

Software Park Thailand
</Code Camp>

SOFTWARE PARK
THAILAND



Express & MVC model

Worawut Worawichwongsa

แนะนำผู้สอน

- วรวิทย์ วรวิชัยวงศ์
- ประวัติการศึกษา
 - ปริญญาตรี สาขาวิทยาการคอมพิวเตอร์
จากจุฬาลงกรณ์มหาวิทยาลัย เกียรตินิยมอันดับหนึ่ง
- ประวัติการเขียนโปรแกรม
 - เขียนโปรแกรมมาตั้งแต่ ป.5
 - งานอดิเรก หาช่องโหว่เกม เขียนโปรแกรมเล่นเกมแทน
 - ภาษาโปรแกรมใช้ VB6, C++, C#, Java, JavaScript, PHP, Python



Software Park Thailand
</Code Camp>

Outline

- Express คืออะไร
- Lab 1 : Hello, world!
- Lab 2 : Routing
- Lab 3 : Static content กับ middlewares
- Lab 4 : Query and Body
- Lab 5 : MVC model
- Daily Project



Learning Outcome

- ผู้เรียนมีความเข้าใจ และสามารถออกแบบระบบ Backend ได้โดยใช้ Express
- ผู้เรียนสามารถออกแบบระบบได้ตาม MVC model
- ผู้เรียนมีพื้นฐาน ที่จะหาความรู้เพิ่มเติมได้เองในอินเทอร์เน็ต



Software Park Thailand
</Code Camp>

Outline

- Express คืออะไร
- Lab 1 : Hello, world!
- Lab 2 : Routing
- Lab 3 : Static content กับ middlewares
- Lab 4 : Query and Body
- Lab 5 : MVC model
- Daily Project



Express คืออะไร

- Express เป็นโมดูลหนึ่งของ Node.js
- Express เป็นตัวดูแล HTTP request ที่ได้รับ
- Express ถูกใช้ในระบบ Backend มากที่สุดของ Node.js
- Express มีส่วนเสริมจำนวนมากให้เลือกใช้ ตอนไม่อยากจะเขียนเองทุกอย่าง

Express คืออะไร

express

4.17.1 • Public • Published 5 months ago

Readme

30 Dependencies

37,038 Dependents

263 Versions

express

Fast, unopinionated, minimalist web framework for **node**.

npm v4.17.1 downloads 45M/month linux passing windows passing coverage 100%

```
const express = require('express')
const app = express()
```

```
app.get('/', function (req, res) {
  res.send('Hello World')
})
```

```
app.listen(3000)
```

install

```
> npm i express
```

± weekly downloads

10,626,622

version

4.17.1

license

MIT

open issues

121

pull requests

59

homepage

expressjs.com

repository

github

last publish

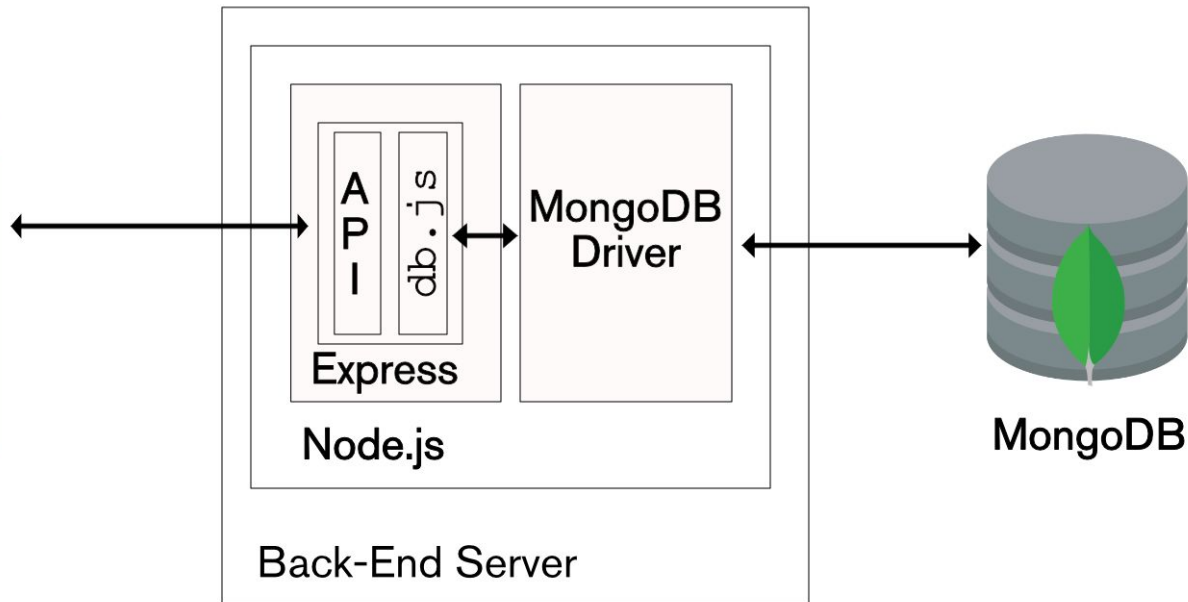
5 months ago



Software Park Thailand
</Code Camp>

Express คืออะไร

```
shell  
  
curl localhost:3000/  
pop/ip  
  
{ "ip": "92.237.136.220" }
```



Express คืออะไร - ทำไมเราต้องการ Express

ทบทวนการทำ Web Server ด้วย node.JS

```
var http = require('http');  
  
http.createServer((req, res) => {  
  res.write('hello world');  
  res.end();  
}).listen(3000);
```

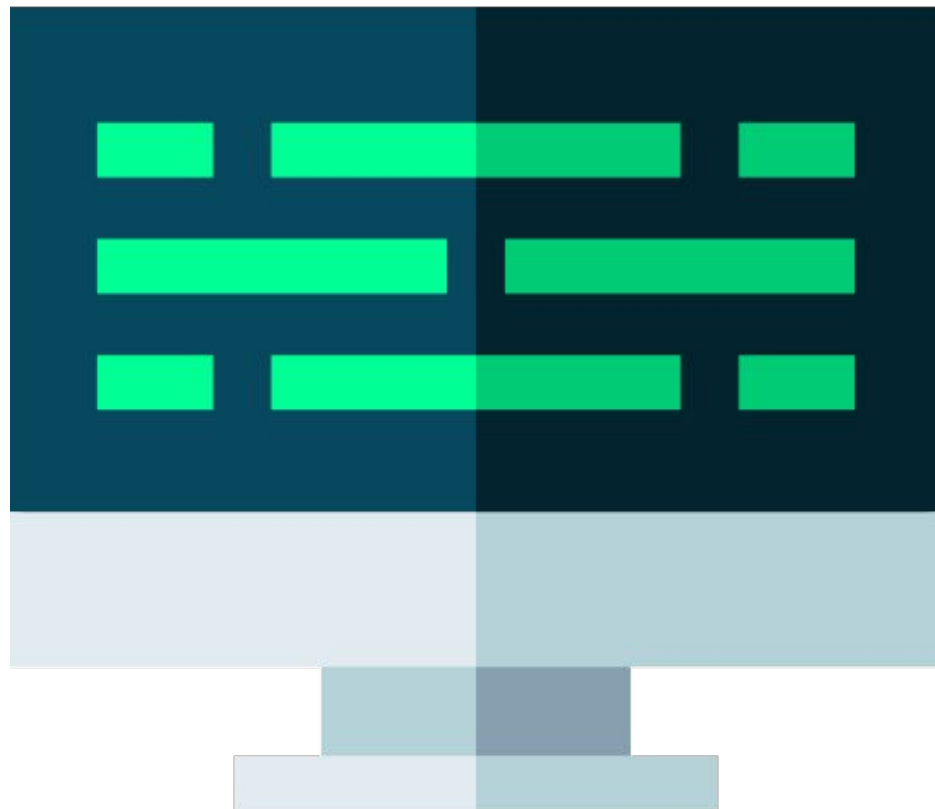


แต่ในความเป็นจริง เราจะต้องมีการจัดการหลาย End Point

เขียนด้วย NodeJS จะได้ประมาณนี้ จะเห็นว่า Code ดูค่อนข้างซับซ้อน

```
var http = require('http');

http.createServer((req, res) => {
  if(req.url == '/') {
    res.write('hello world');
  } else if(req.url == '/tom'){
    res.write('hello tom')
  } else if(req.url == '/joe'){
    res.write('hello joe')
  }
  res.end();
}).listen(3000);
```





วิธีการลง Express

- เปิด cmd ในโฟลเดอร์ที่ต้องการให้ Express ทำงาน
- พิมพ์คำสั่ง
 - `npm i express` แล้วกด enter
- ยินดีด้วย คุณลง Express สำเร็จแล้ว



Software Park Thailand
</Code Camp>

Outline

- Express คืออะไร
- Lab 1 : Hello, world!
- Lab 2 : Routing
- Lab 3 : Static content กับ middlewares
- Lab 4 : Query and Body
- Lab 5 : MVC model
- Daily Project



Lab 1 : Hello, world!

- สร้างไฟล์ชื่อ app.js ที่โฟลเดอร์ที่ลง Express
- ลอกโค้ดต่อไปนี้ลงไฟล์ app.js

```
var express = require('express')
var app = express()

app.get('/', function (req, res) {
  res.send('hello world')
})

app.listen(3000)
```

- เปิด cmd พิมพ์ “node app.js” กด enter

Lab 1 : Hello, world!

- เปิด Web browser ที่คุณชื่นชอบ
- เข้าเว็บ <http://localhost:3000/>

hello world

Lab 1 : Hello, world!

- การใช้ web browser เข้าเว็บ
 - <http://localhost:3000/>
- http:// เชื่อมแบบ http
- localhost เชื่อมไปเครื่องตัวเอง
- :3000 เชื่อมที่ port 3000
- / ขอ path นี้
- การเข้าเว็บแบบนี้ จะส่ง “GET” เสมอ



Lab 1 : Hello, world!

- require('express')
 - เรียก express ที่เพิ่งลง
- express()
 - สร้าง application express

```
var express = require('express')
var app = express()

app.get('/', function (req, res) {
  res.send('hello world')
})

app.listen(3000)
```



Lab 1 : Hello, world!

- app.get('/', function)
 - ดัก “GET” ที่ path /
 - แล้วส่งให้ function

```
function (req, res) {  
    res.send('hello world')  
}
```

เป็น function รับ request
แล้ว reponse 'hello world' คืนไป

```
var express = require('express')  
var app = express()  
  
app.get('/', function (req, res) {  
    res.send('hello world')  
})  
  
app.listen(3000)
```



Software Park Thailand
</Code Camp>

Lab 1 : Hello, world!

- app.listen(3000)
 - รอรับ request ที่ port 3000

```
var express = require('express')  
var app = express()
```

```
app.get('/', function (req, res) {  
  res.send('hello world')  
})
```

```
app.listen(3000)
```

Lab 1 : Hello, world!

- โจทย์ Lab 1
 - ที่เว็บ <http://localhost:5555/bye>
 - ให้แสดงคำว่า “Good bye”



Lab 1.2 : Hello World - Post/Put/Delete

1. เปลี่ยนมาทำให้รองรับ Post
2. เปลี่ยนมาทำให้รองรับ Put
3. เปลี่ยนมาทำให้รองรับ Delete



Software Park Thailand
</Code Camp>

Outline

- Express คืออะไร
- Lab 1 : Hello, world!
- Lab 2 : Routing
- Lab 3 : Static content กับ middlewares
- Lab 4 : Query and Body
- Lab 5 : MVC model
- Daily Project



Lab 2 : Routing

- แก้ app.js ให้เป็นแบบนี้

```
var express = require('express')
var app = express()

app.get('/user/first', function (req, res) {
  res.send('first on user')
})

app.get('/user/second', function (req, res) {
  res.send('second on user')
})

app.listen(3000)
```

Lab 2 : Routing

- ดูเว็บ <http://localhost:3000/user/first>
- ดูเว็บ <http://localhost:3000/user/second>
- ถ้ามีหลาย user หลาย first/second จะดูมันมาก
- <http://localhost:3000/user/first>
- <http://localhost:3000/user/second>
- <http://localhost:3000/item/dosomething>
- <http://localhost:3000/item/omg>



Lab 2 : Routing

- สร้างไฟล์ Route เพิ่ม

```
var express = require('express')
var app = express()

var userRoute = require('./userRoute')
app.use('/user', userRoute)

app.listen(3000)
```

app.js

```
var express = require('express')
var router = express.Router()

router.get('/first', function (req, res) {
  res.send('first on user')
})

router.get('/second', function (req, res) {
  res.send('second on user')
})

module.exports = router
```

userRoute.js



Lab 2 : Routing

- require('./userRoute')
 - เรียกไฟล์ userRoute.js
- app.use('/user', userRoute)
 - ดั๊กที่ “sub-path” /user
 - แล้วส่งต่อให้ userRoute

```
var express = require('express')
var app = express()

var userRoute =
require('./userRoute')
app.use('/user', userRoute)

app.listen(3000)
```

app.js

Lab 2 : Routing

- ถ้า path เป็น /user/first/second

Sub-path	Path ที่ส่งต่อ
/	/user/first/second
/user	/first/second
/user/first	/second

Lab 2 : Routing

- express.Router()
 - สร้าง “Router”
- router.get(...)
 - เหมือน express app
 - ดัก GET แล้วส่งต่อให้ข้างใน
- module.exports = router
 - ส่ง router ออก
 - require คือใช้ตัวนี้แหละ

```
var express = require('express')
var router = express.Router()

router.get('/first', function (req, res) {
  res.send('first on user')
})

router.get('/second', function (req, res) {
  res.send('second on user')
})

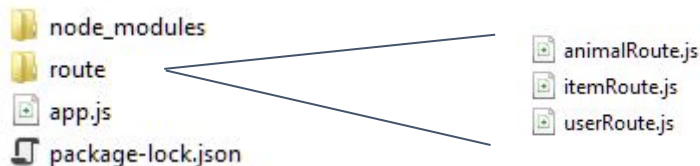
module.exports = router
```

userRoute.js



Lab 2 : Routing

- ข้อสังเกต
 - ใน `app.get` หรือ `router.get` จะใช้ `path`
 - ใน `app.use` หรือ `router.use` จะใช้ `sub-path` หรือข้ามไปเลยก็ได้ (ดักทุก request)
 - เนื่องจากอาจจะมีหลาย router ส่วนใหญ่จะสร้างโฟลเดอร์ใหม่ แล้วเอา router ใส่ใน folder นั้น เพื่อจัดหมวดหมู่โปรแกรม



Lab 2 : Routing

- โจทย์ Lab 2

- แสดง 'dog walk' ที่ <http://localhost:3000/dog/walk/>
- แสดง 'dog play' ที่ <http://localhost:3000/dog/play/>
- แสดง 'cat eat' ที่ <http://localhost:3000/cat/eat/>
- แสดง 'cat sleep' ที่ <http://localhost:3000/cat/sleep/>



Software Park Thailand
</Code Camp>

Outline

- Express คืออะไร
- Lab 1 : Hello, world!
- Lab 2 : Routing
- Lab 3 : Static content กับ middlewares
- Lab 4 : Query and Body
- Lab 5 : MVC model
- Daily Project



Lab 3 : Static content

- เพิ่มบรรทัดนี้ลงไป
- สร้าง folder “public”
- ลองใส่ไฟล์อะไรก็ได้ในโฟลเดอร์ “public”
- ดู http://localhost:3000/ไฟล์

```
var express = require('express')
var app = express()

app.use(express.static('./public/'))

var userRoute =
require('./userRoute')
app.use('/user', userRoute)

app.listen(3000)
```

app.js



Lab 3 : Static content

- `express.static('./public/')`
 - สร้าง “express-middleware”
 - Response ไฟล์จากโฟลเดอร์ public
 - ถ้าไม่มี จะส่ง request ต่อไป
- `app.use(middleware)`
 - ใช้ “express-middleware”
 - ดักทุก request ส่งให้ middleware

```
var express = require('express')
var app = express()

app.use(express.static('./public/'))

var userRoute =
require('./userRoute')
app.use('/user', userRoute)

app.listen(3000)
```

app.js

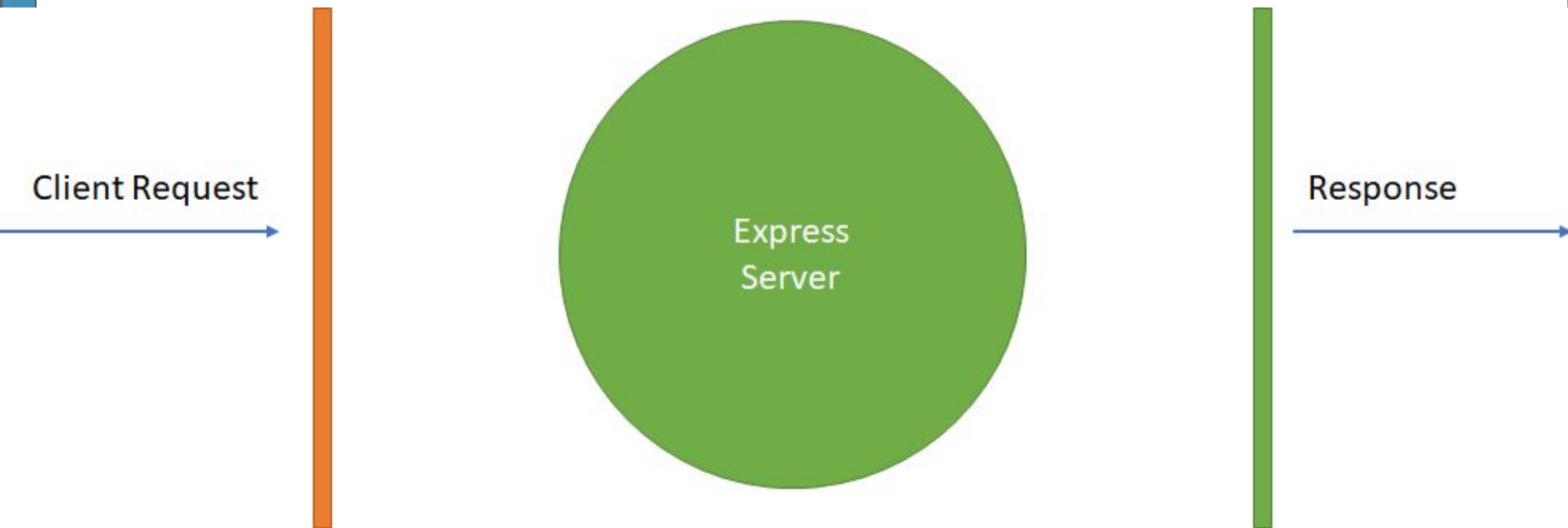
Lab 3 : Middleware

- middleware คืออะไร?
 - เป็น function ที่รับ 1 parameter
 - next (Next function) เรียก Middleware ตัวถัดไป
- express-middleware คืออะไร?
 - เป็น function ที่รับ 3 parameters
 - req (Request) HTTP request
 - res (Response) HTTP response
 - next (Next function) เรียก Middleware ตัวถัดไป



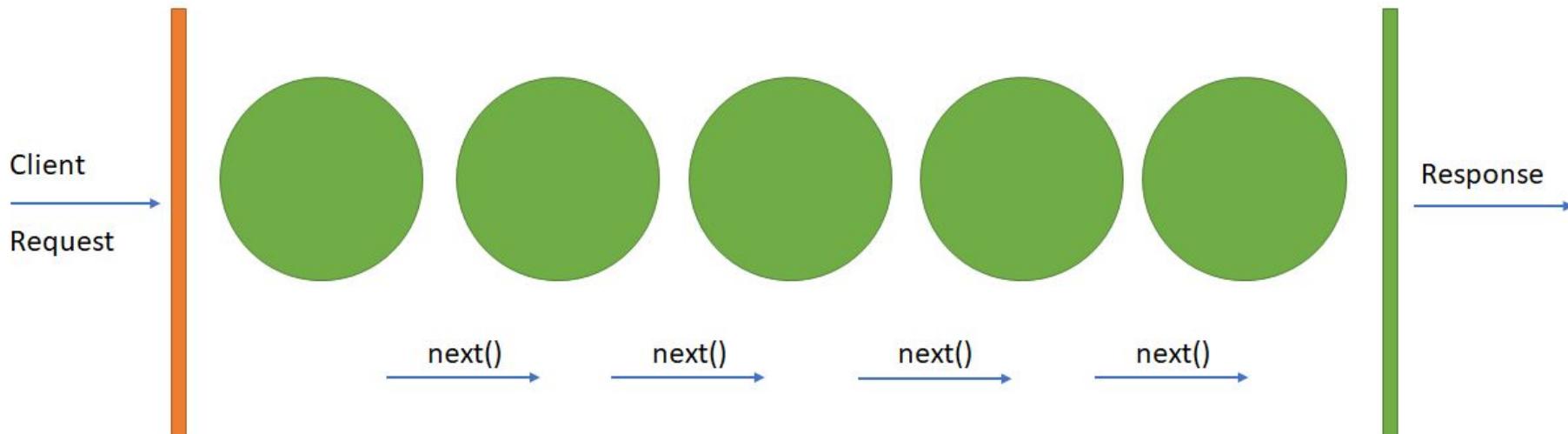
Software Park Thailand
</Code Camp>

การทำงานปกติแบบไม่มี Middleware

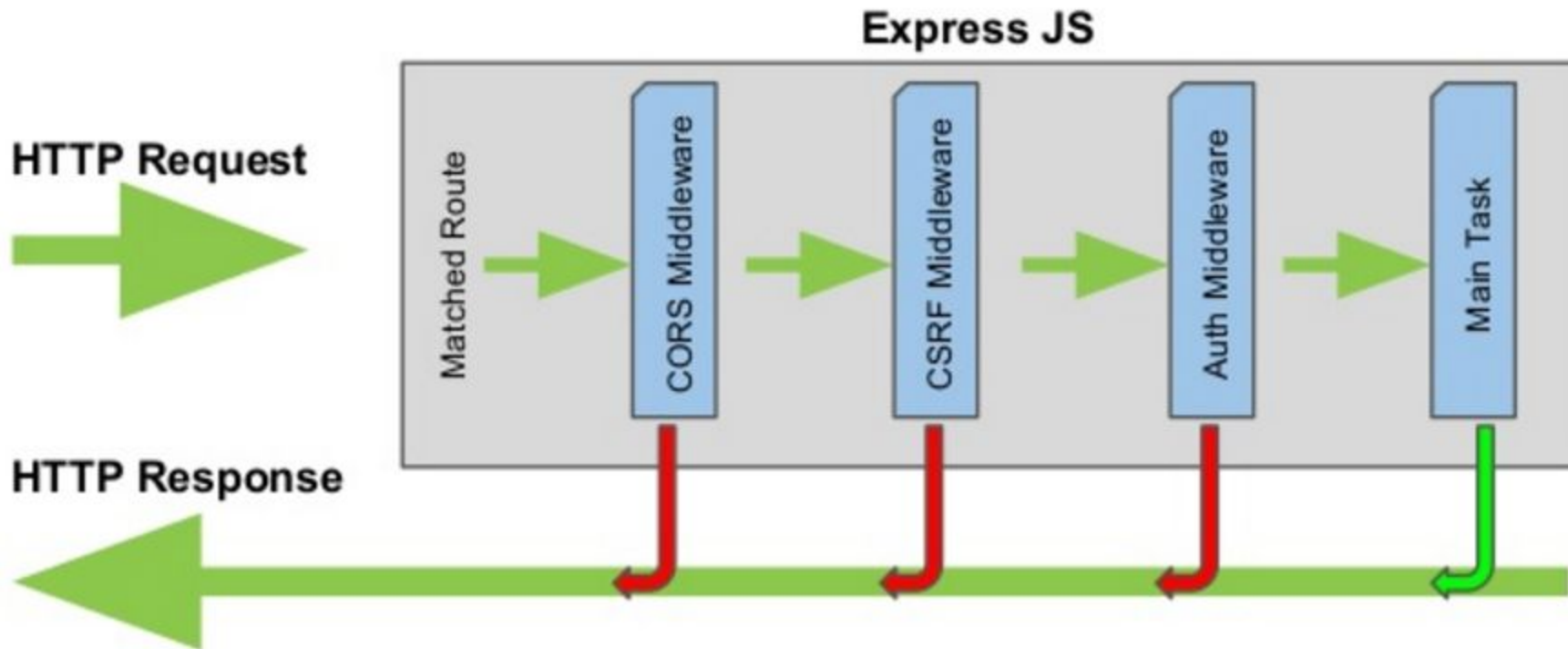


การทำงานแบบมี Middleware

All middleware has access to req,res and next



Sample Express JS Middleware





Middleware ด้วย NodeJS

```
var connect = require('connect'),
    util = require('util');

var interceptorFunction = function(request, response, next) {
  console.log(util.format('Request for %s with method %s',
    request.url, request.method));
  next();
};

var app = connect()
  // .use('/log', interceptorFunction)
  .use(interceptorFunction)
  .use(function onRequest(request, response) {
    response.end('Hello from Connect!');
  }).listen(3001);
```



Lab 3 : Middleware

- ข้อสังเกต
 - Router เป็น middleware
 - app.use ใช้ middleware
 - router.get ใช้ middleware
 - จริงๆ function(req, res) จะเป็น middleware ถ้าเขียนเป็น function(req, res, next)

ทุกอย่างใน express
เป็น middleware ทั้งหมด

(ยกเว้น listen 3000 นะนะ)

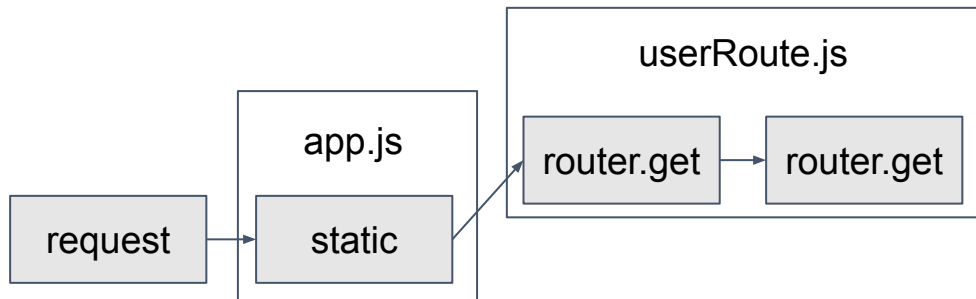
```
var express = require('express')
var app = express()

app.use(express.static('./public/'))

var userRoute =
require('./userRoute')
app.use('/user', userRoute)

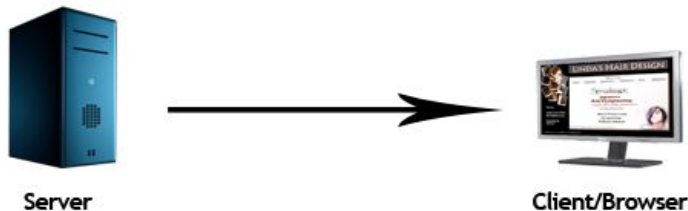
app.listen(3000)
```

Lab 3 : Middleware

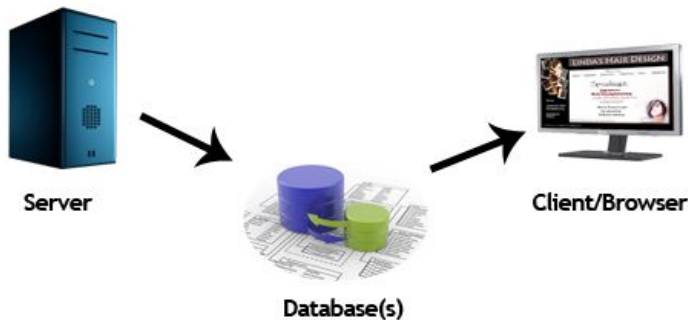


Static Content คืออะไร?

Static Website



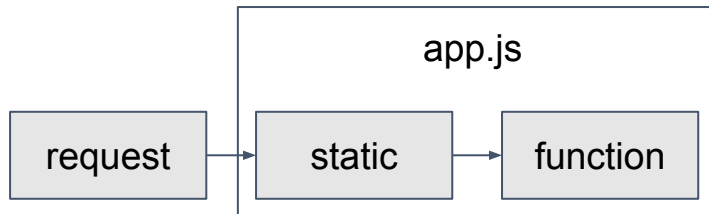
Dynamic Website



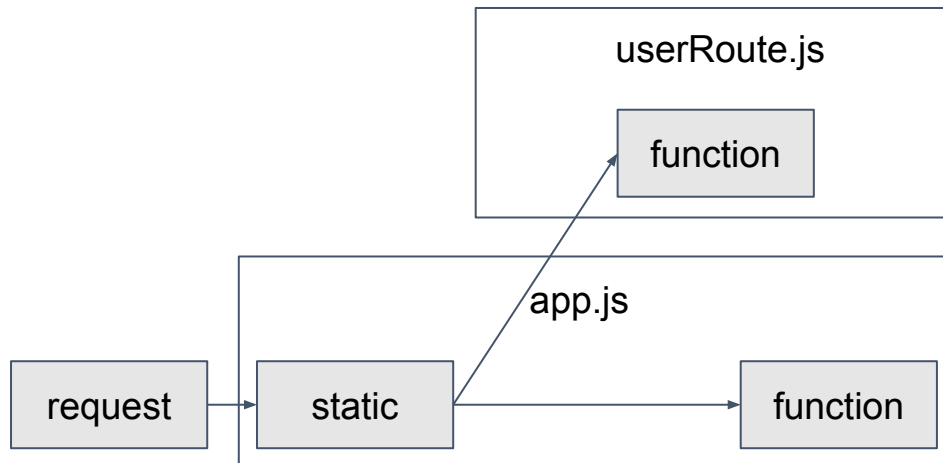
Lab 3 : Static Content กับ Middleware

- โจทย์ Lab 3
 - แสดงรูปอะไรก็ได้ที่ localhost:3000/picture.png
 - นอกจากนั้น ให้แสดงคำว่า “404 Not Found”
- Optional: ไม่ว่าจะเข้าที่ไหนใน localhost:3000/user ให้แสดงคำว่า “404 User Not Found”

Lab 3 : Static Content กับ Middleware



Lab 3 : Static Content กับ Middleware





Software Park Thailand
</Code Camp>

Outline

- Express คืออะไร
- Lab 1 : Hello, world!
- Lab 2 : Routing
- Lab 3 : Static content กับ middlewares
- Lab 4 : Query and Body
- Lab 5 : MVC model
- Daily Project



Lab 4 : Query

- แก้ userRoute.js ให้เป็นแบบนี้
- ลองดู
localhost:3000/user/?id=hello
localhost:3000/user/?id=1234
localhost:3000/user/?id=1&2

```
var express = require('express')
var router = express.Router()

router.get('/', function (req, res) {
  res.send(req.query.id)
})

module.exports = router
```

userRoute.js



Software Park Thailand
</Code Camp>

Lab 4 : Query & Parameter

- แก้ userRoute.js ให้เป็นแบบนี้
- ลองดู
localhost:3000/user/hello
localhost:3000/user/123456
localhost:3000/user/1/2

```
var express = require('express')
var router = express.Router()

router.get('/:id', function (req, res) {
  res.send(req.params.id)
})

module.exports = router
```

userRoute.js



Software Park Thailand
</Code Camp>

Lab 4 : Query & Parameter

- router.get('/:id', ...)
 - **:id** ใส่ข้อความตรงนี้ไว้ที่ req.params.**id**

```
var express = require('express')
var router = express.Router()

router.get('/:id', function (req, res) {
  res.send(req.params.id)
})

module.exports = router
```

userRoute.js



Lab 4 : Query & Parameter

- แก้ userRoute.js
- ลองดู localhost:3000/user/add/1/2

```
var express = require('express')
var router = express.Router()

router.get('/add/:a/:b', function (req, res) {
  var a = req.params.a
  var b = req.params.b
  res.send(a + b)
})

module.exports = router
```

userRoute.js

Lab 4 : Query & Parameter

- แก้ userRoute.js
- ลองดู localhost:3000/user/add/1/2

```
var express = require('express')
var router = express.Router()

router.get('/add/:a/:b', function (req, res) {
  var a = parseInt(req.params.a)
  var b = parseInt(req.params.b)
  res.send(String(a + b))
})

module.exports = router
```

userRoute.js

Lab 4 : Body

1. เปิด cmd
 - a. รัน 'npm i body-parser'
 - i.
2. แก้ตรงนี้ใน app.js
3. bodyParser.urlencoded
 - a. สร้าง middleware
 - i. สำหรับอ่าน content
 - b. แล้วใส่ค่าที่อ่านไว้ที่ req.body

```
var express = require('express')
var bodyParser = require('body-parser')
var app = express()

app.use(express.static('public'))

app.use(bodyParser.urlencoded({
  extended: true
}))

var userRoute = require('./userRoute')
app.use('/user', userRoute)

app.listen(3000)
```

app.js

Lab 4 : Body

1. แก้ userRoute.js ให้เป็นแบบนี้
2. router.post
 - a. ดัก POST ที่ /user/add
3. ลองดู localhost:3000/user/add

```
var express = require('express')
var router = express.Router()

router.post('/add', function (req, res) {
  var a = parseInt(req.body.a)
  var b = parseInt(req.body.b)
  res.send(String(a + b))
})

module.exports = router
```

userRoute.js

Lab 4 : Body

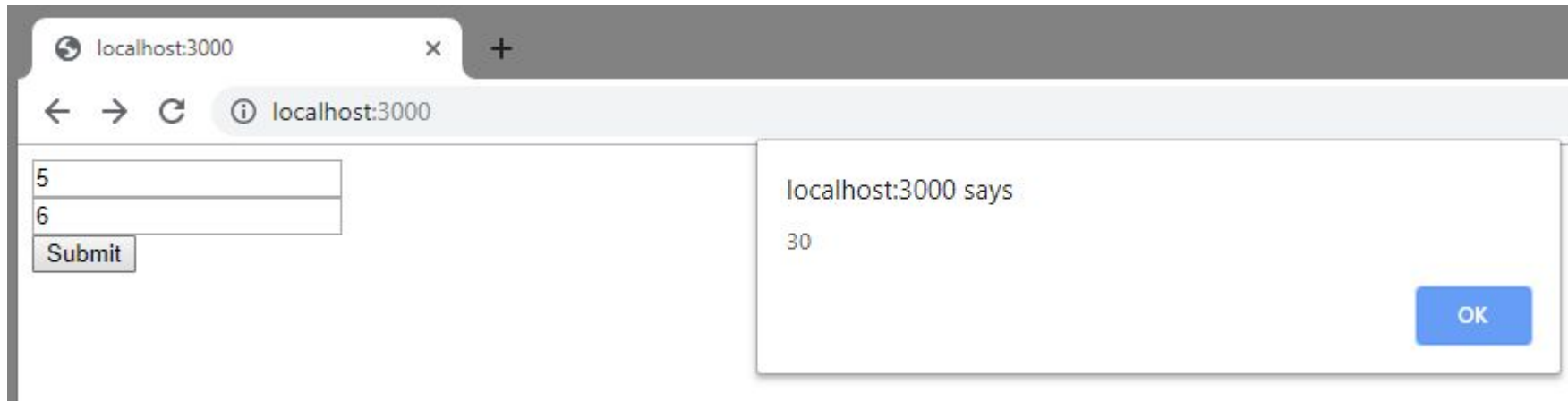
1. เพิ่ม index.html
2. ลองดู localhost:3000

```
<form action="/user/add" method="post">  
  <input type="text" name="a"><br>  
  <input type="text" name="b"><br>  
  <input type="submit" value="Submit">  
</form>
```

public/index.html

Lab 4 : Body

- โจทย์ Lab 4
 - สร้างเว็บที่ใช้ jQuery ทำเว็บคูณเลข





Software Park Thailand
</Code Camp>

Outline

- Express คืออะไร
- Lab 1 : Hello, world!
- Lab 2 : Routing
- Lab 3 : Static content กับ middlewares
- Lab 4 : Query and Body
- Lab 5 : MVC model
- Daily Project



Lab 5 : MVC Model

```
var express = require('express')
var bodyParser = require('body-parser')
var listRoute = require('./listRoute')

var app = express()
app.use(express.static('public'))
app.use(bodyParser.urlencoded({
  extended: true
}))
app.use('/list', listRoute)

app.listen(3000)
```

app.js

```
var express = require('express')
var router = express.Router()

var list = []

router.get('/', function(req, res) {
  res.send(list)
})
router.post('/', function(req, res) {
  list.push(req.body.data)
  res.send(list)
})

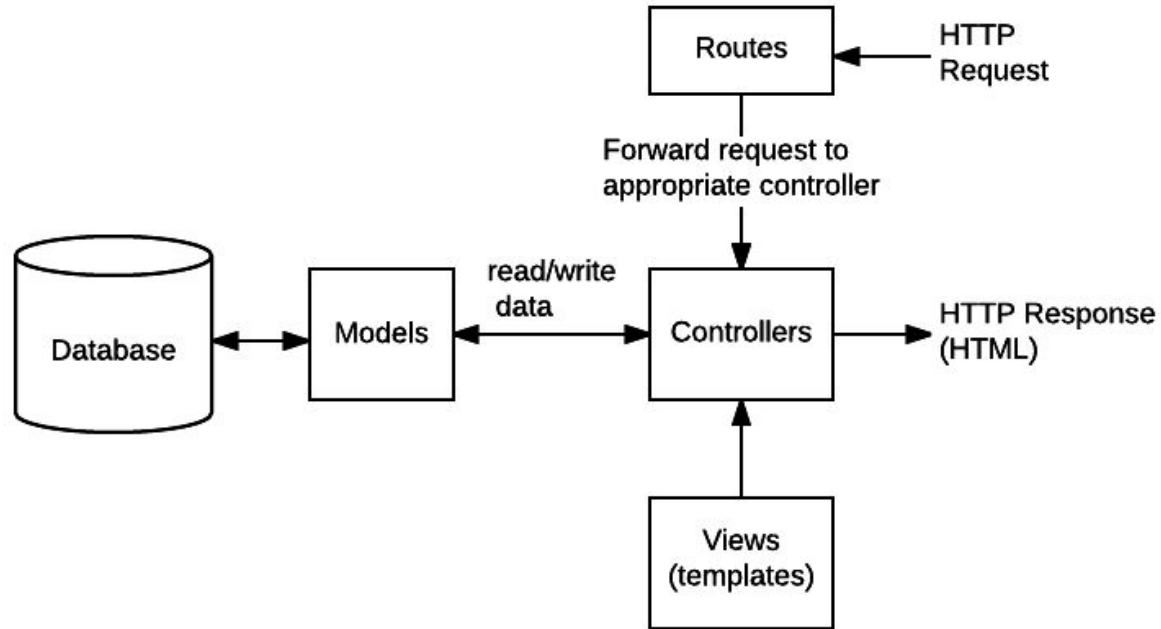
module.exports = router
```

listRoute.js

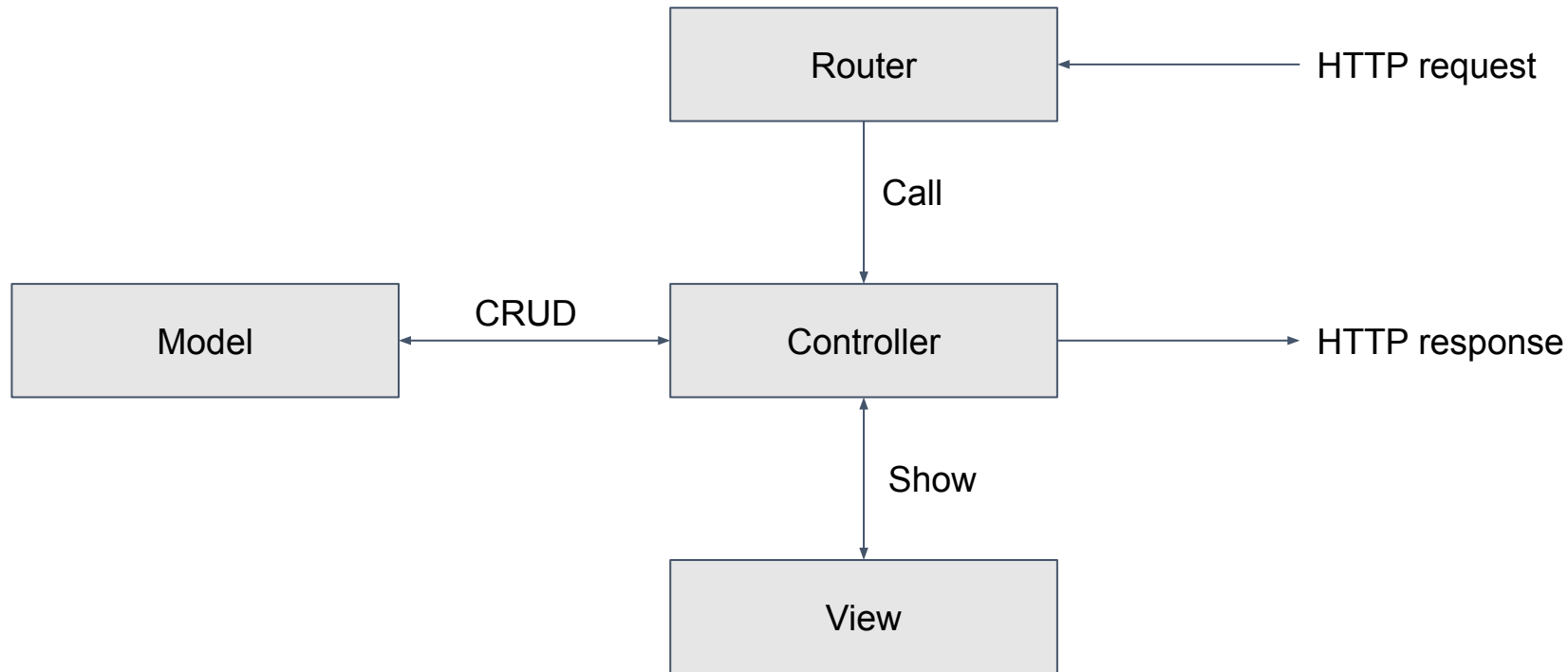


Software Park Thailand
</Code Camp>

Lab 5 : MVC Model



Lab 5 : MVC Model





Lab 5 : MVC Model

```
var express = require('express')
var bodyParser = require('body-parser')
var listRoute = require('./listRoute')

var app = express()
app.use(express.static('public'))
app.use(bodyParser.urlencoded({
  extended: true
}))
app.use('/list', listRoute)

app.listen(3000)
```

app.js

```
var express = require('express')
var router = express.Router()

var list = []

router.get('/', function(req, res) {
  res.send(list)
})
router.post('/', function(req, res) {
  list.push(req.body.data)
  res.send(list)
})

module.exports = router
```

listRoute.js



Software Park Thailand
</Code Camp>

Lab 5 : MVC Model

```
var express = require('express')
var bodyParser = require('body-parser')
var listRoute = require('./listRoute')

var app = express()
app.use(express.static('public'))
app.use(bodyParser.urlencoded({
  extended: true
}))
app.use('/list', listRoute)

app.listen(3000)
```

app.js

```
var express = require('express')
var router = express.Router()

var list = []

router.get('/', function(req, res) {
  res.send(list)
})
router.post('/', function(req, res) {
  list.push(req.body.data)
  res.send(list)
})

module.exports = router
```

listRoute.js



Lab 5 : MVC Model

```
var express = require('express')
var bodyParser = require('body-parser')
var listRoute = require('./listRoute')

var app = express()
app.use(express.static('public'))
app.use(bodyParser.urlencoded({
  extended: true
}))
app.use('/list', listRoute)

app.listen(3000)
```

app.js

```
var express = require('express')
var router = express.Router()

var list = []

router.get('/', function(req, res) {
  res.send(list)
})
router.post('/', function(req, res) {
  list.push(req.body.data)
  res.send(list)
})

module.exports = router
```

listRoute.js



Software Park Thailand
</Code Camp>

Lab 5 : MVC Model

```
var express = require('express')
var bodyParser = require('body-parser')
var listRoute = require('./listRoute')

var app = express()
app.use(express.static('public'))
app.use(bodyParser.urlencoded({
  extended: true
}))
app.use('/list', listRoute)

app.listen(3000)
```

app.js

```
var express = require('express')
var router = express.Router()

var list = []

router.get('/', function(req, res) {
  res.send(list)
})

router.post('/', function(req, res) {
  list.push(req.body.data)
  res.send(list)
})

module.exports = router
```

listRoute.js



Lab 5 : MVC Model

- โจทย์ Lab 5
 - แยกโค้ดนี้ ให้เป็นหลายไฟล์ ตามโครงสร้างของ MVC (+router)

```
var express = require('express')
var bodyParser = require('body-parser')
var listRoute = require('./listRoute')

var app = express()
app.use(express.static('public'))
app.use(bodyParser.urlencoded({
  extended: true
}))
app.use('/list', listRoute)

app.listen(3000)
```

app.js

```
var express = require('express')
var router = express.Router()

var list = []

router.get('/', function(req, res) {
  res.send(list)
})
router.post('/', function(req, res) {
  list.push(req.body.data)
  res.send(list)
})

module.exports = router
```

listRoute.js



Software Park Thailand
</Code Camp>

Outline

- Express คืออะไร
- Lab 1 : Hello, world!
- Lab 2 : Routing
- Lab 3 : Static content กับ middlewares
- Lab 4 : Query and Body
- Lab 5 : MVC model
- **Daily Project**

Homework – Making the API end point as follows., Using Postman for testing.

(<https://chrome.google.com/webstore/detail/postman-rest-client/fdmmgilgnpjigdojojpjoooidkmcomcm>)

1. GET /static -> return “Hello World”
2. GET /staticJSON -> return JSON of {text:”Hello World”}
3. GET /echo?text=Heyyyy -> return any string in text variable, i.e., “Heyyyy”
4. GET /plus?a=3&b=5 -> return the sum of a + b, i.e., 8
5. GET /plusByJSON?jsonText={a:3,b:5} -> return the sum of a+b, parsed the json object from jsonText variable, thus, it should be answer 8
6. GET /plus/6/7 -> return the sum of 6+7, which is 13
7. GET /checkEvenNumber/3

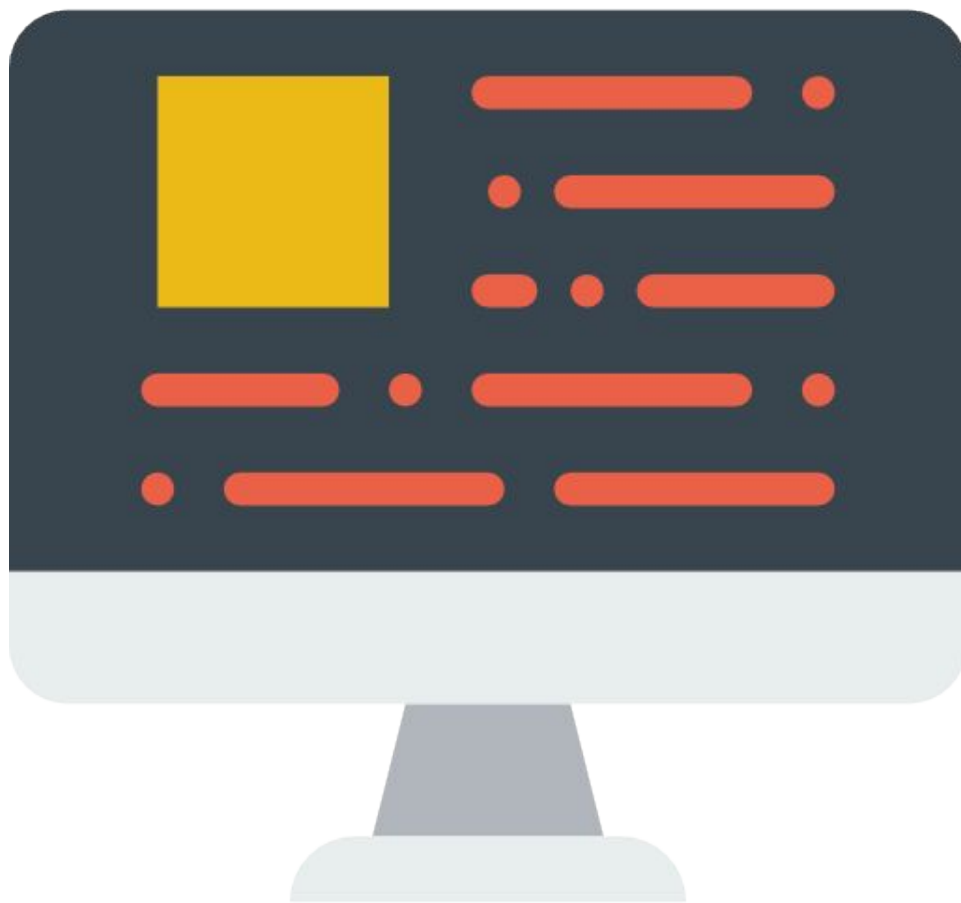
<https://kobkrit.com/learning-guide-for-expressjs-and-exercise-79cead94040a>

Homework

8. Return the Bad input (400) response code if the input number is not even. (in this case, it is 3) thus it is return 400 response code.
9. Return the Success (200) response code if the input number is a even.
10. POST /number/1 for saving a input number, return the all saved numbers as a JSON array. For example
11. POST /number/1 -> [1]
12. POST /number/2 -> [1,2]
13. POST /number/5 -> [1,2,5]
14. DELETE /number/1, remove the number in the array
15. DELETE /number/1 -> [2,5]
16. DELETE /number/2 → [5]
17. PUT /number/5/10, change the number in the array
18. PUT /number/5/10 -> [10]
19. POST /countFields for counting the number of fields that submit via req.body (raw as the JSON object) for example,
20. POST /countFields BODY {"a":1,"b":2,"c":3} -> 3
21. POST /countFields BODY {"a":1,"b":2,"c":3,"d":5} -> 4

Daily Project

- ทำที่เก็บข้อมูลแบบ Key-value (Hint: Map / Object)
- ทำระบบ CRUD
 - Create(From Key create value)
 - Read (From key return value)
 - Update (From key update value)
 - Delete(From key)
- ทำหน้าเว็บนิดนึงด้วยก็ได้ จะได้ทดสอบง่ายขึ้น (front-end / fullstack)



Congrats!