CLOCKWORK

# CLOCKWORK RECRUITING AI PROMPT PLAYBOOK

This AI prompt playbook gives you a brief overview and examples of prompts that you can use to leverage Clockwork + your preferred AI Assistant (Claude, ChatGPT) and connect to Clockwork Recruiting's MCP (Model Context Protocol) server.

## + MCP Server Overview + Reference

A brief overview of what an MCP Server is, and how it connects your data in Clockwork to your preferred AI Assistant.

## + Quick Start Guide - MCP Server

Get instructions on how to install the MCP Server for your AI Assistant and securely connect your Clockwork data.

## + How To Use This Playbook

A brief introduction on how to use this playbook and an introduction to the prompts that are included.

## + Prompt Library

A library of prompts for getting started using your AI Assistant with your Clockwork Data.

## + Frequently Asked Questions

Frequently asked questions about the MCP server and what data can be retrieved from Clockwork.

# What Is an MCP Server? Why Does It Matter for Your Executive Search Firm and CRM?

MCP stands for Model Context Protocol. Think of it as a standard, secure connector that lets an AI Assistant, like Claude or ChatGPT, talk directly to a piece of software—like your executive search CRM—instead of just reasoning about it in the abstract or making sweeping assumptions from your prompts. When connected to Clockwork's MCP Server, your AI Assistant can securely retrieve live recruiting data directly from Clockwork instead of relying only on information you manually provide. With an MCP server, that same request can:

- Look up a live candidacy status instead of you copying and pasting it in
- Pull real project, company, client, and search data to answer accurately, not guess from a prompt
- Take an action in a connected system, like drafting a note or generating a status report, based on real data and records
- Check across multiple systems in one request—CRM, calendar, email inbox—for a fuller picture
- Work the same way across AI Assistants, standardizing the way data is read and ensuring consistent context, since an MCP server is an open standard, not a one-off integration
- Stay current automatically, since it's querying live data instead of a stale snapshot

An MCP server is built by the platform provider—in Clockwork's case, by us—and it exposes a structured, permission-based set of your firm's search data. You control what's connected and who has access, and the AI Assistant doesn't get a copy of your database; it asks for only the data it needs, when it needs it, through the same permissions your team already operates under.

## Why This Matters More in Executive Search Than Almost Any Other Industry

Retained executive search runs on relationship intelligence and real-time data like:

- Who's been placed where
- Who's sitting on a board
- Which client is mid-search and needs an update today
- Which candidates have quietly moved on to new roles

The web has made basic information more accessible than ever. Anyone can search a name or a company and get a reasonable answer in seconds—your competitors included. But that public layer isn't where the advantage lives anymore. The value lives in what you layer on top of it: the notes, touchpoints, past search history, and proprietary relationship intelligence sitting inside your executive search CRM. That context was never public to begin with, and no general search or chatbot has access to it.

This is exactly where an MCP server changes the equation. An AI Assistant without a connection to an MCP server can only work with the public layer—it's still guessing at the rest. An AI Assistant connected through an MCP server combines both: the broad, real-time information available on the web with the deep,

proprietary intelligence unique to your firm. That combination is what turns your AI Assistant from a generic research assistant into a genuinely strategic one—working from real data, producing more reliable outputs, and generating deeper insights.

Here are a few concrete ways this shows up in practice for executive recruiters:

- **Faster, Custom Reports.** Instead of a researcher manually assembling a project, candidate, or status update, an AI Assistant connected through an MCP server can pull live candidacy stages, notes, and timelines directly from the platform and draft the report in minutes. You can also load custom, branded report templates directly into your AI Assistant to standardize and generate client-ready outputs.
- **Clearer Business Development and Meeting Prep.** Before a client call, an AI Assistant connected through an MCP server can check prior project history, past placements, relationships, and open searches for that company, grounding the conversation in real data instead of a generic summary.
- **Less Manual Data Entry**. Because an AI Assistant can read from and write back to your CRM through a governed connection via an MCP server, your team spends less time re-typing information that already exists somewhere in the system.
- **Consistency Across a Growing Team.** As firms add contractors, researchers, and associates, an MCP server gives everyone the same accurate, real-time view of search data—without needing platform training to get there.

## What an MCP Server Is Not

It's worth being precise here about what an MCP server specifically is, because there are a lot of assumptions made about it in general and how it's used. An MCP server is not a chatbot, and it's not a new AI model—it's a protocol: a common language that lets an AI Assistant, like Claude or ChatGPT, and another platform exchange and act on data in a structured, permissioned way. Think of it like a shared map for an AI Assistant, where properties, fields, and data are organized into a broader set of directives for your AI Assistant to work from.

An AI Assistant does the reasoning. An MCP server does the connecting. Your executive search CRM stays the central system of record, where all the work gets done.

This distinction matters for firms evaluating AI Assistants right now. Plenty of AI Assistants can summarize a document or draft an email without an MCP server. But far fewer can see and reach your actual executive search data and work from what's really there. That gap is what an MCP server closes.

We built Clockwork's MCP server because this is the direction the search industry is heading: AI Assistant working agentically, directly inside the systems, data, and processes recruiters already rely on. For specialized retained search firms, that means less time spent translating information between tools and more time spent on the work that actually wins new searches—client relationships, candidate assessment, and the judgment calls no model can make for you.

CLOCKWORK

# Clockwork MCP Tool Reference Guide

Everything the Clockwork connector exposes, grouped by what it touches. Each tool also accepts response_format='json' for machine-readable output and most list tools paginate (limit/offset). You don't need to memorize these — the prompts call the right ones — but this is a fundamental working list of what your AI Assistant can retrieve.

## Session & identity

- clockwork_get_knowledge — Loads Clockwork's data dictionary, behavioral rules, gap atlas, and annotated sample responses. Call once at the start of every session (business-rules and domain at minimum).
- clockwork_get_current_user — Resolves who "I" am for "my projects" / "my candidates" queries. Reads the CLOCKWORK_CURRENT_USER_ID env var or takes an email/name to look up.

## Projects & searches

- clockwork_list_projects — Lists searches with filters: status, user_id/user_email project_type. Supports include (client_company, lead, owner, placement, project_type, closing_reason, tags).
- clockwork_get_user_projects
- clockwork_get_project
- clockwork_get_project_team
- clockwork_get_project_clients
- clockwork_get_project_compensation
- clockwork_get_project_fee
- clockwork_get_project_payments
- clockwork_get_project_location

## Candidacies

- clockwork_list_candidacies — Two modes: a project's full pipeline (project_id), or all of one person's candidacies across every project (person_id). Includes person, status, overview/assessment/next-steps notes, and candidacy_stat.
- clockwork_get_candidacy
- clockwork_get_candidacy_status_history — The full stage-by-stage transition history for a candidacy: from-status → to-status, who changed it, and when. Answers "when did they reach interview?" and "how long in each stage?"
- clockwork_get_candidacy_notes — Notes on a candidacy with date-window filtering (begin_time/end_time) and include=author.

CLOCKWORK

# Clockwork MCP Tool Reference Guide

## People

- clockwork_search_people — Full-text search across names, titles, companies, email, phone, tags, and résumé content. Supports a date-added window.
- clockwork_list_people
- clockwork_get_person
- clockwork_get_person_details — Deep sub-resources for a person including all notes and attachments

## Companies

- clockwork_list_companies
- clockwork_get_company

## Conflict & reference data

- clockwork_run_conflict_check — One-call off-limits check on a person: returns a **GO / CAUTION / STOP** verdict with evidence (do-not-contact, prior placement inside the no-poach window, active involvement on another search).
- clockwork_list_reference_data — The lookup tables. Fetch once per session and cache.

CLOCKWORK

# Quick Start Guide - MCP Server

To get started, make sure that you have properly set-up your AI Assistant by connecting Clockwork's MCP Connector / Server within your AI Assistant. Directions for doing so are linked below:

→ MCP Server: Claude Setup
→ MCP Server: ChatGPT Setup

The following prompts are broken down into several categories for quicker referencing. Each section will begin with a list of prompts and their summaries.

## Daily Workflows

Prompts intended to run every morning (or on demand throughout the day).

+ Daily Project Pulse
+ Daily Meeting Prep
+ Smart Inbox Triaging For Executive Recruiters

## Pipeline and Project Health

Prompts focused on the health of active searches: pipeline depth, stalls, data completeness, etc.

+ Weekly Pipeline Health Report
+ Stalled Candidacy Recovery
+ Diversity Slate Audit
+ Project Candidate Completeness Audit

## Client & Candidate Communication

Prompts that draft or prepare external communication — to clients, candidates, and references.

+ Client Update Email Draft
+ Candidate Outreach Sequencer
+ Reference Call Prep Sheet
+ Project Status Report [Template Builder]
+ Weekly Project Status Report [Generator]
+ Candidate Notes → Client Facing Overview

## Search Execution & Research

Prompts for launching a search, looking people up, mapping a market, and checking conflicts.

+ Search Kickoff Brief
+ Market Map Generator
+ Person Brief 360
+ Phone Number Lookup
+ Off-Limits & Conflict Checks

## Business Development

Prompts focused on generating new business from existing relationships in Clockwork.

+ Generate Leads / Companies to Target

## Team Operations, Performance & Capacity

Prompts focused on the operational health, team performance, who is doing what in Clockwork, and how capacity is distributed.

+ Team Activity Leaderboard
+ Monthly Clockwork Activity Report By User
+ Team Data Hygiene Scorecard
+ Partner Performance Scorecard by Project Outcome
+ Team Capacity & Workload Comparison
+ Monthly Practice Review
+ Project Efficiency Report

CLOCKWORK

# How To Use This Prompt Playbook

Treat each prompt as a starting point. The “Try this next...” note for each prompt lists additional ways or suggestions to further refine and retarget the prompt. The more specific you are about project names, time windows, and tone, the better the output.

**Start every session within your AI Assistant by loading the knowledge base.** The first Clockwork tool call should be:

clockwork_get_knowledge(topic='business-rules') then clockwork_get_knowledge(topic='domain')

Prompts with a "Session setup" block include this as step 0; for others it happens automatically via the tool's built-in reminder. Load once per session — the content is stable.

**Establish your identity once.** For "my projects" / "my candidates" prompts, resolve who you are at session start via clockwork_get_current_user (or set CLOCKWORK_CURRENT_USER_ID in your system prompt). Then list_projects (user_id=<me>) and get_user_projects return your book in a single call.

**For recurring prompts** (Daily, Weekly, Monthly), consider scheduling them or creating a skill in your AI Assistant so the output is waiting in your inbox or Slack each morning.

**Review the first few runs of any new prompt closely** and tune the output format to your taste — small wording changes compound over time, and **you should always check an AI Assistant’s work.**

**On client-facing outputs:** any prompt that generates client content will confirm the stoplightStatus visibility filter before running. Answer once per session — your answer carries through that run and is re-confirmed next time.

**On project display:** all prompts use the <Company> \ <Project> format (e.g., "Mission \ VP of Finance"), matching the Clockwork UI. Confidential searches display as [Confidential] \ <Role Title> in internal contexts.

**On response format:** Clockwork tools support response_format='json' for machine-readable output. For prompts that build lookup maps, compute aggregates, or feed multi-step calculations, request json on those calls. For final outputs you're reading directly, the default markdown is fine.

**Mix and match.** Many prompts chain naturally — a conflict check before an outreach sequence, a market map feeding a kickoff brief, or a completeness audit before a client update. The tools are building blocks; combine them to fit your workflow.

**Use and load branded templates for reports.** When running prompts to deliver client-facing deliverables, upload your firm’s template for the specific report when prompting.

CLOCKWORK

# Clockwork Recruiting - AI Assistant Prompt Playbook

Below is a working library and examples of prompts that executive recruiters and their teams can use with Clockwork Recruiting and their AI Assistant. Each prompt is designed to be pasted into a chat with an AI Assistant that has the Clockwork Recruiting connector (MCP Server) enabled and, where noted, additional connectors such as Google Calendar, Gmail, Google Drive, Slack, LinkedIn, etc.

These prompts are meant to serve as a starting point for leveraging an AI Assistant with Clockwork's MCP server and your Clockwork data - not finished scripts or a finite list of what is possible. Every prompt is written to leverage your Clockwork data but is intended to be adapted to your firm or use case.

For example, change the time window, the project name, the tone, the output format, or the scope and the same prompt does a different job with relevant outputs depending upon your request. Each prompt is designed to be used within your preferred AI Assistant after connecting Clockwork's MCP Server. Each prompt section below includes:

1. **Title** - the action you are looking to prompt your AI Assistant with
2. **Description** - a summary of what you are prompting your AI Assistant to execute
3. **Frequency** - a suggested time-frame for running this prompt
4. **Connectors used** - tools you may want to connect for more seamless outputs
5. **Prompt** (copy/paste ready) - The prompt itself along with the steps to refine and define the directive for your AI Assistant.
6. **Try this next...** - a note listing some of the different directions you can take the prompt.

Treat the following prompt library as a set of building blocks you can combine to fit how you run your executive search process and the necessary actions you can take leveraging your AI Assistant and Clockwork. **Make sure that when copy and pasting the prompt into your AI Assistant, that you insert the appropriate project, candidate, client name or information, etc.**

For using the prompt itself, we recommend copying the text within the section that is outlined in grey. This will include the top-level prompt as well as a series of steps and directives to provide your AI Assistant additional instruction and context.

CLOCKWORK

# AI PROMPTS FOR DAILY WORKFLOWS

Prompts intended to run every morning (or on demand throughout the day) to set the partner up for the day's meetings, calls, and inbox, including:

## + Daily Project Pulse

A 24-hour change report across your active retained searches, surfacing pipeline movement, stage changes, stalls, and items needing attention.

## + Daily Meeting Prep

A one-paragraph prep brief for every meeting on today's calendar, enriched with Clockwork candidate/client context, recent email threads, and relevant docs.

## + Smart Inbox Triaging For Executive Recruiting

Scans my inbox each morning, cross-references senders against Clockwork, and tells me which emails matter most and why.

CLOCKWORK

# Prompt: Daily Project Pulse

**Description:** A 24-hour change report across your active retained searches, surfacing pipeline movement, stage changes, stalls, and items needing attention.
**Frequency:** Daily (suggested 7:30am local)
**Connectors:** Clockwork Recruiting

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

Using Clockwork Recruiting, generate my Daily Project Pulse for the last 24 hours.

**Session setup (do once, cache for this session):** (0) Call clockwork_get_knowledge(topic='business-rules') and clockwork_get_knowledge(topic='domain') if not already done this session, and resolve my identity via clockwork_get_current_user. Then fetch and cache: (a) firm users via list_reference_data(data_type=users) — keep only Partner, Recruiter, Researcher, Admin roles; build a userId → name map for note attribution; (b) statuses via list_reference_data(data_type=statuses) → statusId → {name, category, rank} map; (c) note categories via list_reference_data(data_type=note_categories) → categoryId → name map.

**Scope:** Pull my active retained searches in one call — list_projects(user_id=<me>, status=active, project_type="Retained", include=lead,owner,client_company). Filter out isInternal = true projects.

For each project, output using the <Company> \ <Project> display format:

- **<Company> \ <Project>** — current owner/lead
- **Candidacies updated in last 24h** (name, status.name as current stage, who last touched it) — filter status.rank < 100 to exclude references and benchmarks
- **Stage changes in the period** — for any candidacy updated recently, call get_candidacy_status_history and report concrete moves ("Screened → Client Interview, moved by [user] yesterday"). This is now exact, not inferred.
- **Candidates removed from consideration** (status.category starts with out-) updated in the period (name, reason if captured in notes)
- **New candidacy notes** (1-line summary each, author from cached map, timestamp) via get_candidacy_notes with a 24-hour begin_time/end_time window and include=author
- **Stall flag**: mark any project where no candidacy has been updated in 7+ days
- **Client visibility**: note each updated candidacy's stoplightStatus so you know what's shareable

End with a **"Needs My Attention"** section listing the top 3 items to act on today and why. When ranking urgency, candidates with higher status.rank (farther along) typically warrant attention first.

Output format: Markdown, grouped by project using <Company> \ <Project> headers, sorted by stall risk (highest first).

## Try this next...

Narrow to a single client or search; widen the window to 48–72 hours for a post-weekend catch-up
Add a financials line per search (get_project_fee) to keep revenue in view
Restrict to searches in client-interview stage and beyond (status.rank >= 1300) for an executive-only version
Have it post the summary to Slack instead of returning it inline

CLOCKWORK

# Prompt: Daily Meeting Prep

**Description:** A one-paragraph prep brief for every meeting on today's calendar, enriched with Clockwork candidate/client context, recent email threads, and relevant docs.
**Frequency:** Daily (suggested 6:30am local)
**Connectors:** Calendar, Clockwork Recruiting, Gmail / Outlook, Shared Drive

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

Build my Daily Meeting Prep for today.

Step 1 — Google Calendar: Pull every meeting on my calendar for today, in chronological order. Skip blocks marked "Focus" or "OOO".

Step 2 — For each meeting, identify all external attendees (ignore my own domain).

Step 3 — Clockwork Recruiting lookup for each external attendee. Find them with search_people (by name or email), then use list_candidacies(person_id=...) to pull their full cross-project history in one call:

- Are they a candidate on any active search? If yes: <Company> \ <Project>, candidacy status.name, last note, owning partner.
- Are they a client contact? Cross-reference with get_project_clients on my active searches, or match by their employer being the client company. If yes: <Company> \ <Project>, role being filled, project status, and the 2 candidates farthest along (name + status.name, sorted by status.rank descending).
- Before suggesting any follow-up outreach, check doNotContact on their person record. If true, note the flag instead of suggesting outreach.
- If neither, note "No active Clockwork relationship" but include any prior candidacies or person notes from the last 12 months.

Step 4 — Gmail: Surface the 3 most recent email threads between me and any attendee (subject + 1-line summary + date).

Step 5 — Google Drive: Search for documents referencing the attendee's company or the meeting topic. Return the top 3 with links.

Step 6 — For each meeting, write a one-paragraph prep: who's in the room, the likely purpose, the 1–2 things I should be ready to discuss, and any open loop from prior interactions.

Output format: Markdown, one section per meeting with header ## [Time] — [Meeting Title], attendees line, then the prep paragraph. Save the file to /Daily Meetings/Meeting - YYYY-MM-DD.md.

## Try this next...

Run it for a single upcoming meeting instead of the whole day
Add a "talking points" or "questions to ask" block per meeting
Pull résumés (get_person_details(detail_type=attachments)) for candidate interviews
Include comp context (get_project_compensation) for offer-stage conversations
Generate it the night before for an early start

CLOCKWORK

# Prompt: Smart Inbox Triaging For Executive Recruiting

**Description:** Scans my inbox each morning, cross-references senders against Clockwork, and tells me which emails matter most and why.
**Frequency:** Daily
**Connectors:** Gmail / Outlook, Clockwork Recruiting

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

Pull all unread emails from my Gmail inbox received since yesterday 5pm. For each sender, look them up in Clockwork Recruiting with search_people (by email), then use list_candidacies(person_id=...) to pull their relationships across all searches in one call.

For each matched person, check doNotContact. If true, flag the email "DNC — review before responding" and do not suggest outreach.

Categorize each email: Active candidate, Active client, Past placement, BD target, Internal team, Other.

For the first four categories, surface: the relevant search(es) as <Company> \ <Project>, candidacy status.name or client role, and a one-line "why this matters today". Suggest a recommended action (Reply now / Schedule reply / Delegate / Archive) and, where useful, draft a 2–3 sentence reply.

Output as a markdown triage list, sorted by priority. Skip newsletters and automated notifications.

## Try this next...

Limit to a single label or VIP-sender list
Auto-draft replies for every "Reply now" item; widen to the last 24–48 hours
Fold in calendar context so emails tied to today's meetings rank highest

CLOCKWORK

# AI PROMPTS FOR PIPELINE AND PROJECT HEALTH

Prompts focused on the health of active searches: pipeline depth, stalls, slate quality, and data completeness including:

## + Weekly Pipeline Health Report

Roll-up of every active retained search showing pipeline depth, stage distribution, movement, and at-risk searches.

## + Stalled Candidacy Recovery

Identifies candidates stuck in the same stage too long and proposes specific next actions.

## + Diversity Slate Audit

Snapshot of pipeline diversity across active searches to ensure slates meet firm and client commitments.

## + Project Candidate Completeness Audit

For a single project, audits every in-consideration candidate against a data-completeness checklist so the team knows which records need work before the next client update.

CLOCKWORK

# Prompt: Weekly Pipeline Health Report

**Description:** Roll-up of every active retained search showing pipeline depth, stage distribution, movement, and at-risk searches.
**Frequency:** Weekly (Monday morning)
**Connectors:** Clockwork Recruiting

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

Using Clockwork Recruiting, build a Weekly Pipeline Health Report for all my active retained searches.

Session setup (do once, cache for this session): (0) Load business-rules and domain knowledge if not already done, and resolve my identity via clockwork_get_current_user. Then cache: (a) firm users (Partner/Recruiter/Researcher/Admin only) → userId → name; (b) statuses → statusId → {name, category, rank}; (c) note categories → categoryId → name.

Scope: list_projects(user_id=<me>, status=active, project_type="Retained", include=lead,owner,client_company). Filter out isInternal = true.

For each project, show (display as <Company> \ <Project>): client company, role, project age (weeks since startedAt), total candidacies (excluding rank < 100), count by status.category group (in-outreach, in-contenders, in-offered, in-placed, out-*, research-*), with status.name shown per candidate in the detail view. Also show: candidacies updated this week, the most recent note date across the project, and — using get_candidacy_status_history — any candidates who changed stage this week (concrete from → to moves).

Add a stoplightStatus column: count of green (shareable) / red (hidden) / none (not yet set), so you can see at a glance which slates are ready to share.

If the user asks about candidate color priorities, ask how the firm uses the color slots before filtering or grouping by colorCategory.

Then produce three roll-up sections:

- Healthy (advancing on schedule)
- Watch (slow movement or thin pipeline)
- At Risk (no client-facing activity in 14+ days OR fewer than 3 candidates at in-contenders or higher)

Sort projects by <Company> \ <Project> within each section. Close with a 5-bullet executive summary I can paste into my partner email.

## Try this next...

Run it for one partner's book or one client
Switch the cadence to biweekly or monthly
Add average days-in-stage per search (derived from status history) to spot where candidates bottleneck
Layer in fee/payment status (get_project_fee, get_project_payments) for a combined pipeline-and-revenue view
Output just the At-Risk section as a focused triage list

CLOCKWORK

# Prompt: Stalled Candidacy Recovery

**Description:** Identifies candidates stuck in the same stage too long and proposes specific next actions.
**Frequency:** Weekly
**Connectors:** Clockwork Recruiting

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

Using Clockwork Recruiting, find every candidacy across my active searches where neither the candidacy record nor any note has been touched in 10+ days.

Pull my searches with list_projects(user_id=<me>, status=active, include=client_company). For each, list candidacies and pull the most recent note per candidacy. To pinpoint exactly how long a candidate has been in their current stage, use get_candidacy_status_history — the most recent transition date is the true stage-entry date.

Filter out candidacies with status.rank < 100 (references and benchmarks).

For each stalled candidacy, output: <Company> \ <Project>, candidate name, status.name (current stage), days in current stage (from status history), days since last note, last note (date + author + 1-line summary), and a suggested next action (e.g., "Nudge client", "Re-engage candidate", "Move to declined").

Sort by days-since-last-activity descending. Group by project.

## Try this next...

Tighten or loosen the threshold (7 vs. 14 days)
Scope to a single search before a client call
Focus only on client-facing stages (status.rank >= 1300) where stalls cost the most
Have it draft the nudge emails/notes for the top stalled candidates

CLOCKWORK

# Prompt: Diversity Slate Audit

**Description:** Snapshot of pipeline diversity across active searches to ensure slates meet firm and client commitments.
**Frequency:** Weekly
**Connectors:** Clockwork Recruiting

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

Method note: Clockwork has no first-class diversity field. This prompt works when your firm captures diversity via tags or custom fields with a known convention. Confirm the convention with me first (you can list available tags via list_reference_data(data_type=tags) and custom fields via data_type=firm_fields), and use only those captured attributes — never infer from name, photo, or background.

Using Clockwork Recruiting, audit the diversity composition of every active retained search where I'm a team member. Pull searches with list_projects(user_id=<me>, status=active, project_type="Retained", include=client_company).

For each project, show (display as <Company> \ <Project>): total candidates with status.rank >= 1100 (in active consideration), count and percent by each captured diversity dimension, and whether the slate meets the project's stated commitment (if recorded in the project brief).

Filter out candidacies with status.rank < 100.

Flag any project where the slate at status.rank >= 1300 (client-interview stage and beyond) falls below the commitment, and suggest specific re-balancing actions. Note any data gaps where diversity fields are blank so we can improve capture.

## Try this next...

Run for a single high-priority search
Report only projects below commitment
Track the trend by re-running weekly and comparing
Pair with the Market Map prompt to source additional candidates for under-represented dimensions

CLOCKWORK

# Prompt: Project Candidate Completeness Audit

**Description:** For a single project, audits every in-consideration candidate against a data-completeness checklist so the team knows which records need work before the next client update.
**Frequency:** Ad hoc (before client updates, slate presentations, or weekly project reviews)
**Connectors:** Clockwork Recruiting

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

Project: [project name or ID]. Build a Candidate Completeness Audit using Clockwork Recruiting. Fetch the project with ?include=client_company,project_type and display as <Company> \ <Project> throughout.

Step 1 — Confirm scope. Fetch statuses via list_reference_data(data_type=statuses). Define "in consideration" as any candidacy with status.rank >= 1100. Confirm the boundaries with me if ambiguous.
Step 2 — Pull in-scope candidacies via list_candidacies with include=person,status,overview_note,assessment_note,next_steps_note. Paginate fully.
Step 3 — For each candidate, pull get_person_details for positions, compensation, education, emails, phone_numbers, tags, and notes (include=author).
Step 4 — Score each candidate against this checklist. Mark each ✅ present, ⚠️ partial, or ❌ missing:

- Current position (title + company + start date)
- Work history (3+ prior positions, or full career if shorter)
- Education (at least one entry)
- Compensation (current comp — base, bonus/variable, equity if applicable)
- Contact info (at least one email AND one phone number)
- Résumé / CV attached — verify via get_person_details(detail_type=attachments). The response includes attachment_type, file_file_name, and a signed download_url (time-limited — use immediately, don't cache).
- Overview note on the candidacy (present, non-empty)
- Assessment note on the candidacy (present, non-empty)
- Next steps note (present, dated within last 14 days)
- Recent activity (any candidacy note in last 14 days)
- Tags (at least one on the person record)

Step 5 — Compute a completeness score per candidate (count of ✅ out of 11, as %).

Output:
1. Header — <Company> \ <Project>, total in-scope candidates, average completeness score, audit date.
2. Candidate Completeness Table — one row per candidate, a column per checklist item plus the score, sorted by score ascending (worst first).
3. Top Gaps — 3–5 bullets on the most common missing fields ("6 of 9 candidates are missing compensation data").
4. Action List — prioritized list of candidates and fields to fix, grouped by likely owner (most recent note author). Format: [Candidate] — [missing fields] — suggested owner: [user].
5. Client-Ready Check — a GO / NOT YET verdict on whether the slate is ready to share, with reasoning.

## Try this next...

Add screening-question ratings (get_candidacy(include_ratings=true)) as a completeness dimension
Loosen scope to the whole pipeline for a data-hygiene sweep
Tailor the checklist to your firm's must-haves
Run it across all active searches and rank projects by average completeness

CLOCKWORK

# AI PROMPTS FOR CLIENT & CANDIDATE COMMUNICATION

Prompts that draft or prepare external communication — to clients, candidates, and references including:

### + Client Update Email Draft

Generates a polished client-ready status email for a specific search, drawn directly from Clockwork activity.

### + Candidate Outreach Sequencer

Generates personalized first-touch outreach for a list of target candidates on a specific search.

### + Reference Call Prep Sheet

Pulls everything needed before placing a reference call on a finalist candidate.

### + Project Status Report [Template Builder]

Analyzes an existing status report document to extract its structure and create a reusable template pre-wired with Clockwork data placeholders. Run once (or whenever the format changes) before using next prompt.

### + Weekly Project Status Report - Generator and Distributor

Pulls live Clockwork data for each active retained search, populates the saved template, generates a formatted PDF per project, and emails it to the client contacts. Runs weekly with no manual data entry

### + Candidate Notes → Client Facing Overview

Reads every note on a person — across all their candidacies and their person record — and synthesizes them into a single, consistently formatted candidate overview written for a client audience. Applies the client-visibility filter so nothing internal-only leaks. The output is a paste-ready note (the connector can read notes but cannot write them back into Clockwork).

CLOCKWORK

# Prompt: Client Update Email Draft

**Description:** Generates a polished client-ready status email for a specific search, drawn directly from Clockwork activity.
**Frequency:** Ad hoc / Weekly per active search
**Connectors:** Clockwork Recruiting, Gmail / Outlook

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

When I give you a project name or ID, draft a client update email using Clockwork Recruiting data. Fetch the project with ?include=client_company,closing_reason and display as <Company> \ <Project> throughout.

**Client visibility filter (confirm once):** Before assembling content, confirm: "I'll include only candidates and notes flagged visible to the client (stoplightStatus = green). Keep red and none excluded, or override for this run?" Don't carry an override into later requests without re-confirming.

If isConfidential = true, don't use the client company name in the subject line — use the role title only and confirm recipients before proceeding.

Pull: the project record, all candidacies with include=person,status,assessment_note,overview_note,next_steps_note, and recent candidacy notes (last 14 days) with include=author.

**Apply filters:** candidacies to stoplightStatus = green only (note how many were excluded); filter notes independently (a green candidacy can still have red notes); filter status.rank < 100.

Then draft an email with:

- Warm one-line opener referencing the last client touchpoint (look up in Gmail if possible)
- **Progress this period**: candidacies updated in the window and candidates removed from consideration (status.category starts with out-). For movement, you can cite concrete stage changes from get_candidacy_status_history on green candidacies.
- **Pipeline snapshot**: counts by status.category group, with status.name labels
- **Spotlight**: 2–3 candidates the client should know about (highest status.rank among green candidacies), each with a 2-sentence positioning statement from candidacy notes
- **What's next**: concrete next steps and any decisions needed from the client

Tone: confident, concise, partner-level. Output as a Gmail draft to the client lead, subject [Client Company] \ [Role] — Search Update [Date].

## Try this next...

Generate updates for all active searches at once
Shift tone (formal board update vs. quick founder note)
Produce a short SMS/Slack version alongside the email
Include a finalist comparison table for late-stage searches
Pull the actual recipients with get_project_clients and address each contact by name

CLOCKWORK

# Prompt: Candidate Outreach Sequencer

**Description:** Generates personalized first-touch outreach for a list of target candidates on a specific search.
**Frequency:** Ad hoc (at search kickoff or pipeline refresh)
**Connectors:** Clockwork Recruiting, Gmail / Outlook (or Slack for internal handoff)

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

Project: [project name]. Fetch with ?include=client_company and display as <Company> \ <Project>.

Call list_reference_data(data_type=statuses) to confirm the categories for "not yet contacted" candidates (research-researching and early in-outreach). Pull the project brief and the candidacies in those categories via list_candidacies with include=person,status.

Filter out status.rank < 100.

Before drafting outreach for anyone: run run_conflict_check(person_id=...) and check doNotContact on the person record. Exclude anyone returning STOP or flagged DNC, and mark them "Excluded — [reason]". Surface CAUTION results for my review rather than auto-excluding.

For each remaining person, pull person details and prior notes, then draft a personalized outreach email that:

- Opens with a specific, non-generic hook from their background or a prior Clockwork interaction
- References the role in 1 sentence without naming the client unless the project is non-confidential
- Asks for a 20-minute exploratory call with two proposed time windows
- Avoids recruiter clichés ("rockstar", "ninja", "exciting opportunity")

Output: a table — Name | Current Company | Hook | Draft Email | Conflict/DNC Flag. Flag anyone I've personally interacted with before for review.

## Try this next...

Generate LinkedIn/InMail variants instead of email
Produce a 3-touch sequence (intro, follow-up, break-up) per candidate
Adjust seniority/tone; batch by company for a coordinated approach
Route the drafts to a researcher via Slack for sending

CLOCKWORK

# Prompt: Reference Call Prep Sheet

**Description:** Pulls everything needed before placing a reference call on a finalist candidate.
**Frequency:** Ad hoc (per finalist)
**Connectors:** Clockwork Recruiting

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

When I give you a candidate name and the project they're a finalist on, build a Reference Call Prep Sheet from Clockwork Recruiting. Fetch the project with ?include=client_company and display as <Company> \ <Project>.

Check doNotContact on the candidate's person record first; if true, flag it and don't suggest outreach paths.

Include: candidate snapshot (name, current title/company, status.name on this project), the role and client requirements (from the project brief), the candidate's relevant accomplishments from candidacy notes, any concerns or open questions raised in notes, and 8 tailored reference questions mapped to the role's success criteria. Where possible, suggest the right referee type per question (former manager, peer, direct report, client).

## Try this next...

Generate a referee-specific script (different questions for a former manager vs. a peer)
Add a comp/expectations summary from get_person_details(detail_type=compensation) for the offer conversation
Produce a post-call template to capture answers as a structured note
Build prep sheets for all finalists on a search at once

CLOCKWORK

# Prompt: Project Status Report — Template Builder (run once per report format)

**Description:** Analyzes an existing status report document to extract its structure and create a reusable template pre-wired with Clockwork data placeholders. Run once (or whenever the format changes) before using the next prompt
**Frequency:** One-time setup (or when the report format changes)
**Connectors:** Clockwork Recruiting

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

I'll give you an existing project status report (PDF or Word). Analyze its structure and produce a reusable template that can be populated with live Clockwork Recruiting data each week.

Step 1 — Read the report. Identify every distinct section, heading, table, and recurring data field. Focus on structure and field types, not the content.
Step 2 — Map each field to a source:

- Clockwork — Project: get_project with ?include=client_company,lead,owner,project_type,closing_reason,placement (use <Company> \ <Project> for display). Financials available via get_project_fee, get_project_compensation, get_project_payments; detailed location via get_project_location.
- Clockwork — Candidacies: list_candidacies with include=person,status,overview_note,assessment_note,next_steps_note. status.name for display, status.category for groupings. Filter status.rank < 100. Stage-movement fields available via get_candidacy_status_history.
- Clockwork — Client Visibility: stoplightStatus on each candidacy and note. Mark as a required filter — any client-facing section draws only from green records.
- Clockwork — Notes: get_candidacy_notes with include=author (categoryName is inlined).
- Static / Firm-provided: fields filled manually (logo, partner name, disclaimers).
- Computed: derived fields (total candidates, days open, pipeline counts by category).

Step 3 — Build the template as Markdown. Replace every data field with a {{PLACEHOLDER_NAME}} and annotate each with its source. Examples:

- {{PROJECT_DISPLAY_NAME}} — <Company> \ <Project> from project + client_company
- {{CANDIDATE_TABLE}} — Clockwork candidacies, green filter applied; status.name display, status.category grouping
- {{PARTNER_SIGNATURE}} — static, fill manually

Step 4 — At the top, include a Data Dictionary listing every placeholder, its source, the tool/field that populates it, and required vs. optional.

Step 5 — Save to /Status Reports/templates/project-status-template.md and confirm the path.

Don't guess at content not visible in the document. If a section's purpose is ambiguous, ask before templating it.

## Try this next...

Build separate templates per client or report type (weekly update vs. final placement summary)
Add a financials/fee block for searches where the client tracks spend
Produce an HTML/branded version alongside the Markdown

CLOCKWORK

# Prompt: Weekly Project Status Report — Generator & Distributor (uses template from previous prompt)

**Description:** Pulls live Clockwork data for each active retained search, populates the saved template, generates a formatted PDF per project, and emails it to the client contacts. Runs weekly with no manual data entry.
**Frequency:** Weekly (suggested Friday afternoon or Monday morning)
**Connectors:** Clockwork Recruiting, Gmail / Outlook

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

Generate and distribute the weekly project status reports for all my active retained searches.
Prerequisites: The template must exist at /Status Reports/templates/project-status-template.md. If not, stop and ask me to run that prompt first.

Session setup (do once, cache): Load business-rules and domain knowledge, resolve my identity via clockwork_get_current_user, and cache: firm users (Partner/Recruiter/Researcher/Admin) → userId → name; statuses → statusId → {name, category, rank}; note categories → categoryId → name.

Step 1 — Load the template.
Step 2 — Pull my active retained searches in one call: list_projects(user_id=<me>, status=active, project_type="Retained", include=lead,owner,client_company,closing_reason). Filter out isInternal = true.
Client visibility filter (confirm once before processing): "I'll include only stoplightStatus = green candidates and notes in each report. Keep red and none excluded, or override for specific projects?" Don't carry an override across projects without re-confirming.

For each project:

- get_project with ?include=lead,owner,client_company,project_type,closing_reason,placement → name (as <Company> \ <Project>), client, description, dates. If isConfidential = true, omit the client name and alert me before distribution.
- list_candidacies with include=person,status,overview_note,assessment_note,next_steps_note. Filter: green only; rank >= 100 only.
- get_candidacy_notes over the last 7 days with include=author. Filter: green notes independently.
- For movement language, use get_candidacy_status_history to cite concrete stage changes this week.

Step 3 — Populate a copy of the template per project, replacing every {{PLACEHOLDER}}. Computed fields: totals by status.category and status.name; days since startedAt; most recent note date this week; 2–3 spotlight candidates (highest status.rank among green candidacies with non-empty assessment notes); count of excluded candidacies (red + none) for the footer. Leave "static / fill manually" placeholders as-is or use a firm default.
Step 4 — Generate a PDF per project (reportlab, clean formatting). Save to /Status Reports/YYYY-MM-DD/[Client] - [Role] - Status Report.pdf.
Step 5 — Identify recipients via get_project_clients (the authoritative per-search client contact list with names and emails). Don't search people by company name. If it returns no contacts, flag the project and ask me to confirm recipients.
Step 6 — Draft one Gmail per project to the confirmed recipients: subject [Client Company] \ [Role] — Search Update — [Date]; body a 2–3 sentence summary (green content only); attach the PDF.
Present each draft for review — don't send automatically. If I say "send all", send all approved drafts.
Step 7 — After sending, log a note on each project's most recent candidacy recording the date the update was sent and to whom. Use note category "Client Communication" if it exists.
Output a summary table: <Company> \ <Project> | Recipients | PDF path | Candidates shown (green) | Candidates excluded (red/none) | Email status (Draft / Sent / Flagged).

## Try this next...

Run for a single client or a subset of searches
Switch the deliverable to a branded HTML email instead of a PDF attachment
Change cadence to biweekly
Add a fee/payment status line for clients who track spend
Stop at the draft stage and hand off sending to an assistant

CLOCKWORK

# Prompt: Candidate Notes → Client Facing Overview

**Description:** Reads every note on a person — across all their candidacies and their person record — and synthesizes them into a single, consistently formatted candidate overview written for a client audience. Applies the client-visibility filter so nothing internal-only leaks. The output is a paste-ready note (the connector can read notes but cannot write them back into Clockwork).
**Frequency:** Ad hoc (before sharing a candidate with a client, or building a slate)
**Connectors:** Clockwork Recruiting

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

When I give you a person's name (and optionally the specific search this overview is for), build a client-facing Candidate Overview from their Clockwork notes. If the name is ambiguous, list matches and ask me to confirm.

Session setup: Load business-rules and domain knowledge if not already done this session, and cache statuses (list_reference_data(data_type=statuses)) and the firm-user map (for note attribution).
Step 1 — Gather every note. Pull the person record (get_person) and person notes (get_person_details(detail_type=notes, include=author)). Get all their candidacies via list_candidacies(person_id=...), then pull candidacy notes for each via get_candidacy_notes(include=author). Also pull positions, education, and compensation from get_person_details for factual grounding.
Step 2 — Apply the client-visibility filter (this is a client-facing document). Confirm once: "I'll include only notes and candidacies flagged visible to the client (stoplightStatus = green). Keep red and none excluded, or override for this run?" Then:

- Exclude any candidacy with stoplightStatus ≠ green.
- Exclude any individual note with stoplightStatus ≠ green, even on a green candidacy (filter notes independently).
- Exclude references/benchmarks (status.rank < 100).
- Check doNotContact — if true, flag it at the top and do not include outreach suggestions.
- If the source search is confidential (isConfidential = true), do not name the client company anywhere in the overview.

Step 3 — Synthesize, don't transcribe. Read across the filtered notes and resolve them into themes. Where notes conflict or a comp expectation changed over time, use the most recent and say so. Never invent detail that isn't supported by a note or a structured field; if a section has no shareable source material, write "Not yet captured" rather than guessing.
Step 4 — Output the overview using exactly this structure (this is the standard format — keep the section order and headers consistent every time so overviews are comparable across candidates):

[Candidate Name] [Current Title] · [Current Company] · [Location]

Summary — 2–3 sentence positioning statement: who they are and why they're a credible fit for this type of role.
Relevant Experience — 3–5 bullets of the most relevant accomplishments and scope, drawn from notes and work history.
Strengths — 3–4 bullets on what stands out, grounded in specific note observations.
Motivations & Fit — what's drawing them to a move / this opportunity, and any constraints (location, timing, role scope) that are shareable.
Compensation — current comp and expectations if captured in a green note or field; otherwise "Not yet captured."
Areas to Explore — 2–3 open questions or things still to validate (framed neutrally, client-appropriate — no internal-only concerns).
Recommended Next Step — one clear suggested action for the client.
Close with a one-line footer: how many notes were synthesized and how many were excluded by the visibility filter, so I know what's not shown. Output as clean Markdown I can copy into Clockwork or an email.

## Try this next...

Swap in your firm's preferred section set or house style
Produce an internal version (drop the green filter, add a candid "Recruiter Assessment / Risks" section) for team use
Generate overviews for an entire slate in one pass so they read consistently side by side
Tune length (a tight 150-word version for a slate cover page vs. a full one-pager)
Feed an existing well-formatted overview as a model and have it match that structure instead of the standard one above

CLOCKWORK

# AI PROMPTS FOR SEARCH EXECUTION & RESEARCH

Prompts for launching a search, looking people up, mapping a market, and clearing conflicts before approach.

## + Search Kickoff Brief

Assembles everything needed to launch a new search: similar past searches, reusable target companies, candidates from related pipelines, and a starter target list.

## + Market Map Generator

Builds a market map of a target function at a defined set of companies, drawing on Clockwork people data and supplementing with web research.

## + Person Brief 360

A complete background and relationship dossier on a named individual, combining their profile, every interaction in Clockwork, and their full candidacy history.

## + Phone Number Lookup

Looks up a person in Clockwork by phone number. About as basic as it gets.

## + Off-Limits & Conflict Checks

Pre-flight check before approaching a candidate to verify no off-limits, client conflict, or recent placement constraints exist.

CLOCKWORK

# Prompt: Search Kickoff Brief

**Description:** Assembles everything needed to launch a new search: similar past searches, reusable target companies, candidates from related pipelines, and a starter target list.
**Frequency:** Ad hoc (at every new project launch)
**Connectors:** Clockwork Recruiting, Shared Drive

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

When I name a new project, build a Search Kickoff Brief from Clockwork Recruiting. Fetch the project with ?include=client_company,project_type and display as <Company> \ <Project> throughout.

1. Pull the project record and any client-provided documents (check Google Drive for the client folder). Pull target comp via get_project_compensation to frame the market.
2. Find the 3 most similar past searches (same function, similar seniority/industry). For each, display as <Company> \ <Project> with final placements (use get_project with include=placement, or status.category == "in-placed" on the candidacies), runner-up candidates, and target companies used.
3. From those past searches, surface 15 candidates worth re-engaging — with a 1-line "why now" per person from their candidacy history (list_candidacies(person_id=...) gives each person's full history in one call). Before including anyone, run run_conflict_check(person_id=...). Exclude STOP results and note them separately; surface CAUTION for review. Exclude anyone currently in an active candidacy on another project.
4. Draft a starter target company list (20 companies) from companies tagged or referenced across the similar searches. Check doNotContact on each company (get_company). If DNC with a future doNotContactEndDate, include with the restriction noted; if no end date, exclude.
5. List the top 5 internal team members with the most relevant relationships in this space, ranked by notes authored on related searches.

Output as a markdown brief, ready to drop into the kickoff meeting.

## Try this next...

Generate a sourcing plan (companies → titles → named targets) instead of a brief
Widen or narrow "similar searches" by function/geo/seniority
Add a comp benchmark from placed candidates' record
Feed the target list straight into the Market Map prompt

CLOCKWORK

# Prompt: Market Map Generator

**Description:** Builds a market map of a target function at a defined set of companies, drawing on Clockwork people data and supplementing with web research.
**Frequency:** Ad hoc (at search kickoff or for BD)
**Connectors:** Clockwork Recruiting, Browser Search

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

Build a market map for the role of [role title] across [list of companies or industry segment].

Step 1 — For each company, query Clockwork for people whose current title matches or is adjacent to the target role using search_people (company name or title as query). If you have a LinkedIn URL for a target, use list_people(linkedin_url=...) to check for an exact match before marking them "External — not in Clockwork." Capture name, title, tenure if known, and last interaction.

Step 2 — For each matched person, check doNotContact. If true, flag "DNC" rather than listing as a target.

Step 3 — For companies with thin Clockwork coverage, use WebSearch to identify likely current role holders from public sources, clearly marked "External — not in Clockwork".

Step 4 — Filter out candidacy entries with status.rank < 100 from any Clockwork counts.

Step 5 — Output a table grouped by company: Name | Title | Source (Clockwork / External) | Last Touch | DNC Flag | Notes.

Step 6 — End with a coverage assessment: which companies are well covered in Clockwork and which need sourcing investment.

## Try this next...

Add a "movement" column flagging people who recently changed roles
Cross-reference against an active search to mark who's already in pipeline
Produce a tiered target list (A/B/C) rather than a flat map
Focus on diversity dimensions your firm captures via tags

# Prompt: Person Brief 360

**Description:** A complete background and relationship dossier on a named individual, combining their profile, every interaction in Clockwork, and their full candidacy history.
**Frequency:** Ad hoc (before any meaningful interaction)
**Connectors:** Clockwork Recruiting, Gmail / Outlook

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

When I give you a person's name, build a 360° brief using Clockwork Recruiting. If the name is ambiguous, list matches and ask me to confirm.

Pull: the person record (get_person), person details (get_person_details for notes, positions, emails, tags, education, compensation, attachments), every candidacy they've ever had via list_candidacies(person_id=...) (full cross-project history in one call), and the candidacy notes on those candidacies. Clockwork stores email addresses, not email messages — correspondence comes from Gmail below.

Check doNotContact immediately after the person record. If true, flag prominently at the top and don't suggest outreach.

Résumé / CV: Call get_person_details(detail_type=attachments). If found, present the download_url with the caveat "This link is time-limited — open it now." Don't cache the URL.

Then, using Gmail, look up the last 20 email threads between me (or anyone on my domain) and any of this person's email addresses. Capture subject, date, direction, and a 1-line summary. If more than 20, note the total and date range.

Produce the brief:

1. Snapshot — current title, company, location, best contact info. Flag doNotContact if set.
2. Career Trajectory — concise work-history narrative; call out inflection points.
3. Clockwork History — date created and by whom; total notes split into "My interactions" vs "Other team members" (via note author) with counts and date ranges; recent Gmail correspondence (last 20 threads, most recent 3 individually); last touch.
4. Candidacy History — table of every search: <Company> \ <Project>, outcome (status.category + status.name), year, owning partner.
5. Themes from Notes — recurring observations (motivations, comp expectations, geographic constraints, red flags).
6. Key Takeaways for Next Engagement — 3–5 bullets, each grounded in a specific Clockwork artifact.

If a section has no data, skip it rather than inventing content.

## Try this next...

Trim to a one-page pre-call snapshot
Add a conflict check (run_conflict_check) for off-limits screening
Include screening-question ratings from any active candidacy
Build briefs for a whole slate at once before a client presentation

CLOCKWORK

# Prompt: Phone Number Lookup

**Description:** Looks up a person in Clockwork by phone number. About as basic as it gets.
**Frequency:** Ad hoc (e.g., when an unknown number calls)
**Connectors:** Clockwork Recruiting

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

When I give you a phone number, use search_people with the number as the query (try it with and without country code, dashes, spaces, and parentheses if the first attempt returns nothing).

If exactly one match: return name, current title, current company, preferred email, preferred phone, and the person's Clockwork ID.

If multiple matches: list all with name, title, company, and ID, and ask which I meant.

If no match: say so plainly and suggest a partial number.

## Try this next...

Chain it into the 360° brief once a match is confirmed
Do the same with an email address or LinkedIn URL (list_people(linkedin_url=...))
Have it auto-pull recent candidacy/note activity for a quick "who is this and what's our history" before you call back

CLOCKWORK

# Prompt: Off-limits & Conflict Checks

**Description:** Pre-flight check before approaching a candidate to verify no off-limits, client conflict, or recent placement constraints exist.
**Frequency:** Ad hoc (before any candidate approach)
**Connectors:** Clockwork Recruiting

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

When I give you a candidate name, run a conflict check in Clockwork Recruiting.

Step 1 — Resolve the person with search_people. Step 2 — Run run_conflict_check(person_id=...), which returns a GO / CAUTION / STOP verdict and the supporting evidence in one call: do-not-contact flag, prior placement inside the no-poach window (default 24 months — adjust with no_poach_months if our policy differs), and active involvement on another open search.

Then enrich the verdict:

- If the candidate's current employer is the client on any of my active searches, flag it as a client conflict (find the employer via get_person_details(detail_type=positions), is_current = true; check it against list_projects(company_name=...)).
- Pull person tags and candidacy notes for any confidentiality or sensitivity markers.
- For the employer, check company doNotContact and doNotContactEndDate.

Return the GO / CAUTION / STOP verdict at the top, then the supporting evidence.

> **Try this next...**
> Batch-check an entire target list before a sourcing push
> Tighten or loosen the no-poach window per client agreement
> Run it as the first step of any outreach prompt
> Produce a clearance log you can attach to the search file

CLOCKWORK

# AI PROMPTS FOR BUSINESS DEVELOPMENT

Prompts focused on generating new business from existing relationships in Clockwork.

## + Generate / Source Leads

Identifies companies and contacts in Clockwork that look like strong BD targets based on past relationships and search history.

CLOCKWORK

# Prompt: Generate Leads / Companies To Target

**Description:** Identifies companies and contacts in Clockwork that look like strong BD targets based on past relationships and search history.
**Frequency:** Monthly
**Connectors:** Clockwork Recruiting

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

Using Clockwork Recruiting, build a BD target list for the next 30 days.

Method note: Placement detection is canonical — identify placed candidates via status.category == "in-placed" on a closed project where closingReason.name == "Placement" (or read include=placement on the project). This walks closed projects, so paginate fully.

Logic:

1. Walk closed projects from the last 36 months. Find ones with a placement. For each placed candidate, find their current employer (get_person_details(detail_type=positions), is_current = true). Build a list of companies where we've placed someone, excluding companies with a current active project (check via list_projects(company_name=...)).
2. List companies with 5+ people records in Clockwork that have never been a project client.
3. For each company on both lists: check company.doNotContact (get_company) before including. If DNC with no future end date, exclude; if doNotContactEndDate is in the future, include with "DNC until [date]" and no immediate-outreach suggestion.
4. For companies that pass, pull the strongest contact (most senior, or most recently noted) and summarize the relationship in 2 sentences from person notes.

Output a ranked table: Company | Reason on List | Best Contact | Last Touch | Suggested Angle | DNC Status. Cap at 25 companies.

## Try this next...

Focus on one sector or geography
Build it around alumni of a single portfolio/company
Shorten the lookback for a "recent placements → warm intros" push
Generate a tailored outreach email per top-5 target alongside the list

CLOCKWORK

# AI PROMPTS FOR TEAM OPERATIONS, PERFORMANCE & CAPACITY

Prompts focused on the operational health of the team itself: who is doing what in Clockwork, who is performing well against their book, and how capacity is distributed.

### + Team Activity Leaderboard

Internal-facing weekly view of recruiter activity volume across the team for coaching and load-balancing.

### + Monthly Clockwork Activity Report By User

Per-user activity rollup showing how each team member uses Clockwork over the month — notes logged, candidacies touched, projects touched, and unique people interacted with.

### + Team Data Hygiene Scorecard

Operational quality check on how well the team keeps Clockwork records complete and current — note frequency, stale candidacies, and missing fields.

### + Partner Performance Scorecard by Project Outcome

Compares team members on the health and outcomes of the projects they own, using project status and candidacy progression as proxies for performance. Frequency: Monthly or Quarterly

### + Team Capacity & Workload Comparison

Shows how project and candidacy load is distributed across the team over a window, highlighting who is overloaded, who has headroom, and how it has trended.

### + Monthly Practice Review

Comprehensive monthly review of practice performance: searches launched, closed, time-to-fill, win/loss, and pipeline outlook.

### + Project Efficiency Report

Comprehensive review of a project that measures how efficiently the team ran it and to surface where time, effort, and sourcing capacity were won or lost.

CLOCKWORK

# Prompt: Team Activity Leaderboard

**Description:** Internal-facing weekly view of recruiter activity volume across the team for coaching and load-balancing.
**Frequency:** Weekly
**Connectors:** Clockwork Recruiting, Slack

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

Using Clockwork Recruiting, generate a Team Activity Leaderboard for the past 7 days.

Session setup: Load business-rules and domain knowledge. Fetch the team roster via list_reference_data(data_type=users), keep only Partner/Recruiter/Researcher/Admin, and build a userId → {name, email, role} map. This is your attribution reference; don't re-fetch per user.

Method note: The reliable activity metric is notes logged. For each team member, count notes authored in the last 7 days across active projects' candidacies and person records (get_candidacy_notes with a date window and include=author). Fetch projects with ?include=client_company so references display as <Company> \ <Project>. Show totals, week-over-week change, and a per-project breakdown for anyone whose note volume dropped 30%+ vs last week.

Output as a markdown table, then post a friendly summary (top 3 + shoutouts, no callouts) to the #recruiting-team Slack channel as a draft for my review.

## Try this next...

Switch to monthly or quarterly
Scope to one pod or one partner's reports
Add a stage-movement count per person (now available via get_candidacy_status_history) as a second activity signal
Drop the Slack step and keep it internal

CLOCKWORK

# Prompt: Monthly Clockwork Activity Report by User

**Description:** Per-user activity rollup showing how each team member uses Clockwork over the month — notes logged, candidacies touched, projects touched, and unique people interacted with.
**Frequency:** Monthly (first business day)
**Connectors:** Clockwork Recruiting

**Prompt**: (copy/paste the prompt below in grey into your chat in your AI Assistant)

Method note: The reliable per-user activity metric is notes authored — use it as the spine and label other counts as approximations.

Step 1 — Session setup: Load business-rules and domain knowledge. Fetch the roster via list_reference_data(data_type=users), keep Partner/Recruiter/Researcher/Admin, build a userId → {name, email, role} map. Fetch note categories → categoryId → name map (note: categoryName is inlined on each note, so you can read it directly).

Step 2 — Walk every project active during the month. Fetch with ?include=client_company. For each, pull candidacy notes and person notes in the window via get_candidacy_notes with begin_time/end_time and include=author.

Step 3 — For each user, compute: Notes logged (total); Notes by category (group by inlined categoryName); Distinct candidacies touched; Distinct projects touched (display as <Company> \ <Project>); Distinct people touched; First and last activity date in the month.

Step 4 — Compare to the prior month and show deltas.

Output: a markdown table sorted by total notes descending, plus a 5-bullet narrative calling out the top performer, the most-improved, anyone down 30%+, and any user with zero activity. Label clearly: "Activity metrics are based on notes authored — the reliable per-user activity metric the connector provides."

## Try this next...

Run quarterly for review cycles
Break out by role (researchers vs. partners have different note patterns)
Add stage-movement activity from status history
Focus on a single team member for a 1:1 prep

# Prompt: Team Data Hygiene Scorecard

**Description: O**perational quality check on how well the team keeps Clockwork records complete and current — note frequency, stale candidacies, and missing fields.
**Frequency:** Monthly
**Connectors:** Clockwork Recruiting

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

Build a Team Data Hygiene Scorecard for Clockwork Recruiting.

For each active firm staff member — fetch users via list_reference_data(data_type=users), keep Partner/Recruiter/Researcher/Admin only — across the projects where they're a team member (use get_user_projects(user_id=...) per person, with ?include=client_company), score:

- Note recency — % of their active candidacies with a note in the last 14 days
- Stage freshness — % of their active candidacies with updated_at in the last 14 days
- Stalled candidacies owned — count with no note and no update in 21+ days (filter status.rank < 100)
- Project notes cadence — average days between notes per project they own

Combine into a 0–100 hygiene score (note recency 40, stage freshness 30, stalled inverse 20, cadence 10).

Output a leaderboard with project references as <Company> \ <Project>, then a "Coaching Opportunities" section listing the bottom 3 with the specific projects and candidacies dragging their score down.

## Try this next...

Add a résumé-attached or comp-captured completeness dimension
Scope to one team or one partner's book
Raise the bar (7-day windows) for high-touch searches
Pair with previous prompts to drill from team scores into per-candidate fixes

CLOCKWORK

# Prompt: Partner Performance Scorecard by Project Outcome

**Description:** Compares team members on the health and outcomes of the projects they own, using project status and candidacy progression as proxies for performance.
**Frequency:** Monthly or Quarterly
**Connectors:** Clockwork Recruiting

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

For the trailing 90 days, build a Partner Performance Scorecard. Fetch projects with ?include=lead,owner,client_company,closing_reason,placement and display as <Company> \ <Project> throughout.

Method note: Placement detection is canonical. A filled project is closed, has closingReason.name == "Placement", and has a candidacy with status.category == "in-placed" (include=placement returns the placed person directly).

Step 1 — Pull projects from the trailing 90 days and group by owning team member (owner.id, falling back to lead.id). get_user_projects(user_id=..., status=active,closed) gives each person's book directly.

Step 2 — For each owner, compute: Active projects; Projects closed in window; Filled and fill rate (filled / closed); Approximate average time-to-fill (startedAt → closedAt for filled); Average pipeline depth (candidacies with rank >= 100 per active project); Average days-to-first-client-stage (now derivable precisely from get_candidacy_status_history — the first transition into in-contenders or higher); Stalled project count (active projects with no note in 14+ days).

Output: a ranked table with all metrics, then a 3-paragraph narrative on the standout performer, the partner whose pipeline depth most exceeds throughput (potential bottleneck), and the partner whose throughput most exceeds pipeline depth (running thin).

## Try this next...

Run quarterly or annually for review
Add average time-in-stage across the funnel to find where each partner's candidates stall
Weight by search difficulty or fee size
Focus on a single partner for a deep-dive

CLOCKWORK

# Prompt: Team Capacity & Workload Comparison

**Description:** Shows how project and candidacy load is distributed across the team over a window, highlighting who is overloaded, who has headroom, and how it has trended.
**Frequency:** Weekly or on demand for resourcing decisions
**Connectors:** Clockwork Recruiting

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

Build a Team Capacity & Workload Comparison for the period [default: last 30 days; ask if I want a different window].

Step 1 — Fetch the roster via list_reference_data(data_type=users), keep Partner/Recruiter/Researcher/Admin, cache the map.

Step 2 — For each member, pull their book directly with get_user_projects(user_id=..., status=active, include=lead,owner,client_company). For each, compute:

- Active projects assigned (display as <Company> \ <Project>)
- Total active candidacies (filter rank >= 100; if a project has multiple team members, divide by team size for fair allocation)
- Active candidacies in client-facing stages (status.rank >= 1300; confirm the threshold if the firm's progression differs)
- Notes authored in the window (engagement proxy)
- Distinct people personally interacted with in the window
- New projects added to their plate in the window

Step 3 — Compute a normalized capacity index per user: (weighted_candidacies × 1.0) + (client_facing_candidacies × 1.5) + (new_projects × 5) and express each user's load as a % of the team average and team max.

Step 4 — Trend: compare to the prior equivalent window and show direction (↑ ↓ →) per user.

Output: (1) a markdown table sorted by capacity index descending, with raw counts, index, % of average, and trend; (2) a horizontal ■-bar visualization to eyeball the spread; (3) a Resourcing Recommendations section — who is over capacity (>130% of average), who has headroom (<70%), and which specific active projects (by <Company> \ <Project>) could be reassigned or supported.

## Try this next...

Change the window or weighting to match how your firm thinks about load
Scope to one pod for a resourcing huddle
Add fee-weighting so revenue-heavy searches count more
Run it before assigning a new search to see who has room

CLOCKWORK

# Prompt: Monthly Practice Review

**Description:** Comprehensive monthly review of practice performance: searches launched, closed, time-to-fill, win/loss, and pipeline outlook.
**Frequency:** Monthly (first business day)
**Connectors:** Clockwork Recruiting

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

Using Clockwork Recruiting, build my Monthly Practice Review for the month just ended. Fetch projects with ?include=lead,owner,client_company,closing_reason,project_type,placement and display as <Company> \ <Project> throughout. Filter out isInternal = true.

Method note: Placement detection is canonical. A placement is confirmed by project.status == "closed", closingReason.name == "Placement", and a candidacy with status.category == "in-placed" (include=placement returns the placed person). Time-to-fill uses startedAt → closedAt.

Sections:

1. Searches launched — count, by client (<Company> \ <Project>), by function (projectType.name)
2. Searches closed — count, time-to-fill, and placed candidate name + company (from include=placement)
3. Searches lost or paused — projects closed/paused with no placement; reason from notes where captured
4. Active pipeline — active searches, total candidates (rank >= 100), distribution by status.category
5. Client concentration — % of activity by top 5 clients
6. Trend vs prior month — deltas on launches, closes, pipeline depth
7. Three things to fix next month — your honest read on what the data shows

Output as a markdown report ready to share with my managing partner.

## Try this next...

Run quarterly or annually with the same structure
Add a revenue view (get_project_fee across filled searches) for a billings summary
Add average time-in-stage across the funnel to show where deals slow
Scope to a single practice area or partner

CLOCKWORK

# Prompt: Project Efficiency Report

**Description:** Comprehensive review of a project that measures how efficiently the team ran it and to surface where time, effort, and sourcing capacity were won or lost.
**Frequency:** Upon search completion
**Connectors:** Clockwork Recruiting

**Prompt:** (copy/paste the prompt below in grey into your chat in your AI Assistant)

You are an **executive-recruiting process-efficiency consultant**. Your job is not to summarize the search — it is to measure how efficiently the team ran it and to surface where time, effort, and sourcing capacity were won or lost. Be specific, quantitative, and neutral. Every number must trace to Clockwork data; never estimate a metric you did not compute.

**DATA TO PULL (Clockwork)**

Step 1: get_knowledge → business-rules and domain (once). Respect: exclude isInternal projects; honor doNotContact; use status.category/rank for bucketing and status.name for display; never infer stage-transition history that the API doesn't store.
Step 2: Project record with include=client_company,closing_reason,placement → dates, close reason, placed person.
Step 3: list_reference_data statuses → build statusId → {name, rank, category} map (needed for current status).
Step 4: **All** candidacies for the project with include=candidacy_stat — paginate fully (pages truncate ~20 records; loop until has_more=false). Per candidacy capture: current statusId, stoplightStatus, colorCategory, createdAt; and from candidacyStat: pickupUser/pickupUserId, pickupAt, firstGreenAt, highWatermarkStatus/Rank/Category, highWatermarkAt.
Step 5: For the **placed candidate(s)** (current or high-water category == in-placed): pull get_candidacy_notes (sorted ascending, include=author) and get_candidacy_status_history. Use notes to reconstruct outreach → firm-screen → client-presentation → offer/placement dates when the status-history endpoint is empty (common on migrated data).

**METRICS TO COMPUTE**

Step 1: **Days to Close** = closedAt − startedAt (calendar days). Also report **Days to Placement Decision** = highWatermarkAt(placed) − startedAt, and flag the **administrative tail** (close date minus placement date) if material.
Step 2: **Total candidates** = count of candidacies, **excluding** anyone whose current status category is other-reference (Benchmark / Reference). State both the raw count and the excluded count.
Step 3: **Candidate waterfall** — cumulative furthest stage reached, using **high-water-mark rank** (so a candidate who later dropped still counts at their deepest stage):
- Sourced/Added (all) · Reached Outreach (hw ≥ 1100) · Reached Client/Contender (hw ≥ 1300) · Reached Offer (hw ≥ 1400) · Placed (hw ≥ 1500). Show count and % of sourced at each step.

Step 4: **Sourcing efficiency by user** (efficiency lens #1 — quality of sourcing). Group by pickupUser. For each: total sourced; # advanced to outreach (hw ≥ 1100) = "worked after review"; # left at research (hw ≤ 1010, research-*) = "added but never worked"; **# added → immediately removed** = high-water-mark stayed at Research (research-*) **AND** current status rank is between 110 and 400 (i.e. sourced, never worked, then disqualified); # reaching client (hw ≥ 1300); # made client-visible (firstGreenAt set). Report volume **and** precision (client-rate), and call out the volume-vs-precision contrast and any single-sourcer concentration in the "added → removed" column.
Step 5: **Interview-readiness / qualification efficiency** (efficiency lens #2 — quality of the screening/ qualification decision). Take the subset of candidates whose high-water-mark rank reached **≥ 1238** ("We will Interview" or beyond) — i.e. someone judged them qualified and worth interviewing. Of that subset: **# moved forward** = client-interviewed (hw ≥ 1300) OR cleared for client visibility (firstGreenAt set / stoplight green); **# fell off** = neither interviewed by the client nor green-lit. Report the move-forward rate (a proxy for how sound the qualification calls were), and a per-user cut attributed to the user who advanced the candidate (highWatermarkUserId) — label it directional, since the API does not retain who set the 1238 transition specifically.
Step 6: **Presentation efficiency** — of candidates that cleared outreach (hw ≥ 1100), what % were presented to the client (hw ≥ 1300); plus client-interview → placement conversion.
Step 7: **Placed-candidate time series** (days from project open): time to source, source → first outreach, outreach → client presentation, presentation → placement decision, and → start date. Name the sourcer.

**ALSO CONSIDER (add if the data supports it)**

- Top-of-funnel concentration: bulk day-one sourcing vs. targeted later adds; did the hire come from bulk or from a targeted add?
- Slate quality: client-visible (green) rate vs. total sourced.
- Data-hygiene flags: large administrative close tail; bulk status-tagging that inflates "outreach"; candidacies never advanced past Source/Research.

**OUTPUT**

- Produce a **single self-contained, nicely designed HTML file** (inline CSS, no external libraries): a header band with project + close reason, KPI cards, a horizontal waterfall/funnel, a sourcing-by-user table with inline bars, an interview-efficiency panel, a placed-candidate timeline, and a short "Consultant Observations" section (3–6 crisp findings + recommendations).
- Palette: professional, calm (deep indigo/navy header, teal/green for progression, amber for caution, red for drop-off). Accessible contrast. Print-friendly.
- Footer: data provenance ("Source: Clockwork — N of N candidacies analyzed") and generation date.
- If AUDIENCE = client-facing, include only stoplightStatus = green candidates in any per-person detail and suppress internal sourcer names.

**GUARDRAILS**

- Reconcile every total (buckets must sum to the population). State "Showing N of N."
- Distinguish high-water-mark stage (furthest reached) from current status (final disposition); the report needs both.
- Do not present bulk status-tags as verified individual outreach without noting the caveat.

## Try this next...

Run quarterly or annually with the same structure to review all projects for efficiency

CLOCKWORK

# Frequently Asked Questions

**Is my Clockwork login secure?**
The connection uses Clockwork's standard secure sign-in. You log in exactly as you normally would; your password is never shared with the AI Assistant. You can disconnect at any time.

**Is my data sent to Anthropic or OpenAI?**
Clockwork ESE Inc does not send any data to an AI Assistant. Our MCP Server is used by a user's AI Assistant to retrieve data from their Clockwork environment through the API.

**How much does the MCP Server Cost?**
Access to Clockwork's MCP Server is included at no additional cost. You'll use it with the AI Assistant of your choice, such as ChatGPT, Claude, or Microsoft Copilot. Any subscription or usage fees associated with your chosen AI Assistant are separate and billed directly by that provider. Clockwork does not manage, monitor, or bill for your AI Assistant subscription.

**What does hallucination mean?**
A hallucination occurs when an AI Assistant generates information that sounds believable but is actually incorrect, incomplete, or entirely fabricated. While connecting your AI Assistant to Clockwork through the MCP Server helps ground responses in your live data, it's still good practice to review AI-generated content before relying on it for important decisions.

**Why aren’t my prompts producing the results I expected?**
There are several reasons this can happen:

1. Your request may be too broad or not specific enough
2. The AI Assistant may need additional context about what you're trying to accomplish
3. The information you're asking for may not exist in your Clockwork data
4. The request may need to be phrased differently to match the available data or supported tools

The good news is that prompting is conversational. If you don't get the result you expected, try adding more context, specifying exactly what you're looking for, or breaking a complex request into smaller steps. Small changes in wording can often produce significantly better results.

For further reading and help troubleshooting using Clockwork with your AI Assistant please visit the support articles below.

Go to MCP Server / Clockwork Support Articles →