



สถาบันไอทีจีเนียส
IT GENIUS INSTITUTE

COURSE ID

WDC-112

COURSE OUTLINE · รายละเอียดหลักสูตรอบรม

WEB DEVELOPMENT / PYTHON

Building Microservices with Python

สร้าง Microservices ด้วย Python และ FastAPI ที่สื่อสารกันได้และพร้อมใช้งานจริง

18 ชั่วโมง

3 วัน

พื้นฐาน Python และ REST API เบื้องต้น

ระดับ Intermediate



ระยะเวลา: 3 วัน

18 ชั่วโมง (3 วัน วันละ 6 ชม.) · 09:00-16:00 น.



รูปแบบ: Classroom / Hybrid

Onsite หรือ Online ผ่าน Microsoft Teams · Public / In-house



รายละเอียดหลักสูตร

www.itgenius.co.th

Microservices คือการแบ่งระบบออกเป็น Service ขนาดเล็กหลายตัว แต่ละตัวดูแลงานของตัวเอง มีฐานข้อมูลของตัวเอง และพัฒนา ทดสอบ และ Deploy แยกกันได้ แต่ยังทำงานร่วมกันเป็นระบบเดียว แนวทางนี้ช่วยให้หลายทีมทำงานคู่ขนานและขยายเฉพาะส่วนที่มีการสูงได้ แต่ก็ตามมาด้วยเรื่องที่ต้องจัดการ เช่น การสื่อสารระหว่าง Service ข้อมูลที่กระจายอยู่หลายที่ และการติดตามปัญหาข้ามระบบ Python กับ FastAPI เป็นตัวเลือกที่นิยมเพราะเขียนเร็ว ง่าย และรองรับงานแบบ Async

หลักสูตรนี้พาสร้าง Microservices ด้วย FastAPI ตั้งแต่แนวคิดและการแบ่ง Service สร้าง REST API ที่ตรวจข้อมูลด้วย Pydantic ต่อ PostgreSQL ด้วย SQLAlchemy 2.0 เลือกรูปแบบการจัดการข้อมูล รับ Service ด้วย Docker Compose ให้เรียกกันผ่าน HTTP และผ่าน RabbitMQ ด้วย FastStream ใช้ Nginx เป็นทางเข้าเดียว ยืนยันตัวตนด้วย JWT ทำ Health Check, Log และ Tracing เขียน Test แล้วรวมทั้งระบบใน Workshop (อบรม 3 วัน วันละ 6 ชั่วโมง รวม 18 ชั่วโมง ระดับ Intermediate)

เครื่องมือ



Python 3.14



FastAPI



PostgreSQL 18



RabbitMQ



Docker Compose

ราคาค่าอบรม · PUBLIC / IN-HOUSE

9,500 บาท / ท่าน

ราคายังไม่รวมภาษีมูลค่าเพิ่ม (VAT 7%) · รวมเอกสารประกอบการอบรม Lab Guide ซอร์สโค้ดครบทุก Lab ไฟล์ docker-compose ของทั้งระบบ และ Certificate · รับจัด In-house ขอใบเสนอราคาได้

สมัครผ่านเว็บไซต์ ลด 10%



เพิ่มเพื่อน
LINE

☎ 02-570-8449

☎ 088-807-9770

LINE @itgenius

🌐 www.itgenius.co.th



01 วัตถุประสงค์ Objectives

- เข้าใจข้อดี ข้อเสีย และการแบ่งระบบเป็น Microservices ตามขอบเขตงาน
- สร้าง REST API ด้วย FastAPI และ PostgreSQL ได้
- เลือกรูปแบบการจัดการข้อมูล เช่น Database per Service และ API Composition ได้
- รับ Microservices หลายตัวด้วย Docker Compose ได้
- จัดการทางเข้าและการกระจายภาระด้วย Nginx ได้
- สื่อสารระหว่าง Service ทั้งแบบ HTTP และผ่าน Message Broker ได้
- ยืนยันตัวตนข้าม Service และติดตามระบบด้วย Log และ Tracing ได้
- เขียน Test และสร้าง Pipeline สำหรับ Build Image ได้

02 กลุ่มเป้าหมาย Target Audience

- นักพัฒนา Python ที่ต้องการแยกระบบเป็น Microservices
- นักไอทีและผู้ดูแลระบบที่ต้องรับและดูแลระบบหลาย Service
- นักวิชาการและผู้สอนที่ต้องการเข้าใจสถาปัตยกรรม Microservices
- ทีมที่มีระบบ Monolith และกำลังวางแผนแยกระบบ

03 จุดเด่นของหลักสูตร Highlights

- **FastAPI แบบปัจจุบัน** Pydantic v2, Async และ Lifespan
- **ข้อมูลแยกตาม Service** PostgreSQL 18 และ SQLAlchemy 2.0
- **สื่อสารได้ 2 แบบ** HTTP ด้วย httpx และ RabbitMQ
- **ทางเข้าเดียวด้วย Nginx** Routing และ Load Balancing
- **พร้อมดูแลระบบจริง** JWT, Health Check และ Tracing

04 ความรู้พื้นฐาน Prerequisites

- เขียนภาษา Python ได้ และเข้าใจฟังก์ชันกับ Class เบื้องต้น
- เข้าใจการทำงานของเว็บแอปพลิเคชันและ HTTP เบื้องต้น
- ใช้ฐานข้อมูลและคำสั่ง SQL เบื้องต้นได้
- นำโน้ตบุ๊กที่ติดตั้ง Docker Desktop ได้มาด้วย

05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

รายการ	รายละเอียด
อุปกรณ์	Notebook RAM 16 GB ขึ้นไป เชื่อมต่ออินเทอร์เน็ต พร้อมสักร์ติดตั้งโปรแกรม
ภาษาและเครื่องมือ	Python 3.14, uv, VS Code, Git และ Bruno หรือ Postman
Framework และไลบรารี	FastAPI, Pydantic v2, SQLAlchemy 2.0, Alembic, httpx และ FastStream
บริการบน Docker	PostgreSQL 18, RabbitMQ, Nginx และ Jaeger
Test และ CI	pytest และ GitHub Actions

หมายเหตุ: เครื่องมือทั้งหมดเป็นโอเพนซอร์สหรือใช้ฟรี หลักสูตรนี้ปรับปรุงใหม่ปี 2026 ขยายจากเดิมที่สร้าง Movies และ Casts Service แล้วรวมด้วย Nginx ให้ครบงาน Microservices จริง ใช้ FastAPI รุ่นปัจจุบันที่จัดการ Startup และ Shutdown ด้วย Lifespan แทน on_event ที่เลิกแนะนำแล้ว ใช้ SQLAlchemy 2.0 แบบ Async กับ Alembic และจัดการโปรเจกต์ด้วย uv เพิ่มการสื่อสารผ่าน RabbitMQ ด้วย FastStream, Saga และ Outbox, JWT, OpenTelemetry, Test และ CI ภาพรวม Kubernetes อยู่ในระดับแนะนำเท่านั้น หลักสูตรนี้เป็น Python ต่างจาก Building Microservices with NestJS และ Microservices with Spring Boot

06 ภาพรวมการเรียนรู้ 3 วัน Big Picture

วัน	หัวข้อหลัก
Day 1 Section 1-5	เข้าใจ Microservices และการแบ่ง Service, เตรียมเครื่องมือด้วย uv และ Docker, สร้าง REST API ด้วย FastAPI และ Pydantic, ต่อ PostgreSQL ด้วย SQLAlchemy 2.0 และเลือกรูปแบบการจัดการข้อมูล
Day 2 Section 6-10	รับ Movies และ Casts Service ด้วย Docker Compose, เรียกกันด้วย httpx พร้อม Timeout และ Retry, วาง Nginx เป็น API Gateway, สื่อสารผ่าน RabbitMQ ด้วย FastStream และใช้ Saga กับ Outbox
Day 3 Section 11-15	ยืนยันตัวตนข้าม Service ด้วย JWT, Health Check, Log และ Tracing ด้วย OpenTelemetry, เขียน Test ด้วย pytest, สร้าง Image ด้วย GitHub Actions แล้วรวมทั้งระบบใน Workshop



DAY
1

แนวคิด Microservices และสร้าง Service แรกด้วย FastAPI

เนื้อหาการอบรม · Section 1-5

รวม
6 ชั่วโมง

1 รู้จัก Microservices

- Monolith เทียบกับ Microservices ข้อดีและข้อเสียที่ต้องรับมือ
- เมื่อไรควรแยกระบบ และเมื่อไรยังไม่ควร
- แบ่ง Service ตามขอบเขตงาน (Bounded Context)
- ทำไมต้อง Python และ FastAPI สำหรับ Microservices
- Workshop: แบ่งระบบจองตั๋วภาพยนตร์ออกเป็น Service

2 Lab: เตรียมเครื่องมือ

- ติดตั้ง uv และสร้างโปรเจกต์ Python 3.14
- ติดตั้ง Docker Desktop และทดสอบ Docker Compose
- โครงสร้าง Repository ที่มีหลาย Service
- รับ FastAPI ด้วยคำสั่ง fastapi dev และดูเอกสาร API อัตโนมัติ
- Lab: Service แรกที่ตอบ Health Check

3 Lab: REST API ด้วย FastAPI

- Path, Query และ Request Body พร้อม Type Hint
- ตรวจสอบข้อมูลด้วย Pydantic v2 และแยก Schema รับกับ Schema ส่ง
- แยกโครงสร้างด้วย APIRouter และ Dependency Injection
- จัดการ Error และ Status Code ให้เป็นมาตรฐาน
- Lab: Movies API ที่เพิ่ม ค้นหา แก้ไข และลบได้

4 Lab: ใช้ PostgreSQL กับ FastAPI

- รับ PostgreSQL 18 บน Docker
- SQLAlchemy 2.0 แบบ Async และการจัดการ Session ผ่าน Dependency
- สร้างและอัปเดตตารางด้วย Alembic Migration
- เปิดและปิดการเชื่อมต่อด้วย Lifespan ของ FastAPI
- Lab: Movies Service ที่เก็บข้อมูลลง PostgreSQL

5 รูปแบบการจัดการข้อมูลของ Microservice

- แยกฐานข้อมูลต่อ Service (Database per Service)
- ใช้ฐานข้อมูลร่วมกัน (Shared Database) และความเสี่ยง
- รวมข้อมูลผ่าน API (API Composition)
- การคัดลอกข้อมูลบางส่วนและ Eventual Consistency
- Workshop: ออกแบบข้อมูลของ Movies และ Casts Service



DAY
2

รับหลาย Service และให้สื่อสารกัน

เนื้อหาการอบรม · Section 6-10

รวม
6 ชั่วโมง

6 Lab: Microservice uu Docker

- เขียน Dockerfile สำหรับ FastAPI ที่ใช้ uv
- สร้าง Casts Service พร้อมฐานข้อมูลของตัวเอง
- รวม Movies, Casts และฐานข้อมูลด้วย Docker Compose
- Network, Volume, Environment Variable และ healthcheck
- Lab: รับทั้งสอง Service ด้วยคำสั่งเดียว

7 Lab: เรียกกันผ่าน HTTP

- เรียก Service อื่นด้วย httpx แบบ Async
- Timeout และ Retry อย่างมีขอบเขต
- แนวคิด Circuit Breaker และการตอบกลับเมื่อปลายทางล่ม
- API Composition รวมข้อมูลภาพยนตร์กับนักแสดง
- Lab: หน้ารายละเอียดภาพยนตร์ที่ยังตอบได้แม้ Casts ล่ม

8 Lab: Nginx เป็น API Gateway

- ใช้ Nginx ให้เข้าถึงทุก Service ผ่าน Host Address เดียว
- กำหนดเส้นทางตาม Path ไปยังแต่ละ Service
- Load Balancing เมื่อรับ Service หลาย Instance
- จำกัดอัตราการเรียก (Rate Limit) และ Header ที่ส่งต่อ
- Lab: Gateway ของระบบจองตั๋วภาพยนตร์

9 Lab: สื่อสารผ่าน RabbitMQ

- การสื่อสารแบบ Synchronous เทียบกับ Asynchronous
- Queue, Exchange และ Routing ของ RabbitMQ
- เขียน Publisher และ Subscriber ด้วย FastStream
- ส่ง Event เมื่อข้อมูลเปลี่ยนและรับใน Service อื่น
- Lab: Booking Service ส่ง Event ให้ Notification Service

10 Transaction ข้าม Service

- ทำไมไม่ใช้ Transaction เดียวข้ามหลายฐานข้อมูลไม่ได้
- Saga แบบ Choreography และ Orchestration
- Outbox Pattern ป้องกัน Event หาย
- Idempotency และการรับ Message ซ้ำ
- Lab: จองที่นั่งและคืนที่นั่งเมื่อชำระเงินไม่สำเร็จ



DAY
3

ความปลอดภัย การติดตามระบบ Test และ Workshop

เนื้อหาการอบรม · Section 11-15

รวม
6 ชั่วโมง

11 Lab: ยืนยันตัวตนข้าม Service

- Auth Service ออก Token ด้วย OAuth2 และ JWT
- ตรวจสอบ Token ที่ Gateway หรือในแต่ละ Service
- ส่งข้อมูลผู้ใช้ต่อระหว่าง Service อย่างปลอดภัย
- เก็บค่าลับและรหัสผ่านอย่างถูกวิธี
- Lab: ป้องกัน Booking API ให้ใช้ได้เฉพาะผู้ที่เข้าสู่ระบบ

12 Lab: ติดตามระบบ

- Health Check และ Readiness ของแต่ละ Service
- Structured Log และ Correlation ID ข้าม Service
- Tracing ด้วย OpenTelemetry และดูเส้นทาง Request ใน Jaeger
- Metrics ที่ควรเก็บ เช่น จำนวน Request และเวลาตอบสนอง
- Lab: หาว่า Request ที่ค้างติดอยู่ที่ Service ไหน

13 Test สำหรับ Microservices

- Unit Test และ API Test ด้วย pytest และ TestClient
- แทนที่ Dependency ด้วย dependency_overrides
- Integration Test กับฐานข้อมูลบน Docker
- Contract Test ป้องกัน API เปลี่ยนแล้ว Service อื่นพัง
- Lab: ชุด Test ของ Movies และ Booking Service

14 Build และ Deploy

- สร้าง Image ทุก Service อัตโนมัติด้วย GitHub Actions
- ตั้ง Tag และเก็บ Image ใน Container Registry
- Docker Compose สำหรับเซิร์ฟเวอร์จริงและการอัปเดตทีละ Service
- ภาพรวมเมื่อระบบโตขึ้น เช่น Kubernetes
- Lab: Pipeline ที่รัน Test แล้ว Build Image

15 Workshop: ระบบจองตั๋วภาพยนตร์

- รวม Movies, Casts, Booking, Auth และ Notification Service
- เชื่อมทุกอย่างผ่าน Nginx, RabbitMQ และ PostgreSQL
- ทดสอบกรณีปกติและกรณีที่ Service ล่ม
- ใช้ AI Assistant ช่วยอ่าน Log และอธิบายโค้ด โดยตรวจสอบผลเองทุกครั้ง
- Workshop: นำเสนอสถาปัตยกรรมและสาธิตระบบที่รันได้จริง

07 เรียนจบแล้วได้อะไร Outcomes

- ออกแบบ **Microservices** ได้ แบ่ง Service และวางข้อมูล
- สร้าง Service ด้วย **FastAPI** Pydantic, SQLAlchemy 2.0 และ Alembic
- รันหลาย Service ได้ Docker Compose และ Nginx
- ให้ Service สื่อสารกัน httpx, RabbitMQ และ Saga
- ปลอดภัยและติดตามได้ JWT, Log และ OpenTelemetry
- ส่งมอบอย่างมั่นใจ pytest และ GitHub Actions

08 สิ่งที่คุณเรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรมครบทั้ง 3 วัน และ Lab Guide ทีละขั้นตอน
- ซอร์สโค้ดระบบจองตั๋วภาพยนตร์แบบ Microservices ครบทุก Lab
- ไฟล์ docker-compose, คำตั้ง Nginx และ Workflow ของ GitHub Actions
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion) และช่องทางสอบถามวิทยากรหลังจบหลักสูตร

09 ข้อมูลผู้สอน Instructor

อ.สาธิต โทยม ผู้สอนและนักพัฒนาที่ใช้ Python และ FastAPI สร้างระบบหลาย Service และดูแลแบบ Docker สอนผ่านระบบจองตั๋วภาพยนตร์ชุดเดียวที่เพิ่ม Service ทีละตัว อธิบายข้อดีข้อเสียของแต่ละทางเลือก ชี้ปัญหาที่เจอเมื่อระบบแยกเป็นหลายส่วน และพาลงมือจนผู้เรียนรันและแก้ปัญหาระบบได้เอง

ราคาค่าอบรม 9,500 บาท/ท่าน · สนใจ IN-HOUSE ขอใบเสนอราคา

9,500 บาท / ท่าน (ยังไม่รวม VAT 7%)

สมัครผ่านเว็บไซต์ ลด 10%

☎ 02-570-8449

📞 @itgenius

☎ 088-807-9770

✉ contact@itgenius.co.th



เพิ่มเพื่อน
LINE