

Integration Configuration Guide

Complete step-by-step guides for connecting your lead sources to the n8n workflow.

Table of Contents

- [1. Zillow Integration](#)
 - [2. Realtor.com Integration](#)
 - [3. Facebook Lead Ads](#)
 - [4. Custom Web Forms](#)
 - [5. OpenRouter AI Enhancement](#)
 - [6. Alternative CRM Setups](#)
 - [7. Alternative SMS Providers](#)
-

1. Zillow Integration

Method A: Zillow Premier Agent API (Recommended)

Prerequisites:

- Zillow Premier Agent account
- API access (contact Zillow support)

Setup Steps:

1. Request API Access

- Email: partnertech@zillow.com
- Subject: "API Webhook Integration Request"
- Provide: Business details, use case

2. Get Your n8n Webhook URL

```
https://your-n8n-instance.com/webhook/realtor-lead
```

3. Configure Zillow Webhook

- Log into Zillow Premier Agent dashboard
- Navigate to: **Settings** → **API Integrations**
- Add webhook endpoint
- Select events: `lead.created` , `lead.updated`
- Paste n8n webhook URL

4. Test Integration

- Zillow will send a test payload
- Check n8n execution log for successful receipt
- Verify all fields map correctly

Zillow Payload Structure:

```
{  
  "leadId": "zillow-12345",
```

```
"firstName": "John",
"lastName": "Smith",
"email": "john@example.com",
"phone": "555-1234",
"listingAddress": "123 Main St",
"message": "Interested in viewing",
"timestamp": "2024-01-15T14:30:00Z"
}
```

Field Mapping in n8n:

- `name` ← `firstName` + " " + `lastName`
- `email` ← `email`
- `phone` ← `phone`
- `property_interest` ← `listingAddress`
- `message` ← `message`
- `lead_id` ← `leadId`
- `source` ← "zillow"

Method B: Zillow Email Forwarding (Workaround)

If API access isn't available:

1. Set Up Email Parsing

- Forward Zillow lead emails to n8n Email Trigger
- Use n8n **Email Trigger (IMAP)** node
- Parse email body with Code node

2. Email Parser Code:

```
const email = $input.first().json;
const bodyText = email.text || email.html;

// Extract name (regex pattern varies by email format)
const nameMatch = bodyText.match(/Name: (.+)/);
const emailMatch = bodyText.match(/Email: (.+)/);
const phoneMatch = bodyText.match(/Phone: (.+)/);
const messageMatch = bodyText.match(/Message: (.+)/);

return [{
  json: {
    name: nameMatch ? nameMatch[1].trim() : 'Unknown',
    email: emailMatch ? emailMatch[1].trim() : '',
    phone: phoneMatch ? phoneMatch[1].trim() : '',
    message: messageMatch ? messageMatch[1].trim() : '',
    source: 'zillow',
    lead_id: 'ZILLOW-' + Date.now()
  }
}];
```

2. Realtor.com Integration

Method A: Realtor.com API

Prerequisites:

- Realtor.com Pro+ account
- API access (request from Realtor.com support)

Setup Steps:

1. Request API Credentials

- Contact: <https://www.realtor.com/support/>
- Request: "Lead API webhook integration"
- Receive: API Key + Webhook documentation

2. Configure Webhook

- Realtor.com dashboard → **Lead Settings**
- Enable **API Integration**
- Webhook URL: Your n8n webhook endpoint
- Events: Lead received

3. Authenticate Webhook (if required)

- Realtor.com may require token validation
- Add to webhook URL: `?token=YOUR_SECRET`
- Add IF node in n8n to validate token:

```
// In n8n, add IF node after webhook
{{$json.query.token}} equals "your-secret-token"
```

Realtor.com Payload:

```
{
  "leadId": "realtorcom-67890",
  "contact": {
    "firstName": "Jane",
    "lastName": "Doe",
    "email": "jane@example.com",
    "phone": "(555) 123-4567"
  },
  "property": {
    "address": "456 Oak Avenue",
    "city": "San Francisco",
    "state": "CA",
    "zip": "94102",
    "listingId": "987654"
  },
  "message": "What's the asking price?"
```

```
"timestamp": "2024-01-15T15:00:00Z"
}
```

Method B: Realtor.com Email Forwarding

Similar to Zillow email method - parse forwarded emails.

3. Facebook Lead Ads

Method A: n8n Facebook Lead Ads Trigger (Recommended)

Setup Steps:

1. Create Facebook App

- Go to: <https://developers.facebook.com/apps/>
- Create New App → Business Type
- App Name: "Lead Integration"
- Add Product: **Webhooks**

2. Subscribe to Page

- Webhooks → Page Subscriptions
- Subscribe to: `leadgen`
- Callback URL: Your n8n webhook
- Verify token: Create random string (e.g., `fb-verify-12345`)

3. Configure n8n

- Use **Facebook Lead Ads Trigger** node (built-in)
- Or use **Facebook Graph API** node
- Credentials: App ID, App Secret, Access Token

4. Get Page Access Token

- Tools → Graph API Explorer
- Select your page
- Generate token with `leads_retrieval` permission

5. Test

- Create test lead ad
 - Submit test form
 - Verify n8n receives data
-

Method B: Facebook Webhook (Manual)

1. In Facebook:

- Business Settings → Lead Access
- Enable webhook
- URL: `https://your-n8n.com/webhook/realtor-lead`
- Verify token: `your-verify-token`

2. In n8n Webhook Node:

- Add verification logic:

```
// Add to webhook response
if ($json.query['hub.mode'] === 'subscribe' &&
    $json.query['hub.verify_token'] === 'your-verify-token') {
    return {
        status: 200,
        body: $json.query['hub.challenge']
    };
}
```

Facebook Payload:

```
{
  "entry": [{
    "changes": [{
      "value": {
        "leadgen_id": "12345",
        "form_id": "67890",
        "page_id": "111213",
        "created_time": 1642262400
      }
    }]
  }]
}
```

Then fetch full lead data:

```
// HTTP Request node
GET https://graph.facebook.com/v12.0/{leadgen_id}
?access_token=YOUR_TOKEN
```

4. Custom Web Forms

Basic HTML Form with JavaScript

1. HTML Form:

```
<!DOCTYPE html>
<html>
<head>
  <title>Contact a Real Estate Agent</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      max-width: 600px;
      margin: 50px auto;
      padding: 20px;
    }
  </style>
</head>
<body>
  <div>
    <h1>Contact Us</h1>
    <div>
      <div>
        <input type="text" value="Name" />
      </div>
      <div>
        <input type="text" value="Email" />
      </div>
      <div>
        <input type="text" value="Phone" />
      </div>
      <div>
        <input type="text" value="Address" />
      </div>
      <div>
        <input type="text" value="City" />
      </div>
      <div>
        <input type="text" value="State" />
      </div>
      <div>
        <input type="text" value="Zip" />
      </div>
      <div>
        <input type="text" value="Website" />
      </div>
      <div>
        <input type="text" value="Comments" />
      </div>
      <div>
        <input type="button" value="Submit" />
      </div>
    </div>
  </div>
</body>
</html>
```

```

.form-group {
  margin-bottom: 15px;
}
label {
  display: block;
  margin-bottom: 5px;
  font-weight: bold;
}
input, textarea {
  width: 100%;
  padding: 10px;
  border: 1px solid #ddd;
  border-radius: 4px;
  font-size: 14px;
}
button {
  background-color: #007bff;
  color: white;
  padding: 12px 30px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  font-size: 16px;
}
button:hover {
  background-color: #0056b3;
}
.success-message {
  display: none;
  background-color: #d4edda;
  color: #155724;
  padding: 15px;
  border-radius: 4px;
  margin-top: 20px;
}
</style>
</head>
<body>
<h1>Interested in this property?</h1>
<p>Fill out the form below and an agent will contact you within 5 minutes!</p>

<form id="lead-form">
  <div class="form-group">
    <label for="name">Full Name *</label>
    <input type="text" id="name" name="name" required>
  </div>

  <div class="form-group">
    <label for="email">Email *</label>
    <input type="email" id="email" name="email" required>
  </div>

```

```

<div class="form-group">
  <label for="phone">Phone Number *</label>
  <input type="tel" id="phone" name="phone" required>
</div>

<div class="form-group">
  <label for="property">Property Address</label>
  <input type="text" id="property" name="property" placeholder="123 Main St, City,
State">
</div>

<div class="form-group">
  <label for="message">Message</label>
  <textarea id="message" name="message" rows="4" placeholder="When would you like to
schedule a viewing?"></textarea>
</div>

<button type="submit">Submit Inquiry</button>
</form>

<div class="success-message" id="success">
  ✓ Thank you! An agent will contact you within 5 minutes.
</div>

<script>
document.getElementById('lead-form').addEventListener('submit', async function(e) {
  e.preventDefault();

  // Get form data
  const formData = {
    name: document.getElementById('name').value,
    email: document.getElementById('email').value,
    phone: document.getElementById('phone').value,
    property_address: document.getElementById('property').value,
    message: document.getElementById('message').value,
    source: 'website',
    timestamp: new Date().toISOString(),
    page_url: window.location.href
  };

  // Send to n8n webhook
  try {
    const response = await fetch('https://your-n8n-instance.com/webhook/realtor-lead', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify(formData)
    });

    if (response.ok) {
      // Show success message
    }
  }
});

```

```

        document.getElementById('success').style.display = 'block';
        document.getElementById('lead-form').style.display = 'none';
    } else {
        alert('Error submitting form. Please try again or call us directly.');
```

```
    }
```

```
  } catch (error) {
```

```
    console.error('Error:', error);
```

```
    alert('Error submitting form. Please try again or call us directly.');
```

```
  }
```

```
});
```

```
</script>
```

```
</body>
```

```
</html>
```

2. WordPress Integration:

```
// Add to functions.php in your WordPress theme
```

```
add_action('wp_ajax_submit_lead', 'handle_lead_submission');
```

```
add_action('wp_ajax_nopriv_submit_lead', 'handle_lead_submission');
```

```
function handle_lead_submission() {
```

```
    $lead_data = array(
```

```
        'name' => sanitize_text_field($_POST['name']),
```

```
        'email' => sanitize_email($_POST['email']),
```

```
        'phone' => sanitize_text_field($_POST['phone']),
```

```
        'property_address' => sanitize_text_field($_POST['property']),
```

```
        'message' => sanitize_textarea_field($_POST['message']),
```

```
        'source' => 'wordpress',
```

```
        'timestamp' => current_time('mysql')
```

```
    );
```

```
// Send to n8n webhook
```

```
$response = wp_remote_post('https://your-n8n-instance.com/webhook/realtor-lead', array(
```

```
    'headers' => array('Content-Type' => 'application/json'),
```

```
    'body' => json_encode($lead_data)
```

```
));
```

```
if (is_wp_error($response)) {
```

```
    wp_send_json_error('Failed to submit');
```

```
} else {
```

```
    wp_send_json_success('Lead submitted');
```

```
}
```

```
}
```

5. OpenRouter AI Enhancement (Optional)

Add AI Lead Scoring & Personalization

1. Create OpenRouter Account:

- Sign up: <https://openrouter.ai>
- Get API key from dashboard
- Add credits (pay-as-you-go)

2. Add AI Lead Scoring Node (After "Extract Lead Data"):

```
// Code node: AI Lead Scoring
const leadData = $input.first().json;

const response = await $helpers.httpRequest({
  method: 'POST',
  url: 'https://openrouter.ai/api/v1/chat/completions',
  headers: {
    'Authorization': 'Bearer YOUR_OPENROUTER_API_KEY',
    'Content-Type': 'application/json',
    'HTTP-Referer': 'https://your-website.com',
    'X-Title': 'Real Estate Lead Scoring'
  },
  body: {
    model: 'anthropic/claude-3.5-sonnet',
    messages: [{
      role: 'user',
      content: `Analyze this real estate lead and provide a JSON response:

Lead Information:
- Name: ${leadData.lead_name}
- Email: ${leadData.lead_email}
- Phone: ${leadData.lead_phone}
- Message: ${leadData.property_interest}
- Source: ${leadData.lead_source}

Provide:
1. Lead score (1-10, where 10 is highest quality)
2. Priority level (high/medium/low)
3. Intent (buyer/seller/renter/investor)
4. Urgency (immediate/this_week/this_month/browsing)
5. Estimated budget tier (entry/mid/luxury)
6. Key insights (2-3 bullet points)

Return ONLY valid JSON in this format:
{
  "score": 8,
  "priority": "high",
  "intent": "buyer",
  "urgency": "immediate",
  "budget_tier": "mid",
  "insights": ["Specific property interest", "Timeline mentioned", "Budget indicated"]
}`
    }]
  }
});
```

```
// Parse AI response
const aiContent = response.choices[0].message.content;
const aiAnalysis = JSON.parse(aiContent);

return [{
  json: {
    ...leadData,
    ai_score: aiAnalysis.score,
    ai_priority: aiAnalysis.priority,
    ai_intent: aiAnalysis.intent,
    ai_urgency: aiAnalysis.urgency,
    ai_budget_tier: aiAnalysis.budget_tier,
    ai_insights: aiAnalysis.insights
  }
}];
```

3. Use AI Insights for Routing:

Add IF node after AI scoring:

```
// Route high-priority leads to top agents
{{ $json.ai_priority }} equals "high"
```

4. AI-Personalized Email/SMS:

```
// Code node: Generate personalized response
const leadData = $input.first().json;

const response = await $helpers.httpRequest({
  method: 'POST',
  url: 'https://openrouter.ai/api/v1/chat/completions',
  headers: {
    'Authorization': 'Bearer YOUR_OPENROUTER_API_KEY',
    'Content-Type': 'application/json'
  },
  body: {
    model: 'anthropic/claude-3.5-sonnet',
    messages: [{
      role: 'user',
      content: `Write a brief, friendly SMS message (under 160 characters) for a real estate agent to send to this lead:

Lead: ${leadData.lead_name}
Their message: ${leadData.property_interest}
Agent name: ${leadData.assigned_agent_name}

Make it personal, acknowledge their specific interest, and confirm quick follow-up.`
    }]
  }
});
```

```
const personalizedMessage = response.choices[0].message.content;

return [{
  json: {
    ...leadData,
    personalized_sms: personalizedMessage
  }
}];
```

Then use `{{ $json.personalized_sms }}` in SMS node instead of template.

6. Alternative CRM Setups

Airtable (Easiest)

Setup:

- Create Airtable base:
 - Table: "Leads"
 - Fields: Name, Email, Phone, Source, Agent, Status, Timestamp
- In n8n, use **Airtable** node (built-in):
 - Operation: Create
 - Base ID: From Airtable URL
 - Table: "Leads"
- Map fields:

```
{
  "Name": "={{ $json.lead_name }}",
  "Email": "={{ $json.lead_email }}",
  "Phone": "={{ $json.lead_phone }}",
  "Source": "={{ $json.lead_source }}",
  "Assigned Agent": "={{ $json.assigned_agent_name }}",
  "Status": "new",
  "Timestamp": "={{ $json.timestamp }}"
}
```

Pros: Visual, easy, free tier generous **Cons:** Limited automation compared to HubSpot/Salesforce

Salesforce

Setup:

- Get Salesforce credentials:
 - Setup → Apps → App Manager
 - Create Connected App
 - Get Consumer Key + Consumer Secret
- In n8n, use **Salesforce** node:

- Operation: Create Lead
- Credentials: OAuth2

3. Field mapping:

```
{
  "FirstName": "={{ $json.lead_name.split(' ')[0] }}",
  "LastName": "={{ $json.lead_name.split(' ').slice(1).join(' ') }}",
  "Email": "={{ $json.lead_email }}",
  "Phone": "={{ $json.lead_phone }}",
  "LeadSource": "={{ $json.lead_source }}",
  "Status": "New",
  "Description": "={{ $json.property_interest }}"
}
```

Google Sheets (Simplest)

Setup:

1. Create Google Sheet with columns:
 - Timestamp, Name, Email, Phone, Source, Agent, Status
2. In n8n, use **Google Sheets** node:
 - Operation: Append Row
 - Sheet ID: From sheet URL
 - Range: Sheet1!A:G
3. Map values:

```
{
  "values": [[
    "{{ $json.timestamp }}",
    "{{ $json.lead_name }}",
    "{{ $json.lead_email }}",
    "{{ $json.lead_phone }}",
    "{{ $json.lead_source }}",
    "{{ $json.assigned_agent_name }}",
    "new"
  ]]
}
```

Pros: Free, simple, familiar **Cons:** No advanced CRM features

7. Alternative SMS Providers

Vonage (Nexmo)

Setup:

1. Sign up: <https://dashboard.nexmo.com>
2. Get API Key + API Secret
3. Buy virtual number

n8n Configuration:

```
// HTTP Request node
{
  "method": "POST",
  "url": "https://rest.nexmo.com/sms/json",
  "body": {
    "from": "YOUR_VONAGE_NUMBER",
    "to": "{{ $json.lead_phone }}",
    "text": "Hi {{ $json.lead_name }}! Thanks for your inquiry...",
    "api_key": "YOUR_API_KEY",
    "api_secret": "YOUR_API_SECRET"
  }
}
```

MessageBird

Setup:

1. Sign up: <https://messagebird.com>
2. Get API key
3. Buy phone number

n8n Configuration:

```
// HTTP Request node with Header Auth
Headers: {
  "Authorization": "AccessKey YOUR_MESSAGEBIRD_KEY"
}

Body: {
  "recipients": ["{{ $json.lead_phone }}"],
  "originator": "YOUR_NUMBER",
  "body": "Hi {{ $json.lead_name }}!..."
}

URL: https://rest.messagebird.com/messages
```

AWS SNS (Enterprise)

Setup:

1. AWS Console → SNS
2. Create topic
3. Subscribe phone numbers
4. Get AWS credentials

n8n Configuration:

Use **AWS SNS** node:

- Topic ARN or Phone Number
 - Message: Your SMS text
-

Quick Reference: All Webhook URLs

Once workflow is activated, you'll have this webhook URL:

```
Production: https://your-n8n-instance.com/webhook/realtor-lead
Test: https://your-n8n-instance.com/webhook-test/realtor-lead
```

Send POST requests with JSON body:

```
{
  "name": "Lead Name",
  "email": "lead@example.com",
  "phone": "+1234567890",
  "property_address": "Address or listing URL",
  "message": "Lead's message/inquiry",
  "source": "zillow|realtor|facebook|website"
}
```

Troubleshooting

Webhook receives data but workflow doesn't process

1. Check execution logs in n8n
2. Verify webhook path matches exactly
3. Test with manual execution first
4. Check for errors in Set/Code nodes

CRM not creating contacts

1. Verify API permissions
2. Check field mapping (required vs optional)
3. Test API endpoint with Postman
4. Review error logs in n8n execution

SMS/Email not sending

1. Verify credentials are correct
 2. Check service provider dashboard for errors
 3. Verify phone number format (+1XXXXXXXXXX)
 4. Check rate limits and quotas
-

Need help? Check n8n community forums or the main README-SETUP.md guide.