

## Test results

- Browser

The screenshot shows a browser window for localhost:8080. The page displays the Santa Clara University logo and the tagline "THE JESUIT UNIVERSITY IN SILICON VALLEY". A large banner features the text "innovating with a MISSION" over a background image of a modern university building. Below the banner, the text "The Campaign for Santa Clara University" is visible. A "Learn More" button is at the bottom. The browser's developer tools Network tab is open, showing a timeline with several requests and responses. The Headers section shows a 200 OK status code. The Response section contains the HTML content of the page.

```
Request URL: http://localhost:8080/
Request Method: GET
Status Code: 200 OK
Remote Address: [::1]:8080
Referrer Policy: no-referrer-when-downgrade

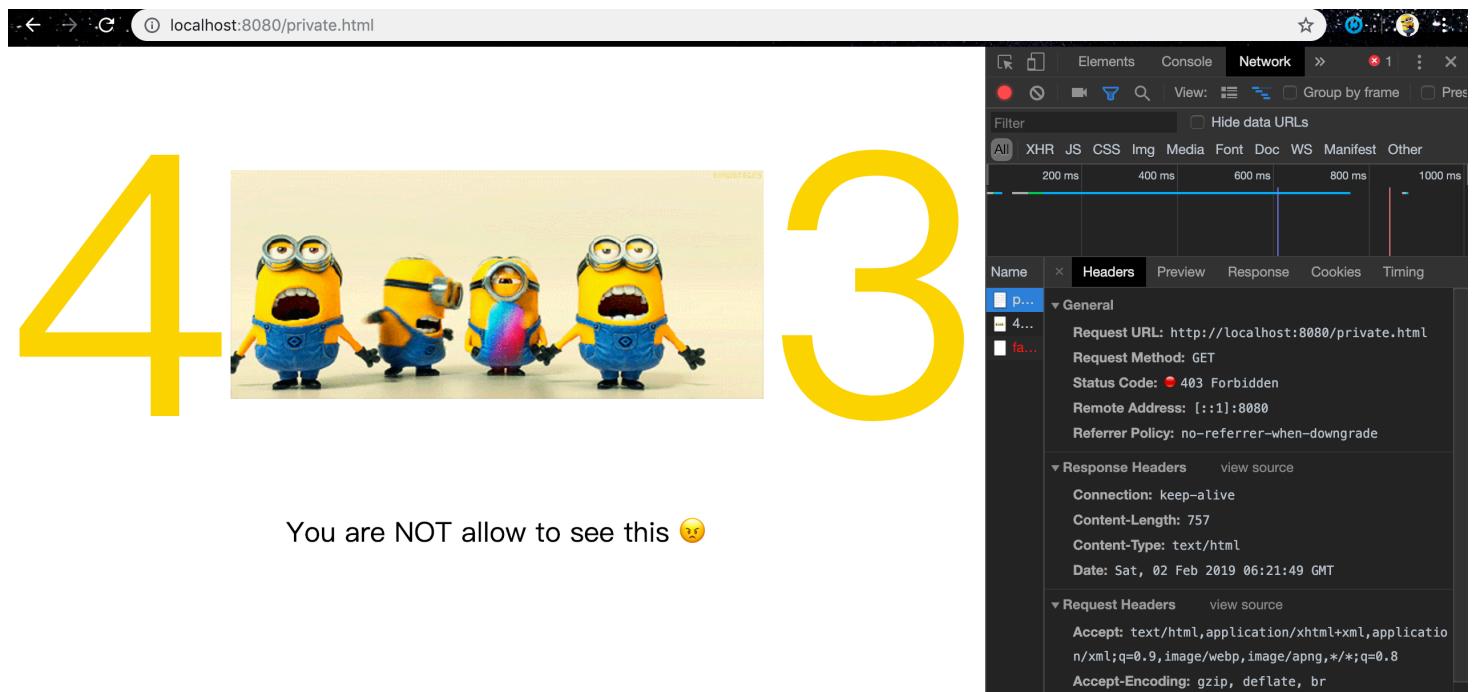
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,zh-TW;q=0.8,zh;q=0.6
7,zh-CN;q=0.6
AlexaToolbar-ALX_NS_PH: AlexaToolbar/alx-4.0.3
Cache-Control: no-cache
Connection: keep-alive
```

The screenshot shows a browser window for localhost:8080/asdadsasd. The page features a large yellow Minion character from the Despicable Me franchise, standing between two large yellow "4" characters. Below the character, the text "Oops...we cannot find your page :(" is displayed. The browser's developer tools Network tab is open, showing a single request with a 404 Not Found status code. The Headers section shows a 404 status. The Response section contains the HTML content of the 404 error page.

```
Request URL: http://localhost:8080/asdadsasd
Request Method: GET
Status Code: 404 Not Found
Remote Address: [::1]:8080
Referrer Policy: no-referrer-when-downgrade

Connection: keep-alive
Content-Length: 807
Content-Type: text/html
Date: Sat, 02 Feb 2019 06:22:01 GMT

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate, br
```



## - Testing Script

```
x Daniel src master npm run test
> static-file-server@1.0.0 test /Users/daniel...
> mocha

  Status 200
    ✓ should serve / as index.html
    ✓ should serve jpg
    ✓ should serve gif

  Status 404
    ✓ should handle non-existing path

  Status 403
    ✓ should reject accessing files with no p

  5 passing (81ms)
```

a simple tool like `telnet` will simply show  
request it.

When you type a URL into a web browser, the browser sends a request for the requested file. If the file is of type *text/html*, the browser will make one connection to the web server. If the file is of type *image/jpeg*, the browser will make four connections to the web server for embedded links (such as images) and one connection to the web server to retrieve the embedded files. Note that the previous statement refers to the HTTP/1.0 protocol, which is what you will be using.

**Extra Credit:** Using HTTP/1.0, a separate file. This implies that the TCP connection is closed. HTTP/1.1 attempts to address this limitation by allowing the server to keep connections to clients open for a longer period of time. It also supports pipelining of client requests. That is, after a client has sent multiple requests (e.g., index.html), the server sends back multiple responses without closing the connection between them.

- Telnet

```
Daniel > src > ↗ master • > telnet localhost 8080
Trying ::1...
Connected to localhost.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.1 200 OK
Last-Modified: Sat, 26 Jan 2019 20:18:36 GMT
Content-Length: 39423
Content-Type: text/html
Date: Sat, 02 Feb 2019 06:14:48 GMT
Connection: close
```

```
x > Daniel > src > ↗ master • > telnet localhost 8080
Trying ::1...
Connected to localhost.
Escape character is '^]'.
GET /private.html HTTP/1.0

HTTP/1.1 403 Forbidden
Content-Length: 757
Content-Type: text/html
Date: Sat, 02 Feb 2019 06:16:37 GMT
Connection: close

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>403 Forbidden</title>
```

requested file. If the file is HTML) and the HTTP/1.0 for embedded links (such web server to retrieve the five separate connections the four image files. Note protocol, which is what yo

**Extra Credit:** Using HT file. This implies that the T phase. HTTP/1.1 attempts server keeps connections pipelining of client reques returned (e.g., index.html) some period of time, a subsequent requests. One

```
x Daniel > src > master telnet localhost 8080
```

```
Trying ::1...
Connected to localhost.
Escape character is '^]'.
GET /afhawfihaiwfhailwf HTTP/1.0
```

```
HTTP/1.1 404 Not Found
Content-Length: 807
Content-Type: text/html
Date: Sat, 02 Feb 2019 06:17:20 GMT
Connection: close
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>404 Not Found</title>
```

When you type a URL into your browser, it sends a request to the web server for the requested file. If the file is an HTML page, the server returns the HTML content. For embedded links (such as images), the server keeps connections open to retrieve the five separate connections required to download the four image files. Note that this is a limitation of the HTTP/1.0 protocol, which is what you are using here.

**Extra Credit:** Using HTTP/1.0, a separate connection is used for each request. This implies that the TCP connections being used never close. In the phase, HTTP/1.1 attempts to address this limitation. When using HTTP/1.1, the server keeps connections to clients open, allowing for "persistent connections". That is, after the results of a single request have been returned (e.g., index.html), the server should by default leave the connection open for some period of time, allowing the client to reuse that connection for subsequent requests. One key issue here is determining how long to keep a connection open. This timeout needs to be configured in the server software. It should be dynamic based on the number of other active connections currently supporting. Thus if the server is idle, it can afford to keep the connection open for a relatively long period of time. If the server is busy, it may need to close the connection sooner (consuming kernel/thread resources) for very long. You should refer to the documentation for your server to determine this timeout.

```
x Daniel > src > master telnet localhost 8080
```

```
Trying ::1...
Connected to localhost.
Escape character is '^]'.
GET / HTTP/1.1
```

```
HTTP/1.1 200 OK
Last-Modified: Sat, 26 Jan 2019 20:18:36 GMT
Content-Length: 39423
Content-Type: text/html
Date: Sat, 02 Feb 2019 06:17:43 GMT
Connection: keep-alive
```

```
<!DOCTYPE html>
<!-- saved from url=(0030)https://www.scu.edu/index.html -->
<html lang="en" class="gr__scu_edu"><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

**Extra Credit:** Using HTTP/1.0, a separate connection is used for each request. This implies that the TCP connections being used never close. In the phase, HTTP/1.1 attempts to address this limitation. When using HTTP/1.1, the server keeps connections to clients open, allowing for "persistent connections". That is, after the results of a single request have been returned (e.g., index.html), the server should by default leave the connection open for some period of time, allowing the client to reuse that connection for subsequent requests. One key issue here is determining how long to keep a connection open. This timeout needs to be configured in the server software. It should be dynamic based on the number of other active connections currently supporting. Thus if the server is idle, it can afford to keep the connection open for a relatively long period of time. If the server is busy, it may need to close the connection sooner (consuming kernel/thread resources) for very long. You should refer to the documentation for your server to determine this timeout.

```
x Daniel src ↵ master telnet localhost 8080
Trying ::1...
Connected to localhost.
Escape character is '^]'.
GET /private.html HTTP/1.1

HTTP/1.1 403 Forbidden
Content-Length: 757
Content-Type: text/html
Date: Sat, 02 Feb 2019 06:18:39 GMT
Connection: keep-alive

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>403 Forbidden</title>
```

requested file. If the file is of type HTML) and the HTTP/1.0 protocol for embedded links (such as images), the web server to retrieve the embedded files. Note that the five separate connections will be released by the four image files. Note that the protocol, which is what you will be using.

**Extra Credit:** Using HTTP/1.0, a client can keep multiple connections open to the same file. This implies that the TCP connection is not released until the entire file has been returned. HTTP/1.1 attempts to address this by allowing the server to keep connections to clients open for a longer period of time, allowing them to send multiple requests. One key issue is how long the connection should be kept open.

```
x Daniel src ↵ master telnet localhost 8080
Trying ::1...
Connected to localhost.
Escape character is '^]'.
GET /akfwguawf HTTP/1.1

HTTP/1.1 404 Not Found
Content-Length: 807
Content-Type: text/html
Date: Sat, 02 Feb 2019 06:18:14 GMT
Connection: keep-alive

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>404 Not Found</title>
```

for embedded links (such as images). Note that the five separate connections will be released by the four image files. Note that the protocol, which is what you will be using.

**Extra Credit:** Using HTTP/1.1, a client can keep multiple connections open to the same file. This implies that the TCP connection is not released until the entire file has been returned. HTTP/1.1 attempts to address this by allowing the server to keep connections to clients open for a longer period of time, allowing them to send multiple requests. One key issue is how long the connection should be kept open. This time should be dynamic based on the request.