By Wilmer Toro

superservicebros.com

Angi Lead Pitbull — Detailed Workflow Breakdown

What this automation solves (real-world)

If you buy leads from Angi, you already know the reality:

You're not competing on skill.
You're competing on **speed**.

Angi leads are brutal because:

- The customer is contacting multiple companies at once
- Whoever responds first usually gets the appointment
- Manual follow-up is too slow (especially after hours)
- A big chunk of leads are **outside your coverage area**, so wasting time on them costs money

This automation gives you:

- ⚡ Speed (instant response)

- 🧠 Intelligence (service-area filtering)
- 📈 Consistency (every lead handled the same way)
- 🛡️ Protection (no chasing bad leads)

The homeowner is sending the same request to multiple companies, and the one who responds first usually wins. If your response is delayed, generic, or inconsistent, that lead is gone.

This automation was built to solve that problem aggressively — but intelligently.

We call it the **Angi Lead Pitbull** because it grabs the lead fast, filters out bad fits, and responds immediately with professional, on-brand messages. A COD lead for us is like a Ribeye for a pitbull.

The Real Problems This Automation Solves

Angi leads come with a few built-in challenges:

- Leads arrive by email and are easy to miss
- Response speed drops after hours or during busy days
- Many leads are **outside your service area**
- Manual responses are inconsistent and slow
- CSRs waste time following up on bad leads

This workflow fixes all of that automatically.

What This Automation Does (Summarized)

Here’s what happens the moment an Angi lead hits your inbox:

1. n8n detects the Angi lead email instantly
2. The email is cleaned and normalized
3. Lead details are extracted (name, phone, address, issue)

4. AI checks if the zip code is inside your service area
5. AI generates a short SMS + professional email
6. If covered → SMS + email are sent immediately
7. If not covered → a polite decline email is sent

No manual review. No copy/paste. No wasted follow-ups.

High-level flow (what happens end-to-end)

Trigger → clean email → extract lead info → add business info → AI generates responses + checks zip coverage → parse output → route based on coverage → send SMS + email (covered) OR email-only (not covered).

Stage 1 — Detect and Normalize the Lead Email

1) Two Gmail Triggers (covers Angi email variations)

You have **two separate Gmail Trigger nodes** watching for Angi lead emails:

Trigger A

- Query: `Subject: New Customer Match: Appliance Repair`
- Sender: `no-reply@angi.com`

Trigger B

- Query: `Subject: New Lead`
- Sender: `no-reply@angi.com`

Why this matters:
Angi sends leads in slightly different formats depending on campaign type or how the lead was generated. Two triggers keeps the workflow from missing leads.

Both triggers funnel into the same cleaning and extraction pipeline.

---

2) “Clean Email” Code Node (HTML/RTF → clean text)

Angi emails can come in:

- HTML
- text
- “textAsHtml”
- sometimes even RTF formatting

This node:

- picks the richest body available (`html` → `textAsHtml` → `text`)
- removes styles/scripts/HTML tags
- converts `<br>` and `</p>` into line breaks
- decodes entities like `&amp;`
- collapses extra whitespace
- includes an RTF-to-text fallback (rare, but smart)

Output:
`email_text` — a clean plain-text version of the lead email.

This step is critical because your next extraction node relies on predictable text patterns.

Stage 2 — Extract the Lead Details (Structured Data)

3) “Extract Lead Info” Code Node (regex-based parsing)

Now that the email is clean, you extract key lead fields using regex.

This node:

- removes soft hyphens (`&#173;`) and `&nbsp;`
- normalizes whitespace again
- uses helper function `extract(regex)` to safely pull matches

Fields extracted:

- **name**
  From: `Customer Information ...`
- **phone**
  Any US phone pattern → converted into digits only (e.g., 8645131828)
- **email**
  Standard email regex
- **address**
  Pattern: `123 Main St, SC 29611`
- **job_number**
  Pattern: `Job #`
- **description**
  Tries to capture issue description text between sections
- **comments**
  Captures free-form notes until "View Lead Details" or footer
- **source** = `"Angi"`
- **raw_text** = the cleaned email body for fallback if extraction misses something

Why this is operator-grade:
Even if some fields fail extraction, you still pass the entire cleaned email to the AI as a backup (`raw_text`), so the workflow remains resilient.

Stage 3 — Inject Business Settings (Template-friendly)

4) "Business Info" Set Node (your template placeholders)

This step is what makes this workflow portable as a template.

It defines:

- `company_name` = `YOUR_COMPANY_NAME`
- `company_phone` = `YOUR_COMPANY_PHONE`

- `website` = `YOUR_WEBSITE`
- `service_area` = `YOUR_SERVICE_AREA`

This is where end users customize the template without touching logic.

Stage 4 — AI Agent: Creates the Responses + Checks Coverage

5) AI Agent (the "Pitbull brain")

This node instructs AI to act as a scheduling assistant.

It must:

1. Read lead details
2. Extract zip code from address
3. Check if zip is in coverage using a Google Sheet tool
4. Output:
    - SMS message (short)
    - Email message (3–5 sentences)
    - Coverage result: `"Covered"` or `"Not Covered"`

The really important part:

You are not "guessing" coverage.
The AI is required to call a tool:

6) "Check Zip Code Coverage" GoogleSheetsTool (AI tool)

This tool node gives the AI access to a **zip code list sheet**:

- "ZipCode list for Map"
- "Shared ZipCode List" tab

So the AI:

- extracts zip from address

- checks the sheet
- returns Covered / Not Covered

This is a big deal because it prevents waste:

- No sending scheduling messages to people you don't serve
- No office time wasted calling outside your area

---

7) Output Contract (Strict JSON)

The AI output must match:

```
{
  "sms_message": "...",
  "email_message": "...",
  "coverage": "..."
}
```

That strict format makes the next step predictable.

---

8) "Parse Output" Code Node

This node simply does:

- `JSON.parse($json.output)`

So now you have real fields available downstream:

- `$json.sms_message`
- `$json.email_message`
- `$json.coverage`

Stage 5 — Routing and Actions (Speed-to-Lead)

9) Switch Node (Covered vs Not Covered)

Routes based on `$json.coverage`:

Path A: “Zip Covered”

- goes to Twilio SMS node
- then to Gmail “Send a message” node

Path B: “Not Covered”

- goes to a separate Gmail node “Send a message1”

This routing prevents sending the wrong message type.

---

10) Covered Path — SMS First (fastest channel)

Twilio node: sends SMS

- `from`: placeholder `YOUR_TWILIO_SMS_NUMBER`
- `to`: extracted lead phone
- `message`: AI-generated SMS

Why SMS first:
SMS is the fastest way to “claim” the lead psychologically. They see it immediately.

---

11) Covered Path — Email Second (professional follow-up)

Gmail node sends a branded HTML email to the lead. It includes:

- Greeting using the lead name
- Confirmation you service their area
- CTA buttons:
  - “Schedule Online” (uses your `website`)
  - “Call” (uses your `company_phone`)
- Footer note explaining why they received the email (matched on Angi)

This email is intentionally polished: it makes you look organized and established.

---

12) Not Covered Path — Email Only (decline cleanly)

If Not Covered, the workflow sends a different email (still branded HTML).

Important note:
In your current template, that "Not Covered" email body text looks like it accidentally contains a contradictory sentence:
"Unfortunately We service your area and have openings available."

That line should be corrected to:

- "Unfortunately we do not service your area…"

This is a small fix but important so you don't confuse the customer.

What makes this workflow "Pitbull" (the real differentiators)

✅ Dual-trigger resilience

Two Angi email types captured.

✅ Aggressive speed-to-lead

SMS + email goes out immediately once covered.

✅ Service area protection

Coverage check prevents wasting money/time on out-of-area leads.

✅ Template-ready

End users only change:

- company name
- phone
- website
- service area text
- Twilio number
- zip code list sheet

✅ Safe fallback via raw_text

If extraction misses fields, AI still gets raw lead content.

How to Configure the Angi Lead Pitbull Automation

This workflow is designed so you **import the template** and only plug in your credentials and business info.

What You'll Need

- An **n8n account**
- A **Gmail account** receiving Angi leads
- An **OpenAI API key**
- A **Twilio account + SMS number**
- A **Google Sheet** containing service-area zip codes
- A **Facebook / Website URL** for CTAs

No coding required.

Step 1 — Import the Workflow

1. Go to **Workflows** → **Import**
2. Upload `Angi Lead Pitbull - Template.json`
3. Save the workflow
4. Leave it **OFF** for now

Step 2 — Connect Gmail Triggers

Open both Gmail trigger nodes and:

- Authenticate your Gmail account
- Confirm the sender is `no-reply@angi.com`
- Adjust subject filters if your Angi emails differ

Step 3 — Update Business Info

Open the **Business Info** node and replace:

- `YOUR_COMPANY_NAME`
- `YOUR_COMPANY_PHONE`
- `YOUR_WEBSITE`
- `YOUR_SERVICE_AREA`

No other logic changes needed.

Step 4 — Connect OpenAI

Open the AI Agent node:

- Add your OpenAI API key
- Do not modify the output format
- Do not remove the coverage check instructions

Step 5 — Configure Zip Code Sheet

1. Create a Google Sheet
2. Add one zip code per row
3. Share it with the connected Google account
4. Open the “Check Zip Code Coverage” tool node
5. Select the spreadsheet and tab

This sheet defines your service area.

Step 6 — Connect Twilio

Open the Twilio SMS node:

- Add your Twilio credentials
- Set the “From” number
- Leave message mapping unchanged

Step 7 — Review Email Templates

Open both Gmail “Send message” nodes:

- Verify wording
- Confirm links (call + schedule)
- Fix the “Not Covered” message if needed

Step 8 — Test the Workflow

Before activating:

- Send yourself a test Angi lead email
- Test one covered zip
- Test one uncovered zip
- Confirm SMS + email routing works correctly

Step 9 — Activate

Once tests pass:

- Turn the workflow **ON**
- Monitor first live leads
- Enjoy beating competitors to the punch

Optional upgrades you could add

If you want this to feel “enterprise-grade” for attendees:

1. **Add a “quiet hours” rule**
    - If lead comes in after 9pm → send SMS next morning at 8am
2. **Add dedupe protection**
    - If same phone/email arrives again within 24h → don’t spam
3. **Log to Google Sheets**
    - lead name, phone, zip, covered/not, timestamp, message sent
4. **Add “call task” for covered leads**
    - if no reply in 10 minutes → create follow-up reminder

color: #fb5607;