สารบัญ

01 ข้อสังเกตจากบอส

02 สิ่งที่ต้องเตรียม

03 ติดตั้งแบบละเอียด

04 ใส่ API Key (AI)

05 เชื่อมต่อ LINE OA

06 Checklist ใช้งานจริง

07 คำสั่งที่ใช้บ่อย

08 Auto-start (Task Scheduler)

09 Backup / Restore

10 Troubleshooting

11 ตำแหน่งไฟล์สำคัญ

12 เอกสารที่เกี่ยวข้อง

เวอร์ชัน

v1 · ฉบับเต็ม 12 หัวข้อ
v2 · 4 ขั้นอัจฉริยะ — ดูที่

```
12  INSTALL_GUIDE.html
```

---

v1 · ฉบับละเอียด 12 ขั้น

คู่มือติดตั้ง Office Chat แบบละเอียดทุกขั้น

คู่มือฉบับเต็มสำหรับติดตั้งระบบ LAN-first แชท + แชร์ไฟล์ + AI + LINE OA ตั้งแต่ศูนย์จนพร้อมใช้งานจริง พร้อม mockup หน้าจอทุกขั้น (วาดจาก UI จริงในโค้ด)

⏱ ใช้เวลาประมาณ 30–45 นาที | Windows / macOS / Linux | Node.js 24+ | รองรับ HTTPS + LAN

0. ข้อสังเกตจากบอส

เกี่ยวกับคำว่า "KPI Key"

บอสพิมพ์ว่า "KPI Key" — ผมตีความว่าหมายถึง

API Key

ของผู้ให้บริการ AI (OpenAI / Google Gemini) เพราะในระบบไม่มีค่าใดชื่อ "KPI Key" หากบอสหมายถึงคีย์อื่น แจ้งกลับได้ครับ จะเพิ่มขั้นตอนนั้นให้

1 สิ่งที่ต้องเตรียมก่อนติดตั้ง

รายการ	ค่าที่ต้องใช้	หมายเหตุ
Node.js	**24 ขึ้นไป**	ตรวจด้วย `node -v` (โปรเจกต์ตั้ง `engines.node ⩾ 24`)
npm	ติดมากับ Node.js	ใช้ `npm -v` ตรวจ
ระบบปฏิบัติการ	Windows / macOS / Linux	คู่มือนี้เขียนบน PowerShell (Windows)
Port ว่าง	`3000` (ดีฟอลต์)	ถ้าใช้อื่น เปลี่ยนใน `.env`
HTTPS domain	ต้องมีเมื่อใช้ LINE OA, OAuth, Web Push	สำหรับ LAN ใช้ `http://IP-เครื่อง` ได้ แต่ LINE ต้อง HTTPS

ตรวจ Node.js ก่อน

```
PS> node -v
v24.6.0 # ถ้าได้เวอร์ชัน 24.x.x ขึ้นไป พร้อมไปต่อ
```

ถ้ายังไม่มี ดาวน์โหลด LTS จาก nodejs.org แล้วติดตั้งแบบปกติ ไม่ต้องเลือกอะไรเพิ่ม

2 ติดตั้งแบบละเอียด (ทีละขั้น)

2.1 เปิด PowerShell ที่โฟลเดอร์โปรเจกต์

```
PS> Set-Location "D:\For sale\Chat OF"
```

2.2 สร้างไฟล์ `.env` จากแม่แบบ

```
PS> Copy-Item .env.example .env
```

> **ไฟล์ `.env`**
> คือที่เก็บค่าตั้งค่าทั้งหมด (รวม API Key และคีย์ LINE) ระบบจะเขียนกลับเข้าไฟล์นี้ทุกครั้งที่กดบันทึกใน Setup Center

2.3 ติดตั้ง dependencies

```
PS> npm install
```

รอจนเสร็จ (ครั้งแรกอาจใช้เวลา 1–3 นาที ขึ้นกับเน็ต)

2.4 ตรวจสภาพเครื่อง (แนะนำ)

```
PS> npm run doctor
```

สคริปต์นี้จะบอก IP ที่ใช้เปิดจากเครื่องอื่นในวง LAN, ตรวจ Node.js, ตรวจ port ว่าง

2.5 เปิดใช้งานครั้งแรก

```
PS> npm start
# หรือใช้สคริปต์ที่เตรียมไว้
PS> .\scripts\start-office-chat.ps1
```

Server จะฟังที่ `http://0.0.0.0:3000`

เปิดเบราว์เซอร์:

- เครื่องเซิร์ฟเวอร์เอง: `http://localhost:3000`
- เครื่องอื่นใน LAN: `http://<IP-เครื่องเซิร์ฟเวอร์>:3000` (ดู IP จาก `npm run doctor`)

2.6 สร้างบัญชี Admin คนแรก

เมื่อเปิดครั้งแรก ระบบจะพาไปหน้า **First-run setup** ให้ตั้ง:

- ชื่อ-นามสกุล
- อีเมล/เบอร์โทร (สำหรับ login)
- รหัสผ่าน (แนะนำ 12 ตัวอักษรขึ้นไป ผสมตัวเลข/ตัวพิมพ์ใหญ่/สัญลักษณ์)

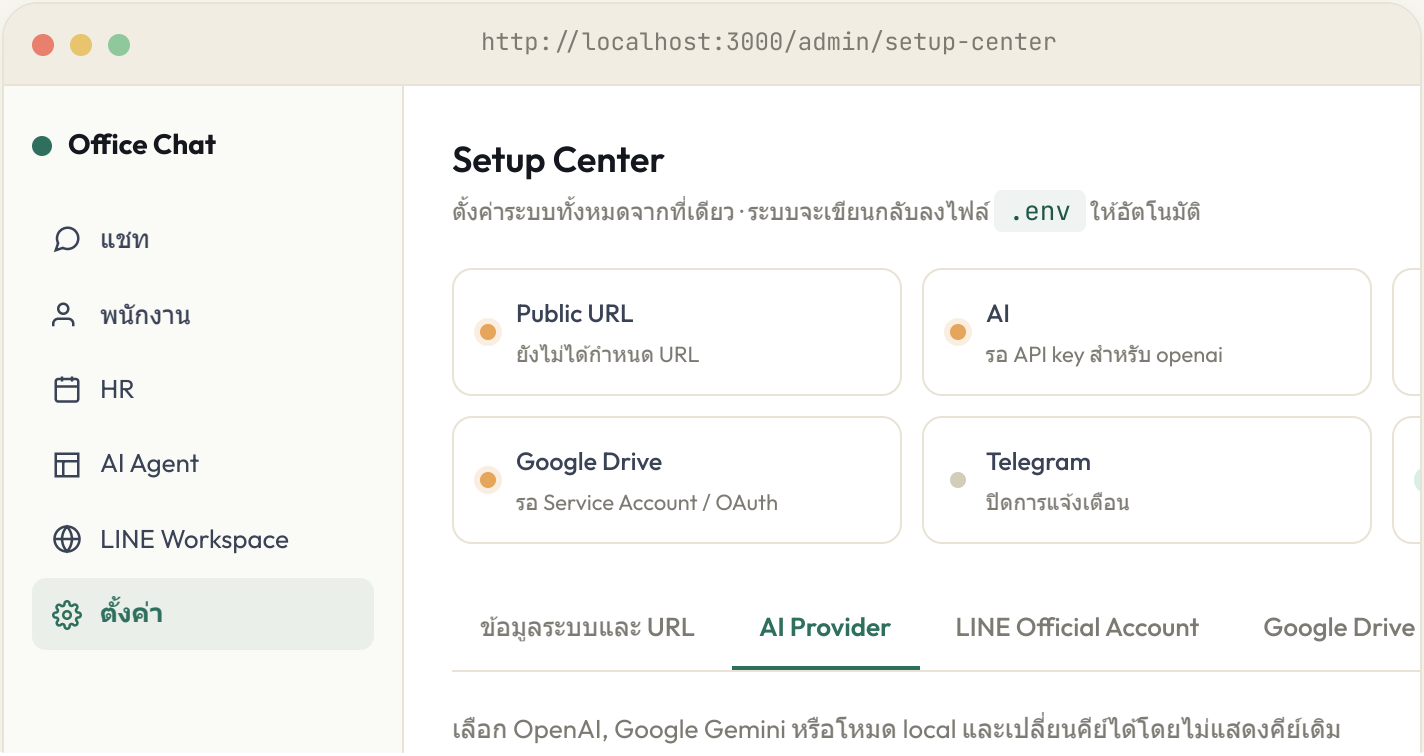
> หลังสร้างเสร็จ ระบบจะสร้างห้องเริ่มต้น (channel กลาง, channel ประกาศ) ให้อัตโนมัติ

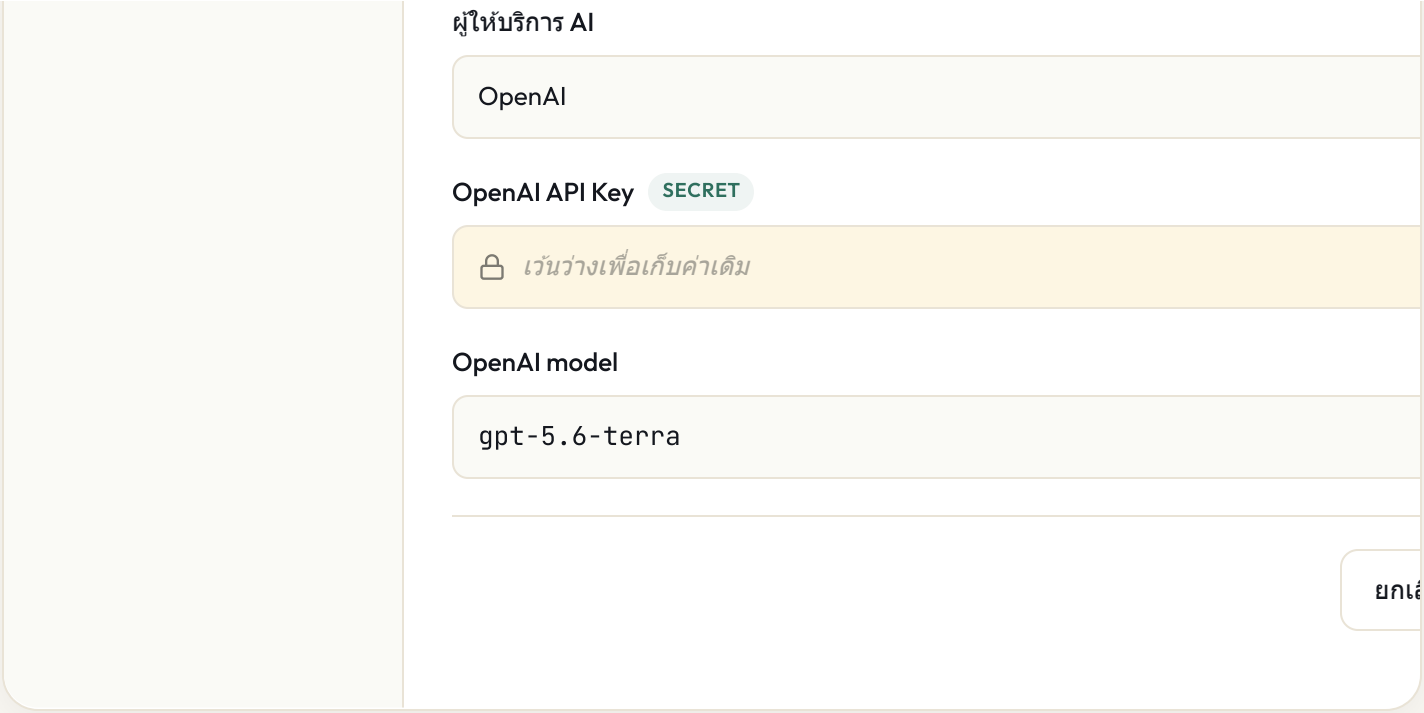
2.7 เปิด Setup Center

จากเมนูบนขวา → **ตั้งค่า** → **Setup Center** (หรือเข้า URL `/admin/setup-center` ตรง ๆ ก็ได้)

หน้านี้คือศูนย์รวมตั้งค่าทุกอย่าง — ไม่ต้องแก้ไฟล์ `.env` ด้วยมืออีก (แม้ระบบจะเขียนลง `.env` ให้เอง)

นี่คือ mockup หน้า Setup Center (ส่วนบน — แสดงสถานะการเชื่อมต่อทั้งหมด):





โครงสร้าง 6 กลุ่มใน Setup Center

1)
ข้อมูลระบบและ URL

· 2)
AI Provider

· 3)
LINE Official Account

· 4)
Google Drive

· 5)
Telegram

· 6)
Web Push & integrations

3 วิธีใส่ API Key (AI Provider)

ส่วนนี้สำหรับตั้งค่า AI ให้น้องแอสลาสตอบลูกค้าและช่วยงานแอดมิน ระบบรองรับ 3 ตัวเลือก:

ตัวเลือก	ตัวแปร	คีย์ที่ต้องใส่
OpenAI	`AI_PROVIDER=openai`	`OPENAI_API_KEY`
Google Gemini	`AI_PROVIDER=google`	`GOOGLE_API_KEY`
Local (ปิด AI ภายนอก)	`AI_PROVIDER=local`	ไม่ต้องใช้คีย์

แนะนำสำหรับงานจริง: `openai` แล้วใช้ `GOOGLE_API_KEY` เป็น fallback

3.2 ขอ OpenAI API Key

1. ไป platform.openai.com/api-keys
2. ล็อกอิน (หรือสมัครใหม่)
3. กด **Create new secret key** → ตั้งชื่อ เช่น `office-chat-server` → **Create**
4. **คัดลอก key ทันที** (ขึ้นต้นด้วย `sk-...`) — จะปิดหน้านี้แล้วดูอีกไม่ได้
5. ตั้ง billing ให้เรียบร้อยที่ platform.openai.com/account/billing

3.3 ขอ Google Gemini API Key (ทางเลือกเสริม)

1. ไป aistudio.google.com/apikey
2. ล็อกอิน Google account
3. กด **Create API key** → เลือก (หรือสร้าง) Google Cloud project
4. คัดลอก key (ขึ้นต้นด้วย `AIza...`)

3.4 ใส่คีย์ใน Setup Center

1. เปิดหน้า **ตั้งค่า** → **Setup Center**
2. เลือกกลุ่ม **AI Provider**

ฟิลด์	ค่าที่ใส่
ผู้ให้บริการ AI	เลือก `OpenAI` หรือ `Google Gemini`
OpenAI API Key	วาง key `sk-...`
OpenAI model	`gpt-5.6-terra` (หรือโมเดลที่ต้องการ)

ฟิลด์	ค่าที่ใส่
OpenAI output tokens	`900` (ค่าตั้งต้นกำลังดี)
Google API Key	วาง key `AIza...` (ใส่คู่กับ OpenAI เพื่อใช้เป็น fallback)
Google model	`gemini-3.5-flash`

4. **สำคัญ:** ช่อง secret ที่เว้นว่าง ระบบจะ **เก็บค่าเดิม** ไว้ ถ้าจะเปลี่ยน key ให้วางใหม่ ถ้าจะลบ key ให้ติ๊ก **"ล้างค่าที่บันทึก"**
5. กด **บันทึกการตั้งค่าทั้งหมด**

> AI/LINE key และโหมดตอบ **เปลี่ยนได้ทันที** ไม่ต้องรีสตาร์ท · ฟิลด์ที่มีป้าย **รีสตาร์ต** เท่านั้นที่ต้องหยุด server แล้ว `npm start` ใหม่

3.5 ตรวจว่าคีย์ใช้ได้

- ดูสถานะบนหน้า Setup Center: การ์ด **AI** ต้องขึ้น "ready"
- ลองคุยกับ **น้องแอสลาส** ในห้องส่วนตัว (ระบบสร้างให้อัตโนมัติทุก user)
- ถ้า error ให้ดู log ที่โฟลเดอร์ `logs/`

4 วิธีเชื่อมต่อ LINE Official Account (LINE OA)

4.1 ภาพรวม

```
ลูกค้า LINE ──► LINE Webhook ──► https://your-domain/api/line/webhook
                                          │
                                          ▼                    Office Chat Server
                                          │
                                          ▼                    AI ตอบในโหมด draft/auto/direct
```

ต้องมี:

1. **LINE Official Account** (สร้างที่ manager.line.biz)

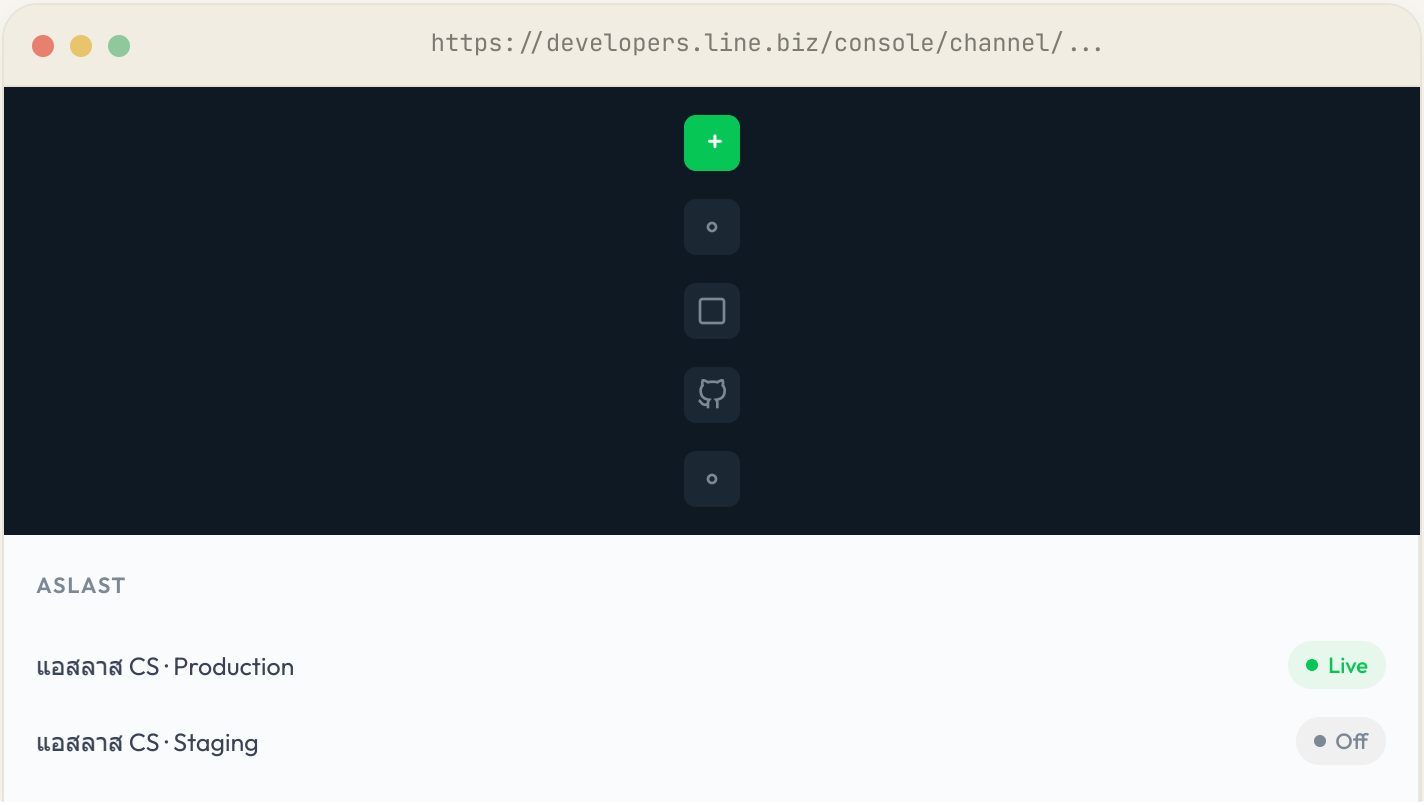
2. **Messaging API channel** (สร้างใน LINE Developers console)
3. **HTTPS domain** ที่ชี้มาที่เซิร์ฟเวอร์ (เพราะ LINE ส่ง webhook ผ่าน HTTPS เท่านั้น)

4.2 สร้าง LINE Official Account + Messaging API

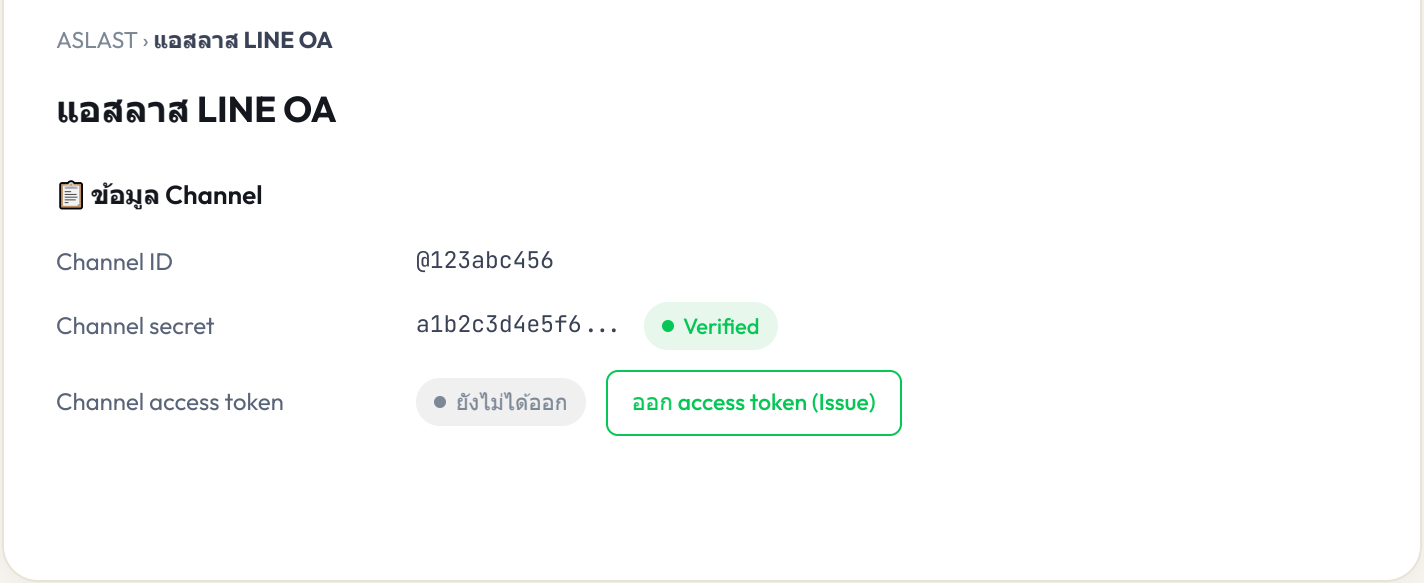
ขั้น A: สร้าง Provider และ Channel

1. ไป developers.line.biz/console
2. ล็อกอินด้วย LINE account เดียวกับที่ผูกกับ LINE OA
3. เลือก Provider (ถ้ายังไม่มี กด **Create a new provider** → ตั้งชื่อบริษัท)
4. ในหน้า Provider → กด **Create a new channel**
5. เลือก **Messaging API**
6. กรอกข้อมูล:
    - Channel name: ชื่อที่ลูกค้าจะเห็น
    - Channel description: คำอธิบาย
    - Category / Subcategory: เลือกให้ตรงธุรกิจ
    - Email: อีเมลผู้ดูแล
7. ยอมรับ terms → **Create**

นี่คือหน้า LINE Developers Console ที่บอสจะเห็นหลังสร้างเสร็จ:



แอสลาส LINE OA



ขั้น B: ผูก Channel กับ LINE OA ที่มีอยู่ (ถ้ามี)

- ถ้ามี LINE OA อยู่แล้ว: LINE Developers → Channel → **LINE Official Account** → เลือก OA ที่ผูก
- ถ้ายังไม่มี OA: ไป manager.line.biz → สร้าง LINE OA ใหม่ → กลับมา LINE Developers console → ผูก

ขั้น C: ตั้งค่า Channel ที่หน้า Channel ของคุณ

แท็บ	ค่าที่ตั้ง
Basic settings	ตรวจ Channel secret (เก็บไว้ใช้ในขั้นต่อไป)
Messaging API	กด **Issue** ในหัวข้อ Channel access token → คัดลอก token (ยาว, ขึ้นต้นด้วย `eyJ...`)
Messaging API → Webhook settings	Webhook URL: `https://<โดเมน>/api/line/webhook` · เปิด **Use webhook**
Messaging API → Response settings	ตั้ง **Response mode** = **Bot** (สำคัญ! ถ้าเป็น Chat จะตอบเองทับ AI)
Messaging API → Auto-response	ปิด **Auto-response messages** และ **Greeting messages**
LINE Official Account features	ปิด **Auto-response** และ **Greeting** ของ OA Manager ด้วย

4.3 เตรียม HTTPS + Public URL

LINE webhook ต้องเป็น HTTPS — มี 3 ทางเลือก:

ทาง A: ใช้ HTTPS domain ที่ชี้มาที่เครื่องเซิร์ฟเวอร์

- ตั้ง DNS: `chat.yourcompany.com` → IP เครื่องเซิร์ฟเวอร์
- ใช้ reverse proxy (Nginx / Caddy) ทำ TLS ด้วย Let's Encrypt

ทาง B: ใช้ Cloudflare Tunnel / ngrok (สำหรับ dev/ทดสอบ)

```
PS> cloudflared tunnel --url http://localhost:3000
```

จะได้ URL `https://xxx.trycloudflare.com` ใช้เป็น Public URL

ทาง C: ใช้ NSSM + Windows Server + certbot — สำหรับ production จริงจังในองค์กร

ตั้ง `OFFICE_CHAT_PUBLIC_URL` ใน `.env` (หรือ Setup Center) เป็น URL HTTPS เต็ม ๆ เช่น `https://chat.yourcompany.com`

4.4 ใส่ค่าใน Setup Center

กลับมาที่ **ตั้งค่า** → **Setup Center** → กลุ่ม **LINE Official Account**:

ฟิลด์	ค่าที่ใส่
โหมดตอบ LINE	เลือกตามต้องการ (ดูหัวข้อ 4.5)
Channel secret	ค่าจาก LINE Developers → Basic settings
Channel access token	token จากขั้น 4.2 (ขึ้นต้น `eyJ...`)
ขอบเขตสินค้า/บริการ	เช่น "เสื้อฮาวาย, กางเกงช้าง, ผ้าห่มพิมพ์ลาย"
พัก AI หลังแอดมินตอบ (ชั่วโมง)	24 (ค่าตั้งต้น)
เวลาเริ่มตอบ / เวลาหยุดตอบ	08:00 / 17:00 (ตามเวลาทำการ)

กด **บันทึกการตั้งค่าทั้งหมด** → การ์ดสถานะ **LINE OA** ต้องขึ้น **"ready"**

4.5 เลือกโหมดตอบ LINE

โหมด	พฤติกรรม	เหมาะกับ
`draft`	AI ร่างคำตอบไว้ใน Office Chat ให้แอดมินกดส่ง	**เริ่มต้นแนะนำ**
`auto`	AI ตอบ LINE ให้อัตโนมัติ + เก็บสำเนาใน Office Chat	ใช้หลังตรวจ Knowledge Base แล้ว
`direct`	AI ตอบ LINE ตรง ไม่สร้างห้องใน Office Chat	ถ้ามีห้อง LINE อยู่แล้ว ไม่อยากให้ระบบสร้างทับ
`off`	ปิดการตอบ LINE	ปิดชั่วคราว

> ⚠ **คำแนะนำ**
> เริ่มด้วย `draft` เสมอ → ให้แอดมินตรวจ draft → ค่อยเปิด `auto` เมื่อมั่นใจ

4.6 ผูก Webhook กลับใน LINE Developers

กลับไปที่ developers.line.biz/console:

1. เลือก Channel ของคุณ → แท็บ **Messaging API**
2. หัวข้อ **Webhook URL** → กรอก `https://<โดเมน>/api/line/webhook`
3. กด **Update** → กด **Verify** (ถ้าสำเร็จจะขึ้น Success)
4. เปิด **Use webhook** ให้เป็นเขียว

4.7 ทดสอบจริง

1. จากมือถือ เพิ่มเพื่อน LINE OA ของคุณ (สแกน QR จาก LINE OA Manager)
2. ส่งข้อความ "ทดสอบ" หรือ "สวัสดี"
3. ตรวจ 3 จุด:
   - ใน LINE: ถ้าโหมด `auto` / `direct` จะได้รับคำตอบจาก AI ทันที
   - ใน Office Chat: ต้องมีห้องใหม่ชื่อลูกค้า (ถ้าโหมดไม่ใช่ `direct`) พร้อม draft หรือสำเนา
   - ในหน้า Admin → LINE Workspace: จะเห็นข้อความเข้า

4.8 คำสั่งแอดมินในห้อง LINE

แอดมินสามารถพิมพ์ในห้องลูกค้า LINE ใน Office Chat:

```
# ส่งข้อความจากแอดมินกลับไปหาลูกค้า + พัก AI อัตโนมัติ
/line ข้อความตอบกลับ

# จัดการ AI เฉพาะลูกค้าคนนี้
/line-ai off # ปิด AI ให้คนตอบเอง
/line-ai on # เปิด AI กลับมา
/line-ai pause # พัก AI 24 ชม.
/line-ai status # ดูสถานะ AI ของลูกค้าคนนี้
```

5 เริ่มเปิดใช้งานจริง (Checklist)

- [ ] Node.js 24+ ติดตั้งแล้ว
- [ ] `npm install` สำเร็จ
- [ ] `.env` ถูกสร้างจาก `.env.example`
- [ ] `npm start` ทำงาน ไม่มี error ใน console
- [ ] `npm run doctor` ผ่าน
- [ ] เข้าหน้าเว็บได้ทั้งจากเครื่องเซิร์ฟเวอร์และเครื่องอื่นใน LAN
- [ ] สร้างบัญชี Admin คนแรกแล้ว
- [ ] ใส่ API Key (OpenAI/Gemini) ใน Setup Center → การ์ด AI ขึ้น ready
- [ ] ทดสอบคุยกับน้องแอสลาสได้
- [ ] ตั้ง HTTPS + Public URL (ถ้าจะใช้ LINE)
- [ ] ใส่ LINE Channel secret + access token → การ์ด LINE OA ขึ้น ready
- [ ] ตั้ง webhook URL ใน LINE Developers → Verify ผ่าน
- [ ] ทดสอบส่งข้อความจาก LINE OA เข้ามา

- [ ] เลือกโหมดตอบ (แนะนำ `draft` ก่อน)
- [ ] ตั้ง firewall เฉพาะ port ที่จำเป็น
- [ ] ตั้ง Task Scheduler / NSSM ให้ server รันอัตโนมัติ (ดูหัวข้อ 7)
- [ ] ตั้ง backup `data/` และ `uploads/` เป็นประจำ

6 คำสั่งที่ใช้บ่อย

```
PS> npm start # รัน server (foreground)
PS> npm run dev # รันแบบ watch (รีโหลดเมื่อโค้ดเปลี่ยน)
PS> npm run doctor # ตรวจสภาพเครื่อง + หา IP สำหรับ LAN
PS> npm test # smoke test
PS> npm run test:unit # unit test
PS> .\scripts\start-office-chat.ps1 # รันแบบมี auto-install
PS> .\scripts\backup-office-chat.ps1 # สำรองข้อมูล
PS> .\scripts\restore-office-chat.ps1 # กู้คืนข้อมูล
```

7 ตั้งให้รันอัตโนมัติบน Windows (Task Scheduler)

1. เปิด **Task Scheduler** → **Create Task**
2. แท็บ **General**:
   - Name: `Office Chat Server`
   - Run whether user is logged on or not: ✓
   - Run with highest privileges: ✓
3. แท็บ **Triggers** → **New**: At system startup
4. แท็บ **Actions** → **New**:
   - Program/script: `powershell.exe`
   - Arguments: `-NoProfile -ExecutionPolicy Bypass -File "D:\For sale\Chat OF\scripts\start-office-chat.ps1"`

5. แท็บ **Conditions**: Start only if network is available: ✓

หรือใช้ NSSM (แนะนำสำหรับ production) — ให้บริการ restart อัตโนมัติเมื่อ crash

8 Backup / Restore

สำรองข้อมูล (ทำทุกวัน):

```
PS> .\scripts\backup-office-chat.ps1
# สำรอง 2 โฟลเดอร์:
# data/ (ฐานข้อมูล SQLite)
# uploads/ (ไฟล์ที่อัปโหลดทั้งหมด)
```

แนะนำ:

- backup รายวันไป NAS / external disk
- เก็บย้อนหลังอย่างน้อย 30 วัน
- ทดสอบ restore เดือนละครั้ง

กู้คืน:

```
PS> .\scripts\restore-office-chat.ps1 -BackupPath "<path-to-backup>"
```

9 Troubleshooting (เจอปัญหาทำยังไง)

อาการ	สาเหตุ / วิธีแก้
ติด `EACCES` ตอน `npm install`	ปิด antivirus ชั่วคราว หรือ run PowerShell แบบ Run as Administrator
`node -v` ได้ v18 หรือต่ำกว่า	ถอน Node.js เก่า แล้วติดตั้ง v24+ ใหม่จาก nodejs.org
เปิด `http://localhost:3000` ไม่ขึ้น	ตรวจ console ว่า server เริ่มจริงไหม · ตรวจว่า port 3000 ว่าง

อาการ	สาเหตุ / วิธีแก้
เครื่องอื่นเข้าไม่ได้	ปิด Windows Firewall ชั่วคราวเพื่อทดสอบ · ถ้าผ่าน → เปิด firewall แล้ว allow port 3000
การ์ด LINE OA ไม่ขึ้น ready	ตรวจว่า Channel secret + access token ถูกต้อง · ตรวจว่าไม่มี space หัว-ท้าย
LINE Verify webhook ไม่ผ่าน	ตรวจว่า `OFFICE_CHAT_PUBLIC_URL` เป็น HTTPS · ตรวจว่า reverse proxy forward ไปที่ port 3000
AI ตอบไม่ได้	ดูการ์ด AI ใน Setup Center · ตรวจว่า API key ถูกต้อง · ตรวจ billing ของ OpenAI/Google
หลังแก้ `.env` แล้วไม่มีผล	ฟิลด์ที่เป็น `restartRequired: true` ต้อง restart server (`Ctrl+C` แล้ว `npm start` ใหม่)
ลืมรหัสผ่าน admin	หยุด server · ลบ user admin เก่าออกจาก `data/office-chat.db` (ผ่าน sqlite3) แล้วให้คนอื่นสมัครใหม่
port 3000 ถูกใช้แล้ว	แก้ `PORT=3001` ใน `.env` แล้ว restart

10 ตำแหน่งไฟล์สำคัญ (อ้างอิงด่วน)

ไฟล์ / โฟลเดอร์	ใช้ทำอะไร
`.env`	ค่าตั้งค่าทั้งหมด — **อย่าแชร์ให้คนอื่น**
`.env.example`	แม่แบบค่าตั้งต้น (มี comment อธิบาย)
`data/`	ฐานข้อมูล SQLite — backup ทุกวัน
`uploads/`	ไฟล์ที่ผู้ใช้อัปโหลด — backup ทุกวัน
`knowledge/`	ข้อมูลโรงงาน/สินค้าให้ AI อ่าน
`logs/`	log ไฟล์ — ดูตอน debug
`secrets/`	เก็บ credential ไฟล์ — **ถูกตัดออกจาก source package อัตโนมัติ**
`scripts/`	สคริปต์ช่วยเหลือ (start/backup/restore)

ไฟล์ / โฟลเดอร์	ใช้ทำอะไร
`docs/`	เอกสารคู่มือ

11 เอกสารที่เกี่ยวข้อง

- `README.md` — ภาพรวมฟีเจอร์และการติดตั้งแบบย่อ
- `docs/BUYER_QUICKSTART_TH.md` — quick start 10 นาที (สำหรับผู้ซื้อซอร์ส)
- `docs/FEATURE_CATALOG_TH.md` — แค็ตตาล็อกฟีเจอร์ทั้งหมด
- `docs/RUNBOOK.md` — คู่มือ operator
- `docs/SELL_SOURCE_CHECKLIST_TH.md` — checklist สำหรับผู้ขายซอร์ส
- `LINE_LOYALTY_HANDOFF.md` — รายละเอียด LINE OA และ loyalty

เวอร์ชันอื่น: INSTALL_GUIDE.html (v2 · ฉบาดอัจฉริยะ 4 ขั้น)