

THE GUIDE

The Missed-Call Money Template

Auto-text every missed call within 60 seconds so you book the job instead of losing it to the next name on Google.

9 sections

copy-paste ready

2026 edition

Owners and operators of local service businesses -- home services, salons, clinics, gyms -- and the freelancers who build automations for them.

Contents

- 01 1. Why a missed call is the most expensive thing in your business
- 02 2. How the missed-call text-back actually works (the whole flow)
- 03 3. Step 1 -- Get your text-back number (Twilio setup, ~10 minutes)
- 04 4. Step 2 -- Build the workflow in n8n (node-by-node, or import the starter)
- 05 5. The Make.com version (if you'd rather not touch n8n)
- 06 6. The text-back scripts (steal these word-for-word)
- 07 7. Your 15-minute setup checklist
- 08 8. What it actually costs (real math, no surprises)
- 09 9. Go live: compliance, gotchas, and the things that trip people up
- 10 Bonus

1. Why a missed call is the most expensive thing in your business

Here's the uncomfortable truth about a local service business: **the moment your phone rings and nobody picks up, that customer is already typing the next name into Google.** They're not going to leave a voicemail and wait. They've got a leak, a flat, a cracked screen, a special occasion, a sick pet -- they want it handled *now*, and whoever answers first usually wins the job.

Run your own numbers (be honest):

Missed calls per month = ____ (check your carrier call log or Google Business Profile)

% of those that were buyers = ~40-70% for most service businesses

Your average job / ticket = \$____

Your close rate on a live lead = ~20-40%

Monthly \$ walking out the door = (missed calls) x (% buyers) x (close rate) x (avg ticket)

Plug in a plumber with 40 missed calls, 60% buyers, a 30% close rate, and a \$350 average job: that's roughly $40 \times 0.60 \times 0.30 \times \$350 =$
 ~\$2,500/month of work that rang once and vanished. Even recovering a *fraction* of it changes your month.

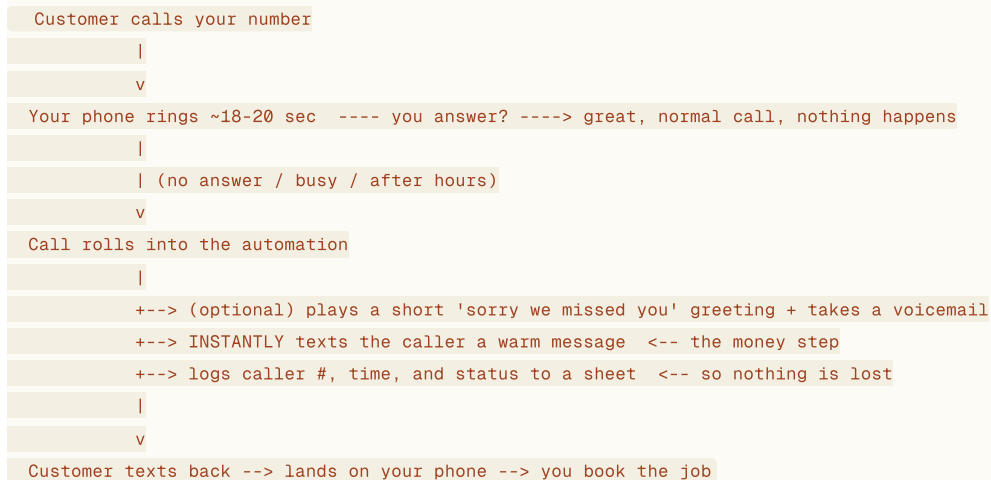
Why texting back wins where a callback doesn't:

- **Speed is the whole game.** Widely cited lead-response research finds the odds of actually connecting with a lead drop sharply the longer you wait past the first few minutes, and that the business who responds *first* wins a large share of the deals. A text that fires in 60 seconds beats a callback in 2 hours.
- **People answer texts.** They'll ignore a voicemail and dodge an unknown callback number, but they'll read and reply to a text almost immediately.
- **A text reopens the door without an awkward call.** They can tell you what they need on their own time, from the grocery store line, without having to pick up the phone again.
- **It makes you look bigger and more on-the-ball** than the competitor who let it go to a full voicemail box.

What this template does, in one sentence: the instant you miss a call, it automatically texts that caller a warm, human message that says 'sorry we missed you, here's how I can help right now' -- and logs the lead so nothing falls through the cracks. That's it. That's the whole money machine.

2. How the missed-call text-back actually works (the whole flow)

No magic, no black box. Here's the entire flow in plain English:



The three moving parts:

1. **A phone number that can text (Twilio).** This is the brain. It receives the call, notices you didn't pick up, and sends the SMS.
2. **An automation engine (n8n, or Make.com).** This is the wiring. It decides '*was this call actually missed?*' and, if so, sends the exact text you wrote and logs the lead.
3. **Your real phone.** The call still forwards to you first -- you always get the chance to answer live. The automation only kicks in when you *don't*.

Two ways to wire your phone number (pick one -- both are covered in the next section):

- **Path A -- Use a new Twilio number as your public 'call/text us' line.** Put it on your Google Business Profile, ads, van, and website. It forwards to your cell when you're available and auto-texts when you're not. Cleanest setup; you also get a textable business number as a bonus.
- **Path B -- Keep your existing number.** Turn on your carrier's *conditional call forwarding* (forward-on-no-answer and forward-on-busy) to the Twilio number. Your published number never changes; unanswered calls quietly roll into the automation.

Everything below builds this exact flow.

3. Step 1 -- Get your text-back number (Twilio setup, ~10 minutes)

Twilio is the phone-number engine. (Prefer an alternative? Telnyx, Plivo, Bandwidth, and Sinch follow the same pattern and are often cheaper per message -- the steps map over almost 1:1.)

1) Create a Twilio account. Sign up at [twilio.com](https://www.twilio.com). You'll get free trial credit. Important: a **trial account can only text numbers you've verified**, and it prepends 'Sent from your Twilio trial account' to messages. That's fine for testing -- **upgrade to pay-as-you-go before you go live** so you can text real customers.

2) Buy a local number with Voice + SMS. In the Console: *Phone Numbers* -> *Buy a number*. Filter for capabilities **Voice** and **SMS**, pick a local area code your customers recognize, and buy it (~\$1.15/month).

3) Register A2P 10DLC (US -- do NOT skip this). In the US, texting from a regular 10-digit number to consumers now requires **A2P 10DLC registration**. Unregistered texts get filtered or blocked by carriers, which quietly kills this whole system. In the Console go to *Messaging* -> *Regulatory Compliance* / *A2P 10DLC* and register:

- a **Brand** (your business info; sole proprietors qualify for a low-volume path), and
- a **Campaign** (use case: 'Customer Care' / conversational reply to inbound callers).

Cost is roughly a ~\$4 one-time brand fee + ~\$1.50/month campaign fee for the low-volume standard tier. Approval can be near-instant or take a day or two -- **start it now and build the rest while it processes**.

4) Grab your API credentials. From the Console dashboard, copy your **Account SID** and **Auth Token** -- you'll paste these into n8n (or Make) in a minute. Treat the Auth Token like a password.

5) Choose your phone path and set it up:

- **Path A (new number):** Just start using the Twilio number as your public number. We'll make it forward to your cell inside the workflow (Step 2), so live calls still reach you.
- **Path B (keep your number):** Turn on **conditional call forwarding** from your existing carrier so that *unanswered* and *busy* calls forward to your new Twilio number. The exact dial codes vary by carrier -- many US carriers use `71 + the Twilio number` to set 'forward on no answer/busy' and `73` to cancel. **Confirm the codes with your carrier**, and confirm the original caller's number passes through (it does on most US networks -- you'll verify in the test step).

6) Point the number's Voice webhook at n8n. In *Phone Numbers* -> *(your number)* -> *Voice Configuration*, set 'A CALL COMES IN' to **Webhook**, method **HTTP POST**, and paste your n8n voice webhook URL (you'll create it in Step 2 -- it looks like `https://your-n8n-domain/webhook/missed-call/voice`). Save.

That's the whole phone side. Now the wiring.

4. Step 2 -- Build the workflow in n8n (node-by-node, or import the starter)

This is the wiring. The design is two webhooks: one answers the incoming call (and rings your phone), the other runs *after* the call and decides whether to text. You can build it by hand in ~5 minutes from the node list below, or paste the import-ready starter JSON at the bottom.

The nodes (in order):

A) Webhook -- 'Incoming Call'

- HTTP Method: `POST` | Path: `missed-call/voice` | Respond: Using 'Respond to Webhook' Node
- This is the URL you paste into Twilio's 'A CALL COMES IN' field.

B) Respond to Webhook -- 'TwiML: Dial Owner'

- Respond With: `Text` | Add header `Content-Type: text/xml`
- Response Body (this rings YOUR phone first, then hands off to the status webhook):

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <Dial timeout="18" callerId="{{ $json.body.To }}"
    action="https://YOUR-N8N-DOMAIN/webhook/missed-call/status" method="POST">
    <Number>+1XXXYOURCELL</Number>
  </Dial>
</Response>
```

- `timeout="18"` = ~4-5 rings before it gives up and rolls to the automation. `+1XXXYOURCELL` = your real phone.

C) Webhook -- 'Call Status'

- HTTP Method: `POST` | Path: `missed-call/status` | Respond: Using 'Respond to Webhook' Node
- Twilio hits this automatically after the Dial finishes, sending a `DialCallStatus` field.

D) IF -- 'Was it missed?'

- Condition (String): `{{ $json.body.DialCallStatus }}` **is not equal to** `completed`
- Anything other than `completed` (`no-answer`, `busy`, `failed`) means you didn't take the call = a lead to save.

E) Twilio -- 'Send Text-Back' (on the TRUE branch)

- Credential: your Twilio (Account SID + Auth Token) | Resource: `SMS` | Operation: `Send`
- **From:** your Twilio number | **To:** `{{ $json.body.From }}` (the caller)
- **Message:** paste Script A from the Scripts section. Tip: type your business name **directly** as plain text; only the `To` field needs an expression. (Keep the message field OUT of expression mode so the `{{business_name}}` style placeholders don't get misread by n8n -- just fill them in with your real words.)

F) Respond to Webhook -- 'TwiML: Voicemail' (chained after Send Text-Back)

- Header `Content-Type: text/xml`. Plays a short greeting and records a voicemail so you get the message too:

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <Say voice="Polly.Joanna">Thanks for calling. Sorry we missed you --
  we just texted you so we can help. Leave a message after the tone if you like.</Say>
  <Record maxLength="120" playBeep="true"/>
  <Hangup/>
</Response>
```

G) Respond to Webhook -- 'TwiML: Empty' (on the FALSE branch -- call was answered, do nothing)

- Header `Content-Type: text/xml`, body: `<?xml version="1.0"?><Response></Response>`

Optional but recommended add-ons:

- **Log the lead:** add a *Google Sheets 'Append Row'* (or *Airtable*) node on the TRUE branch: timestamp `{{ $now }}`, caller `{{ $json.body.From }}`, status `{{ $json.body.DialCallStatus }}`, texted = yes. Now you have a running list of every lead the system caught.
- **Two-way replies reach a human:** add a THIRD webhook for inbound SMS (set it as the number's *Messaging* webhook in Twilio) that forwards the reply to your cell or email -- or manage the number in an app like OpenPhone. **A text-back nobody answers is worse than none.**
- **5-minute follow-up:** add a *Wait (5 min)* node then a second Twilio SMS (follow-up script) for callers who didn't reply.

Import-ready starter workflow. In n8n click *Workflows -> Import from... -> Paste* and drop this in. Then: (1) set your Twilio credential on the 'Send Text-Back' node, (2) replace `YOUR-N8N-DOMAIN`, `+1XXXYOURCELL`, and `+1XXXYOURTWILIONUMBER`, (3) edit the message text, (4) **Activate** the workflow and copy the production webhook URLs into Twilio. (Node versions evolve -- if a node imports with a warning, just open it and re-confirm the fields; the structure is correct.)

```
{
  "name": "Missed-Call Money Template",
  "nodes": [
    {
      "parameters": {
        "httpMethod": "POST",
        "path": "missed-call/voice",
        "responseMode": "responseNode",
        "options": {}
      },
      "id": "b1a1a111-1111-4111-8111-111111111111",
      "name": "Incoming Call (Twilio Voice webhook)",
      "type": "n8n-nodes-base.webhook",
      "typeVersion": 2,
      "position": [
        220,
        240
      ],
      "webhookId": "missed-call-voice"
    },
    {
      "parameters": {
        "respondWith": "text",
        "responseBody": "=<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<Response>\n  <Dial timeout=\"18\" callerId=\"{{ $jsor",
        "options": {
          "responseHeaders": {
            "entries": [
              {
                "name": "Content-Type",
                "value": "text/xml"
              }
            ]
          }
        }
      },
      "id": "b2a2a222-2222-4222-8222-222222222222",
      "name": "Twiml: Dial Owner",
      "type": "n8n-nodes-base.respondToWebhook",
      "typeVersion": 1,
      "position": [
        520,
        240
      ]
    },
    {
      "parameters": {
        "httpMethod": "POST",
        "path": "missed-call/status",
        "responseMode": "responseNode",
        "options": {}
      },
      "id": "b3a3a333-3333-4333-8333-333333333333",
      "name": "Call Status (Dial action webhook)",
      "type": "n8n-nodes-base.webhook",
      "typeVersion": 2,

```

```

    "position": [
      220,
      540
    ],
    "webhookId": "missed-call-status"
  },
  {
    "parameters": {
      "conditions": {
        "options": {
          "caseSensitive": true,
          "typeValidation": "strict"
        },
        "conditions": [
          {
            "leftValue": "={{ $json.body.DialCallStatus }}",
            "rightValue": "completed",
            "operator": {
              "type": "string",
              "operation": "notEquals"
            }
          }
        ],
        "combinator": "and"
      },
      "options": {}
    },
    "id": "b4a4a444-4444-4444-8444-444444444444",
    "name": "Was it missed?",
    "type": "n8n-nodes-base.if",
    "typeVersion": 2,
    "position": [
      520,
      540
    ]
  },
  {
    "parameters": {
      "resource": "sms",
      "operation": "send",
      "from": "+1XXXYOURTWILIONUMBER",
      "to": "={{ $json.body.From }}",
      "message": "Hi, this is Sam at AtlasWorks Plumbing -- sorry we just missed your call. We're out on a job but I can f",
      "options": {}
    },
    "id": "b5a5a555-5555-4555-8555-555555555555",
    "name": "Send Text-Back",
    "type": "n8n-nodes-base.twilio",
    "typeVersion": 1,
    "position": [
      820,
      460
    ],
    "credentials": {
      "twilioApi": {
        "id": "REPLACE_WITH_YOUR_CREDENTIAL_ID",
        "name": "Twilio account"
      }
    }
  }
}

```

```

    },
    {
      "parameters": {
        "respondWith": "text",
        "responseBody": "<?xml version='1.0' encoding='UTF-8'?>\n<Response>\n  <Say voice='Polly.Joanna'>Thanks for ca
        "options": {
          "responseHeaders": {
            "entries": [
              {
                "name": "Content-Type",
                "value": "text/xml"
              }
            ]
          }
        }
      },
      "id": "b6a6a666-6666-4666-8666-666666666666",
      "name": "TwiML: Voicemail",
      "type": "n8n-nodes-base.respondToWebhook",
      "typeVersion": 1,
      "position": [
        1080,
        460
      ]
    },
    {
      "parameters": {
        "respondWith": "text",
        "responseBody": "<?xml version='1.0' encoding='UTF-8'?>\n<Response></Response>",
        "options": {
          "responseHeaders": {
            "entries": [
              {
                "name": "Content-Type",
                "value": "text/xml"
              }
            ]
          }
        }
      },
      "id": "b7a7a777-7777-4777-8777-777777777777",
      "name": "TwiML: Empty (call was answered)",
      "type": "n8n-nodes-base.respondToWebhook",
      "typeVersion": 1,
      "position": [
        820,
        660
      ]
    }
  ],
  "connections": {
    "Incoming Call (Twilio Voice webhook)": {
      "main": [
        [
          {
            "node": "TwiML: Dial Owner",
            "type": "main",
            "index": 0
          }
        ]
      ]
    }
  }
}

```

```

    ]
  ]
},
"Call Status (Dial action webhook)": {
  "main": [
    [
      {
        "node": "Was it missed?",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Was it missed?": {
  "main": [
    [
      {
        "node": "Send Text-Back",
        "type": "main",
        "index": 0
      }
    ],
    [
      {
        "node": "Twiml: Empty (call was answered)",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Send Text-Back": {
  "main": [
    [
      {
        "node": "Twiml: Voicemail",
        "type": "main",
        "index": 0
      }
    ]
  ]
}
},
"settings": {
  "executionOrder": "v1"
}
}

```

5. The Make.com version (if you'd rather not touch n8n)

Prefer Make.com? The exact same two-webhook design ports over. You'll build two small scenarios (or one with two webhooks):

Scenario 1 -- 'Ring the owner' (the voice webhook):

1. **Custom webhook** module -- this URL goes in Twilio's 'A CALL COMES IN' field.
2. **Webhook response** module -- Status 200, header `Content-Type: text/xml`, body = the same `<Dial>` TwiML from Step 2 (rings your cell, `action` points at Scenario 2's webhook URL).

Scenario 2 -- 'Text back if missed' (the Dial action webhook):

1. **Custom webhook** module -- Twilio posts `DialCallStatus`, `From`, `To` here after the dial.
2. **Filter** on the route: only continue if `DialCallStatus` is not equal to `completed`.
3. **Twilio > Send an SMS** module -- From = your Twilio number, To = `{{From}}`, Body = Script A.
4. **Google Sheets > Add a Row** module (optional) -- log timestamp, caller, status.
5. **Webhook response** module -- `Content-Type: text/xml`, body = the voicemail TwiML (or an empty `<Response>` on the completed route).

Cost note: Make's **free tier gives 1,000 operations/month**, which comfortably covers a busy local shop (each missed call is only a few ops). If you outgrow it, the Core plan is roughly **\$9/month for 10,000 ops**. Make is the friendlier click-and-drag option; n8n is cheaper at scale and free to self-host. Either works -- **pick the one you'll actually finish today**.

6. The text-back scripts (steal these word-for-word)

Merge fields in `{{double braces}}` are fill-in-the-blanks -- replace them with your real words before you go live. Common ones: `{{business_name}}`, `{{owner_first_name}}`, `{{booking_link}}`, `{{service_area}}`, `{{quote_window}}`, `{{service_type}}`.

The 5 rules that make a text-back actually book jobs:

1. **Fire within 60 seconds.** The whole edge is speed.
2. **Sound like a human, not a robot.** Use your first name. Contractions. No 'Dear valued customer.'
3. **Ask ONE clear question.** Questions get replies; statements get ignored.
4. **Say who you are** (business name) so it's not a mystery text, and include an opt-out on first contact.
5. **Careful with links in the FIRST text** -- some carriers filter first-contact messages that contain URLs. If deliverability matters, put the booking link in message #2 and keep message #1 link-free.

Script A -- Universal / home services (the default):

Hi, this is `{{owner_first_name}}` at `{{business_name}}` -- sorry we just missed your call. We're out on a job but I can help you right here. What do you need done, and what's your zip? -- Reply STOP to opt out

Script B -- Salon / spa / clinic (appointment-driven):

Hi! This is `{{business_name}}`. Sorry we couldn't grab your call -- the front desk is with a client. Want us to book you in? Reply with a day and time that works and we'll lock it in. (Reply STOP to opt out)

(Follow-up B2, if you're confident in deliverability, add the link in message #2:) Here's our online calendar if that's easier: `{{booking_link}}`

Script C -- After-hours / closed:

Thanks for calling `{{business_name}}`! We're closed right now but we saw your call. Tell us what you need and we'll text you back first thing -- or reply with a good time to call you. (Reply STOP to opt out)

Script D -- Quote-driven trades (roofing, HVAC, plumbing, electrical):

Hi, `{{business_name}}` here -- sorry we missed you. We can usually get you a quote within `{{quote_window}}`. What's the issue and your address? We serve `{{service_area}}` and can often come out the same week. (Reply STOP to opt out)

Script E -- Gym / studio / membership:

Hey! `{{business_name}}` here -- sorry we missed your call. Looking to check us out? Reply YES and I'll text you a free pass + our hours. (Reply STOP to opt out)

Follow-up #2 -- send ~5-10 min later, ONLY if they haven't replied:

Just following up so you're not left waiting -- still happy to help. Want me to text you a quick quote or lock in a time? -- `{{owner_first_name}}`, `{{business_name}}`

Follow-up #3 -- next business day, optional, if still no reply:

Hi, it's `{{owner_first_name}}` at `{{business_name}}`. Didn't want to leave you hanging after yesterday's call. If you still need `{{service_type}}`, just reply and I'll take care of you. If not, no worries at all!

When they reply -- close fast (this is where the money is): have three canned replies ready so you can answer from a job site in one tap: (1) 'What's the address/zip?', (2) your available times or `{{booking_link}}`, (3) 'We can be out `{{day}}` -- want me to lock it in?' The template opens the conversation;

you close it.

7. Your 15-minute setup checklist

Print this. Do it in one sitting. (A2P approval may lag behind -- start it early and keep building; you can finish and test with a verified number while it processes.)

- [] **(0-3 min)** Create/log into Twilio, **upgrade off the trial**, and buy a local number with Voice + SMS.
- [] **(3-5 min)** Start **A2P 10DLC registration** (Brand + a 'Customer Care' Campaign; sole-prop low-volume tier). Kick it off and move on.
- [] **(5-6 min)** Copy your **Account SID + Auth Token** into an n8n (or Make) Twilio credential.
- [] **(6-11 min)** Build or **import the workflow**: 2 webhooks, an IF, the Twilio SMS node, and the voicemail TwiML. Paste your real cell into the `<Dial>` TwiML.
- [] **(11-12 min)** **Activate** the workflow, copy the production webhook URLs, and paste the voice URL into Twilio's 'A CALL COMES IN' field (HTTP POST).
- [] **(12-13 min)** Edit **Script A** with your real `{{business_name}}` and `{{owner_first_name}}`.
- [] **(13-14 min)** Choose your **phone path**: publish the Twilio number (Path A), or set conditional call forwarding from your existing number (Path B).
- [] **(14-15 min)** **TEST IT**. Have a friend call the number and NOT let it forward-answer. Confirm: (1) the text-back arrives, (2) the caller's number came through as `From`, (3) the lead logged (if you added logging). Then let one call through live and confirm it does NOT text (status = completed).

If A2P is still pending during your test, texts may be filtered to strangers but will still reach numbers you've verified on your Twilio account -- test with your own phone.

8. What it actually costs (real math, no surprises)

Approximate US pricing as of 2025-2026 (rates are set by the vendors and change -- verify before you rely on them):

| Item | Approx. cost |

|---|---|

| Twilio local number | ~\$1.15 / month |

| A2P 10DLC (required, US) | ~\$4 one-time brand + ~\$1.50 / month low-volume campaign |

| Outbound SMS (US) | ~\$0.0079 / segment + carrier fee ~\$0.003 |

| Inbound SMS (US) | ~\$0.0075 / segment |

| Voice, if forwarding through Twilio | inbound ~\$0.0085/min + forward leg ~\$0.014/min = ~\$0.022 / connected min |

| n8n | **FREE self-hosted** (Docker on a ~\$5-6/mo VPS) **or** n8n Cloud from ~\$20-25 / month |

| Make.com (alternative) | **FREE** up to 1,000 ops/month, then ~\$9 / month |

Worked example -- a shop with ~40 missed calls/month and ~200 forwarded call-minutes:

Number	\$1.15
A2P campaign	\$1.50
40 missed calls x 2 texts = 80 segs ...	\$0.88 (~\$0.011 each)
~60 inbound replies	\$0.45
200 forwarded minutes x \$0.022	\$4.40

Twilio total	~\$8.40 / month
+ n8n self-hosted	\$0 (or ~\$20-25 on n8n Cloud)
=====	
ALL-IN	~\$8-10/mo self-hosted ~\$30/mo on n8n Cloud

The framing that matters: the cost side is *small and knowable* -- under ten bucks a month if you self-host. The value side depends entirely on you (see the honest notes -- **we promise no jobs and no revenue**). But for most local shops, a single recovered job tends to dwarf a year of running this. The point isn't the math being magic; it's that the downside is tiny and the upside is a booked job you'd otherwise never have known about.

9. Go live: compliance, gotchas, and the things that trip people up

The 20% of details that cause 80% of the 'why isn't it working' messages:

- **A2P 10DLC is mandatory (US).** Unregistered 10-digit SMS gets filtered or blocked. Register before you depend on it. This is the #1 reason a text-back 'sometimes doesn't send.'
- **Trial-account trap.** Twilio trial only texts *verified* numbers and stamps 'Sent from your Twilio trial account' on messages. Upgrade to pay-as-you-go before go-live.
- **Opt-out handling.** Twilio auto-honors STOP/UNSUBSCRIBE on standard messaging -- keep it on, include opt-out language on first contact, and never override it.
- **TCPA, plain-English (not legal advice).** You're replying to someone who *just called you* -- that's customer-initiated and the low-risk zone. **Do not** reuse these captured numbers for cold marketing blasts; that's a different, riskier activity. When in doubt, ask a professional.
- **Caller-ID passthrough (Path B).** With carrier conditional forwarding, confirm the *original* caller's number arrives as **From** (it does on most US carriers). If you ever see your own number as the caller, your forwarding isn't passing original caller ID -- check the carrier setting.
- **Tune the timeout.** 18-20 seconds (~4-5 rings) is the sweet spot. Too long feels dead; too short and you auto-text people you could've answered live.
- **Avoid double-texting repeat callers.** If the same number calls three times in an hour, add a quick Google Sheet lookup (or check Twilio message history) to suppress duplicates within, say, 30 minutes. Optional, but repeat callers notice.
- **Make replies reach a human.** Set the number's *Messaging* webhook (or use an app like OpenPhone) so inbound replies land on your phone. The auto-text starts the conversation; a real person has to finish it.
- **Landline or VoIP business line?** Forwarding still works -- set your line's no-answer/busy forward (or your VoIP app's) to the Twilio number.

Swap-outs: Don't want Twilio? Telnyx, Plivo, Bandwidth, and Sinch all do voice + SMS with the same webhook pattern and are often cheaper. The blueprint doesn't change -- only the credentials do.

That's the whole system. Build it this week, test it with a friend's phone, and let it quietly turn your next missed call into a booked job.

Bonus

The 'first responder' cheat code: the money isn't just in the auto-text -- it's in how fast you *close* after they reply. Two moves that stack on this template: (1) Add a **Wait 5 min** -> **second text** so a busy caller who didn't reply gets one more gentle nudge (script in the Scripts section). (2) Build 3 canned replies into your phone's text shortcuts: 'What's your address/zip?'; your booking link, and 'We can be out `{{day}}` -- want me to lock it in?' The template *opens* the conversation from a job site in one tap; canned replies let you *close* it in ten seconds without stopping work. Bonus-bonus: point your Google Business Profile phone number at this line and turn on call history -- within a week you'll SEE exactly how many calls you were quietly losing, and that number tends to change behavior faster than any pep talk.

<hr>

<p class="endnote">What this is: a real, buildable automation that instantly texts back missed calls, built entirely from standard, documented Twilio + n8n (or Make.com) features. Every step here works; nothing is hypothetical.

What this is NOT: a guarantee of jobs, leads, revenue, or ROI. The dollar figures in this template (average ticket, jobs recovered, payback) are illustrative examples -- plug in your own numbers. Your results depend on your call volume, your close rate, your offer, your market, and how fast you actually follow up after the text. We make zero income claims.

About the stats: widely cited 'speed-to-lead' and 'first responder wins' figures come from third-party industry research (e.g., lead-response-time studies), included as context for why fast text-backs tend to help -- not as a promise about your specific business.

Prices are approximate as of 2025-2026 and are set by Twilio, n8n, and Make -- verify current rates before you rely on them; they change.

Compliance is on you (this is not legal advice): replying by text to someone who just called you is customer-initiated and generally low-risk, but SMS rules (TCPA in the US, plus mandatory A2P 10DLC registration) apply and vary by region. Register A2P 10DLC, keep opt-out (STOP) handling on, don't repurpose these numbers for cold marketing blasts, and confirm your own obligations with a qualified professional.

No strings: nothing here requires you to hire AtlasWorks AI or buy anything. It's yours to build and keep.</p>