

BRIEFED

Build Your AI Client Research Agent

Automatically research every client before you meet them. Brief delivered to your phone. Google Doc created. All before you say hello.

By Miss AI · realmissai.com

Follow this guide to build it yourself.

Rather have it built for you? Visit aiconsultancy.co.nz

WHAT IS BRIEFED

The Problem It Solves

If you run sales calls, discovery calls, or client meetings - you know the pain. You're Googling the person 10 minutes before the call. Skimming their website. Trying to remember what they wrote in the booking form. Half the time you walk in underprepared and the call falls flat.

Briefed fixes this completely. It's an AI agent that activates the moment someone books a call with you. It researches their business, generates a full prep note, sends it to your phone via Telegram, and creates a Google Doc - automatically.

- Every client. Every call. Fully researched before you say hello.

The Full Flow

1

Client books your Calendly

They fill in company, website, and their biggest challenge

2

Webhook fires instantly

Calendly sends their data to your Briefed agent

3

Agent scrapes their website

Reads and analyses their business content

4

Claude generates prep note

7 sections: summary, pain points, solutions, red flags and more

5

Telegram delivers the brief

Full prep note lands on your phone within 60 seconds

6

Google Doc is created

Auto-saved to your Drive for reference during the call

PART 1

Before You Start

What You Need

- ✓ **Calendly account** - Free plan works - calendly.com
- ✓ **Vercel account** - Free plan works - vercel.com
- ✓ **GitHub account** - Free - github.com
- ✓ **Anthropic API key** - Pay per use, very cheap - console.anthropic.com
- ✓ **Telegram account** - Free - telegram.org
- ✓ **Google account** - For Drive and Docs - gmail.com
- ✓ **VS Code** - Free code editor - code.visualstudio.com
- ✓ **Claude Code** - AI coding assistant - run: `claude` in terminal
- ✓ **Node.js** - Free - nodejs.org

■ You don't need to know how to code. Claude Code writes everything for you. You just need to follow the steps accurately and copy things correctly.

Set Up Your Calendly Questions

Add these three questions to your booking form so Briefed has something to research:

- "What is your company or business name?"
- "Please share your website and/or social media links"
- "What do you need help with? What's your biggest challenge right now?"

In Calendly: Click your event type → Edit → Questions → Add each as required.

PART 2

Set Up Your Project

Step 1 - Create a GitHub Repository

Go to github.com → click + → New repository → name it briefed-agent → set to Public → Create repository.

Step 2 - Clone and Open in VS Code

```
# Run these in your Terminal (Mac: Cmd+Space → Terminal)
git clone https://github.com/YOUR_USERNAME/briefed-agent.git
cd briefed-agent
code .
npm init -y
npm install googleapis node-fetch
```

Step 3 - Create vercel.json

Create a file called vercel.json in the root folder:

```
{
  "version": 2,
  "functions": {
    "api/**/*.js": {
      "maxDuration": 30
    }
  }
}
```

Step 4 - Create .gitignore

Create a file called .gitignore - this protects your secret keys:

```
node_modules/
.env
.env.local
google-credentials.json
.DS_Store
```

PART 3

Get Your API Keys

Anthropic API Key

console.anthropic.com → API Keys → Create Key → name it Briefed → copy the key (starts with sk-ant-)

Calendly API Key

calendly.com → profile → Integrations → API & Webhooks → Create token → enable: all Scheduling scopes, webhooks:read, webhooks:write, users:read → copy immediately (shown once only)

Telegram Bot Token

Open Telegram → search @BotFather → type /newbot → follow prompts → copy the token it gives you → open your new bot and press Start

Telegram Chat ID

In Telegram search @RawDataBot → press Start → look for "id" under "chat" in the JSON response → copy that number

Google OAuth Credentials

console.cloud.google.com → New project (name: Briefed) → APIs & Services → Library → enable Google Drive API and Google Docs API → Credentials → Create Credentials → OAuth 2.0 Client ID → Web application → add redirect URI → download JSON

Create Your .env File

Create a file called .env in your project root:

```
ANTHROPIC_API_KEY=your_anthropic_key_here
CALENDLY_API_KEY=your_calendly_key_here
CALENDLY_WEBHOOK_SECRET=add_after_webhook_setup
TELEGRAM_BOT_TOKEN=your_telegram_bot_token
TELEGRAM_CHAT_ID=your_telegram_chat_id
GOOGLE_CLIENT_ID=from_your_oauth_json
GOOGLE_CLIENT_SECRET=from_your_oauth_json
GOOGLE_REFRESH_TOKEN=get_in_next_step
GOOGLE_SHARE_EMAIL=your_google_email@gmail.com
```

PART 4

Get Your Google Refresh Token

This is a one-time step that gives Briefed permanent access to create Google Docs in your Drive. Do this once and you never have to touch it again.

Step 1 - Generate Auth URL

```
# Run in your terminal - replace with your actual values
node -e "
const {google} = require('googleapis');
const oauth2Client = new google.auth.OAuth2(
  'YOUR_GOOGLE_CLIENT_ID',
  'YOUR_GOOGLE_CLIENT_SECRET',
  'https://your-vercel-url.vercel.app/api/oauth/callback'
);
const url = oauth2Client.generateAuthUrl({
  access_type: 'offline',
  scope: [
    'https://www.googleapis.com/auth/drive.file',
    'https://www.googleapis.com/auth/documents'
  ]
});
console.log('Visit this URL:', url);
"
```

Step 2 - Authorise Access

- Copy the URL printed in terminal
- Paste it into your browser
- Log in with the Google account you want docs saved to
- Click Allow
- You'll be redirected to a URL showing an error - that's fine
- Copy everything in the URL after code= and before &scope;

Step 3 - Exchange for Refresh Token

```
node -e "  
const {google} = require('googleapis');  
const oauth2Client = new google.auth.OAuth2(  
  'YOUR_CLIENT_ID', 'YOUR_CLIENT_SECRET',  
  'https://your-vercel-url.vercel.app/api/oauth/callback'  
);  
oauth2Client.getToken('PASTE_YOUR_CODE_HERE')  
  .then(r => console.log(JSON.stringify(r.tokens, null, 2)));  
"
```

Copy the refresh_token value and add it to your .env as GOOGLE_REFRESH_TOKEN.

PART 5

Build the Agent with Claude Code

Open Claude Code in your VS Code terminal by typing `claude` and pressing enter. Then paste this exact prompt:

PASTE THIS PROMPT INTO CLAUDE CODE (Part 1):

Build a Vercel serverless function at `api/briefed-webhook.js`

STEP 1 - Receive Calendly webhook:

- Accept POST requests, parse JSON body
- Extract: name, email, timezone from `body.payload`
- Extract: eventName, startTime, endTime, meetingLink from `body.payload.scheduled_event`
- Extract from `questions_and_answers` array:
 - company (question contains "company")
 - website (question contains "website" or "social")
 - biggestChallenge (question contains "help" or "challenge")

STEP 2 - Research their website:

- Fetch the client website URL with 8 second timeout
- Strip all HTML tags, scripts, and styles
- Keep first 3000 characters of clean text

PASTE THIS PROMPT INTO CLAUDE CODE (Part 2):

```
STEP 3 - Generate prep note with Claude API:  
- POST to https://api.anthropic.com/v1/messages  
- Model: claude-sonnet-4-20250514, max_tokens: 1000  
- System prompt: "You are a meeting prep assistant.  
Generate a 7-section prep note:  
1.Business Summary 2.Industry and Market  
3.Likely Pain Points 4.Suggested Talking Points  
5.Questions to Ask 6.Relevant AI Solutions  
7.Red Flags. Be direct and specific."  
  
STEP 4 - Send Telegram: POST to Telegram API  
- Use TELEGRAM_BOT_TOKEN and TELEGRAM_CHAT_ID env vars  
- Include name, company, website, challenge,  
meeting time, join link, and full prep note  
  
STEP 5 - Create Google Doc using googleapis OAuth2  
- Use GOOGLE_CLIENT_ID, GOOGLE_CLIENT_SECRET,  
GOOGLE_REFRESH_TOKEN env vars  
- Title: "Briefed - [Name] - [Date]"  
- Share with GOOGLE_SHARE_EMAIL env var  
  
STEP 6 - await ALL steps BEFORE returning 200 OK  
- Add console.log after every step for debugging  
- Wrap everything in try/catch blocks  
- Add: module.exports = handler  
- Add: handler.config = {api:{bodyParser:false}}
```

- Let Claude Code write the full file. If it asks questions, answer them. Once done, review the file and move to the next step.

PART 6

Deploy to Vercel

Step 1 - Push to GitHub

```
git add .
git commit -m "Add Briefed agent"
git push origin main
```

Step 2 - Connect to Vercel

- Go to vercel.com and log in
- Click Add New → Project
- Import your briefed-agent GitHub repository
- Click Deploy
- Note your live URL (e.g. briefed-agent.vercel.app)

Step 3 - Add Environment Variables to Vercel

In Vercel → your project → Settings → Environment Variables. Add all 9 keys from your .env file:

KEY	WHERE TO GET IT
<code>ANTHROPIC_API_KEY</code>	console.anthropic.com
<code>CALENDLY_API_KEY</code>	calendly.com → Integrations
<code>CALENDLY_WEBHOOK_SECRET</code>	After webhook setup (next step)
<code>TELEGRAM_BOT_TOKEN</code>	From @BotFather
<code>TELEGRAM_CHAT_ID</code>	From @RawDataBot
<code>GOOGLE_CLIENT_ID</code>	Google Cloud Console
<code>GOOGLE_CLIENT_SECRET</code>	Google Cloud Console
<code>GOOGLE_REFRESH_TOKEN</code>	From Step 4 of this guide
<code>GOOGLE_SHARE_EMAIL</code>	Your Google email address

PART 7

Set Up the Calendly Webhook

Now we tell Calendly where to send booking data. Run these commands in your terminal one at a time.

Step 1 - Get Your Organisation URL

```
curl --request GET \  
  --url https://api.calendly.com/users/me \  
  --header 'Authorization: Bearer YOUR_CALENDLY_API_KEY'  
  
# Copy the "current_organization" URL from the response
```

Step 2 - Create the Webhook

```
curl --request POST \  
  --url https://api.calendly.com/webhook_subscriptions \  
  --header 'Content-Type: application/json' \  
  --header 'Authorization: Bearer YOUR_CALENDLY_API_KEY' \  
  --data '{  
    "url": "https://YOUR_VERCEL_URL/api/briefed-webhook",  
    "events": ["invitee.created"],  
    "organization": "YOUR_ORGANISATION_URL",  
    "scope": "organization"  
  }'  
  
# Success = you see "state": "active" in the response
```

PART 8

Test Everything

The Full Test

- Go to your Calendly booking page
- Book a test meeting - fill in ALL fields with realistic data
- Use a real company name and website URL
- Write a realistic challenge
- Confirm the booking
- Watch your Telegram - brief should arrive within 60 seconds
- Check Google Drive - a new doc should appear

Debugging Guide

If something doesn't work, go to Vercel → your project → Logs and look for error messages.

ERROR	FIX
500 Server Error	Check env var names are spelled exactly right in Vercel
Telegram not delivering	Make sure you pressed Start on your bot in Telegram
Google Doc not creating	Enable Google Docs API in Google Cloud Console
Website returns 403	Site blocks scrapers - brief still generates from other info
Function timeout	Reduce max_tokens in Claude call to 800
Chat not found	Get correct Chat ID from @RawDataBot not @userinfobot

PART 9

Customise for Your Business

The real power of Briefed is in the Claude system prompt. Customise it to match your services, industry knowledge, and typical objections.

Add Your Services

Add this to your Claude system prompt:

The following services can be offered as part of a custom build:

1. AI Strategy and Roadmap
2. Workflow Automation
3. Custom AI Tool Development
4. Done-For-You Implementation

Pricing signals:

- Small business 1-5 staff: \$2,000-\$5,000
- Medium business 5-20 staff: \$5,000-\$15,000
- Enterprise: \$15,000+

Add Industry Pain Points

Common pain points by industry:

Construction: quoting, job costing, site reports

Real estate: listing descriptions, lead follow-up

Healthcare: appointment booking, patient comms

Hospitality: reservations, reviews, rosters

Professional services: proposals, invoicing

Add Objection Prep

Common objections to flag:

"We already use [tool]" → ask what it cannot do

"Too expensive" → calculate time cost of current process

"Not technical" → done-for-you means they do not need to be

"Maybe next quarter" → ask what changes next quarter

REFERENCE

What's Inside Your Brief

Every Briefed prep note contains these 7 sections:

1. Business Summary	What they do, their market position, rough size
2. Industry and Market	Key context about their sector and competitive landscape
3. Likely Pain Points	Based on their industry and what they told you
4. Suggested Talking Points	What to open with and explore in the conversation
5. Questions to Ask	5 specific, useful questions tailored to this client
6. Relevant AI Solutions	Which tools and automations would likely apply to them
7. Red Flags	Complexity, budget concerns, or scope issues to watch for

How to Use Your Brief

Business Summary: Know what they do in 30 seconds - open the call from their world

Pain Points: These are your first questions - lead with empathy not pitch

Talking Points: Use these to show you already understand their industry

Questions: Ask these to uncover budget, urgency, and decision makers

Red Flags: Know what complexity is coming before it surprises you

Don't want to build this yourself? I build custom AI agents for businesses worldwide. If you'd rather have me set up Briefed for you as a done-for-you solution, get in touch at aiconsultancy.co.nz

You built it.

Now walk into every call like you already know them. Because you do.

Don't want to build it yourself?

I build custom AI agents for businesses worldwide. Get in touch and I'll build Briefed for you.

aiconsultancy.co.nz

Want more builds like this?

realmissai.com

@RealMissAI