Cristian Rojas
cristianjrojas@gmail.com

Lead Scoring & Notification

An end-to-end n8n automation that ensures every lead is captured, qualified, stored, and actioned.

Key Capabilities

- **Flexible Lead Intake**
  - Live webhook endpoint (/lead-submit) for real form submissions
  - Cron-driven simulator for QA that generates both valid and error-case leads
- **Robust Validation**
  - Enforces presence and format of full_name, email, budget, interest_level
  - Routes any malformed payloads to a safe error branch
- **Advanced Scoring**
  - Budget tiers (5–40 pts), interest (5–30 pts), company size (0–20 pts)
  - +5 pt bonuses for business-domain emails and provided phone numbers
  - Produces a 0–100 score with clear Cold/Warm/Hot thresholds
- **Automatic Categorization**
  - **Hot** (≥ 70), **Warm** (50–69), **Cold** (< 50)
- **Fail-Safe Data Retention**
  - Upserts scored leads into Airtable
  - Backs up the raw JSON payload in a separate table before processing
- **Smart Notifications**
  - Direct-message to the assigned rep for every **Hot** lead, including an Airtable deep link
  - 2-minute follow-up reminder to keep high-value leads top of mind
- **Comprehensive Error Handling**
  - Archives any validation or runtime errors (with full payload and context) in Airtable
- Sends real-time alerts to your Slack "#errors" channel

This workflow balances reliability (raw back-up, error logging), flexibility (simulated vs. real data), and speed (instant scoring + notifications), so your team never misses an opportunity.

Prerequisites

- **n8n** version ≥ 1.95.2 (Self-hosted or cloud)
- **Airtable** API key or Personal Access Token (PAT)
- **Slack**: Incoming webhook or Bot token with chat permissions
- Node access to a scheduler (for simulated data) and webhooks (for production form)

Installation

Cristian Rojas
cristianjrojas@gmail.com

1. **Clone** this repository:

```
git clone https://github.com/CristianRojas001/Lead-Scoring-and-Notification---
REMWaste/blob/main/Lead_Scoring_and_Notification.json
```

2. **Import** the workflow JSON into n8n:
   - Open the n8n Editor UI.
   - Click **Import**, paste the contents of Lead_Scoring_and_Notification.json.
3. **Set up Credentials**:
   - **Airtable**: Create a credential using your API key/PAT, select or enter your Base ID.
   - **Slack**: Create a Slack credential (Incoming Webhook or Bot Token + default channel or user).
4. **Adjust Configuration**:
   - In the **Lead Validation and Scoring** node, comment/uncomment data-source blocks to switch between:

- **TestingData** (simulated leads)
- **FromForm - Production** webhook (real form submissions at /lead-submit)
   - Toggle the **Schedule Trigger** node to enable or disable simulated leads.

5. **Activate** the workflow (toggle **Active** at the top right) once ready.

Configuration: Switching Between Testing & Production Data Sources

In the **Lead Validation and Scoring** Function node, you must toggle which data source to use by commenting/uncommenting the appropriate code blocks.

1. **For Testing**

Comment out the production block (line 7):

```
//const leadData = $('FromForm - Production').first().json.body.data;
```

Uncomment the testing line (line 4):

```
let leadData = $items("TestingData")[0].json;
```

2. **For Production**

o **Comment out** the testing line (line 4):

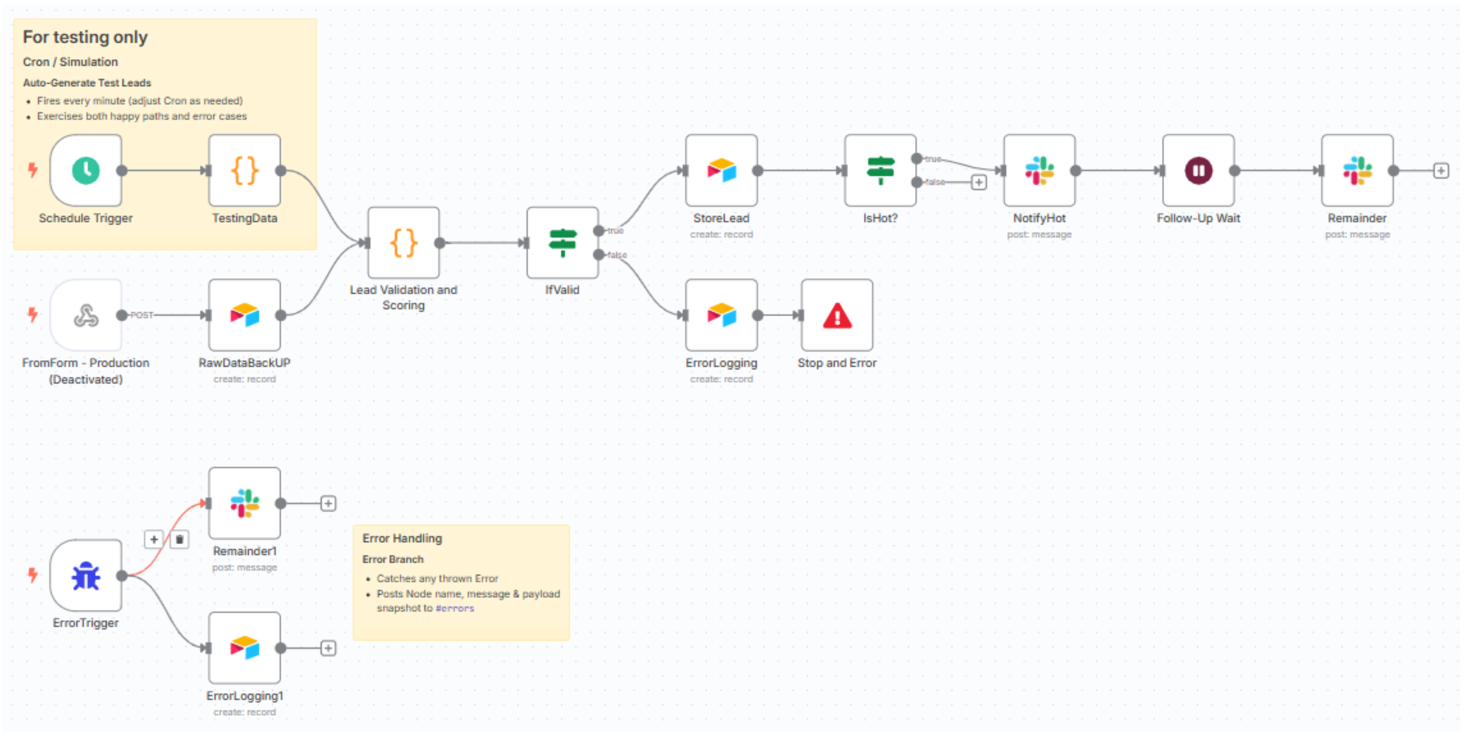
```
// let leadData = $items("TestingData")[0].json;
```

o **Uncomment** the production line 7 as shown above.

Cristian Rojas
cristianjrojas@gmail.com

Note: Always ensure only one data source is active—either TestingData or FromForm - Production—to avoid unexpected behavior.

Workflow Overview


For testing only
Cron / Simulation
Auto-Generate Test Leads
- Fires every minute (adjust Cron as needed)
- Exercises both happy paths and error cases

Error Handling
Error Branch
- Catches any thrown Error
- Posts Node name, message & payload snapshot to #errors

Node Details

TestingData (Function)

- **Purpose:** Generates fake lead records for development and QA.
- **Behavior:** On each trigger, emits one of eight predefined variants—five valid leads plus three crafted error/edge cases (e.g. missing email, malformed budget, wrong data types).
- **Benefit:** Allows you to exercise and validate every branch of the workflow (success, validation fail, runtime error) automatically, without manual form submissions.

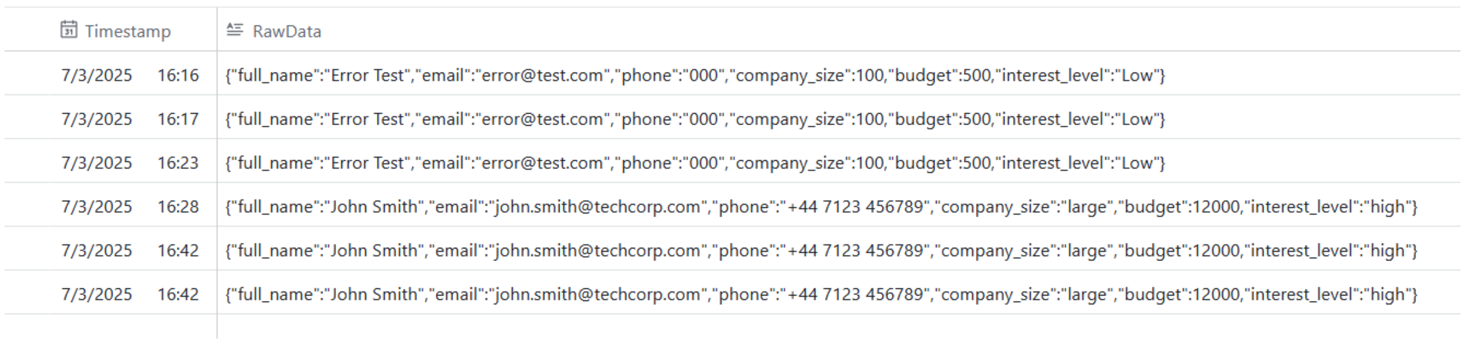
FromForm - Production (Webhook)

- **Purpose:** Receives live lead submissions from your marketing form.
- **Endpoint:** HTTP POST to `/lead-submit`.
- **Payload:** Expects JSON with fields `full_name`, `email`, `phone`, `company_size`, `budget`, `interest_level`.
- **Switching:** Disabled during testing by commenting out its code block in the Function node.

Cristian Rojas
cristianjrojas@gmail.com

RawDataBackUP (Airtable)

- **Purpose:** Immediately archives the incoming raw payload—before any processing—to guarantee no lead data is ever lost.
- **Schema:** Stores the full JSON string plus a generated timestamp.
- **Recovery:** If downstream steps fail, ops can retrieve the original lead details from this table and manually recover any potential deals.



Timestamp	RawData
7/3/2025 16:16	{"full_name":"Error Test","email":"error@test.com","phone":"000","company_size":100,"budget":500,"interest_level":"Low"}
7/3/2025 16:17	{"full_name":"Error Test","email":"error@test.com","phone":"000","company_size":100,"budget":500,"interest_level":"Low"}
7/3/2025 16:23	{"full_name":"Error Test","email":"error@test.com","phone":"000","company_size":100,"budget":500,"interest_level":"Low"}
7/3/2025 16:28	{"full_name":"John Smith","email":"john.smith@techcorp.com","phone":"+44 7123 456789","company_size":"large","budget":12000,"interest_level":"high"}
7/3/2025 16:42	{"full_name":"John Smith","email":"john.smith@techcorp.com","phone":"+44 7123 456789","company_size":"large","budget":12000,"interest_level":"high"}
7/3/2025 16:42	{"full_name":"John Smith","email":"john.smith@techcorp.com","phone":"+44 7123 456789","company_size":"large","budget":12000,"interest_level":"high"}

Lead Validation & Scoring (Function)

Overview

This n8n Code Node processes incoming lead data from web forms, validates required fields, calculates a lead score based on multiple criteria, and categorizes leads as Hot, Warm, or Cold for sales prioritization.

Required Fields

- full_name - Lead's full name
- email - Lead's email address
- budget - Lead's budget (numeric value)
- interest_level - Lead's interest level (high/medium/low)

Optional Fields

- phone - Lead's phone number
- company_size - Company size (enterprise/large/medium/small)

Data Source Configuration

The node supports two modes:

Testing Mode

```
let leadData = $items("TestingData")[0].json;
```

Cristian Rojas
cristianjrojas@gmail.com

Production Mode

```
let leadData;
if ($('FromForm - Production').first().json.body) {
  leadData = $('FromForm - Production').first().json.body;
} else if ($('FromForm - Production').first().json.data) {
  leadData = $('FromForm - Production').first().json.data;
} else {
  leadData = $('FromForm - Production').first().json.body;
}
```

Validation Process

The node validates that all required fields are present and not empty. If validation fails, it returns an error object with:

- validation_error: true
- missing_fields: [] - Array of missing field names
- message - Human-readable error message

Scoring Algorithm

The node calculates a lead score (0-100 points) based on multiple factors:

Budget Scoring (0-40 points)

- $10,000+: 40 points
- $5,000-$9,999: 25 points
- $1,000-$4,999: 15 points
- $500-$999: 10 points
- Under $500: 5 points

Interest Level Scoring (0-30 points)

- High: 30 points
- Medium: 20 points
- Low: 10 points
- Other/Unknown: 5 points

Company Size Scoring (0-20 points)

- Enterprise: 20 points
- Large: 15 points
- Medium: 10 points
- Small: 5 points
- Unknown: 0 points

Bonus Scoring (0-10 points)

- Business email domain (non-consumer): +5 points

Cristian Rojas
cristianjrojas@gmail.com

• Phone number provided: +5 points

Lead Categorization

Based on the total score:

• **Hot** (70+ points): High-priority leads requiring immediate attention
• **Warm** (50-69 points): Medium-priority leads for follow-up
• **Cold** (0-49 points): Low-priority leads for nurturing campaigns

Output Structure

The node returns an enriched lead object containing:

Original Data

• full_name, email, phone, company_size, budget, interest_level

Calculated Fields

• lead_id - Unique identifier (format: LEAD_timestamp_randomstring)
• lead_score - Calculated score (0-100)
• lead_category - Hot/Warm/Cold classification
• scored_at - ISO timestamp of scoring

Scoring Breakdown

• score_breakdown - Object containing individual scoring components for debugging

Validation Status

• validation_passed: true
• processed_at - ISO timestamp of processing

Error Handling

If validation fails, the node returns an error object instead of processed lead data. Subsequent nodes should check for validation_error: true before processing.

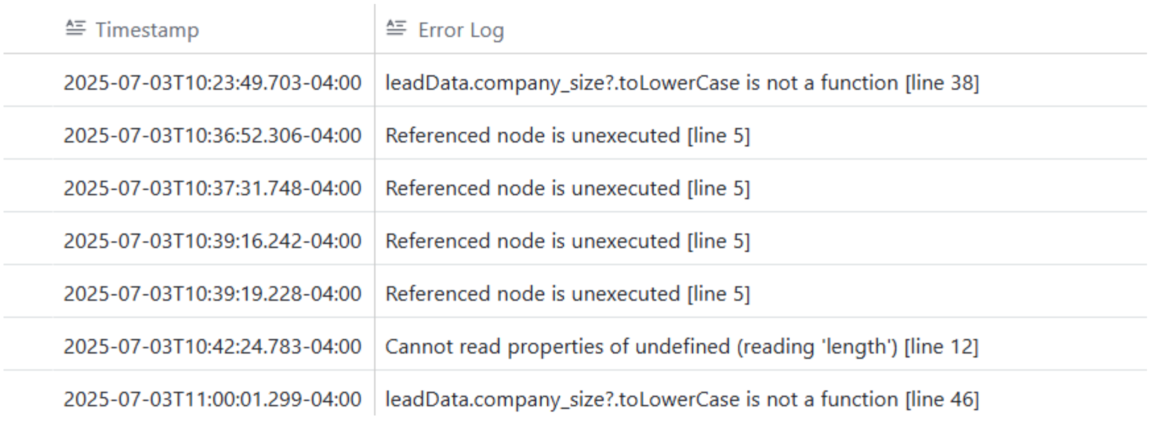
IfValid (If)

• **Condition:**

```
{{$json.validation_error !== true && $json.error === undefined}}
```

Routes only clean, error-free items as **True**.

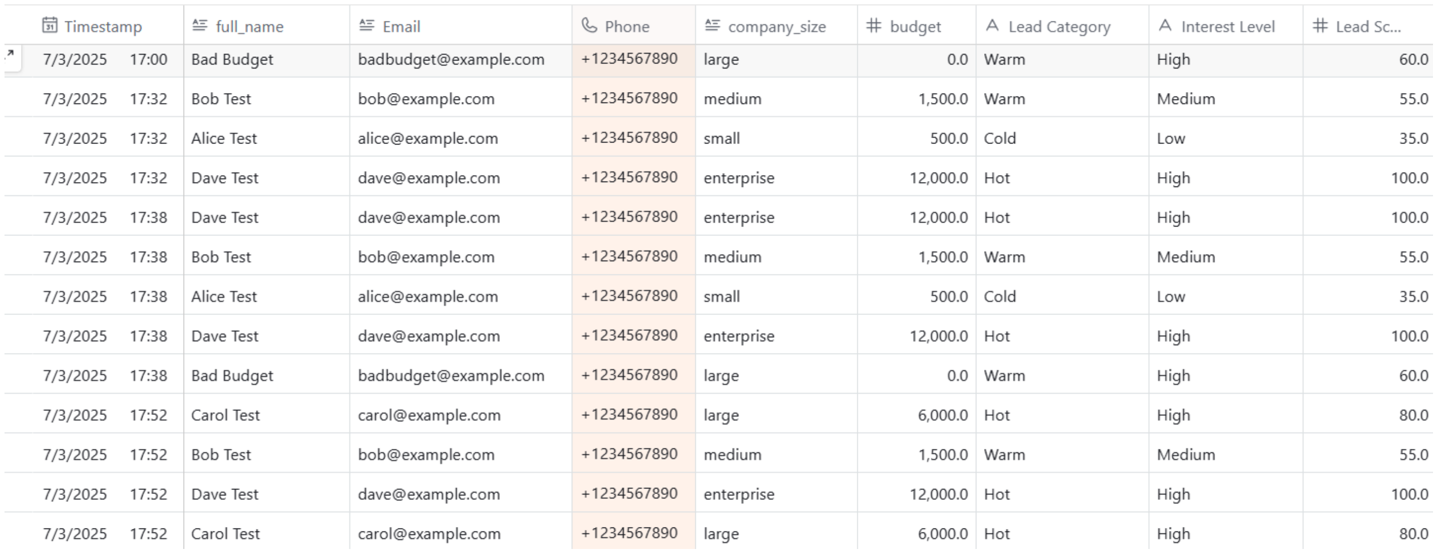
- **False Path:** Any validation or runtime error is sent to the **ErrorLogging** Airtable table—ensuring no lead data is dropped.



Timestamp	Error Log
2025-07-03T10:23:49.703-04:00	leadData.company_size?.toLowerCase is not a function [line 38]
2025-07-03T10:36:52.306-04:00	Referenced node is unexecuted [line 5]
2025-07-03T10:37:31.748-04:00	Referenced node is unexecuted [line 5]
2025-07-03T10:39:16.242-04:00	Referenced node is unexecuted [line 5]
2025-07-03T10:39:19.228-04:00	Referenced node is unexecuted [line 5]
2025-07-03T10:42:24.783-04:00	Cannot read properties of undefined (reading 'length') [line 12]
2025-07-03T11:00:01.299-04:00	leadData.company_size?.toLowerCase is not a function [line 46]

StoreLead (Airtable)

- **Operation:** Upserts the scored lead into the "Leads" base, keyed on email.
- **Fields:** Captures raw JSON, lead_id, lead_score, lead_category, scored_at, plus original form fields.
- **Options: Typecast** enabled so numbers and strings convert automatically to match Airtable field types.
- **Idempotency:** Re-running the workflow won't create duplicates thanks to the upsert key.



Timestamp	full_name	Email	Phone	company_size	budget	Lead Category	Interest Level	Lead Sc...
7/3/2025 17:00	Bad Budget	badbudget@example.com	+1234567890	large	0.0	Warm	High	60.0
7/3/2025 17:32	Bob Test	bob@example.com	+1234567890	medium	1,500.0	Warm	Medium	55.0
7/3/2025 17:32	Alice Test	alice@example.com	+1234567890	small	500.0	Cold	Low	35.0
7/3/2025 17:32	Dave Test	dave@example.com	+1234567890	enterprise	12,000.0	Hot	High	100.0
7/3/2025 17:38	Dave Test	dave@example.com	+1234567890	enterprise	12,000.0	Hot	High	100.0
7/3/2025 17:38	Bob Test	bob@example.com	+1234567890	medium	1,500.0	Warm	Medium	55.0
7/3/2025 17:38	Alice Test	alice@example.com	+1234567890	small	500.0	Cold	Low	35.0
7/3/2025 17:38	Dave Test	dave@example.com	+1234567890	enterprise	12,000.0	Hot	High	100.0
7/3/2025 17:38	Bad Budget	badbudget@example.com	+1234567890	large	0.0	Warm	High	60.0
7/3/2025 17:52	Carol Test	carol@example.com	+1234567890	large	6,000.0	Hot	High	80.0
7/3/2025 17:52	Bob Test	bob@example.com	+1234567890	medium	1,500.0	Warm	Medium	55.0
7/3/2025 17:52	Dave Test	dave@example.com	+1234567890	enterprise	12,000.0	Hot	High	100.0
7/3/2025 17:52	Carol Test	carol@example.com	+1234567890	large	6,000.0	Hot	High	80.0

IsHot? (If)

- **Check:**

```
{{$json.lead_category === 'Hot'}}
```

Only Hot leads follow the True branch for immediate notification.

NotifyHot (Slack)

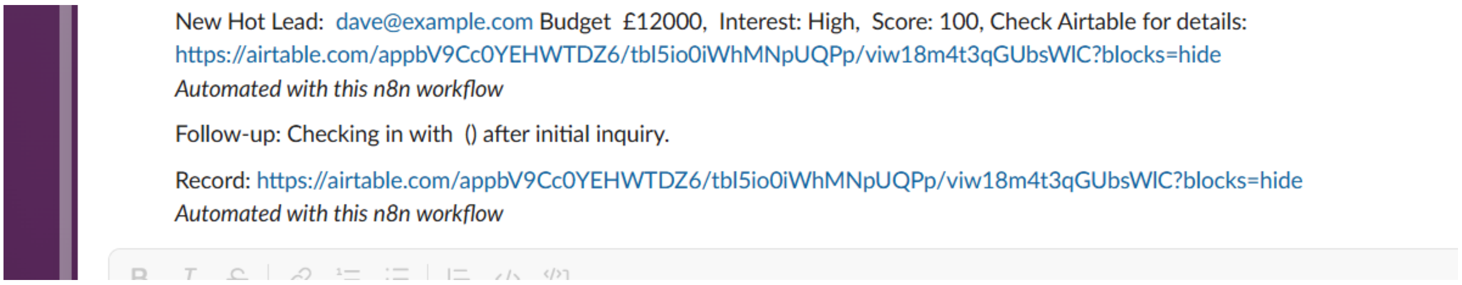
- **Action:** Sends a direct message to the assigned sales rep's Slack user ID.
- **Content:**
  - Lead name, budget, interest level, overall score
  - Deep link to the Airtable record for one-click access
- **Purpose:** Ensures high-value opportunities are surfaced instantly to the right person.

Follow Up Wait (Wait)

- **Pause:** Delays the workflow for **2 minutes** after the Hot notification.
- **Use Case:** Simulates a quick nurture step to re-engage the lead or remind the rep.

Reminder (Slack)

- **Action:** Sends a follow-up Slack message 2 minutes after the initial Hot-lead notification, using the same channel or direct-message thread.
- **Content:**
  - **Restates** the lead's name and email to remind the rep who the prospect is.
  - **Calls out** any key details (budget, interest level, score) to reinforce context.
  - **Includes** a link back to the Airtable record for quick access.
  - **Adds** a gentle prompt (e.g. "Have you had a chance to reach out?") to drive action.
- **Benefit:**
  Prevents high-value leads from slipping through the cracks by issuing a timely reminder, increasing the odds of rapid follow-up and deal conversion.
- **Example:**



New Hot Lead: dave@example.com Budget £12000, Interest: High, Score: 100, Check Airtable for details: https://airtable.com/appbV9Cc0YEHWTDZ6/tbl5io0iWhMNpUQPp/viw18m4t3qGUbsWlC?blocks=hide
Automated with this n8n workflow

Follow-up: Checking in with () after initial inquiry.

Record: https://airtable.com/appbV9Cc0YEHWTDZ6/tbl5io0iWhMNpUQPp/viw18m4t3qGUbsWlC?blocks=hide
Automated with this n8n workflow

ErrorTrigger (Error Trigger)

- **Type:** `n8n-nodes-base.errorTrigger`
- **Purpose:** Acts as a global catch-all for any exception thrown in upstream nodes (validation failures, runtime errors, API timeouts, etc.).
- **Behavior:**
  - Listens on the **error** output of all previous nodes with Error Output enabled.
  - Emits a single item containing an error object with fields like message, nodeName, and the original item.json payload.

ErrorLogging (Airtable)

- **Operation: Create**
- **Mapped Fields:**
    - **Error Message:** {{$json.error.message}}
    - **Failed Node:** {{$json.error.nodeName}}
    - **Raw Payload:** {{ JSON.stringify($json.error.item.json) }}
    - **Timestamp:** {{ $now.toISOString() }}
- **Benefit:** Persists every error with full context so ops can triage and manually recover any lost leads.

Remainder1 (Slack)

- **Type:** n8n-nodes-base.slack
- **Channel/DM:** #errors (or a dedicated incident channel)
- **Message Template:**

```
🚨 *Workflow Error Detected*
• *Node:* {{ $json.error.nodeName }}
• *Message:* {{ $json.error.message }}
• *Payload:* ```{{ JSON.stringify($json.error.item.json, null, 2) }}```
• *Time:* {{ $now.toISOString() }}
```

- **Behavior:** Fires immediately after ErrorLogging, giving your team real-time visibility into failures along with the exact data that caused them.

Robustness & Error-Handling Features

Raw Payload Archival

- **RawDataBackUP** archives every incoming JSON (simulated or live) to Airtable with a timestamp before any processing—so you can always recover leads if later steps fail.

Input Validation Guardrails

- The Function node explicitly checks for full_name, email, budget, and interest_level.
- Any missing fields produce an item with validation_error set, rather than letting bad data proceed.

Runtime Error Branching

- Error Output is **not** enabled on the scoring node, so unhandled exceptions (e.g. malformed budget strings, calling .toLowerCase() on non-strings) don't stop the run.
- Instead, the node emits a normal item with an error property, which is then caught downstream.

IfValid Gatekeeper

- An **If** node checks

Cristian Rojas
cristianjrojas@gmail.com

```
{{$json.validation_error !== true && $json.error === undefined}}
```

and only lets fully valid items continue.

- Anything with `validation_error` or `error` goes to the error branch.

2. **ErrorTrigger + ErrorLogging**

- An **Error Trigger** node (where enabled) catches thrown errors;
- **ErrorLogging** Airtable tables store each error's message, origin node, raw payload, and timestamp for manual triage.

3. **Real-Time Slack Alerts**

- A Slack node posts formatted error details (node name, message, payload) to your #errors channel immediately after logging.

4. **Type-Safe Parsing & Airtable Typecast**

- Budget values are parsed with parseFloat on sanitized strings to avoid NaN.
- Airtable nodes have **Typecast** enabled so mismatched types (strings vs. numbers, select fields) are coerced rather than rejected.

Additional Robustness Features

- **Retry Logic**
  All Airtable nodes, Slack nodes, and the custom Function node are configured to **retry up to 3 times** on failure before finally emitting an error—giving transient API hiccups a chance to recover automatically.
- **Workflow Timeout**
  A global **timeout** is set to **5 minutes**. If the workflow ever loops or stalls beyond that, it auto-terminates to prevent runaway executions.
- **Multi-Source Input Fallback**
  The validation node's input logic checks both production and testing sources in order, ensuring no blank path ever slips through:

```
let leadData;
if ($node["FromForm - Production"].items[0].json.body) {
  leadData = $node["FromForm - Production"].items[0].json.body;
} else if ($node["FromForm - Production"].items[0].json.data) {
  leadData = $node["FromForm - Production"].items[0].json.data;
} else {
  leadData = $node["FromForm - Production"].items[0].json.body;
}
```

This fallback ensures that if one data source is missing or malformed, the next is tried before failing.

Cristian Rojas
cristianjrojas@gmail.com

Customization & Extension

- **Adjust Scoring**: Modify weights and thresholds in Lead Validation and Scoring.
- **Add Enrichment**: Insert an HTTP Request to Clearbit, LinkedIn, or your CRM API.
- **Multi-Touch Nurture**: Chain additional Waits and Email/SMS nodes for Warm/Cold leads.
- **Reporting**: Create a separate Cron → Airtable “Stats” workflow to tally daily volumes.