



# Report

## Assignment 2 - *Web Server*

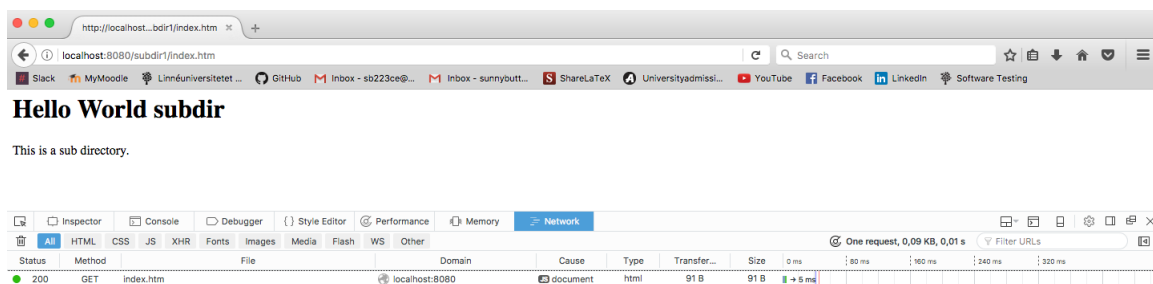
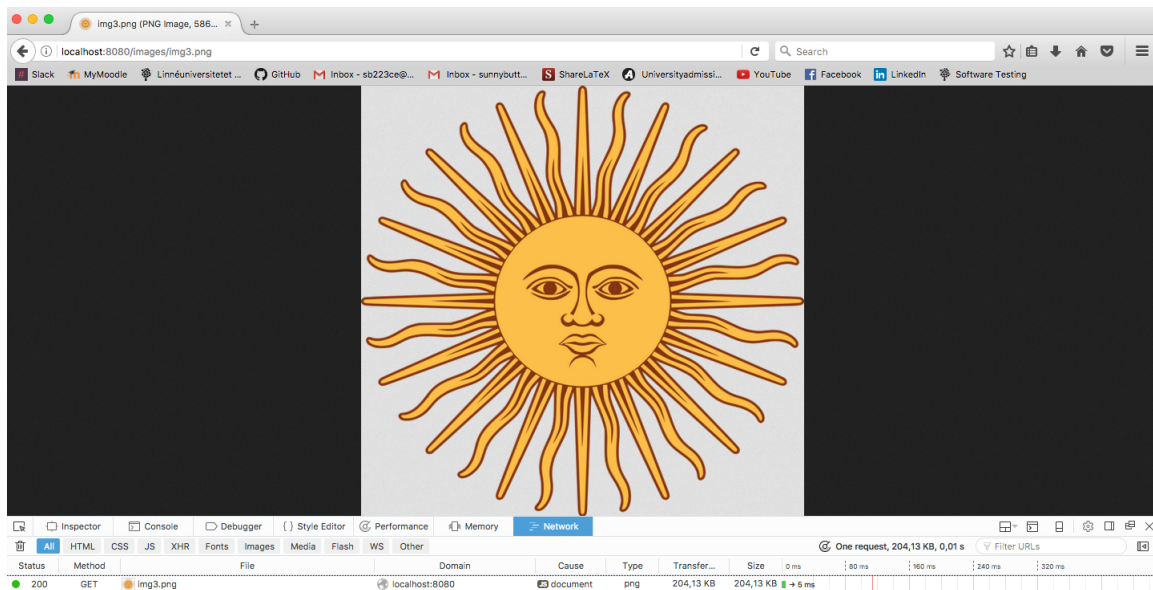
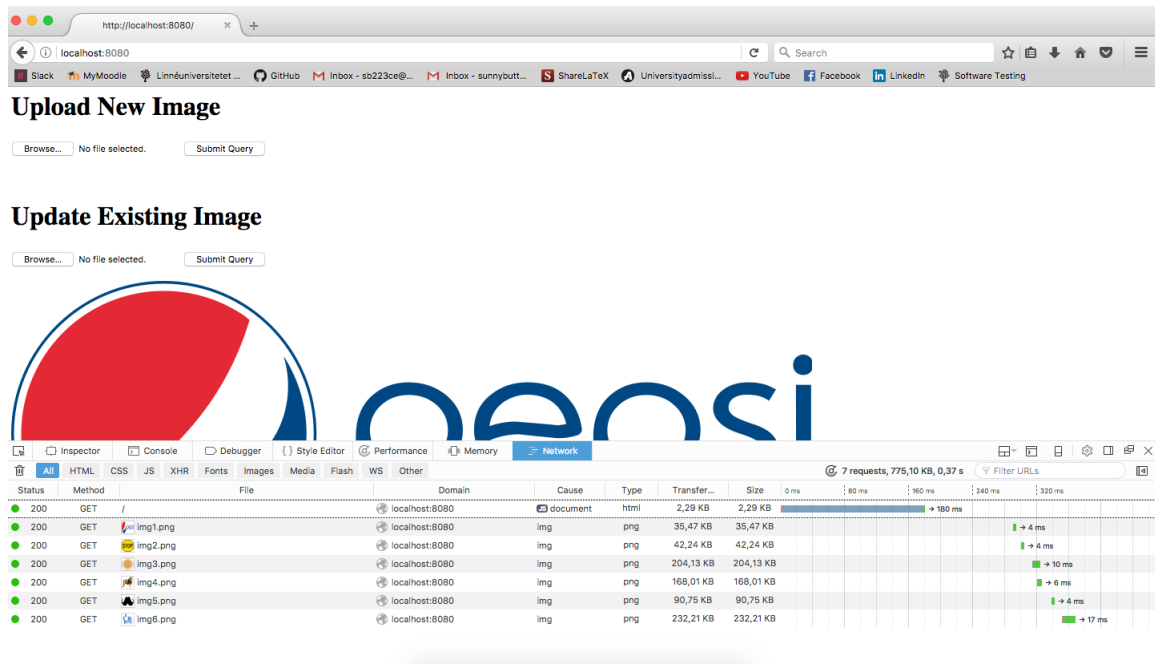


23 Feb, 2017

*Semester:* Spring 2017  
*Course:* Computer Networks  
*Authors:* Sarpreet Singh Buttar

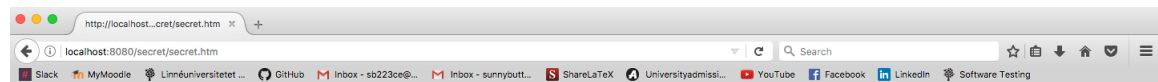
# Problem 1

The below pictures shows the result of GET request made by client using localhost:8080.



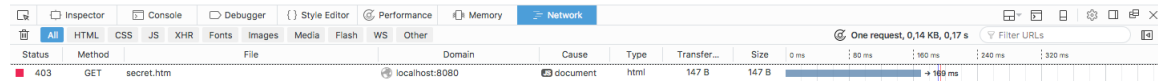
## Problem 2

### 403 Forbidden



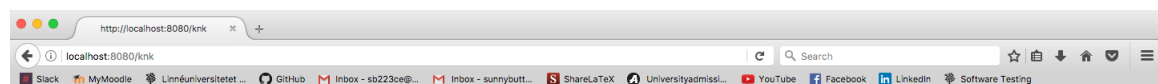
### 403 Forbidden

The request is valid but does not have the necessary permissions for accessing the resource.



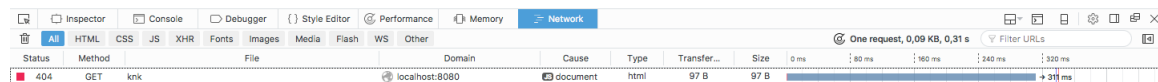
The above picture shows that client tried to access the secret file but server respond with 403. We have made one directory named as *secret*, whenever client will try to access the file in this directory server will respond with 403 because we always check the file parent and if it is *secret* then server return 403.

### 404 Not Found



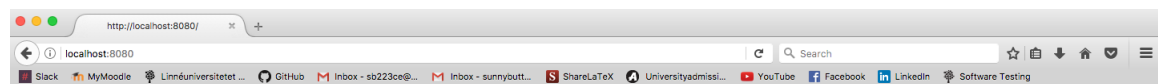
### 404 Not Found

The requested resource could not be found.



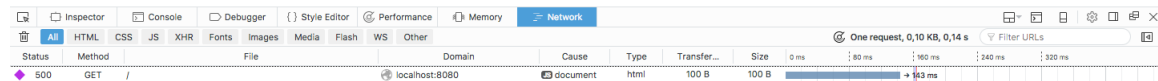
The above picture shows that client tried to access the file which is not present in the directory and server respond with 404.

### 500 Internal Server Error



### 500 Internal Server Error

Server have some internal problem

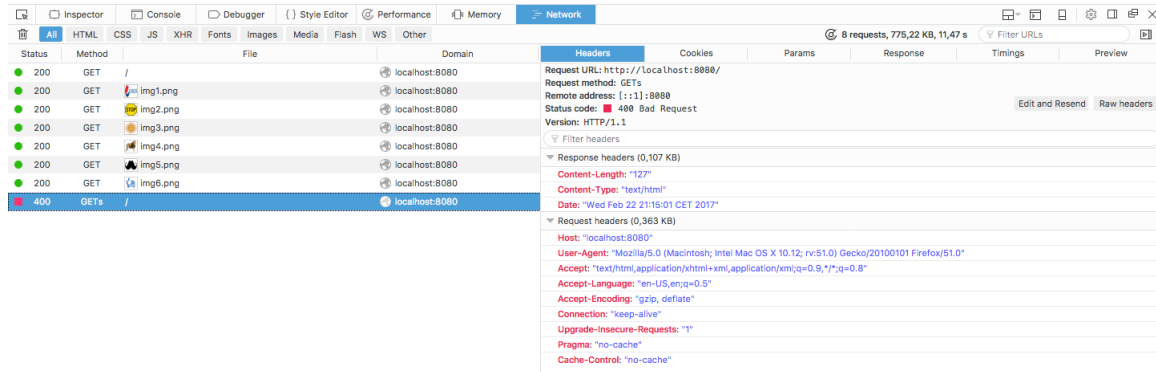


The above picture shows that client tried to access the default index file but server encounter an internal problem. *Note:* For testing this response I through an `IOException` by myself otherwise it is difficult to test it. Below is the screenshot of my self made `IOException`.

```
2    */
3    private String readHeader(BufferedReader reader) throws IOException, NumberFormatException {
4
5        StringBuilder header = new StringBuilder();
6
7        if(header.toString().isEmpty()) throw new IOException();
8
9        while (true) {
10
11            String line = reader.readLine();
12
13            // ... (rest of the code) ...
14        }
15    }
```

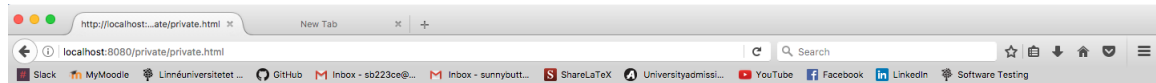
# VG Task 1

## 400 Bad Request



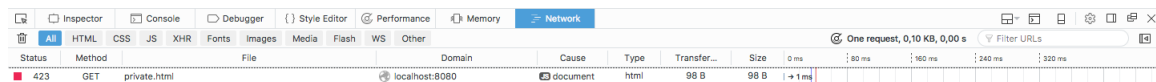
The above picture shows that client sent *GETs* instead of *GET* which results in 400 Bad Request.

## 423 Locked



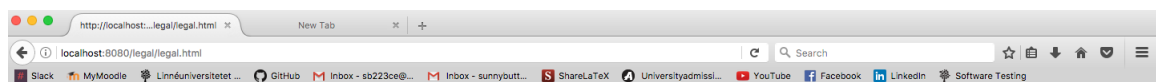
### 423 Locked

The resource that is being accessed is locked.



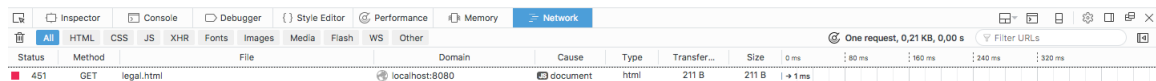
The above picture shows that client trying to access private files and server respond with 423. For this we have made a directory called 'private' and all the files inside the directory will show same response code. However, there is only 1 file but we always check the parent file before responding which means if new files will be added it will be also not visible.

## 451 Unavailable For Legal Reasons



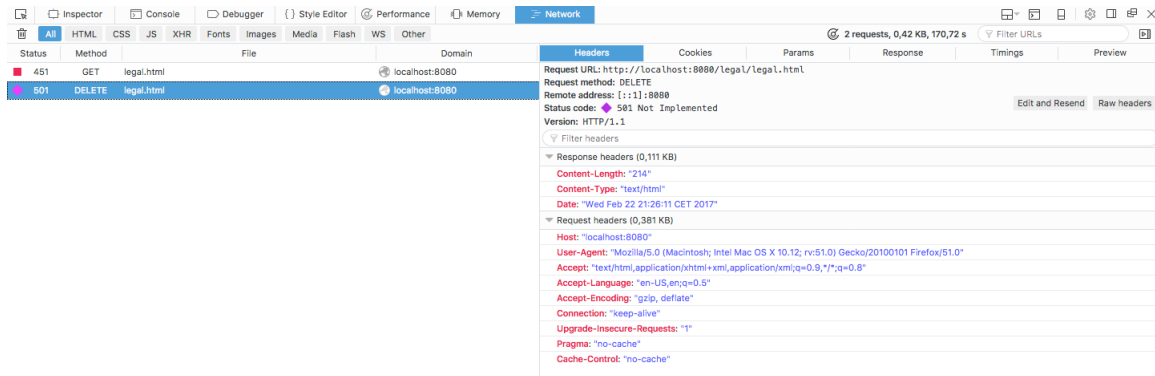
### 451 Unavailable For Legal Reasons

A server operator has received a legal demand to deny access to a resource or to a set of resources that includes the requested resource.



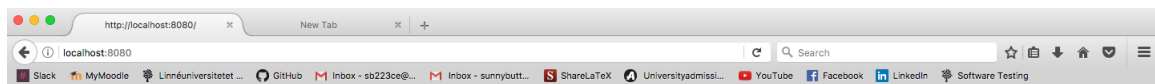
The above picture shows that client trying to access legal files and server respond with 423 because server operator has received a legal demand to deny access to this resource. For this we have made a directory called legal and its logic to deliver the response code is same as 423 Locked but it will show 451.

## 501 Not Implemented

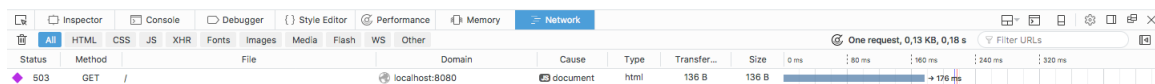


The above picture shows that client sent a *DELETE* request but server does not support it and respond with 501.

## 503 Service Unavailable

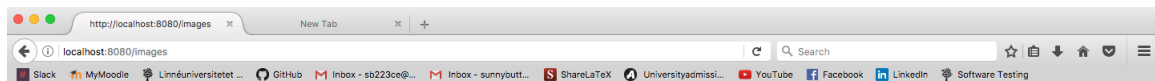


The server is currently unavailable because it is down for maintenance.



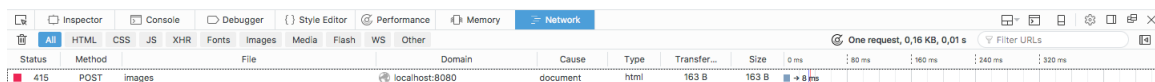
The above picture shows that client tried to connect server but it is not available at the moment because it is down for the maintenance. For producing this response, turn *UNDER\_MAINTAINENCE* variable to *false* in *Server* class.

## 415 Unsupported Media Type



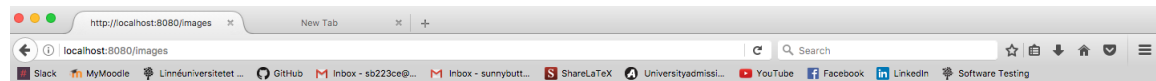
### 415 Unsupported Media Type

The server does not support requested media type. It only support png and jpg/jpeg media types.



The above picture shows that client tried to upload a file which is not supported by server. In this case server only supports png and jpeg/jpg images. For producing this response, please upload the file other than supported types such as .java, .pdf etc.

## 201 Created



### 201 Created

The request has been fulfilled, resulting in the creation of R415.png

Status	Method	File	Domain	Cause	Type	Transfer...	Size	0 ms	80 ms	160 ms	240 ms	320 ms
201	POST	images	localhost:8080	document	html	122 B	122 B					

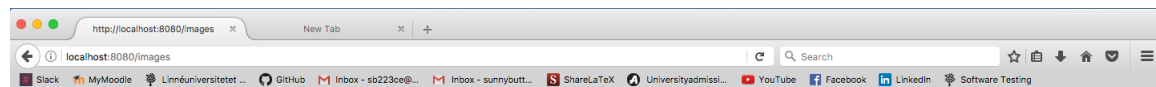
The above picture shows that client upload a file using POST request and server accept it and save it and return 201 which means a new file has been created and client can access this file using localhost:8080/images/<filename>.

## 204 No Content

Status	Method	File	Domain	Cause	Type	Transfer...	Size	0 ms	80 ms	160 ms	240 ms	320 ms
204	POST	images	localhost:8080	document	html	144,81 KB	0 B					

The above picture shows that client update the existing picture and server respond successfully proceed the request and respond with 204 mean request is fulfilled but no content to show. *Note:* We have found that HTML5 does not support PUT method any more that is why client is using POST method for this. For fixing this issue we have added "METHOD-PUT" from our self and then check it while parsing the request and if it is present then we set the request type manually. Some of the students have used ajax for this but I am familiar with javascript or ajax so find this idea to overcome this problem.

## 409 Conflict



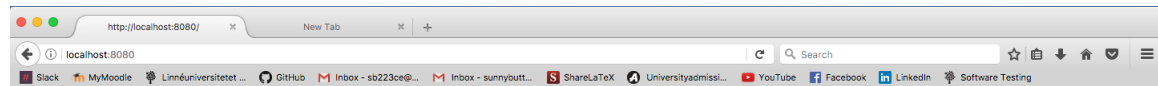
### 409 Conflict

The request could not be processed because R204.png does not exist. Please use POST request.

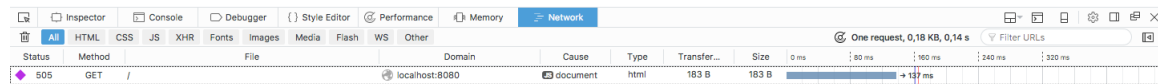
Status	Method	File	Domain	Cause	Type	Transfer...	Size	0 ms	80 ms	160 ms	240 ms	320 ms
409	POST	images	localhost:8080	document	html	145 B	145 B					

The above picture shows that client trying to update the existing file but the file is not present on the server side. Due to this server respond with 409 and suggest client to use POST request instead of PUT.

## 505 HTTP Version not Supported



The server does not support the HTTP protocol version used in the request. Only 1.1 is supported by the server.



The above picture shows that client is using HTTP 1.0 but server only supports HTTP 1.1 version. In result server respond with 505 and suggest the client to use only HTTP 1.1. *Note:* For testing this response we have change the HTTP version manually in the code, otherwise it is difficult to test. Below is the picture of which helped to produce this response

```
private void parseHeader(String header) throws BadRequestException, VersionNotSupportedException,
    ArrayIndexOutOfBoundsException, NumberFormatException {

    String[] totalLines = header.split("\r\n");

    String[] request = totalLines[0].split(" ");

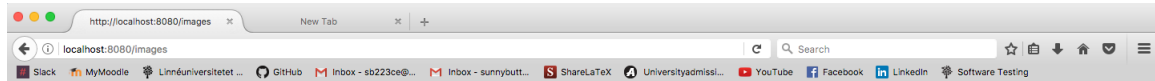
    if (request.length != 3) {
        throw new BadRequestException();
    }

    request[2] = "HTTP/1.0";

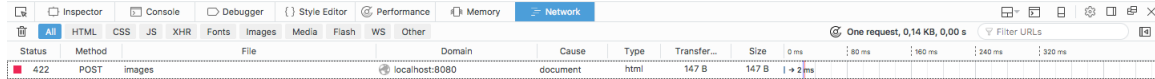
    if (!request[2].equals(HTTP_VERSION)) {
        throw new VersionNotSupportedException();
    }
}
```



## 422 Unprocessable Entity



The request was well-formed but was unable to be followed due to semantic errors.



The above picture shows that client has made proper request but there was some semantic errors which result into 422. This happen if some given data is in incorrect format. *Note:* For testing this error we have change the code manually. We know that content-length is always number, so for producing this error we produce NumberFormatException by reducing the substring size. In normal code it is 16 because "content-length: ".length() is 16 and for testing this we change it to 1 which means character will also in content-length which will result into NumberFormatException. Below is the picture of manually changed code for testing this response

```
private String readHeader(BufferedReader reader) throws IOException, NumberFormatException {  
    StringBuilder header = new StringBuilder();  
    while (true) {  
        String line = reader.readLine();  
        if (line == null || line.equals("\r\n") || line.isEmpty() || line.equals("")) {  
            break;  
        }  
        header.append(line + "\r\n");  
        if (line.startsWith("Content-Length")) {  
            contentLength = Integer.parseInt(line.substring(1));  
        }  
    }  
    return header.toString();  
}
```



## VG Task 2

*Note:* Sometimes, if picture is in mb's then it is half shown while uploading or updating. Trying to find the error but no success. We decided to limit the size but then we were running out of time and skip it.

### Difference between POST and PUT

POST	PUT
It is used for the creation of new resources. According to API it is non-idempotent means every time this request change result beyond the initial application	It is used for updating the existing resources. However it can be also used for creation but it is made for updating the resource. According to API it is idempotent which means this request proceed without changing the result beyond the initial application
It is non-repeatable. If client try to repeat, browser always shows an alert message to warn the client	It is repeatable
It is not safe to be repeatable because it will produce duplicate resources	It is safe to be repeatable because it only update the existing resources. Eg: int a = 10, if we repeat it 10 times 'a' will be still 10.

## Problem 3

### 408 Timeout

```
sunnybuttar — -bash — 89x24
Last login: Wed Feb 22 11:00:09 on ttys000
Sunnys-MacBook-Air:~ sunnybuttar$ telnet localhost 8080
Trying ::1...
Connected to localhost.
Escape character is '^]'.
HTTP/1.1 408 Request Time-out
Date: Thu Feb 23 08:41:32 CET 2017
Content-Length: 107
Content-Type: text/html

<html><body><h1>408 Request Time-out</h1><p>The server timed out waiting for the request.
</p></body></html>Connection closed by foreign host.
Sunnys-MacBook-Air:~ sunnybuttar$
```

### 505 HTTP Version Not Supported

```
sunnybuttar — -bash — 94x24
Last login: Thu Feb 23 08:45:15 on ttys000
Sunnys-MacBook-Air:~ sunnybuttar$ telnet localhost 8080
Trying ::1...
Connected to localhost.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.1 505 HTTP Version Not Supported
Date: Thu Feb 23 08:45:48 CET 2017
Content-Length: 183
Content-Type: text/html

<html><body><h1>505 HTTP Version Not Supported</h1><p>The server does not support the HTTP pro
tocol version used in the request. Only 1.1 is supported by the server.</p></body></html>Conne
ction closed by foreign host.
Sunnys-MacBook-Air:~ sunnybuttar$
```

### 501 Not Implemented

```
sunnybuttar — telnet localhost 8080 — 98x24
Last login: Thu Feb 23 08:45:32 on ttys000
Sunnys-MacBook-Air:~ sunnybuttar$ telnet localhost 8080
Trying ::1...
Connected to localhost.
Escape character is '^]'.
DELETE / HTTP/1.1

HTTP/1.1 501 Not Implemented
Date: Thu Feb 23 08:48:29 CET 2017
Content-Length: 214
Content-Type: text/html

<html><body><h1>501 Not Implemented</h1><p>The server either does not recognize the request method
, or it lacks the ability to fulfill the request. The sever only support GET, POST and PUT methods
</p></body></html>
```

## 200 OK

```
sunnybuttar — telnet localhost 8080 — 86×41
[Sunnys-MacBook-Air:~ sunnybuttar$ telnet localhost 8080
Trying ::1...
Connected to localhost.
Escape character is '^]'.
GET / HTTP/1.1

HTTP/1.1 200 OK
Date: Thu Feb 23 08:50:35 CET 2017
Content-Length: 2350
Content-Type: text/html

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>

    <h1>Upload New Image</h1>
    <form action = "/images" method="post">
        <input id="uploadedFile" type="file" onchange="POST();" />
        <input id="contents" name="" hidden="on"> <input type="submit">
    </form>

    <br/>

    <h1>Update Existing Image</h1>
    <form action = "/images" method="post">
        <input id="updatedFile" type="file" onchange="PUT();" />
        <input id="updateContents" name="" hidden="on"> <input type="submit">
    </form>

    
    
    
    
    
    
```

## 503 Service Unavailable

```
sunnybuttar — -bash — 83x24
Last login: Thu Feb 23 08:50:20 on ttys000
[Sunnys-MacBook-Air:~ sunnybuttar$ telnet localhost 8080
Trying ::1...
Connected to localhost.
Escape character is '^]'.
HTTP/1.1 503 Service Unavailable
Date: Thu Feb 23 08:53:23 CET 2017
Content-Length: 136
Content-Type: text/html

<html><body><h1>503 Service Unavailable</h1><p>The server is currently unavailable
because it is down for maintenance.</p></body></html>Connection closed by foreign h
ost.
Sunnys-MacBook-Air:~ sunnybuttar$
```

## 400 Bad Request

```
sunnybuttar — -bash — 81x24
Last login: Thu Feb 23 08:53:21 on ttys000
[Sunnys-MacBook-Air:~ sunnybuttar$ telnet localhost 8080
Trying ::1...
Connected to localhost.
Escape character is '^]'.
HELLO / HTTP/1.1

HTTP/1.1 400 Bad Request
Date: Thu Feb 23 08:55:29 CET 2017
Content-Length: 127
Content-Type: text/html

<html><body><h1>400 Bad Request</h1><p>The server cannot process the request due
to an apparent client error.</p></body></html>Connection closed by foreign host.
Sunnys-MacBook-Air:~ sunnybuttar$
```

## 404 Not Found

```
sunnybuttar — -bash — 79x24
Last login: Thu Feb 23 09:05:46 on ttys000
[Sunnys-MacBook-Air:~ sunnybuttar$ telnet localhost 8080
Trying ::1...
Connected to localhost.
Escape character is '^]'.
GET any HTTP/1.1

HTTP/1.1 404 Not Found
Date: Thu Feb 23 09:06:15 CET 2017
Content-Length: 97
Content-Type: text/html

<html><body><h1>404 Not Found</h1><p>The requested resource could not be found.
</p></body></html>Connection closed by foreign host.
Sunnys-MacBook-Air:~ sunnybuttar$
```

## Requesting an Image with Telnet

[illegible]

This above picture display a request made by telnet client for getting an image. The content of the image is a stream of bytes, which is shown in telnet as ASCII characters since ASCII characters are 1 byte each. The encoding is "UTF-8" and there for some ASCII characters are shown all greyed out. And an empty space is a received empty character. Some words can often be seen mixed in with the content of the image and these are embedded text encoded into the image deliberately. In this image we can see the Extension "PNG" which is written first, then later we can find that it is made in Photoshop. But these embedded texts are not shown after the image has been decoded into RGB values.