

# หลักสูตร Node.JS Express

## ภาพรวมของการเรียน

Node.JS เป็นโปรแกรม JavaScript ที่ทำคำสั่งภายนอกเว็บเบราว์เซอร์ ช่วยให้ให้นักพัฒนาใช้ JavaScript เขียนคำสั่งฝั่งเซิร์ฟเวอร์ได้ โดยไม่ต้องใช้ภาษาอื่น REST API เป็นอินเทอร์เฟซที่คอมพิวเตอร์สองระบบใช้ในการแลกเปลี่ยนข้อมูลอย่างปลอดภัยผ่านอินเทอร์เน็ต เป็นรูปแบบสถาปัตยกรรมสำหรับสร้างบริการเว็บที่เรียกและส่งข้อมูลผ่านโปรโตคอล HTTP หลักสูตรนี้ผู้เรียนจะได้เรียนการเขียนโปรแกรม JavaScript ขั้นสูง การสร้าง REST API ด้วย Node.JS และ Express.JS การทำงานกับฐานข้อมูล MongoDB จนถึงการแปลงโปรแกรมที่สร้างเสร็จแล้วให้เป็นแอปพลิเคชัน

## ระยะเวลาที่ใช้ในการเรียน

หลักสูตร 18 ชั่วโมง เรียนทั้งหมด 6 ครั้ง (ครั้งละ 3 ชั่วโมง)

## พื้นฐานของผู้เรียน

สามารถเขียนโปรแกรม JavaScript

## โปรแกรมที่ใช้

Visual Studio Code

## หลักสูตรนี้เหมาะสำหรับ

- ผู้ที่ต้องการสร้าง REST API
- โปรแกรมเมอร์
- วิศวกร
- นักศึกษาที่สนใจ

## วิทยากร

อาจารย์ กฤษณ์ อุดมจารุมณี

อาจารย์ประจำหลักสูตร Computer Programming

## คุณวุฒิ :

- ปริญญาตรี วิศวกรรมไฟฟ้า มหาวิทยาลัยสงขลานครินทร์

- ปริญญาโท M.B.A., Drexel University, Pennsylvania, USA
- Microsoft Certified Professional - .NET Framework 3.5 Windows Forms Application.

Node.JS Express Course Outline	
ครั้งที่ 1	<p><b>การเขียนโปรแกรม JavaScript</b></p> <ol style="list-style-type: none"> <li>1. วิธีการกำหนดตัวแปรและค่าคงที่</li> <li>2. ชนิดของตัวแปร (Data Types)</li> <li>3. ตัวดำเนินการ (Operators) <ul style="list-style-type: none"> <li>• ด้านคณิตศาสตร์, ด้านการเปรียบเทียบ, ด้านตรรกะ</li> </ul> </li> <li>4. คำสั่ง document.write, alert, prompt และ confirm</li> <li>5. คำสั่งในการตรวจสอบเงื่อนไข <ul style="list-style-type: none"> <li>• if...else</li> <li>• switch-case</li> </ul> </li> <li>6. คำสั่งในการวนรอบ (Loops) <ul style="list-style-type: none"> <li>• while, do...while,</li> <li>• for loop</li> </ul> </li> <li>7. วิธีสร้างและจัดการข้อมูลชนิดอาร์เรย์ (Array)</li> <li>8. วิธีใช้งานคำสั่งชนิดต่างๆ (Functions) <ul style="list-style-type: none"> <li>• คำสั่งเกี่ยวกับคณิตศาสตร์และตัวเลข,</li> <li>• คำสั่งจัดการข้อความ</li> <li>• คำสั่งเกี่ยวกับวันที่และเวลา</li> </ul> </li> <li>9. วิธีสร้างและเรียกเมธอด (Method)</li> </ol>
ครั้งที่ 2	<ol style="list-style-type: none"> <li>10. การเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming) ด้วย JavaScript <ul style="list-style-type: none"> <li>• วิธีสร้างคลาส (Class)</li> <li>• วิธีสร้างพร็อพเพอร์ตี้ (Property)</li> <li>• วิธีสร้างคอนสตรัคเตอร์ (Constructor)</li> <li>• วิธีสร้างเมธอด (Method)</li> <li>• คำสั่ง Static</li> <li>• ขอบเขตของตัวแปร</li> <li>• วิธีการสืบทอดคลาส (Inheritance)</li> <li>• คำสั่ง Getters กับ Setters</li> </ul> </li> </ol>

	<p>11. วิธีเขียนคำสั่ง Arrow Function</p> <p>12. Asynchronous กับ Promises</p> <ul style="list-style-type: none"> <li>• รูปแบบการจัดการกับโค้ด Asynchronous</li> <li>• วิธีเขียนโค้ด Callbacks</li> <li>• ทำความรู้จักกับ Promises</li> <li>• วิธีแปลง Callbacks เป็น Promises</li> <li>• วิธีใช้คำสั่ง Async และ Await</li> </ul>
ครั้งที่ 3	<p><b>Node.JS</b></p> <p>13. ทำความรู้จักกับ Node.JS</p> <p>14. วิธีติดตั้ง Node.JS</p> <p>15. โมดูล (Module) ของ Node.JS</p> <ul style="list-style-type: none"> <li>• วิธีโหลดและใช้โมดูลใน Node.JS</li> <li>• ตัวแปรและคำสั่งของ Node.JS <ul style="list-style-type: none"> <li>◦ คำสั่ง require</li> <li>◦ คำสั่ง Wrapper</li> </ul> </li> <li>• โมดูลประเภทต่างๆ <ul style="list-style-type: none"> <li>◦ ประเภทเส้นทาง (Path) , ระบบปฏิบัติการ, ระบบไฟล์ และ HTTP</li> </ul> </li> <li>• ทำความเข้าใจกับ Global Object</li> <li>• วิธีสร้างอีเวนต์ (Event)</li> <li>• วิธีสร้างคลาส EventEmitter</li> </ul> <p>16. Node Package Manager (NPM)</p> <ul style="list-style-type: none"> <li>• วิธีติดตั้งและใช้โมดูล NPM</li> <li>• วิธีทำงานกับ Global Packages</li> <li>• วิธีแก้ไขและเผยแพร่ NPM</li> <li>• ทำความรู้จักกับ package.json</li> </ul>
ครั้งที่ 4	<p>17. สร้าง REST API ด้วย Express</p> <ul style="list-style-type: none"> <li>• วิธีสร้างเว็บเซิร์ฟเวอร์ด้วย Express</li> <li>• วิธีใช้งาน Router <ul style="list-style-type: none"> <li>◦ การกำหนดเส้นทาง,</li> <li>◦ กำหนดตัวแปรของเส้นทาง</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>○ เปลี่ยนเส้นทาง (Redirect)</li> <li>• วิธีเพิ่ม Middleware</li> <li>• วิธีใช้คำสั่ง HTTP GET, POST และ PUT</li> <li>• วิธีจัดการกับข้อผิดพลาด</li> <li>• วิธีตรวจสอบความถูกต้องของข้อมูล</li> </ul>
ครั้งที่ 5	<p>18. การทำงานกับฐานข้อมูล MongoDB</p> <ul style="list-style-type: none"> <li>• การติดตั้ง MongoDB</li> <li>• วิธีเชื่อมต่อกับ MongoDB</li> <li>• วิธีกำหนดโครงสร้างและโมเดล</li> <li>• ตัวดำเนินการแบบเปรียบเทียบและแบบตรรกะ</li> <li>• วิธีเรียกข้อมูล (Query)</li> <li>• วิธีเพิ่มข้อมูลไปใน MongoDB</li> <li>• วิธีอ่านและแสดงผลข้อมูลจาก MongoDB</li> <li>• วิธีแก้ไขและลบข้อมูลใน MongoDB</li> <li>• วิธีบันทึกและลบข้อมูล</li> <li>• วิธีแบ่งหน้าเพจ</li> </ul>
ครั้งที่ 6	<p>19. การสร้างความสัมพันธ์ระหว่างข้อมูลด้วย Mongoose</p> <ul style="list-style-type: none"> <li>• วิธีตรวจสอบความถูกต้อง <ul style="list-style-type: none"> <li>○ คำสั่งที่กำหนดมาให้</li> <li>○ คำสั่งที่กำหนดเอง</li> <li>○ คำสั่ง Async</li> </ul> </li> <li>• วิธีตรวจสอบข้อผิดพลาด</li> <li>• วิธีกำหนด Schema</li> <li>• วิธีสร้างแบบจำลองความสัมพันธ์</li> <li>• วิธีใช้อาร์เรย์</li> <li>• วิธีทำ Transaction</li> <li>• วิธีตรวจสอบความถูกต้องของ ID อ็อบเจกต์</li> </ul> <p>20. การแปลงโปรแกรมเป็นแอปพลิเคชัน (Deployment)</p> <ul style="list-style-type: none"> <li>• วิธีใช้งาน Heroku</li> <li>• วิธีเพิ่มโค้ดลงใน Git Repository</li> <li>• วิธีทำแอปพลิเคชันด้วย Heroku</li> </ul>

NetDesign