

LAB2 HTTP and HTML Basic

สิ่งที่นักศึกษาต้องส่ง

1. รายงานผลการทดลองพร้อมคำตอบสำหรับคำถามในแต่ละส่วน
 2. รูปภาพหรือสกรีนช็อตจาก Terminal ที่แสดงผลลัพธ์คำสั่งที่ใช้งาน
-

PART 1 Domain and DNS

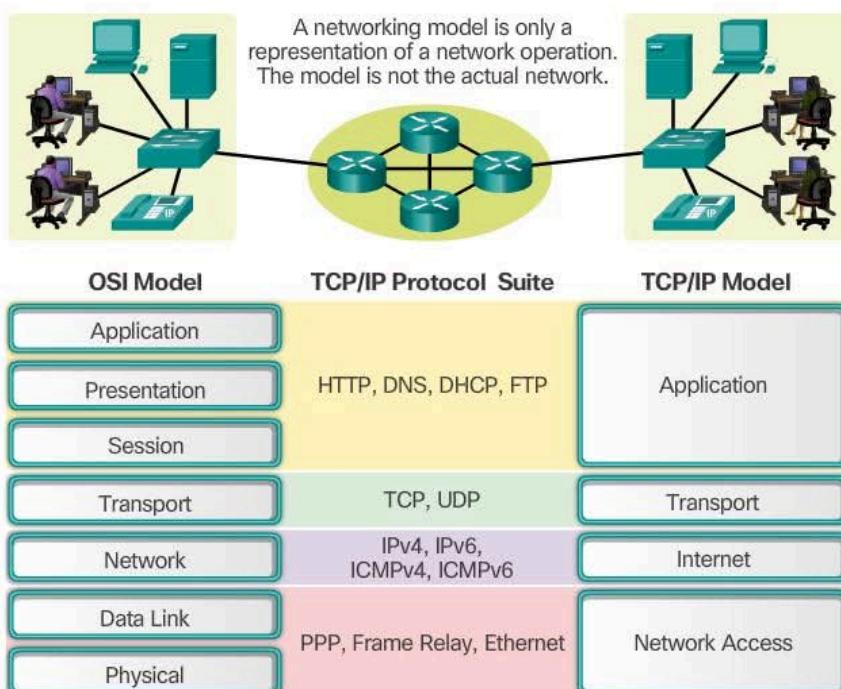
วัตถุประสงค์

1. เพื่อเข้าใจการทำงานของระบบ DNS (Domain Name System)
2. เพื่อฝึกใช้งานคำสั่งพื้นฐานที่เกี่ยวข้องกับ DNS บน macOS
3. เพื่อเรียนรู้วิธีการตรวจสอบและแก้ไขการตั้งค่า DNS

ขั้นตอนการเตรียมความพร้อม

1. เตรียมเครื่อง Mac ที่ใช้ macOS
 - macOS เวอร์ชันล่าสุดที่รองรับคำสั่ง `dig`, `nslookup`, และ `ping`
2. ติดตั้งเครื่องมือเสริม (ถ้าจำเป็น)
 - ใช้ Terminal ซึ่งติดตั้งมาใน macOS
 - หากยังไม่มีเครื่องมือ Developer Tools ให้ติดตั้งผ่านคำสั่ง

`xcode-select --install`

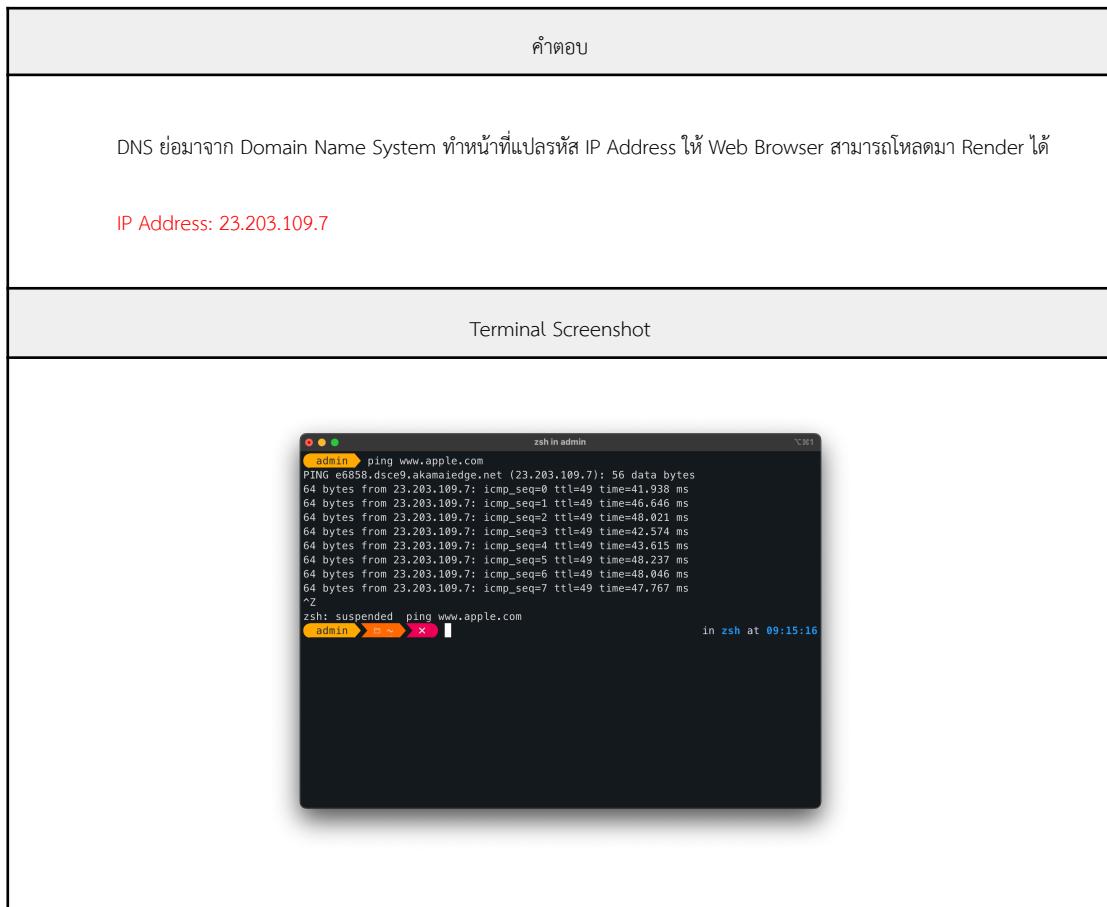


เนื้อหากิจกรรมใน Lab

ส่วนที่ 1: การทดสอบการแปลงชื่อโดเมน

1. เปิด Terminal บน macOS
2. ใช้คำสั่ง **ping** เพื่อตรวจสอบการแปลงชื่อโดเมนเป็น IP Address:
ping www.apple.com

คำถาม: จดบันทึก IP Address ที่ได้รับและอธิบายการทำงานของ DNS



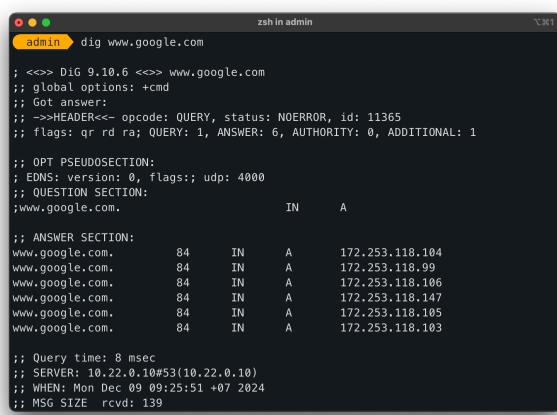
ส่วนที่ 2: ใช้คำสั่ง **dig** ตรวจสอบการทำงานของ DNS

ใช้คำสั่ง **dig** เพื่อตั้งข้อมูล DNS สำหรับโดเมน:

dig www.google.com

คำถาม: ระบุค่า **ANSWER SECTION** และ **AUTHORITY SECTION**

อธิบายว่าแต่ละส่วนหมายถึงอะไร

คำตอบ
Authority: 0
Answer: 6 ไฉไลๆ
www.google.com. 84 IN A 172.253.118.104 www.google.com. 84 IN A 172.253.118.99 www.google.com. 84 IN A 172.253.118.106 www.google.com. 84 IN A 172.253.118.147 www.google.com. 84 IN A 172.253.118.105 www.google.com. 84 IN A 172.253.118.103
Terminal Screenshot
 <pre> zsh in admin admin: dig www.google.com ; <>> DiG 9.10.6 <>> www.google.com ;; global options: +cmd ;; Got answer: ;; ->>HEADER:<- opcode: QUERY, status: NOERROR, id: 11365 ;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ;; EDNS: version: 0, flags:; udp: 4000 ;; QUESTION SECTION: ;www.google.com. IN A ;; ANSWER SECTION: www.google.com. 84 IN A 172.253.118.104 www.google.com. 84 IN A 172.253.118.99 www.google.com. 84 IN A 172.253.118.106 www.google.com. 84 IN A 172.253.118.147 www.google.com. 84 IN A 172.253.118.105 www.google.com. 84 IN A 172.253.118.103 ;; Query time: 8 msec ;; SERVER: 10.22.0.10#53(10.22.0.10) ;; WHEN: Mon Dec 09 09:25:51 +07 2024 ;; MSG SIZE rcvd: 139 </pre>

ส่วนที่ 3: ใช้คำสั่ง **nslookup** ตรวจสอบชื่อโดเมน

ทดลองใช้คำสั่ง **nslookup**:

nslookup www.microsoft.com

คำถาม:

1. เปรียบเทียบข้อมูลที่ได้จาก **nslookup** และ **dig**
2. ระบุว่า DNS Server ที่ใช้งานคืออะไร

ENGSE611 การพัฒนาเว็บด้วยเทคโนโลยีสมัยใหม่

คำตอบ
<ol style="list-style-type: none">1. nslookup และ dig ให้ข้อมูล DNS เมื่อก่อนกันที่ต่างกันคือ ns look up ให้ข้อมูลแบบรับรู้ สามารถนำไปใช้ต่อได้ ง่าย แต่ DIG ให้ข้อมูลแบบเขียนอิบๆ2. DNS Server ที่ใช้<ol style="list-style-type: none">a. Name: e13678.dscc.akamaiedge.netb. Address: 184.25.221.172
Terminal Screenshot

The screenshot shows two terminal windows side-by-side. The left window displays the output of the nslookup command for 'www.microsoft.com'. It shows multiple servers tried and eventually failed with a 'SERVFAIL' error. The right window displays the output of the dig command for the same domain. It shows a successful query with various resource records (RR) including A, CNAME, and NS types, along with their respective values and TTLs.

```
nslookup www.microsoft.com
; Got SERVFAIL reply from 10.22.0.10, trying next server
; Got SERVFAIL reply from 10.8.1.38, trying next server
Servers:  10.8.1.248
Address:  10.8.1.248#53
** server can't find nslookup: SERVFAIL

dig www.microsoft.com
; <>> DIG 9.16.6 <>> www.microsoft.com
; global options: +cmd
; Got answer:
; ->>>HEADER: opcode: QUERY, status: NOERROR, id: 31189
; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1
;
; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: udp: 4000
; QUESTION SECTION:
; www.microsoft.com.           IN      A
;
; ANSWER SECTION:
www.microsoft.com.  2744   IN      CNAME   www.microsoft.com-c-3.edgekey.net
www.microsoft.com-c-3.edgekey.net. 529 IN CNAME www.microsoft.com-c-3.edgekey.net
www.microsoft.com-c-3.edgekey.net.globalredir.akadns.net. 316 IN CNAME e13678.dscc.akamaiedge.net
e13678.dscc.akamaiedge.net. 16 IN      A       184.25.221.172

; Query time: 7 msec
; SERVER: 10.22.0.10#53(10.22.0.10)
; WHEN: Mon Dec 09 09:35:32 +07 2024
```

ส่วนที่ 4: ให้เปลี่ยน WiFi ที่ใช้งานจากของมหาวิทยาลัย เป็น WiFi Hotspot ของมือถือของตัวเอง
หรือ WiFi อื่น ที่ไม่ใช่ของ มหาวิทยาลัย และทำการทดสอบในส่วนที่ 1-3 ใหม่อีกครั้ง บันทึกผล

การทดลอง

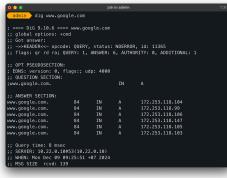
คำถาม:

1. มีสิ่งใดเปลี่ยนแปลงไปบ้างและให้เหตุผลประกอบ
2. กรณีเราใช้คำสั่ง **ping** และ **nslookup** กับ IP Address จะมีผลจะเป็นอย่างไร

ENGSE611การพัฒนาเว็บด้วยเทคโนโลยีสมัยใหม่

คำตอบ
<p>1. เปลี่ยนทั้ง server และ Ip address ที่เปลี่ยนอาจจะมีผลมาจากสองอย่างคือ</p> <ul style="list-style-type: none">a. เป็น Dynamic IP Assignment หรือ ISPs ที่จะทำการเปลี่ยน IP ครบทองเวลาที่เราทำการ request เพื่อความปลอดภัย หรือb. DNS Resolution เป็นการกำหนด IP address ตามขนาดของเครือข่าย เนื่องจาก internet ของผู้ใช้งาน มีขนาดไม่เท่ากัน server จึงมีการเปลี่ยนขนาดของ DNS ให้เหมาะสม <p>2. ping จะได้เป็นคำตอบที่ได้จากการ ping domain name ปกติ ส่วน nslookup ก็สามารถทำได้เช่นเดียวกันแต่จะได้ เป็นคำตอบแบบไม่ได้รับการอนุญาตโดยสมบูรณ์</p>

ENGSE611 การพัฒนาเว็บด้วยเทคโนโลยีสมัยใหม่

ping www.apple.com	
	
dig www.google.com	
	
nslookup www.microsoft.com	
	

PART 2 HTTP Communication**วัตถุประสงค์**

1. เพื่อเข้าใจการทำงานของ HTTP Request แบบ GET
2. เพื่อเข้าใจการทำงานของ HTTP Request แบบ POST
3. เพื่อเข้าใจการทำงานของ HTTP Request และ Response

เนื้อหากิจกรรมใน Lab

ส่วนที่ 1: สร้างตัวดักจับ http request ที่ <https://requestcatcher.com/> โดยทำการใส่ “รหัสประจำตัวนศ.” เข้าไปในช่อง และกด Get started



Debug web hooks, http clients, etc. <https://12345.requestcatcher.com>

Request Catcher will create a subdomain on which you can test an application. All requests sent to any path on the subdomain are forwarded to your browser in real time.

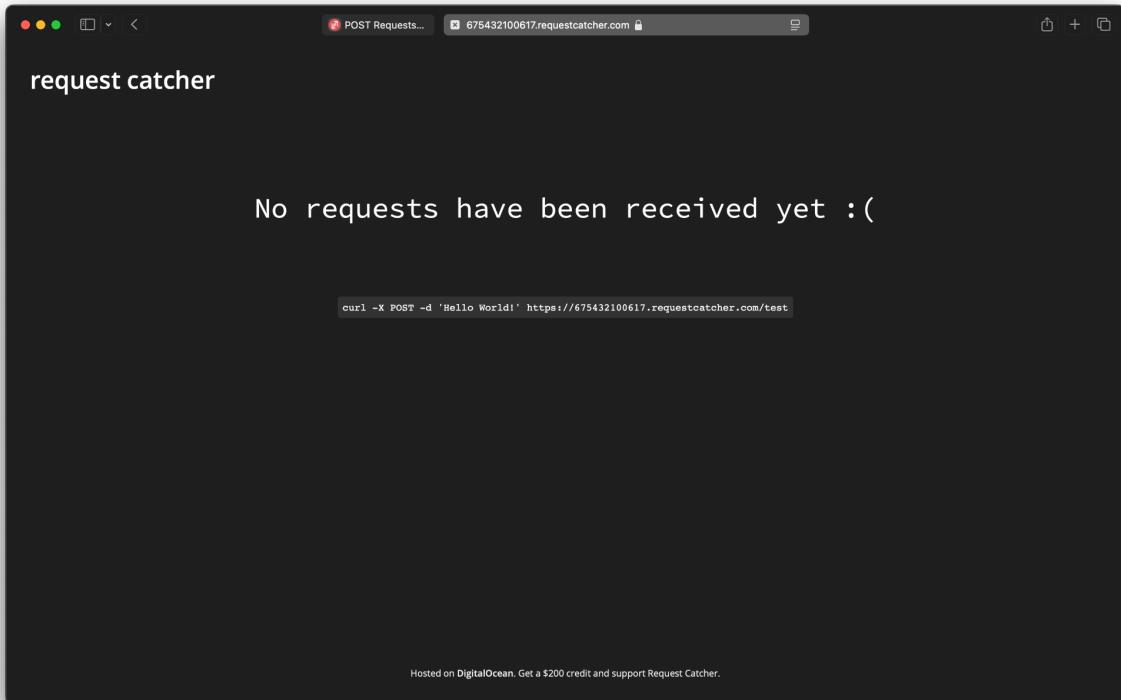
[Get started](#)

จากนั้นจะได้หน้า web ดังรูป และให้เปิดทิ้งไว้ เวลาทำการทดลอง



No requests have been received yet :(

```
curl -X POST -d 'Hello World!' https://12345.requestcatcher.com/test
```



ส่วนที่ 2: HTTP Request แบบ GET

- ให้เข้า web ในลิ้งนี้ อ่านให้เข้าใจและ กด Send

Sending HTTP GET Request

<https://reqbin.com/req/nfilsyk5/get-request-example>

REQBIN

HTTP Curl Python JavaScript PHP JSON XML Contact Account

SEND REQUEST

Sending HTTP GET Request

```
GET /echo HTTP/1.1
Host: reqbin.com
Accept: */*
```

Updated: Jul 01, 2023 Viewed: 79585 times Author: ReqBin

What is HTTP?

The Hypertext Transfer Protocol (HTTP) is the core of the World Wide Web and powers websites and mobile

จากนั้นให้เลือก GET ตามรูป

POST Requests Online

Post requests to the server and check server responses. Fully Online; no desktop app needed. Test Server endpoints by sending HTTP POST, GET, PUT, and HEAD requests directly from your browser. Create PHP, Python, Java, Curl, and JavaScript code snippets from your requests with one click. Share your HTTP requests online, showcase your work, or discuss with colleagues and friends.

The screenshot shows the ReqBin online testing tool interface. At the top, there's a URL input field containing <https://reqbin.com/echo>. To the right of the URL is a dropdown menu with options: GET (which is checked), POST, PUT, PATCH, DELETE, HEAD, and OPTIONS. Below the URL input are tabs for Params, Body, Auth, Headers, Message, and Timings. The Params tab is selected, showing a table titled "Query Params" with three rows. Each row has a checkbox labeled "Key" and a column labeled "Value" containing "value". Under the Body tab, the response is displayed as an HTML document:

```
<html>
<head>
  <title>ReqBin Echo</title>
  <meta name="description" content="ReqBin Echo Interface">
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width">
  <link rel="shortcut icon" href="/static/favicon.ico">
  <style>
    body {font-size: 1.5rem;} li {margin: 0.3rem;}
  </style>
```

At the bottom right of the interface, there are icons for copy, share, and refresh. Above the response, there's a status bar showing 200 (OK), 27 ms, and 0.63 kb.

2. จากนั้นให้ทำการ copy web link ที่ได้จากส่วนที่ 1 มาใส่ในช่องดังรูป กด Send
ให้ทดสอบไปสู่ Query Parameter สังเกตผลลัพธ์ บันทึกผล

ENGSE611 การพัฒนาเว็บด้วยเทคโนโลยีสมัยใหม่

POST Requests Online

Post requests to the server and check server responses. Fully Online; no desktop app needed. Test Server endpoints by sending HTTP POST, GET, PUT, and HEAD requests directly from your browser. Create PHP, Python, Java, Curl, and JavaScript code snippets from your requests with one click. Share your HTTP requests online, showcase your work, or discuss with colleagues and friends.

The screenshot shows the RequestCatcher interface. At the top, there's a URL input field with "https://12345.requestcatcher.com/" and a "Send" button. Below it is a "Request" section with tabs for "Params", "Body", "Auth", "Headers", "Message", and "Timings". Under "Params", there are three entries: "key" with "value", "key" with "value", and "key" with "value". The "Response" section shows a status of "200 (OK)" with a response time of "225 ms" and a size of "1.51 kb". The response body contains an HTML page with various scripts and styles.

This screenshot shows the Response section of RequestCatcher. It displays a status of "200 (OK)", a response time of "225 ms", and a size of "1.51 kb". The response body is an HTML document with several script and style tags.

ตัวอย่างผลลัพธ์

The screenshot shows the ReqBin interface. On the left, there's a browser window showing a captured POST request to "https://12345.requestcatcher.com/?user=thanit&pass=1234". The request details show the method as "POST", the URL as "/?user=thanit&pass=1234", and the timestamp as "2024-12-09T08:55:01+07:00". The request body is "64.227.21.251". On the right, there's another browser window titled "reqbin.com/post-online" showing the "POST Requests Online" interface. It has a "Params" tab selected with a "user" parameter set to "thanit" and a "pass" parameter set to "1234". The response body is identical to the one shown in the first screenshot.

ENGSE611 การพัฒนาเว็บด้วยเทคโนโลยีสมัยใหม่

The screenshot shows the ReqBin website at <https://reqbin.com/post-online>. The interface includes a sidebar with various examples like GET Bearer Token Auth Header, REST API POST Example, and POST JSON Example. The main area has tabs for HTTP, Curl, Python, JavaScript, PHP, JSON, and XML. It shows a URL input field with <https://reqbin.com/echo>, a dropdown for method (GET), and a 'Send' button. Below these are sections for Params, Body, Auth, Headers, and Message. The Body section contains a text area with key-value pairs: key1=value1 and key2=value2. At the bottom, there are tabs for Body (selected), Headers, Message, and Timings. The status bar shows 200 OK, 56 ms, and 1.54 kb. The response body shows an HTML page with a title 'ReqBin Echo' and a message about cookie consent.

The screenshot shows the RequestCatcher tool at <https://12345.requestcatcher.com/?user=thanit&pass=1234>. It displays a table for Query Params with columns for Key, Value, and checked checkboxes. The table rows are: user (thanit), pass (1234), and key (value). There are also tabs for Params, Body, Auth, Headers, and Message.

ส่วนที่ 3: HTTP Request แบบ POST

- ให้เข้า web ในลิ้งนี้ อ่านให้เข้าใจและ กด Send

Sending POST Request

<https://reqbin.com/req/zvtstmpb/post-request-example>

The screenshot shows the ReqBin website at reqbin.com/req/zvtstmpb/post-request-example. The main content area is titled "Sending POST Request". It contains a code snippet for a POST request to the "/echo/post/json" endpoint:

```
POST /echo/post/json HTTP/1.1
Host: reqbin.com
Content-Type: application/json
Content-Length: 80

{
    "Id": 12345,
    "Customer": "John Smith",
    "Quantity": 1,
    "Product": "Laptop"
}
```

Below the code, it says "Updated: Jan 13, 2023 Viewed: 70704 times Author: ReqBin". The top navigation bar includes links for HTTP, Curl, Python, JavaScript, PHP, JSON, XML, Contact, and Account.

จากนั้นให้เลือก POST ตามรูป

POST Requests Online

Post requests to the server and check server responses. Fully Online; no desktop app needed. Test Server endpoints by sending HTTP POST, GET, PUT, and HEAD requests directly from your browser. Create PHP, Python, Java, Curl, and JavaScript code snippets from your requests with one click. Share your HTTP requests online, showcase your work, or discuss with colleagues and friends.

The screenshot shows an online tool for making HTTP requests. The URL is <https://reqbin.com/echo>. A dropdown menu on the right lists methods: GET, POST (which is selected), PUT, PATCH, DELETE, HEAD, and OPTIONS. Below the URL, there are tabs for Params, Body, Auth, Headers, and Message. The Body tab is active, showing a table with three rows of query parameters:

Key	Value
key	value
key	value
key	value

The screenshot shows the response details for the POST request. At the top, it says "Body 27 Headers 16 Message 19 Timings". On the right, it shows the status "200 (OK)" with "27 ms" and "0.63 kb". The "Body" tab is active, displaying the HTML response:

```
<html>
<head>
    <title>ReqBin Echo</title>
    <meta name="description" content="ReqBin Echo Interface">
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width">
    <link rel="shortcut icon" href="/static/favicon.ico">
    <style>
        body {font-size: 1.5rem;} li {margin: 0.3rem;}
    </style>

```

2. จากนั้นให้ทำการ copy web link ที่ได้จากส่วนที่ 1 มาใส่ในช่องตั้งรูป กด Send
สังเกตผลลัพธ์ บันทึกผล

POST Requests Online

Post requests to the server and check server responses. Fully Online; no desktop app needed. Test Server endpoints by sending HTTP POST, GET, PUT, and HEAD requests directly from your browser. Create PHP, Python, Java, Curl, and JavaScript code snippets from your requests with one click. Share your HTTP requests online, showcase your work, or discuss with colleagues and friends.

The screenshot shows the Requestcatcher interface with a red box highlighting the 'Body' tab. The URL in the address bar is <https://12345.requestcatcher.com/>. The 'Body' tab contains the following raw HTTP request:

```
POST / HTTP/1.1
Accept-Encoding: deflate, gzip
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
Host: 12345.requestcatcher.com
Accept: application/json
Content-Type: application/json
Content-Length: 0
```

The screenshot shows the Requestcatcher interface with the 'Response' tab selected. The status is 200 (OK). The response body is a raw HTML document:

```
<html>
<head>
  <title>request catcher</title>
  <meta content="dark light" name="color-scheme">
  <link href="https://fonts.googleapis.com/css?family=Ubuntu+Mono|Source+Code+Pro|Cousine|Open+Sans:400,300,600" rel="stylesheet">
  <link rel="stylesheet" href="assets/static.d40603e4.css?v=0.2">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.24.0/moment.min.js"></script>
  <script type="text/javascript" src="assets/catcher_595c1f61.js"></script>
```

3. จากนั้น กดไปที่ body ให้เลือก Form (url-encoded) ให้ป้อนข้อมูลที่จะส่งตามตัวอย่างที่แสดงไว้ให้ กด Send สังเกตผลลัพธ์ บันทึกผล

POST Requests Online

Post requests to the server and check server responses. Fully Online; no desktop app needed. Test Server endpoints by sending HTTP POST, GET, PUT, and HEAD requests directly from your browser. Create PHP, Python, Java, Curl, and JavaScript code snippets from your requests with one click. Share your HTTP requests online, showcase your work, or discuss with colleagues and friends.

The screenshot shows the Requestcatcher interface with the 'Body' tab selected. The 'Form (url-encoded)' option is selected. The body field contains the following key-value pairs:

```
key1=value1
key2=value2
```

ตัวอย่างผลลัพธ์

ENGSE611 การพัฒนาเว็บด้วยเทคโนโลยีสมัยใหม่

The screenshot shows two side-by-side browser windows. The left window is titled 'request catcher' and displays a log of requests. The right window is titled 'reQBin.com/post-online' and shows a POST request interface.

request catcher Log:

- POST /?**
2024-12-09T08:55:53+07:00
64.227.21.251
- GET /?user=thanit&pass=1234**
2024-12-09T08:55:01+07:00
64.227.21.251

reQBin POST Requests Online:

- URL: https://12345.requestcatcher.com
- Method: POST
- Body:
user=thanit
pass=1234
- Headers:
 - Content-Type: application/x-www-form-urlencoded
 - User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6231.122 Safari/537.36
 - X-Real-Ip: 2405:9800:b910:2d34:4c9e:843d:375d:2c05
- Message: 200 OK 237 ms 1.01 kB

ให้ดูข้อมูลในแต่ละ TAB และ อธิบายเพื่อไม่เติม

Screenshot of the reQBin POST Requests Online interface. A specific POST request is selected:

Params **Body 2** **Auth** **Headers 6** **Message 9**

None JSON Form (url-encoded) XML Custom

Body

```
user=thanit
pass=1234
```

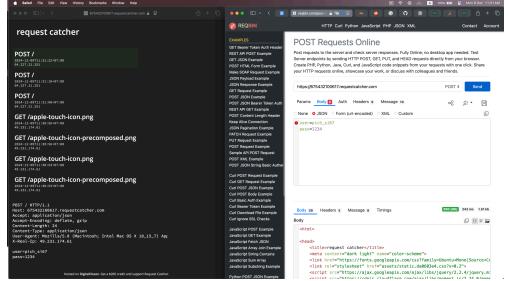
Screenshot of the reQBin interface showing the Body tab. It displays the raw HTML code of the response:

Body 39 **Headers 5** **Message 8** **Timings**

Body

```
<html>
<head>
  <title>request catcher</title>
  <meta content="dark light" name="color-scheme">
  <link href="https://fonts.googleapis.com/css?family=Ubuntu:400,700" rel="stylesheet">
  <link rel="stylesheet" href="assets/static.da0603e4.css?ver=4.2.12" type="text/css" media="all" />
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.24.0/moment.min.js"></script>
  <script type="text/javascript" src="assets/catcher.596cf1fa1.js?ver=4.2.12" />
</head>
```

ENGSE611 การพัฒนาเว็บด้วยเทคโนโลยีสมัยใหม่

Screenshot																	
																	
Screenshot	อธิบาย																
<p>Body 39 Headers 5 Message 8 Timings 200 (OK) 242 ms 1.51 kb</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Key</th><th>Value</th></tr> </thead> <tbody> <tr><td>Accept-Ranges</td><td>bytes</td></tr> <tr><td>Content-Length</td><td>1549</td></tr> <tr><td>Content-Type</td><td>text/html; charset=utf-8</td></tr> <tr><td>Last-Modified</td><td>Sat, 05 Oct 2024 14:46:41 GMT</td></tr> <tr><td>Date</td><td>Mon, 09 Dec 2024 04:31:22 GMT</td></tr> </tbody> </table>	Key	Value	Accept-Ranges	bytes	Content-Length	1549	Content-Type	text/html; charset=utf-8	Last-Modified	Sat, 05 Oct 2024 14:46:41 GMT	Date	Mon, 09 Dec 2024 04:31:22 GMT	Headers: เป็นส่วนที่แสดงข้อมูลรายละเอียด URL ที่มาจาก API				
Key	Value																
Accept-Ranges	bytes																
Content-Length	1549																
Content-Type	text/html; charset=utf-8																
Last-Modified	Sat, 05 Oct 2024 14:46:41 GMT																
Date	Mon, 09 Dec 2024 04:31:22 GMT																
<p>Body 39 Headers 5 Message 8 Timings 200 (OK) 242 ms 1.51 kb</p> <p>Body</p> <pre><head> <title>request catcher</title> <meta content="dark light" name="color-scheme"> <link href="https://fonts.googleapis.com/css?family=Ubuntu+Mono Source+Code+Pro" rel="stylesheet" type="text/css"> <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js"> <script src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.24.0/moment.min.js"> <script type="text/javascript" src="assets/catcher.595c1f61.js"></script> </head></pre>	ส่วน Body เป็นส่วนที่ได้มีการแปลงข้อมูลแล้ว ออกมารูปสิ่งที่สามารถแสดงผลได้ ในที่นี่คือ HTML																
<p>Body 39 Headers 5 Message 8 Timings 200 (OK) 242 ms 1.51 kb</p> <p>HTTP Message</p> <pre>HTTP/1.1 200 OK Accept-Ranges: bytes Content-Length: 1549 Content-Type: text/html; charset=utf-8 Last-Modified: Sat, 05 Oct 2024 14:46:41 GMT Date: Mon, 09 Dec 2024 04:31:22 GMT <html><head><title>request catcher</title><meta content="dark light" name="color-scheme"></head></pre>	Message เป็นการแสดง ข้อความต่างๆที่ส่งมาพร้อมกับ URL นี้ จากการจะเป็นข้อความของ HTTP ที่แสดงสถานะของ HTTP																
<p>Body 39 Headers 5 Message 8 Timings 200 (OK) 242 ms 1.51 kb</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Process</th><th>Time (ms)</th></tr> </thead> <tbody> <tr><td>DNS Resolution:</td><td>14 ms</td></tr> <tr><td>Connecting:</td><td>74 ms</td></tr> <tr><td>TLS Setup:</td><td>88 ms</td></tr> <tr><td>Sending:</td><td>0 ms</td></tr> <tr><td>Waiting:</td><td>74 ms</td></tr> <tr><td>Receiving:</td><td>0 ms</td></tr> <tr><td>Total:</td><td>242 ms</td></tr> </tbody> </table>	Process	Time (ms)	DNS Resolution:	14 ms	Connecting:	74 ms	TLS Setup:	88 ms	Sending:	0 ms	Waiting:	74 ms	Receiving:	0 ms	Total:	242 ms	Timing จะเป็นการบอกเวลาของ Process ที่เกิดขึ้นทั้งหมดตั้งแต่การแปลงค่าของ DNS ถึงการรับค่าอุปกรณ์แสดงผล ซึ่งจะเกิดขึ้นไวมากๆ
Process	Time (ms)																
DNS Resolution:	14 ms																
Connecting:	74 ms																
TLS Setup:	88 ms																
Sending:	0 ms																
Waiting:	74 ms																
Receiving:	0 ms																
Total:	242 ms																

PART 3 HTML Basic**วัตถุประสงค์**

1. ให้ฝึกใช้ github และการสร้าง repository
2. เพื่อฝึกการเขียนภาษา html และใช้งาน html tag เป็นต้น
3. เพื่อทดสอบความเข้าใจการทำงานของ HTTP Request และ Response ที่ตนเองได้สร้างขึ้นด้วยเครื่องมือใน VS Code

เนื้อหากิจกรรมใน Lab**ส่วนที่ 1:** สร้าง repository

1. ให้นักศึกษาสร้าง repository ของวิชา โดยใช้ชื่อว่า engse611 และ public และต้องเพิ่มไฟล์ README.md ตอนที่สร้างด้วย
2. ให้ทำการ clone ลงมาทำงานที่ Local Mac/Notebook แบบ SSH
3. เปิด VS Code ที่ Local Mac/Notebook ณ directory engse611 จากนั้นให้สร้าง folder ใหม่ ชื่อ lab2

ส่วนที่ 2: สร้างและพัฒนา html file

1. ที่ directory lab2 ให้นักศึกษาสร้างไฟล์ชื่อ index1.html และเขียน html code โดยทำการรวม html tag element ในหัวข้อที่ 05. พื้นฐาน HTML ของ Lecture02 เข้าด้วยกันเป็นไฟล์เดียว และจัด format code สวายงาม
2. ที่ directory lab2 ให้นักศึกษาสร้างไฟล์ html และเขียน html code เพื่อให้ได้ผลลัพธ์ ตาม Lab Practical 1 ถึง 7 ดังต่อไปนี้

Lab Practical	Html files
Lab Practical 1	index2-1.html
Lab Practical 2	index2-2.html
Lab Practical 3	index2-3.html
Lab Practical 4	index2-4.html
Lab Practical 5	index2-5.html
Lab Practical 6	index2-6.html
Lab Practical 7	index2-7.html

3. เมื่อทำเสร็จทั้งหมด commit และ push code ใน Lab2 ขึ้น git และแนบลิงค์ git repository ของ Lab2 ในเอกสารส่งงานด้วย

Hello World!

This is a simple paragraph.

Lab Practical 2: Using **Headings** and **paragraphs**

Heading Level 1

This is a paragraph of text.

Heading Level 2

Another paragraph goes here.

Heading Level 3

And yet another paragraph.

Heading Level 4

One more paragraph for this heading.

Heading Level 5

Another small paragraph.

Heading Level 6

The last paragraph on this page.

Lab Practical 3: Create html page using **headings**, **paragraphs** and text formatting tags within the paragraph

Welcome to Our Website

Somthing goes here **This is a strong text.** some other thing goes here *This is emphasized text.* something goes here _{subscript} et ^{superscript} end.

About Us

We are a company dedicated to providing high-quality WWW solutions.

Our Services

Web Design: Creating beautiful and responsive websites.

Content Creation: Crafting engaging and informative content.

SEO: Optimizing websites for search engines.

Contact Information

Email: info@example.com

Phone: +1 (123) 456-7890

Website: www.example.com

Lab Practical 4: Create HTML page using tags such as mark, bold, italic, underline, abbreviation, inserted, deleted tags

Text Formatting Examples

This is a **highlighted** text. You can use the **mark** element to highlight important words or phrases.

This is bold text. You can use the **strong** element to make text bold.

This is italicized text. You can use the *em* element to italicize text.

This is underlined text. You can use the u element to underline text.

HTML is an abbreviation. You can use the abbr element with a "title" attribute to define abbreviations.

This is inserted text. You can use the ins element to indicate text that has been inserted into a document.

~~This is deleted text.~~ You can use the ~~del~~ element to indicate text that has been removed or deleted from a document.

Lab Practical 5: Create HTML page using anchor and hyperlink tags

Welcome to Our Website

Welcome to our website! Here are some important links to various sections of our site:

[About Us](#)

[Our Services](#)

[Contact Us](#)

About Us

We are a company dedicated to providing high-quality services to our clients.

[Contact us](#) to learn more.

Our Services

Explore our wide range of services:

[Web Design](#)

[Content Creation](#)

[SEO Services](#).

Contact Us

You can reach us via email at info@example.com or by phone at +1 (123) 456-7890.

Welcome to Our Website

Welcome to our website! Here's some information about our services and offerings:

About Us

We are a company dedicated to providing a wide range of services to our clients. Our key offerings include:

- Web Design
- Content Creation
- SEO Services

Our Team

Meet our talented team members who work diligently to deliver the best results:

- **John Doe**
Web Designer
- **Jane Smith**
Content Creator
- **David Johnson**
SEO Specialist

Contact Us

If you have any questions or need assistance, feel free to reach out to us:

- Email: info@example.com
- Phone: +1 (123) 456-7890

Welcome to Our Website

Welcome to our website! Here's some information about our services and offerings:

About Us

We are a company dedicated to providing a wide range of services to our clients. Our key offerings include:

Web Design

Creating stunning and responsive websites.

Content Creation

Developing engaging and informative content.

SEO Services

Optimizing websites for search engines.

Our Team

Meet our talented team members who work diligently to deliver the best results:

John Doe

Web Designer

Jane Smith

Content Creator

David Johnson

SEO Specialist

Contact Us

If you have any questions or need assistance, feel free to reach out to us:

Email

info@example.com

Phone

+1 (123) 456-7890

หมายเหตุ: เมื่อทำเสร็จทั้งหมดให้ commit และ push code ใน Lab2 ขึ้น git และแนบลิงค์ git repository ของ Lab2 ในเอกสารส่งงานด้วย

<https://github.com/ltsPitxh/engse611/tree/main/lab2>

End of Lab2