

Lab 05: AI Travel Planner Bot

Hands-on Workshop — Agentic Workflow

วิชา n8n Workflow Automation

ภาคการศึกษาที่ 2/2568

สร้าง AI Travel Planner Bot (Agentic Workflow)

รายละเอียด:

- เวลา: 3 ชั่วโมง
- คะแนน: 100 คะแนน
- Tests: 10 ข้อ
- Nodes: 8 + 2 Sub-nodes

สิ่งที่ต้องเตรียม:

- Discord Account
- n8n (Docker หรือ Server)
- Python 3.x + pip
- API Keys 3 ตัว (ฟรี)

Agentic Workflow คืออะไร?

AI เป็นตัวตัดสินใจ ไม่ใช่แค่ IF/ELSE

Rule-based (Lab 04):

- IF count > 0 → ส่ง
- คนเขียน logic ทั้งหมด
- ผลลัพธ์เหมือนกันทุกครั้ง

Agentic (Lab 05):

- AI รับข้อความ → แยกข้อมูลเอง
- AI สังเคราะห์ + วางแผน
- ผลลัพธ์ปรับตามบริบท

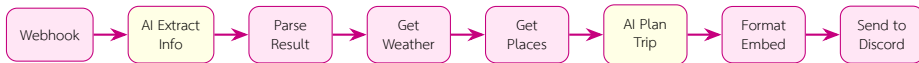
Lab นี้: AI แยก city/days + ดูอากาศ + วางแผนทริปเอง!

เกณฑ์การให้คะแนน

Test	ตรวจอะไร	คะแนน
1	มีไฟล์ workflow.json	8
2	เป็น valid JSON	8
3	มี Webhook Node	10
4	Webhook ใช้ POST method	8
5	มี HTTP Node สำหรับ OpenWeatherMap	10
6	มี AI Node (Basic LLM Chain + OpenRouter)	14
7	มี HTTP Node สำหรับ Geoapify Places	10
8	มี Code Node	10
9	มี HTTP Node สำหรับ Discord	10
10	Code Node มีการสร้าง Embed	12
รวม		100

Workflow ที่ต้องสร้าง

Chat Command — Agentic Travel Planner (8 Nodes + 2 Sub-nodes):



User พิมพ์อะไรก็ได้ → AI แยกข้อมูล + วางแผนทริปให้!

Step 1: Discord Setup

ขั้นตอน:

- ❶ สร้าง Discord Server → สร้าง Channel **#travel-plans**
- ❷ สร้าง Webhook URL (Channel Settings → Integrations)
- ❸ สร้าง Bot Application (Developer Portal → New Application)
- ❹ คัดลอก **Bot Token** + เปิด **Message Content Intent**
- ❺ เชิญ Bot เข้า Server (OAuth2 → bot scope)

เก็บ Webhook URL + Bot Token ไว้ใน Notepad!

Step 1.1: สร้าง Discord Server + Channel

ขั้นตอน:

- 1 เปิด Discord → คลิก + สร้าง Server ใหม่
- 2 เลือก **Create My Own** → **For me and my friends**
- 3 ตั้งชื่อ: n8n Lab Server
- 4 สร้าง Text Channel: **#travel-plans**

Step 1.2: สร้าง Webhook URL

ขั้นตอน:

- 1 คลิกขวาที่ #travel-plans → **Edit Channel**
- 2 ไปที่ **Integrations** → **Create Webhook**
- 3 ตั้งชื่อ: Travel Planner Bot
- 4 คลิก **Copy Webhook URL**
- 5 วางไว้ใน Notepad!

URL นี้ใส่ใน n8n Node “Send to Discord”

Step 1.3: สร้าง Bot Token

ขั้นตอน:

- ❶ เปิด <https://discord.com/developers/applications>
- ❷ คลิก **New Application** → ชื่อ: Travel Planner Bot
- ❸ คลิก **Bot** ในเมนูซ้าย
- ❹ คลิก **Reset Token** → คัดลอกทันที!
- ❺ เปิด **Message Content Intent** → ON
- ❻ Save Changes

Token แสดงครั้งเดียว! ต้องคัดลอกทันที!

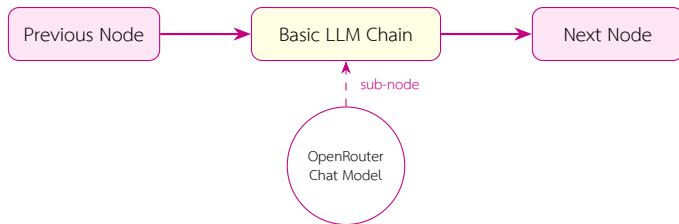
Step 1.4: เชิญ Bot เข้า Server

ขั้นตอน:

- ❶ คลิก OAuth2 → URL Generator
- ❷ Scopes: เลือก **bot**
- ❸ Permissions: View Channels, Send Messages, Embed Links, Read Message History
- ❹ คัดลอก Generated URL → เปิดใน Browser
- ❺ เลือก Server → **Authorize**

Basic LLM Chain คืออะไร?

Basic LLM Chain = n8n node สำหรับเรียก AI/LLM โดยตรง



ประกอบด้วย 2 ส่วน:

- 1 Basic LLM Chain (Main Node) — กำหนด Prompt + รับ Output
- 2 OpenRouter Chat Model (Sub-node) — เชื่อมต่อ LLM model จริง

Lab นี้ใช้ 2 LLM Chain: AI Extract Info + AI Plan Trip

ทำไมเลือก OpenRouter + Gemma?

OpenRouter คืออะไร?

- API Gateway รวม LLM หลายตัว
- สมัครทีเดียว ใช้ได้หลาย model
- เปลี่ยน model แค่เลือกจาก dropdown

ทำไมเลือก Gemma?

- google/gemma-3-27b-it:free
- ฟรี 100% — ไม่ต้องใส่บัตรเครดิต
- รองรับภาษาไทยดี
- ขนาด 27B — เพียงพอสำหรับ Lab

Model อื่นที่ใช้ได้:

Model	ราคา
gemma-3-27b-it	Free
GPT-4o	\$2.50/M
Claude 3.5	\$3.00/M
Llama 3.1	Free

Free model เพียงพอ!

LLM Node: วิธีตั้งค่าใน n8n

1. เพิ่ม Basic LLM Chain:

- กด + → ค้นหา Basic LLM Chain
- Prompt Type: **Define**
- ใส่ Prompt ในช่อง Text

2. เพิ่ม Sub-node (OpenRouter Chat Model):

- คลิก + ใต้ Basic LLM Chain → เลือก **OpenRouter Chat Model**
- Model: google/gemma-3-27b-it:free
- Credentials: กด **Create New** → ใส่ OpenRouter API Key

3. Output:

- ผลลัพธ์อยู่ใน \$json.text หรือ \$json.output
- ไม่ต้อง parse choices[0] เหมือนเรียก API ตรง!

Prompt ของ AI Extract Info

หน้าที่: แยก city + days จากข้อความอิสระ

You are a helpful assistant that extracts travel information from user messages.
Extract the city name and number of travel days.
If not specified, default to 3 days.
City name MUST be in English.

IMPORTANT: Respond with ONLY a valid JSON object:
{ "city": "CityName", "days": 3 }

Examples:

- "อยากไปเชียงใหม่ 3 วัน" -> { "city": "Chiang Mai", "days": 3 }
- "Plan a trip to Tokyo for 5 days" -> { "city": "Tokyo", "days": 5 }
- "แนะนำที่เที่ยวกรุงเทพ" -> { "city": "Bangkok", "days": 3 }

User message: {{ \$json.body.content }}

Respond with ONLY the JSON object:

Prompt ของ AI Plan Trip

หน้าที่: สร้างแผนทริปรายวัน จากข้อมูลจริง

You are an expert travel planner.
Create a day-by-day travel itinerary in Thai language
based on weather data and tourist attractions.
For each day, suggest morning, afternoon, and evening.
Consider weather for outdoor vs indoor activities.

Weather: Temperature {{ \$json.main.temp }}C,
Humidity {{ \$json.main.humidity }}%,
Condition: {{ \$json.weather[0].description }}

Tourist attractions nearby:
{{ \$json.features ?
 \$json.features.map(f=>f.properties.name).join(', '
 : 'No data' }}

Please create the itinerary.

AI ใช้ weather + places จริง → แผนทริปปรับตามสภาพอากาศ!

Step 2: สมัคร API Keys (ทั้งหมดฟรี!)

API	เว็บ	ใช้ทำอะไร	Free Quota
OpenWeatherMap	openweathermap.org	สภาพอากาศ	1,000 calls/day
Geoapify	myprojects.geoapify.com	สถานที่ท่องเที่ยว	3,000 calls/day
OpenRouter	openrouter.ai	Gemini AI (LLM)	Free models

หมายเหตุ:

- OpenWeatherMap: รอ **10 นาที** หลังสมัคร (key ต้อง activate)
- OpenRouter: ใช้ model **google/gemma-3-27b-it:free**
- ทดสอบแต่ละ API ด้วย curl ก่อนใส่ n8n

Step 3.1: Webhook + AI Extract Info

Node 1: Webhook

- Method: POST, Path: travel-planner

Node 2: AI Extract Info (Basic LLM Chain) — หัวใจ Agentic!

- AI แยก **city** และ **days** จากข้อความอิสระของ user
- ตัวอย่าง: “อยากไปเที่ยวเชียงใหม่ 3 วัน”
- AI ตอบ: {"city": "Chiang Mai", "days": 3}

Node 3: Parse Extract Result (Code)

- แปลง AI response เป็น JSON ที่ node ถัดไปใช้ได้
- มี fallback: ถ้า AI ตอบผิด format จะใช้ Bangkok/3 วัน

Step 3.2: Weather + Places + AI Plan Trip

Node 4: Get Weather (OpenWeatherMap)

- URL: ...?q={{ \$json.city }}&appid=KEY&units=metric
- สังเกต: ใช้ \$json.city จาก Parse Extract Result

Node 5: Get Places (Geoapify)

- ใช้ coord จาก Weather API เป็น filter
- categories=tourism.sights, limit=5

Node 6: AI Plan Trip (Basic LLM Chain)

- ส่ง weather + places data ให้ AI
- Prompt: “สร้างแผนทริปรายวัน ภาษาไทย”
- **Agentic:** AI ตัดสินใจเลือกกิจกรรมตามสภาพอากาศเอง!

Step 3.3: Format + Send

Node 7: Code (Format Embed)

- ดึงแผนทริปจาก AI response
- ตั้งสีตามสภาพอากาศ (Clear=Gold, Rain=Blue, ...)
- Truncate ถ้ายาวเกิน 4,096 ตัวอักษร (Discord limit)

Node 8: HTTP Request (Send to Discord)

- Method: **POST**
- URL: Discord Webhook URL
- Body: `{{= $json }}`

รวม 8 Nodes + 2 Sub-nodes (OpenRouter Chat Model)

Step 4: ตั้งค่าและรัน bot.py

1. ติดตั้ง Library:

```
pip install discord.py aiohttp
```

2. แก้ไข bot.py — ใส่ค่า 2 ตัว:

```
BOT_TOKEN = "MTQ3MDk4...xxxxx"  
N8N_WEBHOOK_URL = "https://your-n8n/webhook/travel-planner"
```

3. Activate Workflow ใน n8n ก่อน! 4. รัน + ทดสอบ:

```
python bot.py
```

พิมพ์อะไรก็ได้ใน Discord: อยากไปเที่ยวเชียงใหม่ 3 วัน

Step 5: Export & Push

1. Export workflow.json:

- n8n → Menu → Download → บันทึกเป็น workflow.json

2. Push ขึ้น GitHub:

```
git add workflow.json  
git commit -m "Add AI Travel Planner workflow"  
git push origin main
```

รอ Auto-grading ตรวจ 10 Tests!

Checklist ก่อนส่ง

- ☐ สร้าง Discord Server + Channel #travel-plans
- ☐ สร้าง Webhook URL + Bot Token
- ☐ สมัคร OpenWeatherMap API Key + ทดสอบ
- ☐ สมัคร Geoapify API Key + ทดสอบ
- ☐ สมัคร OpenRouter API Key + ทดสอบ
- ☐ สร้าง Workflow: พิมพ์ข้อความอิสระแล้วตอบกลับสำเร็จ
- ☐ Export workflow.json
- ☐ pytest ผ่าน 10/10 (100 คะแนน)
- ☐ Push ขึ้น GitHub

Troubleshooting

OpenRouter 401 Unauthorized

- API key ผิด → สร้าง key ใหม่ใน openrouter.ai

OpenWeather 401

- Key ยังไม่ active → รอ 10 นาทีหลังสมัคร

AI แยก city ผิด / ตอบไม่ตรงประเด็น

- ปรับ prompt ของ AI Extract Info / AI Plan Trip ให้ชัดเจนขึ้น

Webhook 404 หลัง Activate

- เปลี่ยนจาก /webhook-test/ เป็น /webhook/

มีคำถามไหม?

ดู Lab Instructions: README.md

ดูคู่มือตั้งค่า: SETUP_GUIDE.md

Good Luck!

ขอให้ทำ Lab สำเร็จ!