Maksym Stepanenko



AI Agents for Small Business: Automation Without a Programmer

n8n, Claude, Make.com, and how to put them to work for your business

Maksym Stepanenko · 2026

AI Agents for Small Business: Automation Without a Programmer

n8n, Claude, Make.com, and how to put them to work for your business

Maksym Stepanenko

Innova Publishing · 2026

CC BY-NC-ND 4.0

Book page, latest version and the other guides in the series:
guides.mmix.biz/en/ai-agents

CHAPTER 1

AI Agents: Why 2025 Was the Turning Point

2023 gave us ChatGPT. 2024 brought the first autonomous agents. 2025 is the year agent systems became available to small businesses without million-dollar investments or a team of developers. The technology has matured to the point where a business owner who writes no code at all can automate what two years ago required a team of five programmers.

What is an AI agent? It is a program that receives a task, plans the steps on its own, uses tools (a browser, email, a database, APIs), makes decisions, and does the work. The key difference from ordinary AI (ChatGPT in a browser) is that an agent acts autonomously, with no human involved at every step.

Three generations of automation: 1st generation (Zapier, IFTTT): “if A → then B.” Simple triggers and actions with no logic. 2nd generation (Make.com, n8n): complex workflows with conditions, loops, and data processing, but still deterministic code. 3rd generation (LLM + agent): the AI decides what to do, interprets unstructured data, and adapts to the situation. This is where we are now.

Examples of what small businesses are already doing with AI agents: an agent processes incoming email, classifies requests, answers FAQs automatically, and creates tickets for complex cases—saving 3 hours a day. An agent monitors competitors, scrapes their prices, and updates your own price list. An agent takes leads from the website, qualifies them, and books meetings in the salesperson’s calendar. An agent reads a contract, extracts the key terms, and sends a summary to the lawyer.

Two paths into agent automation: No-code/low-code (n8n, Make.com): you build visually, without code. Open to anyone. The only limit is what the built-in nodes support. Code-assisted (n8n +

JavaScript, Python scripts): you add custom logic where you need it. It takes basic skills, but it opens unlimited possibilities.

Cost: Claude API (Anthropic)—from $3 per million tokens. A typical agent uses 1,000–10,000 tokens per request, so one operation costs $0.003–$0.03. For 1,000 operations a day, that is $3–$30 a day. n8n Cloud—from $20/month. Total for a working agent system: $30–$100/month, instead of $3,000–$10,000 for an employee doing the same work by hand.

Realistic expectations: AI agents are not perfect. They get it wrong in roughly 5–15% of cases at first launch. They need iterative tuning: launch, find the errors, fix the prompts, relaunch. But even at 90% accuracy, an agent is a better deal than a person who spends 100% of their time on routine work and still makes mistakes 5–10% of the time.

CHAPTER 2

n8n: Installation, Your First Workflow, and the Core Nodes

n8n is an open-source workflow automation platform. Unlike Zapier ($20–$100+/month with hard limits), n8n can run on your own server for $5/month on DigitalOcean, or you can use n8n Cloud ($20/month with no limit on workflows). For a small business, this is a big shift in automation.

Self-hosted n8n on a VPS: buy a VPS ($5/month on DigitalOcean, Hetzner, or similar). SSH into the server. Install Docker: `curl -fsSL https://get.docker.com | sh`. Start n8n: `docker run -d --name n8n -p 5678:5678 -v ~/.n8n:/home/node/.n8n n8nio/n8n`. Open http://your-ip:5678. Create an account. Done.

n8n Cloud ($20/month): there is nothing to install. Go to app.n8n.cloud. It is good for getting started without technical skills. As you scale, it is cheaper to move to self-hosted.

Anatomy of a workflow in n8n: a Trigger node is “what starts the workflow”: Schedule Trigger (on a schedule), Webhook (an HTTP request from outside), Gmail Trigger (a new email), Telegram Trigger, Form Submission, and so on. Action nodes are “what we do next”: HTTP Request, Send Email, Google Sheets, Slack, Telegram, GPT/Claude API, Code (JavaScript). A connection is an arrow linking nodes. Data flows from node to node as JSON objects.

Your first workflow: “Email → GPT → reply.” Step 1: Gmail Trigger—fires on every new email. Step 2: Filter node—keep only emails with certain keywords (so you don’t process everything). Step 3: HTTP Request → Claude API—send the email body and get a response. Step 4: Gmail Send—reply to the sender. Result: automatic replies to FAQ emails with no human involved.

Key nodes to learn: HTTP Request—the most powerful. It calls any REST API, with parameterized URLs, custom headers, and a

body. Code node—JavaScript right inside the workflow, for data parsing, complex logic, and transformations. Set node—assigns and transforms fields. IF node—branches on a condition. Loop Over Items—loops over an array. Merge—combines data from different branches. Wait—a pause (wait N seconds or until an external event).

Credentials in n8n: n8n stores API keys and OAuth tokens securely. Create credentials for Google (Gmail, Sheets, Drive), the Telegram Bot API, the Claude or OpenAI API, Slack, Airtable, Notion, and so on. Set up once, use in every workflow.

Debugging: in n8n, every node shows its input and output data after it runs. If something goes wrong, click the node and look at what came in and what went out. Most errors come from a wrong data format or a wrong path to a field ({{$json.email}} instead of {{$json.from.email}}).

CHAPTER 3

Claude and OpenAI in Automations: How to Write Prompts

An AI agent is only as smart as its prompt. A badly written prompt means an unstable, hallucinating agent. A good prompt makes a reliable business tool that runs thousands of times in a row with minimal errors.

Claude vs. GPT-4 vs. other models: Claude (Anthropic) is the best for long documents, following instructions, and formal text. It is stronger in “safe” content and business contexts. GPT-4o (OpenAI) is an excellent balance of speed and quality, with good tooling. Gemini (Google) is fast and cheap and integrates well with Google Workspace. For most business tasks: Claude 3 Haiku (fast, cheap) for classification and simple replies; Claude 3.5 Sonnet (balance of price and quality) for core tasks; Claude Opus (expensive, best quality) for business-critical decisions.

The structure of a good business prompt: [ROLE] You are an assistant for Acme Inc., a company specializing in IT services. You answer only questions related to our services. [TASK] Your task: classify the customer’s incoming request and either answer it directly or escalate it to a manager. [CONTEXT] Our services: [list]. Our prices: [list]. Business hours: [details]. [USER DATA] Request: {{$json.message}} [RESPONSE FORMAT] Reply strictly in JSON format: {"classification": "faq|escalate|spam", "response": "...", "reason": "..."}

Why JSON output is critical for automations: if the AI returns free text, it is hard to parse in the next node. JSON lets you take a specific field ($json.classification), use it in IF nodes, and pass it to other systems. Add this to every prompt: “Respond ONLY in JSON format. No explanations, no markdown, just JSON.”

System prompt vs. user prompt: the system prompt holds stable instructions (role, rules, format) and rarely changes. The user

prompt holds dynamic data (the current email, the user's question) and changes with every request. In an API call: messages: [{role: "system", content: "system prompt"}, {role: "user", content: "data"}].

A few examples of effective prompts:

Email classification: "Classify the email into categories: SUPPORT (technical question), BILLING (payment question), SALES (interest in buying), SPAM. Data: Subject: {{subject}}, Body: {{body}}. Reply in JSON: {"category": "...", "priority": "high|medium|low", "summary": ""}"

Extracting data from a contract: "Extract the following fields from the contract: parties, subject matter, amount, term, termination conditions. Contract text: {{contract_text}}. Reply in JSON."

Generating a product description: "Create an SEO-optimized product description for an online store. Tone: professional, persuasive. Length: 150–200 words. Include: 3 key benefits and a call to action. Product specifications: {{product_data}}."

Temperature and stability: for data extraction and classification, use temperature 0 (as deterministic as possible). For text generation, use temperature 0.7. For creativity, use temperature 1.0. Business automations mostly use temperature 0–0.3.

CHAPTER 4

10 Automations That Change a Business

Theory is good. Here are specific automations that already work in real small businesses and save dozens of hours a week.

Automation 1: Lead generation and qualification. Trigger: a form on the website (Webhook). Actions: Claude qualifies the lead from the fields filled in. If high priority → an instant Telegram notification and a task created in the CRM. If medium → an email offering a call. If spam → archive. Savings: 1–2 hours of qualification a day.

Automation 2: Smart customer support. Trigger: a new ticket in Zendesk or email. Actions: Claude reads the ticket and looks for an answer in the knowledge base (Notion or Google Docs via API). If an answer is found, it replies automatically. If not, it summarizes and prioritizes the ticket for a human agent. Savings: 50–70% of tickets are resolved without a person.

Automation 3: Weekly competitive analysis. Trigger: Schedule Trigger (every Sunday). Actions: HTTP Request → competitor websites (scraping prices, news, job postings). Claude analyzes the week's changes, generates a report, and sends a PDF by email. Savings: 4–8 hours of manual monitoring.

Automation 4: Content machine. Trigger: Google Sheets (a new row with a topic). Actions: Claude writes a draft article. An image is generated through DALL-E or the Stable Diffusion API. The post goes to WordPress or Ghost via API. A tweet is scheduled in Buffer. A notification goes to Telegram. Savings: 2–4 hours per article.

Automation 5: Smart invoice processing. Trigger: Gmail—an email with a PDF attachment. Actions: extract the PDF. Claude reads the invoice and pulls out the supplier, amount, date, and number. It compares them with the order in the system. If everything matches, it adds the invoice to Google Sheets. If there is a discrepancy, it notifies the accountant. Savings: 30 minutes

per incoming invoice.

Automation 6: Onboarding new clients. Trigger: a new row in the CRM (a new client). Actions: a personalized welcome email (Claude generates it from the client's data). Tasks created for the team in Asana or Trello. A welcome call scheduled. A check-in email after 7 days. Savings: 45 minutes of manual onboarding per client.

Automation 7: Review monitoring. Trigger: Schedule (every 4 hours). Actions: scrape Google Reviews, Yelp, and Trustpilot for your company. Claude analyzes the sentiment of each new review. A negative review triggers an immediate alert and a reply template. A positive one is added to a list for reposting on social media. Savings: continuous monitoring at no extra cost.

Automation 8: Smart email newsletter. Trigger: an RSS feed of news in your industry (weekly). Actions: Claude reads 20–30 news items and picks the top 5 most relevant. It writes a short summary of each with your commentary, assembles the newsletter, and sends it through Mailchimp. Savings: 3–4 hours of newsletter preparation.

Automation 9: Sales analytics. Trigger: Schedule (every morning). Actions: pull data from Stripe or WooCommerce. Claude analyzes which products are growing, which are declining, and any anomalies. It generates a 5-line morning report and sends it to Telegram. Savings: 1 hour of daily manual analytics.

Automation 10: Automatic follow-up. Trigger: CRM—a lead has not replied in 3 days. Actions: Claude writes a personalized follow-up based on the original request, the company profile (LinkedIn), and previous communications. It sends the message and logs it in the CRM. If there is still no reply after 5 days, it creates a task. Savings: 90% of follow-ups are automated.

CHAPTER 5

Building an AI Agent for Customer Support

A customer support agent is the most popular first AI agent for a small business. Properly set up, it handles 60–80% of incoming requests without a human, works 24/7, and never gets tired.

Support agent architecture: Input channel (email / Telegram / chat widget / WhatsApp). Classifier (Claude)—determines the type and priority. Knowledge base (Notion / Google Docs / Airtable). Response generator (Claude)—creates a personalized reply. Escalation (a Telegram or Slack notification to a human when needed). Logger (Google Sheets / CRM)—records every interaction.

Step 1: The knowledge base. Create a document (Google Doc or Notion) with: an FAQ (the 50–100 most common questions and answers). Policies (returns, shipping, warranty). A price list. Service descriptions. Contacts and business hours. This document is the “brain” of your agent. You can update it without reprogramming the agent.

Step 2: The classifier prompt. “You classify customer requests. Categories: FAQ (the answer is in the knowledge base), TECHNICAL (needs a technical specialist), BILLING (payment or refund question), URGENT (complaint or unhappy customer), OTHER. Customer request: {{message}}. Reply in JSON: {"category": "...", "confidence": 0.0-1.0, "summary": "...", "language": "..."}”

Step 3: Searching the knowledge base. There are two approaches. The simple one is to pass the entire knowledge base into the prompt (works for bases up to 20,000 words and costs $0.03–$0.10 per request). RAG (Retrieval-Augmented Generation) uses vector search for relevant fragments (harder to set up, but cheaper and it scales to large bases).

Step 4: The response generator prompt. “You are a polite assistant for [Company name]. You answer customer requests based on the information provided. Rules: answer only from the knowledge base. If the information is not there, say you will check and get back to them. Tone: professional, warm, brief. Knowledge base: {{kb_content}}. Customer request: {{message}}. Conversation history: {{history}}. Generate a reply to the customer.”

Step 5: Escalation conditions. Escalate to a human if: the classifier’s confidence is <0.7. The category is URGENT. The customer is clearly upset (sentiment analysis). This is already the third message in a row without a resolution. The customer asks to “talk to a human.”

Step 6: Multilingual support. Claude works very well with Russian, Ukrainian, English, and French. Add this to the prompt: “Reply in the same language as the customer.” The classifier returns a language field for analytics.

Iterative improvement: for the first 2 weeks, monitor every reply the agent sends. Mark the errors. Improve the prompts. Add to the knowledge base the questions the agent could not answer. After a month, most typical tasks are handled accurately. After three, the system runs stably.

CHAPTER 6

Make.com: Fast Automation Without a VPS

Make.com (formerly Integromat) is a powerful no-code automation platform. Unlike n8n, Make.com runs as SaaS: there is nothing to install, and everything happens in the browser. The free plan (1,000 operations/month) is enough for testing. The paid plan starts at $9/month for serious use.

Why Make.com is sometimes better than n8n: the GUI is more intuitive for non-technical users. It has 1,500+ built-in integrations (more than n8n). It has a better debugging UI. You need no VPS (everything is in the cloud). When n8n is the better choice: self-hosting, unlimited scaling, custom JavaScript nodes, and savings ($5/month for a VPS vs. $9+/month for Make).

Anatomy of a Make.com scenario: Watch module—the trigger (the equivalent of a Trigger node in n8n). Action modules—the actions (Search, Create, Update, Send, and so on). Iterator—loops over an array. Router—branching. Aggregator—combines the results from an iterator.

A practical example: “A Telegram bot for your business.” A user writes to the Telegram bot. The Make scenario receives the message through Telegram Watch, sends it to the OpenAI or Claude API, returns the answer to Telegram, and logs it in Google Sheets. The whole scenario takes 30 minutes to build, with no code.

HTTP requests in Make.com: any API call goes through the HTTP module: method (GET/POST/PUT), URL, headers, body. JSON parsing is built in. It is ideal for the Claude API, the Notion API, and any custom systems.

Errors and retry logic: Make.com automatically retries failed operations (configurable). You can set how many times to retry, the pause between attempts, and what to do after a final failure (a

notification by email or Slack).

Data and storage: there is no built-in database. To keep data between scenarios, use: Google Sheets (the simplest option). Airtable (a table-based database). A Notion database. Make Data Store ($9/month extra, limited volume). MySQL/PostgreSQL through the HTTP module.

Make.com + n8n: you don’t have to pick one. Many people use Make.com for simple integrations (Shopify → CRM, Calendar → Slack) and n8n for complex AI agents. Different tools for different jobs.

Limits and pricing: an operation is each module in a scenario. A scenario with 10 modules is 10 operations. 10,000 operations a day costs about $16/month (the Core plan). For agents processing 500+ requests a day, self-hosted n8n is cheaper.

CHAPTER 7

Security, Privacy, and GDPR

When you automate business processes with AI, you inevitably process customer data through third parties. That creates data protection obligations you cannot ignore.

What happens to your data when you call the Claude API: the data is sent to Anthropic's servers (in the US). Anthropic does not use API requests to train its models (by default). Anthropic may store requests for up to 30 days for safety and compliance. Important: if you process the data of EU citizens, you need to sign a Data Processing Agreement (DPA) with Anthropic.

GDPR and Canada's PIPEDA: if your customers are EU residents, the GDPR applies no matter where your business is located. PIPEDA (the Canadian counterpart) applies to all Canadian businesses. Key requirements: data minimization (pass the AI only what is needed, and anonymize where possible). Notice (the customer must know that an AI is processing their data). Consent (for processing special categories of data). A Data Processing Agreement with every subprocessor.

Anonymization before sending to the AI: if you process personal data, add an anonymization step to the workflow. Replace names with [CLIENT], emails with [EMAIL], and addresses with [ADDRESS]. In most cases the AI handles the task without the personal details. This reduces risk and simplifies compliance.

API keys and how to store them: never hardcode them in a workflow. In n8n: the Credentials section (encrypted storage). In Make.com: Connections (similar). Environment variables in .env files for self-hosted n8n. Key rotation: change API keys every 6 months, and immediately when an employee who had access leaves.

Logging and auditing: log every action your agents take. At minimum: the date and time of the operation, the request type, the

result (success/fail), and the client ID (without personal data). Store logs in Google Sheets, Airtable, or any database. Retention period: 12 months (usually enough for compliance).

Human-in-the-loop: for business-critical decisions (finance, legal documents, medical data), the agent should PROPOSE a decision and a human should CONFIRM it. Never fully automate: deletion of client data. Financial transactions >$1,000. Legal communications. Messages where a mistake could harm the client.

Testing before launch: run the agent on test data (not real customers). Check 50+ scenarios, including edge cases. Simulate bad input: what does the agent do with an empty request? An aggressive request? A request in several languages? Prompt injection attempts?

CHAPTER 8

Measuring the ROI of AI Automation

A business invests in automation to earn more or spend less. Without measuring results, you cannot know whether the system works. Here is how to calculate the ROI of AI agents.

Baseline: measure before you automate. How many hours a week does the task take? What rate does the person doing it earn? How many units are processed per day? What is the error rate (quality)?

Cost of automation: direct costs (API cost + platform + VPS). Indirect costs (time for setup and maintenance). For a typical email-processing agent: Claude API: $15/month. n8n Cloud: $20/month. Setup (one time): 10 hours @ $50 = $500. Maintenance: 2 hours/month @ $50 = $100/month. Total for the first month: $635. Following months: $135/month.

Benefit measurement: the agent processes 200 emails a day, while a person processed 50 a day in 4 hours. The agent needs 0 person-hours. Savings: 4 hours/day × 22 working days × $25/hour = $2,200/month. ROI in the first month: ($2,200 - $635) / $635 = 246%. ROI in the second month: ($2,200 - $135) / $135 = 1,530%.

Qualitative metrics: response time (from 4 hours to 2 minutes). 24/7 coverage (the agent works when the office is closed). Consistency (the agent gives equally good answers on Monday morning and Friday evening). Customer satisfaction: compare CSAT before and after.

When automation does NOT pay off: the task happens only 1–2 times a month. The task requires empathy and human judgment that AI cannot reproduce. The volume is too small to pay back the setup. Regulatory requirements oblige you to keep a human in the loop.

A dashboard for monitoring agents: create a Google Sheet or Airtable with: the number of operations per day. Success rate (% without errors). API cost (from the Anthropic dashboard).

Processing time. Escalations (% of requests handed to a human). Update it automatically through n8n (an agent that monitors the other agents).

Iterative improvement: week 1: launch and record every error. Weeks 2–3: fix the prompts and improve the knowledge base. Week 4: measure accuracy again. Month 2: add new scenarios and widen coverage. Quarter 1: evaluate ROI and decide on scaling. Most agents reach a working accuracy of 90%+ by the end of the second month.

CHAPTER 9

Roadmap: From Your First Automation to a Full System

Business automation is not a one-time project but a continuous process. Here is a 12-month roadmap that will help you automate your business step by step, without chaos.

Month 1—Foundation: Install n8n or create a Make.com account. Get Claude and OpenAI API keys. Write down your 3 most painful manual tasks (how much time they take, who does them, how critical they are). Fully automate one of them. Measure the result.

Months 2–3—The core of the system: Automate the handling of incoming requests (email/chat/Telegram). Build a knowledge base for the support agent. Connect a CRM (if you don't have one, Airtable at the minimum). Automate lead qualification and routing.

Months 4–6—Expansion: Content automation (draft generation, social posts). Analytics: morning reports without manual data gathering. Follow-up sequences. Competitor monitoring.

Months 7–9—Integration: Link all the automations into a single system. CRM → Marketing → Support → Analytics—one flow of data. A customer journey map with automation points at every step. A human oversight dashboard.

Months 10–12—Optimization: Audit all agents—what works well and what produces errors. Update prompts for new model versions. Add voice agents (Twilio + Whisper + TTS). Run first experiments with multi-agent systems (agents that call other agents).

What NOT to automate: relationships with key clients—personal contact is essential. Crisis communications—they require human judgment. Strategic decisions—AI helps with analysis, but the decision belongs to a person. Hiring and HR communications—they require real humanity.

Automation culture in the team: the main problem is not technical but human. The team fears the agents will replace them. The right framing: agents take the routine, and people do the interesting work. Involve the team in deciding what to automate—they know best where the routine is. Show that the agent helps rather than replaces: start with 1–2 people for whom the agent saves an hour a day. They become your internal automation “champions.”

Bottom line: after 12 months of doing this right, a typical small business with a team of 3–10 people has: 15–20 active automations. 30–60 person-hours saved per week. 24/7 customer service without additional hires. A competitive advantage over companies that still do everything by hand.

What next

Thank you for reading “AI Agents for Small Business: Automation Without a Programmer”.

Updates to this book, EPUB and PDF files and every guide in the series are on the book page: guides.mmix.biz/en/ai-agents

Maksym Stepanenko, Innova Publishing.