

AI Setup Guide - OpenRouter Integration

Complete guide to adding AI-powered lead scoring, personalization, and automation to your n8n workflow.

Table of Contents

1. [Why AI Makes This a Game-Changer](#)
 2. [OpenRouter Setup \(5 Minutes\)](#)
 3. [Configuring AI in Each Workflow](#)
 4. [Prompt Engineering Guide](#)
 5. [Cost Analysis](#)
 6. [Troubleshooting](#)
-

Why AI Makes This a Game-Changer

Without AI (Basic Template Workflow)

Lead: "Interested in 3-bedroom house, budget \$500k, downtown area, need ASAP"

Response:

"Hi [Name], thanks for reaching out! An agent will contact you soon."

Problem: Generic, misses urgency signals, no personalization.

With AI (This System)

Lead: "Interested in 3-bedroom house, budget \$500k, downtown area, need ASAP"

AI Analysis:

- Score: 9/10 (HOT)
- Budget: \$500k (mid-tier)
- Urgency: IMMEDIATE
- Locality: Downtown
- Property type: House, 3BR

AI Response:

"Hi Sarah! I saw you're looking for a 3-bedroom home downtown around \$500k - great timing, the market just shifted in buyers' favor! I have 2 perfect matches ready to show you TODAY. Calling you in 60 seconds at (555) 123-4567. - Mike"

Result: Personalized, acknowledges specifics, creates urgency, feels human.

OpenRouter Setup (5 Minutes)

What is OpenRouter?

OpenRouter gives you access to **ALL major AI models** (Claude, GPT-4, Gemini, etc.) through one API with pay-as-you-go pricing. No subscriptions, no commitments.

Why OpenRouter vs Direct OpenAI/Anthropic?

- ☒ No monthly subscription (pay per request)
 - ☒ Access to multiple models (Claude often better for real estate)
 - ☒ Cheaper than direct APIs
 - ☒ Better rate limits for small businesses
 - ☒ Automatic fallback if one model is down
-

Step 1: Create OpenRouter Account

1. Go to: <https://openrouter.ai>
 2. Click **"Sign Up"**
 3. Sign up with Google/GitHub or email
 4. **Verify your email** (important!)
-

Step 2: Add Credits

1. Click your profile → **"Keys & Credits"**
2. Click **"Add Credits"**
3. Add **\$5-10 to start** (will last 500-1000 leads)
4. Payment via credit card (instant)

Cost Calculator:

- Claude 3.5 Sonnet: ~\$0.015 per lead analysis
 - GPT-4o: ~\$0.01 per lead
 - **\$10 = ~600-700 leads processed**
-

Step 3: Get Your API Key

1. Go to: **Keys & Credits** → **"API Keys"**
 2. Click **"Create API Key"**
 3. Name it: `n8n-realtor-workflow`
 4. **Copy the key** (starts with `sk-or-v1-...`)
 5. **Save it securely** (you can't see it again!)
-

Step 4: Add Key to n8n Workflows

You'll need to add this key to **3 Code nodes** across the workflows:

Workflow 1: Main Lead Intake (lead-intake-ai-enhanced.json)

Nodes to update:

1. **"AI Lead Scoring & Enrichment"** (Code node)
2. **"AI Personalized Responses"** (Code node)

How to update:

1. Import workflow to n8n
2. Click the **"AI Lead Scoring & Enrichment"** node
3. Find this line in the code:

```
'Authorization': 'Bearer YOUR_OPENROUTER_API_KEY'
```

4. Replace `YOUR_OPENROUTER_API_KEY` with your actual key:

```
'Authorization': 'Bearer sk-or-v1-abc123...'
```

5. Click **Save**

6. Repeat for **"AI Personalized Responses"** node

Workflow 2: Follow-Up Drip Campaign (follow-up-drip-campaign.json)

Node to update:

- **"Generate AI Follow-Up Message"** (Code node)

Same process as above.

Workflow 3: Daily Agent Dashboard (daily-agent-dashboard.json)

Node to update:

- **"Generate AI Summary"** (Code node)

Same process as above.

Configuring AI in Each Workflow

Main Lead Intake Workflow

AI Lead Scoring Node

What it does:

- Analyzes lead message for urgency signals
- Extracts budget, locality, property type
- Scores 1-10 based on quality
- Categorizes as Hot/Warm/Cold

Default Configuration:

```
model: 'anthropic/claude-3.5-sonnet'  
temperature: 0.3 // Lower = more consistent, higher = more creative
```

Customize the scoring criteria:

Find this section in the prompt:

```
Provide:  
1. Lead score (1-10, where 10 is highest quality/urgency)
```

Adjust scoring logic by editing the prompt. Example customization:

```
Lead Scoring Criteria:  
- Score 9-10: Budget mentioned, specific property, timeline < 1 week, phone provided  
- Score 7-8: Budget range, general area, timeline < 1 month  
- Score 4-6: Browsing, no budget, vague timeline  
- Score 1-3: Spam, incomplete info, unrealistic budget
```

Give higher scores to:

- **Luxury buyers** (budget > \$1M)
- **Investors** (multiple property interest)
- **Relocating professionals** (corporate)

AI Personalization Node

What it does:

- Generates custom WhatsApp message (140 chars)
- Generates custom email (HTML formatted)
- References specific property, budget, area from lead

Customize tone:

Find these lines and adjust:

```
// For WhatsApp
Tone: Friendly check-in, not pushy

// Change to:
Tone: Enthusiastic, high-energy, competitive market urgency

// For Email
Tone: Helpful expert, providing value

// Change to:
Tone: Luxury concierge service, white-glove treatment
```

Follow-Up Drip Campaign Workflow

AI Follow-Up Generator

What it does:

- Day 1: Casual WhatsApp nudge
- Day 3: Email with "similar listings"
- Day 7: Market insights email
- Day 14: Graceful final check-in

Customize follow-up schedule:

Find this section in "**Calculate Follow-Up Stage**" node:

```
if (daysSinceReceived === 1 && followupCount === 0) {
  followupStage = 'day1';
}
```

Change to your schedule:

```
// Faster follow-up (aggressive)
if (daysSinceReceived === 0.5 && followupCount === 0) { // 12 hours
  followupStage = 'day1';
}
```

```
} else if (daysSinceReceived === 1 && followupCount <= 1) {
  followupStage = 'day3';
}

// Slower follow-up (luxury market)
if (daysSinceReceived === 3 && followupCount === 0) {
  followupStage = 'day1';
} else if (daysSinceReceived === 7 && followupCount <= 1) {
  followupStage = 'day3';
}
```

Customize message tone per stage:

In "Generate AI Follow-Up Message" node, find:

```
case 'day1':
  promptContext = `This is a DAY 1 FOLLOW-UP...
  Tone: Friendly check-in, not pushy.`
```

Change tone:

```
case 'day1':
  promptContext = `This is a DAY 1 FOLLOW-UP...
  Tone: Urgent market opportunity, limited inventory, act now.`
```

Daily Agent Dashboard Workflow

AI Summary Generator

What it does:

- Creates personalized morning briefing for each agent
- Highlights priorities (hot leads, follow-ups, etc.)
- Motivating/coaching tone

Customize tone:

Find:

```
Tone: Encouraging coach
```

Change to:

```
// For competitive teams
Tone: Competitive leaderboard mindset, rankings, goal-oriented

// For luxury market
Tone: Refined professional, high-stakes opportunities
```

```
// For team managers
Tone: Executive summary, strategic priorities, team optimization
```

Prompt Engineering Guide

Best Practices for Real Estate AI Prompts

1. Be Specific About Output Format

✖ Bad:

```
content: "Analyze this lead"
```

✔ Good:

```
content: "Analyze this lead and return JSON with these exact fields:
score (number 1-10), priority (string: hot/warm/cold),
budget_tier (string: entry/mid/luxury)..."
```

2. Provide Context & Examples

✖ Bad:

```
content: "Write a follow-up message"
```

✔ Good:

```
content: "Write a Day 3 follow-up email for a lead who inquired about
a $450k condo downtown but didn't respond to initial contact.
Include 2-3 similar property suggestions and market insights for
the downtown condo market. Tone: helpful expert."
```

3. Control Creativity with Temperature

```
// For consistent, factual output (scoring, data extraction)
temperature: 0.1-0.3

// For creative writing (emails, summaries)
temperature: 0.6-0.8

// For very creative (marketing copy)
temperature: 0.9-1.0
```

4. Real Estate-Specific Prompt Templates

Lead Scoring Prompt Template:

```
`You are a real estate lead qualification expert with 20 years of experience.
```

```
Analyze this inquiry and score 1-10 based on:
1. Urgency signals (words like "ASAP", "urgent", "this week")
2. Budget clarity (specific number vs "flexible")
3. Property specificity (address vs "something nice")
4. Contact completeness (phone + email vs just one)
5. Intent signals (viewing, offer, just browsing)
```

```
Lead: "${leadData.lead_message}"
```

```
Buyer signals: ${buyerSignals}
Seller signals: ${sellerSignals}
```

```
Return ONLY valid JSON with exact schema provided.`
```

Personalization Prompt Template:

```
`You are ${agentName}, a top-performing real estate agent in ${city}.

Write a brief, warm WhatsApp message to ${leadName} who inquired about:
"${propertyInterest}"

Your message should:
- Use their first name naturally
- Reference the specific property or area
- Create gentle urgency (market moving fast, etc.)
- Confirm you'll call in 60 seconds
- Sound human, not robotic
- Stay under 140 characters

Market context: ${marketCondition}
Lead priority: ${priority}

Write ONLY the message text.`
```

Cost Analysis

Actual Costs Per Lead (OpenRouter)

Lead Intake Workflow (2 AI Calls)

AI Lead Scoring:

- Model: Claude 3.5 Sonnet
- Input tokens: ~400 (lead data + prompt)
- Output tokens: ~200 (JSON response)
- **Cost: \$0.0045**

AI Personalization (WhatsApp + Email):

- Model: Claude 3.5 Sonnet
- Input tokens: ~600
- Output tokens: ~400

- **Cost: \$0.009**

Total per lead: ~\$0.0135 (1.35 cents)

Monthly Costs by Volume

| Leads/Month | AI Cost | Total System Cost* |
|-------------|---------|--------------------|
| 50 leads | \$0.68 | \$30.68 |
| 100 leads | \$1.35 | \$31.35 |
| 500 leads | \$6.75 | \$36.75 |
| 1000 leads | \$13.50 | \$43.50 |

*Total includes n8n (\$20), Twilio (\$10), OpenRouter

Cost Savings vs. Alternatives

Compared to:

- **Virtual Assistant:** \$500-1500/mo → **Save \$456-1456/mo**
- **Lead Response Service:** \$200-400/mo → **Save \$156-356/mo**
- **AI Writing Tools (Jasper, Copy.ai):** \$49-99/mo → **Save \$15-65/mo**

ROI: If AI helps close **ONE additional deal** per year (very conservative), you've made 100-500x return.

Alternative AI Providers

If you prefer NOT to use OpenRouter:

Option 1: Direct OpenAI API

Pros: Official, reliable, well-documented **Cons:** More expensive, requires separate account

Setup:

1. Create account: <https://platform.openai.com>
2. Add payment method
3. Get API key
4. Replace in workflow:

```
url: 'https://api.openai.com/v1/chat/completions'
headers: {
  'Authorization': 'Bearer YOUR_OPENAI_KEY'
}
body: {
  model: 'gpt-4o' // or 'gpt-4-turbo'
}
```

Cost: ~\$0.015-0.02 per lead (slightly higher)

Option 2: Direct Anthropic API

Setup:

- 1. Create account: <https://console.anthropic.com>
- 2. Add payment
- 3. Get API key
- 4. Replace in workflow:

```
url: 'https://api.anthropic.com/v1/messages'
headers: {
  'x-api-key': 'YOUR_ANTHROPIC_KEY',
  'anthropic-version': '2023-06-01'
}
body: {
  model: 'claude-3-5-sonnet-20241022'
}
```

Cost: Similar to OpenRouter (~\$0.01-0.015 per lead)

Option 3: Free/Local AI (Advanced)

Using Ollama (Runs on your computer):

- Completely free
- Privacy (data never leaves your server)
- Requires technical setup

Models: Llama 3, Mistral, etc.

Setup: See [Ollama documentation](#)

Model Comparison for Real Estate

| Model | Best For | Cost | Pros | Cons |
|-------------------|--------------------------------|--------|--|--------------------|
| Claude 3.5 Sonnet | Lead analysis, personalization | \$\$ | Most accurate, best at following JSON schema | Slightly slower |
| GPT-4o | General tasks | \$\$\$ | Fast, good quality | More expensive |
| GPT-4o-mini | High-volume, budget | \$ | Very cheap, fast | Less nuanced |
| Claude 3 Haiku | Simple tasks, budget | \$ | Fast, cheap | Less sophisticated |

Recommendation: Start with **Claude 3.5 Sonnet** for quality, switch to **GPT-4o-mini** if volume/cost becomes an issue.

Troubleshooting

"Authorization failed" Error

Cause: Invalid API key

Fix:

1. Verify key starts with `sk-or-v1-` (OpenRouter) or `sk-` (OpenAI)
 2. Check for extra spaces when copying
 3. Ensure key is active (check OpenRouter dashboard)
 4. Verify credits available
-

"Model not found" Error

Cause: Wrong model name

Fix: Update model name to:

```
// OpenRouter format (with provider prefix)
model: 'anthropic/claude-3.5-sonnet'

// NOT:
model: 'claude-3.5-sonnet' // Wrong - missing provider
```

See full model list: <https://openrouter.ai/models>

AI Returns Non-JSON / Malformed Response

Cause: Model didn't follow instructions

Fix:

1. Make JSON requirement more explicit in prompt:

```
Return ONLY valid JSON. No markdown, no explanations, no text before or after.
Do not wrap in ```json blocks. Start with { and end with }.
```

2. Add JSON parsing with fallback:

```
let aiAnalysis;
try {
  // Try to extract JSON from markdown blocks
  const jsonMatch = aiContent.match(/```json\n?([\s\S]*?)\n?```/);
  const jsonString = jsonMatch ? jsonMatch[1] : aiContent;
  aiAnalysis = JSON.parse(jsonString);
} catch (e) {
  // Fallback to defaults
  aiAnalysis = { score: 5, priority: 'warm', ... };
}
```

High AI Costs

Optimize:

1. **Reduce temperature** (less tokens used):

```
temperature: 0.1 // Instead of 0.7
```

2. Shorten prompts (remove examples):

```
// Before: 800 tokens
content: `Long explanation... examples... context...`

// After: 200 tokens
content: `Analyze lead. Return JSON: {score, priority, budget}`
```

3. Switch to cheaper model for simple tasks:

```
// For lead scoring (needs accuracy)
model: 'anthropic/claude-3.5-sonnet' // Keep

// For follow-up messages (less critical)
model: 'openai/gpt-4o-mini' // 10x cheaper
```

4. Cache common prompts (advanced): OpenRouter supports prompt caching - see their docs.

Slow Performance

Cause: AI API calls add 1-3 seconds

Optimize:

1. Run AI calls in parallel (already done in workflows)
2. Use faster models:

```
model: 'openai/gpt-4o' // Fast
model: 'anthropic/claude-3-haiku' // Fastest
```

3. Set timeout:

```
timeout: 5000 // 5 second max
```

Advanced: Custom AI Agent Personas

Create different AI personalities for different market segments:

```
// Luxury market agent
const luxuryPrompt = `You are Alexandra, a luxury real estate specialist.
Tone: Refined, exclusive, white-glove service.
Vocabulary: "curated properties", "exclusive opportunity", "bespoke"`;

// First-time homebuyer agent
const ftbPrompt = `You are Mike, a friendly first-time homebuyer specialist.
```

```
Tone: Patient, educational, encouraging.
Vocabulary: "Let's find your perfect first home", "I'll guide you`;

// Investor agent
const investorPrompt = `You are David, a real estate investment advisor.
Tone: Data-driven, ROI-focused, market analysis.
Vocabulary: "cash flow", "cap rate", "market appreciation`;

// Select based on lead intent
const agentPersona = leadData.ai_intent === 'investor' ? investorPrompt :
                      leadData.ai_budget_tier === 'luxury' ? luxuryPrompt :
                      ftbPrompt;
```

Testing Your AI Configuration

Test Prompt in OpenRouter Playground

Before deploying to n8n:

1. Go to: <https://openrouter.ai/playground>
2. Select model: **Claude 3.5 Sonnet**
3. Paste your prompt
4. Add sample lead data
5. Click **Run**
6. Verify output format



Sample test data:

Lead: "Looking for 3BR house in downtown, budget around \$500k, need to move by end of month for new job. Can you show me something this week? -Sarah, 555-1234"

Expected output:

```
{
  "score": 9,
  "priority": "hot",
  "intent": "buyer",
  "urgency": "immediate",
  "budget_tier": "mid",
  "locality_preference": "downtown",
  "property_type": "house",
  "insights": ["Specific timeline (end of month)", "Relocation urgency", "Budget clearly stated"],
  "follow_up_hours": 1
}
```

Next Steps

1.  Set up OpenRouter account
2.  Add \$10 in credits

3. ☒ Get API key
 4. ☒ Update all 3 workflows with key
 5. ☒ Test with sample lead
 6. ☒ Review AI responses
 7. ☒ Customize prompts for your market
 8. ☒ Monitor costs in OpenRouter dashboard
-

Support & Resources

- **OpenRouter Docs:** <https://openrouter.ai/docs>
 - **Model Pricing:** <https://openrouter.ai/models>
 - **Prompt Engineering:** <https://docs.anthropic.com/claude/docs/prompt-engineering>
 - **n8n AI Nodes:** <https://docs.n8n.io/integrations/builtin/cluster-nodes/root-nodes/n8n-nodes-langchain.ai/>
-

 **Your AI-powered lead system is ready! Watch the magic happen with your first lead.**