

AI Automation / AI Agents

Design an agent that reads incoming emails, drafts responses, and sends after approval — saves 10+ hours per week.

Difficulty: Intermediate

Model: GPT-4 / Claude / Gemini

Use Case: Email Management, Customer Communication, Support Automation

Updated: May 2026

Why This Prompt Exists

Email is a productivity black hole. Knowledge workers spend 2-3 hours daily on email. An AI agent can draft responses — but needs to sound like you, not a robot.

You get:

- responses that sound generic and robotic (damaging relationships)
- agents that miss context from previous emails in the thread
- drafts that need heavy editing (wasting more time than they save)
- no handling of sensitive or complex emails (escalation missing)
- privacy concerns (email content exposed to AI)

But email agents can be effective:

- classification: what type of email is this? (inquiry, complaint, thank you, spam)
- thread awareness: what's already been discussed?
- tone matching: how does this person usually write?
- knowledge base: company policies, product info, past resolutions
- approval workflow: draft → human review → send

Without design, email agents create more work than they save.

This prompt designs effective email response automation.

The Prompt

Assume the role of an email automation architect who designs AI response agents.

Your task is to create an agent that reads emails and drafts responses.

Generate:

1. EMAIL TYPES TO HANDLE

- Types the agent will handle: [e.g., "general inquiries, scheduling requests, FAQ answers"]
- Types that always escalate: [e.g., "legal threats, executive emails, confidential info"]
- Types to ignore: [e.g., "newsletters, spam"]

2. CLASSIFICATION PROMPT

- How the agent categorizes incoming emails
- Confidence threshold for escalation

3. THREAD AWARENESS

- How far back to read (last 3 messages / entire thread)
- What to track: [previous answers, unresolved questions, tone shifts]

4. RESPONSE GENERATION RULES

- Tone: [professional / friendly / concise / match sender's tone]
- Structure: [acknowledgment + answer + next steps + closing]
- Signature: [use standard signature or none]
- Variables to include: [customer name, order number, ticket ID]

5. KNOWLEDGE SOURCES

- Company policies (where to find them)
- Product documentation
- Past successful responses (examples)
- Do not use: [sources to exclude]

6. APPROVAL WORKFLOW

- Auto-send for: [low-risk email types, confidence > 0.9]
- Human review for: [medium-risk, confidence 0.6-0.9]
- Escalate for: [high-risk, confidence < 0.6]
- Where drafts go: [Gmail draft folder, Slack channel, internal tool]

7. PRIVACY & COMPLIANCE

- Data handling rules (don't store emails longer than needed)
- Opt-out mechanism (users can request human-only responses)
- Audit log requirements

8. READY-TO-USE AGENT PROMPT

- The system prompt for the email agent

INPUTS:

Email volume (daily):

[E.G., "200 emails/day, mostly customer support"]

Email types (examples):

[PASTE 10-20 EXAMPLE EMAILS]

Preferred tone:

[E.G., "Friendly but professional, like a helpful colleague"]

Sensitive topics to escalate:

[E.G., "Legal, security, account deletion"]

RULES:

- Never auto-send for sensitive topics (legal, security, executive communication)
- Always include human approval for first week of deployment
- Log every draft for quality review (sample 5-10% daily)
- Train classification on real emails, not synthetic data
- Include an unsubscribe/opt-out for AI responses
- Comply with data privacy regulations (GDPR, CCPA) for email content

How To Use It

- Start with auto-send disabled — review every draft for the first week to catch issues.
- Train classification on 100+ real emails before going live.
- Start with low-risk email types (general inquiries, FAQs) before expanding.
- Log all drafts for quality review — sample 5-10% daily.
- Include an opt-out mechanism for customers who prefer human-only responses.

Example Input

Email volume:

"150 emails/day — customer support for SaaS product"

Email types:

"Password reset requests, billing questions, feature requests, bug reports"

Preferred tone:

"Friendly and helpful, not robotic. Use customer's name when available."

Sensitive topics to escalate:

"Account deletion, data export requests, legal complaints"

Why It Works

Most email automation attempts fail because they ignore thread context, miss sender tone, or produce robotic responses.

This framework improves outcomes by forcing:

- classification (what type of email is this?)
- thread awareness (what's already been said?)
- tone matching (how should this response sound?)
- approval workflow (who reviews before sending?)
- privacy compliance (how is email data handled?)

Great email agents don't replace humans — they draft responses so humans can review and send with one click.

Build Better AI Systems

Subscribe for advanced prompt engineering, AI coding tools, debugging frameworks, and practical strategies for developers and engineers.

Carefully engineered prompts for people doing real work.

Share this:

- [Share on Facebook \(Opens in new window\) Facebook](#)
- [Share on X \(Opens in new window\) X](#)

See also [Data Analysis Agent](#)