



สถาบันไอจีเนียส
IT GENIUS INSTITUTE

COURSE ID

WDC-121

COURSE OUTLINE · รายละเอียดหลักสูตรอบรม

WEB DEVELOPMENT / TESTING

TDD with Node.js and TypeScript

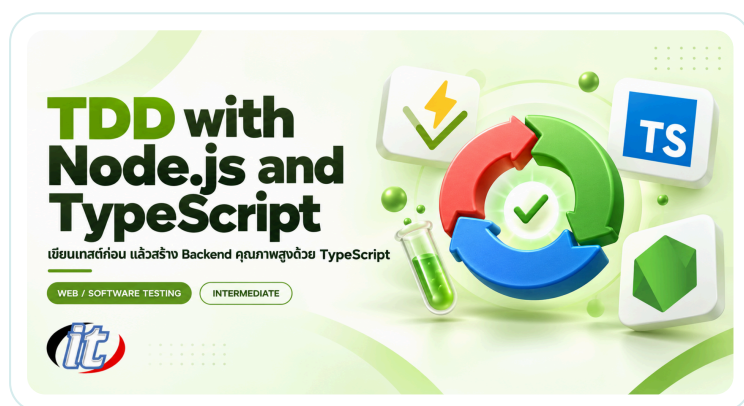
เขียน Test ก่อนโค้ดและส่งมอบ API ที่กล้า Refactor ได้อย่างมั่นใจ

• 18 ชั่วโมง

• 3 วัน

• พื้นฐาน TypeScript และเคยทำ REST API

• 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

ทุกวันนี้ทีมพัฒนาส่งมอบซอฟต์แวร์ได้เร็วขึ้นมาก ทั้งจากเครื่องมือสมัยใหม่และ AI ที่ช่วยเขียนโค้ด แต่ความเร็วมาพร้อมความเสี่ยง โค้ดอาจมีข้อผิดพลาดซ่อนอยู่ การแก้ส่วนหนึ่งทำให้อีกส่วนพังโดยไม่รู้ตัว และชุดทดสอบที่เขียนตามหลังมักไม่ครอบคลุมพฤติกรรมสำคัญ ทีมจึงต้องการวิธีทำงานที่สร้างความมั่นใจในคุณภาพตั้งแต่บรรทัดแรกของโค้ด

หลักสูตรนี้ฝึกวงจร Red-Green-Refactor อย่างเป็นระบบบน Node.js 24 LTS และ TypeScript โดยใช้ Vitest เป็นเครื่องมือหลัก เริ่มจาก Kata สั้น ๆ ต่อยอดสู่ REST API ด้วย Express 5 และ Supertest ทดสอบกับ PostgreSQL จริงด้วย Testcontainers วัดคุณภาพของ Test ด้วย Coverage และ Mutation Testing แล้วนำทั้งหมดเข้าสู่ CI Pipeline ผ่านโปรเจกต์ ParcelPoint Shipment API ชุดเดียวที่พัฒนาด้วย TDD ตลอดหลักสูตร (อบรม 3 วัน วันละ 6 ชั่วโมง รวม 18 ชั่วโมง ระดับ Intermediate)

เครื่องมือ

Vt Vitest

TS TypeScript

Nd Node.js

St Supertest

Zd Zod

ราคาค่าอบรม · PUBLIC / IN-HOUSE

8,500 บาท / ท่าน

ราคายังไม่รวมภาษีมูลค่าเพิ่ม (VAT 7%) · รวมเอกสารประกอบการอบรม Lab Guide Source Code และชุด Test ครบทุก Workshop ไฟล์ CI Pipeline ตัวอย่าง และ Certificate · รับจัด In-house ขอใบเสนอราคาได้

สมัครผ่านเว็บไซต์ ลด 10%



เพิ่มเพื่อน
LINE

☎ 02-570-8449

☎ 088-807-9770

💬 @itgenius

🌐 www.itgenius.co.th



01 วัตถุประสงค์ Objectives

- เข้าใจหลักการและประโยชน์ของ Test Driven Development และใช้วงจร Red-Green-Refactor ได้จนเป็นนิสัย
- ตั้งค่าโปรเจกต์ Node.js และ TypeScript สำหรับการทดสอบด้วย Vitest ได้ถูกต้อง
- เขียน Unit Test ที่อ่านง่าย ดูแลรักษาได้ และทดสอบพฤติกรรมแทนรายละเอียดภายใน
- เลือกใช้ Test Doubles (Stub, Mock, Spy, Fake) ได้เหมาะสมกับสถานการณ์
- พัฒนา REST API ด้วยแนวทาง TDD และทดสอบ HTTP Endpoint ด้วย Supertest
- เขียน Integration Test กับฐานข้อมูลจริงด้วย Testcontainers
- วัดคุณภาพของชุดทดสอบด้วย Coverage และ Mutation Testing
- นำชุดทดสอบเข้าสู่ CI Pipeline และกำหนด Quality Gate ให้ทีม

02 กลุ่มเป้าหมาย Target Audience

- นักพัฒนา Backend ที่ใช้ Node.js และ TypeScript ในงานประจำ
- Full Stack Developer ที่ต้องการยกระดับคุณภาพโค้ดฝั่ง Server
- Tech Lead ที่ต้องการวางมาตรฐานการทดสอบให้ทีม
- QA Engineer ที่ต้องการทำงานใกล้ชิดกับนักพัฒนาในระดับ Unit และ Integration Test
- ทีมที่ใช้ AI ช่วยเขียนโค้ดและต้องการชุดทดสอบเป็นข้อกำหนดที่ตรวจสอบได้

03 จุดเด่นของหลักสูตร Highlights

- ลงมือ TDD ตั้งแต่ชั่วโมงแรก จาก Kata สั้น ๆ สู่ REST API จริง
- เครื่องมือรุ่นปัจจุบัน Node.js 24, TypeScript 7, Vitest และ Express 5
- ทดสอบกับฐานข้อมูลจริง PostgreSQL ชั่วคราวด้วย Testcontainers
- วัดคุณภาพ Test อย่างลึก Coverage และ Mutation Testing ด้วย StrykerJS
- โปรเจกต์เดียวตลอด 3 วัน ParcelPoint API ตั้งแต่ Domain ถึง CI

04 ความรู้พื้นฐาน Prerequisites

- เขียน JavaScript หรือ TypeScript ได้ในระดับใช้งานจริง เช่น Function, Class และ async/await
- เคยพัฒนา REST API ด้วย Node.js เช่น Express มาก่อน
- เข้าใจ SQL และฐานข้อมูลเชิงสัมพันธ์เบื้องต้น ใช้ Git, npm และ Command Line ได้
- ไม่จำเป็นต้องเคยเขียน Test หรือใช้ Vitest มาก่อน ผู้สอนจะปูพื้นฐานการทดสอบให้ตั้งแต่ต้น

05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

รายการ	รายละเอียด
Runtime และภาษา	Node.js 24 LTS และ TypeScript 7 หรือ VS Code พร้อม Extension Vitest
Test Runner	Vitest (เปรียบเทียบกับ node:test ที่มากับ Node.js) และ Vitest Coverage (V8)
API และ Validation	Express 5, Supertest และ Zod (แนวทางเดียวกันใช้กับ Fastify และ Hono ได้)
Database และ Test Quality	PostgreSQL ผ่าน Testcontainers for Node.js, Kysely หรือ Drizzle ORM และ StrykerJS
อุปกรณ์และบัญชี	Notebook RAM 8 GB ขึ้นไป (แนะนำ 16 GB) ติดตั้ง Docker Desktop ได้ และบัญชี GitHub หรือ GitLab ของผู้เรียน

หมายเหตุ: หลักสูตรนี้ต่อยอดจาก Software Testing Foundation และ Basic Node.js โดยเน้นงานของนักพัฒนา Backend ที่เขียน Test ก่อนโค้ดในระดับ Unit และ Integration ส่วน End-to-End Test ของหน้าเว็บอยู่ในหลักสูตร Playwright End-to-End Test Automation เครื่องมือทั้งหมดเป็นโอเพนซอร์สและรันบนเครื่องของผู้เรียน การทดลอง CI ใช้บัญชี GitHub หรือ GitLab แบบฟรีของผู้เรียน เวอร์ชันของเครื่องมือจะปรับตามรุ่นล่าสุด ณ วันอบรม

06 ภาพรวมการเรียนรู้ 3 วัน Big Picture

วัน	หัวข้อหลัก
Day 1 Section 1-5	หลักการ TDD และวงจร Red-Green-Refactor, ตั้งค่า Node.js 24, TypeScript 7 และ Vitest, เขียน Unit Test ที่ดี, ฝึก TDD Kata และพัฒนา Domain Logic ของ ParcelPoint
Day 2 Section 6-10	ออกแบบโค้ดให้ทดสอบง่าย, Test Doubles และ Fake Timers, Outside-in TDD กับ Express 5 และ Supertest, Validation ด้วย Zod และทดสอบสิทธิ์ผู้ใช้
Day 3 Section 11-15	Integration Test กับ PostgreSQL จริงด้วย Testcontainers, Coverage, Mutation Testing ด้วย StrykerJS, Quality Gate ใน CI และ Master Workshop

DAY
1

TDD Fundamentals with Vitest

เนื้อหาการอบรม · Section 1-5

รวม
6 ชั่วโมง

1 หลักการของ Test Driven Development

- ปัญหาของการเขียน Test ตามหลัง และประโยชน์ของการเขียน Test ก่อน
- วงจร Red-Green-Refactor และกฎ 3 ข้อของ TDD
- Testing Pyramid (Unit, Integration, End-to-End) และ Testing Trophy
- TDD ในยุคที่ใช้ AI ช่วยเขียนโค้ด: ใช้ Test เป็นข้อกำหนดที่ตรวจสอบได้
- Lab: วิเคราะห์โค้ดที่ไม่มี Test แล้วเขียน Test แรกเพื่อกำหนดพฤติกรรมที่ต้องการ

2 ตั้งค่าโปรเจกต์ Node.js และ TypeScript

- Node.js 24 LTS และการรัน TypeScript ด้วย Type Stripping
- TypeScript 7 (Native Compiler) กับการตรวจ Type และเครื่องมือที่ยังต้องใช้ TypeScript 6
- ตั้งค่า tsconfig และ ES Modules
- ติดตั้งและตั้งค่า Vitest พร้อมเปรียบเทียบกับ node:test
- Lab: สร้างโปรเจกต์ตั้งต้นที่รัน Test และ Type Check ได้ด้วยคำสั่งเดียว

3 เขียน Unit Test ที่ดี

- โครงสร้าง describe, test และ expect
- รูปแบบ Arrange-Act-Assert และการตั้งชื่อ Test ให้สื่อพฤติกรรม
- Matcher ที่ใช้บ่อย และการทดสอบ Exception และ async Function
- Parameterized Test ด้วย test.each เพื่อลดโค้ดซ้ำ
- Lab: ปรับ Test ที่อ่านยากให้สั้น ชัด และไม่เปราะบาง

4 ฝึก TDD Kata

- ฝึกวงจร TDD ด้วยโจทย์สั้น เช่น String Calculator
- เทคนิค Fake It, Triangulation และ Obvious Implementation
- Refactor อย่างปลอดภัยโดยมี Test ค้ำครอง
- Lab: ทำ Kata เป็นคู่แบบ Ping-Pong Pairing และสลับบทบาทเขียน Test กับเขียนโค้ด

5 Workshop Day 1: Domain Logic ของ ParcelPoint

- คำนวณค่าขนส่งตามน้ำหนักและโซนด้วย TDD
- ใช้คู่มือส่วนลดพร้อมเงื่อนไขและกรณีพิเศษ
- State Machine ของสถานะพัสดุ: Created, Picked Up, In Transit, Delivered และ Cancelled
- Workshop: Refactor Domain Logic ให้อ่านง่ายโดย Test ยังผ่านทั้งหมด

DAY
2

TDD for REST API และ Test Doubles

เนื้อหาการอบรม · Section 6-10

รวม
6 ชั่วโมง

6 ออกแบบโค้ดให้ทดสอบได้

- แบ่ง Layer เป็น Route, Service และ Repository
- Dependency Injection แบบเรียบง่ายโดยไม่ต้องใช้ Framework
- Port and Adapter (Hexagonal Architecture) เบื้องต้น
- Lab: แยก Dependency ของ Service ออกเป็น Interface เพื่อให้ทดสอบได้

7 Test Doubles

- ความแตกต่างของ Dummy, Stub, Spy, Mock และ Fake
- การใช้ vi.fn, vi.spyOn และ vi.mock
- ควบคุมเวลาด้วย Fake Timers
- ข้อควรระวังของการ Mock มากเกินไป (Over-mocking)
- Lab: ทดสอบบริการแจ้งเตือนลูกค้าด้วย Mock และทดสอบเวลาหมดอายุด้วย Fake Timers

8 TDD กับ REST API

- Outside-in TDD เริ่มจาก Endpoint ลงสู่ Service
- ทดสอบ Express 5 ด้วย Supertest: Status Code, Header และ Body
- ตรวจสอบ Input ด้วย Zod และทดสอบ Validation Error
- ทดสอบ Error Handling Middleware
- Lab: สร้าง Endpoint สร้างพัสดุโดยเขียน Test ด้วย Supertest ก่อนเสมอ

9 ทดสอบการยืนยันตัวตนและสิทธิ์

- ทดสอบ Endpoint ที่ต้องใช้ Token ทั้งกรณีปกติและกรณีไม่มีสิทธิ์
- ทดสอบสิทธิ์ตามบทบาทผู้ใช้ (Role-based Access)
- สร้าง Test Helper สำหรับผู้ใช้แต่ละบทบาทเพื่อลดโค้ดซ้ำ
- Lab: เพิ่มกฎให้เฉพาะเจ้าหน้าที่ยกเลิกพัสดุได้ โดยเริ่มจาก Test ที่ล้มเหลว

10 Workshop Day 2: REST API ของ ParcelPoint

- API สร้าง ค้นหา อัปเดตสถานะ และยกเลิกพัสดุ โดยเขียน Test ก่อนทุก Endpoint
- ใช้ In-memory Fake Repository แทนฐานข้อมูล
- ใช้ Mock สำหรับบริการแจ้งเตือนลูกค้า
- Workshop: บททวนชุด Test ร่วมกันและแก้ Test ที่ผูกกับรายละเอียดภายในมากเกินไป

DAY
3

Integration Testing, Test Quality, CI และ Master Workshop

เนื้อหาการอบรม · Section 11-15

รวม
6 ชั่วโมง

11 Integration Testing ด้วย Testcontainers

- ข้อจำกัดของการ Mock ฐานข้อมูล และเหตุผลที่ต้องทดสอบกับฐานข้อมูลจริง
- ใช้ @testcontainers/postgresql สร้าง PostgreSQL ชั่วคราว
- Migrate Schema และเตรียมข้อมูลทดสอบ (Test Fixture)
- แยกข้อมูลระหว่าง Test และลดเวลารันด้วย Global Setup
- Lab: รับ Repository Test กับ PostgreSQL จริงใน Container

12 Code Coverage

- Line, Branch และ Function Coverage ด้วย Vitest Coverage (V8)
- กำหนด Coverage Threshold ให้เหมาะกับโปรเจกต์
- ข้อจำกัดของ Coverage ในการวัดคุณภาพ Test
- Lab: อ่าน Coverage Report และเพิ่ม Test ใน Branch ที่ยังไม่ถูกทดสอบ

13 Mutation Testing ด้วย StrykerJS

- หลักการ Mutation Testing และความหมายของ Mutation Score
- ตั้งค่า StrykerJS ร่วมกับ Vitest Runner
- วิเคราะห์ Mutant ที่รอดชีวิตและปรับปรุง Test
- Lab: เพิ่ม Mutation Score ของ Domain Logic โดยปรับ Assertion ที่อ่อนแอ

14 นำ TDD เข้าสู่ CI และการทำงานของทีม

- รับ Unit และ Integration Test ใน CI ด้วย GitHub Actions หรือ GitLab CI
- Quality Gate จาก Test, Coverage และ Type Check
- นำ TDD ไปใช้กับ Legacy Code ด้วย Characterization Test
- Code Review ที่ให้ความสำคัญกับ Test
- Lab: ตั้ง Pipeline ที่หยุดการ Merge เมื่อ Test ไม่ผ่าน

15 Master Workshop: ParcelPoint ที่พร้อมใช้งาน

- เปลี่ยน Fake Repository เป็น PostgreSQL จริงผ่าน Testcontainers และเขียน Integration Test ของการทำงานหลัก
- รับ Coverage และ StrykerJS เพื่อหา Test ที่อ่อนแอแล้วปรับปรุง
- เพิ่มฟีเจอร์ค่าบริการส่งด่วนด้วย TDD ตั้งแต่ Test แรกจนถึง Endpoint
- ตั้งค่า CI Pipeline พร้อม Quality Gate และนำเสนอผลงานพร้อมวิธีวัดโค้ดร่วมกับ

07 เรียนจบแล้วได้อะไร Outcomes

- ใช้ TDD จนเป็นนิสัย วงจร Red-Green-Refactor ในงานจริง
- ตั้งค่าเครื่องมือทดสอบได้เอง Node.js, TypeScript และ Vitest
- เขียน Test ที่มีคุณภาพ อ่านง่ายและไม่เปราะบางเมื่อ Refactor
- ทดสอบ REST API ได้ครบ Endpoint, Validation, Error และสิทธิ์
- ทดสอบกับฐานข้อมูลจริง Integration Test ด้วย Testcontainers
- วัดและยกระดับคุณภาพ Test Coverage, Mutation Score และ CI

08 สิ่งที่คุณเรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรมครบทั้ง 3 วัน และ Lab Guide รายวันพร้อมโค้ดตั้งต้นของแต่ละวัน
- Source Code และชุด Test ของ ParcelPoint API ที่พัฒนาด้วย TDD ใช้เป็นแม่แบบมาตรฐานของทีมได้
- ไฟล์ตั้งค่า Vitest, StrykerJS และ CI Pipeline ตัวอย่างสำหรับ GitHub Actions และ GitLab CI
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion) และช่องทางสอบถามวิทยากรหลังจบหลักสูตร

09 ข้อมูลผู้สอน Instructor

อ.สามิตร โทยม ผู้สอนและนักพัฒนาซอฟต์แวร์ที่ใช้ Node.js และ TypeScript พัฒนาระบบ Backend ในงานจริง จุดเด่นคือการสอน TDD แบบลงมือเขียนไปพร้อมกัน ทีละรอบของ Red-Green-Refactor อธิบายเหตุผลของการตัดสินใจแต่ละขั้น และพาผู้เรียนเจอปัญหาจริงอย่าง Test ที่เปราะบางหรือ Mock มากเกินไป เพื่อให้กลับไปวางมาตรฐานการทดสอบให้ทีมของตนเองได้

ราคาค่าอบรม 8,500 บาท/ท่าน · สนใจ IN-HOUSE ขอใบเสนอราคา

8,500 บาท / ท่าน (ยังไม่รวม VAT 7%)

สมัครผ่านเว็บไซต์ ลด 10%

☎ 02-570-8449

📞 @itgenius

☎ 088-807-9770

✉ contact@itgenius.co.th



เพิ่มเพื่อน
LINE