**WEB SERVICES**

420-511-VA

#### **LAB-1-PART-2**

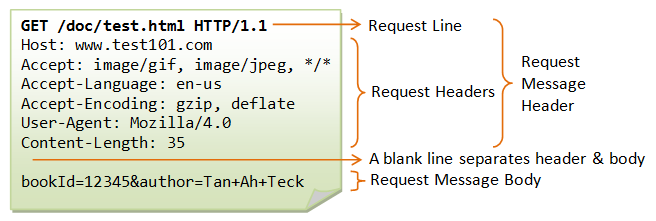
Note: You need to demo your completed LAB to the teacher after completion.

In this part we will simulate an HTTP Request and HTTP Response between a web browser and an HTTP server.

An HTTP Request sent by a browser to an HTTP Server (Apache) has the following components:

* **Request line:** that specifies the HTTP Verb or action to be taken on the server
* **Header and Header fields:** that consist of name value pairs to communicate information about the data being transferred and what needs to be done with it
* **An empty line**
* **Body:** the data being transferred, for example if the request is the result of a user submitting a form the body will contain the form’s data

#### **Sample HTTP GET Request**



1. Using the programs implemented in LAB-1, we will consider program-1 as a web browser and program-2 as a web server. We will send an HTTP Request from program-1 to Program-2 and an HTTP Response from program-2 to program-1.
   * Implement the below code where necessary to achieve our goal.
   * Code 1:

|  |
| --- |
| $request = "GET /page.html HTTP/1.1\r\n";  $request .= "Host: localhost\r\n";  $request .= "Connection: Close\r\n\r\n"; |

* + Code remarks:
    - Note that the \r\n are for carriage return and line feed (CRLF), meaning to specify that the program reading this input will go on a new line.
    - Note the double \r\n at the end of "Connection: Close\r\n\r\n", since after the header there should be an empty line before a possible body as per the HTTP specifications: <https://datatracker.ietf.org/doc/html/rfc2616#section-4.1>
    - Note that we broke down the response value into multiple code lines for clarity, but it is effectively equivalent to: $request = "GET /page.html HTTP/1.1\r\nHost: localhost\r\nConnection: Close\r\n\r\n";
  + Code 2, write a custom response:

|  |
| --- |
| $response = "You've reached the HTTP server. Welcome! \r\n\r\n";  $response .= "This is a response from program 2\r\n\r\n";  $response .= "The purpose of this Lab 1.2 is to simulate an HTTP Request and HTTP Response between a web browser and an HTTP server.\r\n\r\n"; |

* + Note that we could make the program-2 loop and wait for other requests and then send corresponding responses. You do not need to implement this.
  + You could test the program-1 code with a program-2 running on a colleagues machine in the LAB, instead of your own program-2.

1. Communicating with Apache.
   * If we want program-1 to send the HTTP Request to a real web server, Apache what should be done? modify the code to do that.
   * Create the html page page.html, and put in it a simple message, otherwise you will get the html for a 404 error page.
   * Test the code and write the response from Apache below.

|  |
| --- |
| HTTP/1.1 200 OK  Date: Mon, 29 Aug 2022 01:45:07 GMT  Server: Apache/2.4.52 (Win64) OpenSSL/1.1.1m PHP/8.1.2  Last-Modified: Mon, 29 Aug 2022 01:45:03 GMT  ETag: "22-5e7576997198a"  Accept-Ranges: bytes  Content-Length: 34  Connection: close  Content-Type: text/html  This is a message in page.html |

1. What happens if we remove one of the \n from the end of the request?

|  |
| --- |
| It takes a long time to response, and it shows a Client Error which is a 408 Request Timeout. |

1. What happens if we remove "Connection: Close\r\n\r\n" from the end of the request?

|  |
| --- |
| It takes a long time to response, and it shows a Client Error which is a 408 Request Timeout. |

1. What happens if, using the original request, we replace GET with DUMMY and make the request to Apache? What are the status codes and messages?

|  |
| --- |
| It takes a long time to response, and it shows a Server Error which is 501 Not Implemented. |