

Simple HTTP Server

Web Server

เว็บเซิร์ฟเวอร์ (Web Server) หรือ HTTP Server คือ โปรแกรมให้บริการที่เก็บเว็บไซต์บนเครื่องผู้ให้บริการ (Server) แล้วให้ผู้ใช้ (Client) เรียกชมหน้าเว็บไซต์ได้โดยใช้โปรโตคอล HTTP ผ่านทางเว็บเบราว์เซอร์

โดย HTTP โปรโตคอล คือ โปรโตคอลที่ใช้สื่อสารแลกเปลี่ยนข้อมูล (เช่น เอกสาร HTML) ระหว่าง Web Server และ Web Client (Browser) โดยในการสื่อสารนั้นจะใช้ HTTP Request ในการร้องขอเนื้อหาจาก Web Server และ Web Server จะตอบกลับมาในรูปแบบ HTTP Response ซึ่ง HTTP Protocol มีหน้าตา ดังนี้

HTTP Request

```
GET /javafaq/images/cup.gif HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/3.01 (Macintosh; I; PPC)
Host: www.oreilly.com:80
Accept: image/gif, image/x-xbitmap, image/jpeg, */*
<blank line>
```

โดยบรรทัดแรกคือ Request Line ประกอบด้วย HTTP Method, Resource Path จาก URL และ HTTP Protocol Version และบรรทัดต่อๆมาคือ Header บ่งบอกข้อมูลเพิ่มเติม และ Body บ่งบอกเนื้อหาที่แนบมาสำหรับ Method อื่นๆอย่าง Post Method

HTTP Response

```
HTTP/1.0 200 OK           ← Status line
Last-Modified: Monday, 27-Dec-99 22:14 GMT ← Header
Content-Type: text/html
Content-Length: 1639

<HTML>                    ← Entity Body
<HEAD>
<TITLE> Our homepage </TITLE>
</HEAD>
<BODY>
..... // BODY GOES HERE
</BODY>
</HTML>
```

โดยบรรทัดแรกคือ Status line ประกอบด้วย HTTP Protocol version, Status code และ ความหมาย ใช้ในการบ่งบอกสถานะต่างๆ เช่น 4XX คือเกิดความผิดพลาดจากฝั่ง Client 5XX, คือเกิดความผิดพลาดจากฝั่ง Server, 2XX คือ

นาย ปฏิภาณ สมดุลยกร 5821602742, นาย พีรทัต มั่นทะกะ 5821602815 และ นาย ภาณุพงศ์ เต่าแก้ว 5821602823 Q03-3 หมู่ 700

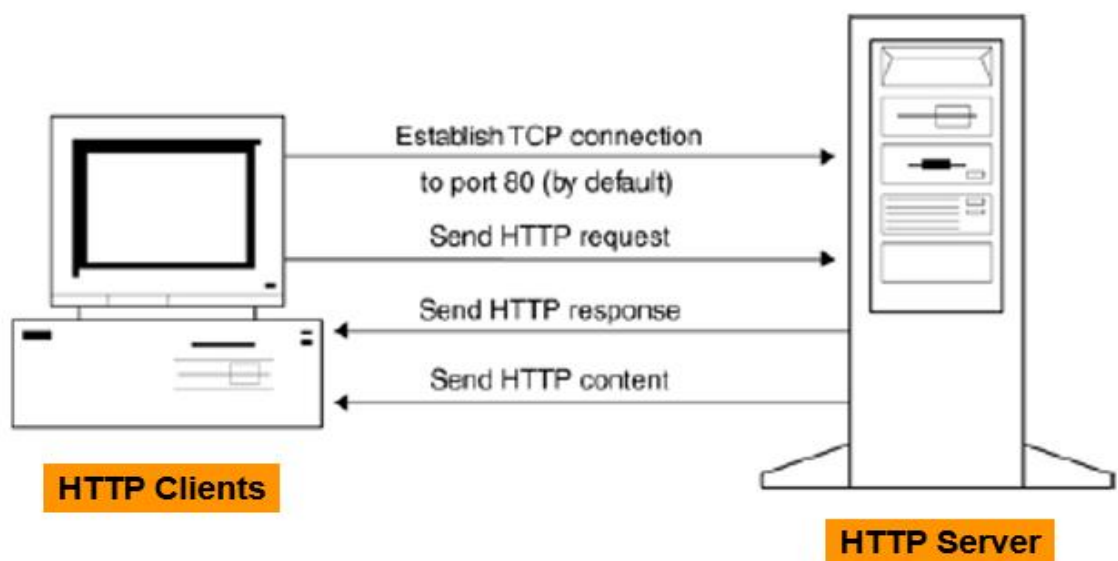
การตอบกลับ OK ไม่มีปัญหา เป็นต้น บรรทัดต่อมาคือ Header ใช้บ่งบอกข้อมูลเพิ่มเติม จากนั้นเว้นบรรทัดแล้วต่อด้วย Entity Body คือ Content ที่ Client ต้องการ

ซึ่ง HTTP Protocol ทำงานอยู่ในระดับ Application Layer บนโปรโตคอล TCP/IP ซึ่งเป็น Layer ระดับล่าง ดังนั้น การสร้าง HTTP Server จึงสร้างบน TCP Server (Socket Server) โดย HTTP Server มีการทำงานดังนี้

HyperText Transfer Protocol



How does HTTP work?



1. Client และ Server จะสร้างการเชื่อมต่อกันด้วย TCP/IP Protocol ที่ IP Address ของ Host และ Port ที่อยู่ของ Web Server ซึ่งปกติจะตั้งอยู่ที่ Port 80
2. ทำการรับส่งข้อมูล โดยการรับส่งข้อมูลนี้คือการสื่อสารด้วย HTTP Protocol โดยจะเริ่มจากฝั่ง Client ส่ง HTTP Request มาที่ฝั่ง Server จากนั้น Server จะทำการแปลความหมายของ HTTP Request นั้นและประมวลผลแล้วตอบกลับด้วย HTTP Response และ Content ที่ Client ต้องการ

สร้าง HTTP Server ด้วยภาษา Java

Class MultithreadHttpServer

```
1 package httpserver;
2
3 import java.io.*;
4 import java.net.*;
5 /**
6  *
7  * @author peeratat.m.p
8  */
9 public class MultithreadHttpServer extends Thread {
10
11     private final Socket clientConnection;
12     private final String host;
13     private final String port;
14
15     public MultithreadHttpServer(Socket clientConnection, InetAddress serverHost, ServerSocket socket) {
16         this.host = serverHost.getHostAddress();
17         this.port = socket.getLocalPort() + "";
18         this.clientConnection = clientConnection;
19     }
20
21     @Override
22     public void run() {
23         try {
24             //get input-output stream
25             InputStreamReader isr = new InputStreamReader(clientConnection.getInputStream());
26             BufferedReader reader = new BufferedReader(isr);
27
28             BufferedOutputStream bos = new BufferedOutputStream(clientConnection.getOutputStream());
29             PrintWriter pw = new PrintWriter(new OutputStreamWriter(bos));
30
31             //Read first line input (Request)
32             String line = reader.readLine();
33             if (line != null) { //Check for NullPointerException : (
34
35                 //http method & url from request
36                 String requestPart[] = line.split(" ");
37                 String url = requestPart[2].substring(0, 4).toLowerCase()
38                     + "://" + this.host
39                     + ":" + this.port
40                     + requestPart[1].substring(0, requestPart[1].length());
41                 String httpMethod = requestPart[0];
42
43                 //Show request information on the console
44                 System.out.println("\nURL : " + url + "\nHttp Method : " + httpMethod);
45                 System.out.println(">>>>>>>Request<<<<<<<");
46                 while (!line.isEmpty()) {
47                     System.out.println(line);
48                     line = reader.readLine();
49                 }
50
51                 //Rout to request handler for a thread
52                 RouterService ts;
53                 ts = new RouterService(pw, url, httpMethod);
```

```

53         ts.process();
54     }
55
56     //shutdown streaming & disconnect
57     pw.flush();
58     bos.flush();
59     clientConnection.close();
60     System.out.println("Disconnected");
61 }catch(IOException iox){
62     iox.printStackTrace();
63 }catch (Exception x) {
64     x.printStackTrace();
65 }
66
67
68 //Program start here
69 public static void main(String[] args) {
70
71     //Create tcp server
72     ServerSocket socket;
73     InetAddress serverHost;
74     int port = 6789;
75     try {
76         socket = new ServerSocket(port);
77         serverHost = InetAddress.getLocalHost();
78         System.out.println("Server Address: " + serverHost.getHostAddress());
79         System.out.println("Server Address: " + serverHost.getHostAddress()
80             + ", Port: " + socket.getLocalPort());
81
82         /* Repeatedly handle requests for processing. */
83         while (true) {
84             Socket clientCon = socket.accept();
85             Thread t = new MultithreadHttpServer(clientCon, serverHost, socket);
86             t.start();
87         }
88     }catch(IOException iox){
89         iox.printStackTrace();
90     }catch (Exception x) {
91         x.printStackTrace();
92     }
93
94 }

```

เนื่องจาก HTTP Protocol ทำงานในระดับ Application Layer จึงต้องสร้าง TCP Server ก่อน

บรรทัดที่ 69 : เริ่มต้นโปรแกรม

บรรทัดที่ 72-79 : สร้าง Socket Server และพิมพ์ข้อความบอก IP Address และ Port Address ทาง Console

บรรทัดที่ 82-86 : วงลูป โดย Server จะรอรับการเชื่อมต่อจาก Client จากนั้นก็จะแตก Thread แล้วส่งการเชื่อมต่อ นั้นไปให้ Thread นั้นๆจัดการ จากนั้นก็กลับมารอรับการขอการเชื่อมต่อจาก Client ถัดไป

บรรทัดที่ 9 : เริ่มต้น Class MultithreadHttpServer

บรรทัดที่ 11-13 : ตัวแปรของ Class ใช้ในการจัดเก็บการเชื่อมต่อและ IP Address กับ Port Address

บรรทัดที่ 15-19 : Constructor method ใช้กำหนดค่าเริ่มต้นให้กับตัวแปรของ Class (ตอนที่สร้าง Instance บรรทัดที่ 84)

บรรทัดที่ 22 : จุดเริ่มต้นการทำงานของ Thread (method run() บรรทัดที่22-66)

นาย ปฏิภาณ สมดุลยกร 5821602742, นาย พีรทัต มั่นทะกะ 5821602815 และ นาย ภาณุพงศ์ เต่าแก้ว

5821602823 Q03-3 หมู่ 700

บรรทัดที่ 25-28 : สร้างท่อสำหรับการส่งข้อมูลจากการเชื่อมต่อโดยมี Instance ของ Class PrintWriter ชื่อว่า pw ใช้ในการส่งข้อมูล และมี Instance ของ Class BufferedReader ชื่อว่า reader ใช้ในการอ่านข้อมูลที่ได้รับเข้ามา

บรรทัดที่ 31-32 : อ่านข้อความที่ส่งมาบรรทัดแรกและตรวจสอบว่าเป็นค่าว่างหรือไม่

บรรทัดที่ 35-44 : เนื่องจากบรรทัดแรกของ HTTP Request นั้นบ่งบอกถึง HTTP Method และ Path จึงนำบรรทัดแรกและ Host Address กับ Port Address มาทำการตัดต่อเพื่อสร้าง URL เพื่อให้สะดวกในการแปลความหมาย HTTP Request และนำไปใช้ในอนาคต จากนั้นแสดงผลที่ Console

บรรทัดที่ 45-48 : วนลูปเพื่ออ่านข้อความ HTTP Request บรรทัดที่เหลือแล้วแสดงข้อความออกทาง Console

บรรทัดที่ 51-53 : สร้าง Instance ของ Class RouterService เพื่อส่งต่อการทำงานของ Thread โดยส่ง Instance ของ Class PrintWriter และ String URL และ HTTP Method ไป

บรรทัดที่ 57-60 : หลังจากทำงานของ Thread เสร็จเรียบร้อยแล้วจึงปิดท่อส่งข้อมูล และปิดการเชื่อมต่อระหว่าง Client กับ Server

Class RouterService

Class นี้มีหน้าที่แปลความหมายของ HTTP request ว่า Client ต้องการอะไรแล้วทำการส่งต่อการทำงานไปให้ส่วนที่ทำหน้าที่จัดการและตอบสนองต่อ Request นั้นๆ

```
1 package httpserver;
2
3 import java.io.*;
4 import java.net.*;
5 import java.util.Date;
6
7 //import content
8 import requesthandler.IndexRequestHandler;
9 import requesthandler.PatiparnRequestHandler;
10 import requesthandler.PeeratatRequestHandler;
11
12 /**
13  * @author peeratat.m.p
14  */
15 public class RouterService {
16
17     private final PrintWriter pw;
18     private final String httpMethod;
19     private final String url;
20
21     public RouterService(PrintWriter printWriter, String url, String httpMethod) {
22         this.pw = printWriter;
23         this.url = url;
24         this.httpMethod = httpMethod;
25     }
26
27     public void process() throws Exception {
28
29         System.out.println(">>>>>>>Process<<<<<<<");
30         URL myUrl = new URL(this.url);
31         if (!myUrl.getProtocol().equalsIgnoreCase("http")) {
32             System.err.println("I only understand http.");
33             return;
34         }
35         if (httpMethod.toLowerCase().equals("get")) {
36             switch (myUrl.getPath()) {
37                 case "/":
38                     System.out.println("Response... " + myUrl.getPath());
39                     IndexRequestHandler index = new IndexRequestHandler(pw, url);
40                     index.execute();
41                     break;
42                 case "/page1":
43                     System.out.println("Response... " + myUrl.getPath());
44                     PatiparnRequestHandler pond = new PatiparnRequestHandler(pw);
45                     pond.execute();
46                     break;
47                 case "/page2":
48                     System.out.println("Response... " + myUrl.getPath());
49                     PeeratatRequestHandler pea = new PeeratatRequestHandler(pw);
50                     pea.execute();
51                     break;
52                 default:
53                     System.out.println("Response... " + myUrl.getPath());
54                     Date today = new Date();
55                     pw.println("HTTP/1.0 404 Not Found");
56                     pw.println(today.toString());
57                     pw.println("Content-Type: text/html");
58                     pw.println();
59                     pw.println("<h1>ERROR 404 Not Found!</h1>");
60             }
61         } else {
62             pw.println("HTTP/1.0 501 Not Implemented");
63             pw.println();
64         }
65     }
66 }
```

บรรทัดที่ 15 : เริ่มต้น Class RouterService

บรรทัดที่ 17-19 : ตัวแปรของ Class ใช้เก็บ PrintWriter object, HTTP Method และ URL

บรรทัดที่ 21-25 : Constructor Method ใช้กำหนดค่าเริ่มต้นให้กับตัวแปรต่างๆของ class

บรรทัดที่ 27-65 : Method process() เป็น method สำหรับการทำงานของ Class RouterService

บรรทัดที่ 30 : สร้าง Instance ของ Class URL จาก URL String เพื่อให้ใช้งาน URL ได้สะดวกขึ้น

บรรทัดที่ 31-34 : ตรวจสอบว่าเป็น HTTP Protocol หรือไม่ ถ้าไม่ให้ส่งข้อความ Error และจบการทำงานของ Method

บรรทัดที่ 35-64 : ตรวจสอบ HTTP method ของ Request ว่าเป็น GET Method หรือไม่ ถ้าไม่ให้ส่ง Response เพื่อบอกว่าไม่รองรับ Method นี้ (Status code 501 Not Implemented) ถ้าใช่ให้เข้าสู่การทำงานของ Switch-Case Statement เพื่อนำ Path จาก URL มาตรวจสอบว่า Client ต้องการอะไรแล้วจึงส่งต่อการทำงานไปให้ Class RequestHandler ที่ต้องจัดการกับ Request นั้นๆ

บรรทัดที่ 42-46 : ถ้าหากมี Request มาที่ Path เป็น page1 ก็จะส่งต่อการทำงานไปที่ Object ของ Class PatiparnRequestHnadler (ส่ง PrintWriter instance ไปให้ใช้ด้วย)

บรรทัดที่ 52-59 : ถ้ามี Request ที่มี Path ไม่ตรงกับที่กำหนดไว้ให้ทำการส่ง HTTP Response กลับไปโดย Status code เป็น 404 Not Found

Class PatiparnRequestHandler

```
1 package requesthandler;
2
3 import java.io.PrintWriter;
4
5 /**
6  *
7  * @author myhome
8  */
9 public class PatiparnRequestHandler extends RequestHandler{
10     private String content = "<!DOCTYPE html>\n" +
11 "<html>\n" +
12 "<title>MR.Patiparn</title>\n" +
13 "<meta charset=\"utf-8\">\n" +
14 "<meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">\n" +
15 "<link rel=\"stylesheet\" href=\"https://www.w3schools.com/w3css/4/w3.css\">\n" +
16 "<style>\n" +
17 "    .center {margin: auto;}\n" +
18 "    .text-center{text-align: center;}\n" +
19 "    .block{display: block;}\n" +
20 "</style>\n" +
21 "<body>\n" +
22 "\n" +
23 "<div class=\"w3-container w3-border w3-border-black w3-margin w3-round\" >\n" +
24 "    <div class=\"w3-container w3-black w3-margin w3-round\">\n" +
25 "        <h1 class=\"center text-center\">ประวัติส่วนตัวของ</h1>\n" +
26 "    </div>\n" +
```



```

27 "    <div class=\"w3-card-4 center w3-round w3-margin-bottom\" style=\"width:70%;\">\n" +
28 "        <header class=\"w3-container w3-black\">\n" +
29 "            <h2 class=\"text-center\">นายปฏิภาณ สมดุลยกัน</h2>\n" +
30 "        </header>\n" +
31 "        \n" +
32 "        <div class=\"w3-container\">\n" +
33 "            <pre>\n" +
34 "                <strong>ชื่อ</strong> ปฏิภาณ <strong>นามสกุล</strong> สมดุลยกัน\n" +
35 "                <strong>รหัสติดต่อ</strong> 5821602742\n" +
36 "                <strong>เชื้อชาติ</strong> ไทย <strong>สัญชาติ</strong> ไทย <strong>ศาสนา</strong> พุทธ\n" +
37 "                <strong>วันเกิด</strong> 8 ธันวาคม พ.ศ.2538 <strong>อายุ</strong> 21 ปี\n" +
38 "                <strong>e-mail</strong> psomdul@gmail.com\n" +
39 "                <strong>การศึกษา</strong>\n" +
40 "                <strong>ที่ผ่านมา</strong>\n" +
41 "                อเนกมล-โรงเรียนปิยะมหาราชาลัย จ.กรุงเทพมหานคร,\n" +
42 "                ประถม-โรงเรียนสาทรศึกษา จ.กรุงเทพมหานคร,\n" +
43 "                มัธยม- โรงเรียนวัฒนาวิทยาลัย จ.กรุงเทพมหานคร\n" +
44 "                <strong>ปัจจุบัน</strong>\n" +
45 "                มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน จังหวัดนครปฐม สาขาวิทยาการคอมพิวเตอร์ ภาคปกติ\n" +
46 "                <strong>ความชอบ</strong>\n" +
47 "                เกมคอมพิวเตอร์, ฟุตบอล, อาหาร, สัตว์, ท่องเที่ยว\n" +
48 "                <strong>เป้าหมายอนาคต</strong>\n" +
49 "                โปรแกรมเมอร์, อาจารย์, ท่องเที่ยวรอบโลก, กินอาหารรอบโลก\n" +
50 "            </pre>\n" +
51 "        </div>\n" +
52 "    </div>\n" +
53 "    \n" +
54 "    </div>\n" +
55 "    \n" +
56 "    </body>\n" +
57 "    </html>;
58
59
60 public PatiparnRequestHandler(PrintWriter pw){
61     super(pw);
62 }
63
64 @Override
65 public String getContent(){
66     return this.content;
67 }
68
69 @Override
70 public void execute(){
71     sendHeader();
72     pw.println(getContent());
73 }
74
75

```

บรรทัดที่ 9 : จุดเริ่มต้นของ Class PatiparnRequestHandler

บรรทัดที่ 10 : ตัวแปรชื่อ content ใช้เก็บเนื้อหาที่จะส่งให้แก่ Client

บรรทัดที่ 60-62 : Constructor Method ใช้กำหนดค่าเริ่มต้นให้กับตัวแปร PrintWriter pw

บรรทัดที่ 65-67 : Method getContent() ใช้ get ข้อมูลจากตัวแปร content

บรรทัดที่ 70-73 : Method execute() ใช้ส่งข้อความ HTTP Response โดยเรียกใช้งาน sendHeader() เพื่อส่งข้อความ

Status line : HTTP/1.0 200 OK และ Header ต่างๆ จากนั้นจะทำงานบรรทัดที่ 72 เพื่อส่ง Content

เสร็จสิ้นการสร้าง Web Server อย่างง่ายด้วยภาษา Java

ทดสอบ

Run:

Server Address: 10.40.35.163, Port: 6789

นาย ปฏิภาณ สมดุลยกัน 5821602742, นาย พีรพัฒน์ มั่นทะกะ 5821602815 และ นาย ภาณุพงศ์ เต่าแก้ว

5821602823 Q03-3 หมู่ 700


Path: /

10.40.35.163:6789

marks Facebook YouTube Twitch Gmail Google โดรน Learning goods DevAhoy : ภาตติเยน Coding Games and



พระองค์จะสถิตอยู่ในดวงใจปวงชนตลอดไป
ขอพระองค์เสด็จสู่สวรรคาลัย
ด้วยเกล้าด้วยกระหม่อมขอเดชะ



ยินดีต้อนรับเข้าสู่หน้าแรก

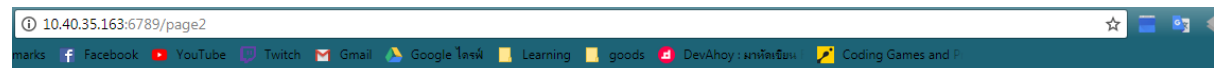
ประวัติส่วนตัว

นาย ปฏิภาณ สมดุลยภนภ

นาย พีรทัต มั่นทะกะ

นาย ปฏิภาณ สมดุลยภนภ 5821602742, นาย พีรทัต มั่นทะกะ 5821602815 และ นาย ภาณุพงศ์ เต่าแก้ว
5821602823 Q03-3 หมู่ 700

Path: /page2



ประวัติส่วนตัวของ

นาย พีรทัต มั่นทะกะ

ชื่อ พีรทัต นามสกุล มั่นทะกะ

รหัสบัตร 5821602815

เชื้อชาติ ไทย สัญชาติ ไทย ศาสนา พุทธ

วันเกิด 5 กรกฎาคม พ.ศ.2538 อายุ 22 ปี

e-mail peeratat1995@gmail.com

การศึกษา

ที่ผ่านมา

โรงเรียนวัดแจ้ง จ.สุราษฎร์ธานี,

โรงเรียนคลองเกลือ จ.นนทบุรี,

โรงเรียนปากเกร็ด จ.นนทบุรี

ปัจจุบัน

มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตกำแพงแสน จังหวัดนครปฐม สาขาวิทยาการคอมพิวเตอร์ ภาคปกติ

ความชอบ

สีเขียว, สีส้ม, สีฟ้า, เมนูอาหารจากไข่, อาหารทะเล

เป้าหมายอนาคต

เที่ยว, software developer, software engineer

นาย ปฏิภาณ สมดุลยภณ 5821602742, นาย พีรทัต มั่นทะกะ 5821602815 และ นาย ภาณุพงศ์ เต่าแก้ว
5821602823 Q03-3 หมู่ 700