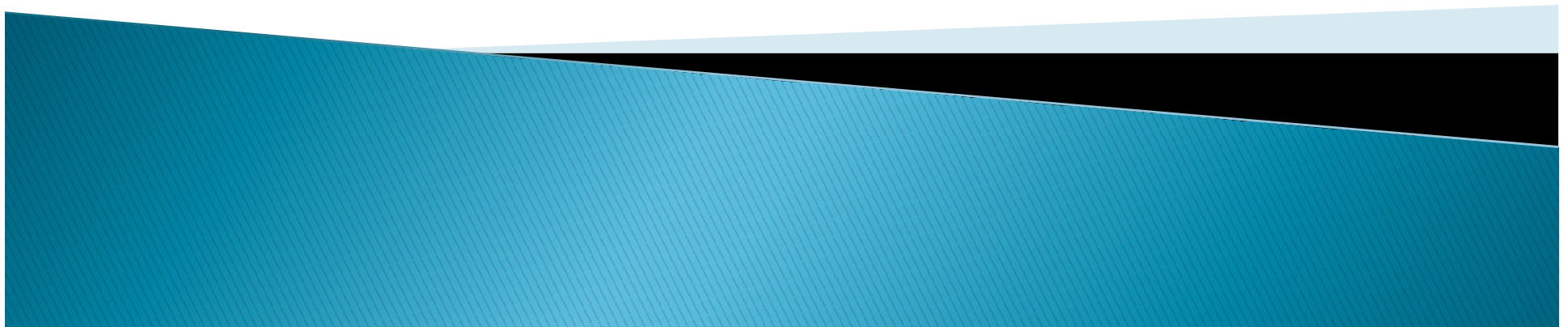


พัฒนา Modern Web Application ด้วย Node.js

โค้ชเอก

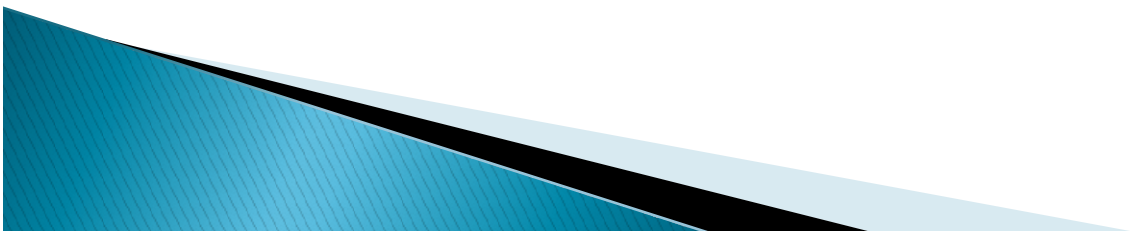
Codingthailand.com



Express.js

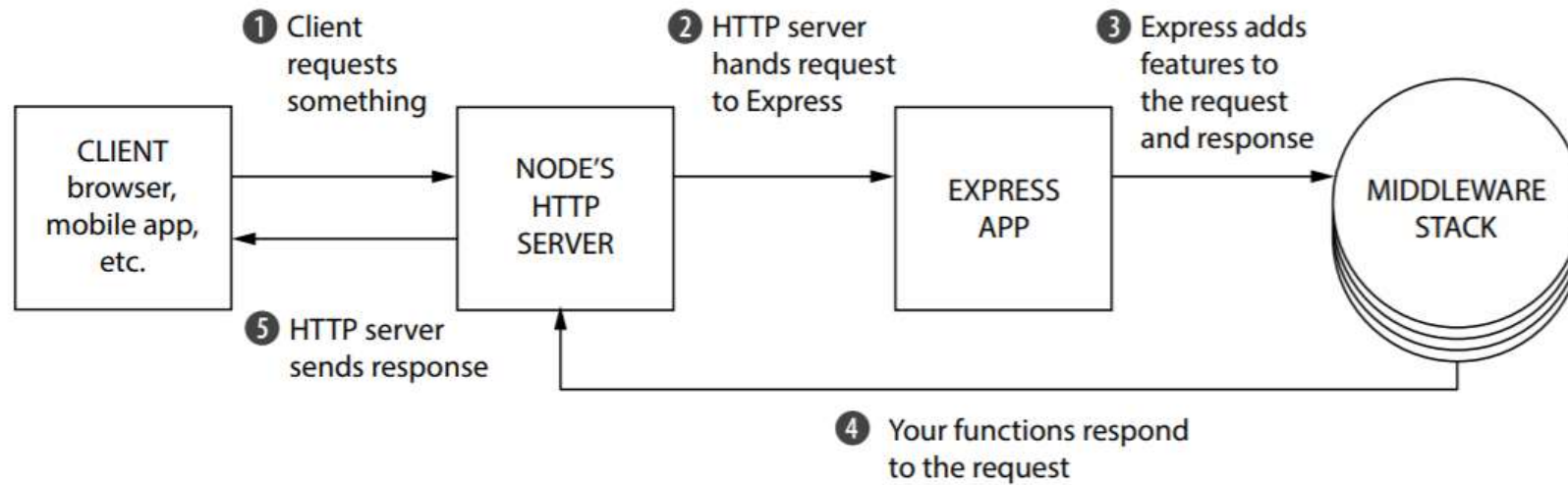
- ▶ <https://expressjs.com/>
- ▶ Express is a minimal and flexible Node.js web application framework.

```
const express = require('express')  
const app = express()  
  
app.get('/', (req, res) => res.send('Hello World!'))  
  
app.listen(3000, () => console.log('Example app listening on port 3000!'))
```



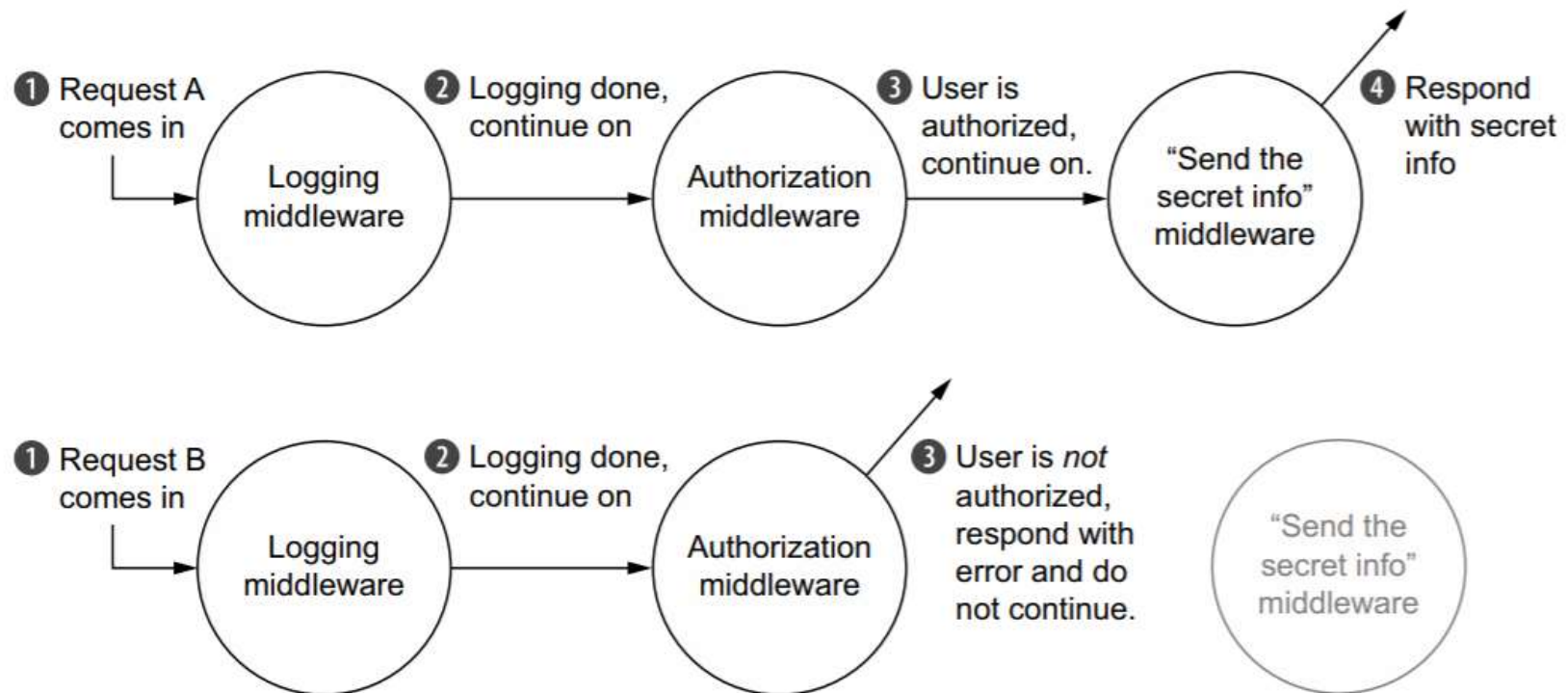
Express.js

- ▶ The flow of a request through an Express.

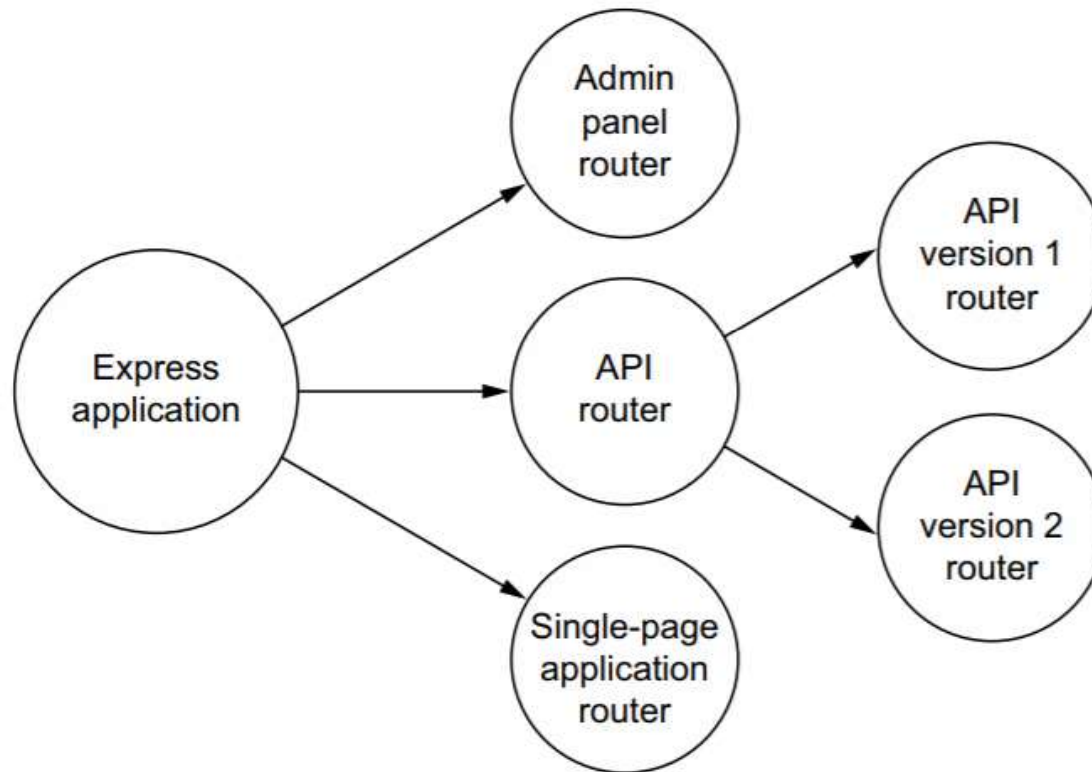


Express.js (Middleware)

- ▶ These smaller request handler functions are **called middleware functions, or middleware.**
- ▶ middleware is a function with three arguments: request (or req), response (res), and next

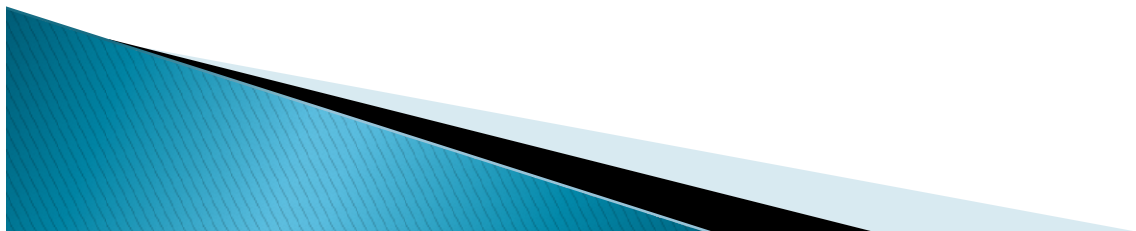


Express.js (Routing)



Express application generator

- ▶ `npm install express-generator -g`
- ▶ `express -h`
- ▶ `express --view=hbs myapp`
- ▶ `cd <ชื่อโฟลเดอร์>`
- ▶ `npm install`
- ▶ <https://expressjs.com/en/starter/generator.html>



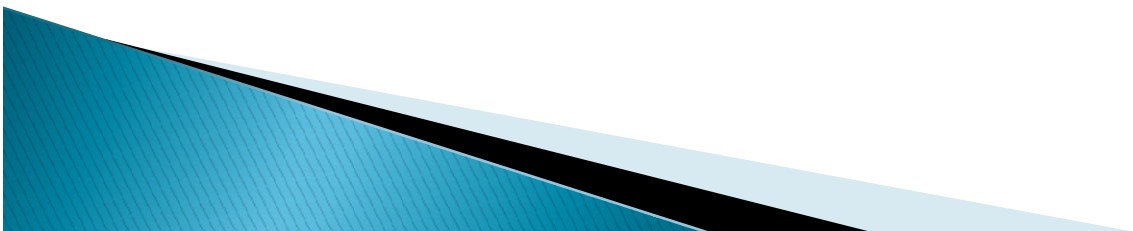
Express.js

- ▶ ติดตั้งและใช้งาน nodemon
- ▶ <https://github.com/remy/nodemon/>
- ▶ `npm install nodemon --save-dev`

```
{  
  "name": "nodeweb",  
  "version": "1.0.0",  
  "private": true,  
  "scripts": {  
    "start": "nodemon ./bin/www"  
  },  
  "dependencies": {  
    "body-parser": "~1.18.2",  
    "cookie-parser": "~1.4.3",  
    "debug": "~2.6.9",  
    "express": "~4.15.5",  
    "express-handlebars": "^3.0.0",  
    "morgan": "~1.9.0",  
    "serve-favicon": "~2.4.5"  
  },  
  "devDependencies": {  
    "nodemon": "^1.17.2"  
  }  
}
```

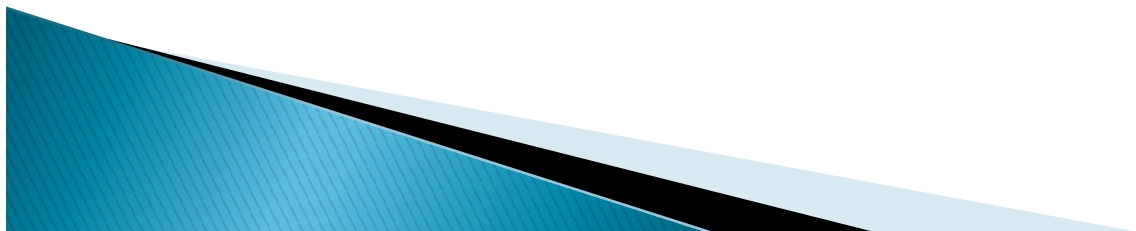
Express.js

- ▶ ทดลองเขียน route



Express Configuration

- ▶ `app.set()` and `app.get()`
- ▶ ตัวอย่างการใช้งาน
- ▶ `app.set('port', process.env.PORT || 3000);`
- ▶ `console.log('Express server listening on port ' + app.get('port'));`



Express.js (Views)

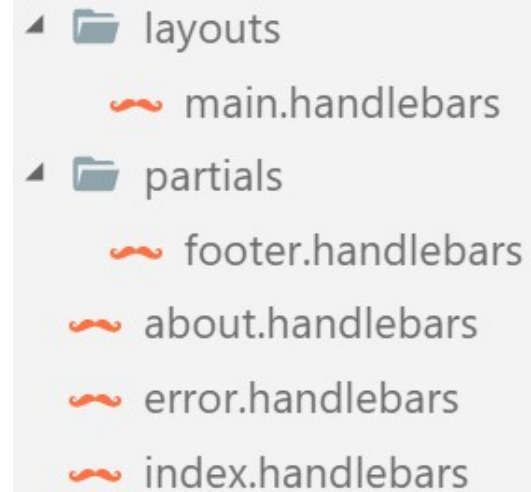
- ▶ View or Template Engine
- ▶ <https://expressjs.com/en/guide/using-template-engines.html>
- ▶ เราเลือกใช้ handlebars
- ▶ <https://github.com/ericf/express-handlebars>
- ▶ Handlebars helper
- ▶ <https://github.com/helpers/handlebars-helpers>
- ▶ เว็บไซต์หลักของ handlebars
- ▶ <https://handlebarsjs.com/>



Express.js (Views)

▶ ทดลองเขียน Views

```
app.engine('handlebars', exphbs({
  defaultLayout: 'main',
  partialsDir: [
    'views/partials/'
  ]
}));
app.set('view engine', 'handlebars');
```



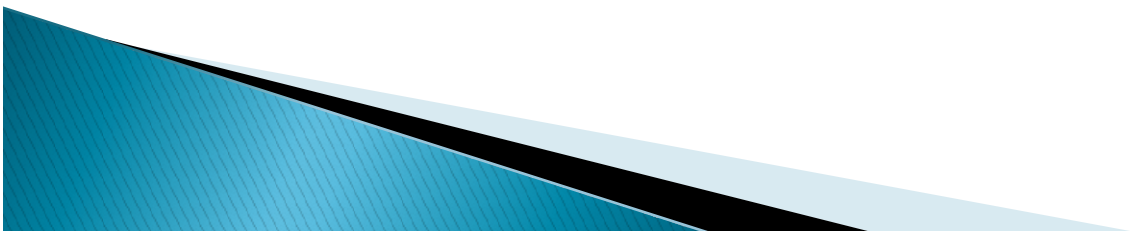
A file explorer view showing the structure of Handlebars templates. It displays two folders: 'layouts' and 'partials'. Under 'layouts', there is a file named 'main.handlebars'. Under 'partials', there are four files: 'footer.handlebars', 'about.handlebars', 'error.handlebars', and 'index.handlebars'. Each file is represented by a small orange mustache icon.

- layouts
 - main.handlebars
- partials
 - footer.handlebars
 - about.handlebars
 - error.handlebars
 - index.handlebars

Express.js (Views)

- ▶ กรณีต้องการเปลี่ยนนามสกุลของไฟล์ view เป็น .hbs

```
app.engine('.hbs', exphbs({  
  extname: '.hbs',  
  defaultLayout: 'main',  
  partialsDir: [  
    'views/partials/'  
  ],  
  helpers: helpers  
}));  
app.set('view engine', '.hbs');
```



Express.js (Views Layouts)

- ▶ ทดลองเขียน Views Layout

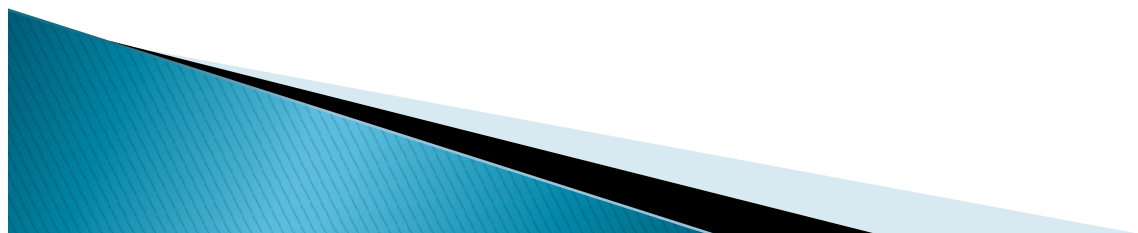
```
<!DOCTYPE html>
<html>
  <head>
    <title>{{title}}</title>
    <link rel='stylesheet' href='/stylesheets/style.css' />
  </head>
  <body>
    {{{body}}}
    {{> footer}}
  </body>
</html>
```

A diagram illustrating the layout structure. Two red arrows point to the placeholders in the code: one points to the `{{{body}}}` placeholder, and the other points to the `{{> footer}}` placeholder.

Express.js (Views) : การใช้งาน handlebars

- ▶ <http://handlebarsjs.com/reference.html>
- ▶ http://handlebarsjs.com/block_helpers.html

```
<ul>
  {{#each p.names}}
    <li>{{@index}} -- {{this.firstName}} {{ this.lastName }}</li>
  {{/each}}
</ul>
```



Express.js

- ▶ ทดลองเขียน Controllers

JS about.js x

```
1 exports.getAbout = (req, res) => {  
2   res.render('about', {  
3     title: 'about'  
4   });  
5 };
```

```
/**  
 * Controllers (route handlers).  
 */  
const aboutController = require('./controllers/about');  
const indexController = require('./controllers/index');  
/**  
 * Primary app routes.  
 */  
app.get('/', indexController.index);  
app.get('/about', aboutController.getAbout);
```

app.js

Express.js

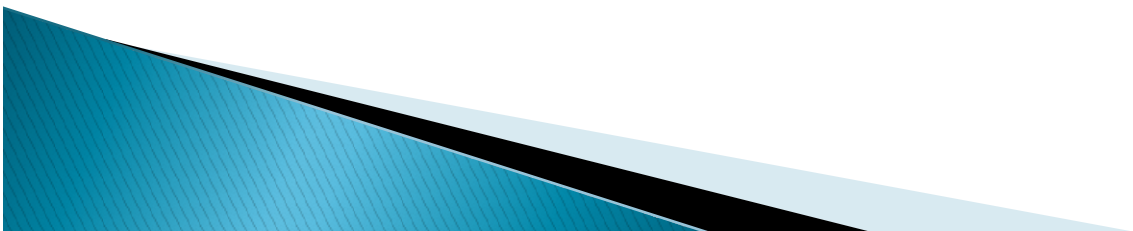
- ▶ ทดลองเขียน Models และเรียกใช้

```
User.js  x
1
2  let User = {
3      "occupation" : "developer",
4      "website" : {
5          "name" : "codingthailand"
6      },
7      "names" : [
8          {"firstName" : "Akenarin", "lastName" : "Komkoon"},
9          {"firstName" : "John" , "lastName" : "Doe"}
10     ]
11 };
12
13
14 module.exports = User;
```

```
index.js  x
1
2  const User = require('../models/User');
3  |
```


Express.js

- ▶ ทดลองนำ Bootstrap 4 เข้ามาใช้งาน
- ▶ <https://getbootstrap.com/>
- ▶ ทดลองสร้าง header และ เมนู



The end

