



COMPUTER NETWORKS CS421

HAMMAD KHAN MUSAKHEL
21801175

REPORT

The GET HTTP request could only be made after a connection (TCP) in our case is formed with the server. In order to form the TCP connection, we use the IP address of the server and the port number (80 in HTTP's case). HTTP requests can only be sent and received after a connection is formed. Since it is a TCP connection, the server allows the same port to be connected to many clients as same port is available to be connected to the client. Moreover, in order to send a request to a server we use the GET HTTP request in this assignment initially. To send a request to the server, we need to specify the request header; request header is an HTTP header that can be used in an HTTP request to provide information about the request context, so that the server can tailor the response. For example, the Accept header indicates the allowed and preferred formats of the response.

The format to send the GET request, which we employed, in python is:

```
"GET " + path + " HTTP/1.1\r\nHost: " + ipAddr + "\r\n" "Range: bytes= " + lowerBound + '-' + upperBound + "\r\n\r\n"
```

The initial header in the request sent is the type of request which the client aims to send to the server. Following that, the client provides the path of the URL to access and the HTTP connection type the client desires: HTTP/1.0 signals at the server to terminate the connection after the response and HTTP/1.1 signals the connection to be live until the client has explicitly signaled to terminate it. The '\n' terms to define further headers if the client wants to list; however, not all headers that can appear in a request are referred to as request headers by the specification. For example, the Content-Type header is referred to as a representation header. Moreover, we provide the header of Host with the IP address of the server, and in the case of getting partial-content byte wise, we specify Range: bytes = lowerbound-upperbound.

The aforementioned is the general format for HTTP requests. In order to send and receive headers only, we send the HEAD request. In that, we necessarily do not change much just the type of request from GET to HEAD; we provided the path and host address to retrieve the header of the request. This allows us to view the content length, content type, and much more.

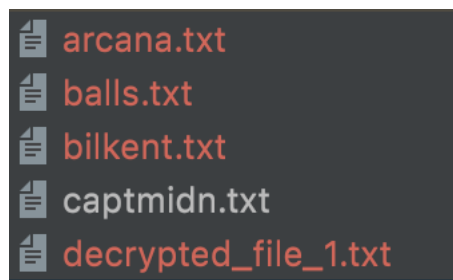
```
"HEAD " + path + " HTTP/1.0\r\nHost: " + ipAddr + "\r\n\r\n"
```

```

www.cs.bilkent.edu.tr/~cs421/fall21/project1/index1.txt
URL of the index file: www.cs.bilkent.edu.tr/~cs421/fall21/project1/index1.txt
Index file is downloaded
There are 9 files in the index
1. www.cs.bilkent.edu.tr/file.txt is not found
2. www.cs.bilkent.edu.tr/folder2/temp.txt is not found
3. www.textfiles.com/100/balls.txt (range = 0-999) is downloaded
4. www.cs.bilkent.edu.tr/~cs421/fall21/project1/bilkent.txt (range = 0-999) is downloaded
5. www.textfiles.com/games/arcana.txt (range = 0-999) is downloaded
6. www.cs.bilkent.edu.tr/~cs421/cs421/abc.txt is not found
7. www.cs.bilkent.edu.tr/~cs421/fall21/project1/files/numbers.txt (range = 0-999) is downloaded
8. www.cs.bilkent.edu.tr/~cs421/fall21/project1/files/decrypted_file_1.txt (range = 0-999) is downloaded
9. www.textfiles.com/100/captmidn.txt (range = 0-999) is downloaded

```

In this assignment, we are tasked to download the files the HTTP GET request demands. The index URL returns a number of Line Text Data. The line text data is then traversed to see if there is a content available at those URLs provided by the index URL. A 200 OK response indicates the availability of content and 404 NOT FOUND indicates otherwise. As shown in the image above, we send HEAD requests to the URLs and got the subsequent responses. Next, only those with 200 OK responses are selected for the GET request. For this example, we didn't provide the upper and lower bounds, thus, as instructed in the assignment, we used the 0-999 bound limits to download the content for the 200 OK ones. Since, the lower bound is 0, all the available content is downloaded



```

arcana.txt
balls.txt
bilkent.txt
captmidn.txt
decrypted_file_1.txt

```

The downloaded content is stored in the same directory with the names provided in the URL.

```

(venv) hammadmacho7@H4MM4D-mach07 FinalCopy % python FileDownloader.py www.cs.bilkent.edu.tr/~cs421/fall21/project1/index2.txt 200-700
www.cs.bilkent.edu.tr/~cs421/fall21/project1/index2.txt
URL of the index file: www.cs.bilkent.edu.tr/~cs421/fall21/project1/index2.txt
Index file is downloaded
There are 13 files in the index
1. www.cs.bilkent.edu.tr/~cs421/fall21/project1/files2/dummy1.txt (size = 11) is not downloaded
2. www.cs.bilkent.edu.tr/~cs421/fall21/project1/files2/dummy2.txt (size = 26) is not downloaded
3. www.cs.bilkent.edu.tr/~cs421/fall21/project1/files2/dummy3.txt (size = 57) is not downloaded
4. www.cs.bilkent.edu.tr/~cs421/fall21/project1/files2/dummy4.txt (size = 152) is not downloaded
5. www.cs.bilkent.edu.tr/~cs421/fall21/project1/files2/dummy5.txt is not found
6. www.cs.bilkent.edu.tr/~cs421/fall21/project1/files2/dummy6.txt (range = 200-700) is downloaded
7. www.cs.bilkent.edu.tr/~cs421/fall21/project1/files2/dummy7.txt (range = 200-700) is downloaded
8. www.cs.bilkent.edu.tr/~cs421/fall21/project1/files2/dummy8.txt (range = 200-700) is downloaded
9. www.cs.bilkent.edu.tr/~cs421/fall21/project1/files2/dummy9.txt (range = 200-700) is downloaded
10. www.cs.bilkent.edu.tr/~cs421/fall21/project1/files2/dummy10.txt is not found
11. www.cs.bilkent.edu.tr/~cs421/fall21/project1/files2/dummy11.txt (range = 200-700) is downloaded
12. www.cs.bilkent.edu.tr/~cs421/fall21/project1/files2/dummy12.txt (range = 200-700) is downloaded
13. www.cs.bilkent.edu.tr/~cs421/fall21/project1/files2/dummy13.txt (size = 0) is not downloaded

```

In the above example, we provide the range as 200-900. As we can see, those with 404 NOT FOUND are labelled with 'is not found' and the rest with 'is not downloaded' are not downloaded as their content length is smaller than the lower bound. However, the ones with content length greater than the upper bound, have the content till the upper bound downloaded.



The downloaded files are stored in the directory.