Green University of Bangladesh
Department of Computer Science and Engineering(CSE)

Faculty of Sciences and Engineering

Semester: (Spring, Year:2021), B.Sc. in CSE (Day)

LAB REPORT NO : 01

Course Title: Computer Networking Lab

Course Code: CSE 312 Section: DB

Lab Experiment Name:

Implement the GET HTTP method using JAVA for the webpage "http://webcode.me/ ".

Student Details

	Name	ID
1.	Hamad Ismail	201902046

Lab Date : 29/10/2022
Submission Date : 05/11/2022
Course Teacher's Name : Rusmita Halim Chaity

1. TITLE OF THE LAB EXPERIMENT

Implement the GET HTTP method using JAVA for the webpage “http://webcode.me/ ”.

2. OBJECTIVES/AIM

The HTTP GET request method is used to request a resource from the server. The GET request should only receive data (the server must not change its state).

3. PROCEDURE / ANALYSIS / DESIGN

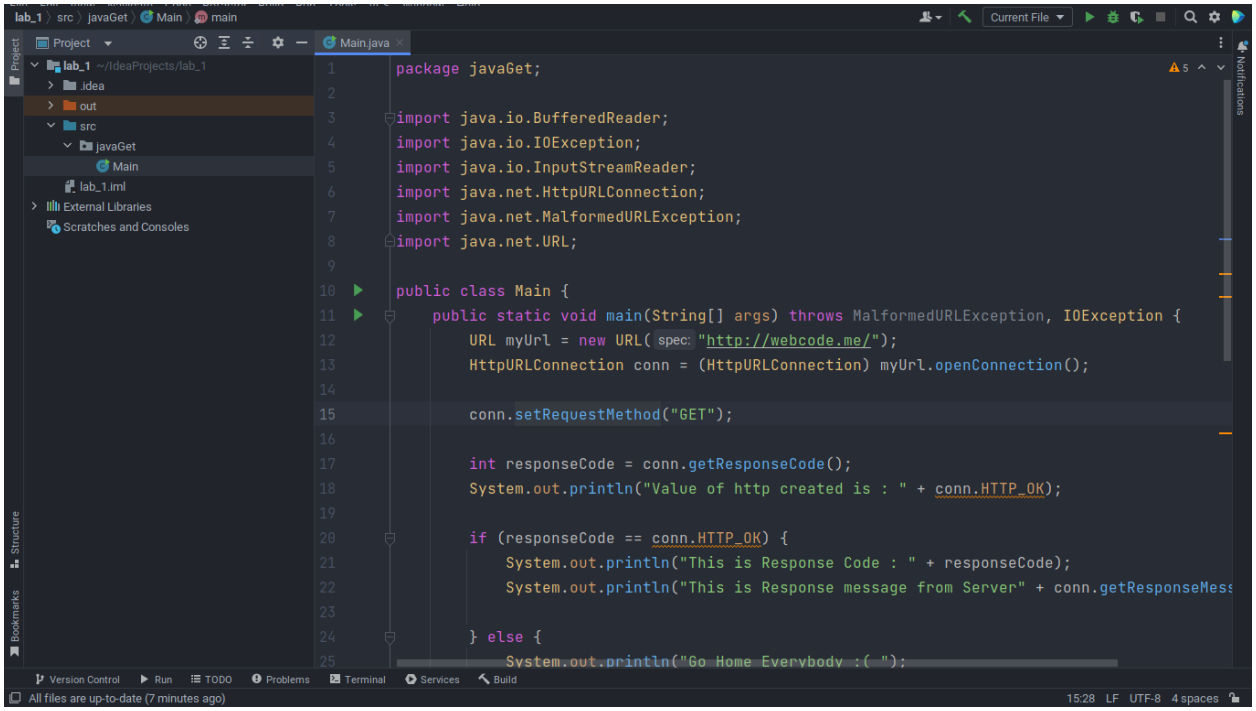
1. Firstly we created a new java project named **lab_1**
2. Then, create a package called **javaGet** and create a java **Main** class under this package.
3. Then under the **main** method create an object called **myUrl** of the **URL** class with the url of the webpage in the constructor.
4. Create an instance of the **HTTPURLConnection** class, made as a return from the function openConnection() of the url instance.
5. Enable the **HTTP** method that the client wants to do, using **setRequestMethod()** function. Here the parameter will be “**GET**”.
6. if **responseCode** == **HTTP_OK** then print out the response code and response messages from the web server.
   Here **HTTP_OK** is a defined value that equals 200.
7. An empty string is initialized to store the read contents from the connection instance created at Step 4.
8. While reading up to the end of the posted content do append each sentence of the posted content with the initialized string.
9. Print the created string, that contains the entire content of the web page.

4. IMPLEMENTATION

```
package javaGet;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;

public class Main {
        public static void main(String[] args) throws
MalformedURLException, IOException {
        URL myUrl = new URL("http://webcode.me/");
```

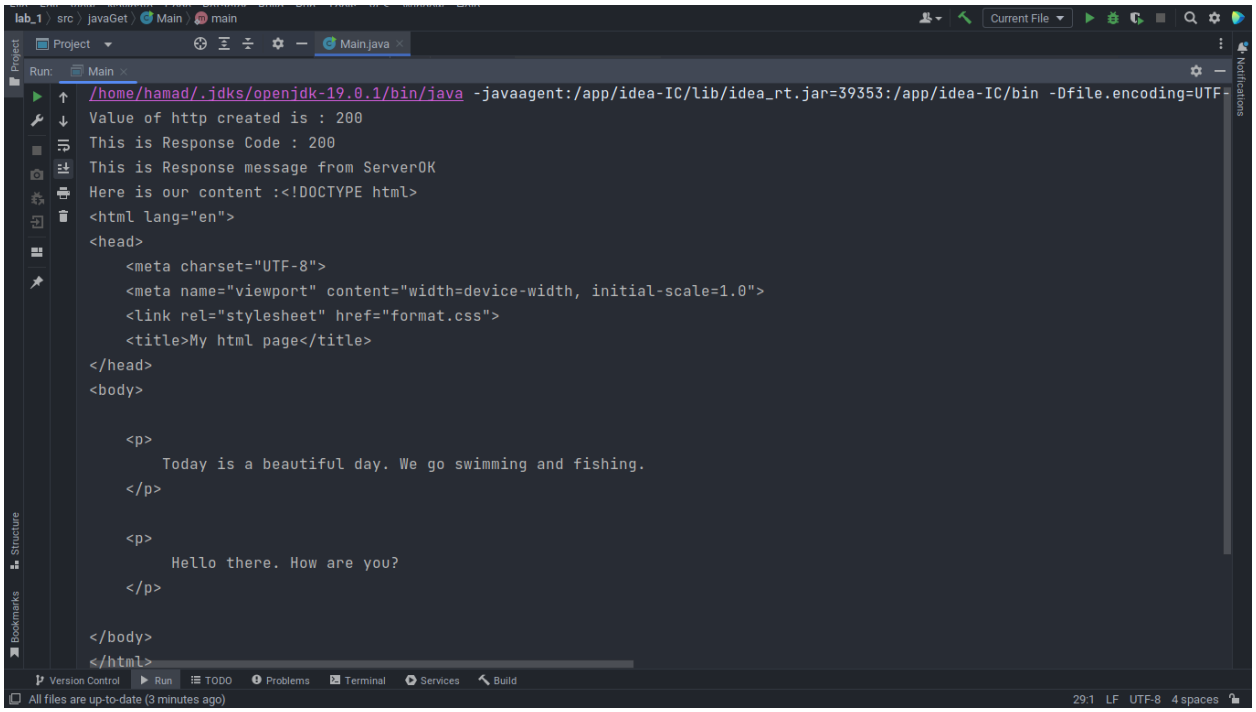
```
            HttpURLConnection conn = (HttpURLConnection)
myUrl.openConnection();

       conn.setRequestMethod("GET");

       int responseCode = conn.getResponseCode();
        System.out.println("Value of http created is : " +
conn.HTTP_OK);

       if (responseCode == conn.HTTP_OK) {
            System.out.println("This is Response Code : " +
responseCode);
              System.out.println("This is Response message
from Server" + conn.getResponseMessage());

       } else {
           System.out.println("Go Home Everybody :( ");
       }


                          InputStreamReader in = new
InputStreamReader(conn.getInputStream());
       BufferedReader buffer = new BufferedReader(in);

       StringBuffer fromServer = new StringBuffer();

       String eachLine = null;

       while ((eachLine = buffer.readLine()) != null) {
           fromServer.append(eachLine);
           fromServer.append(System.lineSeparator());
       }
       buffer.close();


                System.out.println("Here is our content
:"+fromServer);
   }
}
```

5. TEST RESULT / OUTPUT



```
package javaGet;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;

public class Main {
    public static void main(String[] args) throws MalformedURLException, IOException {
        URL myUrl = new URL( spec: "http://webcode.me/");
        HttpURLConnection conn = (HttpURLConnection) myUrl.openConnection();

        conn.setRequestMethod("GET");

        int responseCode = conn.getResponseCode();
        System.out.println("Value of http created is : " + conn.HTTP_OK);

        if (responseCode == conn.HTTP_OK) {
            System.out.println("This is Response Code : " + responseCode);
            System.out.println("This is Response message from Server" + conn.getResponseMess

        } else {
            System.out.println("Go Home Everybody :( ");
```

Fig 5.1 : Implement the GET HTTP Method using Java



```
/home/hamad/.jdks/openjdk-19.0.1/bin/java -javaagent:/app/idea-IC/lib/idea_rt.jar=39353:/app/idea-IC/bin -Dfile.encoding=UTF-
Value of http created is : 200
This is Response Code : 200
This is Response message from ServerOK
Here is our content :<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="format.css">
    <title>My html page</title>
</head>
<body>

    <p>
        Today is a beautiful day. We go swimming and fishing.
    </p>

    <p>
         Hello there. How are you?
    </p>

</body>
</html>
```

Fig 5.2 : Output of the GET HTTP Method.

Here we have got our expected output for the GET method on the given web page, *"http://webcode.me/ "*.
GET HTTP method simply returns the contents of the webpage.

6. ANALYSIS AND DISCUSSION

1. Here we implement **HTTP GET** Method using Java.
2. This method is mainly used if the client simply wants to read the entire contents of any certain web page.
3. Here no need to enforce any content getting ready since, nothing is to be done for writing. Only a simple reading request is pushed towards the server.
4. If the server accepts which is verified using the response code 200 by the client, then the entire web page is returned as a string instance, that is finally simply read by the client.
5. The following section simplifies the steps to implement the GET HTTP method for the web page "http://webcode.me/ ".