

Missed Call Money Saver How It Works

CALL FLOW OVERVIEW

When someone dials your Twilio number, the Express server immediately creates a TwiML Dial verb that forwards the call to your real business line. The timeout is controlled by the `DIAL_TIMEOUT` environment variable so you can decide how long to ring before failing over.

After the dial attempt finishes, Twilio posts back to `/voice/after` with the final status. If the status indicates the call was not answered (busy, no-answer, or failed), the app kicks off the SMS follow-up workflow.

AUTOMATIC SMS CONCIERGE

The `/sms` endpoint uses a lightweight state machine stored in `lib/leadStore.js` to remember each caller. The first message asks for their name, the second asks what they need, and the third shares your Calendly scheduling link. Every reply is also written to Google Sheets so your team can see the lead history instantly.

Built-in responses make it compliant: the bot accepts `HELP` to reiterate commands and `STOP` to opt out automatically. If a lead keeps texting after the booking link is sent, they receive a friendly confirmation and another copy of the self-service booking link.

LEAD LOGGING IN GOOGLE SHEETS

The `services/sheets.js` helper connects with the Google Sheets API using a service account. Each interaction calls `upsertByPhone` to insert or update a row keyed by the caller's phone number. It tracks fields like timestamp, name, stated need, status, appointment start and end times, and the Calendly event URI so your team has a single source of truth.

CALENDLY + REVIEW AUTOMATION

When you provide a Calendly API token, the server subscribes to webhook events at `/calendly/webhook`. A new booking flips the lead status to booked and logs the appointment window; cancellations mark the row as canceled.

A cron job runs every five minutes. Two hours after an appointment end time it will text the lead a customizable review link and update the Sheet so you never send duplicate reminders.

WEB EXPERIENCE & PAYMENTS

The Express app also serves the marketing site from `/public` and exposes helper APIs. `/config` shares publishable Stripe and PayPal keys with the browser, `/api/create-checkout-session` handles the \$150/mo subscription plus one-time setup fee, and optional store routes demonstrate how to sell add-on products through Stripe Checkout.

Front-end pages like `/checkout` and `/thank-you` are static HTML files, making it easy to host the whole stack on a single Node process or behind a platform like Render, Fly.io, or