

Problem 1

the screenshots listed below show the output of a GET request:


The screenshot shows a web browser at `localhost:8080/index.html`. The page content is "Welcome to home page". The Network tab is open, showing a single request to `http://localhost:8080/index.html`. The request method is GET, status is 200 OK, and the remote address is `:::11::8080`. The response headers indicate a content length of 119, content type of `text/html`, and a date of `Sun Mar 03 16:09:31 CET 2019`. The request headers show various browser capabilities and a user agent string.

Name	Status	Type	Initiator	Size	Time	Waterfall
index.html	200	document	Other	329 B	3 ms	

The screenshot shows a web browser at `localhost:8080/user2`. The page content is "This is a heading for user 2" followed by "This is a paragraph.". The Network tab is open, showing a single request to `http://localhost:8080/user2`. The request method is GET, status is 200 OK, and the remote address is `:::11::8080`. The response headers indicate a content length of 119, content type of `text/html`, and a date of `Sun Mar 03 16:09:31 CET 2019`. The request headers show various browser capabilities and a user agent string.

Name	Status	Type	Initiator	Size	Time	Waterfall
user2	200	document	Other	329 B	3 ms	
script.js	200	script	content.js:75	1.5 KB	5 ms	

← → ↻ localhost:8080/google.png



Elements Console Sources Performance Network

Filter: Hide data URLs

XHR JS CSS Img Media Font Doc WS Manifest Other

20 ms 40 ms 60 ms 80 ms 100 ms

Name: google.png

General

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

Response Headers

- Content-Length: 13504
- Content-Type: image/png

Request Headers

- 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
- Cache-Control: max-age=0
- Connection: keep-alive
- Host: localhost:8080

1 request | 1.2 KB transferred

← → ↻ localhost:8080/index.html

welcome to index.html page

Elements Console Sources Performance Memory Application Security Audits AdBlock

Filter: Hide data URLs

XHR JS CSS Img Media Font Doc WS Manifest Other

10 ms 20 ms 30 ms 40 ms 50 ms 60 ms 70 ms 80 ms 90 ms 100 ms 110 ms 120 ms

Name: index.html

General

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

Response Headers

- Content-Length: 123
- Content-Type: text/html

Request Headers

- 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,ar;q=0.8,sv;q=0.7
- Cache-Control: no-cache
- Connection: keep-alive
- Cookie: Idea-4e042964-ed364f23-4d14-4941-8a11-bb0de202afd
- Host: localhost:8080
- Pragma: no-cache
- Upgrade-Insecure-Requests: 1
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3653.76

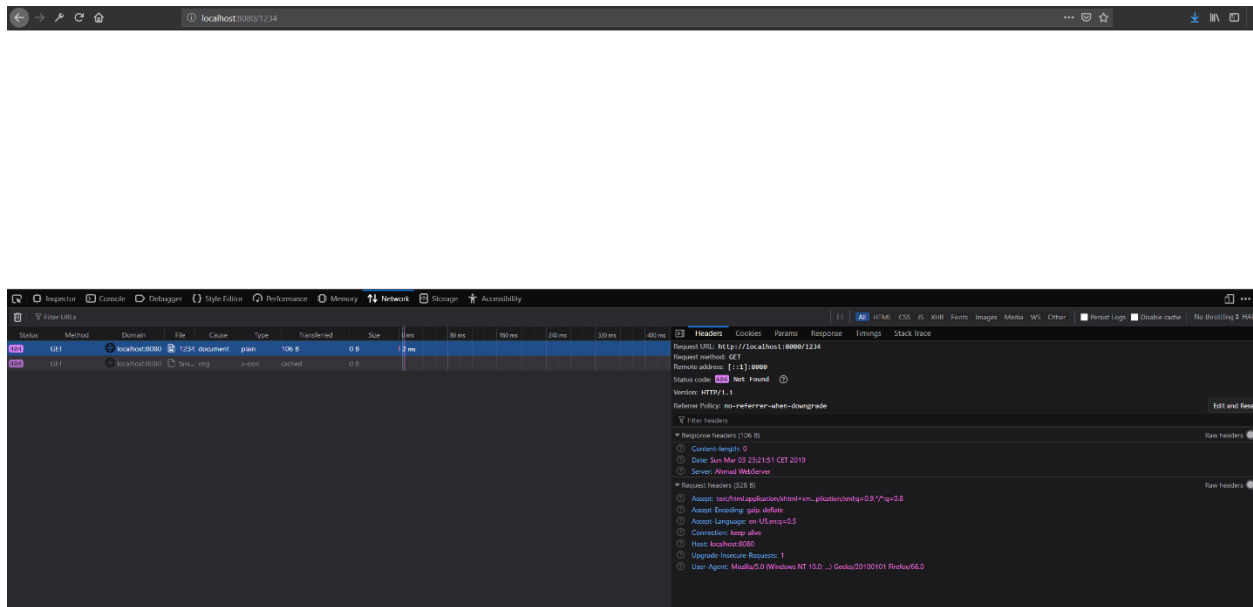
2 requests | 1.6 KB transferred | Finish: 110 ms

Console

Problem 2

404 Not Found

The picture below shows the result when a client tries to access a file that is not available in the directory.



403 Forbidden

the screenshot below shows the results of trying to access the admin file with a response of 403. The admin file directory was restricted if the client enters localhost:8080/admin or localhost:8080/admin/ in the path.

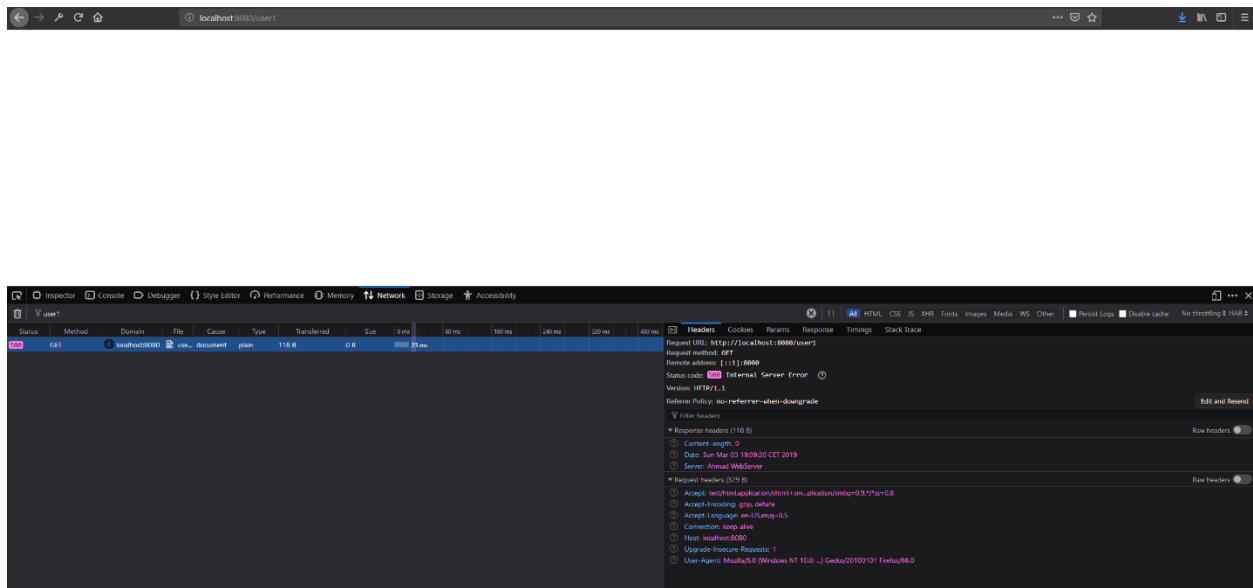
The screenshot shows a web browser window with the address bar set to `localhost:8080/admin`. The page content displays an error message: "Access to localhost was denied" and "You don't have authorization to view this page." Below this, it says "HTTP ERROR 403" and provides a "Reload" button.

The browser's developer tools are open, showing the Network tab. A request to `admin` is selected, and the details pane on the right shows the following information:

- General**
 - Request URL: `http://localhost:8080/admin`
 - Request Method: `GET`
 - Status Code: `403 FORBIDDEN`
 - Remote Address: `[::1]:8080`
 - Referrer Policy: `no-referrer-when-downgrade`
- Response Headers**
 - Content-length: `0`
 - Date: `Sun Mar 03 16:28:18 CET 2019`
 - Server: `Ahmad Sadiq WebServer`
- Request Headers**
 - 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;q=0.9,ar;q=0.8,sv;q=0.7`
 - Cache-Control: `no-cache`
 - Connection: `keep-alive`
 - Cookie: `Idaa-4a042964-ad364f23-4d14-49d1-Ba11-1bb8df202afd`
 - Host: `localhost:8080`
 - Pragma: `no-cache`
 - Upgrade-Insecure-Requests: `1`
 - User-Agent: `Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) 537.36`

500 Internal Server Error

The screenshot below shows the results of an Internal server error. To get a screenshot of an Internal Server Error from the browser we had to generate it by ourselves, because it is tough to catch an internal server error



VG-task 2

Put a text file 1.txt using curl

```
C:\Users\Ahmad\Desktop\curl-7.64.0-win64-mingw\bin>curl -T C:\1.txt localhost:8080
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100    14      0     0  100    14      0     1  0:00:14  0:00:09  0:00:05     0
```

PUT a png image 4.png using curl

```
curl: try 'curl --help' or 'curl --manual' for more information

C:\Users\Ahmad\Desktop\curl-7.64.0-win64-mingw\bin>curl -T C:\4.png localhost:8080
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 209k    0     0  100 209k      0  196k  0:00:01  0:00:01 --:--:-- 196k

C:\Users\Ahmad\Desktop\curl-7.64.0-win64-mingw\bin>
```

Post vs Put

Put:

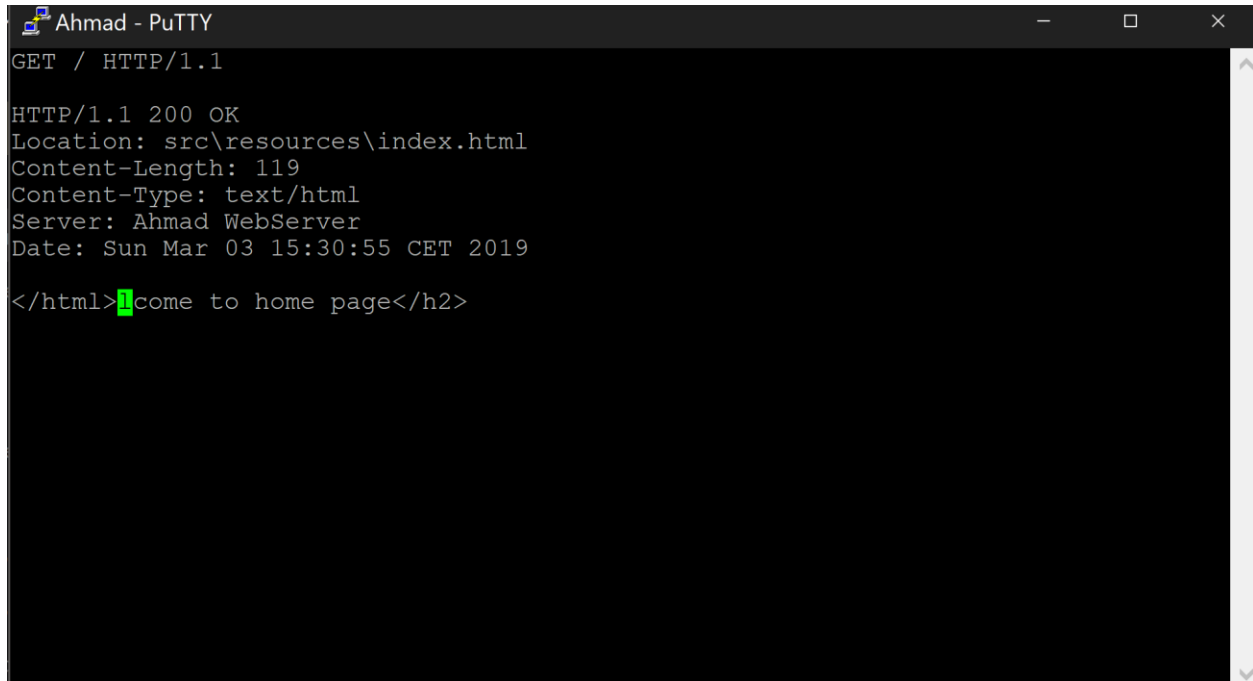
Used for replacing the existing resources or creating new incase if the resources does not exist. It is idempotent which means calling the request once or multiply times has the same effect the request is carried on without modifying the results beyond the initial application. It is repeatable and same, that means that for example if we have $x = 1$ if we repeat it many times the result will be the same $x = 1$.

Post:

Used for creating new resources. It is non-idempotent which means that calling the request once or multiply times may have additional effects the request is carried on and may modify the results beyond the initial application. It is non-repeatable and if requested multiply times an alert message is presented by the browser. It is not safe and non-repeatable due to the method producing several copies of the same resources.

Problem 3

200 OK



```
Ahmad - PuTTY
GET / HTTP/1.1

HTTP/1.1 200 OK
Location: src\resources\index.html
Content-Length: 119
Content-Type: text/html
Server: Ahmad WebServer
Date: Sun Mar 03 15:30:55 CET 2019

</html>come to home page</h2>
```

Using the `GET / HTTP/1.1` we get the index file in the root.

the headers response:

Version of the HTTP 1.1.

Response status: 200 OK.

Location: the path of the file requested by the client.

Content-Length: the size of the file in bytes.

Content-Type: the type of the file requested which is text/html which indicate the browser that the file of a text type.

Server: name of the server

Date: the time and date when the request was created.

The last row is the content of index.html file (the response body).


```
Ahmad - PuTTY
GET /user1 HTTP/1.1

HTTP/1.1 200 OK
Content-Length: 223
Content-Type: text/html

<!DOCTYPE html>
    <html>
        <head>
            <meta charset="utf-8">
        </head>
        <body>
            <form method="post"
" enctype="multipart/form-data">
                <input type="file" name="pic" accept="image/
*" >
                <input type="submit">
            </form>
        </body>
    </html>
```

Using the GET /user1 HTTP/1.1 we get the index file in the file user1.

the headers response:

Version of the HTTP 1.1.

Response status: 200 OK.

Location: the path of the file requested by the client.

Content-Length: the size of the file in bytes.

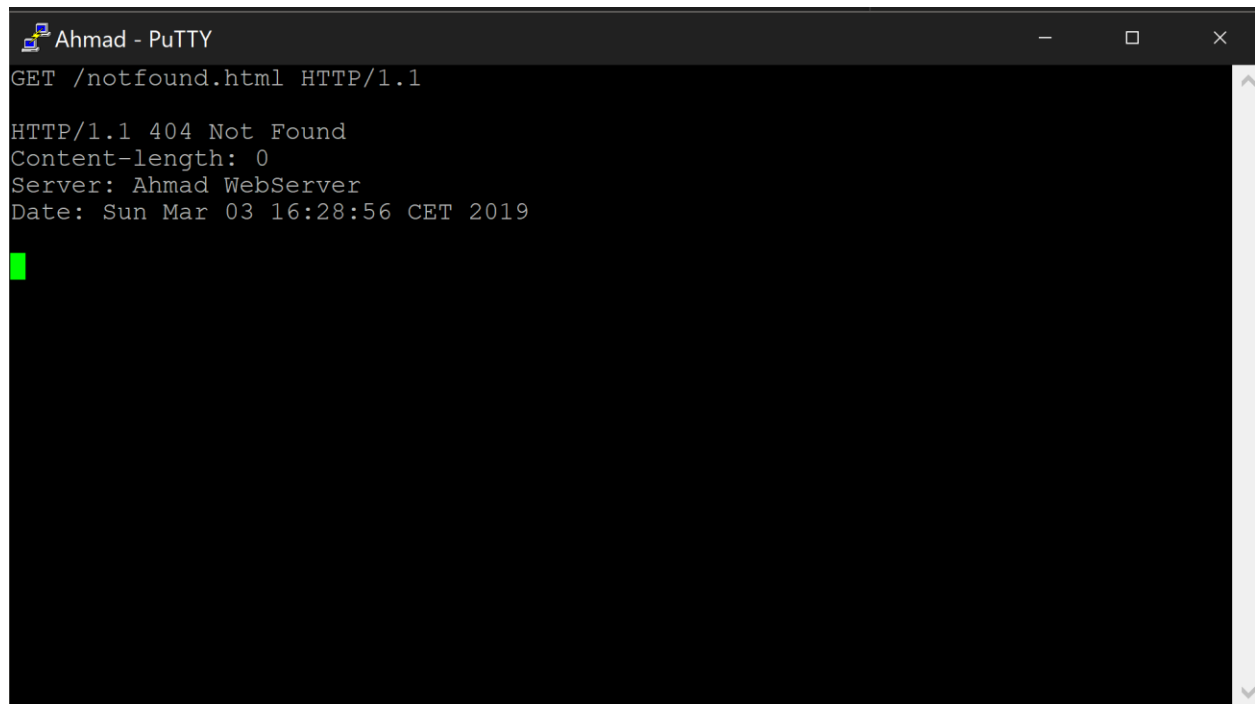
Content-Type: the type of the file requested which is text/html which indicate the browser that the file of a text type.

Server: name of the server.

Date: the time and date when the request was created.

the content of the body is a html tag.

404 Not Found

A screenshot of a PuTTY terminal window titled "Ahmad - PuTTY". The terminal shows an HTTP GET request for "/notfound.html" and the corresponding response headers. The response status is "404 Not Found". The headers include "Content-length: 0", "Server: Ahmad WebServer", and "Date: Sun Mar 03 16:28:56 CET 2019". A green cursor is visible on the line following the headers.

```
Ahmad - PuTTY
GET /notfound.html HTTP/1.1

HTTP/1.1 404 Not Found
Content-length: 0
Server: Ahmad WebServer
Date: Sun Mar 03 16:28:56 CET 2019
█
```

Using the GET `/notfound.html HTTP/1.1` which is a file that is not in the directory.

the headers response:

Version of the HTTP 1.1.

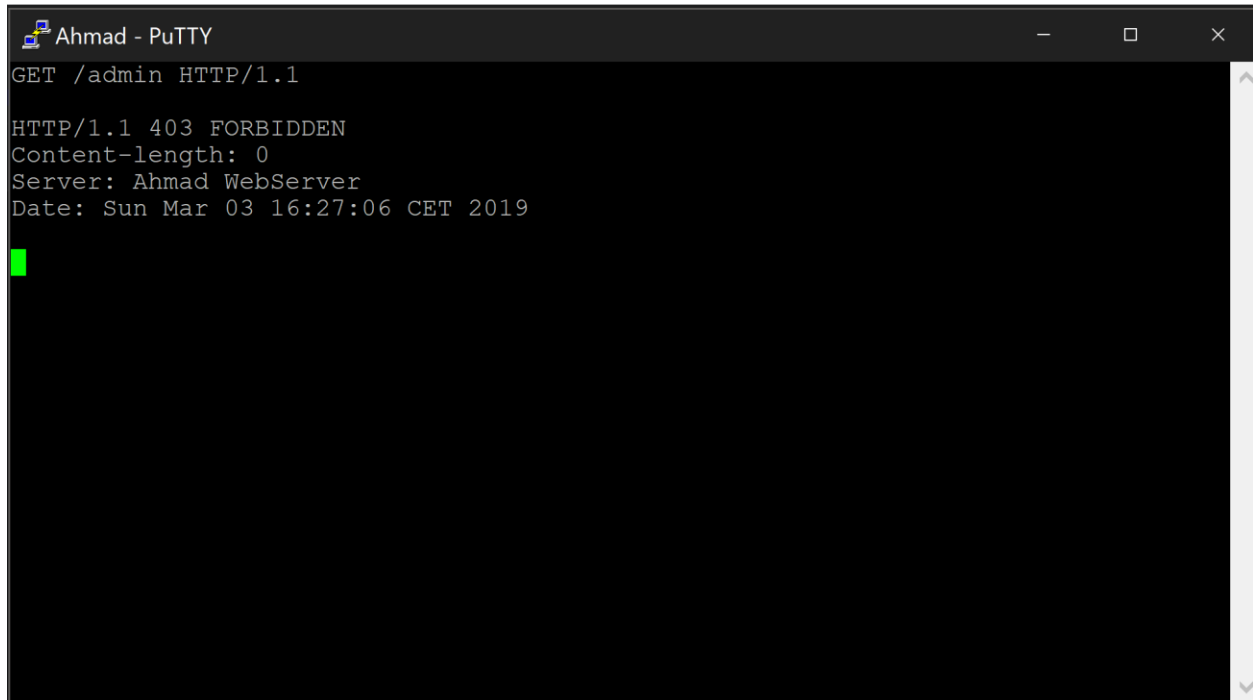
Response status: 404 Not Found.

Content-Length: the size of the file in bytes.

Server: name of the server.

Date: the time and date when the request was created.

403 Forbidden

A screenshot of a PuTTY terminal window titled "Ahmad - PuTTY". The window shows an HTTP request and response. The request is "GET /admin HTTP/1.1". The response is "HTTP/1.1 403 FORBIDDEN" with headers "Content-length: 0", "Server: Ahmad WebServer", and "Date: Sun Mar 03 16:27:06 CET 2019". A green cursor is visible on the line following the response headers.

```
Ahmad - PuTTY
GET /admin HTTP/1.1

HTTP/1.1 403 FORBIDDEN
Content-length: 0
Server: Ahmad WebServer
Date: Sun Mar 03 16:27:06 CET 2019
█
```

Using the GET /admin *HTTP/1.1* denied access when trying to access the admin file.

the headers response:

Version of the HTTP 1.1.

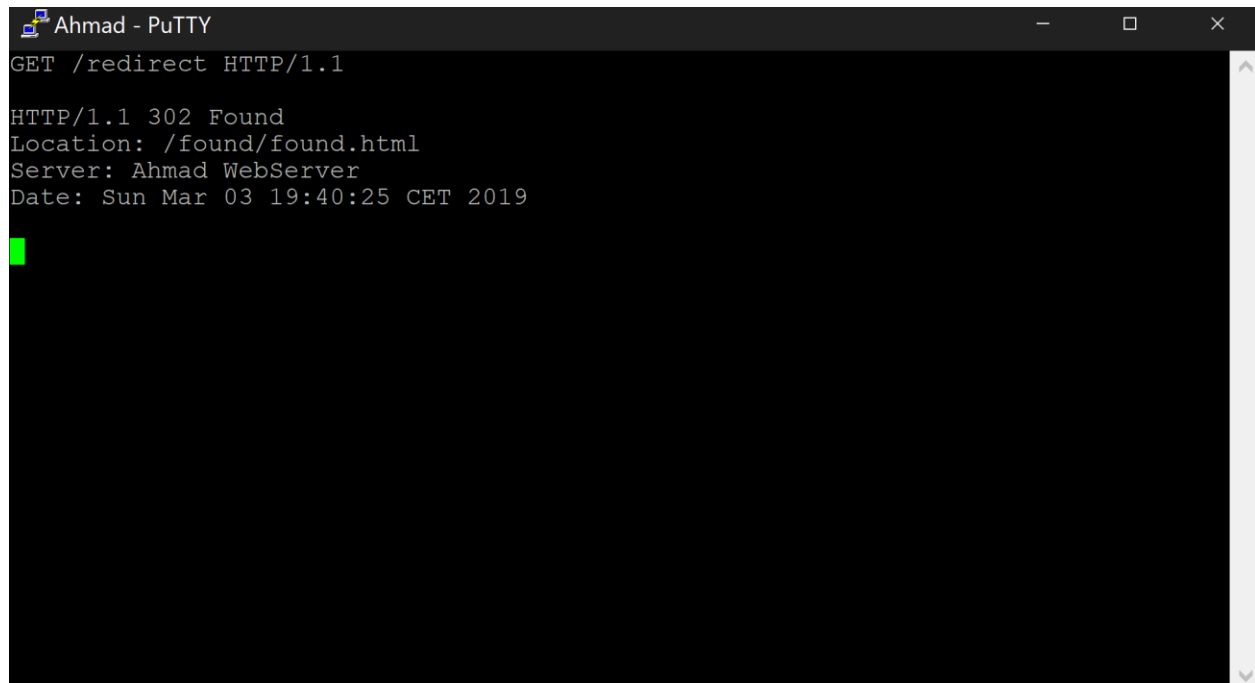
Response status: 403 FORBIDDEN.

Content-Length: the size of the file in bytes.

Server: name of the server.

Date: the time and date when the request was created.

302 Found

A screenshot of a PuTTY terminal window titled "Ahmad - PuTTY". The terminal shows an HTTP GET request for "/redirect" and the corresponding response headers for a 302 Found status. The response headers include the location "/found/found.html", the server name "Ahmad WebServer", and the date "Sun Mar 03 19:40:25 CET 2019". A green cursor is visible on the line following the date.

```
Ahmad - PuTTY
GET /redirect HTTP/1.1

HTTP/1.1 302 Found
Location: /found/found.html
Server: Ahmad WebServer
Date: Sun Mar 03 19:40:25 CET 2019
```

Using the GET `/redirect HTTP/1.1` results in a redirect to the new path.

the headers response:

Version of the HTTP 1.1.

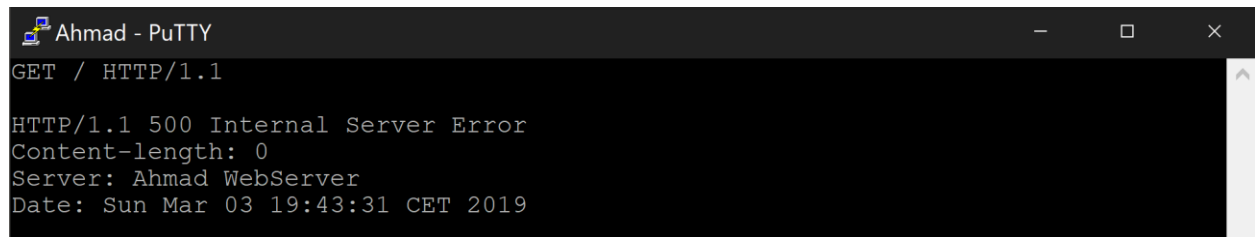
Response status: 302 Found.

Location: the path of the new file requested by the client using the old path.

Server: name of the server.

Date: the time and date when the request was created.

500 Internal Server Error

A screenshot of a PuTTY terminal window titled "Ahmad - PuTTY". The window shows the output of an HTTP GET request. The first line is "GET / HTTP/1.1". The second line is "HTTP/1.1 500 Internal Server Error". The third line is "Content-length: 0". The fourth line is "Server: Ahmad WebServer". The fifth line is "Date: Sun Mar 03 19:43:31 CET 2019".

```
Ahmad - PuTTY
GET / HTTP/1.1

HTTP/1.1 500 Internal Server Error
Content-length: 0
Server: Ahmad WebServer
Date: Sun Mar 03 19:43:31 CET 2019
```

Using the `GET / HTTP/1.1` results in Internal Server Error

(Note: in order to show 500 error we had to generate it by ourselves in the code)

the headers response:

Version of the HTTP 1.1.

Response status: 500 Internal Server Error.

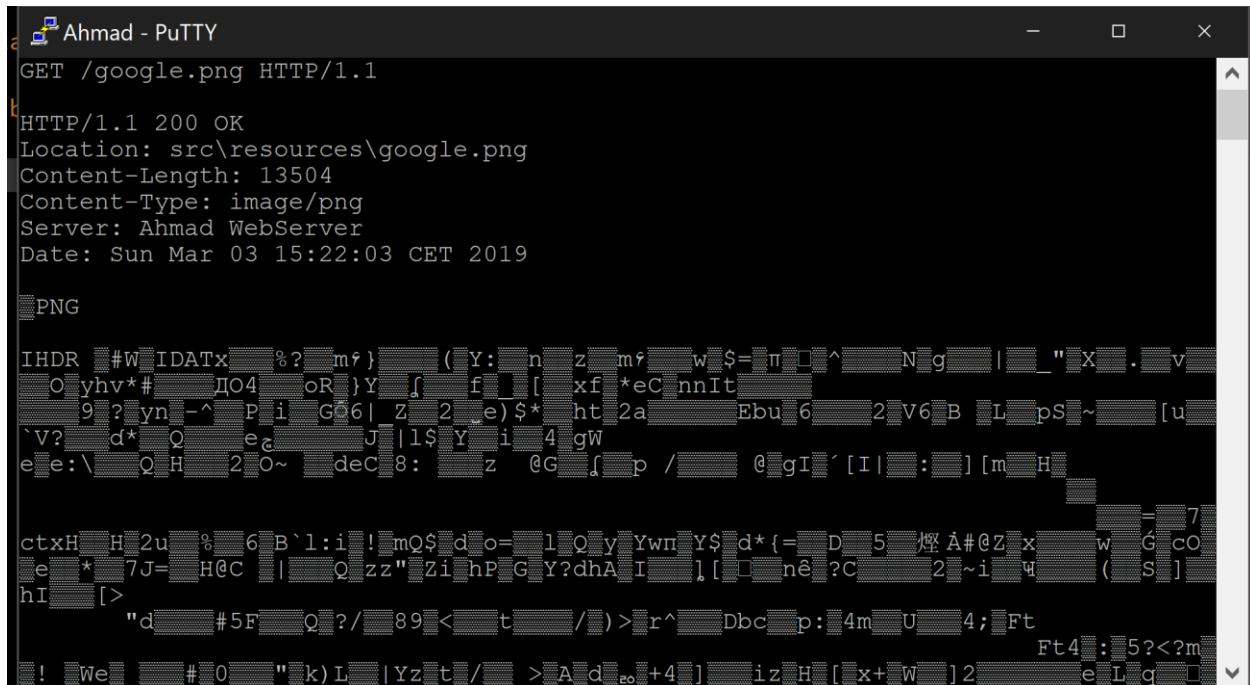
Content-Length: the size of the file in bytes.

Server: name of the server.

Date: the time and date when the request was created.

Get png image using Telnet

the picture below shows a request by telnet for getting a png image. It shows the headers of the response and a part of the response body.



```
Ahmad - PuTTY
GET /google.png HTTP/1.1

HTTP/1.1 200 OK
Location: src\resources\google.png
Content-Length: 13504
Content-Type: image/png
Server: Ahmad WebServer
Date: Sun Mar 03 15:22:03 CET 2019

PNG
IHDR 1W IDATx %? m? } ( Y: n z m? w $= π □ ^ N g | | _ " X . v
O yhv*# 004 oR }Y f f [ xf *eC nnIt
9 ? yn -^ P i G6| Z 2 e)$* ht 2a Ebu 6 2 V6 B L pS ~ [u
`V? d* Q e J |l$ Y i 4 gW
e e:\ Q H 2 O~ deC 8: z @G f p / @ gI ' [I| : ] [m H
7
ctxH H 2u % 6 B`l:i ! mQ$ d o= l Q y Ywn Y$ d* (= D 5 煙 Å#eZ x w G cO
e * 7J= H@C | Q zz" Zi hP G Y?dhA I l [ nê ?C 2 ~i 4 ( S ]
hI [ >
"d #5F Q ?/ 89 < t / ) > r^ Dbc p: 4m U 4; Ft
Ft4 : 5?<?m
! We # 0 " k) L |Yz t / > A d 20 +4 ] iz H [ x+ W ] 2 e L q
```

Using the GET /google.png HTTP/1.1 we get the png image in the file user1.

the headers response:

Version of the HTTP 1.1.

Response status: 200 OK.

Location: the path of the file requested by the client.

Content-Length: the size of the file in bytes.

Content-Type: the type of the file requested which is image/png which indicate the browser that the file of a image type.

Server: name of the server.

Date: the time and date when the request was created.

the content of the body is a binary form of the png.

