

What is an AIOS?

What is an AIOS?

Your business should serve you. Not the other way around.

You started a business for freedom, freedom to travel, freedom to create, freedom to spend your mornings however you want. But somewhere along the way, the business became the boss. You're checking dashboards before coffee. You're in meetings all day just to "stay informed." You're the highest-paid administrator in your own company.

An AI Operating System (AIOS) is the way out.

So What Is It, Exactly?

An AIOS is an autonomous layer of intelligence wrapped around your entire business, your data, your operations, your communications, your decisions.

It's not a chatbot. It's not a prompt library. It's not another SaaS tool you pay \$49/month for and forget about. It's an operating system, like Windows runs your computer, your AIOS runs your business.

Here's what that looks like in practice:

- It knows your business. Your strategy, your team, your processes, your history, all loaded into context so every conversation starts informed. No more repeating yourself.
 - It sees your numbers. Revenue, traffic, subscribers, conversion rates, collected automatically from your actual data sources, every single day. No more logging into 7 different dashboards.
 - It watches everything. Meetings, Slack messages, signals from across your business, synthesized into a morning brief delivered to your phone before breakfast.
 - It lives in your pocket. Send a voice note from Telegram while walking the dog. Get an intelligent, context-aware response in seconds. Make decisions on the go.
 - It works while you sleep. Cron jobs collecting data, monitoring changes, running scheduled tasks, your business operates whether you're at your desk or on a beach.
-

What Makes This Different

You've probably seen AI tools before. ChatGPT, automation platforms, "AI-powered" everything. Most of them are point solutions, they solve one problem, in one silo, disconnected from everything else.

An AIOS is different because it's holistic. It wraps around your entire business. Your data feeds your intelligence. Your intelligence feeds your decisions. Your decisions feed your automations. Everything compounds.

And here's the part that changes everything: you build it yourself. Not with a dev team. Not with months of integration work. You, sitting with Claude Code, describing what you want in plain English. Layer by layer, module by module, until you have a system that

knows your business as well as you do, and handles the work that used to eat your entire day.

The Living Proof

This isn't theory. Liam runs 4 businesses through a single AIOS, 9 data sources, daily intelligence briefs, meeting analysis across 74+ calls daily, autonomous operations. All from his phone.

A team of 50+ people, coordinated by an AI layer that knows the business better than any single employee ever could. The business runs. Decisions happen. Intelligence flows. And Liam spends his time on what actually matters, building, creating, growing.

That's what you're building here. Not a toy. Not a side project. A real operating system for your real business.

The Promise

You will reclaim your bandwidth. Not by working harder. Not by hiring more people. By building a system that thinks, acts, and operates, so you can spend your time on the work that actually matters.

Grow the business. Build something new. Or simply enjoy the freedom you originally started the business for.

"That dream that we wanted with our business, which isn't a job, which is something that allows us to enjoy the freedom of life, to take trips when we need to, to take the morning off if you feel like it." — Liam Ottley

Next lesson: The Operator Trap - the specific problem your AIOS is built to solve.

The Operator Trap

The Operator Trap

You started a business to be free. Instead, you became the operator.

The Problem

You're working IN the business, not ON it. Every day is admin, firefighting, "must dos." 80% of your bandwidth goes to keeping the boat floating. There's nothing left for growth, strategy, or the life you actually wanted.

This is The Operator Trap, the business runs, but only because you're running it. You can't step away. You can't think bigger. You're the bottleneck.

Sound familiar?

- **You check Stripe, then GA4, then your CRM, then Slack, every morning before you've even had coffee.**
- **You sit in meetings not because you add value but because you need to "stay informed"**

- You have a to-do list that never gets shorter
- You tell yourself "I'll work ON the business this quarter," and then don't.
- You're managing people, fixing things they broke, answering questions they could answer themselves
- You still feel like you are the AI architect but still have to manage all the people in your team, still making long hours telling people what to do, or fixing the mess they made

Every one of these is a symptom of the same thing: you're trapped in operations.

The Old Answer (That Doesn't Work)

The traditional playbook says: hire more people, buy more tools, work more hours.

But think about what actually happens:

- Every new hire = more management overhead. More 1-on-1s. More onboarding. More "can you check this?" More people to coordinate, more mistakes to catch, more opinions to manage.
- Every new SaaS tool = more fragmentation. Another login. Another dashboard. Another tab in your browser you're supposed to check regularly. Nothing talks to anything else.
- More hours = you're still doing the work yourself. You're just doing more of it. That's not freedom, that's a treadmill.

More just means more to operate.

The New Answer

AIOS says the answer is less.

- **Less manual work – automate the tasks that eat your day**
- **Less people needed – an AI layer handles what used to require headcount**
- **Less time in operations – your system runs, you architect**
- **More bandwidth for the work that actually matters**

Instead of hiring three more people to manage your operations, you build a system that handles it. Instead of checking seven dashboards every morning, you get a single intelligence brief delivered to your phone. Instead of sitting in every meeting to "stay informed," your AI watches, summarizes, and flags what needs your attention.

You stop being the operator. You become the architect.

The Real Enemy

The enemy isn't AI skeptics. The enemy isn't your competitors. The enemy is the old model of business itself, the model that trades your freedom for revenue, your time for growth, and your bandwidth for "hustle."

AIOS is the escape hatch.

"Still stuck in The Operator Trap? Your competitor just wrapped that with their AIOS."

The Bigger Picture

Here's what most people haven't fully internalized: we don't even know what business is going to look like in 5-10 years. The rate of change is accelerating. Entire business models are being created and destroyed in months, not decades.

The only people who will survive this transition, who will stay profitable, stay alive, and stay ahead, are the ones who have already built these systems. Not because the system itself is the answer to every future challenge, but because of what it gives you: full capacity and bandwidth to respond.

When the next wave hits, the entrepreneur with an AIOS has 80% of their bandwidth free to apply maximum leverage to whatever strategic shift is required. They can pivot. They can adapt. They can move fast.

The entrepreneur without one? Still drowning in operations. Still manually doing things that should have been automated a year ago. They simply cannot react fast enough.

AIOS isn't just about efficiency today. It's the insurance policy for tomorrow.

Next lesson: The Five Layers - how you escape The Operator Trap, step by step.

The Five Layers

The Five Layers

How you escape The Operator Trap - one layer at a time.

You don't build the whole system in a weekend. You add one layer at a time. Each layer is independently valuable. And through this gradual exposure, you become technical without even trying, and you become a wizard one baby step at a time.

The Stack

Layer 1 – Context

Your AI understands your business.

Strategy, team, processes, history, all structured and loaded so every conversation starts informed. Your AI stops asking you basic questions. It knows your team names, your revenue targets, and your current priorities. It stops being "generic AI" and starts being your strategic partner.

***What changes:* Every interaction with Claude is 10x more useful. You stop wasting time explaining background. You start getting strategic answers immediately.**

Layer 2 – Data

Your AI sees your numbers in real-time.

Collectors pull from your actual data sources, Stripe, YouTube, GA4, Google Sheets, or whatever you use, and centralize everything into a single database. Auto generated metrics. A custom dashboard showing YOUR numbers. Query anything in plain English.

***What changes:* You stop logging into 7 different dashboards every morning. You ask "How did we do last week?" and get a real answer with real numbers from your real business.**

Layer 3 – Intelligence

Your AI watches everything and briefs you daily.

Meeting recordings collected and searchable. Slack messages stored and queryable. Your AI accessible from your phone via Telegram, voice notes, photos, or text. And every morning before breakfast, a comprehensive intelligence brief synthesizing everything that happened across your business.

***What changes:* You stop sitting in meetings just to "stay informed." You stop checking Slack obsessively. Your AI does that for you, and tells you what matters. Coffee and your Brief, that's how the day starts.**

Layer 4 – Automate

You audit every task you do. Then start crossing them off.

List everything you do across your business. Every task, every recurring responsibility. Score each one: can AI handle this? Fully automatable? Partially? Not yet? Then start from the top, automate the highest-impact, easiest-to-solve tasks first. One by one, permanently. Each task automated = bandwidth recovered. This is the turning point where the platform starts actively giving you your time back.

***What changes:* You go from "I have a cool AI system" to "I'm actually getting my life back." Your Task Automation % starts climbing. You can feel the difference.**

Layer 5 – Build

Your bandwidth is free. Now you choose.

Keep building new capabilities, a content machine, client systems, custom automations. Launch new initiatives. Work ON the business for the first time in years. Or simply enjoy the freedom. Take the morning off. Leave the laptop at home. Run the business from your phone in 15 minutes a day.

What changes: You're not following instructions anymore. You're an Architect, designing and building whatever your business needs next. This is the layer where you finally have the business you actually wanted.

The Bandwidth Story

Here's how bandwidth recovery works through the layers:

- **Layers 1-3 are the platform. They create an intelligent foundation that shrinks your admin overhead. You're already spending less time on busywork just by having context, data, and intelligence flowing automatically.**
 - **Layer 4 is the active reclamation. You systematically identify what's eating your time and automate it away. This is where the Task Automation % KPI becomes your scoreboard.**
 - **Layer 5 is the payoff. Freed bandwidth applied to growth, or simply to living the life you started the business for in the first place.**
-

The Compounding Effect

Each layer makes the next one more powerful:

- **Context alone is transformational. Your AI goes from generic to strategic.**
- **Add Data and it's a multiplier. Now your AI knows the business AND sees the numbers**
- **Add Intelligence and you're the most informed person in your organization.**
- **Add Automate and you're actively recovering your time.**
- **Add Build and you have the freedom to decide what's next.**

You never need to go further than where you're getting value. But the magic is in the compounding.

The Hidden Benefit

Here's what nobody tells you: by the time you've built Layer 3, you're technical.

Not because you studied. Not because you took a coding course. Because you've been gradually exposed to file systems, APIs, databases, and scripts through hundreds of baby steps. The methodology makes you a builder without you even realizing it's happening.

By Layer 4, you're not just using AI, you're architecting systems. By Layer 5, you're dangerous.

30% of workshop attendees haven't installed Claude Code yet. 45% are "tinkering." That's fine. Layers, Not Leaps. Go at your pace. Each layer independently valuable. Permission to go slow and you'll still get there.

Next lesson: The Five Principles - how to think while you build.

The Five Principles

The Five Principles

How to think while you build.

These aren't rules. They're mental models, the principles that separate people who get massive value from AIOS and people who tinker for weeks without progress. Internalize these and everything else falls into place.

1. Just Ask

The Claude Code Chad mentality.

The single most powerful shift: believe that anything is automatable, and then just ask.

Don't Google it. Don't YouTube "how to build an automation." Don't hire a developer. Open Claude Code and describe what you want in plain English. The only limit is your willingness to ask.

Most people self-censor. "Claude can't do that." "That's too complex." "I'd need a developer for that." Wrong. If you can describe it in plain English, Claude can build it. The person who asks for the impossible gets it, while the person who assumes it's impossible keeps doing it manually.

This is not a technical skill. It's a mindset.

Example: "I want something that knows everything about all my businesses so that when I talk to it about putting together a strategy, it can think about and visualize all the different pieces." That's not a fantasy. That's ContextOS. You just have to ask.

2. Talk, Don't Type

Voice-first interaction.

You're the CEO. You give instructions. You don't sit in a chat box typing carefully formatted messages like a customer support agent.

Hold your FN key, speak for 60 seconds, and let Claude format it. Install Wispr Flow for dictation anywhere on your Mac. Talk to your AI like you talk to a team member.

This is 3x faster than typing (200 words/minute spoken vs 50 typing). But more importantly, it changes the psychology. You stop thinking of Claude as a tool and start thinking of it as a partner. You stop writing prompts and start having conversations.

Non-technical people immediately feel powerful because they don't need to know syntax and they just need to talk.

3. Layers, Not Leaps

Patience. Gradual exposure. Compounding capability.

You don't build the whole system in a weekend. You add one layer at a time. Each layer is independently valuable.

And through this gradual exposure, you become technical without even trying. You become a wizard one baby step at a time.

This principle gives you permission to go slow, while promising you'll still get there. You don't need to understand everything right now. Just build the next layer. The understanding comes through doing.

4. Build for Scale and Security

Everything you build should be robust, safe, and self-improving.

Your AIOS touches real business data, real operations, real money. It needs to be built like infrastructure, not a weekend hack.

Security: Human-in-the-loop by default. Approval gates before anything critical executes. Your data stays local - you control what's sent, what's stored, and what requires your sign-off.

Scalability: Plan before you build. Use → → as your workflow - research, scope, then execute. Every system should include its own documentation and error handling.

Self-healing loops: When Claude updates CLAUDE.md based on mistakes it made - that's a self-healing loop. When your cron job logs its own failures and suggests fixes - that's self-improving infrastructure. The system gets smarter through use, not just through your direct intervention.

This is what separates a toy from a real operating system. Duct-tape automations break at 3am and nobody knows why. An AIOS built with this principle runs while you sleep - and if something goes wrong, it tells you what happened and how to fix it.

5. Borrow Before You Build

The 80/20 rule. Don't reinvent. Adapt.

In most cases, someone has already solved 80% of your problem. Your job is to find it, grab it, and adapt it to your workflow.

Modules are pre-built, battle-tested systems you can drop into your workspace. You don't architect a Telegram bot from scratch, you install the module. You don't design a data pipeline, you plug in the collector template and connect your sources.

The 80/20 rule: 80% of the time, borrow and adapt. 20% of the time, build something new. And when you build something new, share it back, so the next person can borrow from you.

This is why the Module Library exists. This is why the community matters. Every module you borrow is days of work you didn't have to do. Every module you share helps someone else skip that same work.

In Practice

These five principles work together:

1. You Just Ask for what you want (don't self-censor)
2. You Talk, Don't Type to give the instructions (voice-first)
3. You go Layers, Not Leaps and one thing at a time (patience)
4. You Build for Scale and Security, properly, not hacked together (quality)
5. You Borrow Before You Build and check the library first (efficiency)

Internalize these and you'll move faster, build better, and waste less time than someone who tries to figure it all out on their own.

Next lesson: Your Three KPIs - how to measure whether your AIOS is actually working.

Your Three KPIs

Your Three KPIs

How to know if your AIOS is actually working.

Not vanity metrics. Not "how many automations did I build." Three specific numbers that tell you whether you're escaping The Operator Trap, or just rearranging the furniture.

KPI 1: Away-From-Desk Autonomy

How many hours per day can you step away and nothing falls apart?

This is the north star. When you start, it's zero and everything requires you at your desk. As you add layers, this number climbs.

The goal isn't "AI does everything." The goal is freedom. It's taking the morning off to go to the gym. It's leaving your laptop at home for a weekend trip and running your business from Telegram on your phone. It's the closing distance between the business you have and the business you dreamed about.

Test it: Take a Friday off. Leave your laptop closed. Check your Telegram. Can you make the critical decisions from your phone in 15 minutes? Can the rest wait until Monday? If yes - your AIOS is working. If not, you know exactly what to build next.

Starting point: 0 hours

Target: Your business runs while you sleep, travel, or take the morning off

This KPI starts climbing at Layer 3 - once your intelligence layer is built and you're receiving daily briefs.

KPI 2: Task Automation %

What percentage of the tasks you currently do are automated or heavily augmented?

List everything you do across your businesses. Every task, every recurring responsibility. Then cross them off, one by one, as your AIOS takes them over.

Start at 0%. Aim for 60-70%.

This is your progress bar for freedom - and it's the scoreboard for Layer 4 (Automate). It's tangible, motivating, and personal to every builder. Your list is different from everyone else's but the principle is universal.

The exercise:

1. Map every task you do across all business areas
2. Score each one: Can AI handle this? (Fully / Partially / Not yet)
3. Prioritize by impact x ease
4. Start automating from the top of the list

(You'll do this formally with the /task-audit command in Layer 4 but start thinking about it now.)

Starting point: 0%

First milestone: 20-30% (you'll feel the difference)

180d Target: 60-70%

KPI 3: Revenue Per Employee

How much revenue does your business generate per person on the team?

Total revenue divided by total team members (including contractors).

This is the metric that captures the AIOS philosophy in a single number. We're not trying to build bigger companies. We're trying to build leaner, faster, more profitable companies with fewer people, fewer voices, and tighter execution.

When someone goes from 10-15 people to 2 founders + 3 people handling 100 clients, their revenue per employee goes through the roof. That's the AIOS effect. Not replacing people for the sake of it, replacing unnecessary complexity with intelligent automation.

Measure it monthly. Watch it climb as your AIOS takes over operational work that used to require headcount.

This is the long-term proof that Layer 5 is working.

How They Map to Your Journey

- Away-From-Desk Autonomy – Measures hours you can step away. Starts climbing at Layer 3 (Intelligence).
- Task Automation % – Measures tasks crossed off your list. Your scoreboard for Layer 4 (Automate).
- Revenue Per Employee – Measures business leverage. The long-term proof Layer 5 (Build) is working.

Every automation you build, every module you install, every task you cross off - moves one of these three numbers. If it doesn't move a KPI, ask yourself whether it's worth building.

Next lesson: Your Roadmap - the full journey mapped out, module by module.

Your Roadmap

Your Roadmap

Here's exactly what you're going to build, in what order, and what you'll have at each stage.

The Journey

The path is simple:

1. Set up your layers using plug-and-play modules (Sections 3-5)
2. Audit your tasks to see where your time actually goes (Section 6)
3. Borrow from the module library to solve as many as possible (Section 7)
4. Build custom automations for the rest (Section 8)

Each step has a clear outcome. Each module has a clear win. You always know where you are and what's next.

The Full Module Inventory

Layer 1 – Context (Foundation)

- ContextOS – 30-45 min, 0 API keys, Free. AI that knows your business: strategy, team, processes, history.
- InfraOS – 20-30 min, 0 API keys, Free. Version control, backups, documentation system, security practices.

Layer 2 – Data

- DataOS – 30-60 min, 1-7 API keys, Free. Business data pipeline, custom dashboard, daily automated collection.

Layer 3 – Intelligence

- IntelOS – 20-30 min, 1-3 API keys, Free. Meeting + Slack collection, searchable forever.
- CommandOS – 30-45 min, 2-3 API keys, Included. Telegram bot – your AIOS in your pocket, 24/7.
- Daily Brief – 20-30 min, 1 API key, ~\$1-3/mo. Morning intelligence brief delivered before breakfast.

Layer 4 – Automate

- ProductivityOS – 10-15 min, 0 API keys, Free. Complete task management system (GTD).

Layer 5 – Build

- **Slash Commands** – 5-10 min, 0 API keys, Free. /brainstorm + /explore – structured custom building.

Total core setup time: ~3-5 hours across all layers

Total monthly cost: ~\$1-3/mo for most members (just the Gemini API for Daily Brief)

Library Modules (pick what you need)

- **Content Pipeline** – 30-45 min, 0-1 API keys, Free. For you if: you create content regularly and want a strategic pipeline.
 - **Thumbnail Generator** – 30-45 min, 2 API keys, ~\$0-30/mo. For you if: you make YouTube videos and thumbnails are a bottleneck.
 - **Diagram Engine** – 5-10 min, 0 API keys, Free. For you if: you want to visualize systems, architectures, or processes.
 - **AI Landscape Monitor** – 15-20 min, 0-1 API keys, Free. For you if: you need to stay current on which AI tools are best.
-

The Pace

Go as far as you're getting value. Some members will build the full stack in a week. Others will take a month. Both are fine.

Remember: Layers, Not Leaps. Each layer is independently valuable. Context alone is transformational. Data is a multiplier. Intelligence is the game-changer. Everything after that is compounding.

The magic isn't in rushing to the finish. The magic is in the compounding.

Ready? Next section: **Getting Started - choose your path and set up your tools.**

AIOS Starter Kit Explained

AIOS Starter Kit Explained

Before you install anything, let's walk through what you're working with.

Every AIOS starts from the same foundation: the Starter Kit. It's a pre-built workspace structure that Claude Code reads and understands. Think of it as the skeleton of your AI Operating System, the bones that everything else attaches to.

[aios-starter-kit.zip](#) 63.91 KB

The Big Picture

Your AIOS workspace is a folder on your computer. Inside that folder is a set of files and directories that Claude reads at the start of every session. These files tell Claude who you are, what your business does, what your priorities are, and what tools are available. The better these files are, the smarter Claude is.

The Starter Kit gives you this folder structure out of the box, with templates, commands, and instructions already in place. You download it, open it, and start customizing.

What's Inside

CLAUDE.md – The Foundation

This is the most important file in your entire workspace. It's automatically loaded at the start of every Claude session. It tells Claude:

- What this workspace is
- What the AIOS mission is (the Five Layers, Five Principles, Three KPIs)
- What the folder structure looks like
- What commands are available
- How Claude should behave (patient, beginner-friendly, celebrates wins)

As you install modules and customize your workspace, CLAUDE.md grows with it. It's a living document and the single source of truth for how Claude should understand and operate within your system.

context/ – Your Business Brain

This folder holds everything Claude needs to know about you and your business:

- **business-info.md** – What your business does, your services, your market
- **personal-info.md** – Who you are, your role, your background
- **strategy.md** – Current priorities, goals, what you're focused on
- **current-data.md** – Key metrics, current state of the business
- **import/** – Drop any existing docs here (business plans, ChatGPT exports, strategy documents) and Claude will analyze them

These files start empty, the first module you install (ContextOS) walks you through filling them out. Once they're populated, every Claude session starts with full awareness of your business. No more repeating yourself.

.claude/commands/ – Your Slash Commands

These are actions Claude can execute when you type a command. The Starter Kit comes with:

- **/prime** – Run this at the start of every session. Claude reads your context files and confirms it understands your business.
- **/install** – Install an AIOS module. Point it at a module folder and Claude walks you through guided setup.
- **/create-plan** – Create a detailed plan before making changes. For when you want to build something new.
- **/implement** – Execute a plan step by step.
- **/task-audit** – Interview you about every task in your business, score them for automation potential, produce a prioritized audit doc.

As you install modules, new commands get added automatically. By the time you've built a few layers, you'll have a full toolkit of commands tailored to your business.

module-installs/ – Where Modules Live

When you download an AIOS module (a ZIP file), you unzip it into this folder. Then you tell Claude to install it. That's it. More on this in the next lesson.

Other Folders

- plans/ – Implementation plans created by /create-plan
 - outputs/ – Work products, deliverables, reports
 - reference/ – Templates, examples, reusable patterns
 - scripts/ – Automation scripts (added by modules as you install them)
 - .env – Your API keys and credentials (never shared, never committed to Git)
-

How It All Fits Together

1. You open your workspace in Claude Code
2. Claude reads CLAUDE.md and your context files, it knows who you are and what you're building
3. You install modules one at a time and each one adds a new layer to your AIOS
4. Each module adds its own scripts, commands, and context to the workspace
5. Over time, your workspace becomes a fully operational AI Operating System, custom-built for YOUR business

The Starter Kit is the starting line. The modules are the building blocks. And Claude is the builder who reads the instructions and walks you through every step.

Next lesson: Plug and Play Module Installs - how to actually install a module and what to expect.

Plug & Play Module Installs

Plug and Play Module Installs

Every module works the same way. Learn this once, use it forever.

AIOS modules are pre-built systems packaged as ZIP files. Each one adds a new capability to your workspace, a data pipeline, an intelligence layer, a Telegram bot, a content system. You download the ZIP, drop it in your workspace, and tell Claude to install it. Claude reads the instructions inside and walks you through the entire setup.

The Install Flow

Every module follows the same pattern:

1. **Download the ZIP** – grab the module file from the lesson page (attached below each module lesson in this course)
2. **Unzip it into your `module-installs/` folder** – so it ends up at something like *`module-installs/context-os/`*
3. **Open Claude Code in your workspace** – make sure you're in your AIOS workspace
4. **Run the `/install` command** – type: `/install @module-installs/context-os` (or whatever module you're installing)
5. **Follow along** – Claude reads the `INSTALL.md` inside the module and walks you through every step

That's it. Five steps. Same every time. Whether it's your first module or your tenth.

What to Expect During Install

Modules aren't just "click and done." They're consultative. Claude acts like a consultant who's setting up a system for your specific business. Here's what that actually looks like:

- **Conversations and interviews** – Claude will ask you questions. "What data sources does your business use?" "What does your content pipeline look like?" "What's your team structure?" Answer honestly and thoroughly - the quality of your answers directly determines the quality of your system.
 - **Explorations and brainstorming** – Some modules involve discovery. Claude might brainstorm data sources with you, help you figure out your funnel, or explore what automations make sense for your business.
 - **Document creation** – Claude will create files tailored to your answers. Strategy documents. Brand positioning. Funnel maps. Data schemas. These aren't templates, they're custom-built from your conversation.
 - **Technical setup** – Behind the scenes, Claude creates scripts, connects APIs, sets up databases, and configures automations. You don't need to understand the code, Claude explains what it's doing in plain English before it does it.
 - **Verification** – At the end, Claude tests that everything works and shows you the result. "Ask me a question about your business." "Check your dashboard." "Look at your Telegram." You always see proof that it worked.
-

Trust the Process

Every module is designed for complete beginners. The `INSTALL.md` inside each module is written to walk a non-technical business owner through the entire setup step by step.

Claude will:

- Explain what it's about to do before doing it

- **Ask for your permission before making changes**
- **Pause and check in if something needs your input**
- **Tell you clearly when you're done and what your result is**

If you get stuck, just ask Claude what's happening. If something goes wrong, Claude will explain the issue and help you fix it. The modules have been tested and refined, trust the process, follow the steps, and you'll get to the end with a working system.

What Each Module Gives You

Every module has a clear win at the end - a tangible result you can see and use:

- **ContextOS – Ask Claude a strategic question about your business. Watch it answer with full context.**
- **DataOS – Open your custom dashboard and see your actual business metrics.**
- **Daily Brief – Wake up, check Telegram, read your morning intelligence brief with coffee.**
- **CommandOS – Send a voice note from your phone and get an intelligent response in seconds.**

Each module builds on the last. Each one makes your AIOS more powerful. And each one gets you closer to escaping The Operator Trap.

Quick Reference

To install any module:

1. Download the ZIP from the lesson page
2. Unzip into module-installs/
3. Run: `/install @module-installs/[module-name]`
4. Follow Claude's instructions
5. Verify the result

Next lesson: Choose Your Path - Desktop or Builder? Pick how you want to work with your AIOS.

Choose Your Path

Choose Your Path

Two paths. Same modules. Same results. Pick the one that fits how you work.

The Claude Desktop Path

For business owners who want the simplest possible setup.

The Desktop Path uses the Claude Code Desktop app. You download it, open your workspace folder, and start talking. No terminal. No code editor. No command line. Just a conversation with Claude inside your project.

Who this is for:

- You're a business owner focused on automating YOUR business
- You want zero friction and just get it working
- Technical stuff isn't your thing (and you don't want it to be)

- You want to ease into things at your own pace

What you get:

- Virtually no setup, download the app, have a Claude plan, download the Starter Kit, and you're in
- Clean, simple interface, feels like chatting
- Every module works exactly the same way

The trade-off:

- You're limited to one conversation at a time and you can't run multiple agents in parallel
- You're working in a single space, so things like background tasks or multi-agent workflows aren't available
- Less flexibility for advanced customization down the road

For most members, especially if you're here to build an AIOS for your own business, the Desktop Path is the recommended starting point. You can always switch to the Builder Path later as you get more comfortable.

The VS Code Path

For people who want more control - or plan to build AIOS for clients.

The VS Code Path uses VS Code + Claude Code CLI. You work in a proper code editor with a terminal, which gives you access to everything Claude Code can do - parallel agents, background tasks, advanced workflows.

Who this is for:

- You're somewhat technical or ready to learn
- You want to build AIOS systems for clients (not just yourself)
- You want the full power - parallel agents, multi-workspace setups, advanced customization
- You like the idea of learning the technical side - it'll make you significantly more capable long-term

What you get:

- Full power of Claude Code, run multiple agents in parallel, background tasks, advanced automation
- Better for building complex systems and client work
- More flexibility to customize and extend everything
- You'll learn faster and become more technical through daily use

The trade-off:

- Slightly more setup, install VS Code, install the Claude Code extension, get comfortable with the terminal
 - Small learning curve at the start (but the modules walk you through everything)
-

The Bottom Line

Both paths use the same modules. Both paths get the same results. The difference is the interface and how much power you have access to.

If you're not sure, start with Desktop. You can always switch later and your workspace, your files, your modules all carry over. Nothing is lost.

If you're the type who picks things up quickly, or you're planning to build AIOS for clients, go Builder from day one. The extra control will pay off fast.

Pick your path, then follow the setup lesson for your choice below.

Next: Follow the setup lesson for your chosen path - Desktop Path Setup or Builder Path Setup.

What You're Building: Teaching Your AI Your Business

Layer 1 - Context: Teaching Your AI Your Business

After this layer, your AI knows your strategy, team, processes, and history. Every conversation starts informed. Your AI stops asking basic questions.

What You're Building

Right now, Claude is generic, the same for everyone. It doesn't know your business, your team, your revenue targets, or what you're working on this quarter. Every conversation starts from zero.

Layer 1 changes that. You're going to teach Claude everything about your business through structured context files, set up version control so you never lose work, and install the documentation and backup systems that keep your AIOS safe and growing.

What You'll Have After

- **Personalized CLAUDE.md that makes every Claude session aware of your business**
- **4 structured context files covering your business, role, strategy, and current data**
- **Git version control, every change tracked and reversible**
- **GitHub backup, your entire AIOS safe in the cloud**
- **A /commit command that saves, documents, and backs up your work in one step**
- **A documentation system that grows with your workspace**

The Clear Win

Ask Claude: *"What should I focus on this week?"*, and get a strategic, context-aware answer that knows your business, your team, your priorities, and your current data. That's the difference between a generic chatbot and YOUR AI.

Two Modules

1. ContextOS – Claude learns your business through a guided interview
2. InfraOS – Version control, backups, documentation, and security

Both are free. Zero API keys. About 50-75 minutes total. Let's go.

ContextOS

ContextOS

The foundation of your entire AIOS. Without this, Claude is generic. With this, Claude is YOUR strategic partner.

What This Is

ContextOS is a guided interview where Claude asks you about your business across four areas, what your business does, who you are, your strategy, and your current data. It builds structured context files from your answers and creates a personalized CLAUDE.md that gets loaded at the start of every session.

Three ways to provide context: import existing documents (business plans, strategy docs, ChatGPT exports), chat interview (Claude asks, you answer), or paste text. Most people use a mix.

Why This Matters

This is the single most important setup step. The quality of your context directly determines the quality of everything Claude does for you. Every module after this builds on this foundation. Without it, Claude is the same as everyone else's. With it, Claude knows your team names, your revenue targets, your current priorities and every conversation starts from that informed position.

Think of it this way: you're giving Claude the same briefing you'd give a new executive hire on their first day. Except Claude never forgets it.

What You'll Have After

- 4 structured context files in your context/ folder
- A personalized CLAUDE.md tailored to your business
- A working /prime command that loads everything at session start

The win: Ask Claude a strategic question about your business "*What should my top 3 priorities be this quarter?*", and watch it answer with full context. It knows you. It knows your business. That's the moment it clicks.

What You Need

- Prerequisites: Starter Kit set up (Section 2)
 - API Keys: None
 - Setup Time: 30-45 minutes
 - Monthly Cost: Free
-

How to Install

1. Download the context-os-v1.zip file attached to this lesson
2. Unzip it into your module-installs/ folder and you should have a folder called module-installs/context-os/
3. Open Claude Code in your AIOS workspace
4. Run: ***/install module-installs/context-os***
5. Follow Claude's guided interview and it will ask you about your business, your role, your strategy, and your current situation. Answer thoroughly. Use voice (Wispr Flow) if you have it - talking is 3x faster than typing and you'll give richer answers.

The module will walk you through everything. Trust the process, it's designed for complete beginners.

[context-os-v1.zip](#) 8.86 KB

Verify It Worked

1. Run `/prime` and Claude should summarize its understanding of your business
 2. Ask a strategic question: *"Based on what you know about my business, what should I focus on this quarter?"*
 3. If Claude answers with specific knowledge about YOUR business (not generic advice), you're set
-

Workshop Reference

This module was covered in Workshop #1 (Setup + Context OS, Feb 23). Check Section 9 for the full recording.

Next: InfraOS - version control, backups, and documentation. Protect everything you just built.

InfraOS

InfraOS

Version control, backups, documentation, and security. The skeleton that keeps your AIOS safe and organized as it grows.

What This Is

InfraOS sets up Git version control from scratch (explained for complete beginners), connects your workspace to GitHub for cloud backup, installs a merged /commit command that saves and documents your work in one step, and creates a documentation system that grows with your workspace. It also sets up your .env file with core API keys you'll need for future modules.

Why This Matters

You just built your context layer, that's valuable work. InfraOS protects it. Version control means every change is tracked and reversible. If something breaks, you roll back. If you want to see what changed last week, you check the log. GitHub backup means your entire AIOS is safe even if your machine dies.

And the documentation system means your workspace documents itself as it grows. HISTORY.md tracks every change. The docs/ folder stores how things work. Future you (and future Claude sessions) will thank you.

Think of ContextOS as the brain of your AIOS. InfraOS is the skeleton, the structure that holds everything together and keeps it safe.

What You'll Have After

- **Git version control initialized and explained**
- **GitHub account connected, your workspace backed up to the cloud**
- **A `/commit` command that saves, documents, and backs up in one step**
- **HISTORY.md changelog tracking every change**
- **docs/ routing index for technical documentation**
- **.gitignore protecting sensitive files**
- **.env set up with core API keys (Anthropic, OpenAI, Gemini) ready for future modules**

The win: Run `/commit` - your changes are saved, documented, and backed up to the cloud in one command. Layer 1 is complete. Your AIOS has a brain (ContextOS) and a skeleton (InfraOS).

What You Need

- Prerequisites: ContextOS installed
 - API Keys: None (but you'll set up 3 core keys for future modules)
 - Setup Time: 20-30 minutes
 - Monthly Cost: Free
-

How to Install

1. Download the `infra-os-v1.zip` file attached to this lesson
2. Unzip it into your `module-installs/` folder - you should have `module-installs/infra-os/`
3. Run: **`/install module-installs/infra-os`**

4. Follow the guided setup - Claude will walk you through creating a GitHub account (if you don't have one), initializing Git, making your first commit, and setting up the documentation system. Everything is explained in plain English as you go.

[infra-os-v1.zip](#) 18.95 KB

Verify It Worked

1. Run `/commit` – Claude should stage your changes, write a commit message, and push to GitHub
 2. Check your GitHub repository – your workspace files should be there
 3. Check that HISTORY.md has an entry for this session's work
-

Layer 1 complete. Your AI knows your business and your workspace is protected.

Next: Layer 2 - your AI starts seeing your numbers.

What You're Building: Your AI Sees Your Numbers

Layer 2 - Data: Your AI Sees Your Numbers

After this layer, your business metrics flow into a local database daily. You stop logging into 7 dashboards. You have one custom dashboard and can query anything in plain English.

What You're Building

Your AI already knows your business (Layer 1). Now it's going to SEE your numbers, revenue, traffic, subscribers, conversion rates, whatever matters to YOUR business. Collectors pull from your actual data sources every day and centralize everything into a single database on your computer.

No more checking Stripe, then GA4, then your CRM, then your spreadsheet, then Slack. One place. One question. Real answers.

What You'll Have After

- A SQLite database with your actual business data
- Custom collectors for your specific data sources (Stripe, YouTube, GA4, Google Sheets, Bitly - whatever you use)
- Auto-generated key-metrics.md that Claude reads every session
- A custom business dashboard showing YOUR numbers at a glance
- Daily automated collection via cron job - data refreshes while you sleep

The Clear Win

Open your custom dashboard and see your actual business metrics. Or just ask Claude: *"How did we do last week?"* and get a real answer with real numbers from your real business. Collected automatically, updated daily.

One Module

DataOS – the biggest setup of any single module, but also one of the most transformational. It's an interactive workshop where Claude brainstorms your data sources with you, then builds custom collectors tailored to your exact business.

- API Keys: 1-7 (depends on how many data sources you connect, the module walks you through each one)
- Setup Time: 30-60 minutes
- Monthly Cost: Free (all APIs have free tiers sufficient for this)

DataOS

DataOS

Centralize your business data. Build your dashboard. Stop logging into 7 platforms every morning.

What This Is

DataOS is an interactive data workshop. Claude interviews you about your data sources. Where do your numbers live? What matters most? What do you check every day? - then builds custom collectors for each one. Everything flows into a single SQLite database on your machine. Claude generates a key-metrics summary, builds you a custom dashboard, and sets up daily automated collection.

Six phases: Discovery (brainstorm your data sources) → Foundation (database setup) → Connect Sources (build collectors one by one) → Key Metrics (auto-generated summary) → Dashboard Workshop (build your visual dashboard) → Automate (schedule daily collection).

Why This Matters

Right now your data is scattered across 10 different dashboards. You check Stripe here, GA4 there, your CRM somewhere else. Every morning starts with a tour of browser tabs just to know what's going on.

DataOS centralizes everything so Claude sees the full picture - and so do you. The custom dashboard gives you a single view of your business. The daily collection means your numbers are always fresh. And because it's all in a queryable database, you can ask Claude anything: *"Compare this month to last month."* *"What's my conversion rate trending?"* *"Which channel drives the most revenue?"*

This is the data foundation that feeds your Intelligence layer (Layer 3), your Daily Brief, and every data-driven decision you'll make from here.

What You'll Have After

- SQLite database with your actual business data
- Custom collectors for your specific sources (module includes templates for YouTube, Stripe, GA4, Google Sheets, Bitly, and a custom collector template for anything else)
- Auto-generated key-metrics.md loaded every session
- A custom Next.js dashboard tailored to your business
- Daily automated collection via cron job (Mac) or scheduled task (Windows)

The win: Open your dashboard and see YOUR numbers. Or ask Claude *"Show me my key metrics"* and get a real answer with real data, collected automatically.

What You Need

- Prerequisites: ContextOS installed (Layer 1)
 - API Keys: 1-7 (depends on your data sources - module guides you through each one step by step)
 - Setup Time: 30-60 minutes (depends on how many sources you connect)
 - Monthly Cost: Free (all APIs have free tiers)
-

How to Install

1. Download the data-os-v1.zip file attached to this lesson
2. Unzip it into your module-installs/ folder and you should have module-installs/data-os/
3. Run: ***/install module-installs/data-os***
4. Follow the guided workshop – Claude will brainstorm your data sources with you, then walk you through connecting each one. Start with your most important source (usually Stripe or Google Sheets). You can always add more later.

Tip: Don't overthink this. Start with ONE data source, the most important one for your business. You can always come back and add more. The module is designed for iterative setup.

[data-os-v1.zip](#) 45.78 KB

Verify It Worked

1. **Ask Claude: "Show me my key metrics" - you should get real numbers from your database**
2. **Open your dashboard and verify your data is displaying**

3. Check that your daily collection is scheduled and running

Workshop Reference

This module was covered in Workshop #2 (Data OS, Feb 24). Check Section 9 for the full recording.

Layer 2 complete. Your AI knows your business AND sees your numbers. Next: Layer 3 - Intelligence. Your AI starts watching meetings, lives on your phone, and briefs you every morning.

What You're Building: Your AI Watches Everything

Layer 3 - Intelligence: Your AI Watches Everything

After this layer, your AI collects all your meetings and messages, runs on your phone via Telegram, and delivers a morning intelligence brief before breakfast. This is where the magic happens.

What You're Building

Your AI knows your business (Layer 1) and sees your numbers (Layer 2). Now it gets eyes, ears, and a voice. Meeting recordings collected and searchable forever. Slack messages stored and queryable. A Telegram bot that puts your AIOS in your pocket,

accessible from your phone 24/7. And the capstone: a comprehensive morning brief that synthesizes EVERYTHING into one 4-minute read delivered before breakfast.

This is the layer where most members say it "clicks." You stop sitting in meetings just to stay informed. You stop checking Slack obsessively. Your AI does it for you and tells you what matters.

What You'll Have After

- Meeting recordings collected, classified, and searchable in your database
- Slack messages stored and queryable
- A Telegram bot with persistent Claude agents, your AIOS in your pocket
- A daily morning intelligence brief delivered to Telegram before breakfast
- An interactive topic where you can drill into anything from the brief

The Clear Win

Wake up. Check Telegram. Read your Brief with coffee. You know what happened overnight, what your numbers look like, what needs your attention, and what the AI recommends. You're the most informed person in your organization before 8am, without opening a laptop.

Three Modules

1. IntelOS – Meeting and Slack collection (the ears)
2. CommandOS – Telegram bot, your AIOS in your pocket (the voice)
3. Daily Brief – Morning intelligence synthesis (the capstone)

Total setup: ~70-105 minutes across all three. Monthly cost: ~\$1-3 (Gemini API for Daily Brief). Everything else is free.

IntelOS

IntelOS

Your AI gets ears. Every meeting, every Slack message is collected, classified, and searchable forever.

What This Is

IntelOS is the intelligence collection layer. It connects to your meeting recording service (Fireflies, Fathom, or a custom recorder) and your Slack workspace to pull meeting transcripts and messages into your database. Meetings get classified by department. Everything becomes queryable. Cron-based collection keeps data fresh automatically.

This module is about collection and storage, gathering the raw intelligence. The analysis and synthesis happens in the Daily Brief module (the capstone of this layer).

Why This Matters

Right now, information from meetings and Slack disappears into the void. You had an important call last Tuesday, what was decided? Someone mentioned a key insight in Slack three days ago, what was it? Without IntelOS, you either remember it or it's gone.

With IntelOS, every meeting and every message is captured, stored, and searchable. Your Daily Brief (next module after CommandOS) can't synthesize what it can't see. IntelOS gives it eyes and ears across your entire organization.

What You'll Have After

- Meeting recordings collected from Fireflies, Fathom, or custom recorder
- Slack messages stored with per-message granularity and daily transcript queries
- Meeting classification by department
- Query helpers for searching across meetings and messages
- Automated collection on a schedule

The win: Ask Claude *"Find that meeting with [person] last week about [topic]"*, instantly found and summarized. Nothing falls through the cracks again.

What You Need

- Prerequisites: ContextOS + DataOS (Layers 1-2)
 - API Keys: 1-3 (Fireflies or Fathom + Slack)
 - Setup Time: 20-30 minutes
 - Monthly Cost: Free (Fireflies/Fathom free tiers + Slack API free)
-

How to Install

1. Download the intel-os-v1.zip file attached to this lesson
2. Unzip it into your module-installs/ folder and you should have module-installs/intel-os/

3. Run: ***/install module-installs/intel-os***
4. Follow the guided setup – Claude will ask which meeting recording service you use, walk you through the API key setup, connect Slack if applicable, and set up automated collection.

Note: If you don't have meetings recorded yet, that's fine. Set up the connection now and data will start flowing as you record meetings going forward. "No meetings yet" is expected if you just started.

[intel-os-v1.zip](#) 28.3 KB

Verify It Worked

1. Ask Claude to query your meetings: ***"Show me recent meetings"***
 2. If you have Slack connected, ask: ***"What were the key messages in [channel] this week?"***
 3. Check that your collection schedule is set up
-

Next: CommandOS - your AIOS goes in your pocket. Telegram bot, voice notes, photos – your AI accessible from your phone 24/7.

CommandOS

Your AIOS goes in your pocket. Text, voice notes, photos, all processed with full workspace context, accessible from your phone 24/7.

What This Is

CommandOS is a Telegram bot connected to persistent Claude Code agent sessions. You message it from your phone, the text, voice notes, and photos, it responds with the full context of your AIOS workspace. It has session management (`/name`, `/compact`), optional voice transcription via Whisper, and can generate charts, PDFs, and detailed reports. Your AI accessible from anywhere.

Why This Matters

Your AIOS is only as useful as your ability to access it. Right now, you need to be at your computer to talk to Claude. CommandOS changes that. Your AI lives in your pocket.

Make decisions on the go. Ask questions while walking the dog. Check metrics from the airport. Get a full analysis delivered to a Telegram topic while you're at dinner. Send a voice note saying *"Give me a summary of last week's numbers"* and get an intelligent response in seconds.

This is how you run your business from your phone - and it's where the Away-From-Desk Autonomy KPI starts climbing.

What You'll Have After

- A Telegram bot with persistent Claude Code agents
- Full workspace context in every conversation
- Voice note support (optional, needs OpenAI API key for Whisper)
- Photo analysis (send a photo, get insights)
- Chart and PDF generation on demand
- Topic management (`/name`, `/compact`) for organizing conversations
- The delivery channel for your Daily Brief (next module)

The win: Send a voice note from your phone while on a walk *"How did our metrics look this week?"* and get an intelligent, context-aware response in seconds. Your AIOS is always with you.

What You Need

- Prerequisites: ContextOS (CLAUDE.md must exist)
- API Keys: 2-3 (Telegram Bot Token + Anthropic API key required; OpenAI API key optional for voice)
- Setup Time: 30-45 minutes
- Monthly Cost: Included in your Claude plan

How to Install

1. Download the `command-os-v1.zip` file attached to this lesson
2. Unzip it into your `module-installs/` folder and you should have `module-installs/command-os/`

3. Run: `/install module-installs/command-os`
 4. Follow the guided setup — Claude will walk you through creating a Telegram bot (via BotFather), setting up API keys, configuring the bot, and deploying it. The module handles everything - you just follow the steps.
-

Verify It Worked

1. Open Telegram on your phone
 2. Find your bot and send a message: *"What do you know about my business?"*
 3. You should get a context-aware response that shows Claude knows your business, team, and priorities
 4. Try a voice note (if you set up Whisper), speak a question and check the response
-

Workshop Reference

The Telegram bot setup was covered in Workshop #5 (Desktop + Modules, Feb 27). Check Section 9 for the recording.

Next: Daily Brief - the capstone of Layer 3. Everything you've built feeds into a morning intelligence brief delivered to your phone before breakfast.

Daily Brief

Daily Brief

The capstone of Layers 1-3. Everything you've built culminates here. One 4-minute read replaces logging into 7 platforms and sitting in every call.

What This Is

The Daily Brief is an adaptive morning intelligence report that synthesizes your metrics (from DataOS), your meetings and messages (from IntelOS), and your

business context (from ContextOS) into a single Gemini mega-prompt. It produces a dashboard image, a narrative brief, key signals, and strategic recommendations - delivered to a Telegram topic via CommandOS every morning before breakfast.

Three presets: Solo Operator, Small Team, Agency. Each with customizable sections and word budgets. It reads your funnel.md to adapt the dashboard and analysis to YOUR specific business. And the Telegram topic is interactive - reply to drill into anything.

Why This Matters

This is THE moment for most members. Everything you've built across three layers feeds into this single output. Your context, your data, your meetings, your messages, all synthesized into one brief delivered before you've had your first coffee.

Before breakfast, you know: what happened overnight, what your numbers look like, what needs your attention, and what the AI recommends. You're the most informed person in your organization. Without opening a laptop. Without sitting in a single meeting.

For many members, this is the moment the whole thing clicks. This is where you feel the bandwidth coming back.

What You'll Have After

- **An adaptive morning brief delivered to Telegram daily**
- **A dashboard visualization of your key metrics**
- **Narrative synthesis of meetings, messages, and signals**

- **Strategic recommendations based on your actual data**
- **An interactive Telegram topic, reply to drill into anything**
- **Daily cron scheduling, it runs automatically every morning**

The win: Coffee and your Brief. Wake up → open Telegram → read your Brief. You're informed. You're in control. The platform is built.

What You Need

- Prerequisites: ContextOS + DataOS + IntelOS + CommandOS (everything you've built so far)
 - API Keys: 1 (Gemini API - for long-context synthesis)
 - Setup Time: 20-30 minutes
 - Monthly Cost: ~\$1-3/month (Gemini API)
-

How to Install

1. Download the `daily-brief-v1.zip` file attached to this lesson
2. Unzip it into your `module-installs/` folder and you should have `module-installs/daily-brief/`
3. Run: **`/install module-installs/daily-brief`**
4. Follow the guided setup — Claude will help you choose your preset (Solo/Team/Agency), configure your `funnel.md`, set up the Gemini API key, test the brief generation, and schedule the daily cron job.

[daily-brief-v1.zip](#) **29.94 KB**

Verify It Worked

1. Run a test brief generation – Claude will trigger it during setup
2. Check your Telegram – the brief should arrive in its dedicated topic
3. Verify it includes your actual metrics, meeting summaries, and recommendations
4. Reply to the brief with a follow-up question – confirm the interactive topic works

LAYER 3 COMPLETE. Your AIOS platform is built. Your AI knows your business, sees your data, watches your meetings, lives on your phone, and briefs you every morning. The foundation is done.

Next: Layer 4 - Automate. The platform is built. Now audit your tasks and start crossing them off. This is where you go from "I have a cool AI system" to "I'm actually getting my life back."

What You're Building: Reclaim Your Bandwidth

Layer 4 - Automate: Reclaim Your Bandwidth

The platform is built. Your AI knows your business, sees your data, and briefs you every morning. Now it's time to audit what you're actually spending your time on and start automating it away.

What You're Building

Layers 1-3 gave you the platform, an intelligent foundation that already shrinks your admin overhead. But you're still doing most of the work yourself. Layer 4 is where that changes.

You're going to map every task you do across your entire business. Score each one for automation potential. Prioritize by impact. Then start crossing them off, one by one, permanently. Each task automated is bandwidth recovered. This is where you go from "I have a cool AI system" to "I'm actually getting my life back."

What You'll Have After

- A complete Task Audit, every recurring task mapped, scored, and prioritized
 - A GTD productivity system to manage the work of automating everything
 - A clear understanding of what to borrow from the Module Library vs what to build custom
 - Your Task Automation % KPI starts climbing
-

The Shift

This is the turning point. Layers 1-3 were about building the platform. Layer 4 is about active reclamation, systematically identifying what eats your time and automating it away. This is where the IN the business → ON the business transition actually happens.

"This is where you go from 'I have a cool AI system' to 'I'm actually getting my life back.'"

The Task Audit

The Task Audit

You can't automate what you haven't mapped. This is where you get clarity on exactly where your time goes and what to automate first.

What This Is

The Task Audit is a slash command (/task-audit) already built into your Starter Kit. It runs a structured interview where Claude walks through every area of your business, asking about every recurring responsibility, every time sink, every "I have to do this every week" task. It keeps going until you're completely empty.

Then it produces a Task Audit document in your workspace with everything mapped, scored, and prioritized.

Why This Matters

Most business owners have no idea where their time actually goes. They feel busy - overwhelmed, even - but they can't tell you exactly which tasks eat the most hours or which ones could be automated tomorrow.

The Task Audit forces that clarity. And once you see the list, scored and prioritized, the path forward is obvious. You stop guessing what to build and start systematically crossing items off.

This becomes your personal scoreboard for the Task Automation % KPI. Every item you tick off is bandwidth permanently recovered.

What You'll Have After

- **A complete map of every recurring task across all areas of your business**
 - **Each task scored: Fully Automatable / Partially Automatable / Not Yet / Human-Only**
 - **Prioritized by impact and ease - what to tackle first is clear**
 - **Checkboxes to tick off as you automate each one**
 - **A "Solved By" column so you can track what system or module handled each task**
 - **The document lives at context/task-audit.md and Claude reads it and uses it to suggest what to build next**
-

How to Run It

1. **Open Claude Code in your AIOS workspace**

2. Type: */task-audit*
 3. Answer Claude's questions honestly and thoroughly and don't skip the small stuff, those tasks add up
 4. Claude will walk through 9 business areas: Operations, Finance, Sales, Marketing, Content, HR, Client Delivery, Product, and Admin
 5. When you're done, Claude produces the scored and prioritized audit document
-

Verify It Worked

Open `context/task-audit.md` in your workspace. You should see a structured document with all your tasks listed, scored, and prioritized. The highest-impact, easiest-to-automate tasks should be at the top. These are your first targets.

Next lesson: ProductivityOS - the system to manage the work of automating everything on your list.

ProductivityOS

ProductivityOS

The Task Audit shows you WHAT to automate. ProductivityOS is how you MANAGE the work of doing it.

What This Is

ProductivityOS installs a complete Getting Things Done (GTD) system in your workspace. GTD is a productivity methodology trusted by millions, it gives you a simple, repeatable process for capturing everything on your plate, deciding what to do about it, and making sure nothing falls through the cracks.

You get: an inbox for capturing ideas and tasks, a decision tree for processing them, context-based next action lists, project tracking, a waiting-for list, a someday-maybe list for things you're not ready for yet, a weekly review process, and an auto-refreshing dashboard that shows you everything at a glance.

Why This Matters

You've just done the Task Audit. You have a prioritized list of things to automate. Without a system to manage that work, the list becomes another thing you ignore, sitting in a file, getting stale, never acted on.

ProductivityOS turns your Task Audit into active projects. Each automation becomes a tracked item with next actions, deadlines, and progress. The weekly review keeps you honest. The dashboard keeps you focused. Claude manages the whole thing for you.

How GTD Works (Quick Version)

- Capture – Everything goes into the inbox. Ideas, tasks, things someone mentioned, stuff you need to do. Don't organize it - just capture.

- Process – Work through the inbox using the decision tree: Is it actionable? What's the next action? Can it be done in 2 minutes? (Do it now.) Does it need a project? (Create one.) Waiting on someone? (Track it.)
- Organize – Actions go to context lists (@computer, @calls, @errands). Projects get their own tracking. Waiting-for items get followed up. Someday-maybe for things you're not ready for yet.
- Review – Weekly review: process the inbox, walk through all projects and actions, update everything. This is what keeps the system alive.
- Do – Pick from your context list based on where you are and what you have energy for. Your dashboard always shows what to do next.

Claude handles the heavy lifting. You run /process to clear the inbox and /review for the weekly review. Claude walks you through both.

Install It

1. **Download the ZIP below**
2. **Unzip into your module-installs/ folder**
3. **Run: */install module-installs/productivity-os***
4. **Follow Claude's guided setup, it takes 10-15 minutes (the fastest module)**
 - **Prerequisites: ContextOS**
 - **Setup time: 10-15 minutes**
 - **API keys: None**
 - **Cost: Free**

[productivity-os-v1.zip](#) 34.12 KB

Verify It Worked

Run /prime and check that Claude mentions your GTD system. Then try capturing something: tell Claude "Add 'research CRM automation options' to my inbox." Then run /process and Claude should walk you through deciding what to do with it.

Check your GTD dashboard, you should see your projects, next actions, and waiting-for items organized and ready to go.

Next lesson: Borrow Before You Build - before you build anything custom, check the library first.

Borrow Before You Build

Borrow Before You Build

Before you build ANYTHING custom, check the library. 80% of the time, someone has already solved your problem.

The Principle

This is Principle 5: Borrow Before You Build. Look at your Task Audit. For each item you want to automate, ask:

1. Is there an off-the-shelf module for this? Check the Module Library (next section). If there's a module that solves your problem, grab it. Install it. Done.
2. Has anyone in the community built something similar? Ask in the Accelerator community. Chances are someone has faced the same problem and might have a solution or a starting point.

3. Only AFTER exhausting the library and community do you go to custom building (Section 8).
-

Why This Matters

Building from scratch is slow and fragile. You spend 3 days building something that a pre-built module solves in 30 minutes. Modules are battle-tested, documented, and designed for beginners. They've been refined through real use.

Every module you borrow instead of building is time saved. Every time saved is bandwidth recovered. That IS the point of this whole system.

The Workflow

1. Open your Task Audit (context/task-audit.md)
 2. Look at your highest-priority items
 3. Go through the Module Library (next section), match modules to tasks
 4. Install the ones that solve your problems
 5. Come back to the Task Audit and tick off what's been handled
 6. Whatever's left after the library, that's what you build custom in Section 8
-

The Flywheel

Build → Share → Someone else borrows → They improve → Share back. The more people share, the less everyone has to build from scratch. Your contribution grows the library. Someone else's contribution solves your next problem. This is how the AIOS ecosystem compounds.

Next section: The Module Library - off-the-shelf solutions for common problems. Grab what matches your Task Audit.

The Module Library

The Module Library

Off-the-shelf solutions for common problems. Grab what matches your Task Audit, don't build what you can borrow.

How This Works

You've just done the Task Audit. You have a prioritized list of things eating your time. Before you build anything custom, check what's available here.

Each module in this library solves a specific, common problem - content creation, thumbnails, diagrams, keeping up with AI tools. They're packaged the same way as the core modules you've already installed: download the ZIP, drop it in your workspace, run /install. Claude walks you through everything.

What's Different About These

The core modules (Sections 3-5) are the AIOS platform, everyone installs those in order. The library modules are pick and choose. You only grab what's relevant to YOUR business and YOUR Task Audit.

Content creator? Grab Content Pipeline and Thumbnail Generator. Agency builder? AI Landscape Monitor keeps you sharp on which tools to recommend. Visual thinker? Diagram Engine gives you professional architecture diagrams in seconds.

You don't need all of these. Grab what solves YOUR problems.

What's Coming

This library grows over time. New modules get added as they're built, both from the core team and from community members who share their builds (see Section 8.3). Check back regularly. The module that solves your next problem might already be here.

Browse the library below and match modules to your Task Audit items.

Content Pipeline

Content Pipeline

For you if: content creation is on your Task Audit. You create YouTube videos, LinkedIn posts, newsletters, or any regular content and want to systematize ideation, development, and scheduling.

What This Is

Content Pipeline is a full content lifecycle system. It starts with an interactive Brand & Content Workshop where Claude interviews you about your brand, audience, content pillars, and strategy - then builds a personalized content engine around your answers.

From there, you get three commands that handle the entire content lifecycle:

- **/capture – Quick idea capture. See something interesting? Hear something in a meeting? Just tell Claude - it classifies the idea by channel, format, and content pillar, then stores it in your pipeline database.**
- **/develop – Full concept development. Takes a captured idea and develops it into a complete concept - with strategic positioning, title options with viral elements, thumbnail concepts, hook strategies. Uses a 7-day context window of your recent videos, meetings, and voice captures so every concept is grounded in what's actually happening in your business.**
- **/schedule – Batch scheduling. Shows your developed ideas ranked by priority. Pick what to schedule, set publish dates, and the system calculates film-by dates and can push to Notion automatically.**

Supports YouTube and LinkedIn out of the box. Optional Notion sync for visual pipeline management. Optional Telegram integration for capturing ideas on the go.

Why This Matters

Content creation is one of the biggest recurring time sinks for business owners. The blank page problem. The "what should I talk about" loop. The scattered notes that never become anything. Content Pipeline replaces all of that with a structured system, ideas flow in, get developed with full business context, and get scheduled with deadlines.

Never stare at a blank page again. Your pipeline is always full, always strategic, always connected to what's happening in your business right now.

Install It

1. Download the ZIP below
2. Unzip into your `module-installs/` folder
3. Run: `/install module-installs/content-pipeline`
4. Follow the Brand & Content Workshop and Claude will interview you about your content strategy, audience, and pillars (30-45 minutes for a thorough setup)
 - Prerequisites: ContextOS + DataOS
 - Setup time: 30-45 minutes
 - API keys: 0-1 (Notion optional)
 - Cost: Free

[content-pipeline-v1.zip](#) 41.28 KB

Verify It Worked

Try capturing an idea: tell Claude about a content topic you've been thinking about. Then run /develop on it - you should get a fully developed concept with title options, thumbnail concepts, and strategic positioning based on your actual business context.

Check your content pipeline - you should see ideas flowing through capture → develop → schedule stages.

Thumbnail Generator

Thumbnail Generator

For you if: you make YouTube videos and thumbnails are a bottleneck - waiting for designers, spending too long iterating, or not getting the quality you want.

What This Is

Thumbnail Generator creates professional YouTube thumbnails using AI image generation. It uses an intelligent photo selection system, you build a catalog of your photos classified by expression, pose, and setting, and the system picks the best combination for each thumbnail concept.

For each thumbnail, it generates 4 distinct concepts in parallel using different compositions, color schemes, and text treatments. These get laid out in a 2x2 comparison grid so you can pick the winner at a glance. Brand style rules auto-inject into every generation so your thumbnails stay consistent.

The system includes a full thumbnail psychology framework, desire loops, stun gun elements, composition types, plus a YouTube competitor research tool that analyzes what's working in your niche.

Why This Matters

Thumbnails are the single biggest lever for YouTube click-through rate, and they're one of the most time-consuming parts of the content workflow. Waiting for a designer, going through rounds of feedback, or struggling with Canva yourself, that's hours per video that could be minutes.

Thumbnail Generator gives you 4 professional options in minutes. Iterate with feedback until you have the winner. No designer needed, no back-and-forth, no bottleneck.

Install It

1. Download the ZIP below
2. Unzip into your `module-installs/` folder
3. Run: `/install module-installs/thumbnail-generator`
4. Follow Claude's setup, the bulk of the time is classifying your photos for the catalog (expressions, poses, settings)
 - Prerequisites: None (standalone module)
 - Setup time: 30-45 minutes (photo classification is the bulk)
 - API keys: 2 (Gemini + Scrape Creators for YouTube research)

- **Cost: ~\$0-30/month (Gemini free tier covers light usage, Scrape Creators \$29/month for YouTube research)**

[thumbnail-generator-v1.zip](#) 37.3 KB

Verify It Worked

Run `/thumbgen` with a video topic. You should get 4 distinct thumbnail concepts in a 2x2 comparison grid within a few minutes. Give feedback on your favorite - the system iterates and refines. Your brand style should be consistent across all 4 concepts.

Diagram Engine

Diagram Engine

For you if: you want to visualize your AIOS architecture, create system diagrams for clients, or turn any process into a professional diagram.

What This Is

Diagram Engine installs D2, a modern diagramming language and gives Claude everything it needs to create professional architecture diagrams autonomously. Claude writes the D2 source code, renders it to PNG, visually validates the result, and iterates until it looks right.

Includes 5 design patterns (master overview, data pipeline, automation schedule, command system, education funnel) and a render script that handles the build process. You describe what you want in plain English. Claude builds it.

Why This Matters

Diagrams make the complex simple. Whether you're mapping your own AIOS architecture, creating a client deliverable, or documenting a business process, having professional diagrams on demand is a superpower. No Figma, no Miro, no wrestling with drag-and-drop tools. Just describe what you want.

Particularly useful if you're building AIOS systems for clients, a clean architecture diagram communicates more value than any amount of text.

Install It

1. Download the ZIP below
2. Unzip into your `module-installs/` folder
3. Run: `/install module-installs/diagram-engine`
4. Follow Claude's setup, it installs D2, sets up the render pipeline, and creates your first test diagram
 - Prerequisites: None (standalone module)
 - Setup time: 5-10 minutes (the fastest library module)
 - API keys: None
 - Cost: Free

[diagram-engine-v1.zip](#) 13.26 KB

Verify It Worked

Ask Claude to create a diagram of your AIOS architecture, the layers you've built, the modules installed, how data flows between them. You should get a clean PNG rendered in seconds. Try iterating, ask for different layouts, colors, or levels of detail.

AI Landscape Monitor

AI Landscape Monitor

For you if: you need to stay current on which AI tools are best for what, especially if building for clients or choosing tools for your own automations.

What This Is

AI Landscape Monitor is a daily scanning system that tracks the best AI models across 10 categories: text (LLMs), code, vision, image generation, image editing, search, text-to-video, image-to-video, text-to-speech, and speech-to-text.

It pulls live rankings from public leaderboards (LMArena, TTS Arena, Voice Writer), compares them against your documented state, and flags when rankings change. When changes are detected, it spawns parallel research agents to investigate, pricing,

integration code, community sentiment. Your ai-docs/ folder always reflects the current state of the art.

Integrates with /prime so your AIOS automatically knows which tools are best when building anything that uses AI.

Why This Matters

The AI landscape changes weekly. Models that were best-in-class last month get dethroned. New capabilities appear overnight. If you're recommending tools to clients or choosing models for your own automations, you need to know what's current, not what was current 3 months ago when you last checked.

AI Landscape Monitor means your system always knows. Never recommend an outdated model to a client again. Never build with last month's tools when something better launched this week.

Install It

1. Download the ZIP below
2. Unzip into your module-installs/ folder
3. Run: */install module-installs/ai-landscape-monitor*
4. Follow Claude's setup, it creates the ai-docs/ category structure and runs an initial scan across all 10 categories
 - Prerequisites: DataOS
 - Setup time: 15-20 minutes

- API keys: 0-1 (OpenRouter optional, free)
- Cost: Free for daily scans. Deep-dive research runs ~\$5-15 per run.

[ai-landscape-monitor-v1.zip](#)31.89 KB

Verify It Worked

Check your ai-docs/ folder and you should see 10 category folders, each with a state-of-the-art.md file showing current rankings. Run /prime and ask Claude "What's the best model for image generation right now?" it should give you a current, informed answer pulled from the latest rankings.

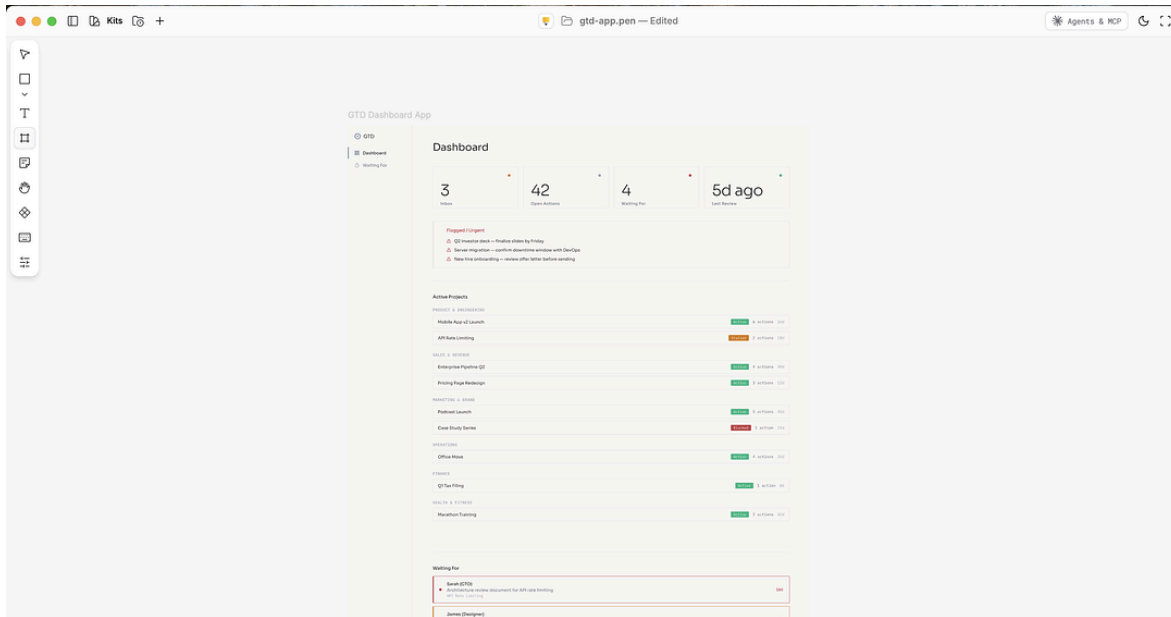
"Create Your Own Integration" Skill

This skill can be invoked to walk you through creating any kind of new non-invokable skill for your workspace e.g. Hubspot integration or podcast scraping capability. See the workshop from Monday 16th of March for more context on what Capabilities are and how to leverage them in your AIOS!

Build a "New App" Skill

This Skill allows you to go from a [Pencil.dev](#) design for an app to properly scoping the tech stack and architecture of it:

[new-app.zip](#)11.15 KB



This skill outputs a full plan file for the build including the phases of implementation involved.

After it creates this file, you must run `"/create-plan phase 1"` and let it create a proper plan file for each phase of implementation. For example my custom DataOS dashboard has 7 phases, so I ran the create plan command 7 times one after the other until I had all 7 plan files.

Then open up as many new Claude Code sessions as you have plan files, run `/prime` in each of them, then run `"/implement @plans/YOUR_PHASE_X_PLAN_FILE_NAME.md"`.

To be safe, build phase 1 to completion before starting phase 2 etc.

Watch the workshop on designing beautiful apps with Pencil.dev from Friday 20th March for a full walkthrough of this in action!

Read the above carefully or it will not work for you!

Frontend Design Skill

Frontend Design

Get polished HTML mockups before you write a single line of production code.

What This Module Does

This module gives Claude a design step that slots into your app-building workflow. You describe what you're building, pick from 9 curated visual styles, and get a working HTML mockup in minutes — before touching any production code.

It fits naturally into the [/new-app workflow](#):

1. Start `/new-app` — scope your app (name, purpose, screens, data)
 2. Get asked about UI references → pause
 3. Run this skill — describe your app, pick a style, get an HTML mockup
 4. Take the mockup back to `/new-app` — "here are the designs, continue"
 5. `/new-app` builds the full implementation plan with your visual locked in
-

The Example Library

9 real HTML designs Claude reads before building anything:

- creative-coder — Neo-brutalist, bold borders, playful

- **cloakclaw** – Dark hacker, red glows, terminal
- **pony-house** – Cinematic SaaS, 3D hero, atmospheric dark
- **stratum** – Industrial skeuomorphic, metal textures
- **ux-jonny** – Swiss minimal, monochrome, editorial
- **jalapao** – Earthy brutalist, rust-orange, gritty
- **sunplanet** – Editorial luxury, warm cream, restrained
- **rinkuu-gopalani** – Dark luxury, forest-green + rose
- **nexus (8 pages)** – Cinematic dark SaaS, full multi-page site

Claude reads the actual source files and draws from their CSS patterns, component structures, and spatial logic then adapts to your brand and content.

What's Included

- The frontend-design skill (auto-activates when you say "design", "mock up", "build a UI")
- Full example library – all 9 HTML designs Claude reads as reference
- Phase 2: design system extraction – once you pick a mockup, Claude extracts `tokens.css`, `components.css`, and `DESIGN.md` for consistent styling across your entire build

Requirements

- Claude Code CLI
 - No API keys, no scripts – just files
 - Setup time: 3–5 minutes
-

How to Install

Download the ZIP below. Drop it into your 'module-installs' folder and tell Claude Code:

```
"/install frontend-design"
```

Claude walks you through the full installation interactively.

Supadata: YouTube, TikTok & IG Transcript Scraping

Supadata: YouTube Transcripts & Web Scraping

Pull transcripts, search YouTube, scrape web pages. Claude gets internet access via the Supadata API.

What This Module Does

By default, Claude can't pull social media data. This module fixes that. Once installed, Claude can pull content from anywhere, YouTube transcripts, social media metadata, web pages, search results any time you ask.

What Claude Can Do After Installing

- Pull the full transcript of any YouTube, TikTok, Instagram, X, or Facebook video
 - Search YouTube and return results with view counts, durations, upload dates
 - Get channel stats – subscriber count, total views, video count
 - Get metadata from any social post – views, likes, comments, author
 - Scrape any webpage and return clean markdown
 - Extract structured data from video content (free during beta)
-

Example Uses

- "Get the transcript of [competitor's video] and summarise the main points"
 - "Search YouTube for AI agency videos uploaded this month, sorted by views"
 - "Scrape [URL] and give me the key points as bullet points"
 - "Pull the transcript of [video URL] and rewrite it as a Twitter thread"
 - "Get the stats for [channel URL]"
-

What's Included

- The Supadata skill (Claude knows all 21 API endpoints)
- A standalone supadata.py Python client for use in your own scripts
- Step-by-step API key setup

Requirements

- Claude Code CLI
- Python 3 + pip install requests python-dotenv
- Free Supadata account at supadata.ai (free tier available)
- Setup time: 5–10 minutes

Cost

Credit-based. Most operations cost 1 credit. Free plan available. 1 credit = one transcript, one search page, one scrape.

How to Install

Download the ZIP below. Drop it into your 'module-installs' folder and tell Claude Code:

"/install supadata"

Claude walks you through the full installation interactively.

Writing Style: Anti-AI-Slop Enforcement

Writing Style: Anti-AI-Slop Enforcement

Stops Claude writing like an AI. 40+ banned words, 12 core rules, self-check on every output.

What This Module Does

Every piece of writing Claude produces has the same tells: em dashes everywhere, words like "leverage" and "delve", uniform paragraph blocks, filler transitions like "Furthermore." It reads like a language model wrote it. Because one did.

This module fixes that. It gives Claude a concrete ruleset and a self-check protocol it runs before delivering any prose. The result sounds like a sharp human writer, not a chatbot.

What It Enforces

- 12 core rules – no em dashes, vary sentence length, start with content, no binary contrasts, no chatbot artifacts, and more
 - 40+ banned words (Tier 1 – never use): leverage, delve, robust, seamless, transformative, cutting-edge, paradigm, tapestry, synergy, and 30+ more
 - Structural pattern bans – dramatic fragmentation, rule of three, compulsive summaries, heavy signposting
 - 8 Claude-specific tells – epistemic hedging, copula avoidance, over-qualification, balanced framing, meta-commentary, nested clauses, verbosity without substance
 - 8-point self-check protocol Claude runs before every prose output, automatically
-

Applies To Everything

Auto-activates on all text output, strategy docs, marketing copy, newsletters, emails, reports, summaries, content scripts, anything Claude writes.

Requirements

- Claude Code CLI only
 - No API keys, no scripts, no dependencies
 - Setup time: 2 minutes
-

How to Install

Download the ZIP below. Drop it into your 'module-installs' folder and tell Claude Code:

```
"/install writing-style"
```

Claude walks you through the full installation interactively.

Firecrawl: Web Scraping & Search

Firecrawl: Web Scraping & Search

Here's a very capable web scraping system that's worth knowing about.

I did a side-by-side comparison between Claude Code's default web search and WebFetch versus Firecrawl. Firecrawl won. It pulls cleaner content, handles JavaScript-heavy pages that the defaults completely fail on, gets through Cloudflare,

reads PDFs, and returns properly structured markdown that Claude can actually use. I've swapped it out entirely and this is now my main web scraping method.

It does cost credits. That's the trade-off. If you're doing serious research, competitor analysis, or building anything that reads the web, it's worth it. If you're just occasionally asking Claude to look something up, the default tools are fine. Your call.

What It Does

One CLI command turns any URL into clean, LLM-ready content. Eight commands total, each one for a different situation:

- **scrape** — pull a single page to markdown
- **search** — search the web and return full page content, not just snippets
- **crawl** — bulk extract an entire site section (all /docs/, all /blog/, etc.)
- **map** — discover every URL on a domain
- **agent** — AI-powered structured extraction (pricing tiers, contact info, specs)
- **browser** — cloud browser for pages that need clicks, logins, or scrolling
- **download** — save an entire site as local files

It handles the cases the defaults can't: Cloudflare-protected sites, JavaScript-rendered pages, PDFs, and anything that requires interaction to load.

Requirements

- **Node.js 18+ and npm**

- Free Firecrawl account at firecrawl.dev (free tier to get started)
 - Works on macOS and Windows
 - Setup time: 5–10 minutes
-

How to Install

Download the ZIP below. Drop it into your module-installs folder and tell Claude Code:

```
/install firecrawl
```

Claude walks you through Node.js, the CLI install, API key setup, and a live test.

Create New OpenClaw AI Employee: /new-employee

OpenClaw AI Employee

This is how you deploy an autonomous AI worker on its own server.

The idea is simple. You have recurring tasks that eat hours every week: research, monitoring, data collection, competitor tracking. Instead of doing them yourself or asking Claude to do them on demand, you spin up a dedicated agent that runs 24/7 on its own \$6/month server, does the work on a schedule you define, and reports findings back to your workspace automatically. You don't touch it. It just runs.

Each employee is a Claude-powered agent running via OpenClaw on a DigitalOcean VPS. You define the role and the schedule. This module handles everything else: provisioning the server, installing OpenClaw, setting up. Tailscale networking so

reports reach you securely, registering cron jobs, and wiring up a webhook so findings land in your database.

The first one takes about 20 minutes to set up. Each one after that takes under 10.

BE CAREFUL WHAT ACCESS YOU GIVE IT! YOU ARE LIABLE FOR ANY MISTAKES.

What You Get

- A webhook receiver that runs on your machine and accepts reports from deployed agents
 - A provisioning script that creates a DigitalOcean VPS with OpenClaw pre-installed
 - A deploy script that pushes your agent spec, sets up Tailscale, registers cron jobs, and installs research tools
 - A verify script that confirms the full pipeline works end to end
 - Agent spec templates to define what each employee does, how it thinks, and what it reports
 - Database tables to track employees and store their reports
-

Requirements

- Python 3.10+ and pip
- DigitalOcean account (droplet creation, \$6-24/month per employee)
- Anthropic API key (the agent's brain)
- Tailscale account (free, handles secure networking between your machine and the VPS)
- Works on macOS and Windows

- **Setup time: 20-25 minutes first time, under 10 minutes per additional employee**
-

How to Install

Download the ZIP below. Drop it into your module-installs folder and tell Claude Code:

```
/install openclaw-ai-employee
```

Claude walks you through accounts, API keys, SSH setup, the webhook receiver, and deploying your first employee.

Reddit: Community Search & Extraction

Search Reddit for any topic, auto-detect the right subreddits, and extract full threads with scored comments. No API key needed.

What This Module Does

Reddit is where practitioners actually talk. The unfiltered version. Problems they're hitting, tools they've tried, pricing they charge, clients they've lost. You won't find this level of honesty on LinkedIn or Twitter.

This module gives Claude full read access to Reddit via public endpoints. No API key, no developer application, no approval wait. Claude can search across all of Reddit or specific subreddits, auto-detect which communities are relevant for your topic, and pull full threads with every comment scored and ranked.

The research() pipeline does the whole thing in one call: finds the right subreddits, searches them, deduplicates, ranks by engagement, and extracts the top threads with comments. Ask Claude a question and you get back the actual conversation happening around that topic, scored and ranked.

What Claude Can Do After Installing

- Search all of Reddit or target specific subreddits with relevance, top, hot, or new sorting
 - Auto-detect subreddits for any topic (finds the communities you didn't know existed)
 - Extract full threads with post body + top 25 comments, each with scores and author info
 - Run full research pipelines that auto-detect, search, deduplicate, rank, and extract in one pass
 - Browse hot and top posts from any subreddit with time filtering
-

Example Uses

- **"What are people saying on Reddit about AI automation agencies?"**
- **"Search r/SaaS for posts about pricing models this year"**

- "Find the top Reddit threads about Claude Code and summarize the main complaints"
 - "Research what small business owners think about AI tools on Reddit"
-

Requirements

- Python 3 and pip
 - No API key needed (uses Reddit's public .json endpoints)
 - Setup time: 2-3 minutes
 - Cost: Free
-

How to Install

Download the ZIP below. Drop it into your module-installs folder and tell Claude Code:

```
/install reddit
```

Claude walks you through the setup and runs a live test. Takes about 2 minutes since there's no API key to configure.

X-Search: X/Twitter Intelligence

Search X for any topic, get synthesized insights with source tweets, and enrich with engagement data. Two layers: Grok discovers, X API delivers the numbers.

What This Module Does

X is where AI practitioners, founders, and thought leaders post in real-time. Product launches, hot takes, technical debates, hiring signals. If something is happening in AI right now, someone on X is talking about it.

This module gives Claude two-layer access to X. Layer 1 uses Grok's `x_search` (via the `xAI Responses API`) for semantic search. You ask a question in natural language and Grok returns a synthesized answer with cited X posts. Layer 2 uses the X API v2 to pull raw engagement data: likes, retweets, replies, impressions, follower counts. Together they give you both the signal and the numbers.

The `discover_and_enrich()` pipeline combines both layers: Grok finds the relevant posts, then the X API pulls the engagement numbers for each one.

What Claude Can Do After Installing

- Semantic search across X with natural language queries, date ranges, and account filters
 - Get synthesized answers from Grok with cited source tweets
 - Pull engagement metrics for any tweet (likes, retweets, replies, impressions)
 - Look up user profiles with follower counts and verification status
 - Search recent tweets with X API operators (by account, hashtag, media type, language)
 - Run full discover-and-enrich pipelines that find and score relevant posts automatically
-

Example Uses

- "What are people saying on X about AI agents this week?"
 - "Find tweets from @AnthropicAI and @karpathy about Claude Code in the last 30 days"
 - "Search X for reactions to the latest OpenAI release and show me the most engaged posts"
 - "Who are the top voices on X talking about AI automation agencies?"
-

Requirements

- Python 3 and pip
 - xAI API key (required, get at console.x.ai)
 - X Bearer Token (optional, for engagement metrics, get at developer.x.com)
 - Setup time: 5-10 minutes
 - Cost: ~\$0.004-0.01 per Grok search call
-

How to Install

Download the ZIP below. Drop it into your module-installs folder and tell Claude Code:

```
/install x-search
```

Claude walks you through the API key setup, client installation, and a live test.

Academic: Research Paper Search

Search 250 million academic papers, get abstracts, traverse citation graphs, and find free PDFs of paywalled research. Zero auth, completely free.

What This Module Does

When you're building AI systems or researching a technical topic, the real answers are often buried in academic papers. Google Scholar is clunky, paywalls block access, and you can't ask Claude to just "go read the latest research" because it has no way into these databases.

This module connects Claude to OpenAlex, the largest open catalog of scholarly works (250M+ papers). Claude can search by topic, filter by year and citation count, traverse citation graphs to find related work, and even locate free legal PDFs of paywalled papers via Unpaywall. If Firecrawl is installed, it can scrape the full text from open access article pages too.

No API key, no account, no cost. OpenAlex and Unpaywall are open infrastructure funded by grants.

What Claude Can Do After Installing

- Search 250M+ papers by topic with filters for year, citation count, open access, and sort order
 - Get paper details including abstracts, authors, institutions, venues, and DOIs
 - Traverse citation graphs to find papers that cite a given paper (follow the research forward)
 - Find free PDFs of paywalled papers via Unpaywall (legal open access copies)
 - Run full research pipelines that search, rank, find free PDFs, and optionally extract full text
-

Example Uses

- **"Find the most cited papers about AI code generation from 2024 onwards"**
 - **"Search for academic research on LLM hallucination detection and give me the top 5 with abstracts"**
 - **"Find papers that cite 'Attention Is All You Need' and sort by citation count"**
 - **"Research the academic literature on AI agent architectures, find free PDFs where possible"**
-

Requirements

- **Python 3 and pip**
 - **No API key needed (OpenAlex and Unpaywall are free and open)**
 - **Optional: Firecrawl module for full text extraction from article pages**
 - **Setup time: 2-3 minutes**
 - **Cost: Free**
-

How to Install

Download the ZIP below. Drop it into your module-installs folder and tell Claude Code:

```
/install academic
```

Claude sets up the client and runs a test search. No API keys to configure, so it's fast.

Podcast Search: Episode Discovery & Transcripts

Find podcast episodes on any topic across all platforms, rank them by quality, and extract full transcripts from the best ones.

What This Module Does

Some of the best expert content lives in podcast interviews. Long-form, unscripted, full of practitioner war stories and real opinions you won't find in blog posts. But finding the right episode on a specific topic across Apple Podcasts, Spotify, YouTube, and hundreds of podcast sites? That's a pain.

This module gives Claude a two-layer discovery system. Layer 1 searches Google via Firecrawl, finding episodes across every podcast platform. Layer 2 searches YouTube directly via Supadata, which is where most podcast interviews also live. Both layers feed into a quality scoring engine (0-100) that ranks episodes by view count, duration, and source authority. Known high-quality shows like Lex Fridman, a16z, Hormozi, and 25+ others get automatic score boosts.

Once ranked, Claude pulls full transcripts from the top YouTube episodes via Supadata. Ask about a topic, get back the best interviews with full transcripts you can actually work with.

What Claude Can Do After Installing

- Search for podcast episodes on any topic across all major platforms
 - Quality-rank results from 0-100 based on views, duration, and source authority
 - Extract full transcripts from YouTube-hosted episodes automatically
 - Search YouTube specifically for podcast interviews with recency and duration filters
 - Run full research pipelines that discover, rank, and extract transcripts in one pass
-

Example Uses

- "Find the best podcast episodes about AI coding agents and get transcripts from the top 3"
 - "Search for podcast interviews with AI agency founders from the last 6 months"
 - "What are the highest-quality podcast episodes discussing Claude Code?"
 - "Research what podcast experts are saying about AI automation and summarize the key themes"
-

Requirements

- Python 3 and pip
 - Firecrawl module installed (for Google search discovery)
 - Supadata module installed (for YouTube search and transcript extraction)
 - Setup time: 5 minutes
 - Cost: Uses Firecrawl credits for search + Supadata credits for transcripts
-

How to Install

Download the ZIP below. Drop it into your module-installs folder and tell Claude Code:

```
/install podcast-search
```

Claude checks that Firecrawl and Supadata are already set up, then installs the podcast search client and skill.

Substack: Newsletter Search & Extraction

Search Substack newsletters on any topic, discover publications, and extract full article content with metadata. No API key needed.

What This Module Does

A lot of the sharpest analysis in AI, business strategy, and product thinking lives on Substack. Ethan Mollick, Gergely Orosz, Ben Thompson, and hundreds of others publish long-form, experience-based writing there that you won't find anywhere else.

This module gives Claude two-layer access to Substack. Discovery happens through Firecrawl (Google search scoped to substack.com), which finds relevant posts across all publications. Content extraction uses Substack's public API endpoints to pull full article text, word counts, reaction counts, and comment counts. No Substack API key required.

The research() pipeline handles everything: searches for posts on your topic, identifies the best matches, extracts full content from the top results, and returns it all in one structured response.

What Claude Can Do After Installing

- Search all of Substack for posts on any topic via Google
 - Extract full article content as clean plain text (HTML stripped automatically)
 - Browse publication archives with sorting by newest or top posts
 - Discover publications writing about a specific topic
 - Get metadata for every post: word count, reactions, comments, audience type (free vs paid)
 - Run full research pipelines that search, discover, and extract in one pass
-

Example Uses

- **"Search Substack for articles about AI agents and summarize the top 3"**
- **"What has Ethan Mollick written about AI coding tools recently?"**
- **"Find Substack newsletters covering the AI agency space"**
- **"Research what Substack authors are saying about AI operating systems"**

Requirements

- Python 3 and pip
- Firecrawl module installed (for Google search discovery)
- No Substack API key needed (uses public endpoints)
- Setup time: 3-5 minutes
- Cost: Firecrawl credits for search only (Substack extraction is free)

How to Install

Download the ZIP below. Drop it into your module-installs folder and tell Claude Code:

```
/install substack
```

Claude checks Firecrawl is working, installs the Substack client and skill, and runs a test search.

Deep Research: Multi-Platform AI Research System

Launch parallel AI agents that search 7 platforms simultaneously, score every source, cross-reference findings, and produce a critically reviewed synthesis report.

What This Module Does

This is the big one. A full multi-agent research system that runs across 7 platforms in parallel.

When you run /deep-research, here's what happens. First, a fast recon agent scans all 7 platforms to find where the signal lives for your topic. You review the recon and confirm the plan. Then parallel topic agents (each running on Claude Opus) go 3+ levels deep on specific angles. Every source gets scored for recency, credibility, specificity, and independence. Claims get triangulated across platforms. A critic agent reviews all reports for quality issues, echo chambers, and unsupported generalizations. Finally, a synthesis agent produces the master report with confidence labels on every finding.

The whole thing runs in 15-30 minutes depending on scope. You get a folder of reports: per-agent deep dives, critic notes, and a master synthesis. What you'd normally spend days on, done in half an hour.

The 7 Research Platforms

- Firecrawl (web) – articles, documentation, competitor pages, blog posts
- Supadata (video) – YouTube transcripts, TikTok, Instagram, video metadata
- Reddit (community) – practitioner discussions, real opinions, subreddit discovery
- X-Search (real-time) – what thought leaders and practitioners are saying right now
- Academic (papers) – 250M+ scholarly papers, citation graphs, free PDFs
- Podcast Search (audio) – expert interviews, quality-ranked, with full transcripts
- Substack (newsletters) – long-form expert analysis and thought leadership

How It Works

The research runs in phases:

1. **Scope Interview** – Claude asks 3-4 questions to understand what you're researching, how recent it needs to be, and whether to go broad or deep
2. **Recon Agent** – a fast Sonnet agent scans all platforms in 2-3 minutes and reports where the signal is (HOT / WARM / COLD per platform)
3. **Human Checkpoint** – you review the recon and confirm or adjust the agent roster
4. **Topic Agents** – 3-6 parallel Opus agents each go deep on a specific angle for 10-20 minutes. They use all 7 platforms, score every source, and chase signal 3+ levels deep
5. **Critic Agent** – reviews all reports for echo chambers, unsupported claims, missing profiles, recency bias, and cross-report contradictions
6. **Synthesis Agent** – produces the master report with confidence-labeled findings, tensions, gaps, key people, and a source registry

What Gets Installed

This is a big module. It installs everything you need for full multi-platform research:

- 5 research platform clients – Reddit, X-Search, Academic, Podcast Search, Substack (full Python clients with all methods)
- 5 matching skill files – so Claude knows how and when to use each platform
- The Deep Research orchestrator skill – the main engine that runs the multi-phase pipeline
- 4 prompt templates – recon, topic-agent, critic, and synthesis (these are the prompts that make the agents actually good)

- The `/deep-research` command – so you can trigger it with a slash command

Firecrawl and Supadata are prerequisites (install them first from the Module Library if you haven't already). The other 5 platforms are bundled in.

Example Uses

- "Run deep research on AI coding agents – who's building them, what's working, what's breaking"
 - "Deep research on the AI automation agency market – pricing, positioning, competition"
 - "Research everything about [person's name] – their work, influence, and what people say about them"
 - "Investigate the state of AI video generation tools – practitioner sentiment, academic frontier, market landscape"
-

Requirements

- Python 3 and pip
- Firecrawl module installed (from the Module Library)
- Supadata module installed (from the Module Library)
- xAI API key for X-Search (get at console.x.ai)
- X Bearer Token (optional, for engagement data, get at developer.x.com)
- Setup time: 10-15 minutes
- Cost: Firecrawl and Supadata credits for each research run, plus ~\$0.01 per X-Search call. Reddit, Academic, Substack, and Podcast Search are free. Claude API usage for running the agents.

How to Install

Download the ZIP below. Drop it into your module-installs folder and tell Claude Code:

```
/install deep-research
```

Claude walks you through the full setup: prerequisites check, API key configuration, all 5 research clients, the orchestrator, and a test run. It's the longest install in the Module Library but Claude handles every step.

What You're Building: Become the Architect

Layer 5 - Build: Become the Architect

You've built the platform. Audited your tasks. Grabbed everything off the shelf. What's left on your Task Audit is unique to YOUR business. Time to build custom.

What You're Building

Layers 1-3 gave you the platform. Layer 4 mapped what needs automating and gave you a system to manage the work. The Module Library (Section 7) handled the common problems. But some tasks on your Task Audit are unique to YOUR business - your specific workflows, your specific integrations, your specific needs. Nobody else has built a module for those. That's what Layer 5 is for.

This is where you become an Architect. You use structured commands to brainstorm what to build, explore how to build it, plan it out, and execute. The same workflow used to build every module in this course.

What You'll Have After

- **Two new commands: /brainstorm (figure out what to build) and /explore (shape a specific idea)**
 - **A structured workflow for going from "I want to automate X" to a working system**
 - **The methodology for building custom automations - properly explored, planned, and documented**
 - **More items ticked off your Task Audit - your Task Automation % keeps climbing**
-

The Shift

Up until now, you've been following instructions, install this module, run this command, follow the steps. Layer 5 is different. This is where you're not following instructions anymore. You're building. You're designing automations that don't exist yet. You're solving problems nobody else has solved.

And here's the thing: by the time you've built through four layers, you're technical - without even trying. The gradual exposure has made you capable of things you couldn't imagine when you started. Trust the process. You're ready.

/explore & /brainstorm Slash Command Toolkit

Slash Command Toolkit

Two commands that give you a structured way to go from "I want to automate X" to a scoped plan you can execute.

What This Is

The Slash Command Toolkit adds two commands to your workspace designed specifically for custom building:

- **/brainstorm** – For when you don't know where to start. It reads your Task Audit (context/task-audit.md), your business context, and your data. Helps you figure out what's possible, what's worth automating, generates options and approaches. "I have 15 tasks left on my audit and I don't know which one to tackle first." Brainstorm helps you explore options, weigh trade-offs, and pick the highest-impact thing to build next.
 - **/explore** – For when you have a specific idea. "I think I want to automate my client onboarding - help me explore what that looks like." Interactive back-and-forth across 5 stages: Discovery → Research → Shape → Scope → Output. Produces a scoped exploration document you can feed into **/create-plan** and then **/implement**.
-

Why This Matters

Building custom automations without a structured process leads to scope creep, wasted time, and abandoned projects. You start excited, get 30% in, realize you didn't think something through, and give up. These commands prevent that by giving you a disciplined workflow:

1. **`/brainstorm`** – figure out **WHAT** to build (or ask for help deciding)
2. **`/explore`** – shape the idea, research feasibility, define scope
3. **`/create-plan`** – turn the exploration into a **step-by-step** plan
4. **`/implement`** – execute the plan with validation at each step

This is the same workflow used to build every module and system in this course. It works.

Install It

1. **Download the ZIP below**
2. **Unzip into your `module-installs/` folder**
3. **Run: `/install module-installs/slash-commands`**
4. **Follow Claude's setup, it installs the commands and verifies they work with your workspace context**
 - **Prerequisites: ContextOS (so the commands have business context to work with)**
 - **Setup time: 5-10 minutes**
 - **API keys: None**
 - **Cost: Free**

[slash-commands-v1.zip](#) 11.89 KB

Verify It Worked

Run `/brainstorm` - Claude should read your Task Audit and business context, then start helping you figure out what to tackle next. If you have a specific idea, try `/explore` and describe it, you should enter an interactive exploration session that takes you through discovery, research, and scoping.

Next lesson: Building Custom Automations - the full workflow for turning your explorations into working systems.

Building Custom Automations

Building Custom Automations

The methodology for going from idea to working system. Properly explored, properly planned, properly built.

The Workflow

Every custom automation follows the same 4-step process:

1. `/brainstorm` – What should I build? Claude reads your Task Audit, your business context, and helps you pick the highest-impact automation to tackle next. Generates options, weighs trade-offs, and recommends where to start.
2. `/explore` – Shape the idea. Interactive 5-stage exploration: Discovery (understand the problem), Research (check what's possible), Shape (design the approach), Scope (define what "done" looks like), Output (produce a scoped document). Takes 10-20 minutes per idea.

3. **/create-plan** – Plan the build. Takes your exploration doc and turns it into a step-by-step implementation plan. Clear chunks, clear validation criteria, clear order of operations.
 4. **/implement** – Build it. Executes the plan chunk by chunk. Each step is validated before moving to the next. When it's done, you have a working automation.
-

The Autonomy Spectrum

Not every automation needs to be fully autonomous from day one. Think of it as a spectrum:

- **Interactive** – You trigger it, Claude helps you do it. (Like **/brainstorm**, you run the command, interact with it, get output.)
- **Scheduled** – Runs at set times automatically. (Like your **Daily Brief** - fires every morning via cron, delivers the result without you lifting a finger.)
- **Triggered** – Runs in response to an event. (Like a **file watcher** - when a new recording appears in a folder, the pipeline kicks off automatically.)
- **Autonomous** – Makes decisions and acts on its own within defined boundaries. (Like your **Telegram bot** - receives messages, processes them, responds, all without your involvement.)

Start interactive. Prove it works. Then decide if you want to schedule it, trigger it, or make it autonomous. You can always move along the spectrum later.

Cron Jobs and Scheduling

When you're ready to make something run automatically, you set up a cron job (or launchd on Mac). This is how your Daily Brief runs every morning, how your data collectors fire daily, how your AI Landscape Monitor scans overnight.

Claude can help you set this up. Just say: "I want this to run every morning at 7am" and Claude will create the scheduling configuration for your platform.

Testing and Documentation

Every automation you build should be tested before you trust it. Run it manually a few times. Check the output. Make sure it handles edge cases. Then automate the scheduling.

And document your HISTORY.md and /commit workflow handle this automatically. Future you (or future Claude sessions) will thank you for the documentation.

The Scoreboard

Every custom automation you build = another task permanently off your plate. Go back to your Task Audit (context/task-audit.md), tick off what's been handled, note what system solved it. Watch your Task Automation % climb.

This is the game. Each task automated is bandwidth recovered, permanently. And the more you build, the better you get at building. The compound effect is real.

Next lesson: Share What You Build - your custom automation might become the next module in the library.