



Building Microservices with Node.js

Microservices ช่วยให้เราสามารถพัฒนาซอฟต์แวร์เป็นชิ้นส่วนเล็กๆ ที่สามารถทำงานร่วมกันได้ โดยสามารถพัฒนาแยกกัน ซึ่งนับเป็นข้อดีที่องค์กรต่าง ๆ เริ่มหันมาใช้แนวทางการพัฒนาแบบนี้ ในช่วงไม่กี่ปีที่ผ่านมา Node.js ได้รับความนิยมในหมู่นักพัฒนาอย่างสูง และได้กลายมาเป็นมาตรฐานหนึ่งในการพัฒนา Microservice เนื่องจาก Node.js มีประสิทธิภาพ ง่าย และมีความยืดหยุ่นสูง ในหลักสูตรนี้แนะนำแนวทางและสถาปัตยกรรมของ Microservice ที่พัฒนาบน Node.js

วัตถุประสงค์:

- Implement caching strategies to improve performance
- Integrate logs for all microservices in one place
- Execute long-running tasks in asynchronous microservices
- Create sophisticated deploy pipelines
- Make all of this production-ready
- Use a service mesh for resiliency

กลุ่มเป้าหมาย:

- นักเรียนนักศึกษา
- ครู อาจารย์ วิทยากรที่สนใจ
- นักวิชาการ นักไอที หรือผู้ดูแลระบบ
- ตลอดจนผู้สนใจทั่วไปใน Microservices with Node.js

ความรู้พื้นฐาน:

- พื้นฐานภาษา JavaScript
- พื้นฐานการใช้งาน Node.JS
- พื้นฐานการใช้งานฐานข้อมูล
- พื้นฐานการใช้งาน Windows



ระยะเวลาในการอบรม:

- 24 ชั่วโมง/4 วัน

ราคาคอร์สอบรม:

- 8,400 บาท (ราคานี้ยังไม่ได้รวมภาษีมูลค่าเพิ่ม)

วิทยากรผู้สอน:

- อาจารย์สนิทวงศ์ กมลภากรณ์

คอร์สที่ควรอบรมก่อนหน้า:

- Basic Node.js (คอร์ส Node.js พื้นฐาน)

เนื้อหาการอบรม:

Module 1: Design Your Microservices Architecture

What Is a Microservice?

- Define what makes up a microservices based architecture
- Explain why we use microservices
- Understand the key differences between microservices and monolithic applications

Design Principles for Microservice Architectures

- Learn about the key design principles
- Explore how to organize microservices using components of business processes
- See the project management, communication channels to use and governance of the services

Meet Your Class Project

- Learn about the Kona Surf and Snorkel, the application that will be built in the course
- See some basic designs for beginning of the project



- Look at some of the codes that will be used to create the first basis of the application's reporting feature.

Node.js for Microservices with Express

- Explore models, views and controllers
- See how the controllers and models will work within the MVC pattern
- Learn about the MVC, Model View Controller pattern

Design the Microservices Architecture

- Evaluate the steps of the design
- Design the microservices architecture

Module 2: Setting Up a Microservices Environment

- Organization of Code: Git Repos and VS Code
- See Git Repository Styles – Multi and Mono Repo
- Look at the Visual Studio Code

Node.js and Express

- Program Environments for Node.js
- Learn Callbacks, Event Emitters and Promises

MongoDB / MySQL / Oracle

- Install MongoDB
- Install MySQL
- Install Oracle
- Connect to MongoDB in Node.js
- Connect to MySQL with Node.js
- Connect to Oracle with Node.js
- Understand Other Database ideas/solutions



Intro to Vue.js

- Overview of Vue.js
- Create application with Vue.js

Module 3: Building the Remaining Microservices

Handling Callbacks with Event Emitters and Promises

- See Callbacks
- Explore Event Emitters
- Understand Promises

Setting Up the Remaining Microservices

- Configure the ports
- Build the remaining Microservices

Creating an API Gateway

- Build the API Gateway
- Identify alternatives like AWS or load balancers as an API gateway

Module 4: Deploying Microservices

Strategies for Microservice Deployment

- Analyze Multiple Services Per Host
- Apply Single Service Per Host
- Learn about the Container and Serverless Deployment

Deploying Microservices Using PM2

- Install PM2
- Explore Manual Deployment



- Set up the ecosystem.json for deployment

Module 5: Scaling Microservices with NGINX

Scaling Microservices

- Understand Load Balancing
- Look at the Decomposing Apps
- See Data Sharding

Installing NGINX on Ubuntu

- Install NGINX
- Analyze NGINX

Load Balancing with NGINX

- Set up the config file
- Start NGINX
- Test the Load Balancer Configuration

Using NGINX as an API Gateway

- Set up the config file
- Reload NGINX
- Test the API Gateway

Module 6: Monitoring and Alerting

Monitoring with PM2 and Keymetrics

- Run PM2 Monitoring from the Command Line
- Register for Keymetrics
- Monitor microservices with Keymetrics



Creating a Monitoring Microservice

- Explore Event Emitter Service Pattern
- Review the code of the Monitoring Microservice

Alerting Upon Microservice Failure

- Set up alerting for the monitoring service
- Review the code of the Send Alert promise

Uptime Robot as an Off-Site Monitoring Service

- Look at the uptime robot overview
- Learn uptime robot register and configure it
- Explore uptime robot Dashboard and alerts configuration