DIRECTUS INTRODUCTION

https://directus.io is a headless CMS that instantly turns your SQL database into REST and GraphQL APIs

Ho Chi Minh, 2025.
Thang Le

TRADITIONAL CMS

- **Monolithic Architecture:** The backend (database and content management) and frontend (templates and rendering) are integrated into a single system.

- **Advantages**:
  - **Ease of Use**: User-friendly for non-technical users, with WYSIWYG editors and pre-built themes.
  - **All-in-One Solution**: No need to build a separate frontend; everything is managed within the CMS.
  - **Quick Setup**: Ideal for simple websites or blogs where the frontend is a single website.

- **Disadvantages**:
- **Limited Flexibility**: Hard to repurpose content for different platforms (e.g., mobile apps, IoT devices) without additional development.
- **Tightly Coupled**: Changes to the frontend often require backend modifications, slowing down development.
- **Vendor Lock-in**: Reliance on a specific CMS platform can make migration difficult.

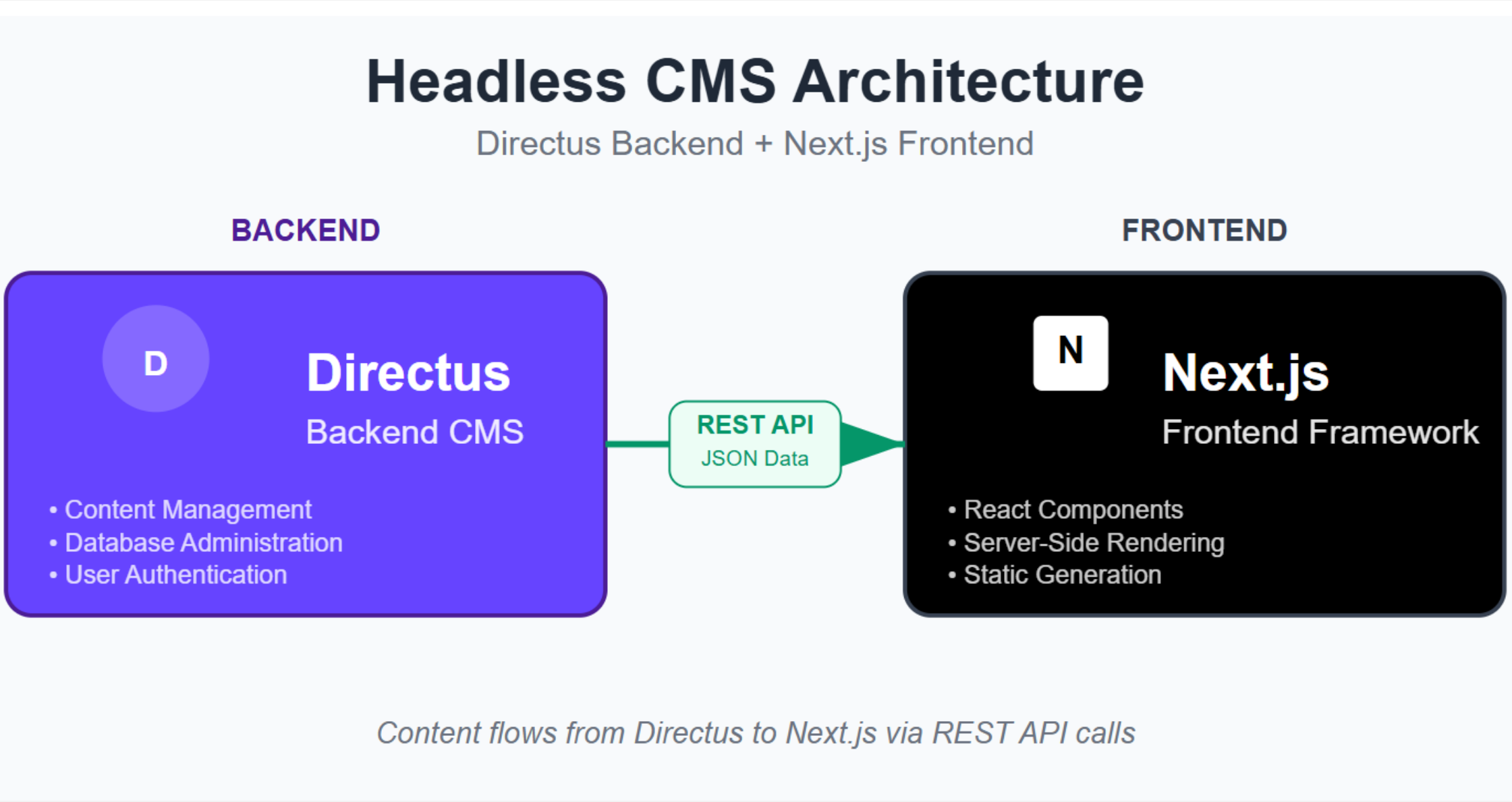
HEADLESS CMS

- A headless CMS decouples the backend (content management and storage) from the frontend (presentation layer). It provides content through APIs (REST or GraphQL), allowing developers to use any frontend technology to display the content.

- **Advantages**:
- **Flexibility**: Developers can choose any frontend framework or technology, enabling modern, dynamic experiences.
- **Multi-Channel Delivery**: Content can be reused across websites, mobile apps, IoT devices, or other platforms without modification.
- **Scalability**: The decoupled architecture allows the backend and frontend to scale independently.
- **Future-Proof**: Easier to adapt to new technologies or platforms since the CMS is not tied to a specific frontend.

- **Disadvantages**:
- **Increased Complexity**: Requires developers to build and maintain the frontend, which can be challenging for non-technical teams.
- **Higher Initial Setup**: Setting up a headless CMS and connecting it to a custom frontend takes more effort than a traditional CMS.
- **Learning Curve**: Non-technical users may find it less intuitive without a built-in frontend editor.

DIRECTUS & NEXT.JS ARCHITECTURE



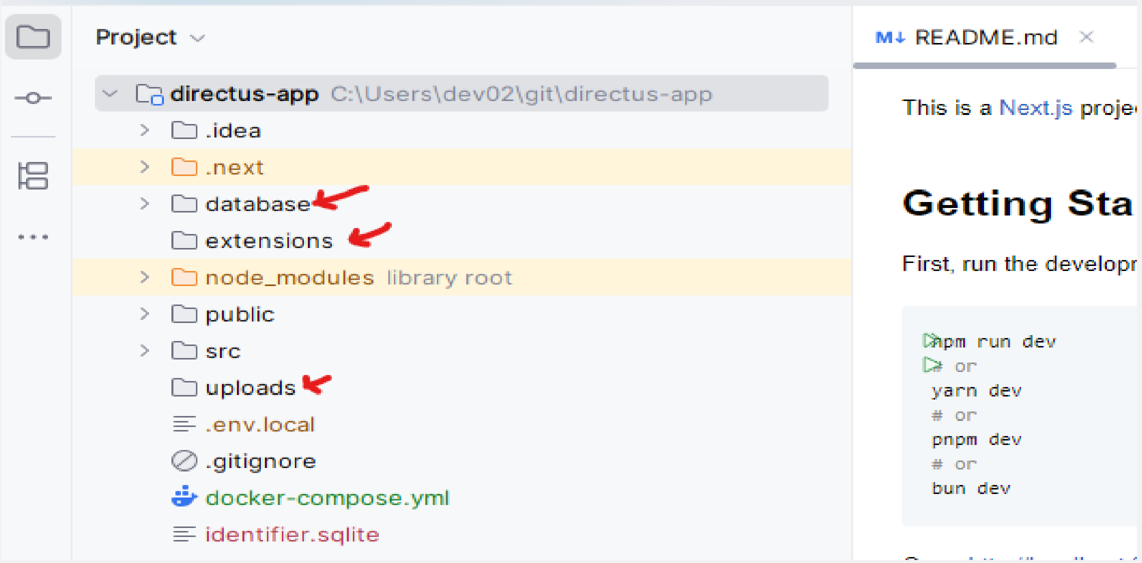
SETUP DIRECTUS & NEXT.JS

1. Setup & Run Directus from Docker

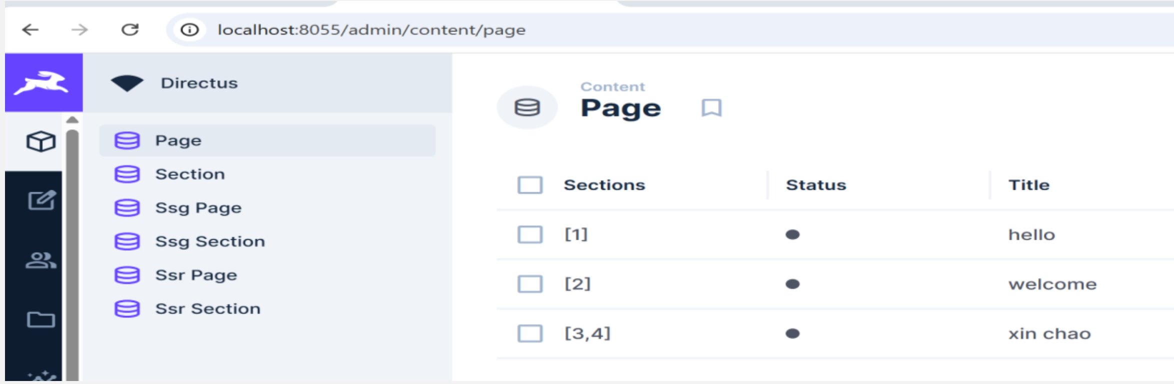
```yaml
version: "3"
services:
  directus:
    image: directus/directus:latest
    ports:
      - 8055:8055
    volumes:
      - ./database:/directus/database
      - ./uploads:/directus/uploads
      - ./extensions:/directus/extensions
    environment:
      KEY: "123456789123456789"
      SECRET: "123456789123456789123456789123456789"
      ADMIN_EMAIL: "admin@example.com"
      ADMIN_PASSWORD: "P@ssw0rd"
      DB_CLIENT: "sqlite3"
      DB_FILENAME: "/directus/database/data.db"
      WEBSOCKETS_ENABLED: "true"
```

2. Setup Next.js

npx create-next-app@latest directus-app



3. Access Directus Admin page: http://localhost:8055/

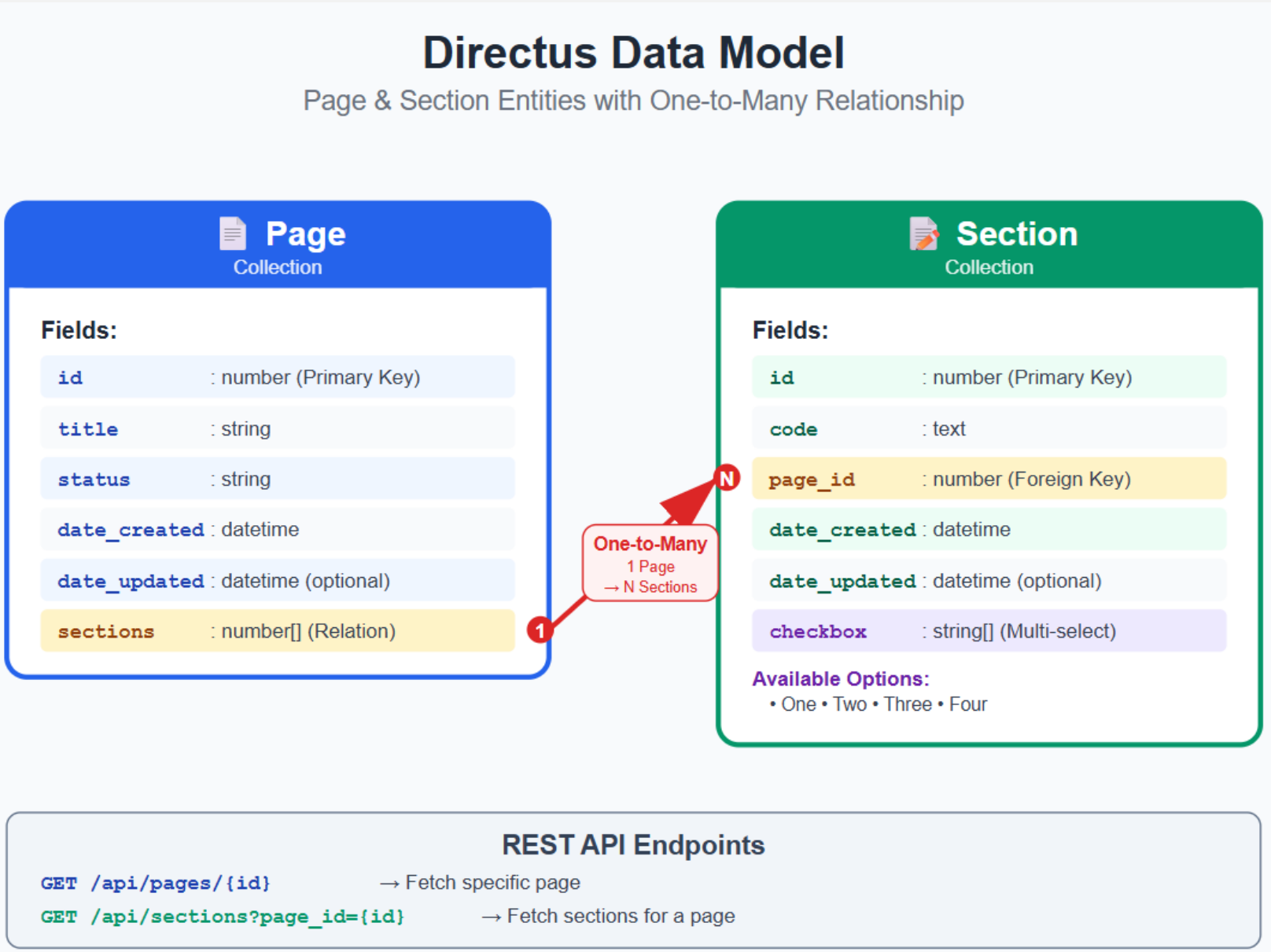


DATA MODEL IN DIRECTUS

Everything in Directus is organized like database tables and relationships.

Each content type becomes a **Collection** (like a table) with defined **Fields** (columns) that store your data

This relational approach makes Directus powerful - you can model any content structure from simple blogs to complex e-commerce systems, all accessible via clean APIs for your frontend applications.



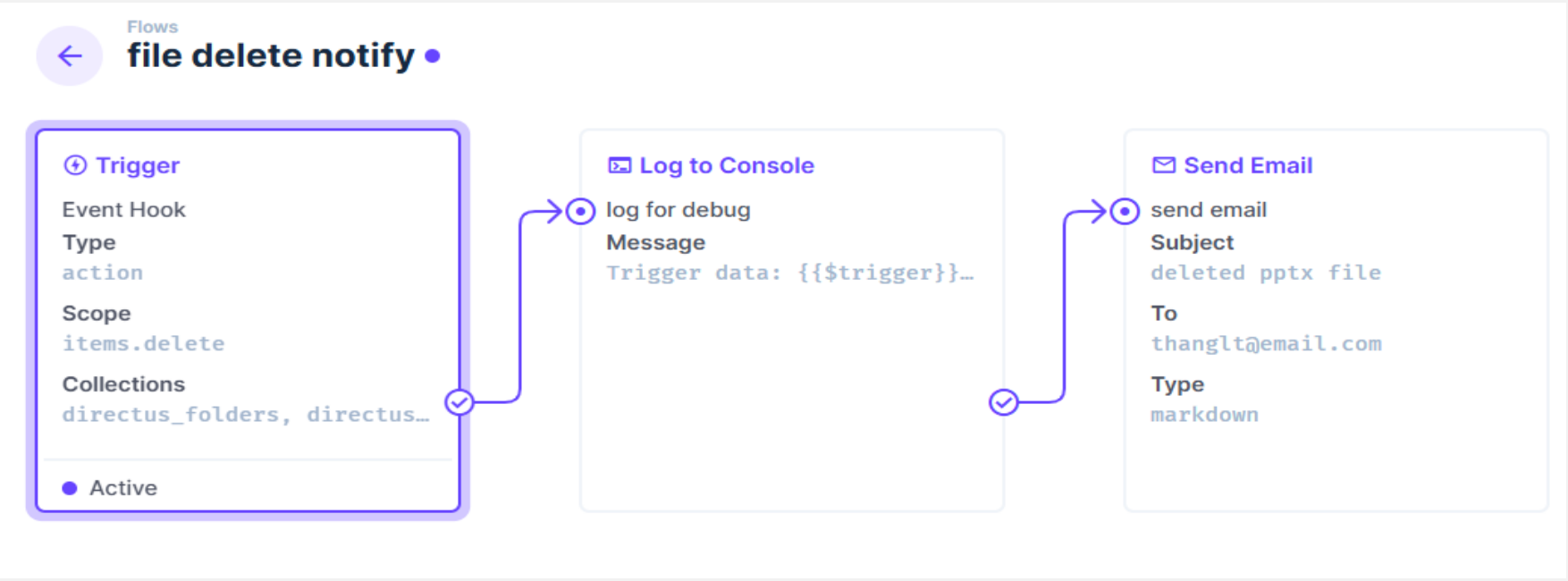
FLOW IN DIRECTUS

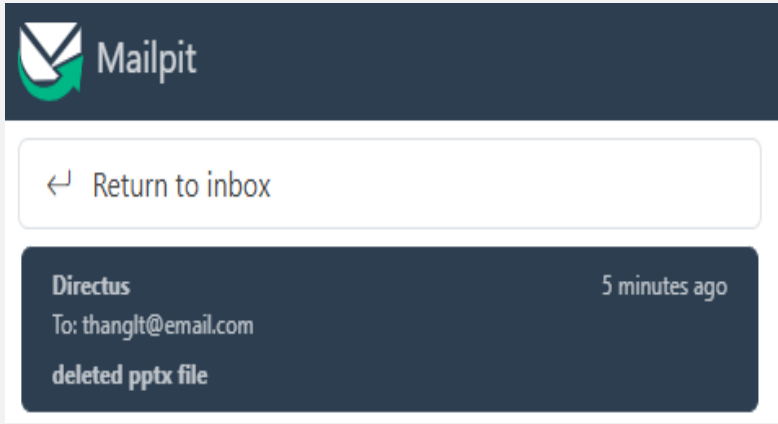
Flows in Directus are visual automation workflows that let you create business logic without writing code. Think of them as "if-this-then-that" automation rules that respond to events in your CMS.

Flows are **event-driven automation pipelines** that execute when specific triggers occur in your Directus instance.

1. **Triggers** - What starts the flow:
   - **Event Hook**: When data changes (create, update, delete items)
   - **Webhook**: External HTTP requests
   - **Schedule**: Time-based triggers (cron jobs)

2. **Operations** - What actions to perform:
   - **Send Email**: Notifications, alerts
   - **Transform Data**: Modify field values
   - **Condition**: If/else logic branching





EXTENSIONS IN DIRECTUS

Directus Extensions allow you to customize and extend the functionality of your Directus instance beyond its default capabilities.

Types of Extensions

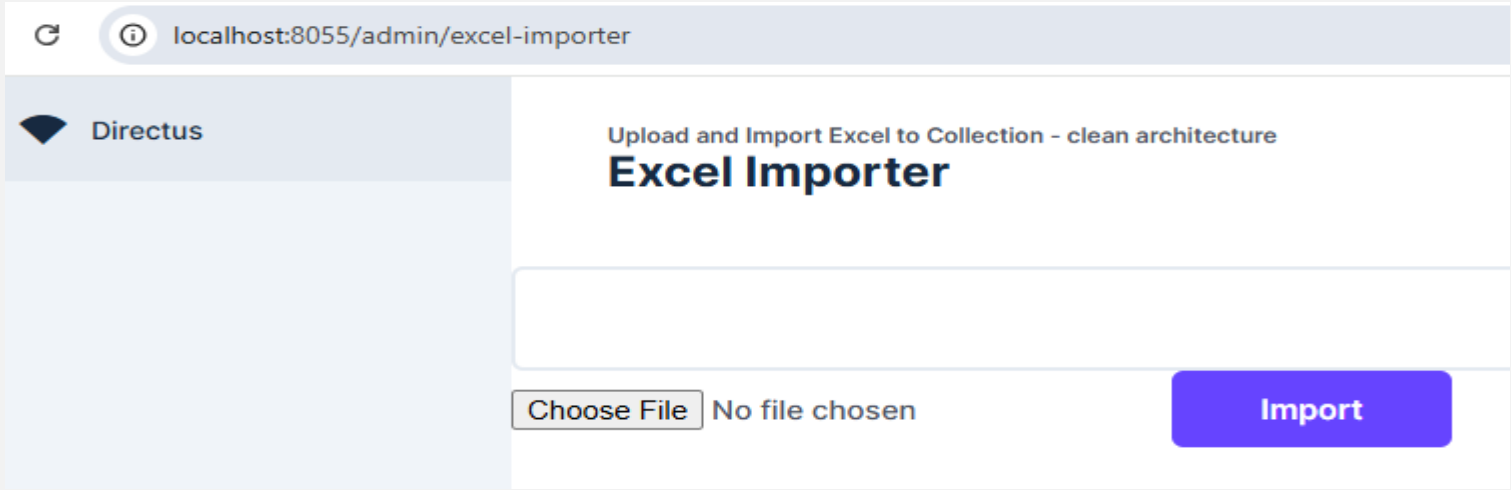
App Extensions (Frontend):

- **Interfaces**: Custom field input components (custom date pickers, rich text editors)
- **Displays**: How data appears in the admin interface
- **Layouts**: Custom ways to view and organize data collections
- **Modules**: Complete admin interface sections (custom dashboards, tools)
- **Panels**: Dashboard widgets and components

API Extensions (Backend):

- **Hooks**: Server-side event listeners (like Flows but in code)
- **Endpoints**: Custom API routes and logic
- **Operations**: Custom Flow operations beyond built-in ones

https://github.com/lengocanh/directus-extension-excel-importer



DEMO & Q&A

https://github.com/letrthang/directus-app