

Strategic Guide for Canopy Tax CRM Automation  
Overcoming API Limitations & Defining Operational Processes  
Prepared by: Yunho

Status: Pre-implementation Analysis

### 1. The Technical Reality

After analyzing the Canopy Tax Public API (v3), the following fact is confirmed:

**No Direct File Trigger:** The API does NOT provide any way to automatically detect or "push" a signal when a client uploads a file.

**Vendor Restriction:** This is a strategic limitation by Canopy. Therefore, we cannot rely on the API alone to know when a file exists.

### 2. Two Operational Solutions (Decision Required)

To bridge this gap, you must choose one of the following two "signal" methods:

#### Option A: Manual Field Update (The "Status" Method)

How it works: Your staff must manually update a "Custom Field" in Canopy (e.g., set File Status to 'Uploaded') after they see a file.

**Pros:** High accuracy since a human verifies it first.

**Cons:** High risk of human error. If staff forget to update the field, the automation never starts.

#### Option B: Email-to-Webhook (The "Sensor" Method) - Recommended

How it works: We turn on Canopy's "Client uploads a file" email notification. My system (n8n) acts as a digital sensor that "listens" to your inbox and triggers the automation.

**Pros:** Much closer to full automation. No extra steps for your staff in the CRM.

**Cons:** Canopy emails are inconsistent. They mix Portal uploads, Secure Link uploads, and other alerts. Designing a precise filter to separate these takes significant development and testing time.

### 3. Recommended Workflow: The "Human-in-the-loop" Filter

Because the API is unreliable, 100% full automation might send wrong messages to clients. I recommend this Hybrid Process:

**Signal:** Canopy sends an email alert (Option B).

**Processing:** n8n parses the email to find the Client ID and details.

**Verification (The Filter):** The system asks a staff member (via a simple button or link) to "Approve" the notification.

**Final Action:** Once approved, the KakaoTalk/Email is sent to the client.

### 4. Why This Requires Significant Time

This project is not about "connecting dots"; it is about "refining raw, unreliable data into a stable business process."

Pattern Analysis: We must analyze various Canopy email subjects and bodies to ensure the "sensor" works every time.

Exception Handling: We must build logic for multiple file uploads, missing client names, or delayed emails.

Operational SOP: We need to align your staff's behavior with the system's logic.

## 5. Conclusion

The success of this project depends on process over code. Please decide if you prefer the Manual Update (A) or Email Sensor (B) model so we can define the final scope.