



สถาบันไอทีจีเนียส  
IT GENIUS INSTITUTE

COURSE ID

WDC-119

COURSE OUTLINE · รายละเอียดหลักสูตรอบรม

WEB DEVELOPMENT / PYTHON

# Python Microservices with Generative AI

สร้างระบบ Microservices ด้วย Python ที่มี AI Assistant ทำงานร่วมได้จริง

- 24 ชั่วโมง
- 4 วัน
- พื้นฐาน Python, REST API, SQL และ Docker เบื้องต้น
- ระดับ Intermediate



ระยะเวลา: 4 วัน

24 ชั่วโมง (4 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](http://www.itgenius.co.th)

องค์กรจำนวนมากไม่ได้ต้องการเพียงระบบที่แบ่งเป็น Service ย่อยเพื่อขยายงานได้ง่าย แต่ยังต้องการนำ Generative AI เข้ามาเป็นส่วนหนึ่งของระบบ เช่น ผู้ช่วยตอบคำถามลูกค้า การค้นหาเอกสารตามความหมาย หรือการสรุปข้อมูลอัตโนมัติ ปัญหาที่ทีมพัฒนาเจอบ่อยคือแบ่ง Service ผิดขอบเขต การสื่อสารระหว่าง Service ที่เปราะบาง และการเรียก LLM กระจัดกระจายจนคุมต้นทุน ความปลอดภัย และการไล่หาปัญหาไม่ได้

หลักสูตรนี้พาผู้เรียนสร้างระบบร้านค้าจำลอง SiamCart ต่อเนื่องตลอด 4 วัน เริ่มจากกำหนดขอบเขต Service ด้วย Bounded Context สร้าง Service ด้วย FastAPI และ Pydantic สื่อสารข้าม Service ด้วย REST, gRPC และ RabbitMQ พร้อม Outbox และ Saga จากนั้นเพิ่ม AI Assistant Service ที่สลับใช้ Cloud LLM หรือ Ollama บนเครื่องได้ ทำ RAG ด้วย PostgreSQL 18 และ pgvector ให้ LLM เรียกข้อมูลคำสั่งซื้อผ่าน Tool Calling แล้วติดตามทั้งระบบด้วย OpenTelemetry จนรันได้ครบด้วย Docker Compose (อบรม 4 วัน วันละ 6 ชั่วโมง รวม 24 ชั่วโมง ระดับ Intermediate)

เครื่องมือ

FA

FastAPI

RM

RabbitMQ

gR

gRPC

pg

pgvector

OI

Ollama

ราคาค่าอบรม · PUBLIC / IN-HOUSE

**11,900** บาท / ท่าน

ราคายังไม่รวมภาษีมูลค่าเพิ่ม (VAT 7%) · รวมเอกสารประกอบการอบรม Lab Guide รายวัน ซอร์สโค้ดโปรเจกต์ SiamCart ไฟล์ Docker Compose และ Certificate · รับจัด In-house ขอใบเสนอราคาได้

สมัครผ่านเว็บไซต์ ลด 10%



เพิ่มเพื่อน  
LINE

☎ 02-570-8449

☎ 088-807-9770

💬 @itgenius

🌐 [www.itgenius.co.th](http://www.itgenius.co.th)

## 01 วัตถุประสงค์ Objectives

- เข้าใจข้อแลกเปลี่ยนของ Microservices เทียบกับ Monolith และ Modular Monolith และรู้ว่าเมื่อใดไม่ควรแยก Service
- แบ่งขอบเขต Service ด้วย Domain-Driven Design, Bounded Context และ Context Map ได้อย่างมีประสิทธิภาพ
- พัฒนา Service ด้วย FastAPI และ Pydantic ตามโครงสร้างที่ดูและทดสอบได้ง่าย พร้อมฐานข้อมูลแยกต่อ Service
- ออกแบบการสื่อสารแบบ REST, gRPC และ Event-driven ผ่าน RabbitMQ ให้เหมาะสมกับสถานการณ์
- ใช้ Timeout, Retry, Circuit Breaker, Idempotency, Outbox และ Saga รับมือความล้มเหลวและความสับสนของข้อมูล
- สร้าง AI Service ที่เรียก LLM ได้ทั้ง Cloud API และ Ollama ทำ RAG ด้วย pgvector และ Tool Calling อย่างปลอดภัย
- ติดตามระบบแบบกระจายด้วย OpenTelemetry ครอบคลุม Trace, Metric, Log และการเรียก LLM
- ทดสอบด้วย pytest และ Testcontainers และเตรียมระบบตาม Checklist ก่อนขึ้น Production

## 02 กลุ่มเป้าหมาย Target Audience

- นักพัฒนา Backend ที่ใช้ Python และต้องการก้าวสู่การออกแบบระบบแบบ Microservices
- นักพัฒนาที่ได้รับมอบหมายให้เชื่อม Generative AI เข้ากับระบบงานเดิมขององค์กร
- Tech Lead และ Software Architect ที่ต้องวางโครงสร้าง Service และ AI Service ให้ดูแลได้ระยะยาว
- ทีม DevOps หรือ Platform ที่อยากเข้าใจมุมมองนักพัฒนาในการสร้าง Service ที่พร้อม Deploy และติดตามได้

## 03 จุดเด่นของหลักสูตร Highlights

- **Microservices + GenAI ในระบบเดียว** AI Service เป็นส่วนหนึ่งของสถาปัตยกรรม
- **สลับ Cloud LLM กับ Local LLM** LLM Client เดียว เปลี่ยนด้วย Config
- **ฝึกรับมือความล้มเหลวจริง** Service ล้ม Message เข้า LLM ตอบช้า
- **เห็นทุก Request ใน Trace เดียว** OpenTelemetry, Jaeger และ Grafana
- **โปรเจกต์เดียวตลอด 4 วัน** SiamCart รับครบด้วย Docker Compose

#### 04 ความรู้พื้นฐาน Prerequisites

- เขียน Python ได้ในระดับใช้งาน เข้าใจ Function, Class, Type Hints และ async/await เบื้องต้น
- เข้าใจ HTTP, REST API และ JSON และใช้คำสั่ง SQL พื้นฐานได้
- ใช้ Git และ Command Line ได้ และเคยรันคอนเทนเนอร์ด้วย Docker มาก่อน
- ไม่จำเป็นต้องมีพื้นฐาน Machine Learning หรือ AI มาก่อน หลักสูตรปูแนวคิด LLM, Embedding และ RAG ที่จำเป็นให้

#### 05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

| รายการ                 | รายละเอียด                                                                                                                                       |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| ภาษาและ Framework      | Python 3.14 ขึ้นไป, uv, Ruff, FastAPI, Pydantic 2, pydantic-settings และ Uvicorn                                                                 |
| การสื่อสาร             | HTTPX, gRPC และ Protocol Buffers, RabbitMQ 4 (aio-pika), Tenacity และ Redis หรือ Valkey                                                          |
| ข้อมูลและ AI           | PostgreSQL 18, SQLAlchemy 2 (async), Alembic, pgvector, Ollama และ LLM API แบบ OpenAI-compatible                                                 |
| Observability และทดสอบ | OpenTelemetry, Jaeger, Prometheus, Grafana, Traefik, pytest และ Testcontainers                                                                   |
| อุปกรณ์ผู้เรียน        | Notebook RAM 16 GB ขึ้นไป (แนะนำ 32 GB หากรัน Local LLM) พื้นที่ว่าง 40 GB พร้อม Docker Desktop หรือ Docker Engine, Git และ VS Code หรือ PyCharm |

**หมายเหตุ:** ต่อยอดจากหลักสูตร Building Microservices with Python (3 วัน) โดยเพิ่ม AI Service, RAG และ Tool Calling เข้าไปในสถาปัตยกรรม หากต้องการเจาะลึก API ตัวเดียวให้ได้มาตรฐานองค์กร ดูหลักสูตร Enterprise REST API with Python · ผู้เรียนใช้ Notebook ของตนเอง Cloud LLM ใช้ API Key ของผู้เรียนหรือองค์กรซึ่งมีค่าใช้จ่ายตามการใช้งาน หรือใช้ Ollama บนเครื่องแทนได้โดยไม่มีค่าใช้จ่าย ชื่อโมเดลและฟีเจอร์ของเครื่องมือเปลี่ยนแปลงบ่อย เนื้อหาจะอิงเวอร์ชันล่าสุด ณ วันอบรม

#### 06 ภาพรวมการเรียนรู้ 4 วัน Big Picture

| วัน                           | หัวข้อหลัก                                                                                                                                                              |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Day 1</b><br>Section 1-5   | Monolith กับ Microservices, แบ่งขอบเขต Service ด้วย DDD และ Event Storming, โครงสร้าง Service ด้วย FastAPI, Database per Service และรันหลาย Service ด้วย Docker Compose |
| <b>Day 2</b><br>Section 6-10  | REST และ gRPC, Retry, Circuit Breaker และ Idempotency, Event-driven ด้วย RabbitMQ, Transactional Outbox และ Saga, Cache และ Read Model                                  |
| <b>Day 3</b><br>Section 11-15 | พื้นฐาน LLM สำหรับนักพัฒนา, AI Service ที่สลับ Cloud LLM กับ Ollama ได้, RAG ด้วย pgvector, Tool Calling และ MCP, Guardrails และการคุมต้นทุน Token                      |
| <b>Day 4</b><br>Section 16-20 | OpenTelemetry ข้าม Service และการเรียก LLM, JWT และ Traefik, ทดสอบด้วย pytest และ Testcontainers, แนวทางสู่ Production และ Master Workshop                              |

DAY  
1

## Microservices Architecture และ Service Design

เนื้อหาการอบรม · Section 1-5

รวม  
6 ชั่วโมง

### 1 Microservices: เมื่อใดควรใช้และเมื่อใดไม่ควร

- เปรียบเทียบ Monolith, Modular Monolith และ Microservices ในมุมมอง การ Deploy และความซับซ้อน
- ลักษณะของ Service ที่ดี: Loose Coupling, High Cohesion และ Deploy แยกกันได้
- ต้นทุนแฝง: Network Latency, Data Consistency และภาระด้าน Operation
- กรณีศึกษา: ระบบที่ควรเริ่มจาก Modular Monolith ก่อนแยก Service

### 2 แบ่งขอบเขต Service ด้วย Domain-Driven Design

- Ubiquitous Language, Subdomain และ Bounded Context
- Event Storming แบบย่อเพื่อค้นหา Domain Event และขอบเขต Service
- Context Map และความสัมพันธ์แบบ Customer/Supplier และ Anti-corruption Layer
- Workshop: ทำ Event Storming และวาด Context Map ของ SiamCart

### 3 Lab: โครงสร้าง Service ด้วย FastAPI

- ตั้งโปรเจกต์ด้วย uv และ Ruff และแบ่งชั้น Router, Service และ Repository
- กำหนด Schema ด้วย Pydantic และจัดการ Config ด้วย pydantic-settings
- Dependency Injection ของ FastAPI, Health Check และ Readiness Endpoint
- Lab: สร้าง Catalog Service ตามโครงสร้างมาตรฐานของทีม

### 4 Lab: Database per Service

- หลักการ Database per Service และผลการ Query ข้าม Service
- SQLAlchemy 2 แบบ async กับ PostgreSQL 18 และ Migration ด้วย Alembic
- เก็บ Snapshot ข้อมูลสินค้าในคำสั่งซื้อแทนการ JOIN ข้ามฐานข้อมูล
- Lab: สร้าง Order Service พร้อมฐานข้อมูลและ Migration ของตนเอง

### 5 Workshop Day 1: รัน SiamCart ด้วย Docker Compose

- Dockerfile แบบ Multi-stage สำหรับ Python Service ที่ใช้ uv
- Docker Compose หลาย Service พร้อม Network, Volume และ Health Check
- แยก Config ตามสภาพแวดล้อมด้วย Environment Variable
- Workshop: รัน Catalog Service, Order Service และ PostgreSQL แยกฐานข้อมูลด้วยคำสั่งเดียว

DAY  
2

## Service Communication: REST, gRPC และ Event-driven

เนื้อหาการอบรม · Section 6-10

รวม  
6 ชั่วโมง

### 6 Lab: Synchronous ด้วย REST และ gRPC

- เรียก Service อื่นด้วย HTTPX แบบ async และกำหนด Timeout ที่เหมาะสม
- ออกแบบ Contract ด้วย Protocol Buffers และสร้าง gRPC Server และ Client ด้วย Python
- เปรียบเทียบ REST กับ gRPC ด้านประสิทธิภาพ การ Versioning และงานภายในระบบ
- Lab: ให้ Order Service ตรวจสอบสินค้าและราคาจาก Catalog ผ่าน gRPC

### 7 Resilience Patterns

- Retry แบบ Exponential Backoff ด้วย Tenacity และความเสี่ยงของ Retry Storm
- Circuit Breaker, Bulkhead และ Fallback เมื่อ Service ปลายทางไม่พร้อม
- Idempotency Key เพื่อป้องกันการทำการซ้ำ
- Lab: ปิด Catalog Service ระหว่างสั่งซื้อแล้วสังเกตพฤติกรรมของระบบ

### 8 Lab: Event-driven ด้วย RabbitMQ

- Event, Command และ Message Broker พร้อมเปรียบเทียบแนวคิด RabbitMQ กับ Apache Kafka
- Exchange, Queue, Routing Key, Quorum Queue และ Dead Letter Queue ใน RabbitMQ 4
- Publisher และ Consumer แบบ async ด้วย aio-pika และการจัดการ Acknowledgement
- Lab: ส่ง Event OrderPlaced ไปยัง Notification Service

### 9 Data Consistency ในระบบกระจาย

- ปัญหา Dual Write และการแก้ด้วย Transactional Outbox
- Saga แบบ Choreography และ Orchestration พร้อม Compensating Transaction
- Consumer ที่รับมือ Message ซ้ำและ Message มาไม่เรียงลำดับ
- Cache ด้วย Redis หรือ Valkey, Cache Invalidation และ Read Model แบบ CQRS เบื้องต้น

### 10 Workshop Day 2: ระบบสั่งซื้อข้าม Service

- เพิ่ม Payment Service (จำลอง) และ Notification Service ที่ทำงานผ่าน Event
- ส่ง Event อย่างน่าเชื่อถือด้วย Outbox
- Saga ยกเลิกคำสั่งซื้ออัตโนมัติเมื่อชำระเงินล้มเหลว
- Workshop: ทดสอบส่ง Message ซ้ำและ Consumer ล่มกลางคัน แล้วตรวจสอบว่าข้อมูลยังถูกต้อง

DAY  
3

## Generative AI Services: LLM, RAG และ Tool Calling

เนื้อหาการอบรม · Section 11-15

รวม  
6 ชั่วโมง

### 11 พื้นฐาน LLM สำหรับนักพัฒนา

- Token, Context Window และ Temperature กับผลต่อคุณภาพและต้นทุน
- Chat Messages, System Prompt และ Structured Output ที่ตรวจด้วย Pydantic
- Cloud LLM กับ Local LLM ด้านความเป็นส่วนตัวของข้อมูล ค่าใช้จ่าย และประสิทธิภาพ
- Lab: เรียก LLM ให้ตอบเป็น JSON ตาม Schema ที่กำหนด

### 12 Lab: AI Service และ LLM Client

- แยก AI Service ออกจาก Business Service และกำหนด Contract ให้ชัด
- LLM Client Abstraction ที่สลับ Cloud API แบบ OpenAI-compatible กับ Ollama ด้วย Config
- รับ Ollama ใน Docker Compose และเลือก Chat Model กับ Embedding Model ให้เหมาะกับเครื่อง
- Streaming Response ผ่าน Server-Sent Events (SSE) จาก FastAPI
- Lab: สร้าง AI Assistant Service ที่ตอบแบบ Streaming

### 13 Lab: RAG ด้วย pgvector

- Embedding, Chunking และ Similarity Search
- ติดตั้ง pgvector บน PostgreSQL 18 สร้างตาราง Vector และ Index แบบ HNSW
- Ingestion Pipeline ที่รับเอกสารผ่าน Event แล้วสร้าง Embedding แบบ Asynchronous
- ประกอบ Prompt จากผลค้นหา อ้างอิงแหล่งที่มา และวัดคุณภาพคำตอบเบื้องต้น
- Lab: นำเอกสารนโยบายร้าน (ข้อมูลจำลอง) เข้า pgvector

### 14 Tool Calling, MCP และความปลอดภัยของ AI Service

- Function/Tool Calling และการกำหนด Schema ของ Tool
- เชื่อม Tool เข้ากับ Order Service โดยส่งต่อสิทธิ์ของผู้ใช้อย่างถูกต้อง
- แนวคิด Model Context Protocol (MCP) และการเปิด Service เป็น MCP Tool (สาธิต)
- Prompt Injection, Data Leakage, การจำกัด Tool และปิดบังข้อมูลส่วนบุคคล
- Timeout, Rate Limit, Fallback Model และบันทึกจำนวน Token เพื่อคุมต้นทุน

### 15 Workshop Day 3: AI Assistant ของ SiamCart

- ตอบคำถามนโยบายร้านแบบ RAG พร้อมอ้างอิงแหล่งที่มา
- ให้ LLM เรียกดูสถานะคำสั่งซื้อของผู้ใช้จาก Order Service ผ่าน Tool Calling
- ทดลองสลับระหว่าง Cloud LLM และ Ollama แล้วเปรียบเทียบคุณภาพ ความเร็ว และต้นทุน
- Workshop: ทดสอบ Prompt Injection ที่พยายามดูคำสั่งซื้อของผู้อื่นและปิดช่องโหว่

DAY  
4

## Observability, Security และ Production Readiness

เนื้อหาการอบรม · Section 16-20

รวม  
6 ชั่วโมง

### 16 Lab: Observability ด้วย OpenTelemetry

- Trace, Span, Metric, Log และ Context Propagation ข้าม HTTP, gRPC และ RabbitMQ
- Auto-instrumentation สำหรับ FastAPI, HTTPX และ SQLAlchemy และการเพิ่ม Custom Span
- OpenTelemetry Collector ส่งข้อมูลไปยัง Jaeger, Prometheus และ Grafana
- ติดตามการเรียก LLM: Latency, จำนวน Token และอัตราความผิดพลาด
- Lab: ดู Distributed Trace ตั้งแต่ผู้ใช้ถามจนถึง LLM และ Order Service

### 17 Security ระหว่าง Service

- Authentication ด้วย JWT ที่ขอบระบบและการส่งต่อ Identity ไปยัง Service ภายใน
- จัดการ Secret และ API Key ของ LLM อย่างปลอดภัย
- บทบาทของ API Gateway ในระบบ Microservices พร้อมตัวอย่างด้วย Traefik
- Lab: เพิ่ม JWT และ Traefik เป็นทางเข้าเดียวของ SiamCart

### 18 Lab: ทดสอบระบบ Microservices

- Unit Test และ Integration Test ด้วย pytest และ Testcontainers สำหรับ PostgreSQL และ RabbitMQ
- Contract Testing ระหว่าง Service
- ทดสอบ AI Service ด้วยการ Mock LLM
- Lab: เขียน Integration Test ของกระบวนการสั่งซื้อ

### 19 แนวทางสู่ Production

- Configuration, Graceful Shutdown และ Health Check สำหรับ Orchestrator
- ภาพรวมการนำระบบขึ้น Kubernetes และสิ่งที่ต้องเตรียมเพิ่มจาก Docker Compose
- Checklist ความพร้อมของ Service และ AI Service ก่อนใช้งานจริง
- Lab: ตรวจสอบ SiamCart ตาม Checklist และสรุปสิ่งที่ต้องเพิ่มก่อนใช้งานจริง

### 20 Master Workshop: SiamCart คนทั้งระบบ

- รวมทุก Service ตั้งแต่ค้นหาสินค้า สั่งซื้อ ชำระเงิน แจ้งเตือน จนถึงสอบถามผ่าน AI Assistant
- จำลอง Payment Service ล่มและ LLM ตอบช้า แล้ววิเคราะห์ต้นเหตุจาก Trace ใน Jaeger
- ปรับ Timeout, Fallback และ Alert จากสิ่งที่พบ
- นำเสนอสถาปัตยกรรมและเหตุการณ์การออกแบบของทีม

## 07 เรียนจบแล้วได้อะไร Outcomes

- ออกแบบ Service อย่างมีเหตุผล DDD, Bounded Context และ Context Map
- สื่อสารข้าม Service ได้หลายแบบ REST, gRPC และ RabbitMQ
- รับมือความล้มเหลวได้ Retry, Circuit Breaker, Outbox และ Saga
- เชื่อม GenAI เข้ากับระบบ RAG ด้วย pgvector และ Tool Calling
- มองเห็นทั้งระบบ OpenTelemetry ข้าม Service และ LLM
- พร้อมต่อยอดสู่ Production Test, Security และ Checklist

## 08 สิ่งที่คุณเรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรมครบทั้ง 4 วัน และ Lab Guide รายวันพร้อมโค้ดติดตั้งสำหรับตามให้ทัน
- ซอร์สโค้ดระบบ SiamCart ทุก Service พร้อมไฟล์ Docker Compose ที่รันได้ด้วยคำสั่งเดียว
- ตัวอย่าง LLM Client, RAG Pipeline, OpenTelemetry Config และ Checklist ความพร้อมก่อนขึ้น Production
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion) และช่องทางสอบถามวิทยากรหลังจบหลักสูตร

## 09 ข้อมูลผู้สอน Instructor

**อ.สามิตร โทยม** วิทยากรผู้พัฒนาระบบ Backend ด้วย Python และสถาปัตยกรรม Microservices ที่ต้องเชื่อม Generative AI เข้ากับระบบงานจริง สอนผ่านโปรเจกต์ต่อเนื่องที่คุณเรียนสร้างเองทีละวัน อธิบายเหตุผลเบื้องหลังการออกแบบแต่ละจุดให้เข้าใจง่าย แล้วให้คุณเรียนทดลองทำให้ระบบล้มเหลวและไล่หาต้นเหตุเอง เพื่อให้กลับไปออกแบบและดูแลระบบขององค์กรได้อย่างมั่นใจ

ราคาค่าอบรม 11,900 บาท/ท่าน · สนใจ IN-HOUSE ขอใบเสนอราคา

**11,900 บาท** / ท่าน (ยังไม่รวม VAT 7%)

สมัครผ่านเว็บไซต์ ลด 10%

☎ 02-570-8449

☎ 088-807-9770

📞 @itgenius

✉ contact@itgenius.co.th



เพิ่มเพื่อน  
LINE