

Capture MCP tools

Every tool an AI app can call on a Capture account, generated from the server's own registry: what each one does, what it takes, and what it returns.

14 tools — 4 read, 10 write — and 4 prompts.

Server: `https://capture-app.graviti.inc/mcp` (Streamable HTTP, stateless).

Protocol versions: 2025-11-25, 2025-06-18, 2025-03-26.

Generated 25 September 2026 from commit 607013e by `scripts/mcp_reference.py`.

Connecting

Every connection is to an account made in the app. **Settings** → **MCP** on the phone makes codes and tokens, shows what is connected, and disconnects. A connection is **read only** or **read & write**, chosen with "Allow changes" when it is made. A read-only connection does not see the write tools at all.

Apps that sign in (ChatGPT, Claude and other OAuth clients): add `https://capture-app.graviti.inc/mcp` as a custom connector. The app registers itself and opens a Capture page; sign in there with Apple or Google, or type a code from Settings → MCP → Get a code.

Tools that use a token (Claude Code, Codex, Gemini CLI, Cursor and others): Settings → MCP → Create a token, then send `Authorization: Bearer cap_mcp_...` to the server URL. Ready-made snippets are in the app and at `https://capture-app.graviti.inc/connect`.

Everything a tool writes goes through the same path as the phone's own changes, with the same validation and sync cursor. A task an app adds reaches the phone at its next sync. Dates and times are given in the account's home time zone, and every due date says how it relates to today. A mistake in a call — an unknown category, a date that is not a date — comes back as a tool error with a sentence saying what to do instead, so the model can correct itself.

At a glance

TOOL	ACCESS	WHAT IT DOES
<code>get_overview</code>	read	Start here.
<code>list_tasks</code>	read	List tasks with filters: category, open/completed, due window (overdue, today, tomorrow, next 7 days, a date range, undated), text search across titles, notes and subtasks, minimum priority, and how they were captured.
<code>get_task</code>	read	Everything stored about one task, including its subtasks and, for a task captured by voice, the transcript of what was said and when.
<code>list_categories</code>	read	Every category (the app's worlds), Inbox first, with its colour and how many open, overdue, due-today and completed tasks it holds.
<code>create_task</code>	write	Add a task, optionally with a category, due date, priority, notes and a checklist of subtasks.
<code>update_task</code>	write	Change a task's title, notes, due date, priority or category.
<code>complete_tasks</code>	write	Mark one or more tasks done.
<code>reopen_tasks</code>	write	Mark one or more completed tasks as not done again.
<code>delete_tasks</code>	write destructive	Delete one or more tasks.
<code>edit_subtask</code>	write	Add, rename, check, uncheck or remove one subtask of a task.
<code>create_category</code>	write	Add a category with an optional colour.
<code>update_category</code>	write	Rename a category or change its colour.

TOOL	ACCESS	WHAT IT DOES
<code>delete_category</code>	<code>write</code> <code>destructive</code>	Delete a category AND every task in it, open or done.
<code>add_to_dictionary</code>	<code>write</code>	Add names, places, jargon or acronyms to the person's Dictionary so Capture spells them right when it transcribes what they say.

Tools

Overview `get_overview`

`read` idempotent: yes

Start here. The person's day at a glance: today's date and time zone, counts (open, overdue, due today, this week, done today), the overdue and due-today tasks, what's coming up in the next seven days, and every category.

ARGUMENTS

None.

RETURNS `OVERVIEW`

NAME	TYPE	NOTES
<code>account</code>	<u><code>AccountView</code></u>	
<code>counts</code>	<u><code>OverviewCounts</code></u>	
<code>overdue</code>	list of <u><code>TaskView</code></u>	
<code>due_today</code>	list of <u><code>TaskView</code></u>	
<code>upcoming</code>	list of <u><code>TaskView</code></u>	
<code>categories</code>	list of <u><code>CategoryView</code></u>	

Find tasks `list_tasks`

`read` idempotent: yes

List tasks with filters: category, open/completed, due window (overdue, today, tomorrow, next 7 days, a date range, undated), text search across titles, notes and subtasks, minimum priority, and how they were captured. Paged; every task includes its category, due date (in the person's time zone, with 'today' / 'in 3 days' / overdue), priority, notes and subtasks.

ARGUMENTS

NAME	TYPE		NOTES
<code>category</code>	text or null	optional	Only this category: its name or id. Omit for all categories.
<code>status</code>	<code>open</code> <code>completed</code> <code>all</code>	default <code>open</code>	open (default), completed, or all.
<code>due</code>	<code>any</code> <code>overdue</code> <code>today</code> <code>tomorrow</code> <code>next_7_days</code> <code>overdue_or_today</code> <code>no_date</code> <code>has_date</code>	default <code>any</code>	Filter by due date, in the person's calendar days: overdue (open and due before today), today, tomorrow, <code>next_7_days</code> (today through six days from now), <code>overdue_or_today</code> , <code>no_date</code> , <code>has_date</code> , or any (default).
<code>due_from</code>	date (YYYY-MM-DD) or null	optional	Due on or after this day (YYYY-MM-DD), inclusive.
<code>due_to</code>	date (YYYY-MM-DD) or null	optional	Due on or before this day (YYYY-MM-DD), inclusive.
<code>search</code>	text (up to 200 characters) or null	optional	Case-insensitive text to find in titles, notes and subtasks.
<code>min_priority</code>	<code>none</code> <code>low</code> <code>medium</code> <code>high</code> or null	optional	Only tasks at or above this priority: low, medium or high.
<code>source</code>	<code>voice</code> <code>manual</code> or null	optional	voice: captured by speaking. manual: typed.
<code>sort</code>	<code>due</code> <code>priority</code> <code>created</code> <code>updated</code> <code>completed</code>	default <code>due</code>	due (soonest first, undated last; default), priority (highest first), created or updated (newest first), completed (most recently completed first).
<code>limit</code>	integer 1–200	default <code>50</code>	Page size, 1-200. Default 50.
<code>offset</code>	integer ≥ 0	default <code>0</code>	Skip this many, for the next page.

RETURNS `TASKPAGE`

NAME	TYPE	NOTES
<code>tasks</code>	list of <code>TaskView</code>	
<code>total</code>	integer	
<code>offset</code>	integer	

NAME	TYPE	NOTES
has_more	true or false	
next_offset	integer or null	
timezone	text	
today	date (YYYY-MM-DD)	

Task details `get_task`

read idempotent: yes

Everything stored about one task, including its subtasks and, for a task captured by voice, the transcript of what was said and when.

ARGUMENTS

NAME	TYPE	NOTES
task_id	id required	The task's id, as returned by list_tasks or get_overview.

RETURNS `TASKRESULT`

NAME	TYPE	NOTES
task	<u>TaskDetail</u>	
note	text or null	

Categories `list_categories`

read idempotent: yes

Every category (the app's worlds), Inbox first, with its colour and how many open, overdue, due-today and completed tasks it holds.

ARGUMENTS

None.

RETURNS `CATEGORYLIST`

NAME	TYPE	NOTES
categories	list of <u>CategoryView</u>	

Add a task `create_task`

`write` idempotent: no

Add a task, optionally with a category, due date, priority, notes and a checklist of subtasks. It appears on the person's phone at the next sync.

ARGUMENTS

NAME	TYPE		NOTES
<code>title</code>	text (1–500 characters)	required	What needs doing, short.
<code>category</code>	text or null	optional	Category name or id. Omit to file it in the Inbox.
<code>notes</code>	text (up to 4000 characters) or null	optional	Longer detail.
<code>due</code>	text or null	optional	When it's due. YYYY-MM-DD means 09:00 that day in the person's time zone; YYYY-MM-DDTHH:MM is that local time; a timestamp with an offset is exact. 'today' and 'tomorrow' also work.
<code>priority</code>	<code>none</code> <code>low</code> <code>medium</code> <code>high</code>	default <code>none</code>	none, low, medium or high.
<code>subtasks</code>	list of text (up to 100 items)	optional	Checklist items to add under the task, in order.
<code>repeat</code>	<code>RepeatArgs</code> or null	optional	Make it repeat; completing it adds the next one. Give it a due date.

where `RepeatArgs` is

The app's repeat rule (TL-18): no "2nd Tuesday", end dates or counts.

NAME	TYPE		NOTES
<code>freq</code>	<code>daily</code> <code>weekly</code> <code>monthly</code> <code>yearly</code>	required	
<code>interval</code>	integer 1–99	default <code>1</code>	Every N days/weeks/months/years.
<code>weekdays</code>	list of <code>mon</code> <code>tue</code> <code>wed</code> <code>thu</code> <code>fri</code> <code>sat</code> <code>sun</code> (1–7 items) or null	optional	Weekly only: which days, e.g. ['mon', 'wed']. Every weekday: mon-fri.
<code>month_day</code>	integer 1–31 or null	optional	Monthly only: day of the month (31 = last day).

RETURNS `TASKRESULT`

NAME	TYPE	NOTES
<code>task</code>	<code>TaskDetail</code>	
<code>note</code>	text or null	

Edit a task `update_task`

`write` idempotent: yes

Change a task's title, notes, due date, priority or category. Only the fields you pass change; pass null for notes or due to clear them.

ARGUMENTS

NAME	TYPE		NOTES
<code>task_id</code>	id	required	The task's id, as returned by <code>list_tasks</code> or <code>get_overview</code> .
<code>title</code>	text (1–500 characters) or null	optional	
<code>notes</code>	text (up to 4000 characters) or null	optional	New notes. Send null to clear them.
<code>due</code>	text or null	optional	New due date (same formats as <code>create_task</code>). Send null to remove it.
<code>priority</code>	<code>none</code> <code>low</code> <code>medium</code> <code>high</code> or null	optional	
<code>category</code>	text or null	optional	Move it to this category.
<code>repeat</code>	<code>RepeatArgs</code> or null	optional	A new repeat rule. Send null to stop it repeating.

where `RepeatArgs` is

The app's repeat rule (TL-18): no "2nd Tuesday", end dates or counts.

NAME	TYPE		NOTES
<code>freq</code>	<code>daily</code> <code>weekly</code> <code>monthly</code> <code>yearly</code>	required	
<code>interval</code>	integer 1–99	default <code>1</code>	Every N days/weeks/months/years.
<code>weekdays</code>	list of <code>mon</code> <code>tue</code> <code>wed</code> <code>thu</code> <code>fri</code> <code>sat</code> <code>sun</code> (1–7 items) or null	optional	Weekly only: which days, e.g. ['mon', 'wed']. Every weekday: mon-fri.
<code>month_day</code>	integer 1–31 or null	optional	Monthly only: day of the month (31 = last day).

RETURNS `TASKRESULT`

NAME	TYPE	NOTES
<code>task</code>	<code>TaskDetail</code>	
<code>note</code>	text or null	

Complete tasks `complete_tasks`

`write` idempotent: yes

Mark one or more tasks done. Already-completed tasks are left as they are.

ARGUMENTS

NAME	TYPE		NOTES
<code>task_ids</code>	list of id (1–100 items)	required	One or more task ids.

RETURNS `BULKRESULT`

NAME	TYPE	NOTES
<code>changed</code>	integer	
<code>results</code>	list of BulkOutcome	

Reopen tasks `reopen_tasks`

`write` idempotent: yes

Mark one or more completed tasks as not done again.

ARGUMENTS

NAME	TYPE		NOTES
<code>task_ids</code>	list of id (1–100 items)	required	One or more task ids.

RETURNS `BULKRESULT`

NAME	TYPE	NOTES
<code>changed</code>	integer	
<code>results</code>	list of BulkOutcome	

Delete tasks `delete_tasks`

`write` `destructive` idempotent: yes

Delete one or more tasks. Confirm with the person first.

ARGUMENTS

NAME	TYPE		NOTES
<code>task_ids</code>	list of id (1–100 items)	required	One or more task ids.

RETURNS `BULKRESULT`

NAME	TYPE	NOTES
<code>changed</code>	integer	
<code>results</code>	list of BulkOutcome	

Edit a checklist item `edit_subtask`

`write` idempotent: no

Add, rename, check, uncheck or remove one subtask of a task.

ARGUMENTS

NAME	TYPE		NOTES
<code>task_id</code>	id	required	The task's id, as returned by <code>list_tasks</code> or <code>get_overview</code> .
<code>action</code>	<code>add</code> <code>rename</code> <code>check</code> <code>uncheck</code> <code>remove</code>	required	add a new item, rename one, check or uncheck it, or remove it.
<code>subtask_id</code>	id or null	optional	The item's id (from <code>get_task</code>). Needed for all but add.
<code>title</code>	text (1–500 characters) or null	optional	For add and rename.

RETURNS `TASKRESULT`

NAME	TYPE	NOTES
<code>task</code>	<u><code>TaskDetail</code></u>	
<code>note</code>	text or null	

Add a category `create_category`

`write` idempotent: no

Add a category with an optional colour.

ARGUMENTS

NAME	TYPE		NOTES
<code>name</code>	text (1–200 characters)	required	
<code>color</code>	<code>blue</code> <code>sage</code> <code>periwinkle</code> <code>slate</code> <code>mauve</code> <code>olive</code> or null	optional	blue, sage, periwinkle, slate, mauve or olive. Omit for the next unused.

RETURNS `CATEGORYRESULT`

NAME	TYPE	NOTES
<code>category</code>	<u><code>CategoryView</code></u>	

Edit a category `update_category`

`write` idempotent: yes

Rename a category or change its colour.

ARGUMENTS

NAME	TYPE		NOTES
<code>category</code>	text	required	A category's name (any case; an unambiguous start of the name works) or its id.
<code>name</code>	text (1–200 characters) or null	optional	New name.
<code>color</code>	<code>blue</code> <code>sage</code> <code>periwinkle</code> <code>slate</code> <code>mauve</code> <code>olive</code> or null	optional	

RETURNS `CATEGORYRESULT`

NAME	TYPE	NOTES
<code>category</code>	<u>CategoryView</u>	

Delete a category `delete_category`

`write` `destructive` idempotent: yes

Delete a category AND every task in it, open or done. The Inbox cannot be deleted. Always confirm with the person first, saying how many tasks go with it.

ARGUMENTS

NAME	TYPE		NOTES
<code>category</code>	text	required	A category's name (any case; an unambiguous start of the name works) or its id.

RETURNS `CATEGORYDELETED`

NAME	TYPE	NOTES
<code>deleted</code>	<u>CategoryRef</u>	
<code>tasks_deleted</code>	integer	

Add to Dictionary `add_to_dictionary`

`write` idempotent: yes

Add names, places, jargon or acronyms to the person's Dictionary so Capture spells them right when it transcribes what they say. Give each word as it should be spelled, optionally with what Capture wrongly types for it. A word already there is not added twice: new misspellings are added to it, and it is switched back on if it was off.

ARGUMENTS

NAME	TYPE		NOTES
<code>words</code>	list of <code>DictionaryWordArgs</code> (1–50 items)	required	One or more words to add.

where `DictionaryWordArgs` is

NAME	TYPE		NOTES
<code>word</code>	text (1–200 characters)	required	The word or phrase spelled the way it should be, e.g. 'Anushka'.
<code>misspellings</code>	list of text (up to 50 items)	optional	What Capture wrongly types for it, e.g. ['Anoushka']. Optional: Capture replaces these with the word when it tidies a transcript.
<code>kind</code>	<code>person</code> <code>project</code> <code>product</code> <code>acronym</code> <code>other</code>	default <code>other</code>	person, project, product, acronym or other (default). The app groups by it.

RETURNS `DICTIONARYRESULT`

NAME	TYPE	NOTES
<code>added</code>	integer	
<code>results</code>	list of <code>DictionaryOutcome</code>	

Prompts

Ready-made conversations a client offers as shortcuts (for example `/mcp__capture__plan_my_day` in Claude Code). Each is only words; the work is done by the tools.

PROMPT	TITLE	WHAT IT ASKS FOR	ARGUMENTS
<code>plan_my_day</code>	Plan my day	What's overdue, what's due today, and a realistic order to do it in.	<code>focus</code> — Anything to prioritise or leave out today (optional).
<code>weekly_review</code>	Weekly review	What got done, what slipped, and what the next seven days hold.	none
<code>tidy_inbox</code>	Tidy my Inbox	File each Inbox task into the right category, with dates where obvious.	none
<code>add_from_text</code>	Add tasks from text	Turn notes, an email or a list into Capture tasks.	<code>text</code> (required) — The notes, email or list to turn into tasks.

What the assistant is told

The instructions every client receives at `initialize`.

Capture is the person's voice-first to-do list. You can see (and, if this connection allows changes, edit) their real tasks, which sync to their iPhone.

- Tasks live in categories. Every account has an Inbox, where uncategorised tasks go.
- Due dates and times are in the person's home time zone, and every task says how its due day relates to today ("today", "tomorrow", "in 3 days", overdue). Overdue means open and due on a day before today. A due date given as a day only is stored at 09:00.
- Priority is none, low, medium or high.
- Tasks captured by voice carry the transcript of what was said (`get_task`).
- The Dictionary holds names and words Capture should spell right when it transcribes; `add_to_dictionary` adds to it.

Start with `get_overview`. Refer to tasks by title when talking to the person, not by id. Before deleting anything, or changing more than a few tasks at once, say what you will do and wait for their OK.

Shapes

The objects the tools return, referenced from the tables above.

AccountView		
NAME	TYPE	NOTES
<code>name</code>	text or null	
<code>timezone</code>	text	
<code>today</code>	date (YYYY-MM-DD)	
<code>weekday</code>	text	
<code>now</code>	text	
<code>access</code>	<code>read</code> <code>read_write</code>	

BulkOutcome		
NAME	TYPE	NOTES
<code>task_id</code>	id	
<code>title</code>	text or null	
<code>outcome</code>	text	

CaptureView

The recording a voice task came from: what was said, and when.

NAME	TYPE	NOTES
id	id	
recorded_at	text or null	
duration_seconds	number or null	
transcript	text or null	
status	text	

CategoryRef

NAME	TYPE	NOTES
id	id	
name	text	

CategoryView

NAME	TYPE	NOTES
id	id	
name	text	
color	ColorView	
is_inbox	true or false	
sort_order	integer	
open_tasks	integer	
overdue_tasks	integer	
due_today	integer	
completed_tasks	integer	
created_at	text	

ColorView

NAME	TYPE	NOTES
index	integer	
name	text	
hex	text	

DictionaryEntry

NAME	TYPE	NOTES
id	id	
word	text	
misspellings	list of text	
kind	text	
used_in_transcription	true or false	

DictionaryOutcome

NAME	TYPE	NOTES
word	text	
outcome	text	
entry	<u>DictionaryEntry</u> or null	

DueView

NAME	TYPE	NOTES
at	text	
date	date (YYYY-MM-DD)	
time	text	
weekday	text	
relative	text	
days_from_today	integer	
overdue	true or false	

OverviewCounts

NAME	TYPE	NOTES
open	integer	
overdue	integer	
due_today	integer	
due_tomorrow	integer	
due_next_7_days	integer	
no_due_date	integer	
completed_today	integer	
completed_last_7_days	integer	

RepeatView

A repeating task's rule (TL-18). Completing it adds the next one.

NAME	TYPE	NOTES
freq	text	
interval	integer	
weekdays	list of text or null	
month_day	integer or null	
summary	text	

SubtaskView

NAME	TYPE	NOTES
id	id	
title	text	
completed	true or false	

TaskDetail

NAME	TYPE	NOTES
id	id	
title	text	
notes	text or null	
category	<u>CategoryRef</u>	
status	open completed	
completed_at	text or null	
due	<u>DueView</u> or null	
repeat	<u>RepeatView</u> or null	
priority	text	
priority_level	integer	
subtasks	list of <u>SubtaskView</u>	
subtasks_done	integer	
subtasks_total	integer	
source	text	
capture_id	id or null	
created_at	text	
updated_at	text	
capture	<u>CaptureView</u> or null	

TaskView

NAME	TYPE	NOTES
id	id	
title	text	
notes	text or null	
category	<u>CategoryRef</u>	
status	open completed	
completed_at	text or null	
due	<u>DueView</u> or null	
repeat	<u>RepeatView</u> or null	
priority	text	
priority_level	integer	
subtasks	list of <u>SubtaskView</u>	
subtasks_done	integer	
subtasks_total	integer	
source	text	
capture_id	id or null	
created_at	text	
updated_at	text	