สถาบันไอทีจีเนียส
IT GENIUS INSTITUTE

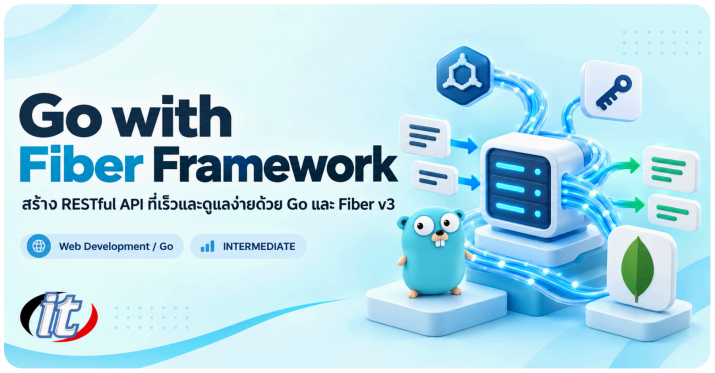
COURSE ID
WDC-108
WEB DEVELOPMENT / GO

COURSE OUTLINE · รายละเอียดหลักสูตรอบรม

Go with Fiber Framework

สร้าง RESTful API ที่เร็วและดูแลง่ายด้วย Go และ Fiber v3

- 30 ชั่วโมง
- 5 วัน
- พื้นฐาน เคยพัฒนา API ภาษาใดก็ได้
- ระดับ Intermediate



ระยะเวลา: 5 วัน
30 ชั่วโมง (5 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

Go เป็นภาษาที่ทีม Backend เลือกใช้มากขึ้นเรื่อย ๆ เพราะคอมไพล์เป็นไฟล์เดียว ทำงานเร็ว ใช้หน่วยความจำน้อย และรองรับงานพร้อมกันจำนวนมากได้ดี ส่วน Fiber เป็นเฟรมเวิร์กที่เขียนคล้าย Express ทำให้นักพัฒนาที่เคยทำ API มาก่อนเริ่มต้นได้เร็ว แต่ตัวอย่างบนอินเทอร์เน็ตจำนวนมากยังเป็น Fiber v2 และ MongoDB Go Driver v1 ซึ่งเปลี่ยนรูปแบบการเขียนไปแล้ว พอนำมาใช้กับรุ่นปัจจุบันจึงเจอโค้ดที่คอมไพล์ไม่ผ่านหรือโครงสร้างที่ดูแลต่อยาก

หลักสูตรนี้พาผู้เรียนสร้าง RESTful API ด้วย Go 1.27 และ Fiber v3 ตั้งแต่ปูพื้นภาษา Go ที่ต้องใช้จริง ทั้ง Struct, Interface, Generics, Error Handling และ Concurrency ด้วย Goroutine, Channel และ context จากนั้นสร้าง API พร้อม Binding, Validation และ Middleware ด้านความปลอดภัย เชื่อมต่อ MongoDB ด้วย Go Driver v2 ทำระบบยืนยันตัวตนด้วย JWT พร้อม Roles และ Permissions จัดโครงโปรเจกต์แบบ Hexagonal Architecture เขียน Test ทั้ง Unit และ Integration แล้วนำแอปขึ้น Docker และ Cloud ปิดท้ายด้วย Capstone สร้าง API ครบวงจรตั้งแต่ต้นจนจบ (อบรม 5 วัน วันละ 6 ชั่วโมง รวม 30 ชั่วโมง ระดับ Intermediate)

เครื่องมือ: Go 1.27 · Fiber v3 · MongoDB · JWT Auth · Docker

ราคาค่าอบรม · PUBLIC / IN-HOUSE

15,500 บาท / ท่าน

สมัครผ่านเว็บไซต์ ลด 10%

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

เพิ่มเพื่อน LINE

02-570-8449 · 088-807-9770 · @itgenius · www.itgenius.co.th

01 วัตถุประสงค์ Objectives

- เขียนภาษา Go 1.27 ได้คล่อง ทั้ง Struct, Interface, Generics และการจัดการ Error
- ใช้ Goroutine, Channel และ context เขียนงานพร้อมกันได้อย่างปลอดภัย
- สร้าง RESTful API ด้วย Fiber v3 พร้อม Binding, Validation และ Middleware
- จัดเก็บและค้นหาข้อมูลใน MongoDB ด้วย MongoDB Go Driver v2
- ทำระบบยืนยันตัวตนด้วย JWT และกำหนดสิทธิ์แบบ Roles และ Permissions
- ออกแบบโครงโปรเจกต์แบบ Hexagonal Architecture ที่ทดสอบและขยายต่อได้ง่าย
- เขียน Unit Test และ Integration Test ให้กับ API ได้
- Build เป็น Docker Image และ Deploy ขึ้นเซิร์ฟเวอร์หรือ Cloud ได้

02 กลุ่มเป้าหมาย Target Audience

- นักพัฒนา Backend ที่ต้องการเริ่มใช้ Go และ Fiber ในงานจริง
- นักพัฒนา Node.js, Python, PHP หรือ Java ที่เคยทำ API และอยากได้ระบบที่เร็วและเบากว่าเดิม
- ทีมพัฒนาซอฟต์แวร์ที่ต้องการวางโครงสร้างโค้ดแบบ Hexagonal Architecture
- Tech Lead ที่กำลังพิจารณานำ Go มาใช้กับบริการใหม่ขององค์กร
- ผู้ที่เคยเรียน Fiber v2 และต้องการอัปเดตเป็น Fiber v3

03 จุดเด่นของหลักสูตร Highlights

- **อัปเดตเป็น Fiber v3** Go 1.27 และ MongoDB Go Driver v2
- **ปูพื้น Go ที่ใช้จริง** Generics, Error Handling และ Concurrency
- **API ที่ปลอดภัย** JWT, Roles, Helmet, CORS และ Rate Limit
- **โครงสร้างที่ดูแลง่าย** Hexagonal Architecture พร้อม Test
- **พร้อมขึ้นระบบจริง** Docker, Health Check และ Deploy

04 ความรู้พื้นฐาน Prerequisites

- เคยพัฒนา RESTful API และใช้งาน JSON มาก่อนด้วยภาษาใดก็ได้
- เข้าใจ HTTP Method, Status Code, Request และ Response
- เขียนโปรแกรมได้อย่างน้อยหนึ่งภาษา เช่น JavaScript, Python, PHP หรือ Java ไม่จำเป็นต้องเคยเขียน Go
- เคยใช้ฐานข้อมูล SQL หรือ NoSQL มาบ้างจะช่วยให้เข้าใจเร็วขึ้น
- โน้ตบุ๊กที่ติดตั้ง Go, Docker Desktop และ VS Code ได้ พร้อมสิทธิ์ติดตั้งโปรแกรม

05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

รายการ	รายละเอียด
อุปกรณ์	Notebook Windows, macOS หรือ Linux เชื่อมต่ออินเทอร์เน็ต ติดตั้ง Docker Desktop ได้
ภาษาและเฟรมเวิร์ก	Go 1.27, Fiber v3, Fiber Contrib (JWT) และ golang-jwt v5
ฐานข้อมูล	MongoDB บน Docker, MongoDB Go Driver v2 และ MongoDB Compass
เครื่องมือพัฒนา	VS Code พร้อม Go Extension, golangci-lint, Air สำหรับ Hot Reload และ Postman หรือ Bruno
ทดสอบและ Deploy	go test, Testcontainers for Go, Docker, Docker Compose และ GitHub Actions

หมายเหตุ: หลักสูตรนี้ปรับปรุงใหม่ปี 2026 จากฉบับเดิมที่สอนบน Fiber v2 เนื้อหาเปลี่ยนเป็น Fiber v3 ที่ใช้ fiber.Ctx แบบ Interface และ c.Bind() แทน BodyParser ย้าย MongoDB Go Driver เป็นรุ่น v2 เพิ่ม Generics, context, Test, Logging ด้วย log/slog และ Health Check และขยายเวลาฝึก Hexagonal Architecture กับการ Deploy เครื่องมือทั้งหมดเป็นโอเพนซอร์สหรือใช้ฟรี การ Deploy ขึ้น Cloud ใช้บัญชีของผู้เรียนเอง (Free Tier หรือ Trial) ซึ่งอาจมีค่าใช้จ่ายตามเงื่อนไขของผู้ให้บริการ

06 ภาพรวมการเรียน 5 วัน Big Picture

วัน	หัวข้อหลัก
Day 1-2 Section 1-8	ปูพื้นภาษา Go 1.27 ทั้ง Struct, Interface, Generics และ Concurrency แล้วสร้าง RESTful API ด้วย Fiber v3 พร้อม Binding, Validation, Middleware และ CRUD ที่มีโครงสร้างชัดเจน
Day 3 Section 9-12	เก็บข้อมูลใน MongoDB ด้วย Go Driver v2, Query และ Aggregation, ทำระบบสมัครสมาชิกและเข้าสู่ระบบด้วย JWT และกำหนด Roles กับ Permissions
Day 4-5 Section 13-20	จัดโครงโปรเจกต์แบบ Hexagonal Architecture, เขียน Test, Logging และ Health Check, Build เป็น Docker Image, Deploy ขึ้นเซิร์ฟเวอร์และ Cloud แล้วปิดท้ายด้วย Capstone API ครบวงจร

DAY 1

พื้นฐานภาษา Go ที่ต้องใช้จริง

เนื้อหาการอบรม · Section 1-4

รวม
6 ชั่วโมง

1 Lab: เริ่มต้น Go 1.27 และเครื่องมือ

- จุดเด่นของ Go สำหรับงาน Backend และจุดที่ต่างจาก Node.js, Python และ Java
- ติดตั้ง Go 1.27 และตั้งค่า VS Code กับ Go Extension และ gopls
- Go Module, go.mod และการจัดการ Dependency ด้วย go get
- go fmt, go vet และ golangci-lint สำหรับรักษาคุณภาพโค้ด
- Lab: สร้างโปรเจกต์แรกและรันโปรแกรมแบบ Command Line

2 ไวยากรณ์ Go ที่ใช้ทุกวัน

- ตัวแปร ชนิดข้อมูล ค่าคงที่ และ Zero Value
- Slice, Map และ for range รวมถึง range over int และ Iterator
- Function แบบคืนค่าหลายตัว, Closure และ defer
- จัดการ Error แบบ Go ด้วย errors.Is, errors.As และการห่อ Error ด้วย %w
- Lab: เขียนโปรแกรมประมวลผลข้อมูลจากไฟล์ JSON

3 Struct, Method, Interface และ Generics

- Struct และ Method แทน Class พร้อม Pointer Receiver
- Embedding เพื่อใช้โค้ดซ้ำแทนการสืบทอด
- Interface แบบ Implicit และหลักการออกแบบ Interface ขนาดเล็ก
- Generics สำหรับฟังก์ชันและชนิดข้อมูลที่ใช้ซ้ำได้
- Lab: ออกแบบ Model และ Interface ของระบบจัดการสินค้า

4 Lab: Concurrency ด้วย Goroutine, Channel และ context

- Goroutine และ Channel พร้อม select สำหรับรอหลายช่องทาง
- sync.WaitGroup, Mutex และ errgroup สำหรับประสานงาน
- ยกเลิกงานและกำหนด Timeout ด้วย context
- ตรวจ Race Condition ด้วย go test -race
- Lab: เรียก API ภายนอกหลายตัวพร้อมกันแล้วรวมผลลัพธ์

DAY 2

สร้าง RESTful API ด้วย Fiber v3

เนื้อหาการอบรม · Section 5-8

รวม
6 ชั่วโมง

5 Lab: รู้จัก Fiber v3 และสร้าง API แรก

- Fiber คืออะไร ทำงานบน fasthttp อย่างไร และเทียบกับ net/http และ Gin
- สิ่งที่เปลี่ยนจาก Fiber v2 เป็น v3 เช่น fiber.Ctx แบบ Interface และ ListenConfig
- Routing, Route Group, Path Parameter และ Query String
- ตั้งค่าแอปด้วย fiber.Config และ Environment Variable
- Lab: สร้าง API แรกพร้อมทดสอบด้วย Postman หรือ Bruno

6 Request, Response และ Validation

- อ่านข้อมูลด้วย c.Bind().Body(), Query() และ URI() แทน BodyParser
- ตรวจสอบข้อมูลด้วย go-playground/validator ผ่าน StructValidator
- ส่ง JSON Response และ Status Code ให้สม่ำเสมอทั้งระบบ
- เขียน Error Handler กลางเพื่อแปลง Error เป็น Response ที่อ่านง่าย
- Lab: API รับข้อมูลสินค้าพร้อมกฎตรวจสอบครบชุด

7 Middleware ใน Fiber v3

- ลำดับการทำงานของ Middleware และการเขียน Middleware เอง
- Logger, Recover, Request ID และ Compression
- ความปลอดภัยด้วย Helmet, CORS และ Limiter
- อ่านค่าจาก Middleware ด้วยฟังก์ชัน FromContext แบบใหม่ใน v3
- Lab: วางชุด Middleware มาตรฐานให้กับ API

8 Lab: CRUD API ที่มีโครงสร้างชัดเจน

- แยก Handler, Service และ Repository ตั้งแต่โปรเจกต์แรก
- Pagination, Filtering และ Sorting ผ่าน Query String
- Upload File ด้วย Multipart Form และตรวจชนิดไฟล์
- Workshop: สร้าง CRUD API ของระบบจัดการสินค้าเก็บข้อมูลในหน่วยความจำ

DAY 3

MongoDB และระบบยืนยันตัวตน

เนื้อหาการอบรม · Section 9-12

รวม
6 ชั่วโมง

9 Lab: MongoDB กับ Go Driver v2

- รัน MongoDB ด้วย Docker และสำรวจข้อมูลด้วย MongoDB Compass
- เชื่อมต่อด้วย MongoDB Go Driver v2 และสิ่งที่เปลี่ยนจาก v1
- Client, Database, Collection และการตั้งค่า Connection Pool
- สร้าง Model ด้วย Struct และ BSON Tag พร้อม ObjectID
- สร้าง Index รวมถึง Unique Index สำหรับอีเมลผู้ใช้

10 Query, Aggregation และ Transaction

- InsertOne, Find, UpdateOne และ DeleteOne พร้อม Filter แบบต่าง ๆ
- Projection, Sort และ Pagination ที่มีประสิทธิภาพ
- Aggregation Pipeline สำหรับสรุปรายงาน
- Transaction บน Replica Set และเมื่อไรควรใช้
- Lab: ย้าย CRUD API จาก Day 2 มาเก็บข้อมูลใน MongoDB

11 Lab: Authentication ด้วย JWT

- ออกแบบระบบ Register และ Login พร้อมเก็บรหัสผ่านด้วย bcrypt
- ออก Access Token และ Refresh Token ด้วย golang-jwt v5
- ตรวจ Token ด้วย JWT Middleware จาก Fiber Contrib
- Logout, Token หมดอายุ และการเก็บ Secret อย่างปลอดภัย
- Lab: ระบบสมัครสมาชิกและเข้าสู่ระบบที่ใช้งานได้จริง

12 Lab: Roles และ Permissions

- ออกแบบ Role และ Permission ให้ตรงกับงานของระบบ
- Middleware ตรวจสิทธิ์ตาม Role และตามความเป็นเจ้าของข้อมูล
- จำกัดจำนวนครั้งการ Login ด้วย Limiter
- ช่องโหว่ API ที่พบบ่อยตาม OWASP API Security Top 10
- Lab: กำหนดสิทธิ์ Admin, Staff และ User ให้ API สินค้า

DAY 4

Hexagonal Architecture และคุณภาพโค้ด

เนื้อหาการอบรม · Section 13-16

รวม
6 ชั่วโมง

13 แนวคิด Hexagonal Architecture

- ปัญหาของโค้ดที่ผูก Business Logic ไว้กับเฟรมเวิร์กและฐานข้อมูล
- Port และ Adapter: Domain, Application และ Infrastructure
- ทิศทางของ Dependency และการวาง Package ใน Go
- เทียบ Layered, Clean และ Hexagonal Architecture แบบพอดีกับขนาดทีม

14 Lab: ปรับโปรเจกต์เป็น Hexagonal

- กำหนด Repository Port เป็น Interface และเขียน MongoDB Adapter
- แยก Fiber Handler เป็น HTTP Adapter ที่เรียก Use Case
- ประกอบ Dependency ใน main ด้วย Constructor Injection
- จัดการ Configuration และ Secret ผ่าน Environment Variable
- Lab: Refactor API สินค้าและผู้ใช้ให้อยู่ในโครงสร้างใหม่

15 Lab: Testing สำหรับ API

- Unit Test แบบ Table-driven ด้วยแพ็กเกจ testing
- Mock Port เพื่อทดสอบ Use Case โดยไม่ต้องมีฐานข้อมูล
- ทดสอบ Route ด้วย app.Test() ของ Fiber
- Integration Test กับ MongoDB จริงด้วย Testcontainers for Go
- วัด Test Coverage และตั้งเกณฑ์ขั้นต่ำของทีม

16 Logging, Health Check และเอกสาร API

- Structured Logging ด้วย log/slog พร้อม Request ID
- Health Check แบบ Liveness และ Readiness ด้วย Middleware ของ Fiber
- Graceful Shutdown เพื่อปิดเซิร์ฟเวอร์โดยไม่ตัดคำขอที่ค้างอยู่
- เขียนเอกสาร API ตามมาตรฐาน OpenAPI ให้ทีม Frontend ใช้งาน
- ภาพรวม OpenTelemetry สำหรับติดตาม Request ข้ามบริการ

DAY 5

Build, Deploy และ Capstone

เนื้อหาการอบรม · Section 17-20

รวม
6 ชั่วโมง

17 Lab: Build และรันโปรเจกต์

- Build เป็น Binary ไฟล์เดียว และ Cross-compile ด้วย GOOS และ GOARCH
- ฝังเลขเวอร์ชันด้วย -ldflags และลดขนาดไฟล์
- Hot Reload ระหว่างพัฒนาด้วย Air
- รวมคำสั่งที่ใช้บ่อยไว้ใน Makefile

18 Lab: Docker และ Docker Compose

- เขียน Dockerfile แบบ Multi-stage Build
- เลือก Base Image แบบ Distroless หรือ Scratch ให้ Image เล็กและปลอดภัย
- Docker Compose สำหรับ API และ MongoDB พร้อม Volume
- ส่งค่า Configuration และ Health Check ผ่าน Compose
- Lab: รันระบบทั้งหมดด้วยคำสั่งเดียว

19 Deploy ขึ้นเซิร์ฟเวอร์และ Cloud

- Deploy บน Linux Server ด้วย systemd และ Nginx เป็น Reverse Proxy
- Deploy Container ขึ้น Cloud เช่น Google Cloud Run, AWS หรือ DigitalOcean
- CI ด้วย GitHub Actions: Lint, Test และ Build Image อัตโนมัติ
- Checklist ความปลอดภัยและประสิทธิภาพก่อนขึ้น Production

20 Workshop: Capstone API ครบวงจร

- เลือกโจทย์ เช่น ระบบจองห้องประชุมหรือระบบคลังสินค้า
- ออกแบบ Domain, Port และ API ตามแนวทาง Hexagonal
- พัฒนาให้ครบทั้ง MongoDB, JWT, Roles และ Middleware
- เขียน Test และ Build เป็น Docker Image พร้อม Deploy
- นำเสนอผลงาน รีวิวโค้ดร่วมกัน และสรุปแนวทางต่อยอด

07 เรียนจบแล้วได้อะไร Outcomes

- **เขียน Go ได้คล่อง** Generics, Error และ Concurrency
- **สร้าง API ด้วย Fiber v3** Binding, Validation และ Middleware
- **ใช้ MongoDB ได้จริง** Go Driver v2, Index และ Aggregation
- **ทำระบบสิทธิ์ได้** JWT, Roles และ Permissions
- **วางโครงสร้างที่ดูแลง่าย** Hexagonal Architecture และ Test
- **นำ API ขึ้นระบบจริง** Docker, CI และ Cloud

08 สิ่งที่ผู้เรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรมฉบับสมบูรณ์ครบทั้ง 5 วัน และ Lab Guide ทีละขั้นตอน
- ซอร์สโค้ดตัวอย่างและโปรเจกต์ต้นแบบแบบ Hexagonal Architecture ครบทุก Lab
- ไฟล์ Dockerfile, docker-compose และ Workflow ของ GitHub Actions พร้อมใช้
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion) และช่องทางสอบถามวิทยากรหลังจบหลักสูตร

09 ข้อมูลผู้สอน Instructor

อ.สามิตร โกยม ผู้สอนและนักพัฒนาระบบที่ใช้ Go สร้างบริการ Backend และ API ในงานจริง จุดเด่นคือการสอนผ่านโปรเจกต์ชุดเดียวที่ค่อย ๆ ต่อเติมตั้งแต่วันแรกจนถึงวันสุดท้าย อธิบายเหตุผลเบื้องหลังแต่ละการออกแบบให้เห็นว่าโค้ดแบบไหนทดสอบง่ายและดูแลต่อได้ และให้ผู้เรียนลงมือทำทุกขั้นจนนำ API ขึ้นระบบได้ด้วยตนเอง

ราคาค่าอบรม 15,500 บาท/ท่าน · สนใจ IN-HOUSE ขอใบเสนอราคา

15,500 บาท / ท่าน (ยังไม่รวม VAT 7%)

สมัครผ่านเว็บไซต์ ลด 10%

02-570-8449
088-807-9770
@itgenius
contact@itgenius.co.th

เพิ่มเพื่อน LINE