

n8n 아키텍처 및 노드 활용의 최종 권위 가이드: 심층 분석 및 엔지니어링 전략

1. 서론: n8n의 워크플로우 아키텍처와 데이터 처리 철학

현대의 자동화 환경은 단순한 선형적 트리거-액션(Trigger-Action) 모델을 넘어, 복잡한 비즈니스 로직과 상태 관리, 그리고 AI 기반의 추론을 요구하는 방향으로 진화하고 있습니다. 이러한 맥락에서 n8n은 '공정한 코드(Fair-code)' 배포 모델과 노드 기반의 시각적 프로그래밍 인터페이스를 결합하여, 개발자와 자동화 엔지니어에게 전례 없는 유연성을 제공합니다. 본 보고서는 n8n의 핵심 구성 요소인 트리거(Trigger), 액션(Action), 유틸리티(Utility), AI/Agent, 그리고 커스터마이징(Customization) 노드 중 가장 사용 빈도가 높은 상위 노드들을 선별하여, 그 기술적 메커니즘과 최적화 전략을 심층적으로 분석합니다.

1.1 노드 기반 실행 모델과 데이터 구조

n8n의 가장 큰 특징은 데이터를 처리하는 독자적인 방식에 있습니다. 모든 데이터는 JSON 객체의 배열로 노드 간에 전달되며, 각 객체는 '아이템(Item)'이라고 불립니다. 이는 단순한 키-값 쌍(Key-Value Pair)의 전달을 넘어, 복잡한 데이터 구조와 바이너리 데이터(이미지, 파일 등)를 동시에 처리할 수 있는 기반을 제공합니다.

아이템 기반 실행(Item-based Execution): 대부분의 노드는 입력으로 들어오는 아이템의 수만큼 실행됩니다. 예를 들어, 10개의 행이 있는 구글 시트 데이터를 읽어오면, 연결된 다음 노드는 기본적으로 10번 실행되거나, 10개의 아이템을 한 번에 처리하도록 설계되어야 합니다.

데이터 링킹(Data Linking): n8n은 워크플로우 내에서 데이터의 계보(Lineage)를 추적합니다. 이를 통해 엔지니어는 직전 노드뿐만 아니라, 흐름상 상위에 위치한 모든 노드의 출력 데이터에 접근할 수 있습니다. 이러한 아키텍처적 이해는 각 노드의 옵션을 설정할 때 필수적입니다. 특히 루프(Loop), 병합(Merge), 그리고 코드(Code) 실행 시 데이터 구조를 어떻게 다루느냐에 따라 자동화의 성능과 안정성이 결정됩니다.

2. 트리거(Trigger) 노드: 워크플로우의 시작점

트리거 노드는 외부 세계의 이벤트나 시간의 흐름을 n8n의 실행 컨텍스트로 변환하는 진입점입니다. 트리거의 선택은 전체 시스템의 반응 속도(Latency)와 리소스 효율성을 결정짓는 핵심 요소입니다. 아키텍처 관점에서 트리거는 크게 '폴링(Polling)' 방식과 '푸시(Push)' 방식으로 나뉩니다.

2.1 Webhook (웹훅) - 실시간 이벤트 수신 표준

Webhook 노드는 n8n을 강력한 백엔드 API 서버로 변환시키는 가장 범용적인 트리거입니다. 외부 시스템(GitHub, Stripe, 사용자 정의 앱 등)에서 발생하는 이벤트를 실시간으로 수신하여 워크플로우를 즉시 실행합니다.

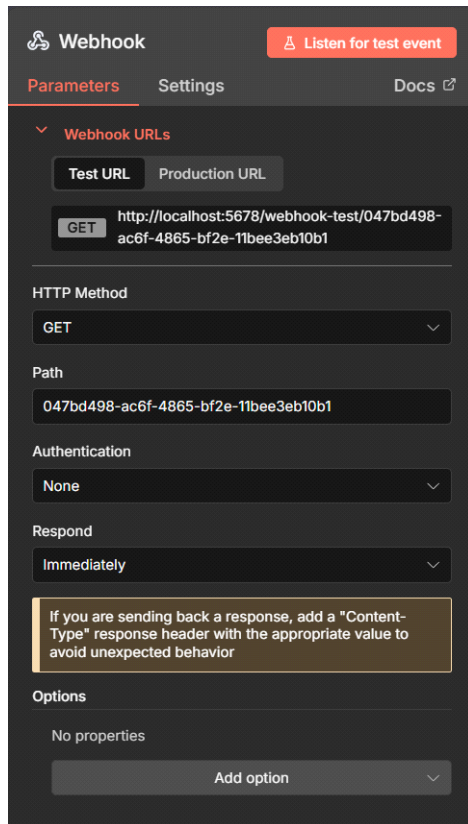
핵심 용도 및 아키텍처

이벤트 기반 아키텍처(Event-Driven Architecture) 구현: 데이터 변경 즉시 반응해야 하는 로직(예: 결제

완료 시 이메일 발송)에 필수적입니다. 폴링 방식과 달리 불필요한 API 호출을 하지 않으므로 리소스 효율이 높습니다.

커스텀 API 엔드포인트: n8n 워크플로우 자체를 마이크로서비스처럼 활용하여, 외부에서 데이터를 요청하고 처리 결과를 반환받는 동기식(Synchronous) API로 구성할 수 있습니다.

주요 옵션 설정 및 고급 구성



The image shows a 'Webhook' configuration panel with a dark theme. At the top, there's a 'Listen for test event' button. Below it are tabs for 'Parameters', 'Settings', and 'Docs'. The 'Webhook URLs' section is expanded, showing 'Test URL' and 'Production URL' tabs. The 'Test URL' is set to 'http://localhost:5678/webhook-test/047bd498-ac6f-4865-bf2e-11bee3eb10b1'. Below this, the 'HTTP Method' is set to 'GET', the 'Path' is '047bd498-ac6f-4865-bf2e-11bee3eb10b1', and 'Authentication' is set to 'None'. The 'Respond' section is set to 'Immediately'. A yellow box contains a note: 'If you are sending back a response, add a "Content-Type" response header with the appropriate value to avoid unexpected behavior'. At the bottom, there's an 'Options' section with 'No properties' and an 'Add option' button.

- **HTTP Method (HTTP 메서드):** GET, POST, PUT, PATCH, DELETE를 지원합니다. 데이터를 수신할 때는 주로 POST가 사용되며, 웹 브라우저의 접근을 처리할 때는 GET이 사용됩니다.
- **Authentication (인증):** 보안은 Webhook의 가장 중요한 고려사항입니다.
- **Basic Auth:** 사용자 이름과 비밀번호를 통한 기본 인증.
- **Header Auth:** API Key 등을 헤더에 포함시켜 검증하는 방식. 보안상 가장 권장되는 방식 중 하나입니다.
- **No Auth:** 공개된 엔드포인트로, 누구나 접근 가능하므로 테스트 외에는 사용을 지양해야 합니다.
- **Response Mode (응답 모드):**
- **On Received:** 요청을 받자마자 200 OK를 반환하고 워크플로우는 비동기로 실행됩니다. 빠른 응답이 필요한 외부 서비스(예: Slack Interactivity)와의 연동에 적합합니다.
- **Last Node:** 워크플로우의 마지막 노드가 실행될 때까지 기다렸다가, 그 결과를 응답으로 반환합니다. 이를 통해 데이터를 가공하여 클라이언트에게 돌려주는 동기식 API를 구축할 수 있습니다.
- **Path (경로):** 기본적으로 생성되는 난수 URL 외에, /webhook/my-custom-path와 같이 경로를 지정할 수 있으나, n8n의 고유 ID가 포함된 URL 구조는 유지됩니다.

2.2 Schedule Trigger (스케줄) - 주기적 실행의 중심

과거 Cron 노드로 불렸던 Schedule Trigger는 특정 시간 간격이나 정해진 시각에 워크플로우를 실행함

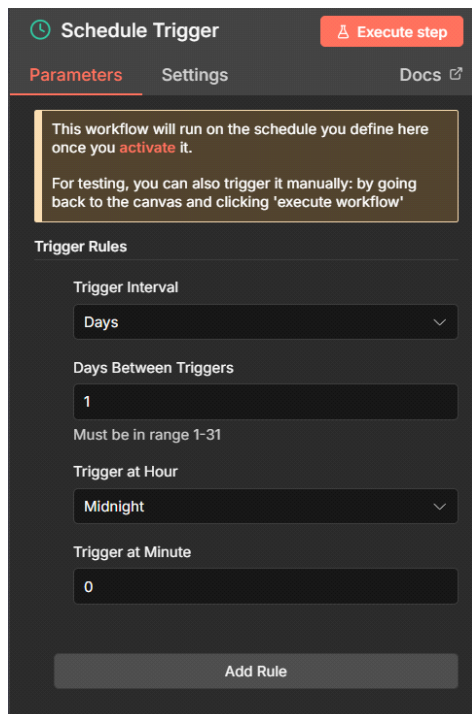
니다. 데이터 백업, 주간 리포트 생성, 정기적인 상태 점검 등에 사용됩니다.

핵심 용도 및 아키텍처

배치 처리(Batch Processing): 실시간성이 필요 없는 대용량 데이터 처리나, API 사용량 제한을 피하기 위해 야간에 작업을 수행해야 할 때 사용됩니다.

폴링 대체: 웹훅을 지원하지 않는 레거시 시스템의 데이터를 주기적으로 확인(Poll)하여 변경 사항을 감지하는 로직의 시작점으로 활용됩니다.

주요 옵션 설정 및 고급 구성



- **Trigger Interval (트리거 간격):**
- **Basic Intervals:** 분, 시, 일, 주 단위의 간편 설정.
- **Custom (Cron Expression):** 리눅스 크론 문법을 사용하여 정교한 스케줄링이 가능합니다. 예: 0 9 * * 1-5 (평일 오전 9시 실행).
- **Timezone (타임존):** 가장 빈번한 오류 원인 중 하나입니다. 기본적으로 n8n 인스턴스의 시스템 타임존을 따르지만, 워크플로우별로 타임존을 오버라이드할 수 있습니다. 글로벌 서비스를 운영할 경우 UTC 기준으로 설정하는 것이 데이터 정합성 유지에 유리합니다.
- **Concurrency (동시성) 관리:** 만약 1분마다 실행되는 워크플로우가 처리에 2분이 걸린다면, 실행이 중첩되어 시스템 부하를 일으킬 수 있습니다. 이를 방지하기 위해 워크플로우 설정에서 동시 실행을 제한하거나, 이전 실행이 완료될 때까지 대기하도록 아키텍처를 설계해야 합니다.

2.3 n8n Form Trigger (폼 트리거) - 사용자 입력 인터페이스

n8n Form Trigger는 별도의 프론트엔드 개발 없이 사용자로부터 데이터를 입력받을 수 있는 웹 폼을 생성합니다. 내부 도구(Internal Tools) 구축이나 간단한 설문 조사, 데이터 수집에 강력한 기능을 제공합니다.

핵심 용도 및 아키텍처

휴먼-인-더-루프(Human-in-the-Loop) 시작점: 자동화 프로세스에 필요한 초기 파라미터를 사람이 직접 입력해야 할 때 사용됩니다.

파일 업로드 및 처리: 사용자가 업로드한 파일을 즉시 워크플로우 내에서 처리(OCR, 저장, 변환)할 수 있습니다.

주요 옵션 설정 및 고급 구성

- **Form Fields (폼 필드):** 텍스트, 숫자, 날짜, 파일 업로드 등 다양한 입력 타입을 드래그 앤 드롭으로 구성할 수 있습니다. 각 필드는 필수 여부와 기본값을 설정할 수 있습니다.
- **Custom Form Styling (커스텀 스타일링):** CSS를 직접 입력하여 폼의 디자인을 기업의 브랜딩에 맞게 수정할 수 있습니다.
- **Form Ending (폼 제출 후 동작):**
 - **Show Message:** 완료 메시지를 표시합니다.
 - **Redirect to URL:** 제출 후 특정 웹페이지로 이동시킵니다.
 - **Return Binary File:** 워크플로우 처리 결과로 생성된 파일(예: PDF 리포트)을 사용자가 즉시 다운로드하게 할 수 있습니다.¹³
- **Test vs Production URL:** 개발 중에는 Test URL을 사용하여 에디터 내에서 실시간으로 데이터를 확인하고, 배포 시에는 Production URL을 사용하여 안정적인 서비스를 제공합니다.

2.4 Google Sheets Trigger (구글 시트 트리거) - 데이터 변경 감지

웹훅을 지원하지 않는 구글 시트의 변경 사항(행 추가, 수정)을 감지하기 위해 폴링 방식을 사용하는 트리거입니다.

핵심 용도 및 아키텍처

로우 코드 데이터베이스 연동: 구글 시트를 간이 DB로 사용하는 조직에서 데이터가 입력될 때마다 후속 작업을 자동화하기 위해 사용됩니다.

상태 관리: n8n은 내부적으로 마지막으로 처리한 행의 정보를 저장하여, 중복 처리를 방지합니다.

주요 옵션 설정 및 고급 구성

The screenshot shows the 'Google Sheets Trigger' configuration panel. It includes tabs for 'Parameters' (selected), 'Settings', and 'Docs'. Key sections include: 'Credential to connect with' with a 'Select Credential' dropdown; 'Poll Times' with a 'Mode' dropdown set to 'Fixed' and a value of 'Every Minute'; 'Document' and 'Sheet' sections, each with 'From list' and 'Choose...' options; 'Trigger On' set to 'Row Added'; and 'Options' with 'No properties' and an 'Add option' button.

- **Poll Interval (폴링 간격):** 얼마나 자주 시트를 확인할지 설정합니다. 너무 짧은 간격(예: 1분)은 구글 API 호출 한도(Quota)를 초과할 위험이 있으므로 주의해야 합니다.
- **Event (이벤트):** 'Row Added' 또는 'Row Updated'를 선택할 수 있습니다. 'Updated'의 경우 특정 열의 변경만 감지하도록 설정하여 불필요한 실행을 줄일 수 있습니다.

2.5 Slack Trigger (슬랙 트리거) - ChatOps 구현

Slack Trigger는 팀 커뮤니케이션 도구인 슬랙에서 발생하는 이벤트(메시지, 멘션, 슬래시 커맨드 등)를 감지합니다.

핵심 용도 및 아키텍처

ChatOps 봇 구축: 슬랙 내에서 /deploy와 같은 명령어를 입력하면 n8n이 이를 수신하여 배포 파이프라인

인을 실행하는 등의 인터랙티브 봇을 구현할 수 있습니다.

이벤트 구독: 특정 채널의 대화를 모니터링하여 감성 분석을 하거나, 특정 키워드가 포함된 메시지를 아카이빙하는 용도로 사용됩니다.

주요 옵션 설정 및 고급 구성

- **Authentication (인증):** Slack App을 생성하고 Bot Token을 발급받아야 합니다. 특히 'Socket Mode'를 사용하지 않는 경우, 슬랙 앱 설정 페이지에서 'Request URL'에 n8n의 웹훅 주소를 등록하고 검증(Challenge) 과정을 거쳐야 합니다.
- **Scopes (권한 범위):** channels:history, groups:history, app_mentions:read 등 필요한 권한을 정확히 부여해야 작동합니다.
- **Trigger On (트리거 조건):** 'On Mention'(봇 멘션 시), 'On Message Post'(모든 메시지 수신) 등 실행 조건을 세밀하게 설정하여 노이즈를 줄여야 합니다.

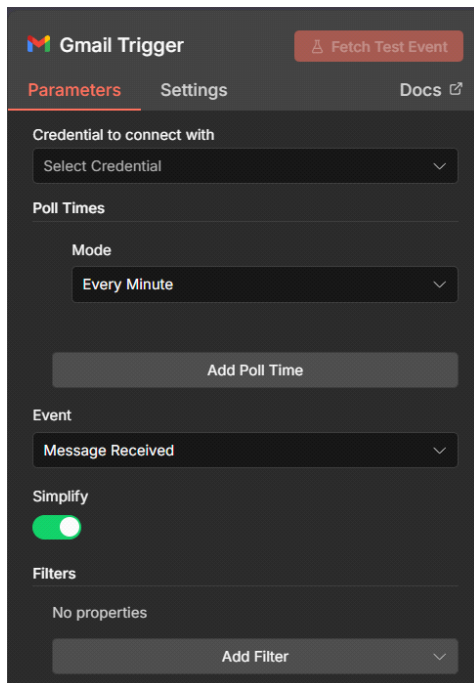
2.6 Email Trigger (이메일 트리거) - IMAP 및 Gmail

이메일 수신을 트리거로 사용하여 첨부파일 처리, 자동 응답, 티켓 생성 등을 수행합니다. Gmail 전용 트리거와 범용 IMAP 트리거로 나뉩니다.

핵심 용도 및 아키텍처

자동화된 인박스 처리: 송장(Invoice)이 첨부된 이메일을 자동으로 인식하여 첨부파일을 드라이브에 저장하고 내용을 OCR 처리하는 등의 워크플로우에 사용됩니다.

주요 옵션 설정 및 고급 구성



- **Protocol (프로토콜):** Gmail 사용자는 OAuth2 인증을 지원하는 Gmail Trigger를 사용하는 것이 보안상 유리합니다. 일반 IMAP은 App Password 설정이 필요할 수 있습니다.
- **Polling Strategy:** 이메일 역시 폴링 방식으로 작동하므로, 'Unread Only'(읽지 않은 메일만 처리) 옵션을 켜고, 처리 후 'Mark as Read'(읽음 처리) 액션을 연결하여 중복 처리를 방지하는 것이 표준 패턴입니다.

2.7 Execute Workflow Trigger (워크플로우 실행 트리거) - 모듈화

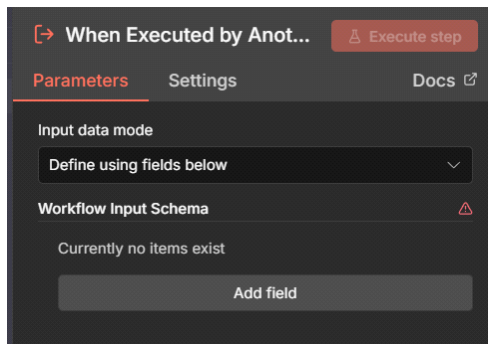
다른 워크플로우에 의해 호출될 때 실행되는 트리거입니다. 복잡한 로직을 재사용 가능한 모듈 (Sub-workflow)로 분리할 때 필수적입니다.

핵심 용도 및 아키텍처

코드 재사용성 향상: 여러 워크플로우에서 공통적으로 사용하는 로직(예: 슬랙 알림 포매팅, 에러 로깅)을 하나의 서브 워크플로우로 만들어 관리 효율을 높입니다.

동기식 처리: 호출한 부모 워크플로우에게 처리 결과를 반환하여 함수(Function)처럼 동작하게 할 수 있습니다.

주요 옵션 설정



- **Input Data Mode:**
- **Accept all data:** 부모 워크플로우가 보내는 모든 데이터를 그대로 수신합니다. 유연하지만 데이터 구조 검증이 필요합니다.
- **Define using fields below:** 수신할 데이터의 스키마(이름, 타입)를 명시적으로 정의하여 타입 안정성을 확보합니다.

3. 액션(Action) 노드: 외부 서비스와의 상호작용

액션 노드는 트리거와 유틸리티 노드를 거쳐 가공된 데이터를 외부 서비스에 생성(Create), 조회(Read), 수정(Update), 삭제>Delete)하는 역할을 수행합니다.

3.1 HTTP Request (HTTP 요청) - 만능 연결 도구

전용 노드가 없는 모든 외부 API와 통신할 수 있는 가장 강력하고 빈번하게 사용되는 노드입니다.

핵심 용도 및 아키텍처

REST/GraphQL API 연동: 전 세계의 거의 모든 웹 서비스와 데이터를 주고받을 수 있습니다.
파일 다운로드 및 업로드: 바이너리 데이터를 받아오거나 외부 서버로 전송하는 데 사용됩니다.

주요 옵션 설정 및 고급 구성

- **Authentication (인증):**
- **Predefined Credential Type:** n8n에 내장된 인증 프리셋을 사용합니다.
- **Generic Credential Type:** Header Auth, Query Auth 등을 직접 구성하여 커스텀 내부 API나 지원되지 않는 서비스에 연결합니다.
- **Pagination (페이징네이션):** 대량의 데이터를 가져올 때, API의 페이징네이션 규칙(Next URL, Offset/Limit)을 설정하여 자동으로 모든 데이터를 순회하며 수집할 수 있습니다. 이는 루프 노드를 수동으로 구성하는 것보다 훨씬 효율적입니다.
- **Error Handling (에러 처리):** 'Never Error' 옵션을 통해 4xx, 5xx 응답이 오더라도 워크플로우를 중단하지 않고 에러 분기 처리를 할 수 있도록 설정하는 것이 운영상 중요합니다.

3.2 Google Sheets (구글 시트) - 데이터 기록 및 관리

가장 대중적인 데이터 저장소인 구글 시트에 데이터를 쓰고 읽습니다.

핵심 용도

간이 데이터베이스: CRM 데이터 백업, 리드 수집, 업무 로그 기록 등에 사용됩니다.

주요 옵션 설정 및 고급 구성

- Operation (작업):
- Append: 데이터를 마지막 행에 추가합니다.
- Upsert (Append or Update): 특정 키(예: 이메일)를 기준으로 데이터가 존재하면 수정하고, 없으면 추가합니다. 중복 데이터 방지에 필수적입니다.
- Get Rows (Lookup): 특정 조건에 맞는 행을 조회합니다. 대량 데이터 조회 시 쿼리 속도 최적화를 위해 필요한 컬럼만 가져오도록 설정해야 합니다.
- Data Mapping: 'Map Automatically'를 사용하면 들어오는 JSON 키와 시트의 헤더가 일치할 경우 자동으로 매핑해주어 설정 시간을 단축시킵니다.

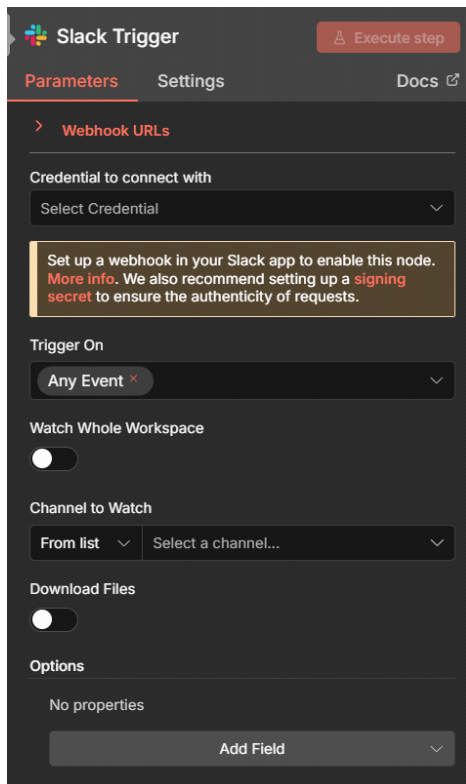
3.3 Slack (슬랙) - 알림 및 메시징

팀 협업을 위한 메시지 발송, 채널 관리 등을 수행합니다.

핵심 용도

실시간 알림: 에러 발생, 중요한 리드 유입, 결제 요청 등을 슬랙으로 즉시 전송합니다.

주요 옵션 설정 및 고급 구성



- **Block Kit:** 단순 텍스트를 넘어 버튼, 이미지, 입력 필드 등이 포함된 풍부한 UI의 메시지를 JSON 형태로 구성하여 전송할 수 있습니다.
- **Dynamic Channel Routing:** 메시지를 보낼 채널 ID를 고정하지 않고, 이전 노드의 데이터(예: 담당 부서)에 따라 동적으로 변경하여 적절한 채널로 라우팅할 수 있습니다.
- **Send and Wait for Response:** 메시지에 버튼을 추가하고, 사용자가 버튼을 클릭할 때까지 워크플로우를 대기시켰다가 클릭 결과에 따라 후속 작업을 진행하는 인터랙티브 워크플로우를 구현할 수 있습니다.

3.4 Airtable (에어테이블) - 구조화된 데이터 관리

구글 시트보다 더 엄격한 스키마를 가진 관계형 데이터베이스 서비스와 연동합니다.

핵심 용도

구조화된 DB 작업: 프로젝트 관리, 재고 관리 등 관계형 데이터 구조가 필요한 작업에 적합합니다.

주요 옵션 설정 및 고급 구성

- **Smart Upsert:** 데이터를 추가하거나 업데이트할 때, 중복을 방지하기 위해 사용됩니다. 다만 에어테이블 API 특성상 타입(Type)이 일치하지 않으면 에러가 발생하므로, 사전에 Edit Fields 노드를 통해 데이터 타입을 정확히 변환(예: 문자열 -> 숫자)해줘야 합니다.
- **List Records:** 필터 수식(Formula)을 사용하여 필요한 레코드만 효율적으로 가져오는 것이 API 호출 제한을 피하는 핵심입니다.

3.5 Postgres (포스트그레스) - 엔터프라이즈 DB 연동

SQL 기반의 강력한 데이터베이스 작업을 수행합니다.

핵심 용도

백엔드 로직 수행: 복잡한 조인(Join) 연산이나 대량 데이터 처리를 DB 레벨에서 수행합니다.

주요 옵션 설정 및 고급 구성

- **Query Parameters (쿼리 파라미터):** SQL 인젝션 공격을 방지하기 위해 \$1, \$2와 같은 파라미터 바인딩을 반드시 사용해야 합니다.
- **Transaction (트랜잭션):** 여러 개의 쿼리를 하나의 트랜잭션으로 묶어, 중간에 오류가 발생하면 모든 변경 사항을 롤백(Rollback)하여 데이터 무결성을 보장할 수 있습니다. 이는 금융 데이터 처리 등에서 필수적인 옵션입니다.

3.6 Gmail (지메일) - 이메일 발송

구글의 네이티브 API를 사용하여 이메일을 발송합니다.

주요 옵션 설정

- **Attachments:** 바이너리 데이터를 매핑하여 파일 첨부 메일을 발송할 수 있습니다.
- **HTML Body:** HTML 태그를 사용하여 디자인된 이메일을 발송할 수 있습니다.
- **Threading:** Thread ID를 지정하여 기존 이메일 대화타래(Thread)에 답장 형태로 메일을 보낼 수 있어 문맥 유지에 유용합니다.

3.7 Telegram (텔레그램) - 챗봇 메시징

개인화된 알림이나 챗봇 응답을 보냅니다.

주요 옵션 설정

- **Chat ID:** 메시지를 받을 대상(사용자 또는 그룹)의 ID를 정확히 지정해야 합니다.

- **Message Types:** 텍스트뿐만 아니라 사진, 비디오, 문서, 위치 정보 등을 전송할 수 있습니다.
- **Keyboard Markup:** 사용자에게 선택지 버튼(Inline Keyboard)을 제공하여 상호작용을 유도할 수 있습니다.

3.8 Notion (노션) - 지식 관리 및 문서화

노션의 데이터베이스에 페이지를 생성하거나 블록을 추가합니다.

주요 옵션 설정

- **Database ID:** 데이터를 추가할 데이터베이스를 선택합니다.
- **Properties Mapping:** 노션의 다양한 속성(Title, Select, Multi-select, Date, Person)에 맞춰 JSON 데이터를 정확히 매핑해야 에러가 발생하지 않습니다. 특히 Select나 Relation 타입은 고유 ID를 요구하는 경우가 많아 주의가 필요합니다.

3.9 OpenAI (OpenAI) - 생성형 AI 액션

AI Agent가 아닌 단독 액션 노드로 사용하여 텍스트 요약, 번역, 감성 분석 등을 수행합니다.

주요 옵션 설정

- **Model:** gpt-4o, gpt-3.5-turbo 등 작업의 난이도와 비용에 맞는 모델을 선택합니다.
- **System Message:** AI에게 역할을 부여하여 일관된 출력을 유도합니다.
- **Response Format:** JSON Object로 설정하여 후속 노드에서 파싱하기 쉬운 정형 데이터를 받도록 강제할 수 있습니다.

3.10 MySQL/MariaDB - 레거시 DB 연동

Postgres와 유사하게 SQL 쿼리를 실행합니다.

주요 옵션 설정

- **Execute Query:** 표준 SQL 문법을 지원합니다.
- **Connection Pooling:** n8n은 연결을 유지하거나 매번 새로 맺을 수 있는 옵션을 제공하는데, 대량의 트래픽 처리 시 DB 부하를 고려하여 타임아웃 설정을 적절히 조절해야 합니다.

4. 유틸리티(Utility) 노드: 데이터 가공 및 제어 로직

유틸리티 노드는 외부와 통신하지 않고, 워크플로우 내부에서 데이터의 흐름을 제어(Flow Control)하거나 형태를 변환(Transformation)하는 핵심 '로직' 블록입니다.

4.1 Edit Fields (필드 편집, 구 Set) - 데이터 구조 변환

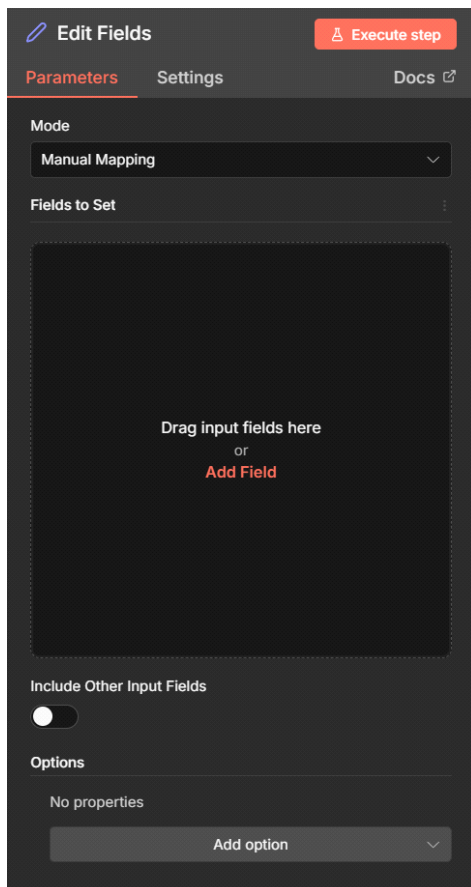
가장 기본적이면서도 중요한 노드로, 데이터의 키-값(Key-Value)을 생성, 수정, 삭제합니다.

핵심 용도 및 아키텍처

변수 선언: 워크플로우 내에서 사용할 상수나 변수를 정의합니다.

데이터 정제: API로 보내기 전 불필요한 필드를 제거하거나, 이름을 변경합니다.

주요 옵션 설정 및 고급 구성



- **Assignment Method:**
- **Manual Mapping:** UI에서 직접 값을 입력하거나 드래그 앤 드롭으로 매핑합니다.

- **JSON Mode:** 복잡한 중첩 객체(Nested Object)나 배열을 생성할 때 직접 JSON 코드를 작성합니다. 고급 사용자는 이 모드를 선호합니다.
- **Include Other Input Fields** (다른 입력 필드 포함):
- **True (기본값):** 기존 데이터에 새로운 필드를 '추가'합니다.
- **False:** 입력 데이터를 모두 버리고, 이 노드에서 정의한 필드만 남깁니다. 보안을 위해 민감한 정보(PII 등)를 제거하고 필요한 데이터만 다음 노드로 넘길 때 유용합니다.

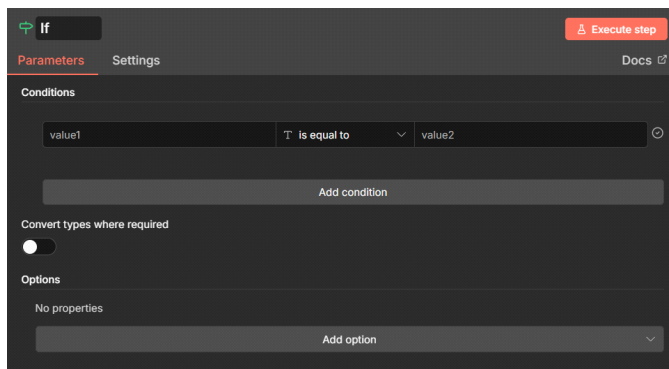
4.2 If (조건 분기) - 불리언 로직

참(True)과 거짓(False) 조건에 따라 실행 경로를 나눕니다.

핵심 용도

조건부 실행: "결제 금액이 100달러 이상이면(True) VIP 슬랙 채널에 알림, 아니면(False) 일반 채널 알림."

주요 옵션 설정 및 고급 구성



- **Conditions:** 문자열(String), 숫자(Number), 날짜(Date), 불리언(Boolean) 등 데이터 타입에 맞는 비교 연산자를 제공합니다.
- **Combine Logic:** AND, OR 조건을 조합하여 복잡한 로직을 구현할 수 있습니다.
- **실행 특성:** n8n의 If 노드는 전통적인 프로그래밍과 달리, 데이터 배치(Batch) 내에서 일부는 True, 일부는 False일 경우 두 경로가 모두 활성화될 수 있음을 이해해야 합니다.

4.3 Switch (스위치) - 다중 경로 라우팅

If 노드의 한계를 넘어, 3개 이상의 경로로 데이터를 분기합니다.

핵심 용도

- **카테고리별 라우팅:** "이메일 유형이 '청구서'면 A경로, '지원'이면 B경로, '기타'면 C경로."
- **가독성 향상:** 중첩된 If 노드(Nested ifs)를 하나의 Switch 노드로 대체하여 워크플로우를 평탄화하고 가독성을 높입니다.

4.4 Merge (병합) - 데이터 동기화 및 결합

갈라진 데이터 흐름을 다시 하나로 합치거나, 서로 다른 소스의 데이터를 결합합니다.

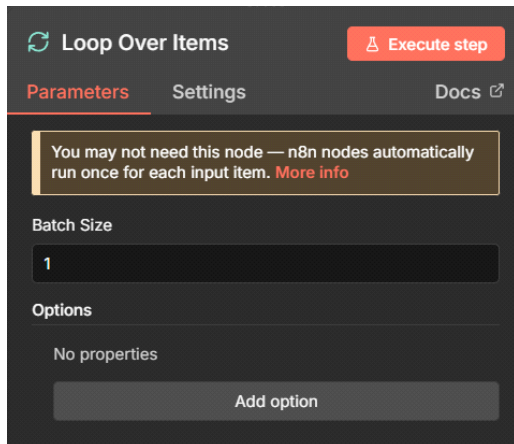
주요 옵션 설정 및 고급 구성 (Merge Modes)

- **Append (추가):** 두 입력의 데이터를 단순히 위아래로 이어 붙입니다. 리스트 병합에 사용됩니다.
- **Combine (Merge by Key):** SQL의 JOIN과 유사합니다. 공통된 키(예: ID)를 기준으로 두 데이터 소스를 가로로 병합하여 풍부한 데이터를 만듭니다.
- **Keep Matches:** 교집합(Inner Join).
- **Keep Non-Matches:** 차집합. 누락된 데이터를 찾을 때 유용합니다.
- **Wait (대기):** 두 입력 브랜치의 실행이 모두 끝날 때까지 기다린 후 실행합니다. 병렬 처리된 작업의 동기화를 맞출 때 필수적입니다.

4.5 Loop Over Items (아이템 반복, 구 Split In Batches) - 배치 처리 및 속도 제어

대량의 데이터를 지정된 크기(Batch Size)만큼 나누어 처리합니다.

핵심 용도 및 아키텍처

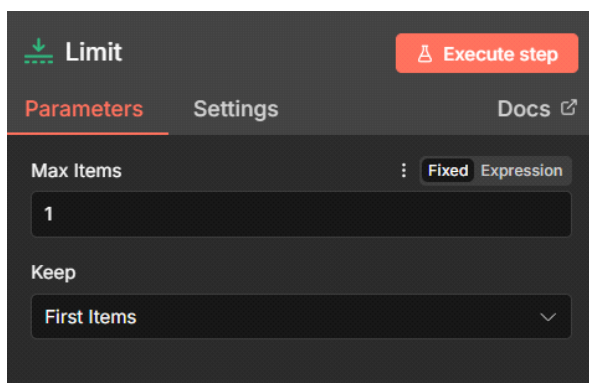


- **API 레이트 리밋(Rate Limit) 준수:** "1초에 5회 요청 제한"이 있는 API를 호출할 때, 배치 사이즈를 1~5로 설정하고 루프 내에 Wait 노드를 두어 속도를 조절합니다.
- **메모리 관리:** 수천 개의 아이템을 한 번에 처리하면 메모리 오버플로우가 발생할 수 있습니다. 이를 작은 배치로 나누어 처리하면 안정성을 확보할 수 있습니다.
- **Context:** `$input.context.noItemsLeft` 변수를 사용하여 루프의 종료 시점을 판단합니다.

4.6 Limit (리미트)

JSON 배열과 개별 아이템 간의 변환을 수행합니다.

주요 옵션 설정

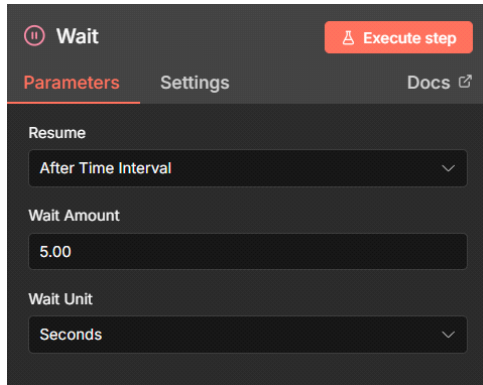


- **Split Out:** 하나의 아이템 안에 있는 배열(Array) 필드를 여러 개의 개별 아이템으로 분리(Explode)합니다. 예를 들어, 주문서(1개) 안에 있는 상품 목록(5개)을 5개의 개별 아이템으로 나누어 처리할 때 사용합니다.
- **Aggregate:** 반대로 여러 개의 아이템을 하나의 배열로 묶습니다. 루프 처리 후 결과를 하나의 리포트로 합쳐서 이메일을 보낼 때 사용됩니다.

4.7 Wait (대기) - 실행 지연 및 일시 정지

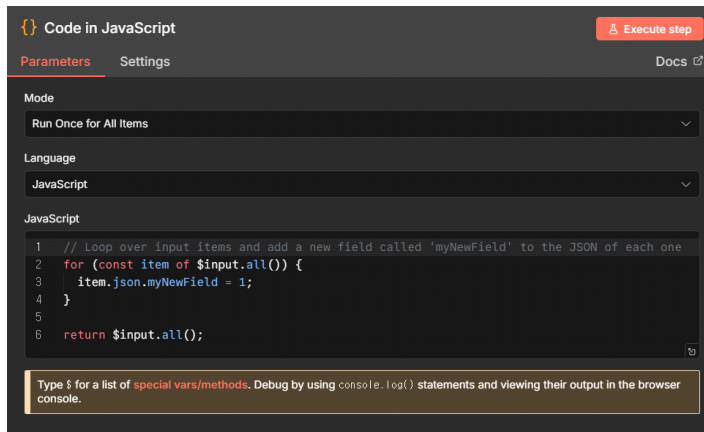
워크플로우 실행을 일정 시간 멈추거나, 외부 신호를 기다립니다.

주요 옵션 설정 및 고급 구성



- **Time Interval:** "1시간 대기"와 같이 단순 지연을 설정합니다.
- **On Webhook Call:** 워크플로우를 일시 정지시키고, 외부에서 특정 URL로 호출이 올 때까지 기다립니다. 이는 승인 시스템(Approval Workflow) 구축의 핵심 패턴입니다.
- **Persistence:** 대기 시간이 길어지면(기본 65초 이상), n8n은 실행 상태를 데이터베이스에 저장하고 메모리에서 내립니다. 이를 위해 SQLite보다는 Postgres와 같은 외부 DB를 사용하는 것이 안정적입니다.

4.8 Code (코드) - 무한한 확장성



자바스크립트(Node.js)나 파이썬을 사용하여 노드로 구현하기 힘든 복잡한 로직을 작성합니다.59 (자세한 내용은 커스터마이징 섹션 참조)

4.9 Crypto (암호화) - 보안 데이터 처리

데이터의 해싱(MD5, SHA), 암호화/복호화(AES), HMAC 서명 생성 등을 수행합니다. API 인증 시 서명(Signature)을 생성해야 할 때 필수적입니다.

4.10 Date & Time (날짜 및 시간) - 시간 계산

날짜 포맷 변경, 시간 더하기/빼기, 두 날짜 간의 차이 계산 등을 수행합니다. 타임존 변환 기능이 강력하여 글로벌 서비스 연동 시 유용합니다.

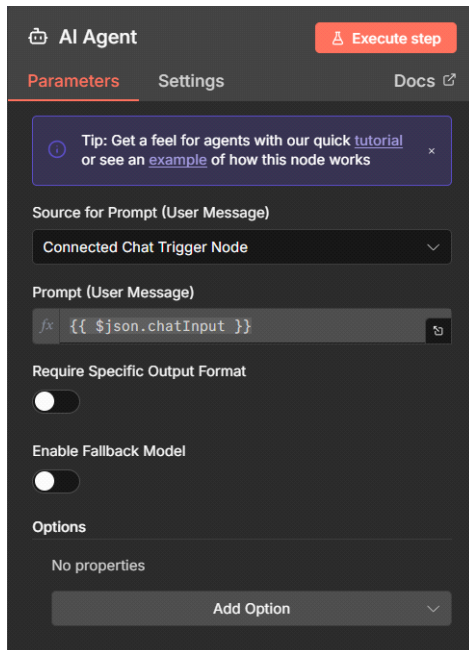
5. AI 및 Agent 노드: 지능형 자동화의 새로운 계층

n8n은 LangChain을 네이티브로 통합하여, 단순 자동화를 넘어선 '추론(Reasoning)' 능력을 갖춘 에이전트 구축을 지원합니다.

5.1 AI Agent (AI 에이전트) - 두뇌

워크플로우의 '두뇌' 역할을 하는 루트 노드입니다. LLM, 메모리, 도구(Tools)를 조율합니다.

주요 옵션 설정 및 고급 구성



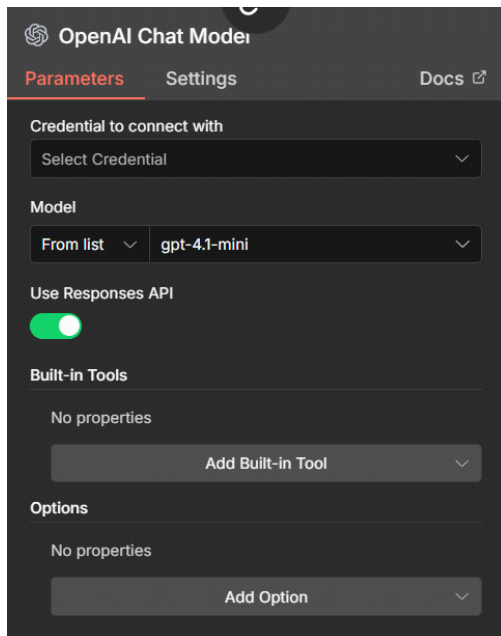
The image shows the configuration panel for an AI Agent. At the top, there's a title 'AI Agent' and an 'Execute step' button. Below the title are tabs for 'Parameters', 'Settings', and 'Docs'. A tip box suggests getting a feel for agents with a quick tutorial or seeing an example. The 'Parameters' tab is active, showing a 'Source for Prompt (User Message)' dropdown set to 'Connected Chat Trigger Node'. Below this is a 'Prompt (User Message)' text area containing the code `fx {{ $json.chatInput }}`. There are two toggle switches: 'Require Specific Output Format' and 'Enable Fallback Model', both currently turned off. At the bottom, there's an 'Options' section with 'No properties' listed and an 'Add Option' button.

- Agent Type:
- **Conversational Agent:** 대화형 챗봇. 도구 사용보다는 대화 맥락 유지에 최적화됨.
- **Tools Agent:** 'ReAct(Reasoning and Acting)' 패턴을 사용하여 사용자의 질문을 해결하기 위해 어떤 도구를 사용할지 스스로 판단하고 실행합니다. 가장 강력한 형태입니다.
- **System Message:** 에이전트의 페르소나와 제약 조건을 정의합니다. (예: "당신은 친절한 고객 지원 봇입니다. 기술적인 문제는 기술 지원 도구를 호출하세요.").

5.2 Chat Model (챗 모델) - 지능 엔진

OpenAI, Anthropic, Mistral 등의 LLM 모델을 연결합니다.

주요 옵션 설정

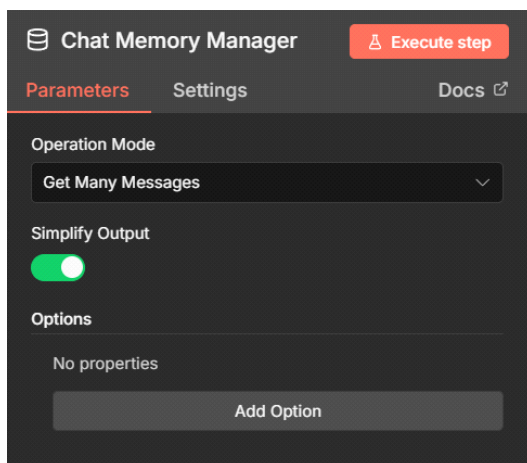


- **Temperature:** 창의성 조절. 0에 가까우면 사실적이고 결정론적인 답변, 1에 가까우면 창의적인 답변을 생성합니다. 데이터 추출 작업에는 0~0.2가 권장됩니다.
- **JSON Mode:** 출력을 반드시 JSON 형태로 강제하여, 후속 노드에서 프로그래밍적으로 처리할 수 있게 합니다.

5.3 Memory (메모리) - 맥락 유지

에이전트가 이전 대화 내용을 기억하게 합니다.

주요 옵션 설정



- **Window Buffer Memory:** 최근 N개의 대화(슬라이딩 윈도우)만 기억하여 토큰 비용을 절약하고 컨텍스트 초과를 방지합니다.
- **Session Key:** 사용자별로 고유한 세션 키(예: 텔레그램 Chat ID)를 할당하여 다중 사용자와의 대화를 구분합니다.

5.4 Vector Store (벡터 스토어) - 장기 기억 및 RAG

Pinecone, Qdrant 등과 연동하여 대량의 문서를 벡터화하여 저장하고 검색합니다.

핵심 용도

The screenshot shows the 'Simple Vector Store' configuration interface. It has tabs for 'Parameters', 'Settings', and 'Docs'. Under 'Parameters', there is a tip box about n8n templates. Below that, 'Operation Mode' is set to 'Get Many'. 'Memory Key' is set to 'vector_store_key' with a dropdown menu. There is a 'Prompt' text input field. 'Limit' is set to '4'. 'Include Metadata' is a toggle switch that is turned on. 'Rerank Results' is a toggle switch that is turned off. At the bottom, there is a warning box stating: 'For experimental use only: Data is stored in memory and will be lost if n8n restarts. Data may also be cleared if available memory gets low, and is accessible to all users of this instance. More info'.

- RAG (Retrieval-Augmented Generation): 기업 내부 문서를 검색하여 시가 그 내용을 바탕으로 답변하게 합니다.

5.5 Embeddings (임베딩) - 텍스트 벡터화

텍스트를 기계가 이해할 수 있는 숫자 벡터로 변환합니다. OpenAI Embeddings (text-embedding-3-small) 등이 사용됩니다.

5.6 Text Splitter (텍스트 분할기) - 청킹(Chunking)

큰 문서를 벡터화하기 전에 작은 조각으로 나눕니다.

주요 옵션 설정

Recursive Character Text Splitter: 문단 -> 문장 -> 단어 순으로 의미 단위로 쪼개는 가장 일반적인 방식입니다.

Chunk Overlap: 문맥이 끊기는 것을 방지하기 위해 조각 간에 일정 부분(예: 200자)을 중복시켜 자릅니다.

5.7 Tools (도구) - 에이전트의 손과 발

에이전트가 사용할 수 있는 기능을 정의합니다.

Custom Code Tool / Call n8n Workflow Tool: 에이전트가 특정 작업을 수행하기 위해 파이썬 코드를 실행하거나 다른 워크플로우를 호출하게 합니다.

Description의 중요성: 도구의 이름과 설명(Description)은 AI에게 전달되는 '프롬프트'입니다. "주식 정보를 가져올 때 사용하세요"와 같이 명확하게 작성해야 에이전트가 적절한 상황에 도구를 호출합니다.

5.8 Calculator (계산기) - 수학 연산

LLM은 수학 연산에 취약하므로, 정확한 계산이 필요할 때 에이전트가 호출하여 사용하는 도구입니다.

5.9 Wikipedia / Search Tool - 정보 검색

외부 지식이 필요할 때 위키피디아나 구글 검색을 수행하도록 연결하는 도구입니다.

5.10 Structured Output Parser - 출력 정형화

AI의 응답을 특정 스키마(JSON)에 맞게 변환하여 시스템 안정성을 높이는 노드입니다.

6. 커스터마이징(Customization) 노드: 한계 없는 확장

6.1 Code Node (JS/Python) - 만능 해결사

n8n 1.0 버전부터 도입된 Code 노드는 데이터 처리 방식을 혁신적으로 개선했습니다.

문법 변화 및 모드

Run Once for All Items: 모든 입력 아이템을 하나의 배열로 받아서 한 번 실행합니다. 데이터 집계(Aggregation)나 전체 데이터셋 비교에 사용됩니다.

문법: `const items = $input.all();`

Run Once for Each Item: 각 아이템마다 코드가 개별적으로 실행됩니다. 단순 데이터 변환(매핑, 정규식 처리)에 유리합니다.

문법: `const item = $input.item;` 또는 `$json.fieldName`.

고급 활용

External Libraries: 셸프 호스팅 버전의 경우, 환경 변수 설정을 통해 npm 모듈을 임포트하여 사용할 수 있습니다. (moment, lodash 등).

Python Support: Pyodide 기반의 파이썬 실행을 지원하여, 데이터 분석 라이브러리(pandas 등 제한적)를 활용할 수 있습니다.

6.2 Execute Command (명령어 실행) - OS 레벨 제어

호스트 운영체제의 셸 명령어를 실행합니다.

핵심 용도 및 보안 경고

로컬 유틸리티 사용: ffmpeg로 동영상을 변환하거나, curl로 복잡한 인증이 필요한 레거시 시스템에 접속할 때 사용합니다.

보안 위험: 이 노드는 호스트 시스템에 대한 임의 명령 실행 권한을 부여하므로, 보안상 매우 위험할 수 있습니다. 클라우드 버전에서는 사용할 수 없으며, 셀프 호스팅 시에도 신뢰할 수 있는 관리자만 접근하도록 제한해야 합니다.

7. 결론 및 모범 사례 (Best Practices)

n8n을 이용한 엔터프라이즈급 자동화 시스템 구축을 위해서는 단순한 노드 연결을 넘어선 아키텍처 설계가 필요합니다.

표 1: 핵심 노드 선택 전략 요약

요구 사항	추천 노드	핵심 구성 전략
외부 시스템에서 실시간 트리거	Webhook	POST 메서드 사용, Header Auth 보안 설정, 비동기 처리를 위한 'Immediate Response' 고려
주기적 배치 작업	Schedule	Cron 표현식 활용, 타임존(Timezone) 설정 확인, 동시성(Concurrency) 제어
복잡한 데이터 변환	Edit Fields / Code	간단한 매핑은 Edit Fields, 복잡한 로직은 Code 노드의 Run Once for Each Item 모드 사용
API 레이트 리밋 제어	Loop Over Items	배치 사이즈 조절, 루프 내 Wait 노드 삽입, \$input.context.noItemsLeft로 종료 조건 관리
데이터 병합 및 조인	Merge	Combine by Key 모드를 사용하여 SQL Join과 같은 데이터 결합 구현
지능형 판단 및 도구 사용	AI Agent (Tools)	Tools Agent 사용, 도구(Tool) 설명(Description)을 명확한 프롬프트로 작성
대기 및 승인 프로세스	Wait	On Webhook Call 모드 사용하여 외부 승인 대기, 장기 대기 시 외부 DB(Postgres) 사용 권장

n8n은 노드 하나하나의 깊이 있는 기능을 이해하고 조합할 때 비로소 그 진가가 발휘됩니다. 특히 **데이터 링킹(Data Linking)**을 통한 전역 변수 접근, **루프(Loop)**와 **배치(Batch)**를 이용한 대용량 처리 안정성 확보, 그리고 AI Agent를 통한 비결정적(Probabilistic) 문제 해결 능력의 결합은 기존 RPA나 자동화 툴이 도달하지 못한 영역을 개척하고 있습니다. 엔지니어는 이러한 도구들을 적재적소에 배치하여, 견고하고 확장 가능한 자동화 파이프라인을 설계해야 합니다.