

Building Oscar

A Step-by-Step Guide to Building Your AI Email Assistant

Build your own AI email assistant using Claude.

This guide walks you through every step from installation to customization.

Updated: This version includes the complete prompt for Section 4: Create Gmail Tools, with detailed tool descriptions and parameters that were missing from the original guide.

Presented by Aloa



Section 4: Create Gmail Tools

Adding tools that let Oscar read, search, and send emails.

What we're doing

Claude needs tools to interact with Gmail. These tools define what actions Oscar can take. Good tool descriptions are critical - they help Claude choose the right tool for each request.

The Prompt

Copy and paste this into Claude Code:

```
Create the Gmail tools for Oscar in packages/backend/src/agent/tools/:
```

```
1. packages/backend/src/agent/tools/listEmails.ts
- Tool name: "list_emails"
- Description: "List emails from the user's inbox with optional
filtering. Use this when the user asks about their emails, inbox,
or wants to see recent messages. Returns subject, sender, date,
and snippet for each email."
- Parameters:
* maxResults (number, optional, default 10): Max emails to return
* labelIds (array of strings, optional): Filter by labels
like ["INBOX", "UNREAD"]
* query (string, optional): Gmail search query for filtering
```

```
2. packages/backend/src/agent/tools/searchEmails.ts
- Tool name: "search_emails"
- Description: "Search emails using Gmail search syntax. Use when
user wants to find specific emails by sender, subject, content,
or date. Examples: 'from:john@example.com', 'subject:meeting',
'is:unread after:2024/01/01', 'has:attachment filename:pdf'.
DO NOT use for general 'show my emails' - use list_emails instead."
- Parameters:
* query (string, required): Gmail search query
* maxResults (number, optional, default 20): Max results
```

```
3. packages/backend/src/agent/tools/readEmail.ts
- Tool name: "read_email"
- Description: "Read the full content of a specific email by its ID.
Use this when the user wants to see the complete text of an email,
not just the preview snippet. Returns subject, sender, recipients,
date, labels, and full body."
- Parameters:
* emailId (string, required): The unique ID of the email to read
```

Section 4: Create Gmail Tools (continued)

```
4. packages/backend/src/agent/tools/sendEmail.ts
- Tool name: "send_email"
- Description: "Send an email on behalf of the user. Use this when
the user explicitly asks to send, compose, or reply to an email.
IMPORTANT: Always confirm the recipient, subject, and content
with the user before sending."
- Parameters:
* to (string, required): Recipient email address
* subject (string, required): Email subject line
* body (string, required): Email body content (plain text)
* cc (string, optional): CC recipients (comma-separated)
* bcc (string, optional): BCC recipients (comma-separated)
* replyToMessageId (string, optional): Message ID for threading
```

```
5. packages/backend/src/agent/tools/index.ts
- Export all tools as an array of GmailTool objects
- Each tool needs: name, description, schema (Zod), execute function
```

Then update packages/backend/src/routes/chat.ts to import and register
these tools with createSdkMcpServer.

Use tool() from @anthropic-ai/claude-code and zod for validation.
Each tool should accept a Gmail client and return a JSON string.

Tools Reference

Tool Name	When to Use
list_emails	"Show my emails", "what's in my inbox"
search_emails	"Emails from John", "unread from last week"
read_email	"Read that email", "what does it say"
send_email	"Send an email to...", "reply to..."

Why Descriptions Matter: Claude decides which tool to use based on descriptions.
Be specific about when to use each tool. Notice how search_emails says "DO NOT use
for general requests" - this prevents Claude from choosing the wrong tool.

Checkpoint

```
# Stop the running backend (Ctrl+C) then:
npm run dev:backend
# Should see: Oscar backend running on port 3001
# No errors about missing tools or imports
```

Troubleshooting

"Cannot find module" → Check file paths and .ts extensions

"tool is not a function" → Import from '@anthropic-ai/clause-code'

Zod errors → Make sure schema types match your parameters