

Node.js (3)

ผศ. ดร. เก็จแก้ว ธเนศวร

kejkaew.tha@mail.kmutt.ac.th

ตารางเรียน

สัปดาห์	วันที่	หัวข้อ/รายละเอียด	การบ้าน (ผู้ตรวจ)	ส่ง Assignment
1	8/8/2566	Introduction to Web programming, Basic HTML 1	Assignment 1 (KT,KN)	1
2	15/8/2566	Basic HTML 2 + CSS 1	Assignment 2 (KT,KN)	1,2
3	22/8/2566	CSS 2	Assignment 3 (KT)	1,2,3
4	29/8/2566	Responsive Web design	Assignment 4 (KT,KN)	2,3,4
5	5/9/2566	JavaScript 1 + DOM	Assignment 5 (KT,KN)	3,4,5
6	12/9/2566	JavaScript 2 + GitHub	Assignment 6 (KT)	4,5,6
7	19/9/2566	JavaScript 3	Assignment 7 (KT,KN)	5,6,7
8	26/9/2566	AJAX + JSON	Assignment 8 (KT,KN)	6,7,8
9	3/10/2566	Node.js 1	Assignment 9 (KT)	6,7,8,9
	10/10/2566	No class	ส่ง portfolio web page (KT)	
10	17/10/2566	Node.js 2	Assignment 10 (KT,KN)	8,9,10
11	24/10/2566			8,9,10
12	31/10/2566	Node.js 3	Assignment 11 (KT,KN)	9,10,11
13	7/11/2566	Node.js 4 + MySQL	Assignment 12 (KT)	10,11,12
14	14/11/2566		online	10,11,12
15	21/11/2566	ส่งความก้าวหน้าโปรเจค	online	12
16	28/11/2566	ส่งโปรเจค (9:00-16:00)	KT (ห้องเรียน)	
		สอบปลายภาค ในตาราง	KT	

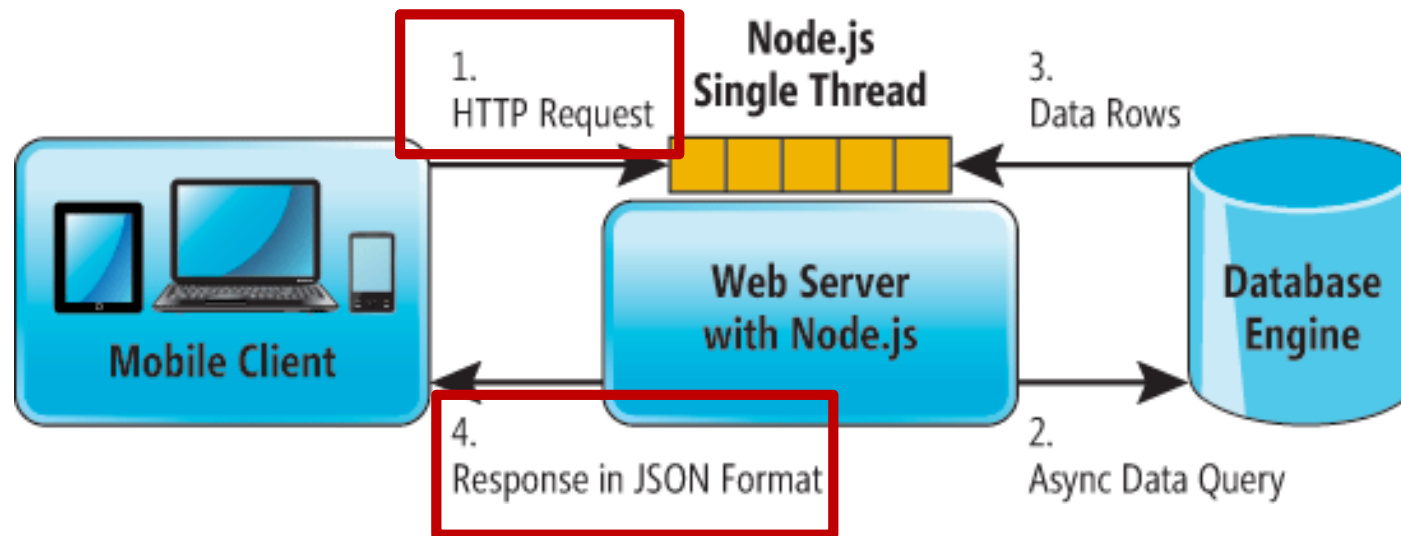
เรื่องที่เรียนไปสัปดาห์ที่แล้ว

- Express module
- Fetch API
- Body-Parser
- Async / Await
- Assignment 10

วันนี้เราจะเรียนเรื่อง

- HTML Form and Express.js
- Cookies
- Upload Image File
- Assignment 11: Login

ทบทวน



- Client – Server relationship

HTML Form and Express.js

- ใช้สำหรับเก็บข้อมูลของ user เพื่อส่งไปให้ server
- สามารถทำงานร่วมกับ Routing ของ express module ได้
- สามารถเข้าถึง ข้อมูลในแต่ละ field ใน form ได้โดยผ่านทาง name attribute ของ field นั้น

ตัวอย่าง HTML Form

```
<form class="form" action="/showname" method="post"
accept-charset="utf-8">
  <fieldset>
    <legend> Enter Your Name </legend>
    Name:<input type="text" name="name" value=""
placeholder="" ><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

Enter Your Name

Name:

ตัวอย่าง

Enter Your Name

Name:

Output: html

your name is:kejkaew

```
1  const express = require('express');
2  const app = express();
3  const hostname = 'localhost';
4  const port = 3001;
5  const bodyParser = require('body-parser');
6
7  app.use(express.static(__dirname));
8  app.use(bodyParser.json());
9  app.use(bodyParser.urlencoded({extended: false}));
10
11  app.post('/showname', (req,res) => {
12      console.log(req.body.name);
13      res.end("your name is:" + req.body.name);
14  })
15
16  app.listen(port, hostname, () => {
17      console.log(`Server running at http://${hostname}:$
18      {port}/`);
19  });
```

Output: server

kejkaew

Cookies

- เป็นข้อมูลขนาดเล็กที่ถูกเก็บไว้ที่ web browser
- สร้างโดย server จากการร้องขอของ client
- ใช้สำหรับ
 - จัดการการเข้าใช้งาน ตามเวลาที่กำหนด (Session)
 - จัดการเก็บข้อมูลบางอย่าง ตามที่กำหนด (Personalization)
 - User tracking

cookie-parser

- การใช้ cookies ใน Express เราต้องมี cookie-parser module ก่อน
- To install it, use the following code

```
npm install cookie-parser
```

- การใช้งาน cookie-parser ต้องทำการ import

```
var cookieParser = require('cookie-parser');  
app.use(cookieParser());
```

Cookie ใน web browser

- สำหรับ client เราสามารถเข้าถึง cookie ได้โดยใช้คำสั่ง `document.cookie`
- ตัวอย่าง ที่ console ใน browser

```
> document.cookie  
< "img=1234.jpg"
```

ตัวอย่าง ปุ่ม cookie

```
<form class="form" action="/cookies" method="get">
  <fieldset>
    <legend>Cookies</legend>
    <input type="submit" value="Set Cookies">
  </fieldset>
</form>
```



ตัวอย่าง เมื่อเริ่มรันครั้งแรก

index.js

```
2 let cookiename = getCookie("name");
3 console.log(cookiename);
4
5 function getCookie(name){
6     var value = "";
7     try{
8         value = document.cookie.split("; ").find(row =>
9             row.startsWith(name)).split('=')[1]
10         return value
11     }catch(err){
12         return false
13     }
14 }
```

Output: browser

Navigated to <http://localhost:3001/>
false

เมื่อกดปุ่ม set cookies แล้วรันอีกที

server.js

```
21 const express = require('express');
22 const app = express();
23 const hostname = 'localhost';
24 const port = 3001;
25
26 app.use(express.static(__dirname));
27 var cookieParser = require('cookie-parser');
28 app.use(cookieParser());
29
30 app.get('/cookies', (req, res) => {
31   res.cookie('name', 'keroro');
32   res.cookie('img', '1234.jpg')
33   res.end("cookies set");
34 });
35
36 app.listen(port, hostname, () => {
37   console.log(`Server running at http://\${hostname}:\${port}/`);
38 });
```

Output: html

cookies set

Output: server

keroro

การกำหนด expire ให้ cookie

- ใช้ **maxAge** property ในการกำหนดว่า cookie จะมีอยู่ใน web browser นานเท่าไร
- เช่น กำหนดว่าให้อยู่ 10 วินาที

```
res.cookie('name', 'keroro',{maxAge: 10000},'path=/');
```

```
30 app.get('/cookies', (req, res) => {  
31   res.cookie('name', 'keroro',{maxAge: 10000},'path=/');  
32   res.cookie('img', '1234.jpg')  
33   res.end("cookies set");  
34 });
```

Upload Image File

- สามารถทำการ upload file โดยใช้ HTML form และ ใช้ multer module ของ Node.js
- ทำการ install multer module

`npm install multer`

- การใช้งาน multer ต้องทำการ import ก่อน

`const multer = require('multer');`

Upload image file: setup

```
40 // upload image file
41 const express = require('express');
42 const app = express();
43 const hostname = 'localhost';
44 const port = 3001;
45 const bodyParser = require('body-parser');
46 const multer = require('multer');
47 const path = require('path');
48
49 app.use(express.static(__dirname));
50 app.use(bodyParser.json());
51 app.use(bodyParser.urlencoded({extended: false}));
```

File storage

- ในการใช้ multer ต้องกำหนดว่าเราจะจัดเก็บ file ที่ไหน โดยใช้ `diskStorage()` method
- ทำการกำหนดดังนี้

```
53  const storage = multer.diskStorage({
54    destination: (req, file, callback) => {
55      callback(null, 'uploads/');
56    },
57    //file.fieldname แทน file.originalname ก็ได้
58    filename: (req, file, cb) => {
59      cb(null, file.fieldname + '-' + Date.now() +
60        path.extname(file.originalname));
61    });
```

Is it an image file?

- ทำการเช็คว่ file ที่อัปเข้ามาเป็น image file หรือเปล่า โดยใช้ function ต่อไปนี้

```
63 const imageFilter = (req, file, cb) => {  
64   // Accept images only  
65   if (!file.originalname.match(/\.(jpg|JPG|jpeg|JPEG|png|PNG|  
66     gif|GIF)$/)) {  
67     req.fileValidationError = 'Only image files are  
68       allowed!';  
69     return cb(new Error('Only image files are allowed!'),  
70       false);  
71   }  
72   cb(null, true);  
73 };
```

Upload a file via form

- single method ใน multer จะทำการ save file ที่กำหนดจาก fieldname ของ form และ save ใส่ req.file object
- ทำการ check errors ต่างๆ ที่อาจเกิดขึ้นด้วย

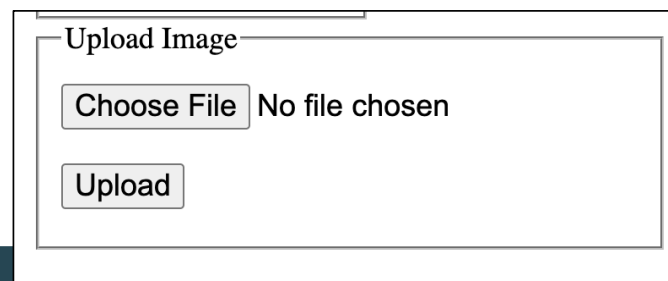
Upload a file via form

```
72 app.post('/upload', (req, res) => {
73   let upload = multer({ storage: storage, fileFilter:
    imageFilter }).single('avatar');
74
75   upload(req, res, (err) => {
76
77     if (req.fileValidationError) {
78       return res.send(req.fileValidationError);
79     }
80     else if (!req.file) {
81       return res.send('Please select an image to upload')
82       ;
83     }
84     else if (err instanceof multer.MulterError) {
85       return res.send(err);
86     }
87     else if (err) {
88       return res.send(err);
89     }
90
91     res.send('You uploaded this image filename: ' + req.
    file.filename);
92   });
93 }
```

Upload image file: HTML form

- set `enctype = "multipart/form-data"` ใน form tag

```
<form class="form" action="/uppic" method="post" id="formId"
enctype="multipart/form-data">
  <fieldset>
    <legend> Upload Image</legend>
    <input type="file" name="avatar" > <br>
    <input type="submit" value="Upload">
  </fieldset>
</form>
```



Upload Image

Choose File No file chosen

Upload

ผลการรัน

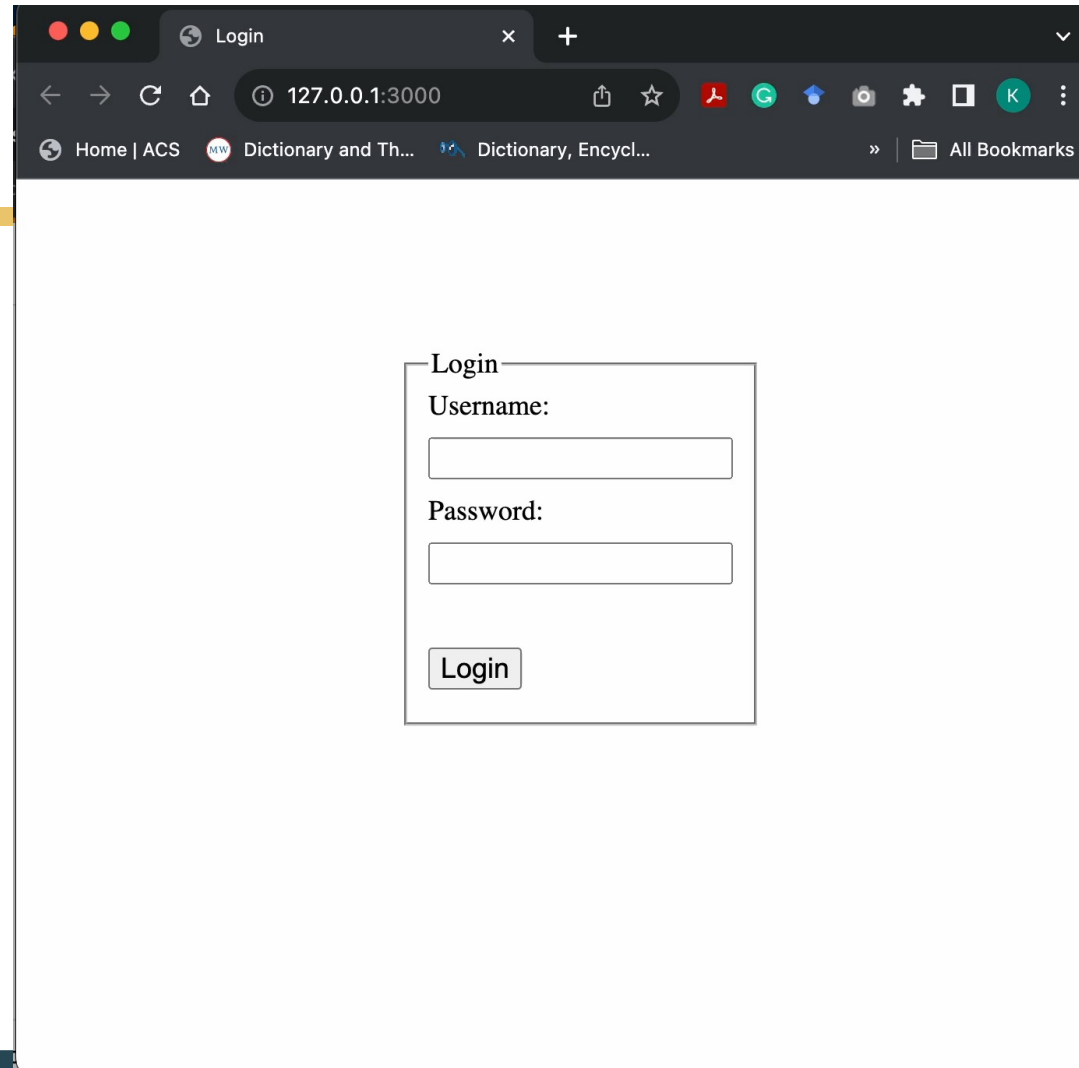
You uploaded this image filename: avatar-1617872282385.png

Assignment 11

- Download assignment11.zip file
- เราจะทำการสร้างระบบ Login สำหรับ social media โดย Login ของเราจะทำการตรวจสอบ username and password จากข้อมูลใน js/userDB.json เพื่อเข้าไปหน้า feed.html หลังจากนั้นในหน้า feed เราจะทำการเรียก postDB.json เพื่อแสดงข้อความที่เคย post ไปแล้วออกมา
- มีการ check cookie ก่อนที่จะเข้าหน้า feed ถ้าไม่มี cookie ที่ชื่อ username จะไม่สามารถเข้าหน้า feed ได้

-
- ทำระบบ Post ข้อความให้สมบูรณ์โดย save เพิ่มใน postDB.json และ post บน
ลงใน พื้นที่ที่กำหนดให้
 - สร้าง profile picture สำหรับ social media ของเรา โดยการ
 - Upload image file ลง server
 - แก้ไขข้อมูลใน userDB.json เมื่อมีการ อัปเดต image file
 - แสดง รูป profile picture ที่เรา upload ลงไปในพื้นที่ที่กำหนดให้
 - ส่งด้วยนะคะ (แค่ 2 files คือ feed.js และ server.js)

ตัวอย่าง



A screenshot of a web browser window displaying a login page. The browser's address bar shows the URL "127.0.0.1:3000". The page has a dark header with navigation links: "Home | ACS", "Dictionary and Th...", and "Dictionary, Encycl...". A "Login" button is visible in the top right corner of the header. The main content area is white and contains a centered login form. The form has a title "Login", a "Username:" label followed by a text input field, a "Password:" label followed by a text input field, and a "Login" button at the bottom.

Login

Username:

Password:

Login

Week 11: Classroom game

- ทำ classroom game ใน LEB2 ก่อนสัปดาห์หน้า