📘 คู่มือการตั้งค่าระบบ taskyapp Bot & Dashboard

ฉบับสมบูรณ์ สำหรับการนำระบบไปใช้งานจริงในบริษัท

📘 คู่มือการตั้งค่าและใช้งานระบบ taskyapp (ฉบับต่อตรง HTTPS — ไม่ใช้ Apps Script)

> คู่มือนี้จะพาคุณตั้งค่าระบบ **taskyapp** (ตัวช่วยจัดการงาน Jira ผ่าน Google Chat และ Dashboard) ตั้งแต่ขั้นตอนแรกสุดจนระบบรันได้สำเร็จ โดยเชื่อมต่อตรงกับเซิร์ฟเวอร์ด้วยระบบความปลอดภัย HTTPS ทันที (ไม่ต้องใช้ Google Apps Script เป็นตัวกลาง)

📑 สารบัญ

1. เตรียมความพร้อมก่อนเริ่ม
2. ขั้นตอนที่ 1: ตั้งค่าฐานข้อมูล Supabase
3. ขั้นตอนที่ 2: เตรียม API Key ของ Jira Cloud
4. ขั้นตอนที่ 3: สร้าง Webhook สำหรับ Google Chat (การแจ้งเตือน)
5. ขั้นตอนที่ 4: สร้าง Gemini API Key (สมองบอท)
6. ขั้นตอนที่ 5: รันสคริปต์ตั้งค่าระบบอัตโนมัติ
7. ขั้นตอนที่ 6: อัปโหลดโค้ดขึ้น GitHub และ Deploy ไป Vercel
8. ขั้นตอนที่ 7: เชื่อมบอทเข้า Google Chat (ต่อตรง HTTPS — No Apps Script)
9. ขั้นตอนที่ 8: ตั้งค่า Jira Webhook (ส่งแจ้งเตือนกลับห้องแชท)
10. ขั้นตอนที่ 9: ลงทะเบียนชื่อเล่นทีมใน Dashboard
11. คู่มือคำสั่งสั่งงานบอทภาษาไทย (NLP Command Guide)
12. แก้ปัญหาที่พบบ่อย (Troubleshooting)

---

1. เตรียมความพร้อมก่อนเริ่ม (Prerequisites)

เพื่อเริ่มทำโปรเจกต์นี้ให้เสร็จและพร้อมใช้งานจริง คุณต้องเตรียมสิ่งต่อไปนี้:

- **Node.js** (เวอร์ชัน 18 ขึ้นไป) ติดตั้งลงในเครื่องคอมพิวเตอร์ของคุณ
- บัญชี **GitHub** และบัญชี **Vercel** (สำหรับโฮสติ้งเซิร์ฟเวอร์เว็บฟรี)
- บัญชี **Supabase** (สำหรับเป็นฐานข้อมูล PostgreSQL ฟรี)
- บัญชี **Jira Cloud** (ที่มีสิทธิ์ผู้ดูแลระบบ เพื่อเข้าถึง API และตั้งค่า Webhook)
- สเปซใน **Google Chat** ที่ต้องการเพิ่มบอท

ขั้นตอนที่ 1: ตั้งค่าฐานข้อมูล Supabase

1. เข้าไปที่ 👉 Supabase ล็อกอินด้วยบัญชี GitHub ของคุณ
2. คลิก `New Project` -> เลือก Organization ของคุณ
3. ตั้งชื่อโครงการ เช่น `taskyapp-db` และ **ตั้งรหัสผ่านฐานข้อมูล (Database Password) จดใส่ Notepad ไว้ ห้ามลืม!**
4. เลือก Region เป็น `Southeast Asia (Singapore)` -> คลิก `Create new project`
5. รอให้โปรเจกต์สร้างเสร็จ (ประมาณ 1 นาที) จากนั้นคลิกปุ่ม `Connect` ที่แถบเมนูด้านบนสุด
6. เลือกประเภท `Transaction pooler` -> เลือกแถบย่อย `URI`
7. คัดลอกลิงก์เชื่อมต่อฐานข้อมูลมาเก็บไว้ใน Notepad และ**เปลี่ยนคำว่า** `[YOUR-PASSWORD]` **ในลิงก์เป็นรหัสผ่านฐานข้อมูลจริงของคุณ** ตัวอย่างเช่น:

```
postgresql://postgres.xxxx:your_password@aws-0-ap-southeast-1.pooler.supabase.com:6543/postgres
```

ขั้นตอนที่ 2: เตรียม API Key ของ Jira Cloud

1. **JIRA_DOMAIN:** ดูจาก URL เวลาล็อกอิน Jira ของคุณ เช่น `your-company.atlassian.net`
2. **JIRA_EMAIL:** อีเมลที่คุณใช้ในการล็อกอินเข้าใช้งาน Jira
3. **JIRA_API_TOKEN:**

- เข้าไปที่ 👉 Atlassian API Tokens

- คลิก `Create API token` -> ตั้งชื่อว่า `taskyapp-token` -> กด `Create`

- คัดลอกรหัส Token ที่แสดงขึ้นมาบันทึกเก็บไว้ทันที (รหัสจะขึ้นโชว์ครั้งเดียวเท่านั้น)

4. **JIRA_PROJECT_KEY:** ตัวย่อหน้าชื่อตั๋วงานของโครงการคุณ เช่น `KAN` หรือ `DEV`

ขั้นตอนที่ 3: สร้าง Webhook สำหรับ Google Chat (การแจ้งเตือน)

> ส่วนนี้ใช้ส่งแจ้งเตือนจากบอร์ด Jira วิ่งเข้ามาโพสต์บอกสมาชิกในห้องแชท

1. เปิดแอป **Google Chat** บนคอมพิวเตอร์ของคุณ
2. เข้าไปในห้องสเปซ (Space) ที่ต้องการรับการแจ้งเตือน
3. คลิกที่ **ชื่อห้องด้านบนสุด** -> เลือก `Apps & integrations` (แอปและการผสานรวม)
4. คลิก `Webhooks` -> กด `Add webhook` (เพิ่ม Webhook)
5. ตั้งชื่อบอทผู้แจ้งเตือน เช่น `taskyapp Notifier` -> กด `Save`
6. คัดลอก URL Webhook ยาวๆ ที่ระบบให้มาบันทึกเก็บไว้ใน Notepad

---

ขั้นตอนที่ 4: สร้าง Gemini API Key (สมองบอท)

1. เข้าไปที่เว็บ 👉 Google AI Studio
2. คลิกปุ่ม `Create API key`
3. เลือกบัญชีหรือสร้าง Project ฟรี จากนั้นคัดลอกรหัส **API Key (ขึ้นต้นด้วย AIzaSy...)** เก็บไว้

ขั้นตอนที่ 5: รันสคริปต์ตั้งค่าระบบอัตโนมัติ

1. แตกไฟล์โปรเจกต์ `taskyapp-handoff.zip` ออกมา
2. เปิด Terminal หรือ Command Prompt ในโฟลเดอร์โครงการ (โฟลเดอร์เดียวกับที่มีสคริปต์ `setup.bat` และ `setup.sh`)
3. รันตัวช่วยตั้งค่า:

- **ระบบ Windows:** ดับเบิลคลิกไฟล์ `setup.bat`
- **ระบบ Mac/Linux:** รันคำสั่ง `chmod +x setup.sh` แล้วรัน `./setup.sh`

4. สคริปต์จะติดตั้ง Dependencies (`npm install`) และเปิดหน้าถามตอบการตั้งค่าแบบทีละขั้น
5. นำค่าที่คุณจดบันทึกไว้ใน Notepad กรอกตอบคำถามให้ครบทั้ง 7 รายการ
6. เมื่อตอบคำถามเสร็จ ระบบจะสร้างไฟล์ `.env` ขึ้นมาโดยอัตโนมัติ พร้อมต่อเข้าฐานข้อมูลเพื่อ**สร้างตารางข้อมูลใน Supabase ให้สำเร็จทั้งหมด 7 ตาราง**ทันที

ขั้นตอนที่ 6: อัปโหลดโค้ดขึ้น GitHub และ Deploy ไป Vercel

6.1 นำโค้ดขึ้น GitHub

1. เข้าไปสร้าง Repository ใหม่แบบ `Private` บน GitHub ของคุณ ตั้งชื่อว่า `taskyapp`
2. อัปโหลดเฉพาะไฟล์ที่อยู่**ภายในโฟลเดอร์** `nextjs-app` (ไม่ต้องอัปโหลดโฟลเดอร์ `node_modules` และไม่ต้องเอาตัวโฟลเดอร์แม่ `nextjs-app` ขึ้น เอาขึ้นเฉพาะไฟล์ข้างในทั้งหมด) ขึ้นไปยัง GitHub

6.2 Deploy ขึ้น Vercel

1. เข้าเว็บ Vercel -> ล็อกอินด้วยสิทธิ์ GitHub
2. กด `Add New...` -> เลือก `Project` -> คลิกปุ่ม `Import` หลัง Repository `taskyapp`
3. ⚠️ **สำคัญมากที่สุด (ห้ามลืม!):** ในเมนูตั้งค่า Project หน้า Vercel ให้มองหาหัวข้อ `Root Directory` -> คลิกปุ่ม `Edit` -> พิมพ์คำว่า `nextjs-app` เข้าไปแล้วกดบันทึก
4. เลื่อนลงมาที่หัวข้อ `Environment Variables`
5. เปิดไฟล์ `.env` ที่อยู่ในเครื่องคอมพิวเตอร์ของคุณ (ที่สร้างไว้ในขั้นตอนที่ 5) กดคลุมทั้งหมดแล้วคัดลอก จากนั้นมาคลิกขวาและวางในช่องป้อนของ Vercel ระบบจะตรวจจับและกรอกค่าตัวแปรทั้งหมดให้คุณโดยอัตโนมัติ
6. คลิกปุ่ม `Deploy` และรอ 1-2 นาทีจนเสร็จสิ้น จากนั้นให้จดจำ **ลิงก์หน้าเว็บของคุณ (Vercel URL)** เอาไว้ใช้งาน

ขั้นตอนที่ 7: เชื่อมบอทเข้า Google Chat (ต่อตรง HTTPS — No Apps Script)

> ในอดีตระบบส่วนนี้ต้องใช้ Google Apps Script ทำหน้าที่รับสัญญาณ แต่โปรเจกต์นี้ได้รับการพัฒนาให้สามารถต่อตรงเข้า HTTPS ได้โดยไม่ต้องมี Apps Script เพื่อความเสถียรและเร็วขึ้นสูงสุด

1. เข้าสู่หน้า 👉 Google Cloud Console เลือก Project ของคุณ
2. ที่ช่องค้นหาด้านบน พิมพ์ค้นหาคำว่า `Google Chat API` -> คลิกเข้ามาแล้วกดเปิดใช้งาน (`Enable`)
3. คลิกแถบ `Configuration` (การตั้งค่า) ทางด้านซ้าย และกรอกรายละเอียดดังนี้:

- **App name:** `taskyapp`

- **Avatar URL:** (สามารถเว้นว่างได้)

- **Description:** `บอทอัจฉริยะประมวลผลงาน Jira ด้วย AI`

- **Functionality:** ติ๊กเครื่องหมายถูกทั้ง **[x] Receive 1:1 messages** และ **[x] Join spaces...**

- **Connection settings:** คลิกเลือก `HTTPS URL`

- **URL:** ใส่ลิงก์เว็บ Vercel ของคุณต่อท้ายด้วย `/api/webhook`

ตัวอย่างการกรอก: `https://your-app.vercel.app/api/webhook`

- **Visibility:** เลือกให้คนทุกคนในองค์กรเห็นบอทได้

4. คลิกปุ่ม `Save` ด้านล่างสุด
5. เปิดห้องสเปซใน Google Chat -> คลิกหัวข้อห้องแชทด้านบน -> เลือก `Apps & integrations` -> กด `Add apps` ค้นหาบอทชื่อ `taskyapp` แล้วกดเชิญเข้าร่วมกลุ่ม

ขั้นตอนที่ 8: ตั้งค่า Jira Webhook (ส่งแจ้งเตือนกลับห้องแชท)

เพื่อให้ตั๋วงานใน Jira เชื่อมโยงกลับมาแสดงผลทันทีที่ห้องแชท Google Chat เมื่อมีคนทำงานเสร็จหรือเคลื่อนไหวบอร์ด:

1. เข้าหน้าเว็บ Jira Cloud ของคุณ -> คลิกรูปฟันเฟือง **Settings** มุมขวาบน -> เลือกหัวข้อ `System`
2. เลื่อนแถบเมนูด้านซ้ายลงล่างสุด คลิกเลือกเมนู `Webhooks`

3. คลิกปุ่ม `Create a Webhook`
4. ตั้งชื่อ Webhook และกรอกข้อมูลสำคัญ:

- **URL:** กรอกลิงก์ Vercel ของคุณตามด้วย `/api/jira/webhook`

ตัวอย่าง: `https://your-app.vercel.app/api/jira/webhook`

5. ในช่องติ๊กเลือก Event (เหตุการณ์) ให้เลื่อนมาหาหัวข้อ **Issue** และติ๊กเลือกเหตุการณ์ดังนี้:

- [x] **created**

- [x] **updated**

6. เลื่อนลงไปด้านล่างสุด แล้วกดปุ่ม `Create` เพื่อบันทึก

ขั้นตอนที่ 9: ลงทะเบียนชื่อเล่นทีมใน Dashboard

บอทอัจฉริยะจะไม่รู้ว่าชื่อเล่นที่คุณสั่งงานในห้องแชท (เช่น "สมศักดิ์" หรือ "pm") เชื่อมโยงกับผู้ใช้งานคนไหนในระบบ Jira Cloud คุณต้องผูกข้อมูลไว้ล่วงหน้าก่อนสั่งงานครั้งแรก:

1. ล็อกอินเข้าใช้งาน Dashboard ของคุณผ่านหน้าเว็บเบราว์เซอร์หลัก (ล็อกอินด้วยบัญชีสิทธิ์ผู้ดูแลระบบ เช่น `admin` / `admin123`)
2. คลิกไปที่แท็บ `Team` (จัดการสมาชิกทีม)
3. กดเพิ่มสมาชิกใหม่ กรอกชื่อเล่นภาษาไทย (เช่น `สมศักดิ์`) และค้นหาเลือกชื่อบัญชีผู้ใช้บน Jira Cloud ให้ตรงกันเพื่อผูกข้อมูล
4. กดบันทึกข้อมูล ทีมงานทุกคนจะพร้อมสั่งงานผ่านบอทได้ทันที

คู่มือคำสั่งสั่งงานบอทภาษาไทย (NLP Command Guide)

คุณสามารถแท็กชื่อบอทและพิมพ์ภาษาไทยหรือภาษาอังกฤษได้อย่างเป็นธรรมชาติ AI (Gemini) จะทำการจัดวิเคราะห์เจตนาและสั่งการระบบเบื้องหลังให้อัตโนมัติ:

1. คำสั่งสร้างตั๋วงานใหม่ (Create Issue)

- **การสั่งสร้าง Epic (งานหลัก):**

- `@taskyapp สร้างงานหลัก ออกแบบ UI หน้าแรก`
- `@taskyapp ช่วยเปิด Epic เขียนโครงสร้างฐานข้อมูลใหม่`
  - **การสั่งสร้าง Task (งานย่อย/จ๊อบ):** (ต้องมีคำคีย์เวิร์ดบอกว่าเป็นงานย่อย เช่น งานย่อย, จ๊อบ, job, sub-task)
- `@taskyapp สร้างจ๊อบ ออกแบบปุ่มสมัครสมาชิก พ่อคือ KAN-5` (ผูกเข้ากับงานหลัก KAN-5)
- `@taskyapp สร้างงานย่อย เขียน API เชื่อมระบบ ส่งมอบให้ สมศักดิ์`

2. คำสั่งแก้ไขข้อมูลงาน (Update)

- **สั่งเปลี่ยนข้อมูลใดๆ ของตั๋วงาน:**

- `@taskyapp แก้ไขตั๋ว KAN-12 เปลี่ยนกำหนดส่งเป็นวันที่ 2026-07-31`
- `@taskyapp ย้ายงาน KAN-3 ให้ สมศักดิ์ รับผิดชอบแทน`
- `@taskyapp อัปเดต KAN-4 ความสำคัญ สูง`

3. คำสั่งเปลี่ยนสถานะงาน (Transition)

- **สั่งเลื่อนบอร์ดข้ามสถานะ:**

- `@taskyapp ปิดงาน KAN-5` (เปลี่ยนเป็น Done)
- `@taskyapp ย้าย KAN-6 ไปกำลังทำ` (เปลี่ยนเป็น In Progress)
- `@taskyapp เลื่อน KAN-7 เป็นรอรีวิว` (เปลี่ยนเป็น In Review)

4. คำสั่งเรียกดูข้อมูลและค้นหา (Read & Search)

- **ดูรายละเอียดตั๋วชิ้นใดชิ้นหนึ่ง:** `@taskyapp ขอรายละเอียดตั๋ว KAN-8`
- **ค้นหางานที่มีเนื้อหาเกี่ยวข้อง:** `@taskyapp ค้นหาคำว่า บั๊กล็อกอิน`
- **ดูสรุปสถิติจำนวนงานค้าง:** `@taskyapp สรุปภาพรวมงานค้างทั้งหมดให้หน่อย`

แก้ปัญหาที่พบบ่อย (Troubleshooting)

❌ พิมพ์สั่งงานบอทแล้ว บอทนิ่งเฉยไม่มีข้อความตอบกลับ

- **สาเหตุที่ 1:** ปัญหาเซิร์ฟเวอร์หลับ (Vercel Serverless Function Sleep) หากบอทไม่ได้ถูกพิมพ์ใช้งานเป็นระยะเวลานาน การส่งสัญญาณครั้งแรกอาจใช้เวลานานกว่า 10 วินาทีจนทำให้เกิด Connection Timeout

- **วิธีแก้ไข:** ลองเปิดหน้าเว็บ Dashboard ของคุณให้ระบบตื่นตัว จากนั้นกลับมาพิมพ์แท็กบอทสั่งงานใหม่อีกครั้ง

- **สาเหตุที่ 2:** กรอกข้อมูลใน Google Cloud Console ผิดพลาด

- **วิธีแก้ไข:** ตรวจสอบให้มั่นใจว่าตัวเลือก Connection settings ในหน้า Google Chat API ถูกกำหนดเป็น `HTTPS URL` และตัวสะกดของ URL ปลายทางลงท้ายด้วย `/api/webhook` อย่างถูกต้องไม่มีเว้นวรรค

- **สาเหตุที่ 3:** การกรอกรหัส `GEMINI_API_KEY` ใน Environment Variables ผิดพลาด ทำให้ AI ไม่ทำงาน

- **วิธีแก้ไข:** ตรวจสอบรหัส API Key ของคุณในแท็บ Settings ของ Vercel

❌ บอทตอบกลับในห้องแชทว่าทำตามคำสั่งสำเร็จ แต่ไม่มีความเปลี่ยนแปลงในหน้า Jira จริง

- **สาเหตุ:** ข้อมูลยืนยันสิทธิ์ของ Jira ล้มเหลว (`JIRA_API_TOKEN` ผิดพลาด หรือสิทธิ์การใช้อีเมล `JIRA_EMAIL` ไม่ผ่าน)
- **วิธีแก้ไข:** ไปที่ Vercel -> ตรวจสอบว่าในตัวแปรระบบมี 4 ค่าสำคัญที่ถูกต้องครบถ้วน และทดสอบลิงก์ `/api/health` บนเบราว์เซอร์เพื่อดูว่าสถานะการเชื่อมต่อ Jira แจ้งว่า `connected` หรือไม่

— จบคู่มือ —