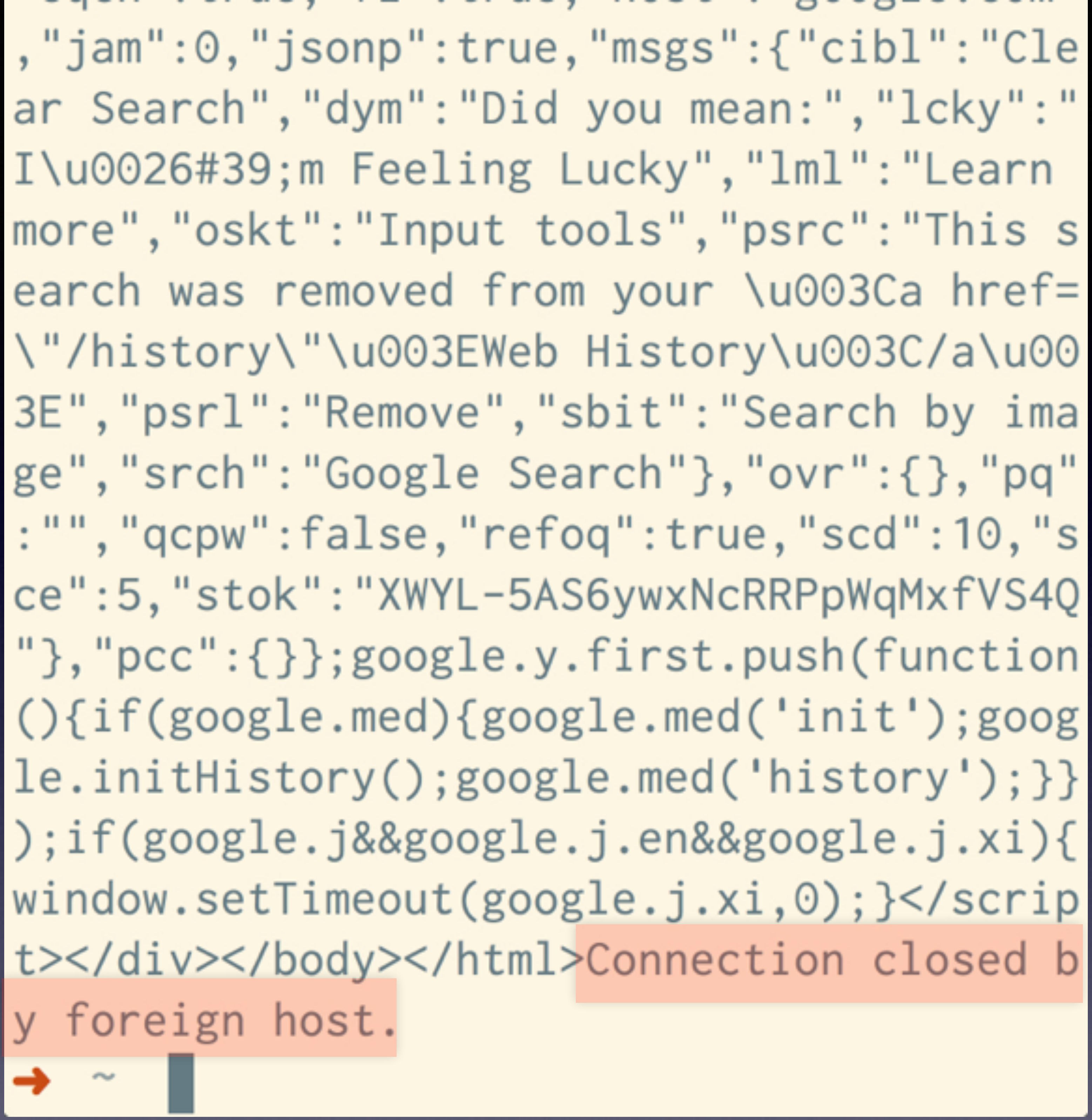
Sprite all the things!!

HTTP/1.0 Spec

One connection, one response

"requires … the connection be … closed by the server after sending the response."

–HTTP/1.0 Spec



```
,"jam":0,"jsonp":true,"msgs":{"cibl":"Cle
ar Search","dym":"Did you mean:","lcky":"
I\u0026#39;m Feeling Lucky","lml":"Learn 
more","oskt":"Input tools","psrc":"This s
earch was removed from your \u003Ca href=
\"/history\"\u003EWeb History\u003C/a\u00
3E","psrl":"Remove","sbit":"Search by ima
ge","srch":"Google Search"},"ovr":{},"pq"
:"","qcpw":false,"refoq":true,"scd":10,"s
ce":5,"stok":"XWYL-5AS6ywxNcRRPpWqMxfVS4Q
"},"pcc":{}};google.y.first.push(function
(){if(google.med){google.med('init');goog
le.initHistory();google.med('history');}}
);if(google.j&&google.j.en&&google.j.xi){
window.setTimeout(google.j.xi,0);}</scrip
t></div></body></html>Connection closed b
y foreign host.
➜  ~  
```

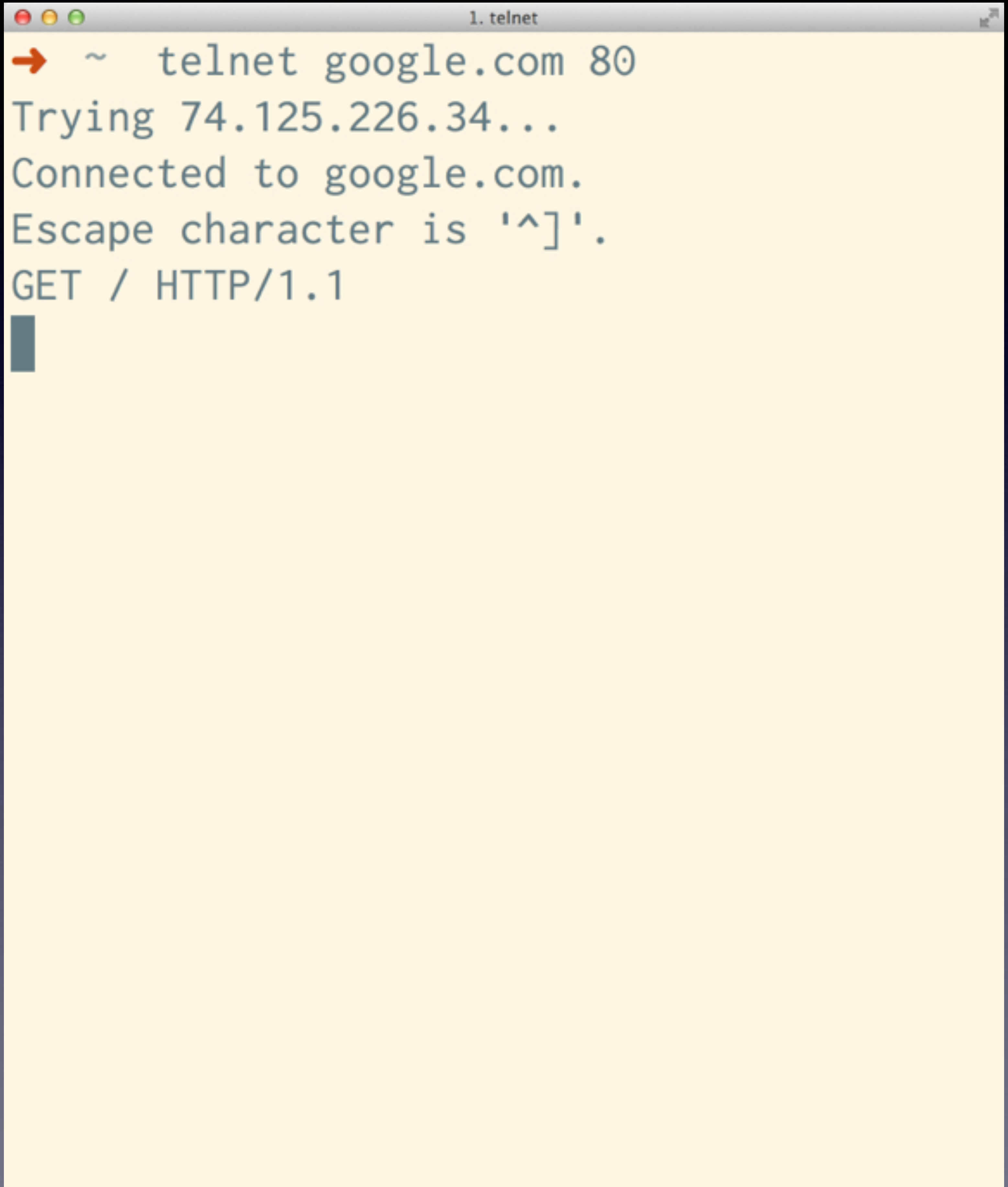
HTTP/1.1 Spec

"HTTP/1.1 servers SHOULD maintain persistent connections"

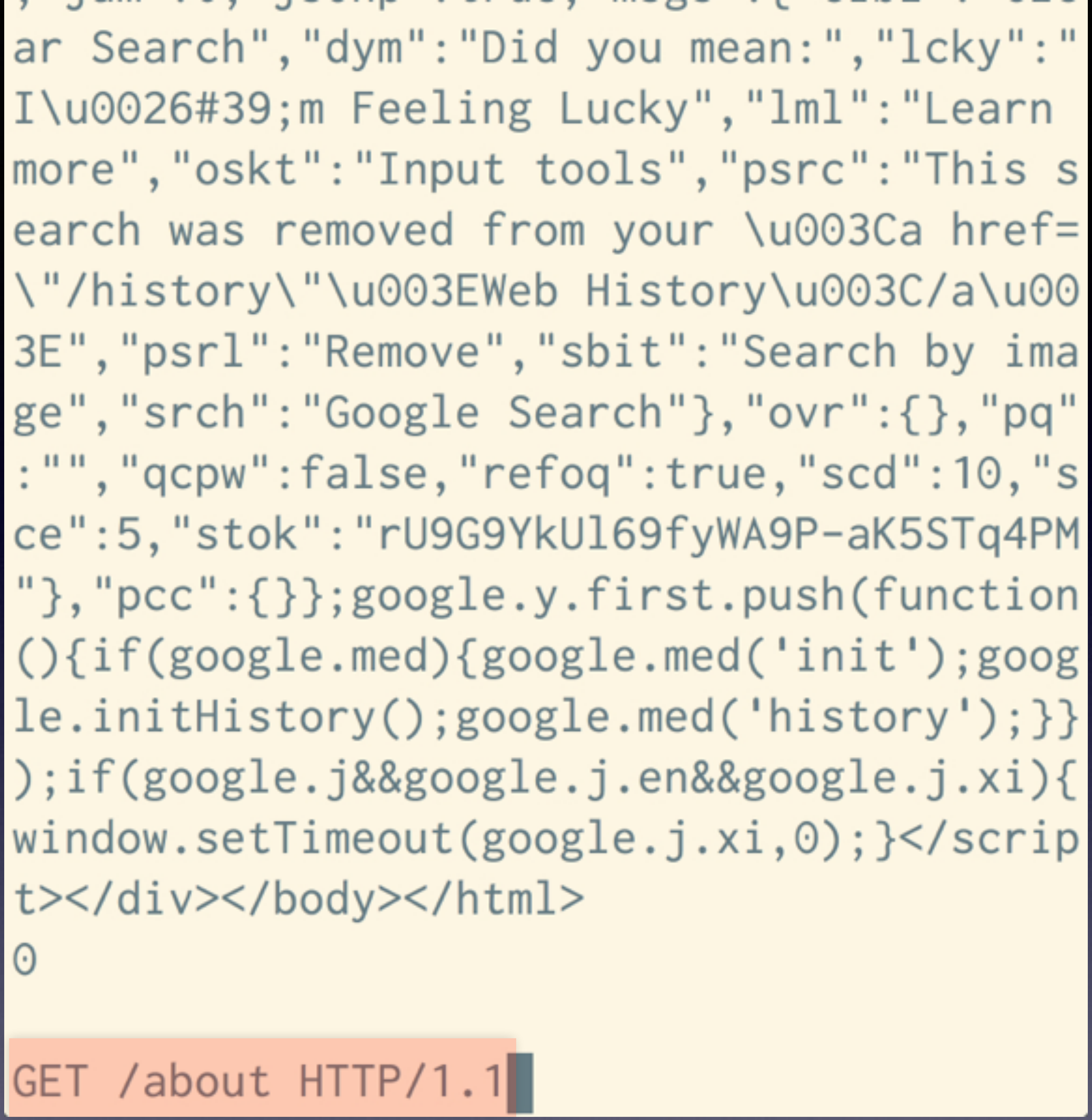
–HTTP/1.1 Spec

Let’s get some HTML

HTTP/1.1



```
➜  ~  telnet google.com 80
Trying 74.125.226.34...
Connected to google.com.
Escape character is '^]'.
GET / HTTP/1.1
```



```
ar Search","dym":"Did you mean:","lcky":"
I\u0026#39;m Feeling Lucky","lml":"Learn
more","oskt":"Input tools","psrc":"This s
earch was removed from your \u003Ca href=
\"/history\"\u003EWeb History\u003C/a\u00
3E","psrl":"Remove","sbit":"Search by ima
ge","srch":"Google Search"},"ovr":{},"pq"
:"","qcpw":false,"refoq":true,"scd":10,"s
ce":5,"stok":"rU9G9YkUl69fyWA9P-aK5STq4PM
"},"pcc":{}};google.y.first.push(function
(){if(google.med){google.med('init');goog
le.initHistory();google.med('history');}}
);if(google.j&&google.j.en&&google.j.xi){
window.setTimeout(google.j.xi,0);}</scrip
t></div></body></html>
0

GET /about HTTP/1.1
```

Persistent connection!

HTTP/1.1 Spec

Pipelining

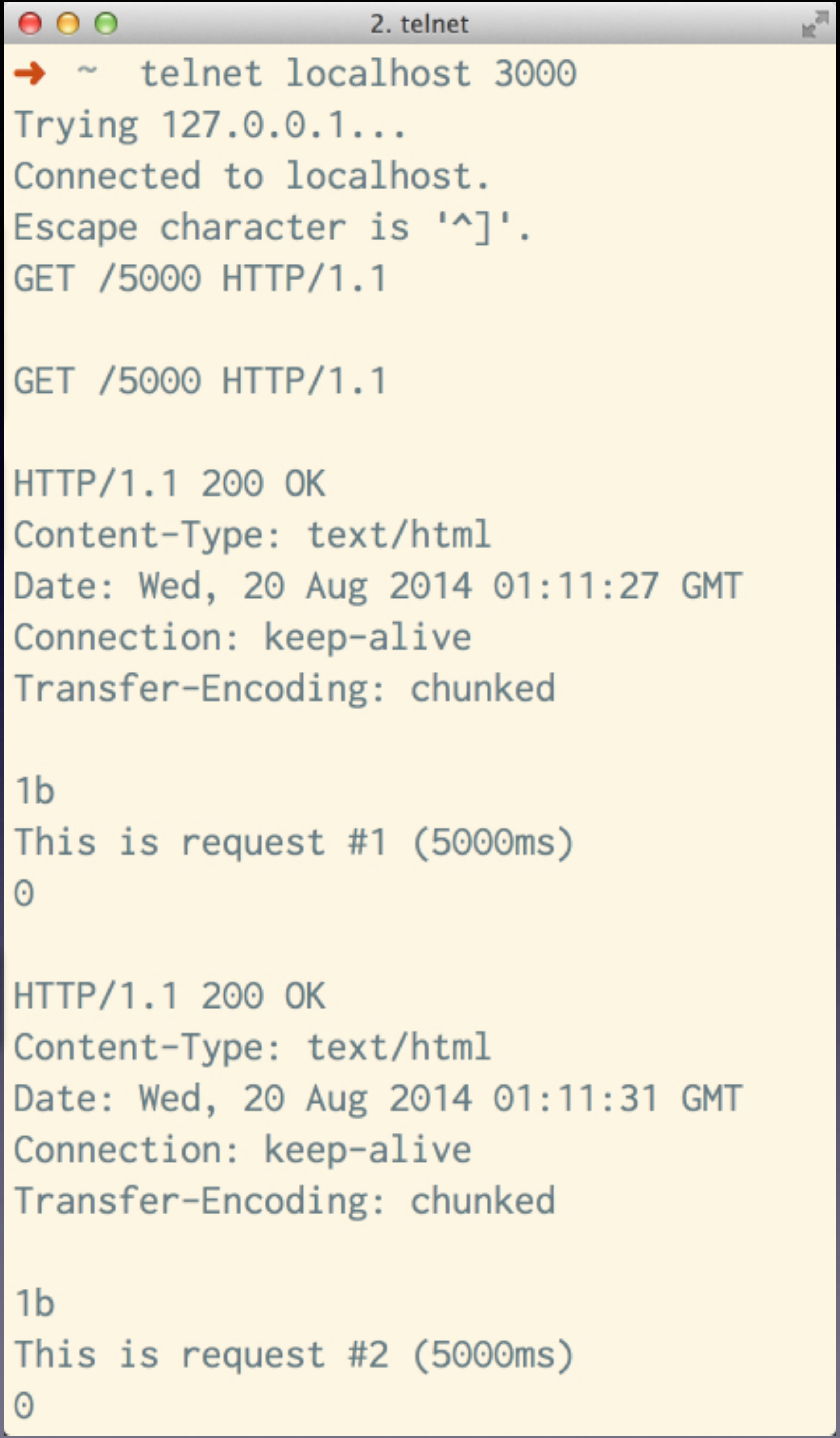
“A client … MAY … send multiple requests without waiting for each response.”

“A server MUST [respond] in the same order that the requests were received.”

–HTTP/1.1 Spec

Let’s get some (local) HTML

HTTP/1.1 Pipelining



Request #1

Request #2

Response #1

Response #2

```
➜  ~  telnet localhost 3000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET /5000 HTTP/1.1

GET /5000 HTTP/1.1

HTTP/1.1 200 OK
Content-Type: text/html
Date: Wed, 20 Aug 2014 01:11:27 GMT
Connection: keep-alive
Transfer-Encoding: chunked

1b
This is request #1 (5000ms)
0

HTTP/1.1 200 OK
Content-Type: text/html
Date: Wed, 20 Aug 2014 01:11:31 GMT
Connection: keep-alive
Transfer-Encoding: chunked

1b
This is request #2 (5000ms)
0
```

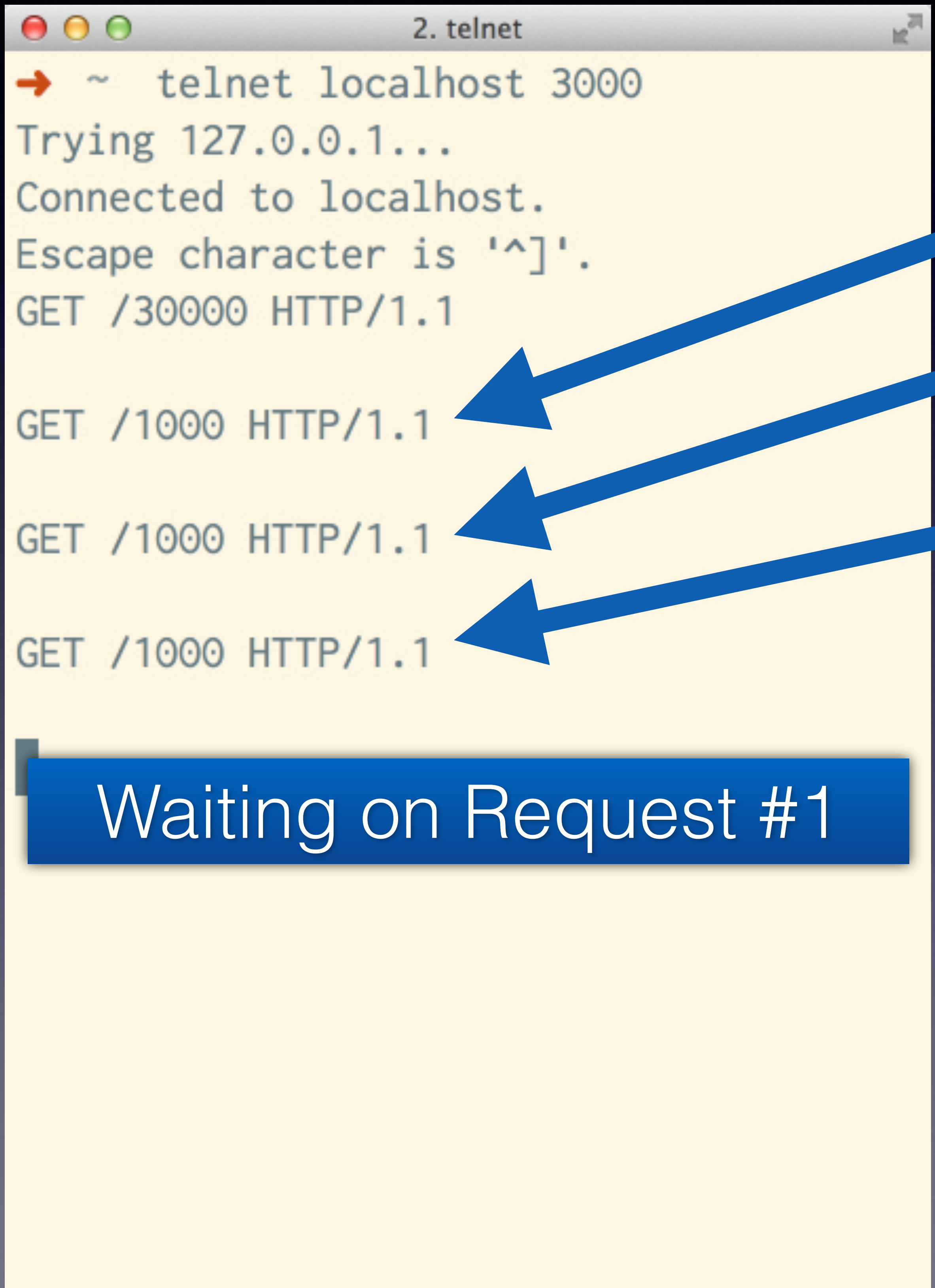
“I’ll let the browser pipeline all my assets.”

–Web Developer Guy

Let’s get some (blocked) HTML

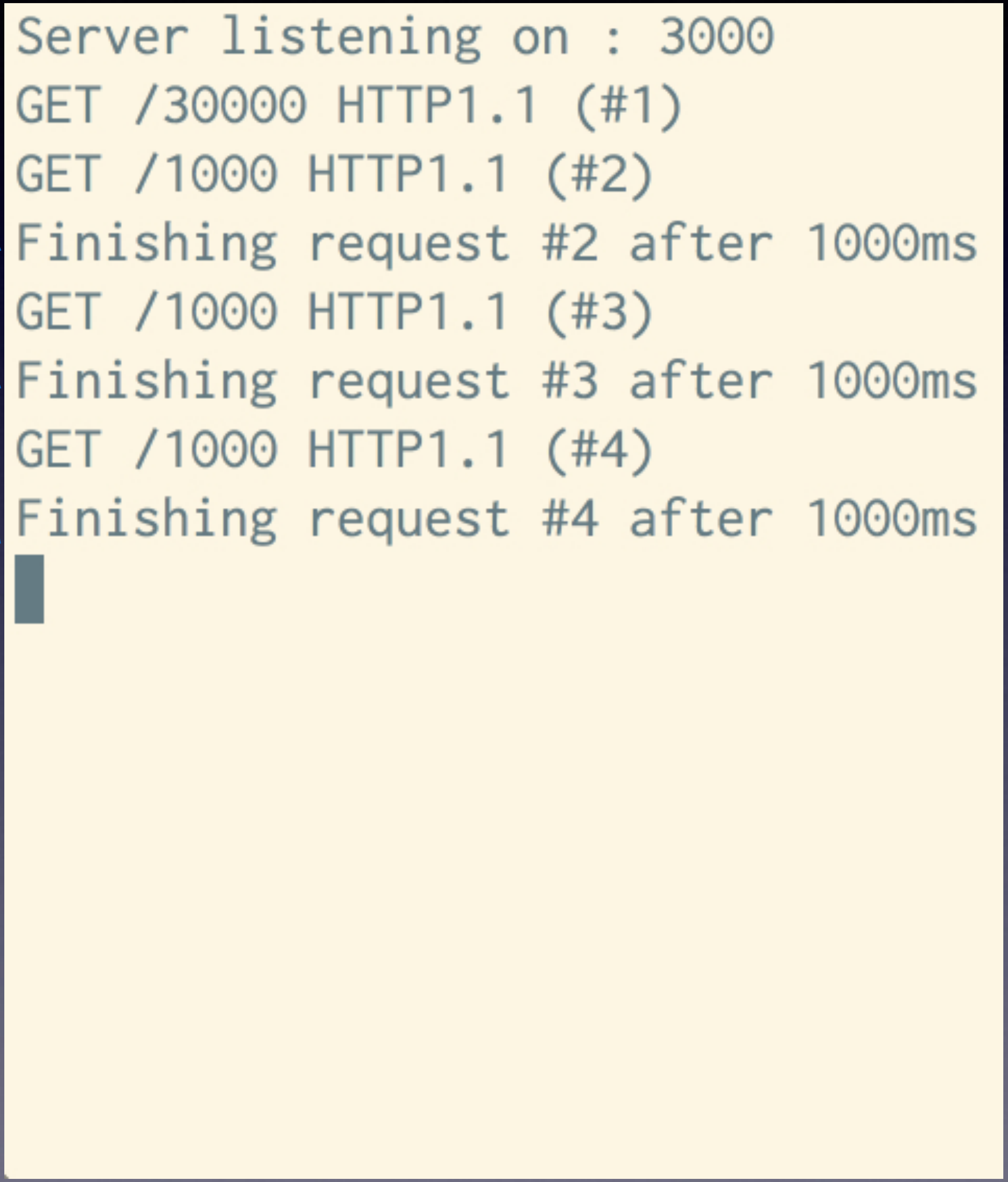
HTTP/1.1 Head-of-Line Blocking

Head-of-Line Blocking



```
➜  ~  telnet localhost 3000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET /30000 HTTP/1.1

GET /1000 HTTP/1.1

GET /1000 HTTP/1.1

GET /1000 HTTP/1.1

```

Waiting on Request #1



```
Server listening on : 3000
GET /30000 HTTP1.1 (#1)
GET /1000 HTTP1.1 (#2)
Finishing request #2 after 1000ms
GET /1000 HTTP1.1 (#3)
Finishing request #3 after 1000ms
GET /1000 HTTP1.1 (#4)
Finishing request #4 after 1000ms
```

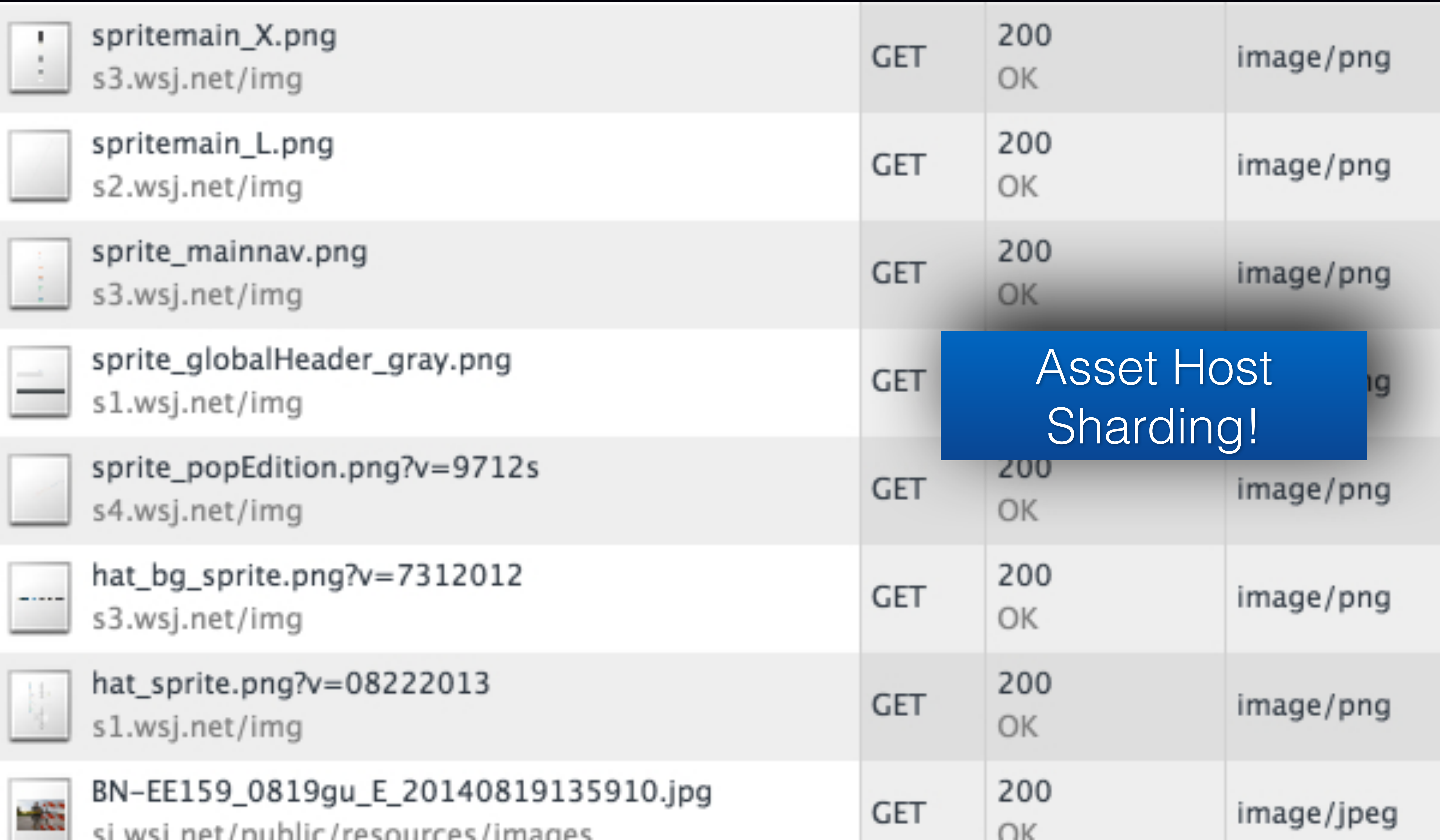
HTTP/1.1 Spec

Pipelining

"A server MUST [respond] in the same order that the requests were received."

–HTTP/1.1 Spec

How we (a)buse HTTP



spritemain_X.png s3.wsj.net/img	GET	200 OK	image/png
spritemain_L.png s2.wsj.net/img	GET	200 OK	image/png
sprite_mainnav.png s3.wsj.net/img	GET	200 OK	image/png
sprite_globalHeader_gray.png s1.wsj.net/img	GET	[illegible]	[illegible]g
sprite_popEdition.png?v=9712s s4.wsj.net/img	GET	200 OK	image/png
hat_bg_sprite.png?v=7312012 s3.wsj.net/img	GET	200 OK	image/png
hat_sprite.png?v=08222013 s1.wsj.net/img	GET	200 OK	image/png
BN-EE159_0819gu_E_20140819135910.jpg si.wsj.net/public/resources/images	GET	200 OK	image/jpeg

Asset Host Sharding!

Let’s Serve some HTML

HTTP/1.1

HTTP/1.1 Spec

Response

```
Response      = Status-Line               ; Section 6.1
                *(( general-header        ; Section 4.5
                 | response-header        ; Section 6.2
                 | entity-header ) CRLF)  ; Section 7.1
                                CRLF
                [ message-body ]          ; Section 7.2
```

HTTP/1.1 Spec

Status-Line

Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF

HTTP/1.1 Spec

Status-Line

Status-Line = HTTP/1.1 201 Created

HTTP/1.1 Spec

Message Headers

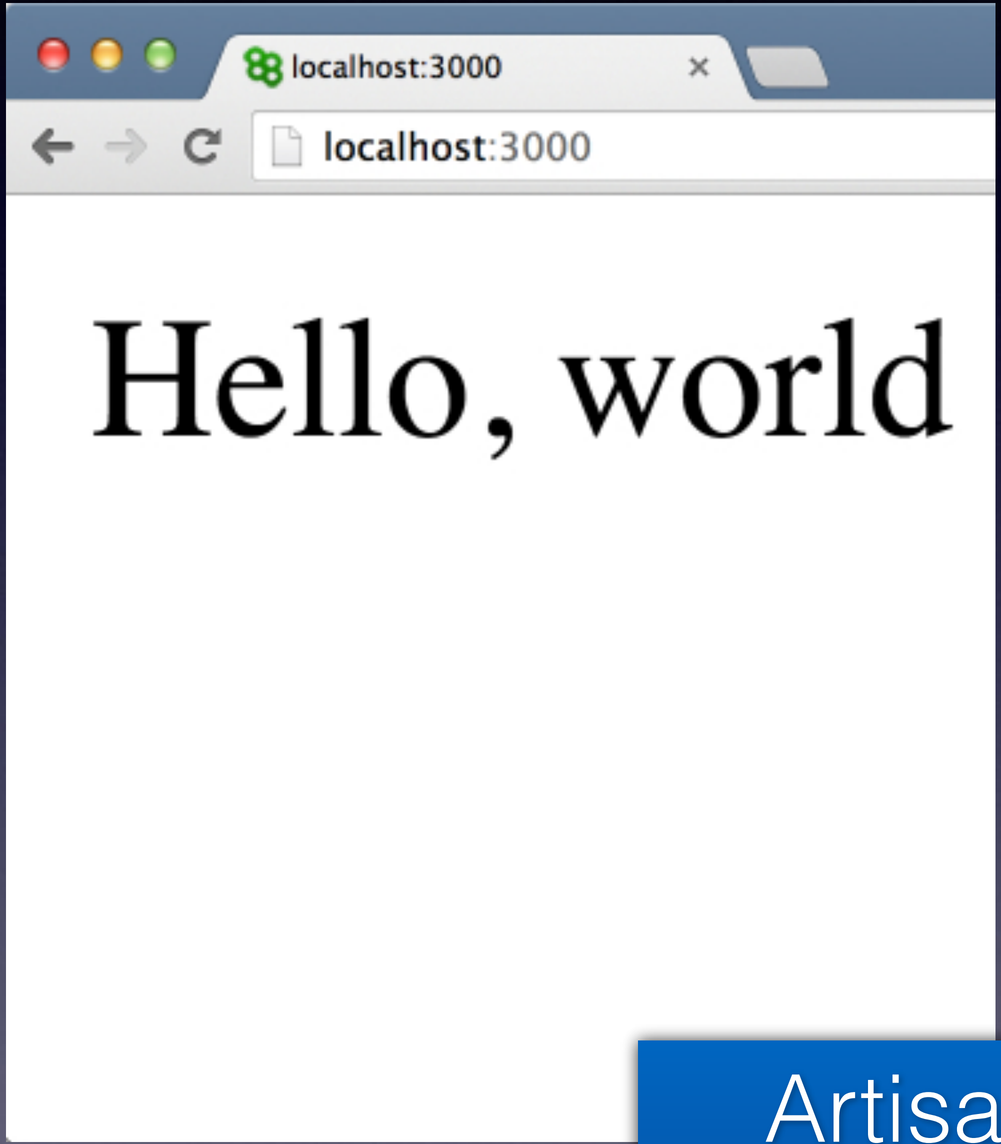
The presence of a message-body … is signaled by the inclusion of a Content-Length or Transfer-Encoding header field

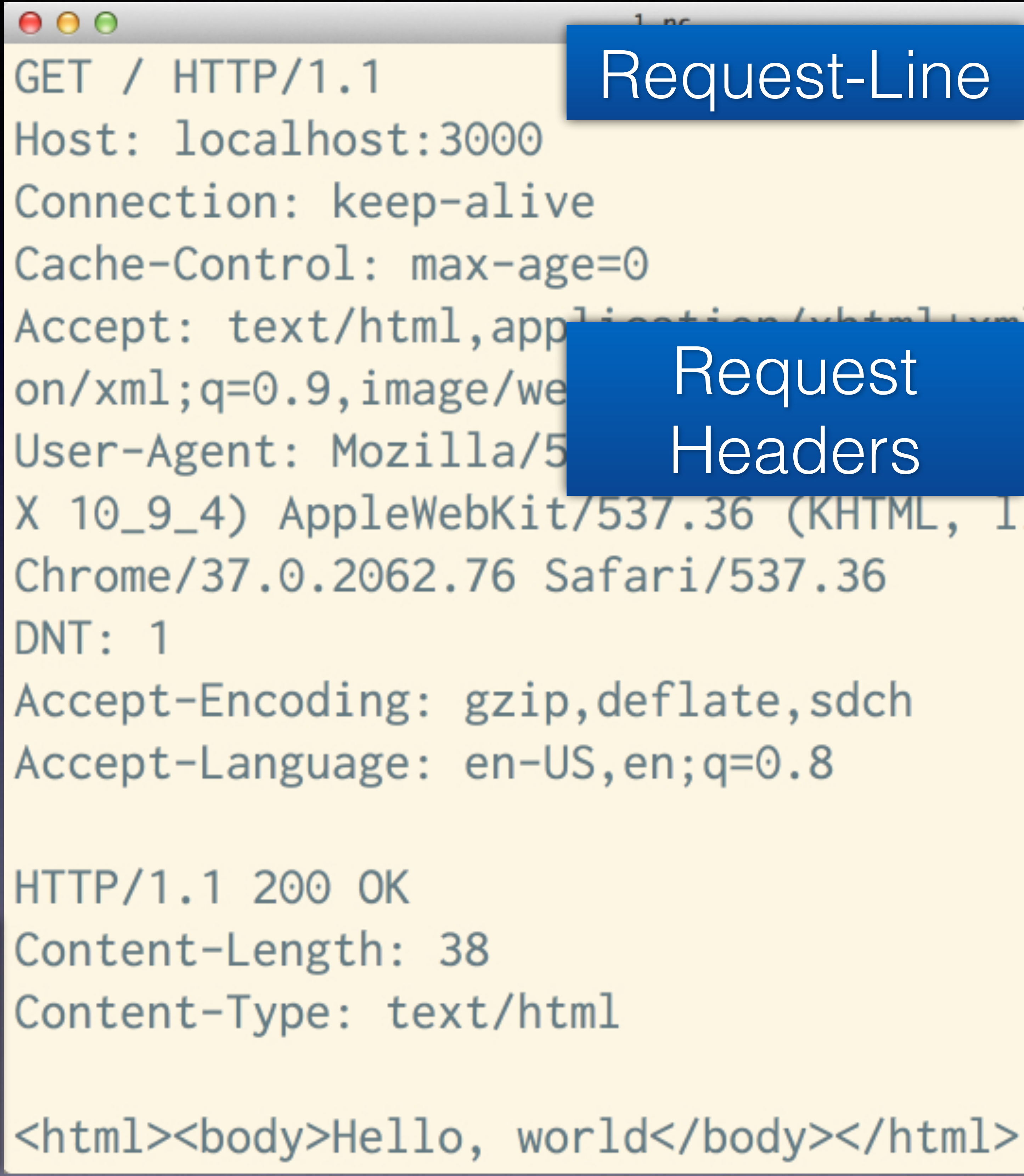
–HTTP/1.1 Spec

HTTP/1.1 Spec

	Example HTTP Response
Status-Line	HTTP/1.1 200 OK
Headers	Content-Type: text/html Content-Length: 38
message-body	<html> <body>Hello, world</body> </html>

$ nc -l 3000





```
GET / HTTP/1.1
Host: localhost:3000
Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,app
on/xml;q=0.9,image/we
User-Agent: Mozilla/5
X 10_9_4) AppleWebKit/537.36 (KHTML, l
Chrome/37.0.2062.76 Safari/537.36
DNT: 1
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8

HTTP/1.1 200 OK
Content-Length: 38
Content-Type: text/html

<html><body>Hello, world</body></html>
```

Request-Line

Request Headers

Artisanal, Small-batch HTTP

Let’s Serve (dynamic-length) HTML

HTTP/1.1

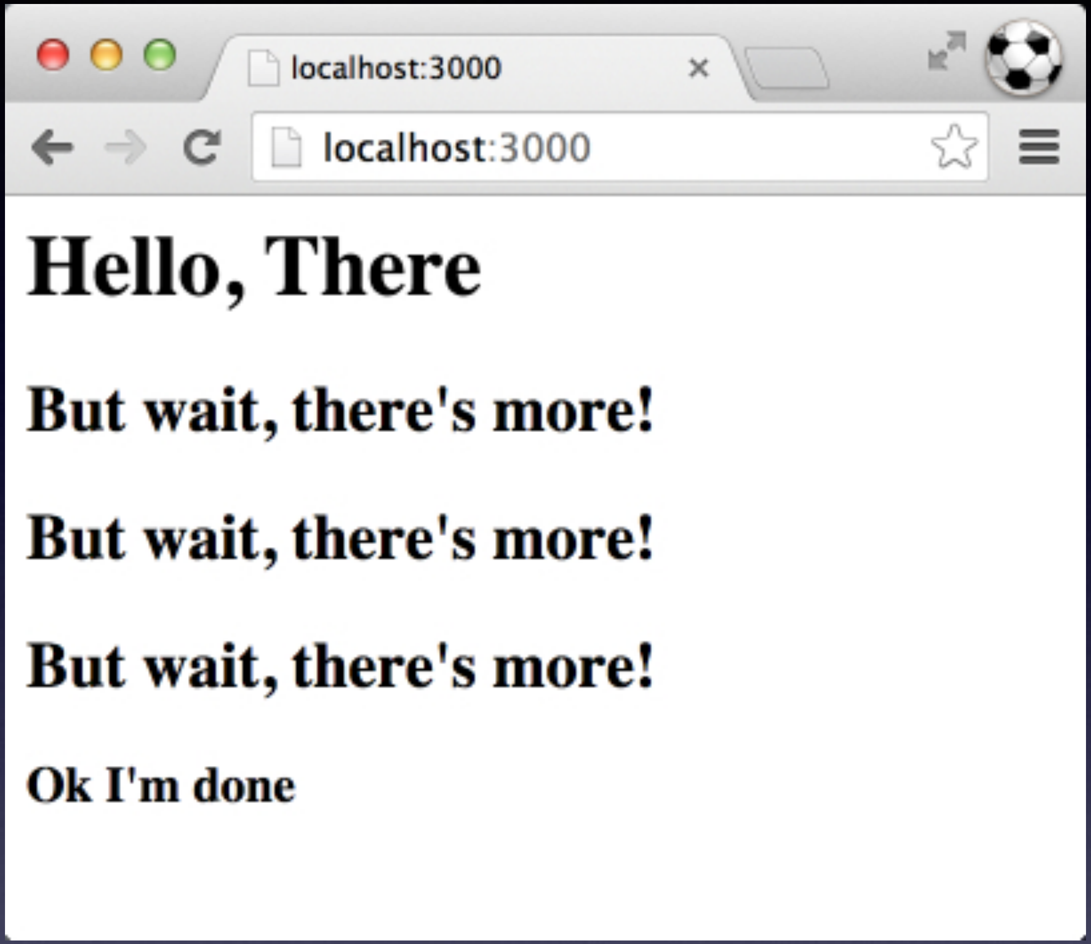
Transfer-Encoding: chunked

HTTP/1.1

Transfer-Encoding: chunked

<chunk-length>
chunk
<chunk-length>
chunk
0

$ nc -l 3000



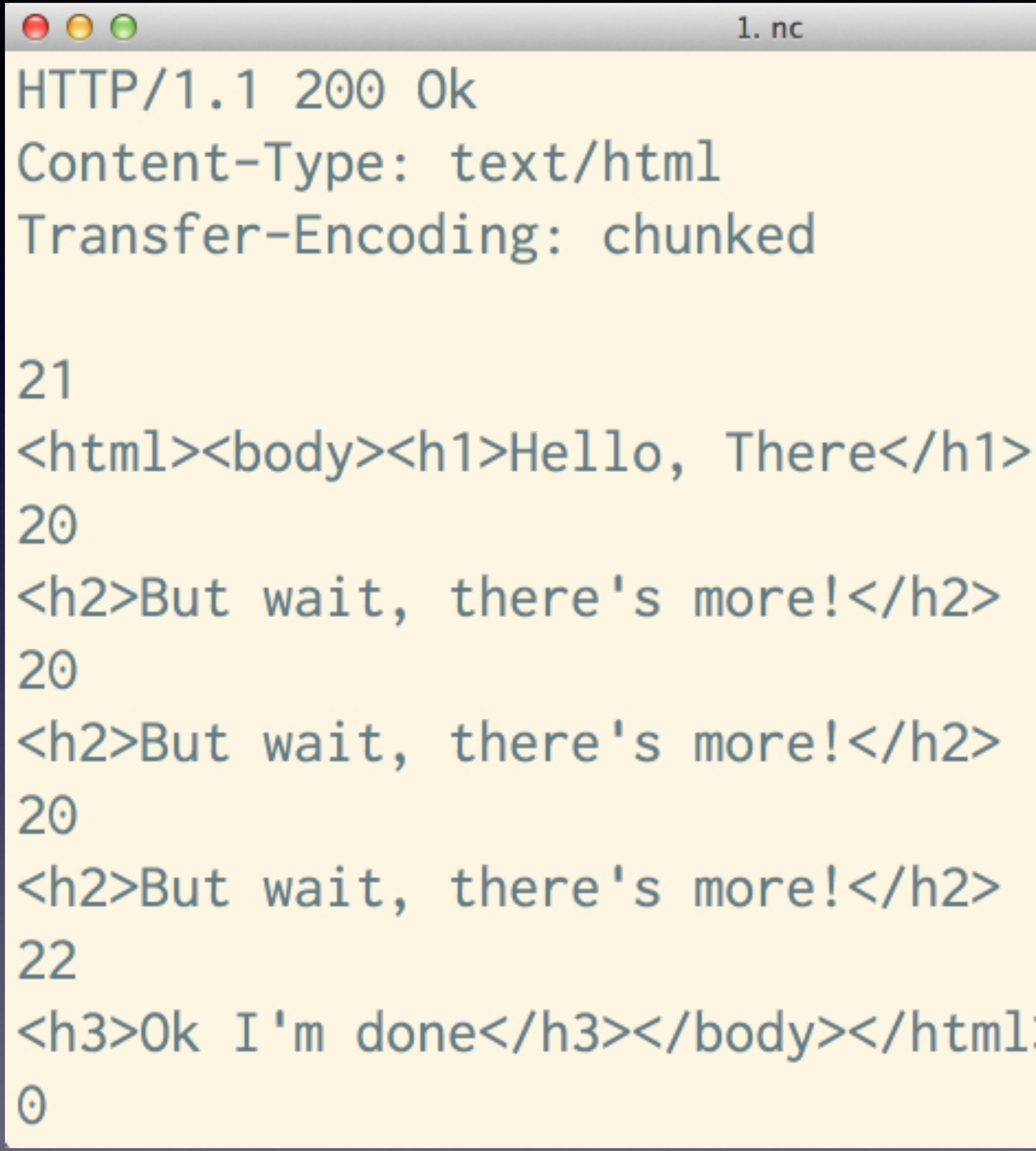
Hello, There

But wait, there's more!

But wait, there's more!

But wait, there's more!

Ok I'm done



```
HTTP/1.1 200 Ok
Content-Type: text/html
Transfer-Encoding: chunked

21
<html><body><h1>Hello, There</h1>
20
<h2>But wait, there's more!</h2>
20
<h2>But wait, there's more!</h2>
20
<h2>But wait, there's more!</h2>
22
<h3>Ok I'm done</h3></body></html
0
```

HTTP/2

What is HTTP/2 not?

- Same HTTP methods (GET, PUT, etc)
- Same usage of headers
- Same use cases
- Still one client, one server

What is HTTP/2 is?

- One TCP connection
- Binary! (Different transfer mechanism)
- Header compression
- Upgrade path

- One TCP connection
    - Requests and Responses can cross
    - Server push
    - Prioritization

- One TCP connection: implications
  - Asset Host Sharding: bad!
  - CSS/JS Concatenation: Unnecessary/bad!
  - Image spriting: Unnecessary/bad!

- Binary
  - HTTP/2: same semantics, different “on-the-wire” transport
  - Can we still make small-batch HTTP/2? (Sorta?)
  - More compact, easier to parse
  - Mandatory compression

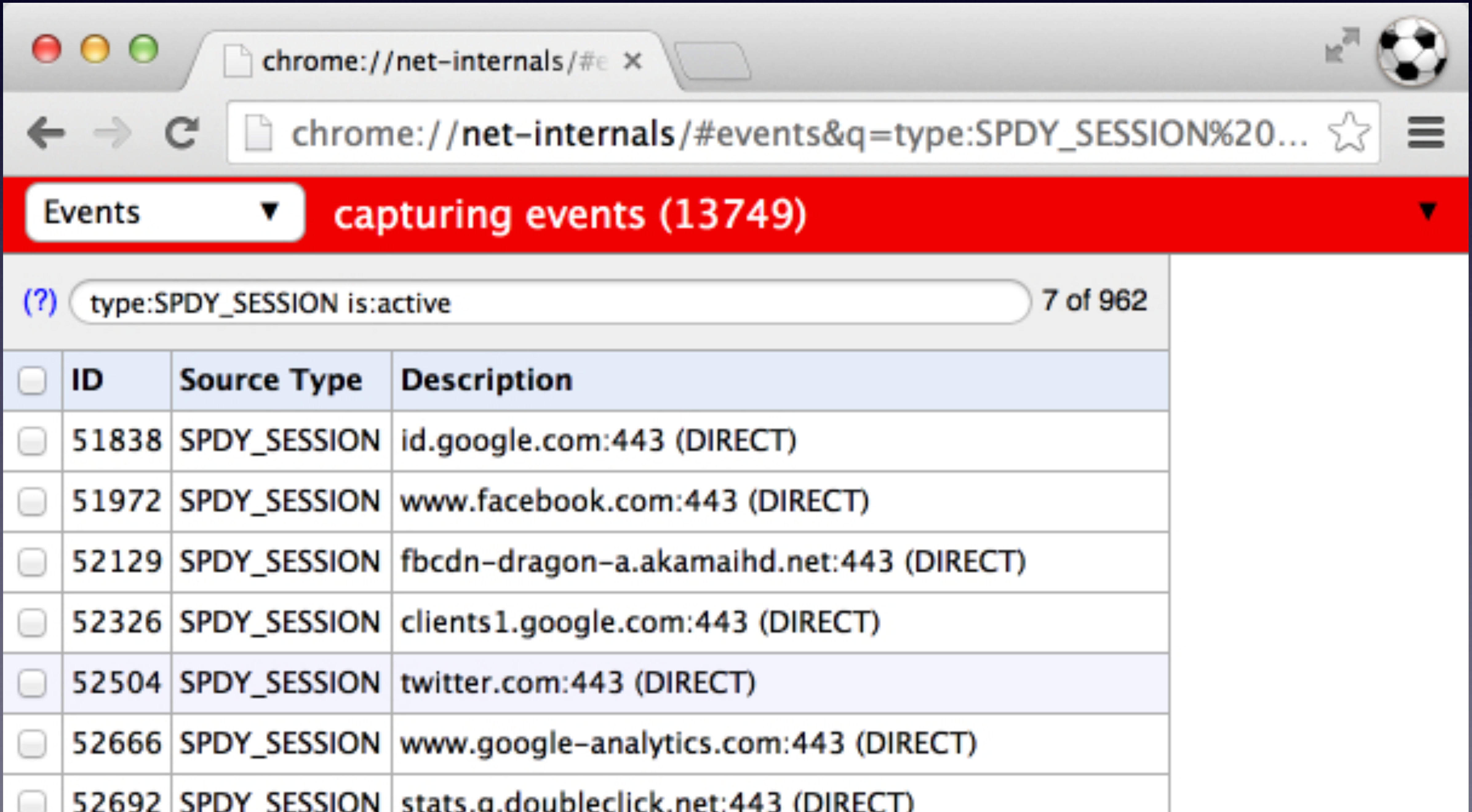
Where is HTTP/2?

SPDY

- ~ 1% of all servers (2013)
- Google
- Facebook
- Twitter
- CloudFlare

SPDY: In your browser

chrome://net-internals/#events


chrome://net-internals/#e
chrome://net-internals/#events&q=type:SPDY_SESSION%20...
Events
capturing events (13749)
(?)

type:SPDY_SESSION is:active

7 of 962

ID	Source Type	Description
51838	SPDY_SESSION	id.google.com:443 (DIRECT)
51972	SPDY_SESSION	www.facebook.com:443 (DIRECT)
52129	SPDY_SESSION	fbcdn-dragon-a.akamaihd.net:443 (DIRECT)
52326	SPDY_SESSION	clients1.google.com:443 (DIRECT)
52504	SPDY_SESSION	twitter.com:443 (DIRECT)
52666	SPDY_SESSION	www.google-analytics.com:443 (DIRECT)
52692	SPDY_SESSION	stats.g.doubleclick.net:443 (DIRECT)

HTTP by **Hand**

Thank you!

@bantic
Cory Forsyth